Lecture Notes in Computer Science 4404

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Simeon J. Simoff Michael H. Böhlen
Arturas Mazeika (Eds.)

Visual Data Mining

Theory, Techniques and Tools for Visual Analytics

Springer

Volume Editors

Simeon J. Simoff
University of Western Sydney
School of Computing and Mathematics
NSW 1797, Australia
E-mail: s.simoff@uws.edu.au

Michael H. Böhlen
Arturas Mazeika
Free University of Bozen-Bolzano
Faculty of Computer Science
Dominikanerplatz 3, 39100 Bozen-Bolzano, Italy
E-mail: {boehlen,arturas}@inf.unibz.it

Library of Congress Control Number: 2008931578

CR Subject Classification (1998): H.2.8, I.3, H.5

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

ISSN 0302-9743
ISBN-10 3-540-71079-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-71079-0 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12280612 06/3180 5 4 3 2 1 0

Foreword

Visual Data Mining—Opening the Black Box

Knowledge discovery holds the promise of insight into large, otherwise opaque datasets. The nature of what makes a rule interesting to a user has been discussed widely[1] but most agree that it is a subjective quality based on the practical usefulness of the information. Being subjective, the user needs to provide feedback to the system and, as is the case for all systems, the sooner the feedback is given the quicker it can influence the behavior of the system.

There have been some impressive research activities over the past few years but the question to be asked is why is visual data mining only now being investigated commercially? Certainly, there have been arguments for visual data mining for a number of years – Ankerst and others[2] argued in 2002 that current (autonomous and opaque) analysis techniques are inefficient, as they fail to directly embed the user in dataset exploration and that a better solution involves the user and algorithm being more tightly coupled. Grinstein stated that the "*current state of the art data mining tools are automated, but the perfect data mining tool is interactive and highly participatory*," while Han has suggested that the "*data selection and viewing of mining results should be fully interactive, the mining process should be more interactive than the current state of the art and embedded applications should be fairly automated*[2]." A good survey on techniques until 2003 was published by de Oliveira and Levkowitz[3].

However, the deployment of visual data mining (VDM) techniques in commercial products remains low. There are, perhaps, four reasons for this. First, VDM, as a strong sub-discipline of data mining only really started around 2001. Certainly there was important research before then but as an identifiable subcommunity of data mining, the area coalesced around 2001. Second, while things move fast in IT, VDM represents a shift in thinking away from a discipline that itself has yet to settle down commercially. Third, to fully contribute to VDM a researcher/systems developer must be proficient in both data mining and visualization. Since both of these are still developing themselves, the pool from which to find competent VDM researchers and developers is small. Finally, if the embedding is to be done properly, the overarching architecture of the knowledge

[1] Geng, L., Hamilton, H.J.: Interestingness measures for data mining: A survey. ACM Computing Surveys 38 (2006)

[2] Ankerst, M.: The perfect data mining tool: Automated or interactive? In: Panel at ACM SIGKDD 2002, Edmonton, Canada. ACM, New York (2002)

[3] de Oliveira, M.C.F., Levkowitz, H.: From visual data exploration to visual data mining: a survey. IEEE Transactions on Visualization and Computer Graphics 9, 378–394 (2003)

discovery process must be changed. The iterative paradigm of *mine and visualize* must be replaced with the data mining equivalent of direct manipulation[4].

Embedding the user within the discovery process, by, for example, enabling the user to change the mining constraints, results in a finer-grained framework as the interaction between user and system now occurs *during* analysis instead of *between* analysis runs. This overcomes the computer's inability to incorporate evolving knowledge regarding the problem domain and user objectives, not only facilitating the production of a higher quality model, but also reducing analysis time for two reasons. First, the guidance reduces the search space at an earlier stage by discarding areas that are not of interest. Second, it reduces the number of iterations required. It also, through the Hawthorn Effect, has the effect of improving the user's confidence in, and ownership of, the results that are produced[5].

While so-called *guided* data mining methods have been produced for a number of data mining areas including clustering[6], association mining[4,7], and classification[8], there is an architectural aspect to guided data mining, and to VDM in general, that has not been adequately explored and which represents an area for future work.

Another area of future work for the VDM community is quantification. Although the benefits that VDM can provide are clear to us, due to its subjective nature, the benefits of this synergy are not easily quantified and thus may not be as obvious to others. VDM methods can be more time-consuming to develop and thus for VDM to be accepted more widely we must find methods of showing that VDM demonstrates either (or both of) a time improvement or a quality improvement over non-visual methods.

This book has been long awaited. The VDM community has come a long way in a short time. Due to its ability to merge the cognitive ability and contextual awareness of humans with the increasing computational power of data mining systems, VDM is undoubtedly not just a future trend but destined to be one of the main themes for data mining for many years to come.

April 2008 John F. Roddick

[4] Ceglar, A., Roddick, J.F.: GAM - a guidance enabled association mining environment. International Journal of Business Intelligence and Data Mining 2, 3–28 (2007)

[5] Ceglar, A.: Guided Association Mining through Dynamic Constraint Refinement. PhD thesis, Flinders University (2005)

[6] Anderson, D., Anderson, E., Lesh, N., Marks, J., Perlin, K., Ratajczak, D., Ryall, K.: Human guided simple search: combining information visualization and heuristic search. In: Workshop on new paradigms in information visualization and manipulation; In conjunction with the 8th ACM international conference on Information and Knowledge Management, Kansas City, MO, pp. 21–25. ACM Press, New York (2000)

[7] Ng, R., Lakshmanan, L., Han, J., Pang, A.: Exploratory mining and pruning optimizations of constrained association rules. In: 17th ACM SIGACT-SIGMOD-SIGART Symposium on the Principles of Database Systems, Seattle, WA, pp. 13–24. ACM Press, New York (1998)

[8] Ankerst, M., Ester, M., Kriegel, H.P.: Towards an effective cooperation of the user and the computer for classification. In: 6th International Conference on Knowledge Discovery and Data Mining (KDD 2000), Boston, MA, pp. 179–188 (2000)

Preface

John W. Tukey, who made unparalleled contributions to statistics and to science in general during his long career at Bell Labs and Princeton University, emphasized that seeing may be believing or disbelieving, but above all, data analysis involves visual, as well as statistical, understanding. Certainly one of the oldest visual explanations in mathematics is the visual proof of the Pythagorean theorem. The proof, impressive in its brevity and elegance, stresses the power of an interactive visual representation in facilitating our analytical thought processes. Thus, visual reasoning approaches to extracting and comprehension of the information encoded in data sets became the focus of what is called *visual data mining*. The field emerged from the integration of concepts from numerous fields, including computer graphics, visualization metaphors and methods, information and scientific data visualization, visual perception, cognitive psychology, diagrammatic reasoning, 3D virtual reality systems, multimedia and design computing, data mining and online analytical processing, very large databases last, and even collaborative virtual environments.

The importance of the field had already been recognized in the beginning of the decade. This was reflected in the series of visual data mining workshops, conducted at the major international conferences devoted to data mining. Later, the conferences and periodicals in information visualization paid substantial attention to some developments in the field. Commercial tools and the work in several advanced laboratories and research groups across the globe provided working environments for experimenting not only with different methods and techniques for facilitating the human visual system in examination and patterns discovery, and understanding of patterns among massive volumes of multi-dimensional and multi-source data, but also for testing techniques that provide robust and statistically valid visual patterns. It was not until a panel of more than two dozen internationally renowned individuals was assembled, in order to address the shortcomings and drawbacks of the current state of visual information processing, that the need for a systematic and methodological development of visual analytics was placed in the top priorities on the research and development agenda in 2005.

This book aims at addressing this need. Through a collection of 21 chapters selected from more than 46 submissions, it offers a systematic presentation of the state of the art in the field, presenting it in the context of visual analytics. Since visual analysis is such a different technique, it is an extremely significant topic for contemporary data mining and data analysis.

The editors would like to thank all the authors for their contribution to the volume and their patience in addressing reviewers' and editorial feedback. Without their contribution and support the creation of this volume would have been impossible. The editors would like to thank the reviewers for their thorough reviews and detailed comments.

Special thanks go to John Roddick, who, on short notice, kindly accepted the invitation to write the Foreword to the book.

April 2008

Simeon J. Simoff
Michael Böhlen
Arturas Mazeika

Table of Contents

Part 3 – Tools and Applications

Visual Data Mining: An Introduction and Overview

Simeon J. Simoff[1,2], Michael H. Böhlen[3], and Arturas Mazeika[3]

[1] School of Computing and Mathematics, College of Heath and Science
University of Western Sydney,
NSW 1797, Australia
s.simoff@uws.edu.au
[2] Faculty of Information Technology, University of Technology, Sydney
PO Box 123 Broadway NSW 2007 Australia
[3] Faculty of Computer Science, Free University of Bolzano-Bozen, Italy
{arturas, boehlen}@inf.unibz.it

1 Introduction

In our everyday life we interact with various information media, which present us with facts and opinions, supported with some evidence, based, usually, on condensed information extracted from data. It is common to communicate such condensed information in a visual form – a static or animated, preferably interactive, visualisation. For example, when we watch familiar weather programs on the TV, landscapes with cloud, rain and sun icons and numbers next to them quickly allow us to build a picture about the predicted weather pattern in a region. Playing sequences of such visualisations will easily communicate the dynamics of the weather pattern, based on the large amount of data collected by many thousands of climate sensors and monitors scattered across the globe and on weather satellites. These pictures are fine when one watches the weather on Friday to plan what to do on Sunday – after all if the patterns are wrong there are always alternative ways of enjoying a holiday. Professional decision making would be a rather different scenario. It will require weather forecasts at a high level of granularity and precision, and in real-time. Such requirements translate into requirements for high volume data collection, processing, mining, modelling and communicating the models quickly to the decision makers. Further, the requirements translate into high-performance computing with integrated efficient interactive visualisation. From practical point of view, if a weather pattern can not be depicted fast enough, then it has no value. Recognising the power of the human visual perception system and pattern recognition skills adds another twist to the requirements – data manipulations need to be completed at least an order of magnitude faster than real-time change in data in order to combine them with a variety of highly interactive visualisations, allowing easy remapping of data attributes to the features of the visual metaphor, used to present the data. In this few steps in the weather domain, we have specified some requirements towards a visual data mining system.

2 The Term

As a term visual data mining has been around for nearly a decade. There is some variety in what different research groups understand under this label. “The goal of

S.J. Simoff et al. (Eds.): Visual Data Mining, LNCS 4404, pp. 1–12, 2008.

visual data mining is to help a user to get a feeling for the data, to detect interesting knowledge, and to gain a deep visual understanding of the data set" [1]. Niggemann [2] looked at visual data mining as visual presentation of the data close to the mental model. As humans understand information by forming a mental model which captures only a gist of the information, then a data visualisation close to the mental model can reveal hidden information encoded in that data [2]. Though difficult to measure, such closeness is important, taking in account that visualisation algorithms map data sets that usually lack inherent 2D and 3D semantics onto a physical display (for example, 2D screen space or 3D virtual reality platform). Ankerst [3], in addition to the role of the visual data representation, explored the relation between visualisation and the data mining and knowledge discovery (KDD) process, and defined visual data mining as "a step in the KDD process that utilizes visualisation as a communication channel between the computer and the user to produce novel and interpretable patterns." Ankerst [3] also explored three different approaches to visual data mining, two of which are connected with the visualisation of final or intermediate results and the third one involves interactive manipulation of the visual representation of the data, rather than the results of the algorithm.

The three definitions recognise that visual data mining relies heavily on human visual processing channel, and utilises human cognition. The three definitions also emphasise, respectively, the key importance of the following three aspects of visual data mining: (i) tasks; (ii) visual representation (visualisation); and (iii) the process.

Our work definition looks at visual data mining as the process of interaction and analytical reasoning with one or more visual representations of an abstract data that leads to the visual discovery of robust patterns in these data that form the information and knowledge utilised in informed decision making. The abstract data can be the original data set or/and some output of data mining algorithm(s).

3 The Process

Fig. 1 illustrates a visual data mining process that corresponds to our definition. The visual processing pipeline is central to the process. Each step of this pipeline involves interaction with the human analyst, indicated by the bi-directional links that connect each step in the process and the human analyst. These links indicate that all the iterative loops in the process close via the human analyst. In some cases, data mining algorithms can be used to assist the process. Data mining algorithm(s) can be applied to the data: (a) before any visualisation has been considered, and (b) after some visual interaction with the data. In the first case, any of the output (intermediate and final) of the data mining algorithm can be included in the visual processing pipeline, either on its own, or together with visualisation of the original data. For example, Fig. 2 illustrates the case when the output of a data mining algorithm, in this case an association rule mining algorithm, is visualised, visually processed and the result then is fed into the visual processing pipeline (in this example we have adapted the interactive mosaic plots visualisation technique for association rules [4]). In another iteration, the analyst can take in account the visualisation of the output of the data mining algorithm when interacting with raw data visualisation [5] or explore the association rule set using another visual representation [6].

Central to the "Analytical reasoning" step in Fig. 1 is the sense-making process [7]. The process is not a straight-forward progression but has several iterative steps that are not shown in Fig. 1.

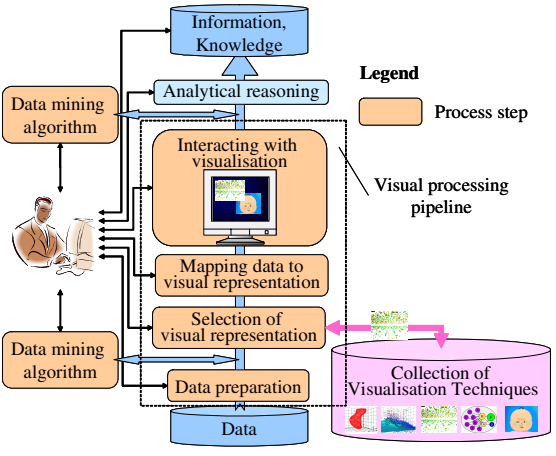


Fig. 1. Visual data mining as a human-centred interactive analytical and discovery process

As the visual data mining process relies on visualisation and the interaction with it, the success of the process depends on the breadth of the collection of visualisation techniques (see Fig. 1), on the consistency of the design of these visualisations, the ability to remap interactively the data attributes to visualisation attributes, the set of functions for interacting with the visualisation and the capabilities that these functions offer in support of the reasoning process. In Fig. 1 the "Collection of Visualisation Techniques" consists of graphical representations for data sets coupled with user interface techniques to operate with each representation in search of patterns. This coupling has been recognised since the early days of the field (for instance, see the work done in Bell Laboratories/Lucent Technologies [8] for an example of two graphical representations for the area of mining of telecommunications data: one for calling communities and the other for showing individual phone calls, and the corresponding interfaces that were successfully applied for telecommunications fraud detection through visual data mining). Keim [9] emphasised further the links between information visualisation and the interactivity of the visual representation in terms of visual data mining, introducing a classification relating the two areas, based on the data type to be visualised, the visualisation technique and the interaction and distortion technique. Though interaction has been recognised, there is a strong demand on the development of interactive visualisations, which are fundamentally different from static visualisations. Designed with the foundations of perceptual and cognitive theory in mind and focused on supporting the processes and methods in visual data mining, these visual data representations are expected to be relevant to the visual data mining tasks and effective in terms of achieving the data mining goals.

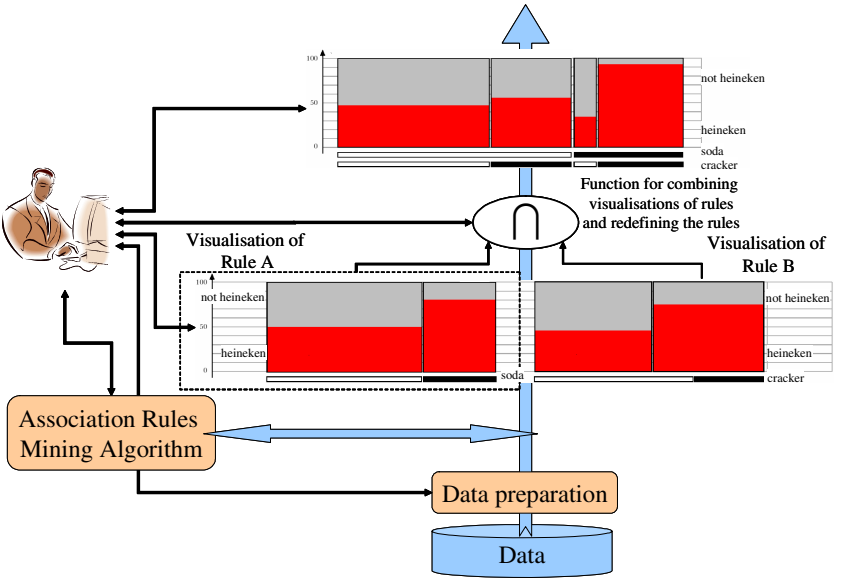


Fig. 2. Interaction with the visual representation of the output of an association rule mining algorithm as part of the visual reasoning process (the visualisation technique has been adapted from [4])

The separation of the visual analysis and visual results presentation tasks in visual data mining is reflected in recent visual data mining tools, for instance, Miner3D[1], NetMap[2] and NetMiner[3], which include two type of interfaces: (a) visual data mining interface for manipulating visualisations in order to facilitate human visual pattern recognition skills; and (b) visual presentation interface for generating visual (static and animated) reports and presentations of the results, to facilitate human communication and comprehension of discovered information and knowledge.

As the ultimate goal is that the visual data mining result is robust and not accidental, visualisation techniques may be coupled with additional statistical techniques that lead to data visualisations that more accurately represent the underlying properties of the data. For example, the VizRank approach [10] ranks and selects two-dimensional projections of class-labelled data that are expected to be the better ones for visual analysis. These projections (out of the numerous possible ones in multi-dimensional large data sets) then are recommended for visual data mining.

4 The Book

Visual data mining has not been presented extensively in a single volume of works. Westphal and Blaxton [11] present an extensive overview of various data and

[1] http://www.miner3d.com/
[2] http://www.netmap.com.au/
[3] http://www.visualanalytics.com/

information visualisation tools and their visual displays in the context of visual data mining. The comprehensive survey by Oliveira and Levkowitz [12] of various visualisation techniques and what actually can be done with them beyond just navigating through the abstract representation of data, places a convincing case for the need for tighter coupling of data and information visualisation with the data mining strategies and analytical reasoning. During 2001-2003 there had been a series of workshops on visual data mining [9, 13-15] with proceedings focused on the state-of-the-art in the area and the research in progress. Soukup and Davidson's monograph on visual data mining [16] has taken a business perspective and practice-based approach to the process and application of visual data mining, with a limitted coverage of visual data representations and interactive techniques.

This book aims at filling the gap between the initiative started with the workshops and the business perspectives [11, 16]. The collection of chapters from leading researchers in the field presents the state-of-the-art developments in visual data mining and places them in the context of visual analytics. The volume is solely devoted to the research and developments in the field.

The book is divided into three main parts: Part 1 – Theory and Methodologies, Part 2 – Techniques, and Part 3 – Tools and Applications.

Part 1 includes five chapters that present different aspects of theoretical underpinnings, frameworks and methodologies for visual data mining. Chapter "The 3DVDM Approach: A Case Study with Clickstream Data" introduces an approach to visual data mining, whose development started as part of an interdisciplinary project at Aalborg University, Denmark in the late 90s. The 3DVDM project explored the development and potential of immersive visual data mining technology in 3D virtual reality environments. This chapter explores the methodology of the 3DVDM approach and presents one of the techniques in its collection – state-of-the-art interaction and analysis techniques for exploring 3D data visualisation at different levels of granularity, using density surfaces. The state-of-the-art is demonstrated with the analysis of clickstream data – a challenging task for visual data mining, taking into account the amount and categorical nature of the data set.

Chapter "Form-Semantics-Function – A Framework for Designing Visual Data Representations for Visual Data Mining" addresses the issue of designing consistent visual data representations, as they are a key component in the visual data mining process [17, 18]. [16] aligned visual representations with some visual data mining tasks. It is acknowledged that in the current state of the development it is a challenging activity to find out the methods, techniques and corresponding tools that support visual mining of a particular type of information. The chapter presents an approach for comparison of visualisation techniques across different designs.

Association rules are one of the oldest models generated by data mining algorithms – a model that has been applied in many domains. One of the common problems of the association rule mining algorithms is that they generate a large amount of frequent itemsets and, then, association rules, which are still difficult to comprehend due to their large quantity. The book includes several chapters providing methodologies, techniques and tools for visual analysis of sets of association rules.

In chapter "A Methodology for exploring visual association models" Alipio Jorge, João Poças and Paulo Azevedo present a methodology for visual representation of association rule models and interaction with the model. As association rule mining

algorithms can produce a large amount of rules this approach enables visual exploration and to some extent mining of large sets of association rules. The approach includes strategies for separation of the rules into subsets and means for manipulating and exploring these subsets. The chapter provides practical solutions supporting the visual analysis of association rules.

In chapter "Visual exploration of frequent itemsets and association rules" Li Yang introduces a visual representation of frequent itemsets and association rules based on the adaptation of the popular parallel coordinates technique for visualising multidimensional data [19]. The analyst has control on what is visualised and what is not and by varying the border can explore patterns among the frequent itemsets and association rules.

In chapter "Visual analytics: Scope and challenges" Daniel Keim, Florian Mansmann, Jörn Schneidewind, Jim Thomas and Hartmut Ziegler present an overview of the context in which visual data mining is presented in this book – the field of visual analytics. Visual data mining enables discovering of regularities in data sets visually or regularities in the output of data mining algorithms, as above discussed. Visual decision making produces decisions that rely heavily on visual data mining, but useful regularities are only a part of the entire decision-making process (see chapters 1-3 in [20] for detailed analysis of the relations between information visualisation, visual data mining and visual decision making). With respect to visual analytics, visual data mining offers powerful techniques for extraction of visual patterns. Visual analytics adds on top of it a variety of analytical techniques that look for making sense and discoveries from the visual output of the visual data mining systems. Visual data mining systems that support visual analytics combine a collection of information visualisation metaphors and techniques with aspects of data mining, statistics and predictive modelling.

Part 2 of the book groups chapters that present different techniques for visual data mining. Some of them are focused on revealing visual structures in the data, others on the output of data mining algorithms (association rules). The later techniques can be viewed as "visual extensions" of the data mining algorithms. Another interesting trend in this part is the tight coupling of interactive visualisations with conventional data mining algorithms which leads to advanced visual data mining and analytics solutions.

Chapter "Using nested surfaces for visual detection of structures in databases" presents a technique for facilitating the detection of less obvious or weakly expressed visual structures in the data by equalising the presence of the more and less pronounced structures in a data set. The chapter presents the technical details of the algorithms. Though, the practical realisation of the work is part of the 3DVDM environment, the algorithms are valid for any visual data mining system utilising 3D virtual reality environments.

In chapter "Visual mining of association rules" Dario Bruzzese and Cristina Davino present a framework for utilising various visualisation techniques to represent association rules and then to provide visual analysis of these rules. The chapter promotes the smorgasbord approach to employ a variety of visual techniques to unveil association structures and visually identify within these structures the rules that are relevant and meaningful in the context of the analysis.

In chapter "Interactive decision tree construction for interval and taxonomical data" François Poulet and Thanh-Nghi Do provide a framework for extending conventional decision tree-building algorithms into visual interactive decision tree classifier builders. The authors demonstrate it on the modification of two decision tree inducing algorithms. The approach offers the benefit of incorporating the statistical underpinning mechanisms for guiding the classifier induction with the interactive visual intervention and manipulation capabilities, which allow the deep visual exploration of the data structures and relating those structures to specific statistical characteristics of the patterns.

Support vector machines (SVMs) are another popular classifier. The winners in Task 2 in the KDD Cup 2002[4], Adam Kowalczyk and Bhavani Raskutti, applied a proprietary SVM algorithm. In chapter "Visual Methods for Examining SVM Classifiers" Doina Caragea, Dianne Cook, Hadley Wickham and Vasant Honavar look at coupling data visualisation with an SVM algorithm in order to obtain some knowledge about the data that can be used to select data attributes and parameters of the SVM algorithm. The visual technique complements conventional cross-validation method. The last chapter of the book - François Poulet's "Towards effective visual data mining with cooperative approaches", also considers the interaction between the SVM technique and visualisations, though as part of a broader philosophy of integrated visual data mining tools.

In Chapter "Text visualization for visual text analytics" John Risch, Anne Kao, Stephen Poteet and Jason Wu explore visual text analytics techniques and tools. Text analytics couples semantic mapping techniques with visualisation techniques to enable interactive analysis of semantic structures enriched with other information encoded in the text data. The strength of the techniques is in enabling visual analysis of complex multidimensional relationship patterns within the text collection. Techniques, discussed in the chapter, enable human sense making, comprehension and analytical reasoning about the contents of large and complexly related text collections, without necessarily reading them.

Chapter "Visual discovery of network patterns of interaction between attributes" presents a methodology, collection of techniques and a corresponding visual data mining tool which enables visual discovery of network patterns in data, in particular, patterns of interaction between attributes, which usually are assumed to be independent in the paradigm of conventional data mining methods. Techniques that identify network structures within the data are getting an increasing attention, as they attempt to uncover linkages between the entities and their properties, described by the data set. The chapter presents a human-centred visual data mining methodology and illustrates the details with two case studies – fraud detection in insurance industry and Internet traffic analysis.

In chapter "Mining patterns for visual interpretation in a multiple views environment" José Rodrigues Jr., Agma Traina and Caetano Traina Jr. address the issue of semantically consistent integration of multiple visual representations and their interactive techniques. The chapter presents a framework which integrates three techniques and their workspaces according to the analytical decisions made by the analyst. The analyst can identify visual patterns when analysing in parallel multiple subsets of the

[4] http://www.biostat.wisc.edu/~craven/kddcup/winners.html

data and cross link these patterns in order to facilitate the discovery of patterns in the whole data set.

Chapter "Using 2D hierarchical heavy hitters to investigate binary relationships" presents an original technique for identification of hierarchical heavy hitters (HHH). The concept of hierarchical heavy hitters is very useful in identifying dominant or unusual traffic patterns. In terms of the Internet traffic, a heavy hitter is an entity which accounts for at least a specified proportion of the total activity on the network, measured in terms of number of packets, bytes, connections etc. A "heavy hitter" can be an individual flow or connection. Hierarchical accounts for the possible aggregation of multiple flows/connections that share some common property, but which themselves may not necessarily be heavy hitters (similar to hierarchical frequent itemsets). The chapter presents visual data mining technique which addresses the space complexity of HHHs in multidimensional data streams [21]. The tool utilises the 3DVDM engine discussed in details in Part 3.

Chapter "Supporting visual data mining of time dependent data through data sonification" explores the potential integration of the visual and audio data representation. It presents the results of experimental study of different sonification strategies for 2D and 3D time series. As sound is a function of time, intuitively sonification seems to be suitable extension for mining time series and other sequence data. Following a human-computer interaction approach the study looks at the potential differentiation in the support that different sonification approaches provided to the analysts in addition to the visual graphs. There has been an increasing interest in experiments with data sonification techniques Though there are no systematic recommendations for mappings between data and sound, in general, these mappings can be split into three types: mapping data values or functions of them to (i) an audio frequency; (ii) musical notes and combinations of them; and (iii) specific sound patterns (like drum beat), where changes in the data are reflected in changes in the parameters of those sound patterns. The parameter selection process is complicated by constraints on the quality of the generated sound. For example, if the analyst decides to map data onto musical notes and use the tempo parameter to rescale too detailed time series, then, when selecting the range of the tempo and the instrument to render sonified data, the analyst has to be careful to avoid the introduction of distortions in the sound. Similar to the sensitivity of visual data representations to the cognitive style of the analyst, sonification may be sensitive towards the individual abilities to hear, which is one of the hypotheses of the study in this chapter.

In chapter "Context visualisation for visual data mining" Mao Lin Huang and Quang Vinh Nguyen focused on the methodology and visualisation techniques which support the presence of history and the context of the visual exploration in the visual display. Context and history visualization plays an important role in visual data mining especially in the visual exploration of large and complex data sets. Incorporating in the visual display context and history information can assist the discovery and the sense making processes, allowing to grasp the context in which a pattern occurs and possible periodical reoccurrence in similar or changed context. Context and history facilitate our short- and long-term memory in the visual reasoning process. The importance of providing both context and some history has been reemphasised in the visual data mining framework presented in [22].

Chapter "Assisting human cognition in visual data mining" brings in the focus of the reader the issue of visual bias, introduced by each visualisation technique, which may result in biased and inaccurate analytical outcomes. Visual bias is a perceptual bias of the visual attention introduced by the visualisation metaphor and leading to selective rendering of the underlying data structures. As interaction with visual representations of the data is subjective, the results of this interaction are prone to misinterpretation, especially in the case of analysts with lesser experience and understanding of the applicability and limitations of different visual data mining methods. Similar to the chapter "Visual discovery of network patterns of interaction between attributes", this chapter takes the viewpoint on visual data mining as a "reflection-in-action" process. Then subjective bias within this framework can be addressed in two ways – one is to enable the human visual pattern recognition through data transformations that possess particular, known in advance, properties and facilitate visual separation of the data points. The other approach that authors propose to use in addressing subjective bias is a corrective feedback, by validating visual findings through employing another method to refine and validate the outcomes of the analysis. The two methods, labelled as "guided cognition" and "validated cognition", respectively, are presented through examples from case studies.

Part 3 – Tools and Applications includes chapters, whose focus is on specific visual data mining systems. This is a selective, rather than a comprehensive, inclusion of systems or combinations of tools that enable visual data mining methodologies. A fairly comprehensive design criteria for visual data mining systems is presented in [22].

In chapter "3DVDM: A system for exploring data in virtual reality", Henrik Nagel, Erik Granum, Søren Bovbjerg and Michael Vittrup present an immersive visual data mining system which operates on different types of displays, spanning from a common desktop screen to virtual reality environments as a six-side CAVE, Panorama and PowerWall. The 3DVDM system was developed as part of the interdisciplinary research project at Aalborg University, Denmark, mentioned earlier in the chapter, with partners from the Department of Computer Science, the Department of Mathematical Sciences, the Faculty of Humanities, and the Institute of Electronic Systems. The system has an extensible framework for interactive visual data mining. The chapter presents the basic techniques implemented in the core of the system. The pioneering work and experiments in visual data mining in virtual reality demonstrated the potential of immersive visual data mining, i.e. the immersion of the analyst within the visual representation of the data. This is substantially different situation in comparison to viewing data on conventional screens, where analysts observe the different visual representations and projections of the data set from outside the visualisation. In the 3DVDM system, the analyst can walk in and out of the visual presence of the data, as well as take a dive into specific part of the data set. The value of the research and development of the 3DVDM system is in the numerous research topics that it has opened, spanning from issues in building representative lower dimension projections of the multidimensional data that preserve the properties of the data through to issues in human-data interaction and visual data mining methodologies in virtual reality environments.

In chapter "DataJewel: Tightly integrating visualization with temporal data mining", Mihael Ankerst, David Jones, Anne Kao and Changzhou Wang present the

architecture and the algorithms behind a visual data mining tool for mining temporal data. The novelty in the architecture is the tight integration of the visual representation, the underlying algorithmic component and the data structure that supports mining of large temporal databases. The design of the system enables the integration of temporal data mining algorithms in the system. DataJewel – that is the catchy name of the tool, offers an integrating interface, which uses colour to integrate the activities both of the algorithms and the analyst. The chapter reports on experiments in analysing large datasets incorporating data from airplane maintenance and discusses the applicability of the approach to homeland security, market basket analysis and web mining.

In chapter "A visual data mining environment" Stephen Kimani, Tiziana Catarci, and Giuseppe Santucci present visual data mining system VidaMine. The philosophy behind the design of VidaMine is similar to the philosophy underpinning the design of DataJewel: (i) an architecture, open for inclusion of new algorithms and (ii) a design, which aims at supporting the analyst throughout the entire discovery process. The chapter presents the details of the architecture and looks at different data mining tasks and the way the environment supports the corresponding scenarios. The chapter includes a usability study of a prototype from the point of view of the analyst – a step that should become a standard for the development of systems for visual data mining process and, in general, any system that supports human-centred data mining process.

Chapter "Integrative visual data mining of biomedical data: Investigating cases in chronic fatigue syndrome and acute lymphoblastic leukaemia" presents the application of visual data mining as part of an overall cycle for enabling knowledge support of clinical reasoning. The authors argued that similar to the holistic approach of the Eastern medicine, knowledge discovery in the biomedical domain requires methods that address the integrated data set of many sources that can potentially contribute to the accurate picture of the individual disease. Presented methods and algorithms, and the integration of the individual algorithms and methods, known informally as the "galaxy approach" (as opposed to the reductionist approach), addresses the issue of the complexity of biomedical knowledge, and, hence, the methods used to mine biomedical data. Data mining techniques are not only part of the patterns discovery process, but also contribute to relating models with existing biomedical knowledge bases and the creation of a visual representation of discovered existing relations in order to formulate hypotheses to question biomedical researchers. The practical application of the methodology and the specific data mining techniques are demonstrated in identifying the biological mechanisms of two different types of diseases: Chronic Fatigue Syndrome and Acute Lymphoblastic Leukaemia, respectively, which share the structure of collected data.

In the last chapter of the book – "Towards effective visual data mining with cooperative approaches", François Poulet addresses the issues of tight coupling of the visualisation and analytical processes and forming an integrated data-mining tool that builds on the strengths of both camps. The author demonstrates his approach on coupling two techniques, some aspects of which have been discussed earlier in the book: the interactive decision tree algorithm CIAD (see chapter "Interactive decision tree construction for interval and taxonomical data") and relating visualisation and SVM (see chapter "Visual Methods for Examining SVM Classifiers"). In this chapter these techniques are linked together – on the one hand, SVM optimises the interactive split

in the tree node, on the other hand, interactive visualisation is coupled with SVM to tune SVM parameters and provide explanation of the results.

5 Conclusions and Acknowledgements

The chapters that span this book draw together the state-of-the-art in the theory, methods, algorithms, practical implementations and applications of what constitutes the field of visual data mining. Through the collection of these chapters the book presents a selected slice of the theory, methods, techniques and technological development in visual data mining – a human-centric data mining approach, which has an increasing importance in the context of visual analytics [17]. There are numerous works on data mining, information visualisation, visual computing, visual reasoning and analysis, but there has been a gap in placing related works in the context of visual data mining. Human visual pattern recognition skills based on detection of changes in shape, colour and motion of objects have been recognised, but rarely positioned in-depth in the context of visual data mining ([20] is an exception). The purpose of this book is to fill these gaps. The unique feature of visual data mining is that it emerges during the interaction with data displays. Through such interaction, classes of patterns are revealed both by the visual structures formed in the displays and the position of these structures within the displays. As an interdisciplinary field, visual data mining is in a stage of forming its own niche, going beyond the unique amalgamation of the disciplines involved.

References

1. Beilken, C., Spenke, M.: Visual interactive data mining with InfoZoom - the Medical Data Set, In: Proceedings 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD 1999, Prague, Czech Republic (1999)
2. Niggemann, O.: Visual Data Mining of Graph-Based Data. Department of Mathematics and Computer Science. University of Paderborn, Germany (2001)
3. Ankerst, M.: Visual Data Mining, in Ph.D. thesis, Dissertation.de: Faculty of Mathematics and Computer Science, University of Munich (2000)
4. Hofmann, H., Siebes, A., Wilhelm, A.F.X.: Visualizing association rules with interactive mosaic plots. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM Press, Boston (2000)
5. Zhao, K., et al.: Opportunity Map: A visualization framework for fast identification of actionable knowledge. In: Proceedings of the ACM Fourteenth Conference on Information and Knowledge Management (CIKM 2005), Bremen, Germany (2005)
6. Blanchard, J., Guillet, F., Briand, H.: Interactive visual exploration of association rules with rule-focusing methodology. Knowledge and Information Systems 13(1), 43–75 (2007)
7. Pirolli, P., Card, S.: Sensemaking processes of intelligence analysts and possible leverage points as identified through cognitive task analysis. In: Proceedings of the 2005 International Conference on Intelligence Analysis, McLean, Virginia (2005)
8. Cox, K.C., Eick, S.G., Wills, G.J.: Visual Data Mining: Recognizing Telephone Calling Fraud. Data Mining and Knowledge Discovery 1, 225–231 (1997)

9. Keim, D.A.: Information visualization and visual data mining. IEEE Transactions on Visualization and Computer Graphics 7(1), 100–107 (2002)
10. Leban, G., et al.: VizRank: Data visualization guided by machine learning. Data Mining and Knowledge Discovery 13(2), 119–136 (2006)
11. Westphal, C., Blaxton, T.: Data Mining Solutions: Methods and Tools for Solving Real-World Problems. John wiley & Sons, Inc., New York (1998)
12. Oliveira, M.C.F.d., Levkowitz, H.: From Visual Data Exploration to Visual Data Mining: A Survey. IEEE Transactions On Visualization And Computer Graphics 9(3), 378–394 (2003)
13. Simoff, S.J., Noirhomme-Fraiture, M., Böhlen, M.H. (eds.): Proceedings of the International Workshop on Visual Data Mining VDM@PKDD 2001, Freiburg, Germany (2001)
14. Simoff, S.J., Noirhomme-Fraiture, M., Böhlen, M.H. (eds.): Proceedings International-Workshop on Visual Data Mining VDM@ECML/PKDD 2002, Helsinki, Finland (2002)
15. Simoff, S.J., et al. (eds.): Proceedings 3rd International Workshop on Visual Data Mining VDM@ICDM 2003, Melbourne, Florida, USA (2003)
16. Soukup, T., Davidson, I.: Visual Data Mining: Techniques and Tools for Data Visualization and Mining. John Wiley & Sons, Inc., Chichester (2002)
17. Thomas, J.J., Cook, K.A.: Illuminating the Path: The Research and Development Agenda for Visual Analytics. IEEE CS Press, Los Alamitos (2005)
18. Keim, D.A., et al.: Challenges in visual data analysis. In: Proceedings of the International Conference on Information Visualization (IV 2006). IEEE, Los Alamitos (2006)
19. Inselberg, A.: The plane with parallel coordinates. The Visual Computer 1, 69–91 (1985)
20. Kovalerchuk, B., Schwing, J. (eds.): Visual and Spatial Analysis: Advances in Data Mining, Reasoning, and Problem Solving. Springer, Dordrecht (2004)
21. Hershberger, J., et al.: Space complexity of hierarchical heavy hitters in Multi-Dimensional Data Streams. In: Proceedings of the Twenty-Fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (2005)
22. Schulz, H.-J., Nocke, T., Schumann, H.: A Framework for Visual Data Mining of Structures. In: Twenty-Ninth Australasian Computer Science Conference(ACSC2006). Conferences in Research and Practice in Information Technology, Hobart, Tasmania, Australia. CPRIT, vol. 48 (2006)

The 3DVDM Approach: A Case Study with Clickstream Data

Michael H. Böhlen[1], Linas Bukauskas[2], Arturas Mazeika[1], and Peer Mylov[3]

[1] Faculty of Computer Science, Free University of Bozen-Bolzano, Dominikanerplatz 3, 39100 Bozen, Italy
[2] Faculty of Mathematics and Informatics, Vilnius University, Naugarduko 24, 03225 Vilnius, Lithuania
[3] Institute of Communication, Aalborg University, Niels Jernes Vej 12, 9220 Aalborg Ãst, Denmark

Abstract. Clickstreams are among the most popular data sources because Web servers automatically record each action and the Web log entries promise to add up to a comprehensive description of behaviors of users. Clickstreams, however, are large and raise a number of unique challenges with respect to visual data mining. At the technical level the huge amount of data requires scalable solutions and limits the presentation to summary and model data. Equally challenging is the interpretation of the data at the conceptual level. Many analysis tools are able to produce different types of statistical charts. However, the step from statistical charts to comprehensive information about customer behavior is still largely unresolved. We propose a density surface based analysis of 3D data that uses state-of-the-art interaction techniques to explore the data at various granularities.

1 Introduction

Visual data mining is a young and emerging discipline that combines knowledge and techniques form a number of areas. The ultimate goal of visual data mining is to devise visualizations of large amounts of data that *facilitate the interpretation* of the data. Thus, visual data mining tools should be expected to provide informative but not necessarily nice visualizations. Similarly, visual data mining tools will usually enrich or replace the raw data with model information to support the interpretation. These goals, although widely acknowledged, often get sidelined and are dominated by the development of new visual data mining tools. Clearly, data analysis tools are important but it is at least as important to design principles and techniques that are independent of any specific tool.

We present an interdisciplinary approach towards visual data mining that combines mature and independent expertise from multiple areas: database systems, statistical analysis, perceptual psychology, and scientific visualization. Clearly, each of these areas is an important player in the visual data mining process. However, in isolation each area also lacks some of the required expertise. To illustrate this, we briefly recall three very basic properties that any realistic visual data mining system must fulfill. Typically each property is inherent to one area but poorly handled in the other areas.

Relevant data. At no point in time is it feasible to load or visualize a significant part, let alone all, of the available data. The data retrieval must be limited to the relevant

S.J. Simoff et al. (Eds.): Visual Data Mining, LNCS 4404, pp. 13–29, 2008.

part of the data. The relevant part of the data might have to depend on the state of the data mining process, e.g., be defined as the part of the data that is analyzed at a specific point in time.

Model information. It is necessary to visualize model information rather than raw data. Raw data suffers from a number of drawbacks and a direct visualization is not only slow but makes the quality depend on the properties of the data. A proper model abstracts from individual observations and facilitates the interpretation.

Visual properties. It is very easy to overload visualizations and make them difficult to interpret. The effective use of visual properties (size, color, shape, orientation, etc.) is a challenge [16]. Often, a small number of carefully selected visual properties tuns out to be more informative than visual properties galore.

This paper describes a general-purpose interactive visual data mining system that lives up to these properties. Specifically, we explain *incremental observer relative data extraction* to retrieve a controlled superset of the data that an observer can see. We motivate the use of *density information* as an accurate model of the data. We visualize *density surfaces* and suggest a small number of interaction techniques to explore the data. The key techniques are animation, conditional analyzes, equalization, and windowing. We feel that these techniques should be part of any state-of-the-art visual data mining system.

Animation. Animation is essential to explore the data at different levels of granularity and get an overview and feel for the data. Without animations it is very easy to draw wrong conclusions.

Windowing. For analysis purposes it is often useful to have a general windowing functionality that allows to selectively collapse and expands individual dimensions and that can be used to change the ordering of (categorical) attribute values.

Equalization. The data distribution is often highly skewed. This means that visualizations are dominated by very pronounced patterns. Often these patterns are well known (and thus less interesting) and they make it difficult to explore other parts of the data if no equalization is provided.

Conditional Analyzes. Data sets combine different and often independent information. It must be possible to easily perform and relate analyzes on different parts of the data (i.e., conditional data analyzes).

In Section 4 we will revisit these techniques in detail and discuss how they support the interpretation of the data. Throughout the paper we use clickstream data to illustrate these techniques and show how they contribute to a robust interpretation of the data. Clickstream data is a useful yardstick because it is abound and because the amount of data requires scalable solutions. We experienced that a number of analysis tools exhibit significant problems when ran on clickstream data, and the performance quickly became prohibitive for interactive use.

There are a number of tools that offer advanced visual data mining functionality: GGobi [20], MineSet [2], Clementine [11], QUEST [1], and Enterprise Miner [3]. Typically, these are comprehensive systems with a broad range of functionalities. Many of them support most of the standard algorithms known from data mining, machine learning, and statistics, such as association rules, clustering (hierarchical, density-based, Kohonen, k-means), classification, decision trees, regression, and principal components.

None of them supports data mining based density surfaces for 3D data and advanced interaction techniques for the data analyst. Many of the systems also evolved from more specialized systems targeted at a specific area.

Data access structures are used to identify the relevant data and are often based on the B- and R-tree [9]. R-tree based structures use minimum bounding rectangles to hierarchically group objects and they support fast lookups of objects that overlap a query point/window. Another family of access structures is based only on space partitioning as used in the kd-tree [17]. Common to all these structures is a spatial grouping. In our case the visible objects are not necessarily located in the same area. Each object has its own visibility range and therefore the visible objects may be located anywhere in the universe. In addition to the lack of a spatial grouping of visible objects the above mentioned access structures also do not support the incremental extraction of visible objects, which we use to generate the stream of visible objects.

For the model construction we estimate the probability density function (PDF). Probability density functions and kernel estimation have been used for several decades in statistics [18,19,6,5,23]. The main focus has been the precision of the estimation, while we use it to interactively compute density surfaces. We use the density surfaces to partition the space. Space partitioning has been investigated in connection with clustering [10,8,22,24,4,12] and visualization. The visualization is usually limited to drawing a simple shape (dot, icon glyph, etc.) for each point in a cluster [4,12] or drawing (a combination of) ellipsoids around each cluster [21,7]. The techniques are usually not prepared to handle data that varies significantly in size, contains noise, or includes multiple not equally pronounced structures [15]. Moreover, a different density levels often requires a re-computation of the clusters, which is unsuitable for interactive scenarios let alone animation.

Structure of the paper: Section 2 discusses clickstreams. We discuss the format of Web logs, show how to model clickstreams in a data warehouse, and briefly point to current techniques for analyzing clickstreams. Section 3 discusses selected aspects of the 3DVDM System, namely the use of incremental observer relative data extraction to generate a stream of visible observations, and the model computation from this stream. Section 4 uses the clickstream data to illustrate data mining based on density surfaces.

2 Clickstreams

One of the most exciting data sources are Web logs (or clickstreams). A clickstream contains a record for every page request from every visitor to a specific site. Thus, a clickstream records every gesture each visitor makes and these gestures have the potential to add up to comprehensive descriptions of the behavior of users. We expect that clickstreams will identify successful and unsuccessful sessions on our sites, determine happy and frustrated visitors, and reveal the parts of our Web sites are (in)effective at attracting and retaining visitors.

2.1 Web Server Log Files

We start out by looking at a single standardized entry in a log file of a web server:

```
ask.cs.auc.dk  [13/Aug/2002:11:49:24 +0200]
  "GET /general/reservation/cgi-bin/res.cgi HTTP/1.0"
  200  4161  "http://www.cs.auc.dk/general_info/"
  "Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)"
```

Such an entry contains the following information.

1. The IP address of the visitor. Possibly a cookie ID is included as well.
2. The date and time (GMT) of the page request.
3. The precise HTTP request. The type of request is usually `Get` or `Submit`.
4. The returned HTTP status code. For example, code 200 for a successful request, code 404 for a non-existing URL, code 408 for a timeout, etc.
5. The number of bytes that have been transferred to the requesting site.
6. The most recent referring URL (this information is extracted from the referrer field of the HTTP request header).
7. The requesting software; usually a browser like Internet Explorer or Netscape, but it can also be a robot of a search engine.

Web logs do not only contain entries for pages that were explicitly requested. It is common for a page to include links to, e.g., pictures. The browser usually downloads these parts of a document as well and each such download is recorded in the Web log. Consider the two example entries below. The first entry reports the download of the root page. The second entry is the result of downloading an image that the root page refers to.

```
0x50c406b3.abnxx3.adsl-dhcp.tele.dk
  [13/Aug/2002:11:49:27 +0200]
  "GET / HTTP/1.1"
  200  3464  "-"
  "Mozilla/4.0 (compatible; MSIE 5.0; Windows NT; DigExt)"

0x50c406b3.abnxx3.adsl-dhcp.tele.dk
  [13/Aug/2002:11:49:27 +0200]
  "GET /images/educate.jpg HTTP/1.1"
  200  9262  "http://www.cs.auc.dk/"
  "Mozilla/4.0 (compatible; MSIE 5.0; Windows NT; DigExt)"
```

2.2 Modeling Clickstreams in a Data Warehouse

We are a long way from inferring user behavior just by analyzing the entries in a web log. We have to clean the raw log data and organize it in a way that supports the business perspective. A well known approach is to use ETL (extract transform load) techniques to pre-process and load the data into a data warehouse that supports business analyzes. Figure 1 shows an example data warehouse schema.

The schema in Figure 1 is a classical data warehouse start schema [13]. The Clickstream fact table contains an entry for every record in the Web log and is several GB large. Each entry has a number of key attributes that point to descriptive entries in the

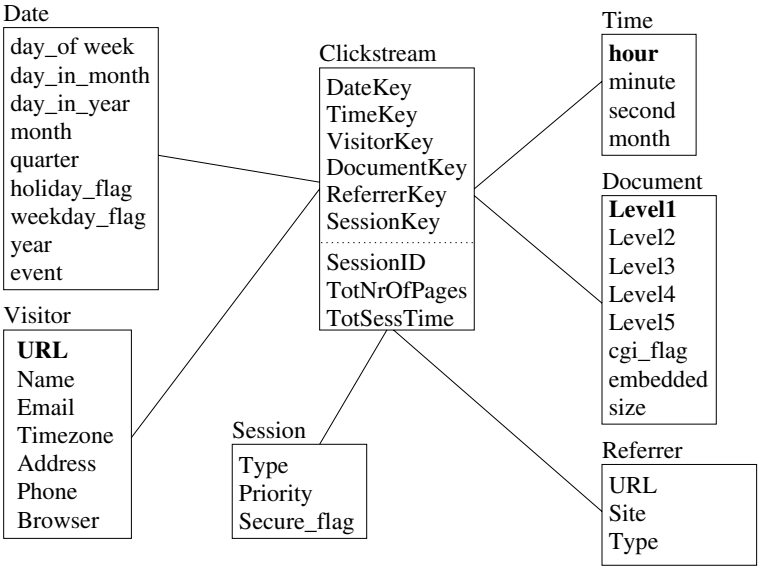


Fig. 1. Star Schema of a Clickstream Data Warehouse

dimension tables. The measure attributes carry information about individual sessions (ID, total number of pages accessed during a session, total session time). Note that these attributes are not available in the log file. They have to be inferred during the data preprocessing phase. A common approach is to sort the log on IP and time, and then use a sequential scan to identify sequences of clicks that form a session. Below we briefly describe those aspects of the dimensions that have unique properties related to the analysis of clickstreams [14].

Date Dimension. The date dimension is small and has a few thousand records at most (365 days per year). The business requirements determine the attributes that should be included. For example, we should record for each day whether it is a holiday or not if some of our analyzes will treat holidays separately.

Time Dimension. All times must be expressed relative to a single standard time zone such as GMT that does not vary with daylight savings time. The time dimension has 86,400 records, one for each second of a given day. A separate time dimension allows analyzes across days and makes it easy to constrain the time independent of the day.

Visitor Dimension. It is possible to distinguish three types of visitors. Anonymous visitors are identified by their IP addresses. In many cases, the IP address only identifies a port on the visitor's Internet service provider. Often, these ports are dynamically reassigned, which makes it difficult to track visitors within a session let alone across sessions. A cookie visitor is one who has agreed to store a cookie. This cookie is a reliable identifier for a visitor machine. With a cookie, we can be pretty sure that a given machine is responsible for a session, and we can determine when the machine will visit us again. An authenticated visitor is the human-identified visitor who not only

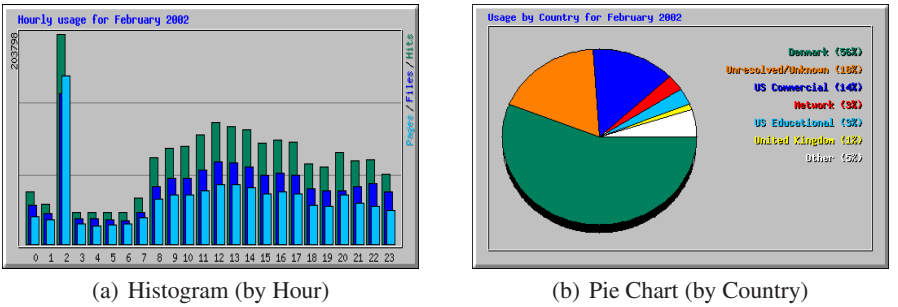
has accepted our cookie but sometime in the past has revealed the name and other information. The visitor dimension is potentially huge but its size can often be reduced significantly. For example, anonymous visitors can be grouped according to domain (and sub-domain). For cookie and authenticated visitors it is likely that we want to build up individual profiles, though.

Document Dimension. The document dimension requires a lot of work to make the clickstream source useful. It is necessary to describe a document by more than its location in the file system. In some cases, the path name to the file is moderately descriptive (e.g., a .jpg extension identifies a picture), but this is certainly not always the case. Ideally, any file on a Web server should be associated with a number of categorical attributes that describe and classify the document.

Session Dimension. The session dimension is a high-level diagnosis of sessions. Possible types are student, prospective student, researcher, surfer, happy surfer, unhappy surfer, crawler, etc. The session information is not explicitly stored in the Web log file. Basically, it has to be reverse engineered from the log file and added during data preprocessing.

2.3 Analyzing Clickstreams

There are a number of standard statistical tools available that are being used to analyze web logs. These tools are used to summarize the data and display the summaries. Two typical charts are shown in Figure 2.



(a) Histogram (by Hour) (b) Pie Chart (by Country)

Fig. 2. Statistical Analysis of a Web Log

Typically, the summaries are computed for one dimension as illustrated by the charts in Figure 7(a). 2D (or even 3D) summaries are more rare, although some of the tools offer (2D) crosstabing. As a result the information is often spread over a large number of tables and graphs, and it is difficult to analyze a clickstream comprehensively.

We propose to combine much of this information into a single graph and use state-of-the-art interaction techniques to analyze the data from different perspectives and at different granularities.

3 The 3DVDM System

3.1 Overall Architecture

As mentioned in the introduction visual data mining is fundamentally inter-disciplinary. Any visual data mining system should make this a prime issue for the basic design of the system. An obvious approach is to go for a modular system with independent components as illustrated in Figure 3.

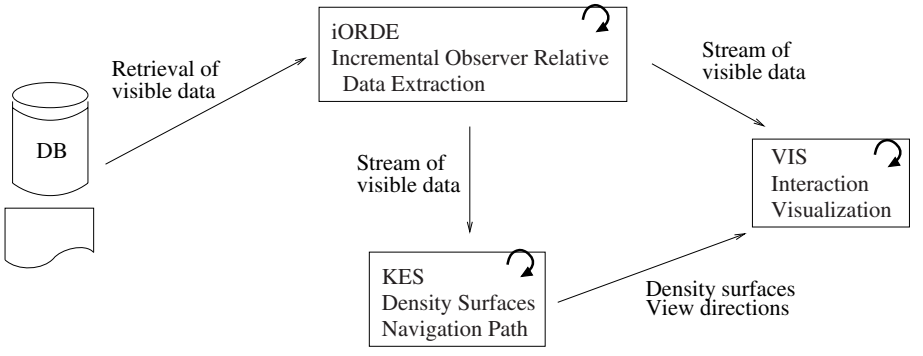


Fig. 3. 3DVDM System Architecture

The modules are implemented as separate processes that communicate through streams. Each module has an event handler that reacts to specific events. These events can be triggered by any of the other modules. Other than this no cooperation between or knowledge of the modules is required, which is an asset for a true interdisciplinary system.

The 3DVDM System is being used in environments (Panorama, Cave) where the data analyst has the possibility to explore the data from the inside and outside. BElow we describe techniques to retrieve and stream the data during such explorations and show how to process the stream and compute model information.

3.2 Streaming Visible Data

Visible objects are identified by the distance in which the object is seen. We use the *visibility range* to define the visibility of an object. In the simplest case it is proportional to the size of an object: $VR(o_i) = o_i[s] \cdot c$. Here, $VR(o_i)$ is the visibility range of object o_i, $o_i[s]$ is the size of the object, and c is a constant. Thus, object o_i will be visible in the circular or hyper-spherical area of size $VR(o_i)$ around the center of o_i.

We write $\mathcal{V}_{P_l}$ to denote the set of visible objects from the point P_l. Objects that become visible when an observer moves from position P_l to position P_{l+1} are denoted by $\Delta^+_{P_l,P_{l+1}}$. With this the stream of visible data from position P_0 to P_k is defined as a sequence $\mathcal{S}(P_0, P_k) = \langle \mathcal{V}_{P_0}, \Delta^+_{P_0,P_1} \ldots \Delta^+_{P_{k-2},P_{k-1}} \rangle$. Here, k is a number of observer positions and the number of stream slices. The definition says that we stream all visible data for the initial observer position. Any subsequent slices are produced as increments.

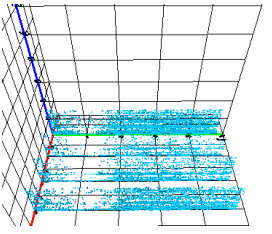
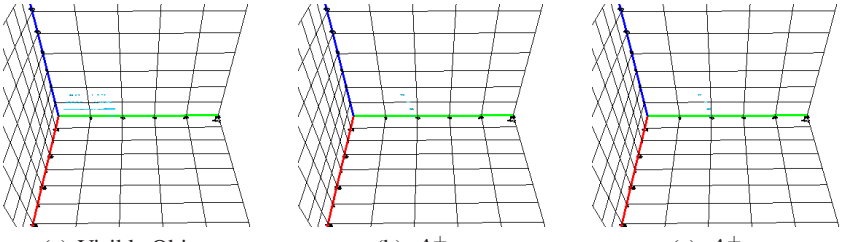


Fig. 4. The Universe of Data



(a) Visible Objects (b) $\Delta^{+}_{P_0,P_1}$ (c) $\Delta^{+}_{P_1,P_2}$

Fig. 5. Stream of Visible Data

Figure 3.2 shows a part of the clicktream data. Specifically, the clicks from four domains are shown: .it, .de, .uk, and .es. We assume equally sized visibility ranges for all clicks (alternatively, the VR could be chosen to be proportional to, e.g., the size of the downloaded document) and let the observer move along the clicks from .it.

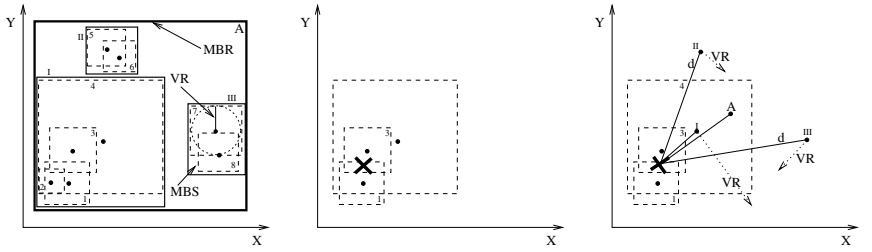
The observer starts at the time position 5 and the VR is 4. Initially, 1728 visible object are received. In Figure 5(b) and 5(c) the incremental slices are shown. $\Delta^{+}_{P_0,P_1}$ contains 503 and $\Delta^{+}_{P_1,P_2}$ 363 newly visible objects.

To provide efficient streaming of visible objects we use the VR-tree, an extension of the R-tree, together with the Volatile Access Structure (VAST). The structures enable a fast and scalable computation of Δ^{+} when the observer is moving.

VAST is called *volatile* because it is initialized and created when the observer starts moving. It is destroyed if the observer no longer needs streaming support. Figure 6 shows an example of VAST that mirrors a subset of the VR-tree. Figure 6(a) shows the hierarchical structure of the VR-Tree. Figure 6(b) shows only the visible objects. These are all objects with a VR that includes the observer position. VAST mirrors the visible part of the VR tree as illustrated in Figure 6(c). In contrast to the VR tree, which uses absolute Cartesian coordinates, VAST uses an observer relative representation with distances and angles. This representation is more compact and supports a moving observer.

3.3 Model Computation

The visible data is streamed to the model computation module where it is processed in slices to estimate the PDF. The estimation algorithm starts with a coarse uniform grid data structure and processes the first slice of the data. The estimation process can be in



(a) VR-Tree (b) Visible Objects (c) VAST

Fig. 6. The VR-Tree migration

two states: the estimation or skipping state. In the estimation state the algorithm refines the PDF by adding new kernels and refining the grid structure in regions where the PDF is non-linear. The process continues until the desired precision is reached. When the estimation is precise enough it enters the skipping state. In the skipping state the algorithm skips the slices until it finds new information that is not yet reflected in the PDF. If such a slice is found the processing is resumed. The individual steps of the stream-base processing are shown in code fragment 1.

Code Fragment 1. *Estimation of the Probability Density Function*

```
Algorithm: estimatePDF
Input:
  vDataset: sequence of data points
  ε: accepted precision
  InitG: initial number of grids

Output:
  APDF tree a
Body:
  1. Initialize a
  2. skipState = FALSE
  3. FOR EACH slice s_i DO
       3.1 IF !skipState THEN
             3.1.1 Process slice s_i.
             3.1.2 Split the space according to the precision of the estimation
             3.1.3 IF precisionIsOK(a) THEN
                     skipState = TRUE
                   END IF
       3.2 ELSIF newStream(s_i) THEN
             3.2.1 skipState = FALSE
           END IF
     END FOR
```

The density surface module can be in one of two states: active (in the process mode) and inactive (waits for an wake up event). The module is active if: (i) a change of the input parameters has triggered the recalculation of the PDFs and/or density surfaces or (ii) the animation of density surfaces is switched on.

Pseudo Code Fragment 2 sketches the event handling. The first block handles new data. A new (or changed) data set triggers the re-computation of the model. The block deletes previous model information if requested (sub-block 1.1), splits the dataset into conditional datasets, and calculates the PDFs for the individual datasets (sub-block 1.3).

The second block handles animation events. It calculates the next density surface level (sub-blocks 2.1-2.2) and computes the corresponding density surfaces (sub-blocks 2.3-2.4). The third block handles events triggered by a change of the density level. This requires a recalculation of the density surfaces at the requested density level. The 4th block visualizes the computed density surfaces and, optionally, corresponding data samples.

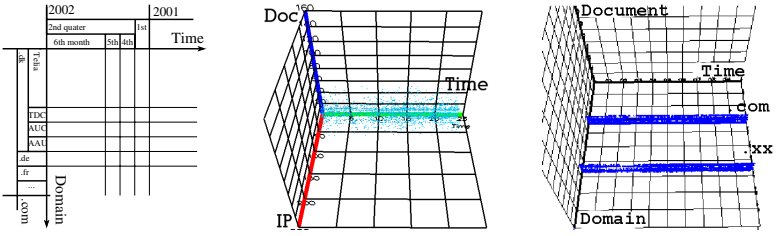
Code Fragment 2. *Dispatcher of the DS module*

```
1. IF new(vDataset) THEN
   1.1 IF reset THEN
         vPDF = vSurfaces = vPlacement = ∅
       END IF
   1.2 IF bConditional THEN
         Split vDataset according to the categorical attribute into D1,...,Dk.
       ELSE
         D1 = vDataset
       END IF
   1.3 FOR EACH Dataset Di DO
         vPDF = vPDF ∪ estimatePDF(Di, EstErr, InitG)
         vPlacement = vPlacement ∪ cPlacement
       END FOR
   END IF

2. IF bAnimate THEN
      2.1 animateDSLevel += 1.0 / animateDSFrames
      2.2 IF animateDSLevel >= 1.0 THEN
            animateDSLevel = 1.0 / animateDSFrames
      END IF
      2.3 vSurface = ∅
      2.4 FOR EACH vPDFi DO
          vSurface = vSurface ∪
          calculateDS(vPDFi, animateDSLevel, iDSGridSize, bEqualized)
          END FOR
      2.5 Request for calculation
    END IF

3. IF !bAnimate AND changed(fDSLevel) THEN
      3.1 FOR EACH vPDFi DO
          vSurface = vSurface ∪
           calculateDS(vPDFi, fDSLevel, iDSGridSize, bEqualized)
   END IF

4. visualizeObs()
```

4 Data Analyzes and Interpretation

This section discusses and illustrates the four key interaction techniques: animation, conditional analyzes, equalization, and windowing. These are general techniques that support the interpretation of the data and are useful in other settings as well. Figure 7 illustrates the challenges the data mining process has to deal with. Neither the visualization of the entire data set (Figure 7(b)) nor the visualization of two domains only (Figure 7(c)) provide detailed insights. We discuss how the above techniques help to investigate and interpret this data.

Below we use I for the interpretation of a visualization. We use I to relate different visualization techniques. Specifically, we formulate theorems that relate the information content of different density surface visualizations.

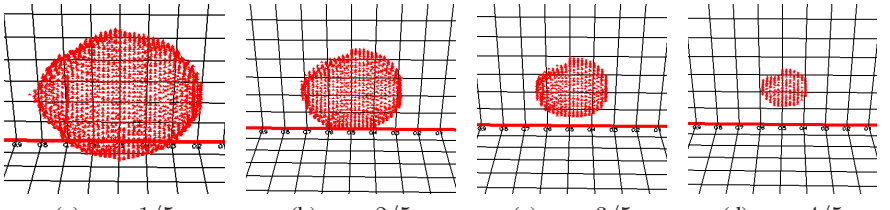


(a) Warehouse (b) All data (c) .dk and .com

Fig. 7. Direct Visualization of a Clickstream Data Warehouse

4.1 Animation

An animation of density surfaces is a (cyclic) visualization of a sequence of density surfaces with a decreasing density level. For example if the number of frames is chosen to be four then the density surfaces at levels $\alpha_1 = 1/5$, $\alpha_2 = 2/5$, $\alpha_3 = 3/5$, and $\alpha_4 = 4/5$ will be visualized in sequence. Since the printed version of the manual is limited and cannot show the animation we present the animation of density surfaces by a sequence of snapshots at different density levels. Figure 8 shows an animated solid sphere (a 3D normal distributions)[1].



(a) $\alpha = 1/5$ (b) $\alpha = 2/5$ (c) $\alpha = 3/5$ (d) $\alpha = 4/5$

Fig. 8. Density Surfaces for Different Surface Grid Size g_s

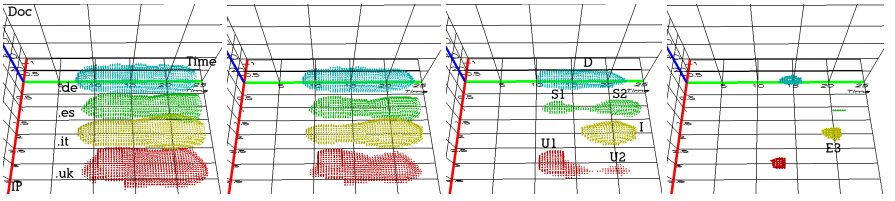
Theorem 1. *Animated density surfaces are strictly more informative than any individual density surface:*

$$\forall k(I[\cup_i DS(D, \alpha_i)] \sqsupset I[DS(D, \alpha_k)])$$

Figure 8, which exhibits a very simple and regular data distribution, illustrates Theorem 1 quite nicely. It is necessary to look at the animated density surfaces to confirm the normal distribution. None of the four snapshots would allow us to infer the same interpretation as the four snapshots together. Note that for any given density level α_k it is possible to construct a dataset D', such that $I[DS(D, \alpha_k)] = I[DS(D', \alpha_k)]$ and $\forall i \neq k : I[DS(D, \alpha_i)] \neq I[DS(D', \alpha_i)]$.

Figure 9 shows animated density surfaces for a subset of the clickstream data set. The figure shows the clicks from the following domains: Italy (30%), Germany (28%), the UK (22%), and Spain (20%).

[1] Because of rescaling the sphere is deformed and resembles an ellipsoid.



(a) $\alpha = 0.2$ (b) $\alpha = 0.3$ (c) $\alpha = 0.5$ (d) $\alpha = 0.9$

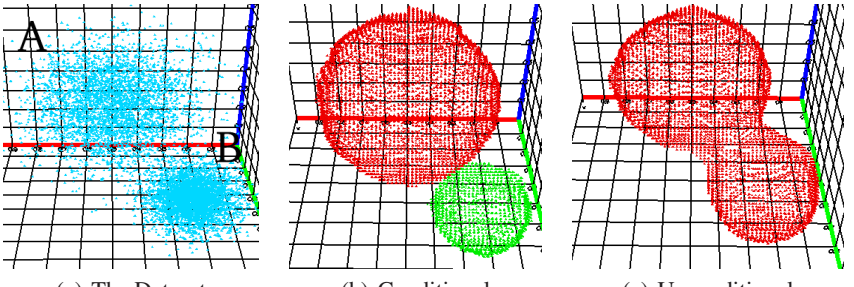
Fig. 9. .it, .de, .uk, .es

The four countries are very different in culture and working style. The density surfaces nicely reflect this. For example in Spain people have a long break (siesta) in the middle of the day. The siesta divides the density surface for Spain into two similar sized surfaces (cf. $S1$ and $S2$, Figure 9(c)). Italy produces most of the clicks during the second part of the day (cf. surface I, Figure 9(c)). In contrast most of the UK clicks are received in the first part of the day (cf. surface $U1$, Figure 9(c)). The German domain is bound by the working hours and peaks in the afternoon. Figure 9(d) shows the peaks for the individual domains.

4.2 Conditional Density Surfaces

Conditional density surfaces are the result of splitting a data set into several subsets and computing a model (i.e., density surfaces) for each of them. A common approach is to split a data set according to the values of a categorical attribute (e.g., sex, country, etc).

An example of conditional density surfaces is shown in Figure 10. The data set contains two normal distributions. A binary attribute W was added to assign the observations to the clusters. All observations that fall into the cluster A have $W = 1$, while observations that fall into cluster B have $W = 2$. Figures 10(c) shows a non-conditional density surface. A single density surface encloses both structures in space. Figure 10(b) shows conditional density surfaces. The conditional analysis separates the data points and yields two independent (but possibly intersecting) density surfaces.



(a) The Dataset (b) Conditional (c) Unconditional

Fig. 10. Difference between Conditional and Unconditional Analysis

Theorem 2. *Let $D_1, ..., D_k$ be independent data sets with identical schemas $\mathbf{D_i}$, and let $D = \cup_{i=1}^{k} \pi[\mathbf{D_i}, i](D_i)$ be the union of these data sets with schema $\mathbf{D} = \mathbf{D_i} \cup \{W\}$. If a conditional data set contains independent structures then an appropriate conditional analysis is more informative than an unconditional analysis.*

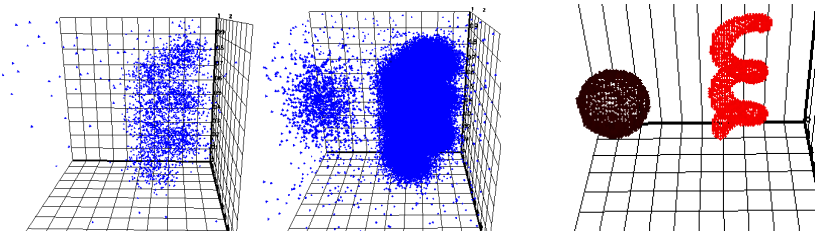
$$I[\cup_i DS(\sigma[W = i](D), \alpha)] \sqsupseteq I[DS(D, \alpha)])$$

Basically, a conditional analysis will yield the same result as an (unconditional) analysis of the individual data sets. If the data sets are independent this is the optimal result. In practice, conditional analyzes can also be useful if independence is not achieved completely. For example, it can be appropriate to treat the clicks from different countries as being independent even if this is not completely accurate. Often it is best to try out both types of analysis, which means that switching between consitional and unconditional analyzes should be supported by the interaction tools available to the data analyst.

4.3 Equalization of Structures

Many real world data sets contain very skewed data. For example, it is likely that a web server will handle a huge amount of local traffic. This does not necessarily mean that non-local domains are less important for the data analysis. In order to support the exploration of not equally pronounced structures it is necessary to equalize the structures before they are displayed.

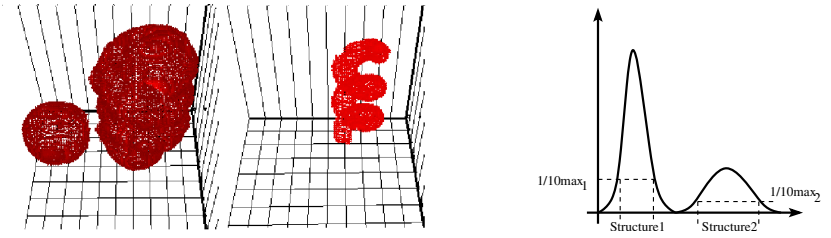
The data set in Figure 11 contains two structures: a spiral (80% of all observations) and a sphere (20% of all observations). In Figure 11(a) we chose the number of observations that yields the best visualization of the spiral. The result is a scatter plot that does not show the sphere. In Figure 11(b) we increase the number of observations until the sphere can be identified. This yields a scatter plot that makes it difficult to identify the spiral. Clearly, it is much easier to identify the structures if equalized density surfaces are used (cf. Figure 11(c)).



(a) $n = 3'000$ (b) $n = 100'000$ (c) Equalized DS

Fig. 11. Scatterplots versus Equalized Density Surfaces

Since the densities of the structures are very different, an overall best density level does not exist as illustrated in Figure 12(a). A low density yields an appropriate sphere but only shows a very rough contour of the spiral. A high density shows a nice spiral but does not show the sphere at all. Basically, we experience the same problems as with the scatterplot visualization. The density information makes it fairly straightforward to



(a) Non-Equalized Density Surfaces (b) Lazy Equalization

Fig. 12. Equalization of Structures

equalize the two structures. The idea is illustrated in Figure 12(b). The structures are equalized by adaptively (i.e., based on the density of a structure) calculating the density level for each structure (e.g., 10% of the maximum density for each structure).

Equalization requires a separation of structures. This comes for free if we use conditional density surfaces. Equalization and conditional density surfaces therefore complement each other nicely. With unconditional analyzes the separation has to be implemented as a separate process.

Note that, in contrast to density surfaces, scatterplot visualizations are not easy to equalize. In principle, one has to determine the optimal number of observations that shall be visualized for each structure. This number depends on many factors and cannot be determined easily.

Theorem 3. *Any data set D can be completely dominated:*

$$\forall D \exists D'(I[DS(D', \alpha)] = I[DS(D \cup D', \alpha)])$$

Figure 12(a) illustrates how the spiral dominates the sphere. Here the domination is not complete because there is still a density level at which the sphere can be identified. If the density of the spiral was increased further the sphere will eventually be treated as noise.

Theorem 4. *Equalization preserves the interpretation of individual structures:*

$$\forall D, D'(I[DS^{eq}(D \cup D', \alpha)] = I[DS(D, \alpha)] \cup I[DS(D', \alpha)])$$

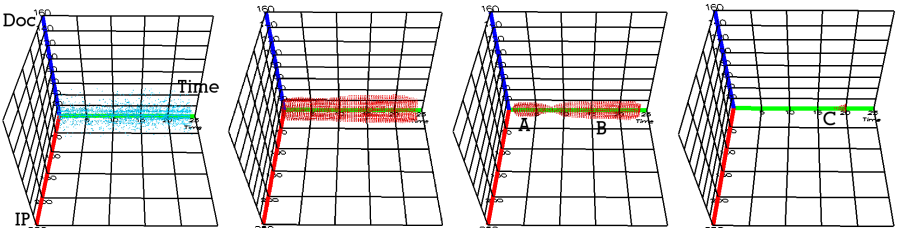
The preservation is illustrated in Figure 11(c). Here the two structures are shown as if each of them was analyzed individually.

4.4 Windowing

The window functionality is used if we want to investigate a subset of the data in more detail. Technically, the window function filters out the observations that fall outside the selected window and re-scales the data to the unit cube. Conceptually, the window functionality enables an investigation at the micro level (a "smart" zoom into the dataset). In contrast to standard zooming techniques, windowing allows to zoom in on a single dimension. For example, it is possible to restrict the analysis to a few domains but preserve the complete time dimension. Note that windowing must trigger a recalculation

of the model. This makes it possible that a single surface at the macro level can split into several surfaces at micro level, as one expects it to be.

Figure 13 is a direct visualization of the clickstream and illustrates the problem. Because the coding preserves the natural ordering of the data all frequent domains (.dk, .com, etc.) are clustered at the beginning of the axis. This yields a very skewed plot that can only be used for a simple overall load analysis: The work-load is high during the very early morning hours and the local working hours (surfaces A and B in Figure 13(c)), and the work-load peaks around 8PM (surface C in Figure 13(d)).

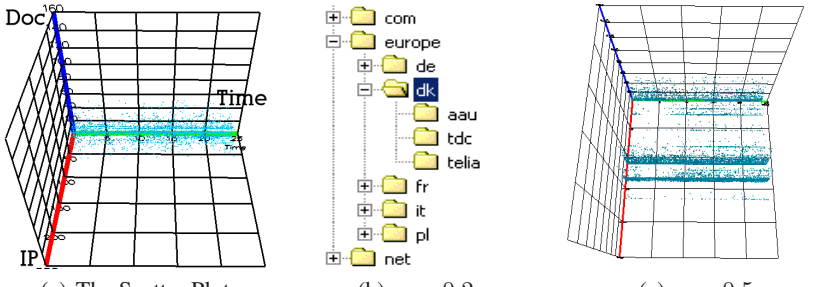


(a) Scatterplot (b) $\alpha = 0.2$ (c) $\alpha = 0.5$ (d) $\alpha = 0.9$

Fig. 13. The Sunsite Dataset (Natural Ordering)

What we want is that the cluster starts to "unfold" as we move closer, i.e., we expect the domains to separate. Standard zooming techniques do not provide this functionality directly. Typically, when moving closer we not only narrow the IP domain but also loose the overview of the time domain. Moreover, zooming does not trigger a re-computation of the model, which means that the surfaces will simply be stretched.

Figure 14 illustrates the idea of the windowing functionality. A simple windowing tool is shown in Figure 14(b). The menu allows to select the domains that shall be visualized (i.e., the window). It supports dimension hierarchies and does not require that the selected domains are contiguous. The effect of zooming in is illustrated in Figure 14(c). The original cluster of points has unfold and it is now possible to analyze selected domains/regions.



(a) The Scatter Plot (b) $\alpha = 0.2$ (c) $\alpha = 0.5$

Fig. 14. The Clickstream Data from www.sunsite.dk

Another useful feature of windowing is re-ordering. If the data analyst wants to change the ordering of the domains this can be very easily done using the tool shown in Figure 14(b).

5 Summary and Future Work

We investigated the potential of density surfaces of 3D data to analyze clickstream data. Density surfaces accurately summarize the data and yields a good model. Density surfaces are simple visual structures that are easy to perceive and support the interpretation of the data. We showed that animation, conditional analyzes, equalization, and windowing are crucial interaction techniques. They make it possible to explore the data at different granularity levels, which leads to a robust interpretation.

In the future it would be interesting to classify density surfaces and come up with an alphabet. The analysis of high-dimensional data is another interesting issue. One could for example investigate projection techniques before the density surfaces are computed, or the (moderate) use of additional visual properties.

References

1. Agrawal, R., Mehta, M., Shafer, J.C., Srikant, R., Arning, A., Bollinger, T.: The quest data mining system. In: Proceedings of ACM SIGKDD, 2-4, 1996, pp. 244–249. AAAI Press, Menlo Park (1996)
2. Brunk, C., Kelly, J., Kohavi, R.: MineSet: an integrated system for data mining. In: Proceedings of SIGKDD, pp. 135–138. AAAI Press, Menlo Park (1997)
3. Cerrito, P.B.: Introduction to Data Mining Using SAS Enterprise Miner. SAS Publishing (2006)
4. Davidson, I., Ward, M.: A Particle Visualization Framework for Clustering and Anomaly Detection. In: Proceedings of Workshop on Visual Data Mining in conjunction with SIGKDD (2001)
5. van den Eijkel, G.C., van der Lubbe, J.C.A., Backer, E.: A Modulated Parzen-Windows Approach for Probability Density Estimation. In: IDA (1997)
6. Farmen, M., Marron, J.S.: An Assesment of Finite Sample Performace of Adaptive Methods in Density Estimation. In: Computational Statistics and Data Analysis (1998)
7. Gross, M.H., Sprenger, T.C., Finger, J.: Visualizing information on a sphere. Visualization (1997)
8. Guha, S., Rastogi, R., Shim, K.: CURE: an Efficient Clustering Algorithm for Large Databases. In: Proceedings of SIGMOD, pp. 73–84 (1998)
9. Guttman, A.: R-Trees: A Dynamic Index Structure for Spatial Searching. In: Proceedings of SIGMOD, pp. 47–57. ACM Press, New York (1984)
10. Hinneburg, A., Keim, D.A.: Optimal Grid-Clustering: Towards Breaking the Curse of Dimensionality in High-Dimensional Clustering. The VLDB Journal, 506–517 (1999)
11. Clementine SPSS Inc. Data mining system: Clementine 12.0 (2008)
12. Keahey, T.A.: Visualization of High-Dimensional Clusters Using Nonlinear Magnification. In: Proceedings of SPIE Visual Data Exploration and Analysis (1999)
13. Kimball, R.: The Data Warehouse Toolkit. John Wiley & Sons, Inc., Chichester (1996)
14. Kimball, R., Merz, R.: The Data Webhouse Toolkit—Building the Web-Enabled Data Warehouse. Wiley Computer Publishing, Chichester (2000)

15. Mazeika, A., Böhlen, M., Mylov, P.: Density Surfaces for Immersive Explorative Data Analyses. In: Proceedings of Workshop on Visual Data Mining in conjunction with SIGKDD (2001)
16. Nagel, H.R., Granum, E., Musaeus, P.: Methods for Visual Mining of Data in Virtual Reality. In: Proceedings of the International Workshop on Visual Data Mining, in conjunction with ECML/PKDD 2001 (2001)
17. Robinson, J.T.: The K-D-B-Tree: A Search Structure For Large Multidimensional Dynamic Indexes. In: Edmund Lien, Y. (ed.) Proceedings of SIGMOD, pp. 10–18. ACM Press, New York (1981)
18. Scott, D.W.: Multivariate Density Estimation. Wiley & Sons, New York (1992)
19. Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Chapman & Hall, London (1986)
20. Swayne, D.F., Lang, D.T., Buja, A., Cook, D.: Ggobi: Evolving from Xgobi into an Extensible Framework for Interactive Data Visualization. Comput. Stat. Data Anal. 43(4), 423–444 (2003)
21. Sprenger, T.C., Brunella, R., Gross, M.H.: H-BLOB: a Hierarchical Visual Clustering Method using Implicit Surfaces. Visualization (2000)
22. Wang, W., Yang, J., Muntz, R.R.: STING: A Statistical Information Grid Approach to Spatial Data Mining. The VLDB Journal, 186–195 (1997)
23. Wand, M.P., Jones, M.C.: Kernel Smoothing. Chapman & Hall, London (1985)
24. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: an Efficient Data Clustering Method for Very Large Databases. In: Proceedings of SIGMOD, pp. 103–114 (1996)

Form-Semantics-Function – A Framework for Designing Visual Data Representations for Visual Data Mining

Simeon J. Simoff

School of Computing and Mathematics
College of Heath and Science
University of Western Sydney
NSW 1797, Australia
s.simoff@uws.edu.au

Abstract. Visual data mining, as an art and science of teasing meaningful insights out of large quantities of data that are incomprehensible in another way, requires consistent visual data representations (information visualisation models). The frequently used expression "the art of information visualisation" appropriately describes the situation. Though substantial work has been done in the area of information visualisation, it is still a challenging activity to find out the methods, techniques and corresponding tools that support visual data mining of a particular type of information. The comparison of visualisation techniques across different designs is not a trivial problem either. This chapter presents an attempt for a consistent approach to formal development, evaluation and comparison of visualisation methods. The application of the approach is illustrated with examples of visualisation models for data from the area of team collaboration in virtual environments and from the results of text analysis.

1 Introduction

In visual data mining large and normally incomprehensible amounts of data are reduced and compactly represented visually through the use of visualisation techniques based on a particular metaphor or a combination of several metaphors. For instance, digital terrain models, based on the landscape metaphor and various geographical frameworks [1, 2] and CAD-based architectural models of cities take the metaphor of urban design into information visualisation. Their popularity has demonstrated that multi-dimensional visualisation can provide a superior means for exploring large data-sets, communicating model results to others and sharing the model [3]. Techniques based on animation and various multimedia support [4] appear frequently in the research and development radar [5]. Dr Hans Rosling[1], a professor in global health at Sweden's Karolinska Institute, has markedly demonstrated the power of animation and visual computing [6, 7] for visually extracting knowledge out of publicly available data, often drawn from United Nations data.

[1] http://www.ted.com/

S.J. Simoff et al. (Eds.): Visual Data Mining, LNCS 4404, pp. 30–45, 2008.

The design of visualisation models for visual data mining, in broad sense, is the definition of the rules for conversion of data into graphics. Generally, the visualisation of large volumes of abstract data, known as 'information visualisation' is closely related but sometimes contrasted, to scientific visualisation, which is concerned with the visualisation of (numerical) data used to form concrete representations [7]. The frequently used expression "the art of visualisation" appropriately describes the state of research in that field. Currently, it is challenging activity for information designers to find out the methods, techniques and corresponding tools available to visualise a particular type of information. The comparison of visualisation techniques across different designs is not a trivial problem [8]. Partially, current situation is explained by the lack of generic criteria to access the value of visualisation models. This is a research challenge, since most people develop their own criteria for what makes a good visual representation. The design of visualisation schemata is dominated by individual points of views, which has resulted in a considerable variety of ad hoc techniques [5]. A recent example is the visualisation of association rules proposed in [9]. On the other hand, an integral part of information visualisation is the evaluation of how well humans understand visualised information as part of their cognitive tasks and intellectual activities, efficiency of information compression and level of cognitive overload. [10] has investigated some aspects of developing visualisation schemata from cognitive point of view.

With the increasing attention towards development of interactive visual data mining methods the development of more systematic approach towards the design of visualisation techniques is getting on the "to do" list of the data mining research. The success of a visual data mining method depends on the development of an adequate computational model of selected metaphor. This is especially important in the context of communicating and sharing of discovered information, and in the context of the emerging methods of computer-mediated collaborative data mining (CMCDM). This new visual data mining methodology is based on the assumption that individuals usually may respond with different interpretations of the same information visualisation [11]. A central issue is the communicative role of abstract information visualisation components in collaborative environments for visual data mining. In fact, "miners" can become part of the visualisation. For example, in virtual worlds this can happen through their avatars[2] [12]. In a virtual world, a collaborative perspective is inevitable, thus a shared understanding of the underlying mapping between the semantic space and the visualisation scheme [13] becomes a necessary condition in the development of these environments. The results of CMCDM are heavily influenced by the adequate formalisation of the metaphors that are used to construct the virtual environment, i.e. the visualisation schemata can influence the behavior of collaborators, the way they interact with each other, the way that they reflect on the changes and manipulations of visualisations, and, consequently, their creativity in the discovery process.

Currently, it is a challenging task for designers of visual data mining environments to find the strategies, methods and corresponding tools to visualise a particular type of information. Mapping characteristics of data into a visual representation in virtual

[2] 3D representations of people and autonomous software agents in virtual worlds. Avatar is an ancient Sanskrit term meaning 'a god's embodiment on the Earth'.

worlds is one promising way to make the discovery of encoded relations in this data possible. The model of semantically organised place for visual data exploration can be useful for the development of computer support for visual information querying and retrieval [14, 15] in collaborative information filtering. The development of a representational and computational model of selected metaphor(s) for data visualisation will assist the design of virtual environments, dedicated to visual data exploration.

The formal approach presented in this chapter is based on the concept of *semantic visualisation* defined as a visualisation method, which establishes and preserves the semantic link between form and function in the context of the visualisation metaphor. Establishing a connection between form and functionality is not a trivial part of the visualisation design process. In a similar way, selecting the appropriate form for representing data graphically, whether the data consists of numbers or text, is not a straightforward procedure as numbers and text descriptions do not have a natural visual representation. On the other hand, how data are represented visually has a powerful effect on how the structure and hidden semantics in the data is perceived and understood. An example of a virtual world, which attempts to visualise an abstract semantic space, is shown in Fig. 1.

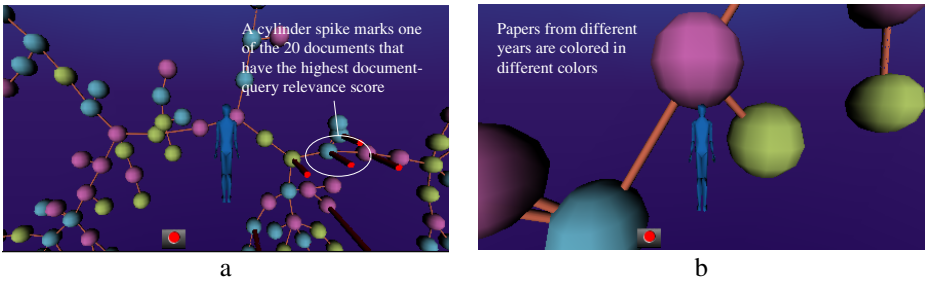


Fig. 1. An example of visualising abstract semantic structure encoded in publication data utilising virtual worlds metaphor and technology (adapted from [16]

Though the work appeared a decade ago, the way the visual representation model has been designed is a typical example of the problems that can be encountered. The visualisation of the semantic space of the domain of human-computer interaction is automatically constructed from a collection of papers from three consecutive ACM CHI conference proceedings [5]. The idea of utilising virtual worlds comes from the research in cognitive psychology, claiming that we develop cognitive (mental) maps in order to navigate within the environments where we operate. Hence the overall "publication" landscape is designed according to the theory of cognitive maps (see [17]in support of the theory of cognitive maps). In such a world, there is a variety of possibilities for data exploration. For example, topic areas of papers are represented by coloured spheres. If a cluster of spheres includes every colour but one, this suggests that the particular topic area, represented by the missing coloured sphere, has not been addressed by the papers during that year. However, without the background knowledge of the semantics of coloured spheres, selected information visualisation scheme does not provide cues for understanding and interpreting the environment landscape. It is not clear how the metaphor of a "landscape" has been formalised and

represented, what are the elements of the landscape. Associatively, this visualisation is closer with the visualisation of molecular structures.

Semantic visualisation is considered in the context of two derivatives of visualisation - *visibilisation* and *visistraction* [18]. Visibilisation is visualisation focusing on the presentation and interpretation which complies with rigorous mapping from physical reality. By contrast, visistraction is the visualisation of abstract concepts and phenomena, which do not have a direct physical interpretation or analogy. Visibilisation has the potential to bring key insights, by emphasising aspects that were unseen before. The dynamic visualisation of the heat transfer during the design of the heat-dissipating tiles cover of the underside of the space-shuttle is an early example of the application of visibilisation [19]. Visistracton can give a graphic depiction of intuition regarding objects and relationships. The 4D simulation of data flow is an example of visistraction, which provides insights impossible without it. In a case-base reasoning system, visistraction techniques can be used to trace the change of relationships between different concepts with the addition of new cases.

Both kinds of semantic visualisation play important role in visual data mining. However, semantic visualisation remains a hand-crafted methodology, where each case is considered separately. This chapter presents an attempt to build a consistent approach to semantic visualisation based on a cognitive model of metaphors, metaphor formalisation and evaluation. We illustrate the application of this approach with examples from visistraction of communication and collaboration data. Further, the chapter is presents the Form-Semantics-Function framework for construction and evaluation of visualisation techniques, an example of the application of the framework towards the construction of a visualisation technique for identifying patterns of team collaboration, and an example of the application of the framework for evaluation and comparison of two visualisation models. The chapter concludes with the issues for building visualization models that support collaborative visual data mining.

2 Form-Semantics-Function: A Formal Approach Towards Constructing and Evaluating Visualisation Techniques

The Form-Semantic-Function (FSF) approach includes the following steps: *metaphor analysis*; *metaphor formalisation*; and *metaphor evaluation*. Through the use of metaphor, people express the concepts in one domain in terms of another domain [20, 21]. The closest analogy is VIRGILIO [22], where the authors proposed a formal approach for constructing metaphors for visual information retrieval. The FSF framework develops further the formal approach towards constructing and evaluating visualisation techniques, approaching the metaphor in an innovative way.

2.1 Metaphor Analysis

During metaphor analysis, the content of the metaphor is established. In the use of metaphor in cognitive linguistics, the terms *source* and *target*[3] refer to the conceptual

[3] In the research literature the target is variously referred to as the primary system or the topic, and the source is often called the secondary system or the vehicle.

spaces connected by the metaphor. The target is the conceptual space that is being described, and the source is the space that is being used to describe the target. In this mapping the structure of the source domain is projected onto the target domain in a way that is consistent with inherent target domain structure [21, 23]. In the context of semantic visualisation, the consistent use of metaphor is expected to bring an understanding of a relatively abstract and unstructured domain in terms of more concrete and structured visual elements through the visualisation schemata.

An extension of the source-target mapping, proposed by [24] includes the notion of generic space and blend space. Generic space contains the skeletal structure that applies to both source and target spaces. The blend space often includes structure not projected to it from either space, namely emergent structure on its own. The ideas and inspirations developed in the blend space can lead to modification of the initial input spaces and change the knowledge about those spaces, i.e. to change and evolve the metaphor. The process is called conceptual blending - it is the essence in the development of semantic visualisation techniques.

In presented approach, the form-semantics-function categorisation of the objects being visualised, is combined with the [24] model. The form of an object can express the semantics of that object, that is, the form can communicate implicit meaning understood through our experiences with that form. From the form in the source space we can connect to a function in the target space via the semantics of the form. The resultant model is shown in Fig. 2.

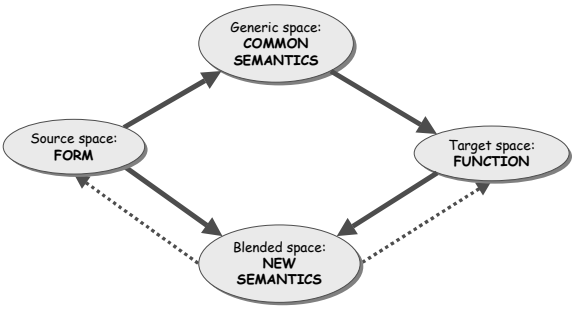


Fig. 2. A model for metaphor analysis for constructing semantic visualisation schemata

The term "visual form" refers to the geometry, colour, texture, brightness, contrast and other visual attributes that characterise and influence the visual perception of an object. Thus, the source space is the space of 2D and 3D shapes and the attributes of their visual representation. "Functions" (the generalisations of patterns, discovered in data) are described using concepts from the subject domain. Therefore the target space includes such concepts associated with the domain functions. This space is constructed from the domain vocabulary. The actual transfer of semantics has two components - the common semantics, which is carried by notions that are valid in both domains and what is referred as new semantics - the blend, which establishes the unique characteristics revealed by the correspondence between the form metaphor and functional characteristics of that form. The schema illustrates how metaphorical inferences produce parallel knowledge structures.

2.2 Metaphor Formalisation

The common perception of the word "formalisation" is connected with the derivation of some formulas and equations that describe the phenomenon in analytical form. In this case, formalisation is used to describe a series of steps that ensure the correctness of the development of the representation of the metaphor. Metaphor formalisation in the design of semantic visualisation schemes includes the following basic steps:

- *Identification of the source and target spaces of the metaphor* - the class of forms and the class of features or functions that these forms will represent;
- *Conceptual decomposition of the source and target spaces* produces the set of concepts that describe both sides of the metaphor mapping. As a rule, metaphorical mappings do not occur isolated from one another. They are sometimes organized in hierarchical structures, in which `lower' mappings in the hierarchy inherit the structures of the `higher' mappings. In other words, this means that visualisation schemes, which use metaphor are expected to preserve the hierarchical structures of the data that they display. In visistraction, these are the geometric characteristics of the forms from the source space, and other form attributes like colours, line thickness, shading, etc. and the set of functions and features in the target space associated with these attributes and variations;
- *Identifying the dimensions of the metaphor* along which the metaphor operates. These dimensions constitute the common semantics. In visistraction this can be for instance key properties of the form, like symmetry and balance with respect to the center of gravity, that transfer semantics to the corresponding functional elements in the target domain;
- *Establishing semantic links, relations and transformations* between the concepts in both spaces, creating a resemblance between the forms in the source domain and the functions in the target domain.

2.3 Metaphor Evaluation

In spite of the large number of papers describing the use of the metaphor in the design of computer interfaces and virtual environments, there is a lack of formal evaluation methods. In the FSF framework metaphor evaluation is tailored following the [25] model, illustrated in Fig. 3.

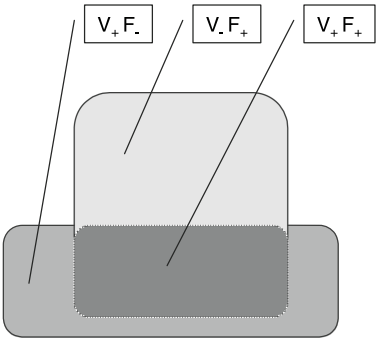


Fig. 3. Model for evaluating metaphor mapping (based on [25])

The "V" and "F" are labels for visualisation and function features, respectively. The "VF" label with indices denotes numbers of features, namely:

- V_+F_+ - function features that are mapped to the visualisation schema;
- V_-F_+ - function features that are not supported by the visualisation schema;
- V_+F_- - features in the visualisation schema, not mapped to the functional features.

The ratio $\frac{V_-F_+}{V_+F_+}$ provides an estimate of the quality of the metaphor used for the visualisation - the smaller the better.

The elements of the Form-Semantics-Function approach are illustrated in the following examples. The first example illustrates metaphor analysis and formalisation for the creation of visualisation form and mapping it to the functional features. In the second example two different forms for visualising the same set of functional features are considered.

3 Constructing Visualisation Schema for Visual Data Mining for Identifying Patterns in Team Collaboration

Asynchronous communication is an intrinsic part of computer-mediated teamwork. Among the various models and tools supporting this communication mode [13], perhaps the most popular in teamwork are bulletin (discussion) boards. These boards support multi-thread discussion, automatically archiving communication content. One way to identify patterns in team collaboration is via content analysis of team communications. However, it is difficult to automate such analysis, therefore, especially in large scale projects, monitoring and analysis of collaboration data can become a cumbersome task.

In the research in virtual design studios [13, 26] there have been identified two extremes (labeled as "Problem comprehension" and "Problem division") in team collaboration, shown in Fig. 4. In "Problem comprehension" collaborative mode the resultant project output - a product, solution to a problem, etc., is a product of a continued attempt to construct and maintain a shared conception and understanding of the problem. In other words each of the participants is developing own view over the whole problem and the shared conception is established during the collaborative process via intensive information exchange.

In "Problem division" mode the problem is divided among the participants in a way where each person is responsible for a particular portion of the investigation of the problem. Thus, it does not necessarily require the creation of a single shared

Fig. 4. Two extremes in team collaboration

conception and understanding of the problem. The two modes of collaboration are two extreme cases. In general, the real case depends on the complexity of the problem.

A key assumption in mining and analysis of collaboration data is that this two extreme styles should be some how reflected in the communication of the teams. Thus, different patterns in team communication on the bulletin board will reflect different collaboration modes. Fig. 5 shows a fragment of a team bulletin board.

Course: Computer Based Design | **Bulletin Board:** Team 2 | **Venue:** Virtual Design Studio

4 Lighting etc - **Derek** *08:46:10 10/16/97* (1)	(M_{14})
• Re: Lighting etc - **Sophie Collins** *10:43:18 10/17/97* (0)	(M_{24})
3 Seating - **Derek Raithby** *15:18:57 10/14/97* (2)	(M_{13})
• Re: Seating - **marky** *17:22:56 10/14/97* (1)	(M_{23})
• Re: Seating - **Sophie Collins** *09:03:27 10/15/97* (0)	(M_{33})
2 Product Research. - **Derek Raithby** *14:37:43 10/14/97* (1)	(M_{12})
• Re: Product Research. - **mark** *17:20:16 10/14/97* (0)	(M_{22})
1 Another idea - **Sophie Collins** *14:24:18 10/14/97* (1)	(M_{11})
• Re: another idea - **Derek Raithby** *14:40:00 10/14/97* (0)	(M_{21})

Fig. 5. Bulletin board fragment with task-related messages, presented as indention graph

The messages on the board are grouped in threads. [27, 28] propose a threefold split of the thread structure of e-mail messages in discussion archives in order to explore the interactive threads. It included (i) reference-depth: how many references were found in a sequence before this message; (ii) reference-width: how many references were found, which referred to this message; and (iii) reference-height: how many references were found in a sequence after this message. In addition to the threefold split, [29] included the time variable explicitly. Fig. 6 shows the formal representation of the bulletin board fragment in Fig. 5.

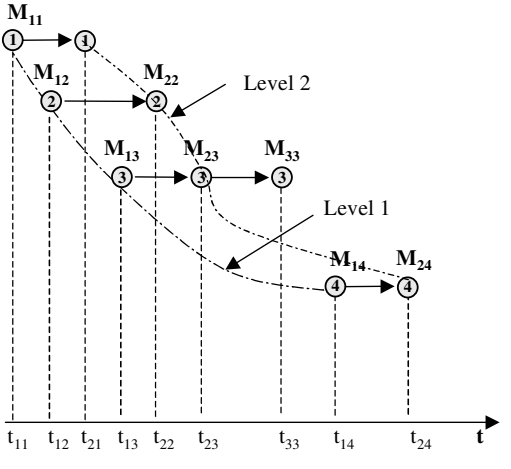


Fig. 6. Formal representation of the thread structure in the fragment presented in Fig. 5

3.1 Metaphor Analysis

Fig. 7 shows the Form-Semantics-Function mapping at the metaphor formalisation stage as a particular case of the [24] model applied to the visualisation of communication utterances data. The source space in this case is the space of 2D geometric shapes, rectangles in particular. The target space includes the concepts associated with the functions that are found in the analysis of a collaborative design session.

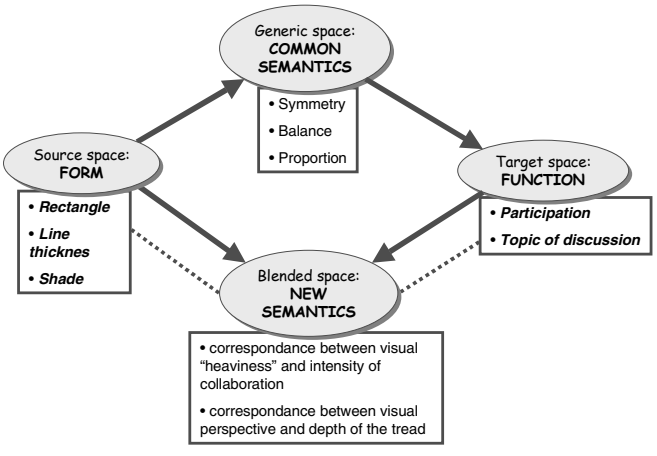


Fig. 7. The Form-Semantics-Function mapping for the source space of nested rectangles and the target space of bulletin board discussion threads

3.2 Metaphor Formalisation

Below is a brief illustration of the metaphor formalisation in this example.

- *Identification of the source and target spaces of the metaphor* - rectangles are the forms that will be used and the class of features or functions that these forms will represent are the messages on a bulletin board;
- *Conceptual decomposition of the source and target spaces* leads to the notion of nested rectangles, whose centers of gravity coincide, with possible variation of the thickness of their contour lines and the background color. Each rectangle corresponds to a message within a thread. Rectangle that corresponds to a message at a level $(n + 1)$ is placed within the rectangle that corresponds to a message at level n. Messages at the same level are indicated by a one step increase of the thickness of the contour line of the corresponding rectangle. Thus, a group of nested rectangles can represent several threads in a bulletin board discussion;
- *Identifying the dimensions of the metaphor* - visual balance and the "depth" or "perspective" of the nested rectangles are the dimension of the metaphor, transferring via the visual appearance the semantics of different communication styles;
- *Establishing semantic links, relations and transformations* - this is connected with the identification of typical form configurations that correspond to typical patterns of collaboration. For example, Fig. 8 illustrates two different fragments A and B (each of one thread). Fig. 9 illustrates the visualisation of this fragments according to the developed visualisation schema.

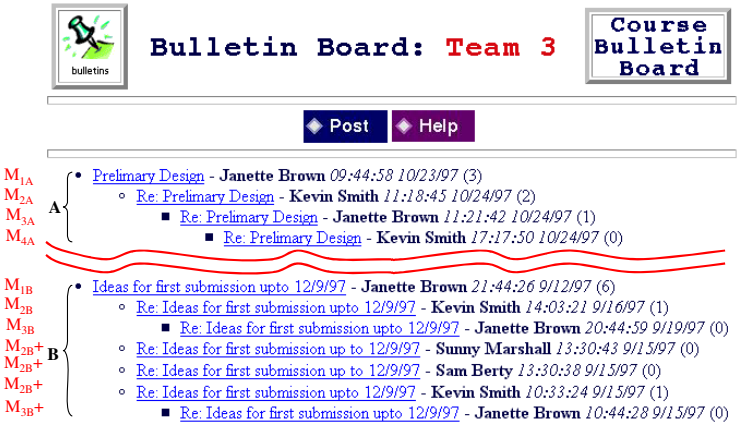


Fig. 8. Bulletin board fragment with task-related messages, presented as indention graph

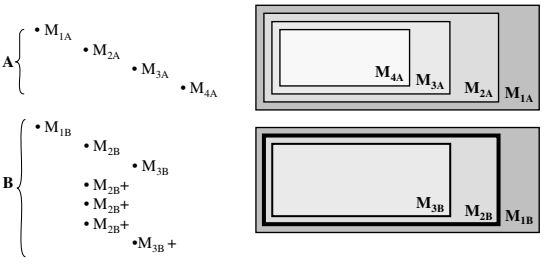
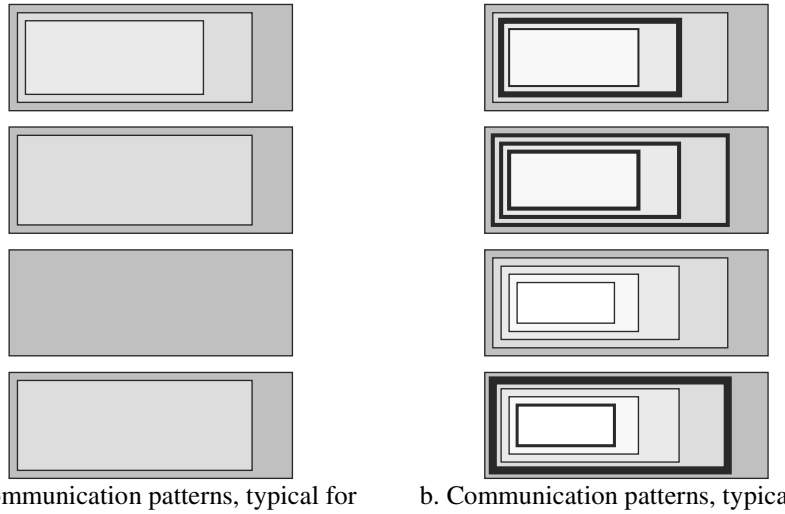


Fig. 9. Visualisation of fragments A and B in Fig. 8



a. Communication patterns, typical for "problem division" collaboration

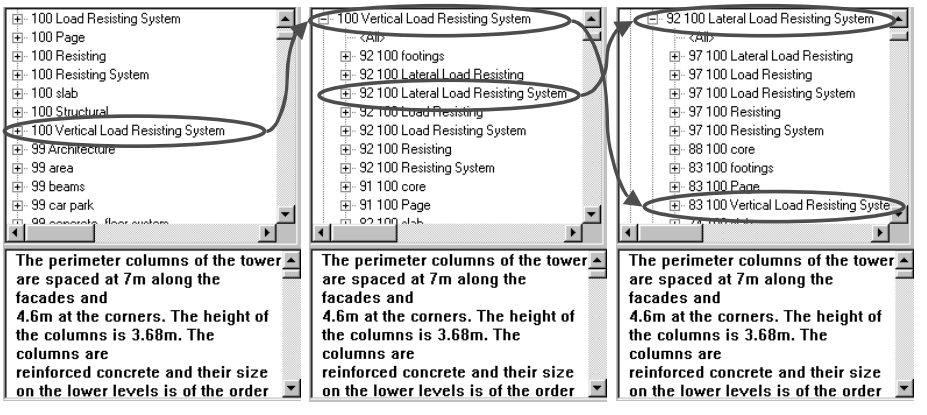
b. Communication patterns, typical for "problem comprehension" collaboration

Fig. 10. Communication patterns, corresponding to different collaboration styles

The visualisation schema has been used extensively in communication analysis. Fig. 10 illustrates communication patterns corresponding to different collaboration styles. An additional content analysis of communication confirmed the correct identification of collaboration patterns.

4 Evaluation and Comparison of Two Visualisation Schemata

We illustrate the idea by evaluating examples of semantic visualisation of textual data and objects in virtual environments. The role of visistraction in concept relationship analyis is to assist the discovery of the relationship between concepts, as reflected in the available text data. The analysis uses word frequencies, their co-occurence and other statistics, and cluster analysis procedures. We investigate two visual metaphors - "Euclidian space" and "Tree", which provide a mapping from the numerical statistics and cluster analysis data into the target space of concepts and relations between them. The visualisation features for both metaphors and the function features of the target space are shown in Table 1. Examples of the two visualisation metaphors operating over the same text data set are shown in Fig. 11 and Fig. 12, respectively.



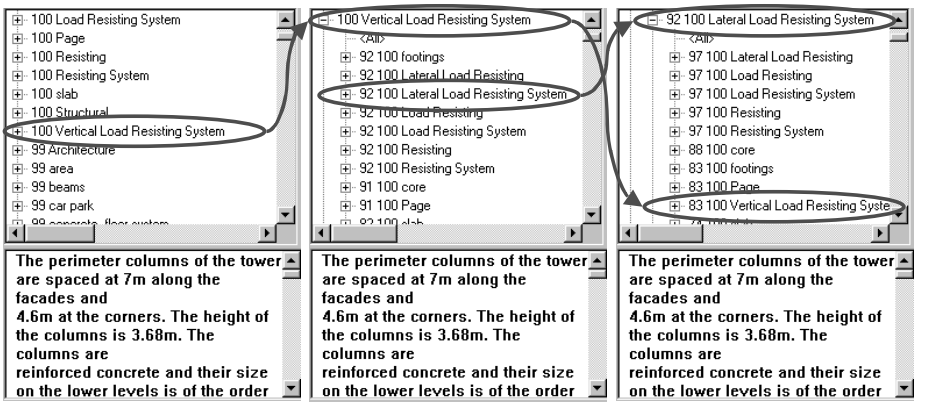


Table 1. Visualisation and function features

Visualisation features of Euclidian space metaphor	Visualisation features of tree metaphor	Function features
- point - alphanumeric single-word point labels - axes - plane - color - line segment	- nodes - alphanumeric multi-word node labels - signs "+" and "-" - branches - numeric labels for branches	- simple/complex concept - subject key word - hierarchical relationship - context link - link strength - synonymy - hyponymy

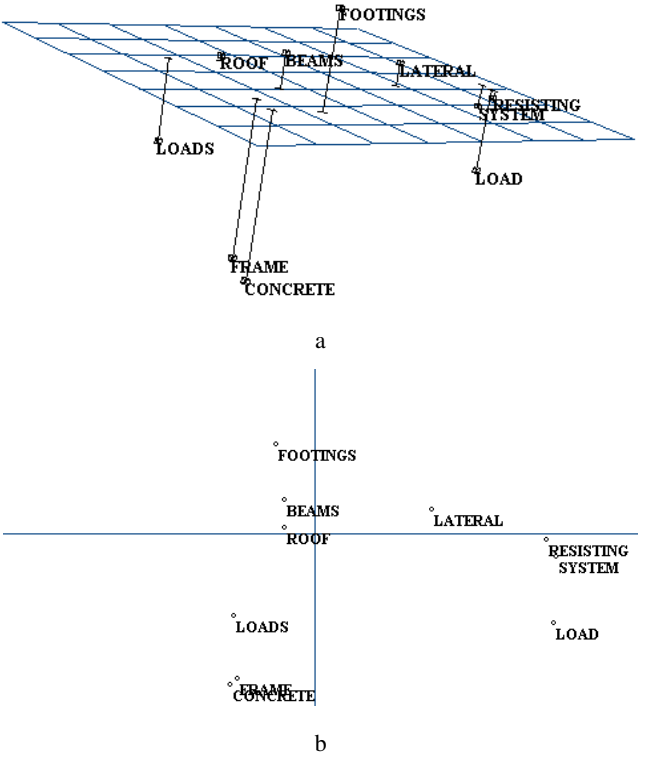


Fig. 11. Visualisation of the 10 most frequent words in the description of lateral load resisting system in one of the wide-span building cases[4]

[4] This visualisation is used in TerraVision, which is part of the CATPAC system for text analysis by Provalis Research Co.

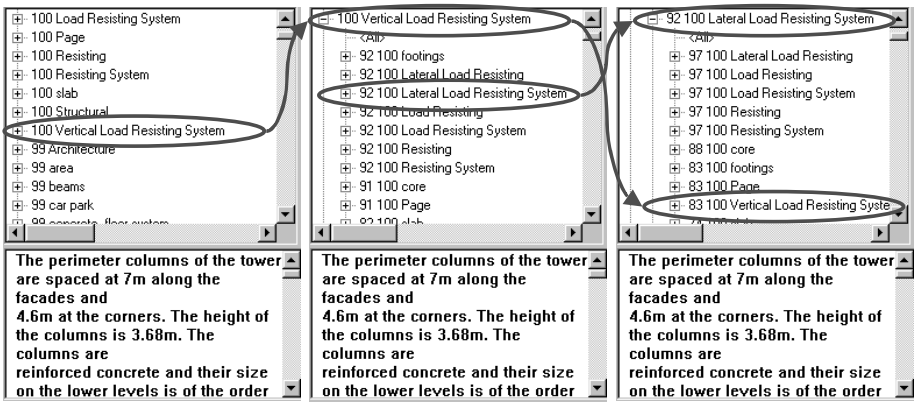


Fig. 12. Visualisation of part of a semantic net[5] for the term "load resisting system"

The source text is the collection of building design cases, which is part of the SAM (Structure and Materials) case library and case-based reasoning system [30].

4.1 Metaphor Evaluation

The first scheme[6] maps the source domain of Euclidean space (coordinates of points in 2D/3D space) to the target domain of word statistics. The blending semantics is that the degree, to which the terms are related to each other, can be perceived visually from the distance between the corresponding data points - the closer the points the tighter is the relationship between the words. The second scheme[7] maps the source domain of the topology of linked nodes to the same target domain of words statistics. This mapping generates one of the possible visualisations of semantic networks. This visualisation includes nodes with single- and multiple-word labels, numeric values of

Table 2. Visualisation support for function features in Euclidean space and tree metaphors

Function features	**Support by the Euclidean space metaphor**	**Support by the Tree metaphor**
Simple/complex concept	-	+
Subject key word	+	+
Hierarchical relationship	-	+
Context link	-	+
Link strength	-	+
Synonymy	-	-
Hyponymy	-	-

[5] Semantic networks in our study are visualised in TextAnalyst by Megaputer Intelligence, Inc.
[6] The schema is used in the CATPAC qualitative analysis package by Terra Research Inc.
[7] The schema is used in TextAnalyst by Megaputer Intelligence (see www.megaputer.com).

each link between terms and the weight of the term among the other terms in the tree. The results of the comparison between the two metaphors are presented in Table 2 and Table 3. The Euclidean space metaphor has a poor performance for visualisation of concept relationships. What is the meaning of such closeness - it is difficult to make a steady judgement about what the relation is and whether we deal with simple (one word) or complex (more than one word) terms. The distance to the surface, proportional to the frequency of the words, can convey the message that a word is a key word. However, there is no feature in the visualisation, which shows context links between words, the strength of this links and other relations between words.

Table 3. Comparison of in Euclidean space and tree metaphors

	Euclidean space metaphor	**Tree metaphor**
V_+F_+	1	5
V_-F_+	6	2
$\frac{V_-F_+}{V_+F_+}$	6	0.4

5 Conclusion and Future Directions

The Form-Semantics-Function framework resented in this work is an attempt to develop a formal approach towards the use of metaphors in constructing consistent visualisation schemes. In its current development, the evaluation part of the framework does not include the analysis of the cognitive overload from the point of information design. Some initial work in that direction has been started in [31].

Currently the research on the FSF framework is further developed in the context of supporting collaborative visual data mining in virtual worlds. The different perceptions of a visualisation model in such environment may increase the gap between individuals as they interact with it in a data exploration session. However, individual differences may lead to a potential variety of discovered patterns and insights in the visualised information across participants. Consequently, current research within the FSF framework is focused on exploring:

- whether people attach special meanings to abstract visualisation objects;
- what are the design criteria towards visualisation objects, engaged in visual data exploration, that people can effectively construct and communicate knowledge in visual data mining environments;
- what are the necessary cues that should be supported in semantically organised virtual environments;
- how individual differences in visual perspectives can be channelled to stimulate the creation of “out of the box” innovative perspectives.

References

1. Hetzler, B., Harris, W.M., Havre, S., Whitney, P.: Visualising the full spectrum of document relationships, in Structures and Relations in Knowledge Organisation. In: Proceedings of the Fifth International Society for Knowledge Organization (ISKO) Conference, Lille, France (1998)
2. Hetzler, B., Whitney, P., Martucci, L., Thomas, J.: Multi-faceted insight through interoperable visual information analysis paradigms. In: Proceedings of the 1998 IEEE Symposium on Information Visualization. IEEE Computer Society, Washington, DC (1998)
3. Brown, I.M.: A 3D user interface for visualisation of Web-based data-sets. In: Proceedings of the 6th ACM International Symposium on Advances in Geographic Information Systems. ACM, Washington, D.C (1998)
4. Noirhomme-Fraiture, M.: Multimedia support for complex multidimensional data mining. In: Proceedings of the First International Workshop on Multimedia Data Mining (MDM/KDD 2000), in conjunction with Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD 2000. ACM Press, Boston (2000)
5. Chen, C.: Information Visualization: Beyond the Horizon. Springer, London (2004)
6. Gross, M.: Visual Computing: The Integration of Computer Graphics. Springer, Heidelberg (1994)
7. Nielson, G.M., Hagen, H., Muller, H.: Scientific Visualization: Overviews, Methodologies, and Techniques. IEEE Computer Society, Los Alamitos (1997)
8. Chen, C., Yu, Y.: Empirical studies of information visualization: A meta-analysis. International Journal of Human-Computer Studies 53(5), 851–866 (2000)
9. Hofmann, H., Siebes, A.P.J.M., Wilhelm, A.F.X.: Visualizing association rules with interactive mosaic plots. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD 2000. ACM, Boston (2000)
10. Crapo, A.W., Waisel, L.B., Wallace, W.A., Willemain, T.R.: Visualization and the process of modeling: A cognitive-theoretic approach. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD 2000. ACM, New York (2000)
11. Snowdon, D.N., Greenhalgh, C.M., Benford, S.D.: What You See is Not What I See: Subjectivity in virtual environments. In: Proceedings Framework for Immersive Virtual Environments (FIVE 1995). QMW University of London, UK (1995)
12. Damer, B.: Avatars. Peachpit Press, an imprint of Addison Wesley Longman (1998)
13. Maher, M.L., Simoff, S.J., Cicognani, A.: Understanding virtual design studios. Springer, London (2000)
14. Del Bimbo, A.: Visual Information Retrieval. Morgan Kaufmann Publishers, San Francisco (1999)
15. Gong, Y.: Intelligent Image Databases: Towards Advanced Image Retrieval. Kluwer Academic Publishers, Boston (1998)
16. Börner, K., Chen, C., Boyack, K.: Visualizing knowledge domains. Annual Review of Information Science &Technology, 179–355 (2003)
17. Kumaran, D., Maguire, E.A.: The human hippocampus: Cognitive maps or relational memory? The Journal of Neuroscience 25(31), 7254–7259 (2005)
18. Choras, D.N., Steinmann, H.: Virtual reality: Practical applications in business and industry. Prentice-Hall, Upper Saddle River (1995)
19. Gore, R.: When the space shuttle finally flies. National Geographic 159, 317–347 (1981)
20. Lakoff, G., Johnson, M.: Metaphors We Live By. University of Chicago Press, Chicago (1980)

21. Lakoff, G.: The contemorary theory of metaphor, in Metaphor and Thought. In: Ortony, A. (ed.), pp. 202–251. Cambridge University Press, Cambridge (1993)
22. L'Abbate, M., Hemmje, M.: VIRGILIO - The metaphor definition tool, in Technical Report: rep-ipsi-1998-15. 2001, European Research Consortium for Informatics and Mathematics at FHG (2001)
23. Turner, M.: Design for a theory of meaning. In: Overton, W., Palermo, D. (eds.) The Nature and Ontogenesis of Meaning, pp. 91–107. Lawrence Erlbaum Associates, Mahwah (1994)
24. Turner, M., Fauconnier, G.: Conceptual integration and formal expression. Journal of Metaphor and Symbolic Activity 10(3), 183–204 (1995)
25. Anderson, B., Smyth, M., Knott, R.P., Bergan, M., Bergan, J., Alty, J.L.: Minimising conceptual baggage: Making choices about metaphor. In: Cocton, G., Draper, S., Weir, G. (eds.) People and Computers IX, G, pp. 179–194. Cambridge University Press, Cambridge (1994)
26. Maher, M.L., Simoff, S.J., Cicognani, A.: Potentials and limitations of virtual design studios. Interactive Construction On-Line 1 (1997)
27. Berthold, M.R., Sudweeks, F., Newton, S., Coyne, R.: Clustering on the Net: Applying an autoassociative neural network to computer-mediated discussions. Journal of Computer Mediated Communication 2(4) (1997)
28. Berthold, M.R., Sudweeks, F., Newton, S., Coyne, R.: It makes sense: Using an autoassociative neural network to explore typicality in computer mediated discussions. In: Sudweeks, F., McLaughlin, M., Rafaeli, S. (eds.) Network and Netplay: Virtual Groups on the Internet, pp. 191–220. AAAI/MIT Press, Menlo Park, CA (1998)
29. Sudweeks, F., Simoff, S.J.: Complementary explorative data analysis: The reconciliation of quantitative and qualitative principles. In: Jones, S. (ed.) Doing Internet Research, pp. 29–55. Sage Publications, Thousand Oaks (1999)
30. Simoff, S.J., Maher, M.L.: Knowledge discovery in hypermedia case libraries - A methodological framework. In: Proceedings of the Fourth Australian Knowledge Acquisition Workshop AKAW 1999, in conjunction with 12th Australian Joint Conference on Artificial Intelligence, AI 1999, Sydney, Australia (1999)
31. Chen, C.: An information-theoretic view of visual analytics. IEEE Computer Graphics and Applications 28(1), 18–23 (2008)

A Methodology for Exploring Association Models*

Alipio Jorge[1], João Poças[2], and Paulo J. Azevedo[3]

[1] LIACC/FEP, Universidade do Porto, Portugal
amjorge@liacc.up.pt
[2] Instituto Nacional de Estatística, Portugal
joao.pocas@ine.pt
[3] Departamento de Informática, Universidade do Minho, Portugal
pja@di.uminho.pt

Abstract. Visualization in data mining is typically related to data exploration. In this chapter we present a methodology for the post processing and visualization of association rule models. One aim is to provide the user with a tool that enables the exploration of a large set of association rules. The method is inspired by the hypertext metaphor. The initial set of rules is dynamically divided into small comprehensible sets or pages, according to the interest of the user. From each set, the user can move to other sets by choosing one appropriate operator. The set of available operators transform sets of rules into sets of rules, allowing focusing on interesting regions of the rule space. Each set of rules can also be then seen with different graphical representations. The tool is web-based and dynamically generates SVG pages to represent graphics. Association rules are given in PMML format.

1 Introduction

Visualization techniques are mainly popular in data mining and data analysis for data exploration. Such techniques try to solve problems such as the dimensionality curse [13], and help the data analyst in easily detecting trends or clusters in the data and even favour the early detection of bugs in the data collection and data preparation phases. However, not only the visualization of data can be relevant in data mining. Other two important fields for visual data mining are the graphical representation of data mining models, and the visualization of the data mining process in a visual programming style [6].

The visualization of models in data mining potentially increases the comprehensibility and allows the post processing of those models. In this chapter, we describe a tool and methodology for the exploration/post processing of large sets of association rules. Small sets of rules are shown to the user according to preferences the user states implicitly. Numeric properties of the rules in each rule subset are also graphically represented.

* This work is supported by the European Union grant IST-1999-11.495 Sol-Eu-Net and the POSI/2001/Class Project sponsored by Fundação Ciência e Tecnologia, FEDER e Programa de Financiamento Plurianual de Unidades de I & D.

S.J. Simoff et al. (Eds.): Visual Data Mining, LNCS 4404, pp. 46–59, 2008.

This environment also takes advantage of PMML (Predictive Model Markup Language) being proposed as a standard by the Data Mining Group [6]. This means that any data mining engine producing association rules in PMML can be coupled with the tool being proposed. Moreover, this tool can be easily used simultaneously with other post processing tools that read PMML, for the same problem.

Association Rule (AR) discovery [1] is many times used in data mining applications like market basket analysis, marketing, retail, study of census data, design of shop layout, among others [e.g., 4, 6, 7, 10]. This type of knowledge discovery is particularly adequate when the data mining task has no single concrete objective to fulfil (such as how to discriminate good clients from bad ones), contrarily to what happens in classification or regression. Instead, the use of AR allows the decision maker/ knowledge seeker to have many different views on the data. There may be a set of general goals (like "what characterizes a good client?", "which important groups of clients do I have?", "which products do which clients typically buy?"). Moreover, the decision maker may even find relevant patterns that do not correspond to any question formulated beforehand. This style of data mining is sometimes called "fishing" (for knowledge), or undirected data mining [3].

Due to the data characterization objectives, association rule discovery algorithms produce a complete set of rules above user-provided thresholds (typically minimal support and minimal confidence, defined in Section 2). This implies that the output is a very large set of rules, which can easily get to the thousands, overwhelming the user. To make things worse, the typical association rule algorithm outputs the list of rules as a long text (even in the case of commercial tools like SPSS Clementine), and lacks post processing facilities for inspecting the set of produced rules.

In this chapter we propose a method and tool for the browsing and visualization of association rules. The tool reads sets of rules represented in the proposed standard PMML [6]. The complete set of rules can then be browsed by applying operators based on the generality relation between itemsets. The set of rules resulting from each operation can be viewed as a list or can be graphically summarized.

This chapter is organized as follows: we introduce the basic notions related to association rule discovery, and association rule space. We then describe PEAR (Post-processing Environment for Association Rules), the post processing environment for association rules. We describe the set of operators and show one example of the use of PEAR, and then proceed to related work and conclusion.

2 Association Rules

An association rule $A \rightarrow B$ represents a relationship between the sets of items A and B. Each item I is an atom representing a particular object. The relation is characterized by two measures: support and confidence of the rule. The support of a rule R within a dataset D, where D itself is a collection of sets of items (or itemsets), is the number of transactions in D that contain all the elements in $A \cup B$. The confidence of the rule is the proportion of transactions that contain $A \cup B$ with respect to the transactions containing A. Each rule represents a pattern captured on the data. The support is the commonness of that pattern. The confidence measures its predictive ability.

The most common algorithm for discovering AR from a dataset *D* is APRIORI [1]. This algorithm produces all the association rules that can be found from a dataset *D* above given values of support and confidence, usually referred to as *minsup* and *minconf*. APRIORI has many variants with more appealing computational properties, such as PARTITION [18] or DIC [3], but that should produce exactly the same set of rules as determined by the problem definition and the data.

In this work we used Caren (Classification and Association Rules ENgine) [2], a java based implementation of APRIORI. This association rule engine optionally outputs derived models in PMML format, besides Prolog, ASCII, and CSV.

2.1 The Association Rule Space

The space of itemsets *I* can be structured in a lattice with the $\subseteq$ relation between sets. The empty itemset $\varnothing$ is at the bottom of the lattice and the set of all itemsets at the top. The $\subseteq$ relation also corresponds to the generality relation between itemsets.

To structure the set of rules, we need a number of lattices, each corresponding to one particular itemset that appears as the antecedent, or to one itemset that occurs as a consequent. For example, the rule $\{a,b\} \rightarrow \{c,d\}$, belongs to two lattices: the one of the rules with antecedent $\{a,b\}$, structured by the generality relation over the consequent, and the lattice of rules with $\{c,d\}$ as a consequent, structured by the generality relation over the antecedents of the rules (Figure 1).

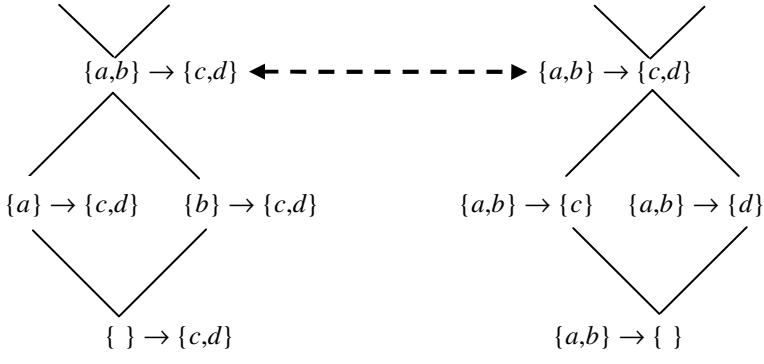


Fig. 1. The two lattices of rule $\{a,b\} \rightarrow \{c,d\}$

We can view this collection of lattices as a grid, where each rule belongs to one intersection of two lattices. The idea behind the rule browsing approach we present, is that the user can visit one of these lattices (or part of it) at a time, and take one particular intersection to move into another lattice (set of rules).

3 PEAR: A Web-Based AR Browser

To help the user browsing a large set of rules and ultimately find the subset of interesting rules, we developed PEAR (Post processing Environment for Association

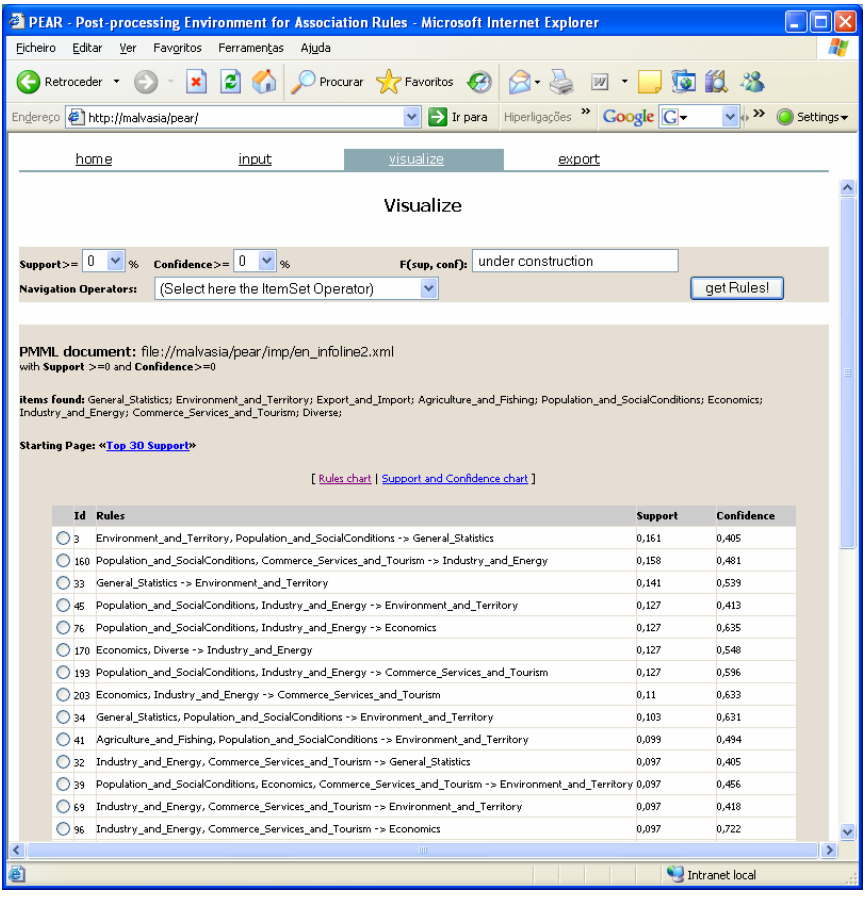


Fig. 2. PEAR screen showing some rules. On the top we have the "Support>=", "Confidence>=" and user defined metric ("F(sup,conf)") parameter boxes. The "Navigation Operators" box is used to pick one operator from a pre-defined menu. The operator is then applied to the rule selected by clicking the respective circle just before the Id. When the "get Rules!" button is pressed, the resulting rules appear, and the process may be iterated.

Rules) [10]. PEAR implements the set of operators described below that transform one set of rules into another, and allows a number of visualization techniques. PEAR's server is run under an http server. A client is run on a web browser. Although not currently implemented, multiple clients can potentially run concurrently.

PEAR operates by loading a PMML representation of the rule set. This initial set is displayed as a web page (Figure 2). From this page the user can go to other pages containing ordered lists of rules with support and confidence. All the pages are dynamically generated during the interaction of the user with the tool. To move from page (set of rules) to page, the user applies restrictions and operators. The restrictions can be done on the minimum support, minimum confidence, or on functions of the support and confidence of the itemsets in the rule. Operators can be selected from a

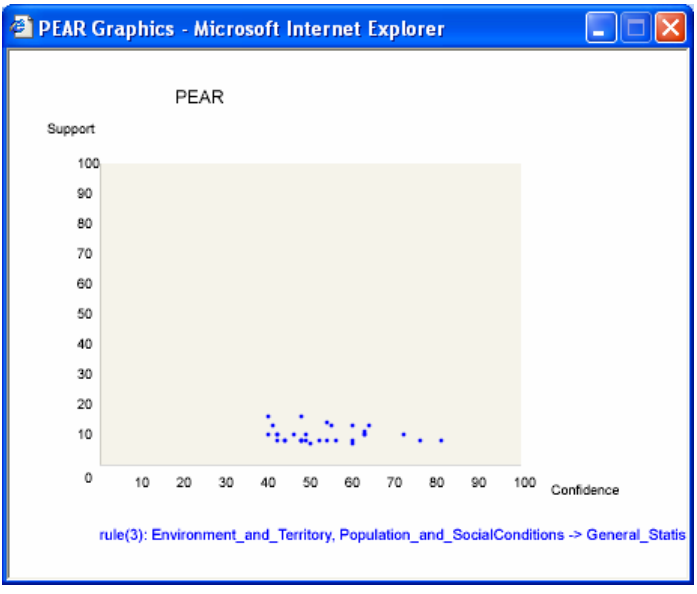


Fig. 3. PEAR plotting support × confidence points for a subset of rules. The rule is identified when the mouse flies over the respective x-y point. On the chart above, the selected point is for the rule with Id 3.

list. If it is a $\{Rule\} \rightarrow \{Sets\ of\ Rules\}$ operator, the input rule must also be selected.For each page, the user can also select a graphical visualization that summarizes the set of rules on the page. Currently, the available visualizations are Confidence × Support x-y plot (Figure 3) and Confidence / support histograms (Figure 4). The produced charts are interactive and indicate the rule that corresponds to the point under the mouse.

4 Chunking Large Sets of Rules

Our methodology is based on the philosophy of web browsing, page by page following hyperlinks. The ultimate aim of the user is to find interesting rules in the large rule set as easily as possible. For that, the set R of derived rules must be divided into small subsets that are presented as pages, and can be perceived by the user. In this sense, small means a rule set that can be seen in one screen (maximum 20 to 50 rules). Each page then presents some hyperlinks to other pages (other small sets of rules) and visual representations.

The first problem to be solved is *how to divide a set of rules into pages*? Our current approach is to start with a set of rules that presents some diversity, and then use operators that link one rule in the current subset to another subset of rules. Currently proposed operators allow focusing on the neighbourhood of the selected rule. Other operators may have other effects.

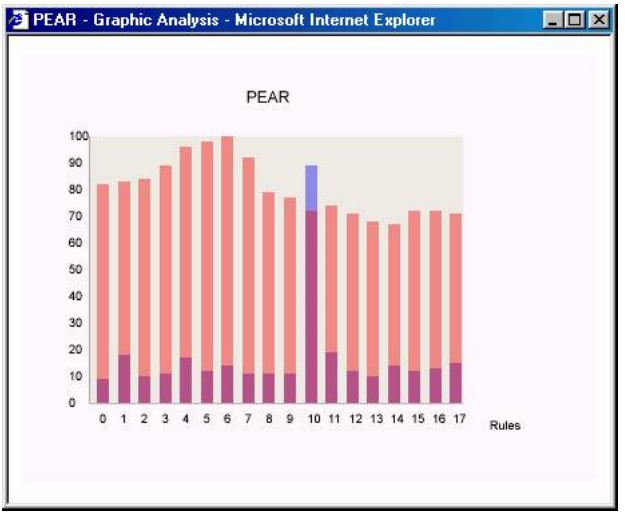


Fig. 4. PEAR showing a multi-bar histogram. Each bar represents the confidence (lighter color) and the support (super-imposed darker color) of one rule. Again, the rule can be identified by flying over the respective bar with the mouse.

The second problem is *how easily interesting rules can be found*? Since the user is searching for interesting rules, each page should include indications of the interest of the rules included. Besides the usual local metrics for each rule, such as confidence and support, global metrics can be provided. This way, when the user follows a hyperlink from one set of rules to another, the evolution of such metrics can be monitored. Below we will make some proposals of global metrics for association rule sets.

The third problem is *how to ensure that any rule can be found from the initial set of rules*. If the graph of rules defined by the available operators is connected, then this is satisfied for any set of initial rules. Otherwise, each connected subgraph must have one rule in the initial set.

4.1 Global Metrics for Sets of Rules

To compare the interest of different sets of rules, we need to numerically characterize each set of rules as an individual entity. This naturally suggests the need of global metrics (measuring the value of a set of rules), in addition to local metrics (characterizing individual rules). Each global metric provides a partial ordering on the family of rule sets.

The value of a set of rules can be measured in terms of *diversity* and *strength*. The diversity of a set of rules may be measured, for instance, by the number of items involved in the rules (item coverage). Example coverage is also a relevant diversity measure.

The strength of a set of rules is related to the actionability of the rules. One simple example is the *accuracy* that a set of association rules obtains on the set of known

examples, when used as a classification model. Other global measures of strength may be obtained by combining local properties of the rules in the set. One example of such a measure is the weighted χ^2 introduced in [13].

4.2 The Index Page

To start browsing, the user needs an index page. This should include a subset of the rules that summarize the whole set. In other words: a set of rules with high diversity. In terms of web browsing, it should be a small set of rules that allows getting to any page in a limited number of clicks. For example, a candidate for such a set could be the smallest rule for each consequent. Each of these rules would represent the lattice on the antecedents of the rules with the same consequent. Since the lattices intersect, we can change to a focus on the antecedent on any rule by applying an appropriate operator.

Similarly, we could start with the set of smallest rules for each antecedent. Alternatively, instead of the size, we could consider the support, confidence, or other measure. All these possibilities must be studied and some of them implemented in our system, which currently shows, as the initial page, the set of all rules.

Another possibility for defining the starting set of rules, is to divide the whole set of rules into clusters with rules involving similar items. Then, a representative rule from each cluster is chosen. The set of representative rules is the starting page. The number of rules can be chosen according to the available screen space. In [10], hierarchical clustering is used to adequately divide a set of association rules. The representative rules are the closest to the centroids of each group of rules.

5 Operators for Sets of Association Rules

The association rule browser helps the user to navigate through the space of rules by viewing one set of rules at a time. Each set of rules corresponds to one page. From one given page the user moves to the following by applying a selected operator to all or some of the rules viewed on the current page. In this section we define the set of operators to apply to sets of association rules.

The operators we describe here transform one single rule $R \in \{Rules\}$ into a set of rules $RS \in \{Sets\ of\ Rules\}$ and correspond to the currently implemented ones. Other operators may transform one set of rules into another. In the following we describe the operators of the former class.

Antecedent generalization

$$AntG(A \rightarrow B) = \{A' \rightarrow B \mid A' \subseteq A\}$$

This operator produces rules similar to the given one but with a syntactically simpler antecedent. This allows the identification of relevant or irrelevant items in the current rule. In terms of the antecedent lattice, it gives all the rules below the current one with the same consequent.

Antecedent least general generalization

$AntLGG(A \rightarrow B) = \{A' \rightarrow B \mid A'$ is obtained by deleting one atom in $A\}$

This operator is a stricter version of the *AntG*. It gives only the rules on the level of the antecedent lattice immediately below the current rule.

Consequent generalization

$ConsG(A \rightarrow B) = \{A \rightarrow B' \mid B' \subseteq B\}$

Consequent least general generalization

$ConsLGG(A \rightarrow B) = \{A \rightarrow B' \mid B'$ is obtained by deleting one atom in $B\}$

Similar to *AntG* and *AntLGG* respectively, but the simplification is done on the consequent instead of on the antecedent.

Antecedent specialization

$AntS(A \rightarrow B) = \{A' \rightarrow B \mid A' \supseteq A\}$

This produces rules with lower support but finer detail than the current one.

Antecedent least specific specialization

$AntLSS(A \rightarrow B) = \{A' \rightarrow B \mid A'$ is obtained by adding one (any) atom to $A\}$

As *AntS*, but only for the immediate level above on the antecedent lattice.

Consequent specialization

$ConsS(A \rightarrow B) = \{A \rightarrow B' \mid B' \supseteq B\}$

Consequent least specific specialization

$ConsLSS(A \rightarrow B) = \{A \rightarrow B' \mid$ B' is obtained by adding one (any) atom to $B\}$

Similar to *AntS* and *AntSS*, but on the consequent.

Focus on antecedent

$FAnt(A \rightarrow B) = \{A \rightarrow C \mid C$ is any$\}$

Gives all the rules with the same antecedent. $FAnt(R) = ConsG(R) \cup ConsS(R)$.

Focus on consequent

$FCons(A \rightarrow B) = \{C \rightarrow B \mid C \text{ is any}\}$

Gives all the rules with the same consequent. $FCons(R) = AntG(R) \cup AntS(R)$.

6 Example of the Application of the Proposed Methodology

We now describe how the method being proposed can be applied to the analysis of downloads from the site of the Portuguese National Institute of Statistics (INE). This site (www.ine.pt/infoline) serves as an electronic store, where the products are tables in digital format with statistics about Portugal.

From the web access logs of the site's http server we produced a set of association rules relating the main thematic categories of the downloaded tables. This is a relatively small set set of rules (211) involving 9 items that serves as an illustrative example. The aims of INE are to improve the usability of the site by discovering which items are typically combined by the same user. The results obtained can be used in the restructuring of the site or in the inclusion of recommendation links on some pages. A similar study could be carried out for lower levels of the category taxonomy.

The rules in Figure 5 show the contents of one index page, with one rule for each consequent (from the 9 items, only 7 appear). The user then finds the rule on "Territory_an_Environment" relevant for structuring the categories on the site. By applying the ConsG operator, she can drill down the lattice around that rule, obtaining all the rules with a generalized antecedent.

Rule	Sup	Conf
Economics_and_Finance <= Population_and_Social_Conditions & Industry_and_Energy & External_Commerce	0,038	0,94
Commerce_Tourism_and_Services <= Economics_and_Finance & Industry_and_Energy & General_Statistics	0,036	0,93
Industry_and_Energy <= Economics_and_Finance & Commerce_Tourism_and_Services & General_Statistics	0,043	0,77
Territory_and_Environment <= Population_and_Social_Conditions & Industry_and_Energy & General_Statistics	0,043	0,77
General_Statistics <= Commerce_Tourism_and_Services & Industry_and_Energy & Territory_and_Environment	0,040	0,73
External_Commerce <= Economics_and_Finance & Industry_and_Energy & General_Statistics	0,036	0,62
Agriculture_and_Fishing <= Commerce_Tourism_and_Services & Territory_and_Environment & General_Statistics	0,043	0,51

Fig. 5. First page (index)

From here, we can see that "Population_and_Social_Conditions" is not relevantly associated to "Territory_and_Environment". The user can now, for example, look into rules with "Population_and_Social_Conditions" by applying the FAnt (focus on antecedent) operator (results not shown here). From there she could see what the main associations to this item are.

Rule	Sup	Conf
Territory_and_Environment <= Population_and_Social_Conditions & Industry_and_Energy & General_Statistics	0,043	0,77
Territory_and_Environment <= Population_and_Social_Conditions & Industry_and_Energy	0,130	0,41
Territory_and_Environment <= Population_and_Social_Conditions & General_Statistics	0,100	0,63
Territory_and_Environment <= Industry_and_Energy & General_Statistics	0,048	0,77
Territory_and_Environment <= General_Statistics	0,140	0,54

Fig. 6. Applying the operator ConsG (consequent generalization)

The process would then iterate, allowing the user to follow particular interesting threads in the rule space. Plots and bar charts summarize the rules in one particular page. The user can always return to an index page. The objective is to gain insight on the rule set (and on the data) by examining digestible chunks of rules. What is an interesting or uninteresting rule depends on the application and the knowledge of the user. For more on measures of interestingness see [20, 21].

7 Implementation

Currently, PEAR server runs under Microsoft Microsoft Internet Information Server (IIS), but is browser independent. The server can also run on any PC with a Microsoft OS using a Personal WebServer.

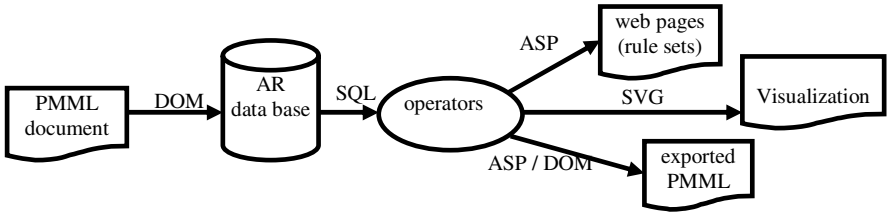


Fig. 7. General architecture of PEAR

On the server side, Active Server Pages (ASP) allow the definition of dynamic and interactive web pages [17]. These integrate html code with VbScript and implement various functionalities of the system. JavaScript, is used for data manipulation on the client-side, for its portability. With JavaScript we create and manipulate PMML or SVG (Scalable Vector Graphics) [5] documents using Document Object Model (DOM). Both PMML and SVG are XML (eXtensible Markup Language) documents.

We use the DOM to read and manipulate the original PMML document (XML document that represents a data mining model), to export a new PMML document and also to create and manipulate the graphical visualization [23]. Interactive graphical visualizations in PEAR are implemented as Scalable Vector Graphics (SVG) [24]. This is an XML-based language that specifies vector graphics that can be visualized by a web browser.

7.1 Scalable Vector Graphics

Scalable Vector Graphics (SVG) is an XML-based language that specifies and defines two dimensional vector graphics that can be visualized by a web browser (Table 1). In PEAR, to produce a visualization of a set of rules, the user clicks on a hyperlink shown on the browser. On the server side, an XML/SVG file is dynamically generated. With the appropriate plug-in, freely available, the browser transforms the SVG source to a graphical representation. Using SVG distributes the work load between server and client and represents a relatively small data transfer.

Table 1. Example of an SVG source

```
<?xml version="1.0" ?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20000802//EN"
"http://www.w3.org/TR/2000/CR-SVG-20000802/DTD/svg-20000802.dtd">

<svg width="250" height="200">
     <path d="M 30 0 L 100 100" />
     <text x="20" y="80" style="font-size:40; font-weight:400; font-
     family:Verdana; font-style:italic; fill:red">Texto</text>
</svg>
```

7.2 Representing Associations Rules with PMML

Predictive Model Markup Language (PMML) is an XML-based language. A PMML document provides a complete non-procedural definition of fully trained data mining models. This way, models can be shared between different applications. The universal, extensible, portable and human readable character of PMML allows users to develop models within one application, and use other applications to visualize, analyze, evaluate or otherwise use the models.

PEAR can read an AR model specified in a PMML document. The user will be able to manipulate the AR model, creating a new rule space based on a set of operators, and export a subset of selected rules to a new PMML document. Internally, rules are stored in a relational database. Operators are implemented as SQL queries.

7.3 Performance

Experiments with PEAR showed good response times when operators are applied to sets with more than 6000 rules (2 seconds locally (server and client in the same computer), 4 seconds remotely (server and client in separate computers)). The times for loading the PMML file to the relational database are still too high for such a number of rules (above 10 minutes) [18]. This deserves some investigation on the implementation techniques used, namely the DOM and the interface with the relational database.

8 Related Work

One of the first proposals (1994)for the visual browsing of association rule sets was the Rule Visualizer [14]. In this proposed tool, association rules could be viewed under different perspectives, such as the rule graph, the rule browser and the rule selector. This tool, however, does not seem to have had any more recent developments.

The system DS-WEB [10] uses the same sort of approach as the one we propose here. DS-WEB and PEAR have both the aim of post processing a large set of AR through web browsing and visualization. DS-WEB relies on the presentation of a reduced set of rules, called direction setting or DS rules, and then the user can explore the variations of each one of these DS rules. In our approach, we rely on a set of general operators that can be applied to any rule, including DS rules as defined for DS-WEB. The set of operators we define is based on simple mathematical properties of

the itemsets and have a clear and intuitive semantics. PEAR also has the additional possibility of reading AR models as PMML.

VisWiz is the non-official name for a PMML interactive model visualizer implemented in Java [25]. It displays graphically many sorts of data mining models. AR are presented as an ordered list together with color bars to indicate the values of measures. The user sets the minimal support and confidence through very intuitive gauges. This visualizer can be used directly in a web browser as a java plug-in.

Lent et al [12] describe an approach to the clustering of association rules. The aim is to derive information about the grouping of rules obtained from clustering. As a consequence one can replace clustered rules by one more general rule. For a given attribute in the consequent, the proposed algorithm constructs a 2D grid where each axis corresponds to an attribute in the antecedent. The algorithm tries to find "the best" clustering of rules for non-overlapping areas of the 2D grid. The approach only considers rules with numeric attributes in the antecedents.

9 Future Work and Conclusions

Association rule engines are often rightly accused of overloading the user with very large sets of rules. This applies to any software package, commercial or non-commercial, that we know.

In this chapter we describe a web based environment that allows the user to browse the rule space, which is organized by generality, by viewing one relevant set of rules at a time. A set of simple operators allows the user to move from one set of rules to another. Each set of rules is presented in a page and can be graphically summarized. In the following we describe the main advantages, limitations and future work of the proposed approach.

The main advantages are:

- PEAR enables selection and browsing across the set of derived AR.
- It enables plotting numeric properties of each subset of rules found.
- Browsing is done by a set of well-defined operators with a clear and intuitive semantics.
- Selection of Association Rules by an user is an implicit form of conveying domain knowledge, that can be later used, for example, in selecting rules for a classifier made out of a subset of rules.
- PEAR adheres to the PMML format standard, which allows the import of rule sets produced by many association rule engines.

The main limitations are:

- Visualization techniques are always difficult to evaluate. This one is no exception.
- The current implementation requires, on the server-side, the use of an operating system from one specific vendor.

Future work:

- Develop metrics to measure the gains of this approach.
- Develop a version of the environment that runs under a linux based web server (A prototype of this new version, which can be easily extended with new graphics and operators, is now available).
- Extend and formalize the notion of global metrics for sets of rules.
- Employ the studied global metrics to improve the initial set of rules (index page).
- Expand the set of available visualization techniques.

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast Discovery of Association Rules. Advances in Knowledge Discovery and Data Mining, 307–328 (1996)
2. Azevedo, P.J.: CAREN – A Java based Apriori Implementation for Classification Purposes, Technical Report, Departamento de Informática, Universidade do Minho, http://www.di.uminho.pt/~pja/class/caren.html
3. Berry, M.J.A., Linoff, G.S.: Data Mining Techniques: For Marketing, Sales, and Customer Support. John Wiley & Sons, Chichester (1997)
4. Brijs, T., Swinnen, G., Vanhoof, K., Wets, G.: Building an Association Rules Framework to Improve Product Assortment Decisions. Data Min. Knowl. Discov. 8(1), 7–23 (2004)
5. Brin, S., Motwani, R., Ullman, J.D., Tsur, S.: Dynamic itemset counting and implication rules for market basket data. SIGMOD Record (ACM Special Interest Group on Management of Data) 26(2), 255 (1997), http://citeseer.nj.nec.com/brin97dynamic.html
6. Chen, M.-C., Lin, C.-P.: A data mining approach to product assortment and shelf space allocation. In: Expert Systems with Applications. Elsevier, Amsterdam (2006)
7. Chang, H.-J., Hung, L.-P., Ho, C.-L.: An anticipation model of potential customers' purchasing behavior based on clustering analysis and association rules analysis. In: Expert Systems with Applications. Elsevier, Amsterdam (2007)
8. Clementine Software, SPSS, http://www.spss.com
9. Data Mining Group (PMML development), http://www.dmg.org/
10. Demiriz, A.: Enhancing Product Recommender Systems on Sparse Binary Data. Data Min. Knowl. Discov. 9(2), 147–170 (2004)
11. Jorge, A.: Hierarchical Clustering for thematic browsing and summarization of large sets of Association Rules. In: Proceedings of the Fourth SIAM International Conference on Data Mining, pp. 178–187. SIAM press, Philadelphia (2004)
12. Jorge, A., Poças, J., Azevedo, P.: Post-processing operators for browsing large sets of association rules. In: Lange, S., Satoh, K., Smith, C.H. (eds.) DS 2002. LNCS, vol. 2534, pp. 414–421. Springer, Heidelberg (2002)
13. Lent, B., Swami, A., Widom, J.: Clustering Association Rules. In: Gray, A., Larson, P. (eds.) Proc. of the Thirteenth International Conference on Data Engineering, ICDE 1997, IEEE Computer Society, Birmingham (1997)
14. Kandogan, E.: Visualizing Multi-dimensional Clusters, Trends and Outliers using Star Coordinates. In: Proceedings of KDD 2001, ACM Press, New York (2001)

15. Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., Verkamo, A.: Finding interesting rules from large sets of discovered association rules. In: Nabil, R., et al. (eds.) Proceedings of 3rd International Conference on Information and Knowledge Management, pp. 401–407. ACM Press (1994)
16. Li, W., Han, J., Pei, J.: CMAR: Accurate and Efficient Classification Based on Multiple-Class-Association Rules. In: IEEE International Conference on Data Mining (2001), http://citeseer.nj.nec.com/li01cmar.html
17. Ma, Y., Liu, B., Wong, K.: Web for Data Mininig: Organizing and Inter-preting the Discovered Rules Using the Web, School. SIGKDD Explorations, ACM SIGKDD 2(1) (July 2000)
18. Microsoft Web Site (JScript and JavaScript) and ASP, http://support.microsoft.com, http://support.microsoft.com
19. Poças, J.: Um ambiente de pós-processamento para regras de associação. MSc. Thesis in Portuguese, Mestrado em Análise de Dados e Sistemas de Apoio à Decisão (2003)
20. Savasere, A., Omiecinski, E., Navathe, S.: An efficient algorithm for mining association rules in large databases. In: Proc. of 21st Intl. Conf. on Very Large Databases (VLDB) (1995)
21. Silberschatz, A., Tuzhilin, A.: On subjective measures of interestingness in knowledge discovery. In: Proceedings of the First International Conference on Knowledge Discovery and Data Mining, pp. 275–281 (1995), http://citeseer.nj.nec.com/silberschatz95subjective.html
22. Tan, P.-N., Kumar, V.: Interestingness measures for association patterns: a perspective. In: Proceedings of the Workshop on Post-processing in Machine Learning and Data Mining, associated to KDD 2000 (2000)
23. Toivonen, H.: Sampling large databases for association rules. In: Proc. of 22nd Intl. Conf. on Very Large Databases (VLDB) (1996), http://citeseer.nj.nec.com/toivonen96sampling.html
24. W3C DOM Level 1 specification, http://www.w3.org/DOM/
25. W3C, Scalable Vector Graphics (SVG) 1.0 Specification, W3C Recommendation (September 2001), http://www.w3.org/TR/SVG/
26. Wettshereck, D.: A KDDSE-independent PMML Visualizer. In: Bohanec, M., Mladenic, D., Lavrac, N. (eds.) Proc. of IDDM 2002, workshop on Integration aspects of Decision Support and Data Mining. associated to the conferences ECML/PKDD (2002)

Visual Exploration of Frequent Itemsets and Association Rules

Li Yang

Department of Computer Science, Western Michigan University
li.yang@wmich.edu

Abstract. Frequent itemsets and association rules are defined on the powerset of a set of items and reflect the many-to-many relationships among the items. They bring technical challenges to information visualization which in general lacks effective visual technique to describe many-to-many relationships. This paper describes an approach for visualizing frequent itemsets and association rules by a novel use of parallel coordinates. An association rule is visualized by connecting its items, one on each parallel coordinate, with polynomial curves. In the presence of item taxonomy, an item taxonomy tree is displayed as coordinate and can be expanded or shrunk by user interaction. This interaction introduces a border in the generalized itemset lattice, which separates displayable itemsets from non-displayable ones. Only those frequent itemsets on the border are displayed. This approach can be generalized to the visualization of general monotone Boolean functions on lattice structure. Its usefulness is demonstrated through examples.

1 Introduction

Association rule mining[1,2] has been extensively studied in data mining. Traditional applications include cross-marketing, attachment mailing, catalog design, store layout, and customer segmentation. Finding frequent itemsets in association rule mining has also been found as a generic approach to solve problems in many other areas such as data classification and summarization[3,4,5].

Association rule mining often produces too many rules for humans to read over. The answer to this problem is to select the so-called most "interesting" rules[6,7]. Because interestingness is a subjective measure, finding the most interesting rules is inherently human being's work. It is thus expected that information visualization may play an important role in managing and exploring association rules and in identifying the most interesting ones.

When applying visualization techniques to frequent itemsets and association rules, however, people realize immediately that they are difficult to visualize. The difficulty comes from the following features of frequent itemsets and association rules: First, they are defined on the power set of a set of items that reflect the many-to-many relationships among the items. Although information visualization has been studied for many years, there seems no effective visual metaphor that is applicable to many many-to-one, never to say many-to-many,

S.J. Simoff et al. (Eds.): Visual Data Mining, LNCS 4404, pp. 60–75, 2008.

relationships. Second, frequent itemsets and association rules have their inherent closure properties. For example, any subset of a frequent itemset *abc* (*abc* is the abbreviation we use to represent the set $\{a, b, c\}$) is also a frequent itemset. A visualization technique should have room to reflect these closure properties. Third, the challenge here is to visualize many such itemsets or association rules.

This paper describes an approach of visually exploring frequent itemsets and association rules by using parallel coordinates. We have found that a novel use of parallel coordinates is generically suitable to visualize frequent itemsets whose subsets are frequent, and to visualize association rules where the results of dropping items from the RHS (right-hand side) of a rule or moving items from the RHS to the LHS (left-hand side) of the rule are also valid rules. A frequent itemset or an association rule is visualized by connecting items, one on each parallel coordinate. In the situation of visualizing generalized association rules where items are organized into item taxonomy, each coordinate can further be used to display an expandable taxonomy tree where leaf nodes are items and non-leaf nodes are item categories. This approach is capable of visualizing a large number of frequent itemsets and association rules by interactively selecting those ones whose items are interesting to the user.

In principle, all subsets (the power set) of a set of items form a lattice structure using subset inclusion as the partial order. Frequent itemsets define a monotone Boolean function on the lattice structure. The problem of visualizing such a function is nontrivial because of its monotonicity on this unique domain. Frequent itemsets are downward closed in the generalized itemset lattice which considers both subset and ancestor relationships as partial orders. Association rules produced from a frequent itemset are upward closed according to their LHSs in the sublattice formed by the frequent itemset. These closure properties will be investigated in this paper together with algorithms for generating and pruning association rules from the discovered frequent itemsets.

Our work on visual exploration of frequent itemsets and association rules was published in [8]. Since then, we have realized that the approach has a broader applicability and can be used to visually explore general monotone Boolean functions defined on a lattice structure. This paper gives a summary of our work on visual exploration of frequent itemsets and association rules. It also discusses the ideas of visual exploration of iceberg data cubes as another example of monotone Boolean functions. Iceberg data cube is an important operation in data warehousing.

2 Basic Concepts

We use market basket analysis as an example application to introduce basic concepts in association rule mining. Let $I = \{i_1, i_2, \ldots, i_k\}$ be a set of k elements, called items. A set of items $A \subseteq I$ is called an *itemset*. We use lower case letters to denote items and upper case letters to denote itemsets. Let $T_1, T_2, \ldots, T_n$ be a list of n subsets of I. We call each $T_i \subseteq I$ a transaction. We say that T_i supports an itemset A if $A \subseteq T_i$. Given a list of transactions, the *support* $P(A)$ of an

itemset A is the percentage of transactions that support A. A *frequent itemset* is an itemset whose support is above or equal to a user-specified minimum support value. An *association rule* is an expression $A \rightarrow B$, where A and B are itemsets and $A \cap B = \phi$. $P(A \cup B)$ is called *support* of the rule. $P(A \cup B)/P(A)$ is called *confidence* of the rule. The intuitive meaning of such a rule is that transactions that contain items in A tend to contain items in B. The problem of association rule mining is to find all association rules whose supports are above or equal to a user-specified minimum support and whose confidences are above or equal to a user-specified minimum confidence. An example rule is: *80% of the customers who buy diapers and baby powder also buy baby oil and baby food. The rule is supported by 12% of all transactions.*

An *item group* is a transitive closure of items in a set of frequent itemsets. Item groups are distinct and do not share items.

Generalized association rules come from the introduction of item taxonomy. An *item taxonomy* IT is a directed tree whose leaf nodes are items and whose non-leaf nodes are item categories. We call an item category $\acute{a}$ a *parent* of a (and a a *child* of $\acute{a}$) if there is an edge from $\acute{a}$ to a in IT. We call an item category $\hat{a}$ an *ancestor* of a (and a a *descendant* of $\hat{a}$) if there is a path from $\hat{a}$ to a in IT. An *ancestor itemset* of an itemset is obtained by replacing one or more items in the itemset with their ancestors. Formally, we call $\widehat{A}$ an *ancestor itemset* of A (and A a *descendant itemset* of $\widehat{A}$) if (1) $\widehat{A} \neq A$; (2) For all $a \in A$, either $a \in \widehat{A}$ or there exists an ancestor $\hat{a}$ of a such that $\hat{a} \in \widehat{A}$ (i.e. $\widehat{A}$ covers all items in A); (3) For all $a \in \widehat{A}$, there exists $b \in A$ so that $a = b$ or a is an ancestor of b (i.e. $\widehat{A}$ contains no extra item). A transaction T supports an item or item category a if $a \in T$ or a is an ancestor of some item in T. A transaction T supports an itemset A if T supports every element of A. The definitions of frequent itemsets and association rules need no change except for the following addition: for an association rule, no item in one side of the rule should be an ancestor of any item in the other side of the rule. Item taxonomy introduces another dimension of closure property. For example, an ancestor itemset of a frequent itemset is also frequent. If $A \rightarrow B$ is an association rule, then $A \rightarrow \widehat{B}$ is also an association rule. However, $A \rightarrow B$ does not imply that $\widehat{A} \rightarrow B$ is an association rule. Mining generalized association rules with item taxonomy was proposed in [9].

3 Itemset Lattice and Closure Properties

Without considering item taxonomy, the power set $\mathcal{P}(I)$ of a set I of items forms a lattice structure $< \mathcal{P}(I), \subseteq >$ where subset inclusion $\subseteq$ specifies the partial order. Figure 1 shows a Hasse diagram of the lattice on a set $I = \{a, b, c, d\}$ of items. Frequent itemsets are downward closed according to the subset relationship: if an itemset is frequent, so is every subset of it; if an itemset is infrequent, so is every superset of it. Therefore there exists a border on the itemset lattice, which separates frequent itemsets from infrequent ones. The existence of such a border is guaranteed by the downward closure property. It is independent of any particular database or minimum support value. The Apriori principle [1,2]

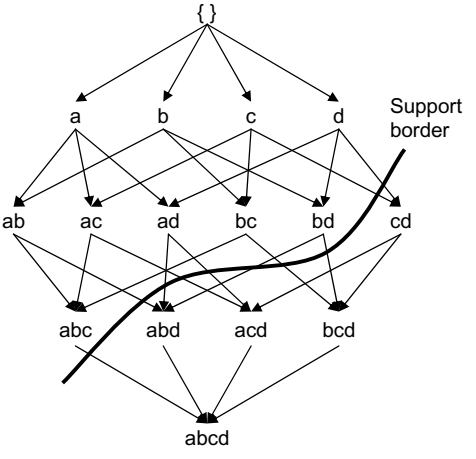


Fig. 1. The lattice for $I = \{a, b, c, d\}$

of finding frequent itemsets employs this property to efficiently prune the search space in generating frequent itemsets.

The above closure property can be extended to generalized frequent itemsets when item taxonomy is considered. Let I be a set of items and IT an item taxonomy on I. Define the *generalized power set* $\mathcal{GP}(I, IT)$ as $\mathcal{GP}(I, IT) = P(I) \cup \{\text{all ancestor itemsets of } A | \forall A \in \mathcal{P}(I)\}$, that is, $\mathcal{GP}(I, IT)$ contains all itemsets and their ancestor itemsets. Define a partial order $\preceq$ as: (1) $A \preceq B$ if $A \subseteq B$; (2) $\widehat{A} \preceq A$. Then $< \mathcal{GP}(I, IT), \preceq >$ is a lattice. It is easy to verify that $P(A) \geq P(B)$ if $A \preceq B$. Therefore there is a border in $< \mathcal{GP}(I, IT), \preceq >$ that separates frequent itemsets from infrequent ones. For example, assume the items $I = \{a, b, c, d\}$ are organized into an item taxonomy tree IT shown in Figure 2. Figure 3 shows a Hasse diagram of the lattice $< \mathcal{GP}(I, IT), \preceq >$ which is an extension of the itemset lattice in Figure 1. In Figure 3, we use straight lines to denote subset relationships and arcs to denote ancestor relationships.

For association rules, the confidence measure does not have such a closure property in the itemset lattice. If we consider the sub-lattice formed by a frequent itemset, however, association rules generated from the frequent itemset have a closure property: taking the lattice structure in Figure 1 for example, if $a \rightarrow bc$ is a valid association rule, then $ab \rightarrow c$ and $ac \rightarrow b$ are also association rules that can pass the same minimum support and the same minimum confidence. In fact, let A be a frequent itemset and $B \subseteq A$, then $B \rightarrow (A - B)$ is an association rule if the support $P(B)$ does not exceed $P(A)/minconf$ where *minconf* is the user-defined minimum confidence. Furthermore, $C \rightarrow (A - C)$ is also an association rule if C satisfies $B \subseteq C \subseteq A$. This means that the association rules generated from a frequent itemset are upward closed according to their LHSs in the sub-lattice formed by the frequent itemset. This fact suggests that we can find the border of LHSs for a given frequent itemset by using a reverse search algorithm

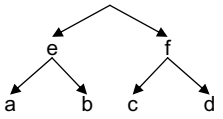


Fig. 2. A simple item taxonomy tree

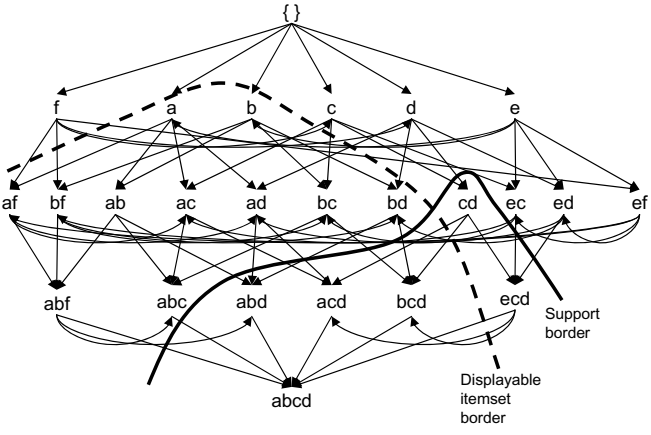


Fig. 3. The lattice $< \mathcal{GP}(I, IT), \preceq >$

on the sub-lattice. The algorithm is illustrated in Algorithm 1. We denote the border LHSs of association rules generated from a frequent itemset A as

Association rule generation is then straightforward: For each frequent itemset A and each $B \in \mathit{BorderLHSs}(A, c)$, we generate the rule $B \rightarrow (A - B)$. However, the rules generated from A may be redundant if A is a subset or an ancestor itemset of another frequent itemset. For example, if $a \rightarrow bc$ is a rule, then $a \rightarrow b$, $a \rightarrow c$, $a \rightarrow \hat{b}c$, and $a \rightarrow b\hat{c}$ are all valid association rules. In fact, let A be a subset or an ancestor itemset of C, i.e. $A \preceq C$, and B lies in both $\mathit{BorderLHSs}(A, \mathit{minconf})$ and $\mathit{BorderLHSs}(C, \mathit{minconf})$. Then the rule $B \rightarrow (A - B)$ is redundant with respect to $B \rightarrow (C - B)$. Therefore, non-redundant association rules can be generated by pruning the border LHSs of an itemset so that it does not share any itemsets with the border LHSs of its supersets or descendent itemsets. The corresponding algorithm is shown as Algorithm 2, where only association rules generated in Line 7 are visualized using visualization techniques discussed in the following.

4 Parallel Coordinates

Parallel coordinates [10,11,12,13,14,15] were originally used to visualize relational records all with equal number of attributes. However, it can also be used to visualize data with variable lengths such as frequent itemsets and association rules. Basic elements of itemsets or association rules are sets of items, which can

Algorithm 1. BorderLHSs(A)

```
1: Initialize: FIFO ← {A}; BorderLHSs(A) ← ∅;
2: while FIFO ≠ ∅ do
3:    dequeue B from the head of FIFO;
4:    onBorder ← TRUE;
5:    for all (|B| − 1)-subset C of B do
6:       if P(C) ≤ P(A)/minconf then
7:          onBorder ← FALSE;
8:          If C is not in FIFO, enqueue C to the end of FIFO;
9:       end if
10:   end for
11:   If onBorder = TRUE, add B to BorderLHSs(A);
12: end while
```

Algorithm 2. GenerateRule(X)

```
1: for all A ∈ X do
2:    LHSs(A) ← BorderLHSs(A);
3:    for all C ∈ X such that C is a (|A| + 1)-superset or a child itemset of A do
4:       LHSs(A) ← LHSs(A) − BorderLHSs(C);
5:    end for
6:    for all B ∈ LHSs(A) do
7:       generate and visualize the rule B → (A − B)
8:    end for
9: end for
```

be handled by listing all items along a vertical coordinate. The resulting coordinate is then repeated evenly in the horizontal direction until there are enough coordinates to host the longest frequent itemset or the longest association rule. An itemset or an association rule can be visualized as a polygonal line connecting all items in the itemset or the rule. Parameters such as support factor and confidence can be mapped to graphics features such as line-width and color.

For example, Figure 4(a) illustrates three frequent itemsets *adbe*, *cdb* and fg as polygonal lines. One important feature of this display is that it provides a way to visualize only the frequent itemsets at the border in the itemset lattice. A visualized frequent itemset implies that any subset is also frequent. Figure 4(b) illustrates an association rule $ab \rightarrow cd$ as one polygonal line for its LHS, followed by an arrow connecting another polygonal line for its RHS. This visualization handles nicely the upward closure property of association rules: subsets of the RHS are absorbed and are not displayed. For example, $ab \rightarrow cd$ implies that $abc \rightarrow d$, $abd \rightarrow c$, $ab \rightarrow c$, and $ab \rightarrow d$ are valid association rules. The implied association rules are not displayed.

If two or more itemsets or rules have parts in common, for example, *adbe* and *cdb* in Figure 4(a), we can use polynomial curves instead of polygonal lines to distinguish them. Cubic Bezier curves offer C^1 continuity at the joint points when connected with each other. A cubic Bezier curve needs four control points, $\mathbf{p} = [p_0, p_1, p_2, p_3]^T$, among which p_0, p_3 are two end points and p_1, p_2 specify

Fig. 4. Visualizing frequent itemsets(a) and an association rule(b)

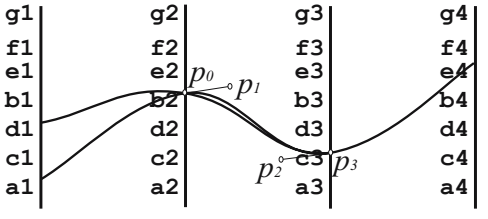


Fig. 5. Visualizing association rules using Bezier curves

derivatives at the end points, i.e. $\mathbf{p}'(0) = 3(p_1 - p_0)$ and $\mathbf{p}'(1) = 3(p_3 - p_2)$. The curve itself is a blending of Bernstein polynomials, $\mathbf{p}(u) = b(u)^T\mathbf{p}$ for $0 \leq u \leq 1$ where $b(u) = [(1-u)^3, 3u(1-u)^2, 3u^2(1-u), u^3]^T$.

An example rule, $ab \rightarrow ce$, is visualized in Figure 5 by using three Bezier curves. We use a_i to denote the position of the item a in the i-th coordinate. The figure shows four control points p_0, p_1, p_2, p_3 of the b_2c_3 segment. p_0 is chosen as the position of b_2. p_3 is chosen as the position of c_3. p_1 is chosen so that $p_1 - p_0$ is in parallel to $c_3 - a_1$. p_2 is chosen so that $p_3 - p_2$ is in parallel to $e_4 - b_2$. In this way, we keep C^1 continuity of the sequence of Bezier curves at all coordinates.

Another association rule illustrated in Figure 5 is $db \rightarrow ce$, which has three items shared with the rule $ab \rightarrow ce$. The b_2c_3 segments of the two association rules are visualized as two different Bezier curves to keep C^1 continuity at the position of the first shared item b. However, the second shared segments c_3e_4 in the two rules coincide completely with each other. We think this is acceptable to distinguish two association rules with parts in common while keeping the display reasonably clean.

We organize items along a coordinate in the following way: First, items are arranged by item groups so that the items belonging to the same group are displayed together. Since item groups do not share items, Bezier curves visualizing itemsets in different item groups will never intersect with each other. In this way, the curves are organized into "horizontal bands" according to item groups. Second, items within the same group are arranged upward in descending order according to their supports. In the same way, items in an itemset are also arranged in descending order according to their supports. In this way, we make

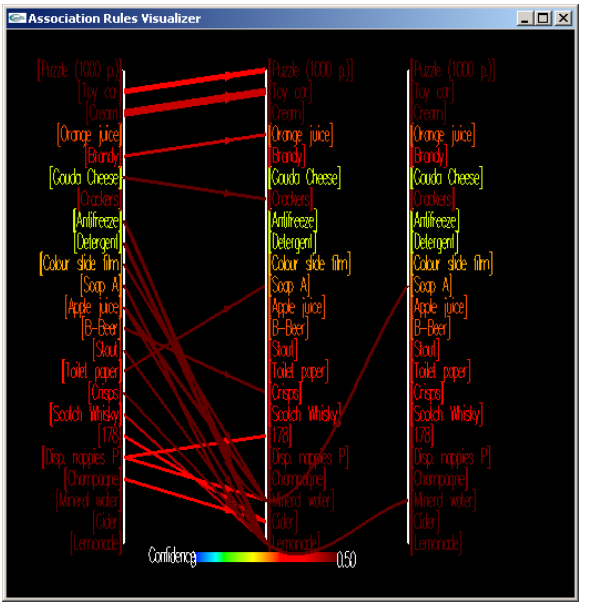


Fig. 6. Visualizing association rules discovered from a transaction dataset

sure that the slopes of a polygonal line are always positive. This helps to reduce the chance of tangled intersection with other polygonal lines.

We tested this approach by using a supermarket transaction dataset in IBM DB2 Intelligent Miner. We used Intelligent Miner to generate frequent itemsets and developed our own program to discover many-to-many association rules using the algorithms presented in Section 3. Figure 6 visualizes the set of discovered association rules when the minimum support is set to 2.8% and the minimum confidence is set to 30%. Support of a rule is represented by line width. Confidence is represented by color. Items are organized into groups. Association rules are aligned according to where the RHSs separate from LHSs. In this example, the leftmost coordinate represents the LHSs of the rules while the right two coordinates represent the RHSs of the rules. Item names are annotated on the left side of the LHS coordinates and on the right side of the RHS coordinates. The visualization is interactive such that each displayed rule is selectable by mouse clicking. The selected rules and all of its implied rules will then be displayed.

5 Dealing with Item Taxonomy

The above approach of visualizing frequent itemsets and association rules requires that all items be arranged along a coordinate. This could be a problem when there are many items in the database. In real world applications, fortunately, items are often organized into item taxonomy. This gives us a way of

visualizing itemsets and rules by replacing each parallel coordinate with a display of the item taxonomy tree.

Each node of the item taxonomy tree has an `expand` flag which can be turn on or off interactively by the user. Its child nodes will be displayed only if the flag is on and the node itself is displayed. An itemset is called *displayable* if all items in the itemset have the `expand` flags set in all of their ancestors, that is, all items in the itemset must be shown in the visualization of the taxonomy tree. This displayable property is downward closed in the generalized itemset lattice $< \mathcal{GP}, \preceq >$. Similar as frequent itemsets, if an itemset A is displayable, any itemset B such that $B \preceq A$ is also displayable. Therefore we have now two borders in the generalized itemset lattice $< \mathcal{GP}, \preceq >$: one border separates frequent itemsets from infrequent ones; the other border separates displayable itemsets from indisplayable ones. For example, let us assume that only items c, d, e, f are displayable in the item taxonomy in Figure 2, this specifies a border of displayable itemsets in the generalized lattice, which has been shown in Figure 3.

As we expand or shrink the displayed item taxonomy tree, the border of displayable itemsets will change. This gives us a way to selectively visualize the frequent itemsets whose items are among what we are interested in. A visualization system can be designed such that only the non-redundant displayable frequent itemsets are displayed. In the generalized itemset lattice $< \mathcal{GP}, \preceq >$, these non-redundant displayable frequent itemsets must reside on the border which separates displayable frequent itemsets from the rest itemsets. This border marks the intersection of all frequent itemsets and all displayable itemsets. Taking the two borders in Figure 3 as example, ec and ed are two displayable frequent itemsets that are visualized. The other displayable frequent itemsets are implied by these two itemsets. By interactively expanding or shrinking the displayed taxonomy tree, therefore, we can control the set of frequent itemsets to display. Specifically, a frequent itemset is displayed if and only if:

1. the frequent itemset is displayable; and
2. the frequent itemset is not the subset of any displayable frequent itemsets; and
3. the frequent itemset is not the ancestor itemset of any displayable frequent itemsets.

Similarly, an association rule is called *displayable* if all items in the rule have the `expand` flag set in all of their ancestors, that is, the frequent itemset from which the association rule is generated must be displayable. We prune association rules generated from displayable frequent itemsets by following the procedure described in Section 3. Association rules generated by Algorithm 2 from displayable frequent itemsets are then visualized. In this way, we keep our visualization readable by reducing the number of rules to be displayed to only those ones that are representative of the other rules.

In summary, we have introduced two kinds of interactions in our visualization. First, the displayed item taxonomy tree can be expanded or shrunk by clicking a node in the tree. This interaction changes the displayable border in the generalized itemset lattice and, consequently, selects another set of frequent

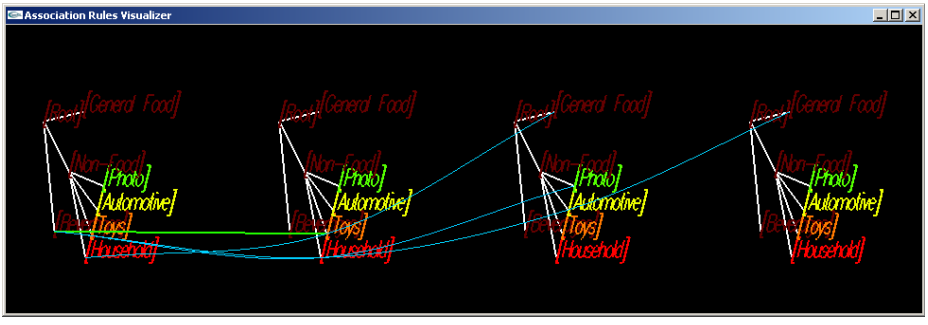


Fig. 7. Visualizing frequent itemsets on the displayed item nodes of the taxonomy tree

itemsets or association rules to visualize. By interactively selecting items to display, we visualize only a minimal set of frequent itemsets or association rules whose items are among those that we are interested in. Second, when we click a displayed association rule, the other rules implied and pruned by the rule will be printed together with their support and confidence values. This interaction gives us more information about the pruned association rules. Although the pruned rules are implied by the displayed rule, this does not mean that the pruned rules are not interesting. From the user's perspective, the pruned rules could be more interesting than the displayed ones because they have higher supports and/or confidences.

6 Experiments and Screen Snapshots

We illustrate our approach through examples using a supermarket transaction data set in IBM DB2 Intelligent Miner as test data. The data set contains 80 items which are leaf nodes of a 4-level taxonomy tree. 496 frequent itemsets are discovered when the minimum support is set to 5%. Initially, our visualization displays only the [Root] nodes. Figure 7 visualizes frequent itemsets on the expanded taxonomy tree. In the taxonomy, [Root] has three children: [Beverage], [Non-Food] and [General Food]. They are listed in the descending order of their support factors. The color of the name of each item or item category represents its support. Displayable frequent itemsets are visualized as Bezier curves. Line-width of a curve represents the support value of the corresponding itemset. Items in a frequent itemset are pre-arranged according to the order that they are arranged on the taxonomy tree.

Frequent itemsets are pruned according to the strategy presented in Section 5. For example, the longest frequent itemset shown in Figure 7 is {[Beverages] [Household] [Automotive] [General Food]}. It implies that all of its subsets are frequent. It also implies that its ancestor itemsets, for example, {[Beverages] [Non-Food] [General Food]}, are also frequent. Those implied itemsets are not visualized. The user can pick up an itemset by clicking on its curve segments.

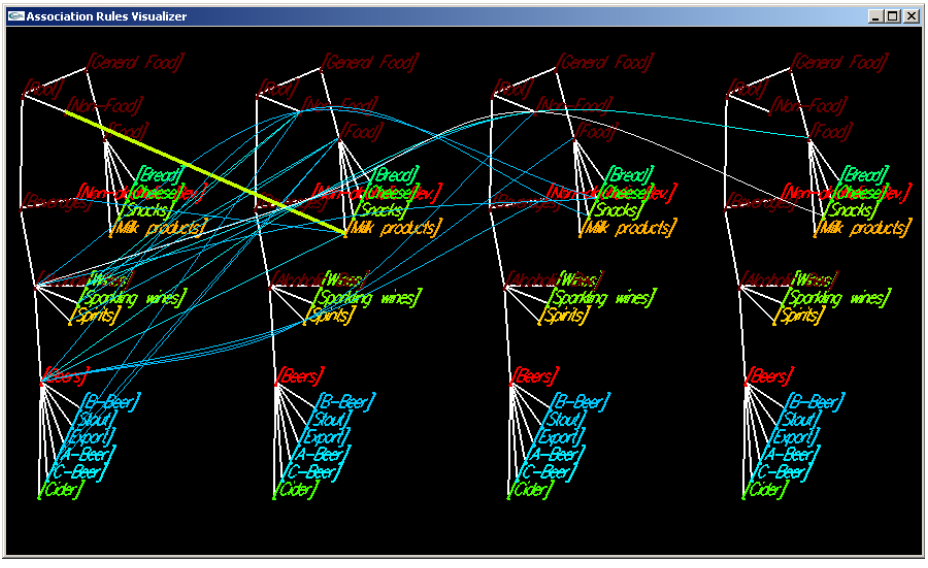


Fig. 8. Frequent itemsets drawn primarily on the selected item nodes: Beers and Foods

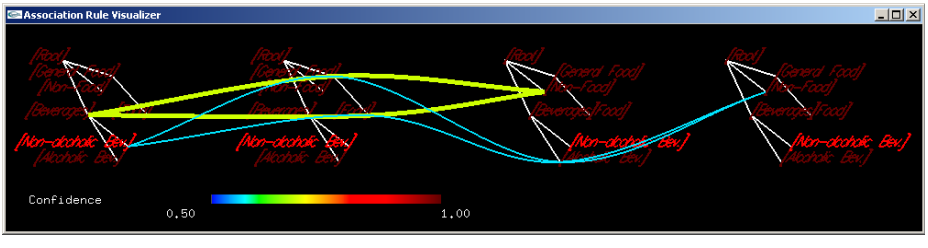


Fig. 9. Visualizing an association rule

The itemset and its implied itemsets will then be printed with their support values. We can drill down by interactively expanding the item taxonomy tree. Figure 8 visualizes frequent itemsets when we expand the taxonomy tree along [Beverages], [Alcoholic], [Beers] and along [General Food], [Food].

Figure 9 visualizes the discovered association rules with item taxonomy when the minimum support is set to 5% and the minimum confidence is set to 50%. Association rules are aligned according to where the RHSs separate from the LHSs. In this example, the left two coordinates represent the LHSs of the rules and the right two coordinates represent the RHSs. Support is represented by line width and confidence is represented by color. As we discussed, implied rules are not displayed. Figure 10 shows the result when we expand the taxonomy tree along [Beverages], [Alcoholic], [Beers] and along [General Food], [Food].

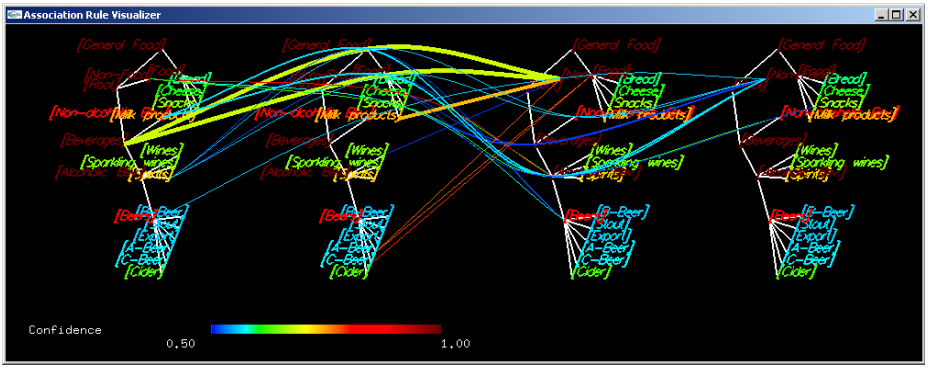


Fig. 10. Association rules drawn primarily on the selected item nodes: Beers and Foods

7 Visualization of Iceberg Data Cubes

This section discusses the idea of visually exploring iceberg data cubes as another example of monotone Boolean functions. Data cube [16] is an important operation in online analytical processing. It can be logically thought as the union of all group-by's of a relational table, where each group-by is obtained by grouping on a subset of aggregating attributes. For a relation with d aggregating dimensions $D_1, \ldots, D_d$, data aggregation on the d dimensions would have $\prod_{i=1}^{d} |D_i|$ cells, which constitute a base cuboid in the data cube on the d dimensions. Assume each dimension D_i has c_i levels of concept hierarchy, a full data cube would contain $\prod_{i=1}^{d} |c_i|$ cuboids. Clearly, data cube computation soon becomes prohibitive as the number of aggregating dimensions increases. To solve this problem, iceberg cube [17] was proposed to compute only dense cells, that is, cells that contain more data records than a user-specified threshold.

A class hierarchy C_i on dimension D_i is a lattice structure (in most cases it is simply a chain of layers with total order). A cube lattice on the dimensions $D_1, \ldots, D_d$ is then a product lattice $C_1 \times \cdots \times C_d$ where the partial order is defined as $c'_1 \times \cdots \times c'_d \preceq c''_1 \times \cdots \times c''_d$ if $c'_i \preceq c''_i$, where $c'_i, c''_i \in C_i$ for $1 \leq i \leq d$, respectively. Figure 11 gives example class hierarchies on three dimensions, Product, Location, Date of a year, and shows a Hasse diagram of their cube lattice which contains $3 \times 3 \times 4 = 36$ cuboids.

An iceberg data cube yet contains a large number of cube cells. We expect that visualization plays an important role in exploring these data cells. In the same way as frequent itemsets, iceberg data cells define a monotone Boolean function on the data cube lattice. If a data cell is dense, so does every ancestor data cell up the data cube lattice. Therefore, ideas similar to the ones we used in visualizing frequent itemsets can be used to visually explore iceberg data cells. Specifically, each dimension bears a lattice structure of class hierarchy where data values or ranges of data values can be organized into a taxonomy tree. Unlike the visualization of frequent itemsets where there is a single item taxonomy tree, however, the taxonomy tree of each dimension can be displayed on each

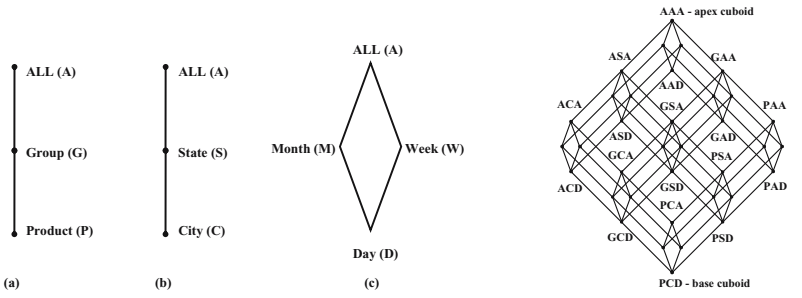


Fig. 11. Concept hierarchy lattices of dimensions: (a) Product; (b) Location; (c) Date; and a Hasse diagram of their product lattice. Cuboids containing Month (M) and Week (W) are not marked to help keep the diagram clean.

coordinate and a data cell is visualized as a sequence of Bezier curves. Similar pruning strategies can be applied to visualize only non-implied data cells. This technique of data cube visualization is currently under investigation.

8 Related Work

The problem of association rule pruning has been studied by many researchers. Klemettinen *et al.*[7] studied association rules at the border of frequent itemsets and used templates to specify what users want to see. The problem of hypergraph transversal was studied by Gunopulos *et al.*[18]. Similarly, the upper and lower support-confidence borders of association rules have been studied by Bayardo and Agrawal[19,6]. Global partial orders of sequential data have been discussed by Mannila and Meek[20]. Toivonen *et al.*[21] introduced rule cover for pruning association rules. Although visualization was not the subject of discussion of these papers, each of them suggested ways to visualize rules, either explicitly or implicitly. As mining the most interesting rules became a concern, interestingness measure has received much attention. [22] introduced a method of pruning association rules by using the maximum entropy principle. Chi-square(χ^2) test was introduced in [23] to measure the correlation of items and was used in [3] for pruning and summarizing large amounts of association rules that have a single Boolean attribute on the RHSs.

The rule pruning method discussed in this paper is influenced by [24,23]. Aggarwal and Yu [24] introduced technique to remove two types of redundant rules. The first type of redundancy is called simple redundancy. It says that $ab \rightarrow c$ should be pruned in the presence of $a \rightarrow bc$. The second type of redundancy is called strict redundancy. It says that $a \rightarrow c$ should be pruned in the presence of $a \rightarrow bc$. However, these redundancies did not consider item taxonomy. Silverstein *et al.*[23] introduced correlation rule mining by using chi-square test. This correlation measure is upward closed in the itemset lattice while the support is downward closed. An algorithm was designed to find a minimal set of correlated frequent itemsets.

There are many methods to visualize frequent itemsets and association rules of simplified forms, for example, by simplifying the problem to two or few dimensions. A straightforward yet useful way to visualize one-to-one association rules is a two-dimensional matrix with LHS items in one dimension and RHS items in the other dimension. Wong *et al.*[25] gave an alternative of arranging dimensions where all rules are placed in one dimension and all items are placed in the other dimension. A rule is visualized as bar charts against all items in the rule. This approach is good for visualizing large itemsets. But it is difficult to work once the number of itemsets becomes large. Fukuda *et al.*[26] and Yoda *et al.*[5] permit association rules to contain two numeric attributes as the LHS and a Boolean attribute as the RHS. This approach enables them to transform a set of tuples into a bitmap image. It provides a method for visualizing association rules with up to two items in the LHS.

A directed graph is another prevailing technique to depict item associations. The nodes of a directed graph represent the items, and the edges represent the associations. This technique works when only a few items and associations are involved. However, the graph may quickly turn into a tangled mess [7] with as few as a dozen rules.

9 Conclusion and Future Work

Frequent itemsets and association rules present technical challenges to information visualization because they are defined on the power set of a set of items and specify many-to-many relationships among the items. Our objective is to visualize a large number of such itemsets or rules. We believe that this step is a must to make a visualization approach practically useful in real world data mining applications. This paper has described an approach of using parallel coordinates as a simple and effective metaphor to visualize frequent itemsets and association rules. The closure properties among frequent itemsets and association rules are embedded in the visualization. Each coordinate can be represented as an expandable item taxonomy tree where items are organized into item taxonomy. With such a graphic encoding, finding interesting itemsets and rules would be enhanced with visual information that helps humans to interpret data mining results. A software system has been developed. It enables three kinds of interaction: (1) The displayed item taxonomy tree is clickable to expand or shrink. This enables us to visualize only those itemsets or rules on the selected items. (2) A displayed itemset or rule is clickable to list all itemsets or rules it implies. (3) Panning and zooming are supported to focus on a particular area in the visualization. All these together provide a feasible approach to visually explore large frequent itemsets and many-to-many association rules.

The fundamental problem in the visualization of frequent itemsets and association rules is that, depending on the support value, there is a long border of frequent itemsets in the itemset lattice and there is no visual technique directly applicable to displaying many many-to-many relationships. We overcame this problem by using an expandable item taxonomy tree to organize items.

Basically, this introduces a displayable itemset border in the generalized item lattice, which separates displayable itemsets from non-displayable ones. Only those frequent itemsets that are on this border are displayed. By changing this border through expanding or shrinking the displayed item taxonomy tree, we selectively visualize the frequent itemsets and association rules that we are interested in.

Visualizing many-to-many association rules is a challenging problem. Our approach of visualization brings issues for future research on how interaction and redundancy might be exploited toward visualizing frequent itemsets and association rules. One research topic is how to explore more effectively closure properties and borders to selectively visualize itemsets or rules. The changes of these borders should be associated with meaningful user interactions. Another topic is how to prune uninteresting itemsets and rules. Summarization techniques would be needed to better summarize and prune the discovered itemsets and rules. Finally, we have realized that our approach is applicable to visually explore general monotone Boolean functions defined on a lattice structure. It is part of our future work to discover its applications in many other areas.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proc. ACM SIGMOD Inter. Conf. on Management of Data (SIGMOD 1993), Washington, D.C, pp. 207–216 (1993)
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proc. 20th Inter. Conf. on Very Large Data Bases (VLDB 1994), Santiago, Chile, pp. 207–216 (1994)
3. Liu, B., Hsu, W., Ma, Y.: Pruning and summarizing the discovered associations. In: Proc. 5th ACM SIGKDD Inter. Conf. on Knowledge Discovery and Data Mining (KDD 1999), San Diego, CA, pp. 145–154 (1999)
4. Mannila, H., Toivonen, H.: Multiple uses of frequent sets and condensed representations. In: Proc. 2nd Inter. Conf. on Knowledge Discovery and Data Mining (KDD 1996), Portland, OR, pp. 189–194 (1996)
5. Yoda, K., Fukuda, T., Morimoto, Y., Morishita, T., Tokuyama, T.: Computing optimized rectilinear regions for association rules. In: Proc. 3rd Inter. Conf. on Knowledge Discovery and Data Mining (KDD 1997), Newport Beach, CA, pp. 96–103 (1997)
6. Bayardo, R.J., Agrawal, R.: Mining the most interesting rules. In: Proc. 5th ACM SIGKDD Inter. Conf. on Knowledge Discovery and Data Mining (KDD 1999), San Diego, CA, pp. 145–154 (1999)
7. Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., Verkamo, I.: Finding interesting rules from large sets of discovered association rules. In: Proc. 3rd ACM Inter. Conf. on Information and Knowledge Management (CIKM 1994), Gaithersburg, MD, pp. 401–407 (1994)
8. Yang, L.: Pruning and visualizing generalized association rules in parallel coordinates. IEEE Trans. Knowledge and Data Engineering 17, 60–70 (2005)
9. Srikant, R., Agrawal, R.: Mining generalized association rules. In: Proc. 21st Inter. Conf. on Very Large Data Bases (VLDB 1995), Zurich, Switzerland, pp. 407–419 (1995)

10. Inselberg, A.: The plane with parallel coordinates. The Visual Computer 1, 69–91 (1985)
11. Inselberg, A., Reif, M., Chomut, T.: Convexity algorithms in parallel coordinates. Journal of the ACM 34, 765–801 (1987)
12. Inselberg, A.: Parallel coordinates: A tool for visualizing multi-dimensional geometry. In: Proc. 1st IEEE Conf. on Visualization, San Francisco, CA, pp. 361–375 (1990)
13. Inselberg, A., Dimsdale, B.: Multi-dimensional lines. SIAM J. Applied Mathematics 54, 559–596 (1994)
14. Inselberg, A.: Visualizing high dimensional datasets and multivariate relations (tutorial). In: Proc. 6th ACM SIGKDD Inter. Conf. on Knowledge Discovery and Data Mining (KDD 2000), Boston, MA (2000)
15. Martin, A., Ward, M.O.: High dimensional brushing for interactive exploration of multivariate data. In: Proc. IEEE Conf. on Visualization, Atlanta, GA, pp. 271–278 (1995)
16. Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pellow, F., Pirahesh, H.: Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. Data Mining and Knowledge Discovery 1, 29–53 (1997)
17. Beyer, K., Ramakrishnan, R.: Bottom-up computation of sparse and iceberg cube. In: Proc. ACM SIGMOD Conf. Management of Data (SIGMOD), Philadelphia, PA, pp. 359–370 (1999)
18. Gunopulos, D., Khardon, R., Mannila, H., Toivonen, H.: Data mining, hypergraph transversals, and machine learning. In: Proc. 16th ACM Symp. on Principles of Database Systems (PODS 1997), Tucson, AZ, pp. 209–216 (1997)
19. Bayardo, R.J.: Efficiently mining long patterns from databases. In: Proc. ACM SIGMOD Inter. Conf. on Management of Data (SIGMOD 1998), Seattle, WA, pp. 85–93 (1998)
20. Mannila, H., Meek, C.: Global partial orders from sequential data. In: Proc. 6th ACM SIGKDD Inter. Conf. on Knowledge Discovery and Data Mining (KDD 2000), Boston, MA, pp. 161–168 (2000)
21. Toivonen, H., Klemettinen, M., Ronkainen, P., Hatonen, K., Mannila, H.: Pruning and grouping of discovered association rules. In: ECML 1995 Workshop on Statistics, Machine Learning, and Knowledge Discovery in Databases, Heraklion, Crete, Greece, pp. 47–52 (1995)
22. Jaroszewicz, S., Simovici, D.A.: Pruning Redundant Association Rules Using Maximum Entropy Principle. In: Chen, M.-S., Yu, P.S., Liu, B. (eds.) PAKDD 2002. LNCS (LNAI), vol. 2336, pp. 135–147. Springer, Heidelberg (2002)
23. Silverstein, C., Brin, S., Motwani, R.: Beyond market baskets: Generalizing association rules to dependence rules. Data Mining and Knowledge Discovery 2, 39–68 (1998)
24. Aggarwal, C.C., Yu, P.S.: Online generation of association rules. In: Proc. 14th Inter. Conf. on Data Engineering (ICDE 1998), Orlando, FL, pp. 402–411 (1998)
25. Wong, P.C., Whitney, P., Thomas, J.: Visualizing association rules for text mining. In: Proc. IEEE Symp. on Information Visualization (InfoVis 1999), San Franscisco, CA, pp. 120–123 (1999)
26. Fukuda, T., Morimoto, Y., Morishita, S., Tokuyama, T.: Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization. In: Proc. ACM SIGMOD Inter. Conf. on Management of Data (SIGMOD 1996), Montreal, Canada, pp. 13–23 (1996)

Visual Analytics: Scope and Challenges

Daniel A. Keim, Florian Mansmann, Jörn Schneidewind,
Jim Thomas, and Hartmut Ziegler

University of Konstanz
{keim, mansmann, schneide, ziegler}@informatik.uni-konstanz.de
http://infovis.uni-konstanz.de
Pacific Northwest National Laboratory,
National Visualization and Analytics Center (NVAC)
nvac@pnl.gov
http://nvac.pnl.gov

Abstract. In today's applications data is produced at unprecedented rates. While the capacity to collect and store new data rapidly grows, the ability to analyze these data volumes increases at much lower rates. This gap leads to new challenges in the analysis process, since analysts, decision makers, engineers, or emergency response teams depend on information hidden in the data. The emerging field of visual analytics focuses on handling these massive, heterogenous, and dynamic volumes of information by integrating human judgement by means of visual representations and interaction techniques in the analysis process. Furthermore, it is the combination of related research areas including visualization, data mining, and statistics that turns visual analytics into a promising field of research. This paper aims at providing an overview of visual analytics, its scope and concepts, addresses the most important research challenges and presents use cases from a wide variety of application scenarios.

1 Introduction

The information overload is a well-known phenomenon of the information age, since due to the progress in computer power and storage capacity over the last decades, data is produced at an incredible rate, and our ability to collect and store these data is increasing at a faster rate than our ability to analyze it. But, the analysis of these massive, typically messy and inconsistent, volumes of data is crucial in many application domains. For decision makers, analysts or emergency response teams it is an essential task to rapidly extract relevant information from the flood of data. Today, a selected number of software tools is employed to help analysts to organize their information, generate overviews and explore the information space in order to extract potentially useful information. Most of these data analysis systems still rely on interaction metaphors developed more than a decade ago and it is questionable whether they are able to meet the demands of the ever-increasing mass of information. In fact, huge investments in time and money are often lost, because we still lack the possibilities to properly interact with the databases. Visual analytics aims at bridging this gap

S.J. Simoff et al. (Eds.): Visual Data Mining, LNCS 4404, pp. 76–90, 2008.

by employing more intelligent means in the analysis process. The basic idea of visual analytics is to visually represent the information, allowing the human to directly interact with the information, to gain insight, to draw conclusions, and to ultimately make better decisions. The visual representation of the information reduces complex cognitive work needed to perform certain tasks. People may use visual analytics tools and techniques to synthesize information and derive insight from massive, dynamic, and often conflicting data by providing timely, defensible, and understandable assessments.

The goal of visual analytics research is to turn the information overload into an opportunity. Decision-makers should be enabled to examine this massive, multi-dimensional, multi-source, time-varying information stream to make effective decisions in time-critical situations. For informed decisions, it is indispensable to include humans in the data analysis process to combine flexibility, creativity, and background knowledge with the enormous storage capacity and the computational power of today's computers. The specific advantage of visual analytics is that decision makers may focus their full cognitive and perceptual capabilities on the analytical process, while allowing them to apply advanced computational capabilities to augment the discovery process. This paper gives an overview on visual analytics, and discusses the most important research challenges in this field. Real world application examples are presented that show how visual analytics can help to turn information overload as generated by today's applications into useful information.

The rest of the paper is organized as follows: section 2 defines visual analytics and discusses its scope. The visual analytics process is formalized in section 3. Section 4 covers the 10 most important application challenges in the field and presents some approaches addressing these problems. It is followed by the 10 most important technical challenges in section 5. Finally, section 6 concludes our work and gives a short outlook of the future of visual analytics.

2 Scope of Visual Analytics

In general, *visual analytics* can be described as "the science of analytical reasoning facilitated by interactive visual interfaces" [1]. To be more precise, visual analytics is an iterative process that involves information gathering, data preprocessing, knowledge representation, interaction and decision making. The ultimate goal is to gain insight in the problem at hand which is described by vast amounts of scientific, forensic or business data from heterogeneous sources. To reach this goal, visual analytics combines the strengths of machines with those of humans. On the one hand, methods from knowledge discovery in databases (KDD), statistics and mathematics are the driving force on the automatic analysis side, while on the other hand human capabilities to perceive, relate and conclude turn visual analytics into a very promising field of research.

Historically, visual analytics has evolved out of the fields of information and scientific visualization. According to Colin Ware, the term visualization is meanwhile understood as "a graphical representation of data or concepts" [2], while

the term was formerly applied to form a mental image. Nowadays fast computers and sophisticated output devices create meaningful visualizations and allow us not only to mentally visualize data and concepts, but also to see and explore an exact representation of the data under consideration on a computer screen. However, the transformation of data into meaningful visualizations is not a trivial task that will automatically improve through steadily growing computational resources. Very often, there are many different ways to represent the data under consideration and it is unclear which representation is the best one. State-of-the-art concepts of representation, perception, interaction and decision-making need to be applied and extended to be suitable for visual data analysis.

The fields of information and scientific visualization deal with visual representations of data. The main difference among the two is that scientific visualization examines potentially huge amounts of scientific data obtained from sensors, simulations or laboratory tests. Typical scientific visualization applications are flow visualization, volume rendering, and slicing techniques for medical illustrations. In most cases, some aspects of the data can be directly mapped onto geographic coordinates or into virtual 3D environments. We define Information visualization more generally as the communication of abstract data relevant in terms of action through the use of interactive interfaces. There are three major goals of visualization, namely a) presentation, b) confirmatory analysis, and c) exploratory analysis. For presentation purposes, the facts to be presented are fixed a priori, and the choice of the appropriate presentation technique depends largely on the user. The aim is to efficiently and effectively communicate the results of an analysis. For confirmatory analysis, one or more hypotheses about the data serve as a starting point. The process can be described as a goal-oriented examination of these hypotheses. As a result, visualization either confirms these hypotheses or rejects them. *Exploratory data analysis* as the process of searching and analyzing databases to find implicit but potentially useful information, is a difficult task. At the beginning, the analyst has no hypothesis about the data. According to John Tuckey, tools as well as understanding are needed [3] for the interactive and usually undirected search for structures and trends.

Visual analytics is more than only visualization. It can rather be seen as an integral approach combining visualization, human factors and data analysis. Figure 1 illustrates the detailed scope of visual analytics. Concerning the field of visualization, visual analytics integrates methodology from information analytics, geospatial analytics, and scientific analytics. Especially human factors (e.g., interaction, cognition, perception, collaboration, presentation, and dissemination) play a key role in the communication between human and computer, as well as in the decision-making process. In this context, *production* is defined as the creation of materials that summarize the results of an analytical effort, *presentation* as the packaging of those materials in a way that helps the audience understand the analytical results in context using terms that are meaningful to them, and *dissemination* as the process of sharing that information with the intended audience [4]. In matters of data analysis, visual analytics furthermore profits from methodologies developed in the fields of data management &

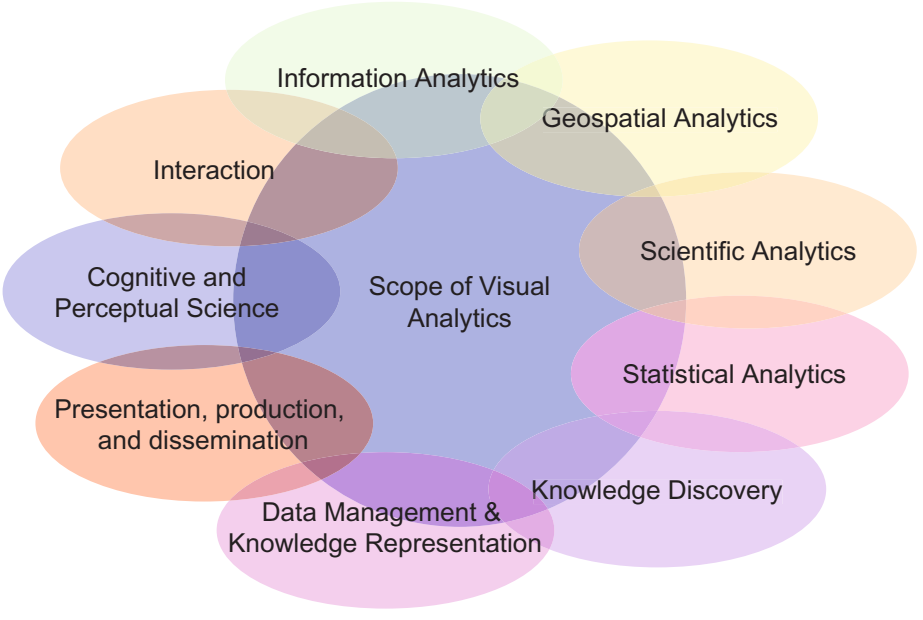


Fig. 1. The Scope of Visual Analytics

knowledge representation, knowledge discovery and statistical analytics. Note that visual analytics, is not likely to become a separate field of study [5], but its influence will spread over the research areas it comprises.

According to Jarke J. van Wijk, "visualization is not 'good' by definition, developers of new methods have to make clear why the information sought cannot be extracted automatically" [6]. From this statement, we immediately see the need for the visual analytics approach using automatic methods from statistics, mathematics and knowledge discovery in databases (KDD) wherever they are applicable. Visualization is used as a means to efficiently communicate and explore the information space when automatic methods fail. In this context, human background knowledge, intuition and decision-making either cannot be automated or serve as input for the future development of automated processes.

Overlooking a large information space is a typical visual analytics problem. In many cases, the information at hand is conflicting and needs to be integrated from heterogeneous data sources. Moreover, the system lacks knowledge that is still hidden in the expert's mind. By applying analytical reasoning, hypotheses about the data can be either affirmed or discarded and eventually lead to a better understanding of the data, thus supporting the analyst in his task to gain insight. Contrary to this, a well-defined problem where the optimum or a good estimation can be calculated by non-interactive analytical means would rather not

be described as a visual analytics problem. In such a scenario, the non-interactive analysis should be clearly preferred due to efficiency reasons. Likewise, visualization problems not involving methods for automatic data analysis do not fall into the field of visual analytics.

The fields of visualization and visual analytics both build upon methods from scientific analytics, geospatial analytics and information analytics. They both profit from knowledge out of the field of interaction as well as cognitive and perceptual science. They do differentiate in so far as visual analytics furthermore integrates methodology from the fields of statistical analytics, knowledge discovery, data management & knowledge representation and presentation, production & dissemination.

3 Visual Analytics Process

In this section we provide a formal description of the visual analytics process. As described in the last section the input for the data sets used in the visual analytics process are heterogeneous data sources (i.e., the internet, newspapers, books, scientific experiments, expert systems). From these rich sources, the data sets $S = S_1, \ldots, S_m$ are chosen, whereas each $S_i, i \in (1, .., n)$ consists of attributes $A_{i1}, \ldots, A_{ik}$. The goal or output of the process is insight I. Insight is either directly obtained from the set of created visualizations V or through confirmation of hypotheses H as the results of automated analysis methods. We illustrated this formalization of the visual analytics process in Figure 2. Arrows represent the transitions from one set to another one.

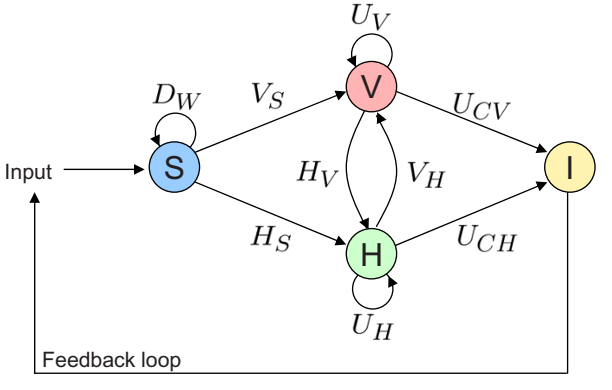


Fig. 2. Visual Analytics Process

More formal the visual analytics process is a transformation $F : S \rightarrow I$, whereas F is a concatenation of functions $f \in \{D_W, V_X, H_Y, U_Z\}$ defined as follows:

D_W describes the basic data pre-processing functionality with $D_W : S \rightarrow S$ and $W \in \{T, C, SL, I\}$ including data transformation functions D_T, data

cleaning functions D_C, data selection functions D_{SL} and data integration functions D_I that are needed to make analysis functions applicable to the data set.

$V_W, W \in \{S, H\}$ symbolizes the visualization functions, which are either functions visualizing data $V_S : S \rightarrow V$ or functions visualizing hypotheses $V_H : H \rightarrow V$.

$H_Y, Y \in \{S, V\}$ represents the hypothesis generation process. We distinguish between functions that generate hypotheses from data $H_S : S \rightarrow H$ and functions that generate hypotheses from visualizations $H_V : V \rightarrow H$.

Moreover, user interactions $U_Z, Z \in \{V, H, CV, CH\}$ are an integral part of the visual analytics process. User interactions can either effect only visualizations $U_V : V \rightarrow V$ (i.e., selecting or zooming), or can effect only hypotheses $U_H : H \rightarrow H$ by generating a new hypotheses from given ones. Furthermore, insight can be concluded from visualizations $U_{CV} : V \rightarrow I$ or from hypothesis $U_{CH} : H \rightarrow I$

The typical data pre-processing applying data cleaning, data integration and data transformation functions is defined as $D_P = D_T(D_I(D_C(S_1, \ldots, S_n)))$. After the pre-processing step either automated analysis methods $H_S = \{f_{s1}, \ldots, f_{sq}\}$ (i.e., statistics, data mining, etc.) or visualization methods $V_S : S \rightarrow V, V_S = \{f_{v1}, \ldots, f_{vs}\}$ are applied to the data, in order to reveal patterns as shown in Figure 2.

The application of visualization methods can hereby directly provide insight to the user, described by U_{CV}; the same applies to automatic analysis methods U_{CH}. However, most application scenarios may require user interaction to refine parameters in the analysis process and to steer the visualization process. This means that after having obtained initial results from either the automatic analysis step or the visualization step, the user may refine the achieved results by applying another data analysis step, expressed by U_V and U_H. Furthermore visualization methods can be applied to the results of the automated analysis step to transform a hypotheses into a visual representation V_H or the findings extracted from visualizations may be validated through an data analysis step to generated a hypotheses H_V. F(S) is rather an iterative process than a single application of each provided function, as indicated by the feedback loop in Figure 2. The user may refine input parameters or focus on different parts of the data in order to validate generated hypotheses or extracted insight.

We take a visual analytics application for monitoring network security as an example. Within the network system, four sensors measure the network traffic resulting in four data sets $S_1, \ldots, S_4$. While preprocessing, the data is cleaned from missing values and unnecessary data using the data cleaning function d_c, integrated using d_i (each measurement system stores data slightly different), and transformed in a format suitable for our analysis using d_t. We now select UDP and TCP traffic for our analysis with the function d_s, resulting in $S' = d_s(d_t(d_i(d_c(S_1, \ldots, S_4))))$. For further analysis, we apply a data mining algorithm h_s to search for security incidents within the traffic generating a hypothesis $h' = h_s(S')$. To better understand this hypothesis, we visualize it using the function v_h: $v' = v_h(h')$. Interactive adjustment of the parameters results in $v'' = u_v(v')$, revealing a correlation of the incidents from two specific source

networks. By applying the function h_v, we obtain a distribution of networks where similar incidents took place $h'' = h_v(v'')$. This leads to the insight that a specific network worm tries to communicate with our network from 25 source networks $i' = u_{ch}(h'')$. Repeating the same process at a later date by using the feedback loop reveals a much higher spread of the virus, emphasizing the need to take countermeasures.

Unlike described in the information seeking mantra ("overview first, zoom/ filter, details on demand") [7], the visual analytics process comprises the application of automatic analysis methods before and after the interactive visual representation is used like demonstrated in the example. This is primarily due to the fact that current and especially future data sets are complex on the one hand and too large to be visualized straightforward on the other hand. Therefore, we present the visual analytics mantra:

"Analyse First -
Show the Important -
Zoom, Filter and Analyse Further -
Details on Demand"

4 Application Challenges

For the advancement of the research field of visual analytics several application and technical challenges need to be mastered. In this section, we present the ten most significant application challenges and discuss them in the context of research projects trying to solve the challenges. Both the application (this section), as well as the technical challenges (next section) were identified by the panel discussion on the Workshop on Visual Analytics in 2005 [8].

4.1 Physics and Astronomy

One major field in the area of visual analytics covers physics and astronomy, including applications like flow visualization, fluid dynamics, molecular dynamics, nuclear science and astrophysics, to name just a few of them.

Especially the research field of astrophysics offers a wide variety of usage scenarios for visual analytics. Never before in history scientists had the ability to capture so much information about the universe. Massive volumes of unstructured data, originating from different directions of the orbit and covering the whole frequency spectrum, form continuous streams of terabytes of data that can be recorded and analysed. The amount of data is so high that it far exceeds the ability of humans to consider it all. By common data analysis techniques like knowledge discovery, astronomers can find new phenomena, relationships and useful knowledge about the universe, but although a lot of the data only consists of noise, a visual analytics approach can help separating relevant data from noise and help identifying unexpected phenomena inside the massive and dynamic data streams. One celebrated example is the Sloan Digital Sky Survey [10] and the COMPLETE project [11], generating terabytes of astrophysics

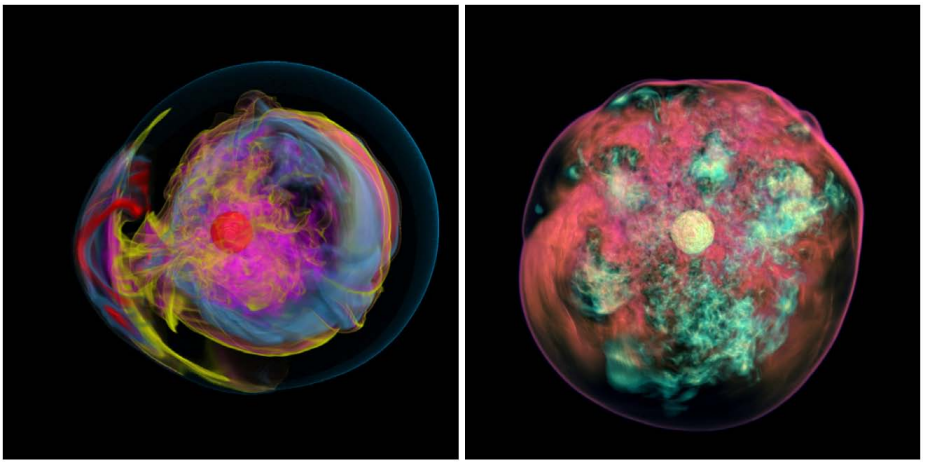


Fig. 3. A visual approach to illustrate the complex relationships within a Supernova (© 2005 IEEE) [9]. The 3D simulation processes tens of terabytes of data (turbulence, rotation, radiation, magnetic fields, gravitational forces) to generate a visual output that can then be analysed to discover further insights.

data each day, or the Large Hadron Collider (LHC) at CERN which generates a volume of 1 petabyte of data per year.

One example for a visual analytics application is the simulation of a Supernova. The SciDAC program has brought together tremendous scientific expertise and computing resources within the Terascale Supernova Initiative (TSI) project to realize the promise of terascale computing for attempting to answer some of the involved questions [9]. A complete understanding of core collapse supernovae requires 3D simulations of the turbulence, rotation, radiation, magnetic fields and gravitational forces, producing tens of terabytes of data per simulation. As an examination of this massive amount of data in a numeric format would simply exceed human capabilities and would therefore not give an insight into the processes involved, a visual approach (see Fig. 3) can help analyzing these processes on a higher aggregated level in order to draw conclusions and extract knowledge from it.

4.2 Business

Another major field in the area of visual analytics covers business applications. The financial market with its thousands of different stocks, bonds, futures, commodities, market indices and currencies generates a lot of data every second, which accumulates to high data volumes throughout the years. The main challenge in this area lies in analyzing the data under multiple perspectives and assumptions to understand historical and current situations, and then monitoring the market to forecast trends and to identify recurring situations. Visual analytics applications can help analysts obtaining insights and understanding into

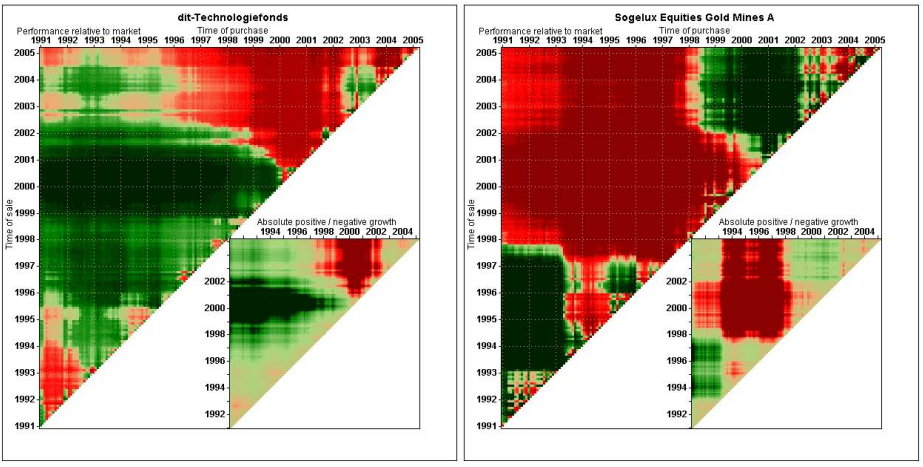


Fig. 4. Visual analysis of financial data with the FinDEx system [12]. The growth rates for time intervals are triangulated in order to visualize all possible time frames. The small triangle represents the absolute performance of one stock, the big triangle represents the performance of one stock compared to the whole market.

previous stock market development, as well as supporting the decision making progress by monitoring the stock market in real-time in order to take necessary actions for a competitive advantage, with powerful means that reach far beyond the numeric technical chart analysis indicators or traditional line charts. One popular application in this field is the well-known Smartmoney [13], which gives an instant visual overview of the development of the stock market in particular sectors for a user-definable time frame. A new application in this field is the FinDEx system [12] (see Fig. 4), which allows a visual comparison of a fund's performance to the whole market for all possible time intervals at one glance.

4.3 Environmental Monitoring

Monitoring climate and weather is also a domain which involves huge amounts of data collected throughout the world or from satellites in short time intervals, easily accumulating to terabytes per day. Applications in this domain most often do not only visualize snapshots of a current situation, but also have to generate sequences of previous developments and forecasts for the future in order to analyse certain phenomena and to identify the factors responsible for a development, thus enabling the decision maker to take necessary countermeasures (like the global reduction of carbon dioxide emissions in order to reduce global warming). The applications for climate modeling and climate visualization can cover all possible time intervals, from daily weather forecasts which operate in rather short time frames of several days, to more complex visualizations of climate changes that can expand to thousands of years. A visual approach can easily help to interpret these massive amounts of data and to gain insight into

the dependencies of climate factors and climate change scenarios that would otherwise not be easily identified. Besides weather forecasts, existing applications for instance visualize the global warming, melting of the poles, the stratospheric ozone depletion, hurricane warnings or oceanography, to name just a few.

4.4 Disaster and Emergency Management

Despite the slowly arising environmental changes like global warming that have been mentioned above, environmental or other disasters can face us as sudden major catastrophes. In the domain of emergency management, visual analytics can help determining the on-going progress of an emergency and can help identifying the next countermeasures (construction of physical countermeasures or evacuation of the population) that must be taken to limit the damage. Such scenarios can include natural or meteorological catastrophes like flood or waves, volcanos, storm, fire or epidemic growth of diseases (bird flu), but also human-made technological catastrophes like industrial accidents, transport accidents or pollution. Depending on the particular case, visual analytics can help to determine the amount of damage, to identify objectives, to assign priorities, and to provide effective coordination for various organizations for more efficient help in the disaster zone.

4.5 Security

Visual analytics for security is an important research topic and is strongly supported by the U.S. government. The application field in this sector is wide, ranging from terrorism informatics over border protection to network security. In these fields, the challenges lie in getting all the information together and linking numerous incidents to find correlations.

A demonstrative example of work in the field is the situational awareness display *VisAware* [14] which is built upon the w^3 premise, assuming that every incident has at least the three attributes what, when, and where (see Fig. 5). In this display, the location attribute is placed on a map, the time attribute indicated on concentric circles around this map, and the classification of the incident is mapped to the angle around the circle. For each incident, the attributes are linked through lines. Other examples in the field are [15] and [16].

4.6 Software Analytics

Visual software analytics has become a popular research area, and as modern software packages often consist of millions of code lines it can support a faster understanding of the structure of a software package with its dependencies. Visual analytics tools can not only help revealing the structure of a software package, but can also be used for various other tasks like debugging, maintenance, restructuring or optimization, therefore reducing software maintenance costs. Two applications in this field are CVSscan [17] for interactively tracking the changes of a software package over time, or the Voronoi treemaps [18] for visualization of software metrics.

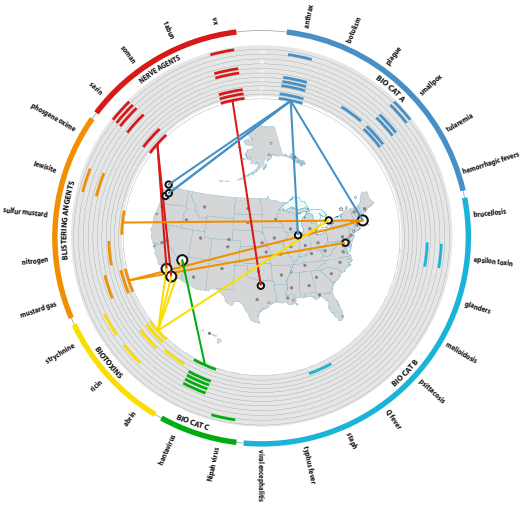


Fig. 5. VisAware for BioWatch (© 2005 IEEE) [14]

4.7 Biology, Medicine and Health

The research fields in biology and medicine offer a very wide variety of applications. As computer tomography and ultrasound imaging in the medical area for 3-dimensional digital reconstruction and visualization have been widely used for years, especially the emerging area of bio-informatics now offers a lot of possible applications for visual analytics. From the early beginning of sequencing, scientist in these areas face unprecedented volumes of data, like in the Human Genome Project with three billion base pairs per human. Other new areas like Proteomics (studies of the proteins in a cell), Metabolomics (systematic study of unique chemical fingerprints that specific cellular processes leave behind) or combinatorial chemistry with tens of millions of compounds even enlarge the amount of data every day. A brute-force computation of all possible combinations is often not possible, but visual approaches can help to identify the main regions of interest and exclude areas that are not promising. As traditional visualization techniques can not cope with these amounts of data, new and more effective visualizations are necessary to analyze this amount of data ([19], [20]).

4.8 Engineering Analytics

The application field in engineering analytics covers the whole range from engineering to construction, with a lot of parallels to physics (see above). The most important application is also flow visualization, regarding the automotive industry for example optimization of the air resistance of vehicles, optimization of the flows inside a catalytic converter or diesel particle filter, or computation of optimal air flows inside an engine [21]. Instead of only solving these problems algorithmically, visual analytics can help to understand the flows, and to

interactively change construction parameters to optimize the flows. Another application in the automotive industry is the simulation of a car crash, where the frame of a car is represented as a grid of hundreds of thousands of points and the crash is simulated inside a computer. As an optimal car frame cannot be fully automatically computed, visual analytics can help engineers to understand the deformation of the frame during a crash step by step, and to identify the key-points where optimization of the frame is necessary for a better overall stability.

4.9 Personal Information Management

The field of personal information management has many facets and is already affecting our everyday life through digital information devices such as PDAs, mobile phones, and laptop computers. However, there are many further possibilities where research might help to form our future. One example is the IBM Remail project [22], which tries to enhance human capabilities to cope with email overload. Concepts like "Thread Arcs", "Correspondents Map", and "Message Map" support the user to efficiently analyse his personal email communication. MIT's project Oxygen [23] goes one step further, by addressing the challenges of new systems to be pervasive, embedded, nomadic, adaptable, powerful, intentional and eternal. Many of those challenges reflect the visual analytics approach to combine human intelligence and intuition with computational resources.

4.10 Mobile Graphics and Traffic

As an example for traffic monitoring, we consider an ongoing project at University of Illinois-Chicago National Center for Data Mining [24]. In this project, traffic data from the tri-state region (Illinois, Indiana, and Wisconsin) are collected from hundreds of embedded sensors. The sensors are able to identify vehicle weights and traffic volumes. There are also cameras that capture live video feeds, Global Positioning System (GPS) information from selected vehicles, textual accident reports, and weather information. The research challenge is to integrate this massive information flow, provide visualizations that fuse this information to show the current state of the traffic network, and develop algorithms that will detect changes in the flows. Part of this project will involve characterizing normal and expected traffic patterns and developing models that will predict traffic activity when stimulus to the network occurs. The changes detected will include both changes in current congestion levels and differences in congestion levels from what would be expected from normal traffic levels.

5 Technical Challenges

To complete the list of challenges of the previous section, we briefly list the 10 most important technical challenges.

The first technical challenge lies in the field of *problem solving, decision science, and human information discourse*. The process of problem solving supported by technology requires understanding of technology on the one hand, but

also comprehension of logic, reasoning, and common sense on the other hand. Intuitive displays and interaction devices should be constructed to communicate analytical results through meaningful visualizations and clear representations.

User acceptability is a further challenge; many novel visualization techniques have been presented, yet their wide-spread deployment has not taken place, primarily due to the users' refusal to change their working routines. Therefore, the advantages of visual analytics tools need to be communicated to the audience of future users to overcome usage barriers, and to eventually tap the full potential of the visual analytics approach. After having developed a system, its *evaluation* is crucial for future reference. Clear comparisons with previous systems to assess its adequacy and objective rules of thumbs to facilitate design decisions would be a great contribution to the community.

To automatically derive *semantics* out of large document collections is still a challenge. On the one hand, some expert systems have been successfully built for specialized fields, but on the other hand the researched methods only perform reasonably within a limited scope. Unlike human comprehension, automatic methods often fail to recognize complex coherences for which they have not been explicitly trained. Modeling of semantics to better deal with conflicting and incomplete information is therefore a challenging field.

Data quality and uncertainty is an issue in many domains, ranging from terrorism informatics to natural sciences, and needs to be taken into account when designing algorithms and visualization metaphors. Semiotic misinterpretations can occur easily. *Data provenance* as the science of understanding where data has come from and why it arrived in the user's database [25] is closely connected to the latter topic. In application fields such as biology, experimental data is made publicly available on the web, copied into other databases, and transformed several times (data curation). Seldom, this information about the transformations and the origins of the data under consideration is properly documented, although it is indispensable for the reproducibility of scientific results. Another challenge lies in *data streams* producing new data at astonishing pace. In this field, especially the timely analysis of the data streams plays an important role. In many cases, e.g. network traffic monitoring, detailed information is abundant and in the long term storage capacities do not suffice to log all data. Thus, efficient and effective methods for compression and feature extraction are needed.

Due to improved measurement methods and decreasing costs of storage capacities, data sets keep on growing. Eventually, *scalability* becomes a major problem in both, automatic as well as visual analysis ([26], [27]), as it becomes more and more challenging to analyze these data sets. For more details see [1], page 24ff "The Scalability Challenge".

Real-world applications often consist of a series of heterogeneous problems. While solving one or more of these problems might still be accomplishable, their correlation make it very difficult to solve the overall problem, thus turning *synthesis of problems* into another challenge. It soon becomes apparent that *integration* with automated analysis, databases, statistics, design, perception, etc. comprises the last of the technical challenges.

6 Conclusion

Visual analytics is an emerging field of research combining strengths from information analytics, geospatial analytics, scientific analytics, statistical analytics, knowledge discovery, data management & knowledge representation, presentation, production and dissemination, cognition, perception and interaction. Its goal is to gain insight into homogeneous, contradictory and incomplete data through the combination of automatic analysis methods with human background knowledge and intuition.

In this paper, we defined the scope of this emerging field and took a closer look at the visual analytics process. By presenting a formal model of the process, we identified the key concepts (data sets, hypotheses, visualizations and insight) and transition functions from one concept to another. To represent the iterative character of the process, a feedback-loop was introduced starting the process over again. To better understand the new analysis methodology, we presented the visual analytics mantra "analyse first - show the important - zoom, filter and analyse further - details on demand". By means of the top 10 application challenges and the top 10 technical challenges, we gave an overview of the current state of the field and its challenges.

References

1. Thomas, J., Cook, K.: Illuminating the Path: Research and Development Agenda for Visual Analytics. IEEE Press, Los Alamitos (2005)
2. Ware, C.: Information Visualization - Perception for Design, 1st edn. Morgan Kaufmann Publishers, San Francisco (2000)
3. Tuckey, J.W.: Exploratory Data Analysis. Addison-Wesley, Reading (1977)
4. Thomas, J.J., Cook, K.A.: A Visual Analytics Agenda. IEEE Transactions on Computer Graphics and Applications 26(1), 12–19 (2006)
5. Wong, P.C., Thomas, J.: Visual analytics. IEEE Computer Graphics and Applications 24(5), 20–21 (2004)
6. van Wijk, J.J.: The value of visualization. IEEE Visualization, 79–86 (2005)
7. Shneiderman, B.: The eyes have it: A task by data type taxonomy for information visualizations. In: IEEE Symposium on Visual Languages, pp. 336–343 (1996)
8. Keim, D.A., Kohlhammer, J., Thomas, J.: Workshop on visual analytics (2005), http://infovis.uni-konstanz.de/events/ws_visual_analytics_05/
9. Ma, K.-L., Lum, E., Yu, H., Akiba, H., Huang, M.-Y., Wang, Y., Schussman, G.: Scientific discovery through advanced visualization. In: Proceedings of DOE SciDAC 2005 Conference, San Francisco (June 2005)
10. Sloan Digital Sky Survey (2007), http://www.sdss.org/
11. COMPLETE - the COordinated Molecular Probe Line Extinction Thermal Emission survey of star forming regions. (2007), http://cfa-www.harvard.edu/COMPLETE/index.html
12. Keim, D.A., Nietzschmann, T., Schelwies, N., Schneidewind, J., Schreck, T., Ziegler, H.: FinDEx: A spectral visualization system for analyzing financial time series data. In: EuroVis 2006: Eurographics/IEEE-VGTC Symposium on Visualization, Lisbon, Portugal, May 8-10 (2006)

13. Wattenberg, M.: Visualizing the stock market. In: CHI 1999: CHI 1999 extended abstracts on Human factors in computing systems, pp. 188–189. ACM Press, New York (1999)
14. Livnat, Y., Agutter, J., Moon, S., Foresti, S.: Visual correlation for situational awareness. In: IEEE Symposium on Information Visualization, pp. 95–102 (2005)
15. Teoh, S.T., Jankun-Kelly, T., Ma, K.-L., Wu, S.F.: Visual data analysis for detecting flaws and intruders in computer network systems. IEEE Transactions on Computer Graphics and Applications, September/October 2004, 27–35 (2004)
16. Goodall, J.R., Lutters, W.G., Rheingans, P., Komlodi, A.: Preserving the big picture: Visual network traffic analysis with TNV. In: Proceedings of IEEE Workshop on Visualization for Computer Security, pp. 47–54 (2005)
17. Voinea, S., Chaudron, A.T.M.: Version-centric visualization of code evolution. In: Proceedings of Eurographics/IEEE-VGTC Symposium on Visualization (2005)
18. Balzer, M., Deussen, O.: Voronoi treemaps. In: IEEE Symposium on Information Visualization (InfoVis 2005), pp. 7–14 (2005)
19. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. Journal on Molecular Biology 215(3), 403–410 (1990)
20. Tatusova, T., Madden, T.: Blast2 sequences - a new tool for comparing protein and nucleotide sequences. FEMS Microbiology Letter 174(2), 247–250 (1999)
21. Doleisch, H., Mayer, M., Gasser, M., Wanker, R., Hauser, H.: Case study: Visual analysis of complex, time-dependent simulation results of a diesel exhaust system. In: 6th Joint IEEE TCVG -EUROGRAPHICS Symposium on Visualization (VisSym 2004), May 2004, pp. 91–96 (2004)
22. IBM Remail - reinventing email (2005), `http://www.research.ibm.com/remail/`
23. MIT Project Oxygen (2007), `http://oxygen.lcs.mit.edu/`
24. Pantheon Highway Gateway (2007), `http://highway.lac.uic.edu/`
25. Buneman, P., Khanna, S., Tan, W.-C.: Why and Where: A Characterization of Data Provenance. In: Van den Bussche, J., Vianu, V. (eds.) ICDT 2001. LNCS, vol. 1973, p. 316. Springer, Heidelberg (2000)
26. Chen, C.: Top 10 unsolved information visualization problems. IEEE Transactions on Computer Graphics and Applications 25(4), 12–19 (2005)
27. Eick, S.G., Karr, A.F.: Visual scalability. Journal of Computational & Graphical Statistics, 22–43 (March 2002)

Using Nested Surfaces for Visual Detection of Structures in Databases

Arturas Mazeika[1,2], Michael H. Böhlen[1], and Peer Mylov[2]

[1] Faculty of Computer Science, Free University of Bozen-Bolzano, Dominikanerplatz 3, 39100 Bozen, Italy
{arturas,boehlen}@inf.unibz.it
[2] Institute of Communication, Aalborg University, Niels Jernes Vej 14, 9220 Aalborg, Denmark
mylov@hum.auc.dk

Abstract. We define, compute, and evaluate *nested surfaces* for the purpose of visual data mining. Nested surfaces enclose the data at various density levels, and make it possible to equalize the more and less pronounced structures in the data. This facilitates the detection of multiple structures, which is important for data mining where the less obvious relationships are often the most interesting ones. The experimental results illustrate that surfaces are fairly robust with respect to the number of observations, easy to perceive, and intuitive to interpret. We give a topology-based definition of nested surfaces and establish a relationship to the density of the data. Several algorithms are given that compute surface grids and surface contours, respectively.

1 Introduction

Visual data mining exploits the human perceptual faculties to detect interesting relationships in the data. To support the detection of relationships it is important to visualize data in a form that is easy understandable to humans. It is common to use scatter plots for this purpose [2]. Employing scatter plots is intuitive as each observation is faithfully displayed. Scatter plots have successfully been used for detecting relationships in two dimensions. For higher dimensions scatter plots are combined with grand tour methods. A grand tour displays a smooth rotation of two dimensional projections that eventually covers the entire high dimensional search space.

Scatter plots hit limitations if the dataset is big, noisy, or if it contains multiple structures. With lots of data the amount of displayed objects makes it difficult to detect any structure at all. Noise easily blurs the picture and can make it impossible to find interesting relationships. Finally, with multiple structures it often happens that one structure is more pronounced than another. In this situation the less pronounced structures easily get lost. For the purpose of data mining this is particularly bad as it is usually the less obvious relationships that are the most interesting ones. Surfaces equalize the more and less pronounced structures and thus support the detection of less obvious relationships.

In this chapter we explore the potential of nested surfaces to analyze data sets. Nested surfaces enclose the data at varying densities. Humans are used to perceive surface information and to abstract surfaces from individual observations. This greatly simplifies the interpretation of the data. Nested surfaces do not suffer if the amount of

S.J. Simoff et al. (Eds.): Visual Data Mining, LNCS 4404, pp. 91–102, 2008.

data is big, and the nesting supports the detection of multiple structures. We provide a topology-based definition of surfaces and prove that the boundary $\partial C_\alpha = \partial\{(x, y, z) : f(x, y, z) \geq \alpha\}$ is a surface if the density function f has a continuous derivative. This provides the basis for an algorithm that computes nested level surfaces. Given a density estimation, which has continuos derivative, and a density level α we give algorithms that compute surface grids and surface contours, respectively. The described methods have been implemented and integrated into the 3D Visual Data Mining (3DVDM) System. The 3DVDM System is used for explorative data analyses in a 6-sided Cave, a 180° Panorama, and on regular computer monitors. It can be downloaded from `http://www.cs.auc.dk/3DVDM` and runs on SGI and PC/Linux computers.

The nested surface method works well with continuous datasets that contain multiple structures. We expect that the method will also work fine for categorical data. In this case, the ordering of dimensions and other parameters may be significant and can yield different visual results.

Usually, high number of observations overloads scatter plots. In contrast, nested surfaces produce nice results. The visualization is not affected by a high number of observations and it is continuously improving as the number of observations increases.

The computation of nested surfaces for the purpose of data mining has only received scant attention. Mostly surfaces have been investigated in connection with advanced visualization techniques, such as rendering, lighting, transparency, or stereoscopy [4,7,8,12]. These approaches focus on methods and data structures related to visualization aspects. Our goal is the computation of the defining structure of surface that emphasize the structure of the data.

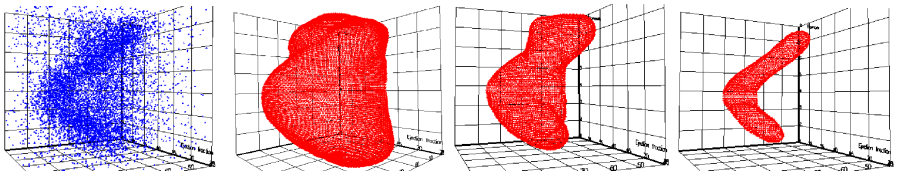
The chapter proceeds as follows. We motivate our method in Section 2. Section 3 provides background material about probability density functions (PDFs), kernel estimations, clustering, and outliers. Section 4 defines surfaces. Section 5 gives algorithms for computing PDF estimates, level grid surfaces, and level contour surfaces. The algorithms are evaluated in Section 6. Section 7 discusses experimental results. Section 8 summarizes the chapter and points to future work.

2 Motivation

Scatter plots are used to find structures in data. These structures are usually described as an accumulation of points. Scatter plots are good in getting a first impression of the data set, but they have a number of disadvantages. On one hand it is hard to understand very dense regions since data points hide each other. On the other hand it is also difficult to investigate sparse regions since data points in sparse areas are easily perceived as noise.

To illustrate our method we use the Spiral–Line data set presented in Figure 1(a). The data set consist of a vertical line in the middle (4'000 points), a spiral curve around the line (4'000 points) and uniformly distributed noise (2'000 points). The data points around the spiral curve form the most dense and notable region. Since the data points around the vertical line have a higher spreadness it is easy to treat it is as noise and not pay attention to it.

Figures 1(b) to 1(d) present the surfaces for different density levels α. Figure 1(b) shows the surface for the lowest density level. This Figure can be used for the detection



(a) Dataset (b) Surface for $\alpha = 1/10m$ (c) Surface for $\alpha = 5/10m$ (d) Surface for $\alpha = 9/10m$

Fig. 1. Spiral–Line DB and Associated Surfaces. m denotes the maximum density in DB.

of outliers. Figures 1(c) and 1(d) show surfaces for higher density levels. Together with Figure 1(b) they emphasize the structure of the data set. Note that the surfaces in Figure 1(c) clearly identify the vertical line *and* the spiral (the quality is much better on the monitor, see also Figures 5 and 6).

In contrast to scatter plot visualizations, surfaces do not deteroriate if the amount of observations increases. Nested surfaces are often easier to interpret than the raw data. Moreover multiple nested surfaces at different density levels facilitate the analysis of the data at different levels of detail. This gives the ability to explore the internal structure of data regions.

3 Preliminaries

3.1 Probability Density Function

Throughout, we assume that the data has been normalized to the three-dimensional unit cube, i.e., each coordinate falls into the [0,1] interval.

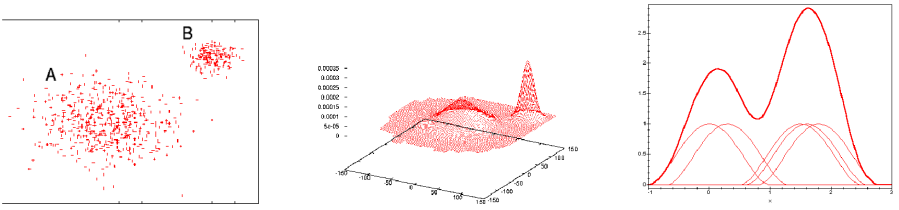
Definition 1. *(Probability density function) Let X be a 3 dimensional random vector with distribution function F. A 3-dimensional real value function f with*

$$F(x, y, z) = \int_{\infty}^{x} \int_{\infty}^{y} \int_{\infty}^{z} f(t, s, q)\, dt\, ds\, dq$$

is a probability density function (PDF).

Figure 2(a) shows a dataset with two clusters: A and B. The corresponding PDF is shown in Figure 2(b). The PDF shows the density of the dataset. Since the density of region A is lower than the density of region B the PDF value for region B is higher than for region A. The PDF also shows that region A is more spread than region B.

In general, we have to estimate the PDF because we are dealing with random datasets for which the true PDF is unknown. Different PDF estimates were proposed in the literature, with the kernel method being one of the most general ones [3,1,11,10,5]. The essence of the kernel method is that each observation increases the chances of having another observation nearby. Therefore, we draw a symmetric kernel with an area equal to 1 around each observation. Adding all kernels (cf. Figure 2(c)) yields an estimate for the PDF. To control the influence of one observation on the overall estimation a



(a) Dataset D (b) PDF for Dataset D (c) Kernel Addition

Fig. 2. Dataset and Corresponding PDF

smoothing parameter h is introduced. The kernel estimate [9] for a a set of observations, $(X_i, Y_i, Z_i), i = 1, \ldots, n$, at point (x, y, z) is defined as follows:

$$\hat{f}_K(x, y, z) = \frac{1}{nh^3} \sum_{k=1}^{n} K\Big(\frac{x - X_i}{h}, \frac{y - Y_i}{h}, \frac{z - Z_i}{h}\Big), \tag{1}$$

where K is a function (kernel) with $K \geq 0$, $\int K = 1$, and $K(x) = K(-x)$.

Various kernels K have been proposed in the statistical literature. Examples include square wave or Gaussian functions. It has been shown [11] that the accuracy of the estimation depends mostly on the smoothing parameter h and less on the choice of the kernel K. Parzen [1] showed that the smoothing parameter

$$h = h_{opt} = c(K, \sigma_1, \sigma_2, \sigma_3)/n^{-1/7} \tag{2}$$

minimizes the mean integrated square error (MISE):

$$\text{MISE} = \mathbf{E} \iiint (\hat{f}(x, y, z) - f(x, y, z))^2 \, dx \, dy \, dz. \tag{3}$$

c is constant for a given dataset and depends on the variance $(\sigma_1^2, \sigma_2^2, \sigma_3^2)$ of the random vector (X_1, X_2, X_3) and the kernel function K.

3.2 Clusters and Outliers

Density functions are also used to define clusters and outliers. Clusters and outliers are important characteristics of a dataset, and they are often used for data analysis. In the next section we will establish a relationship between clusters and surfaces. Let D be a set of 3 dimensional points, and let $(\mathbf{x}, \mathbf{x}^*] = \{\mathbf{x}t + \mathbf{x}^*(1-t), t \in (0, 1]\}$ be an interval in the three-dimensional space.

Definition 2. *(Cluster) A cluster for a set of local maxima M of the density function f and threshold ξ is the subset*

$$C = \{\mathbf{x} \in D \mid \forall \mathbf{x}^* \in M \wedge \forall \mathbf{y} \in (\mathbf{x}, \mathbf{x}^*] : f(\mathbf{y}) \geq \xi\}.$$

Definition 3. *(Outliers) The points $O \subseteq D$ are* outliers *iff for all local maxima $\mathbf{x}^*$ of the density function f*

$$O = \{\mathbf{x} \in D \mid \forall \mathbf{y} \in (\mathbf{x}, \mathbf{x}^*] : f(\mathbf{y}) < \xi\}.$$

Thus, a cluster is a set that contains PDF center (maxima) points together with all surrounding points that "exceed noise level ξ".

4 Surface Definition

We use a topological approach to define a surface. Intuitively, a surface is a set of points iff the neighbourhood of any point is similar to a two–dimensional open disk. To define the resemblance with an open disk we use a homeomorphic (one-to-one, continuous inverse) function.

Definition 4. *(Elementary surface) Let f be a function that maps an open disc D^2 to a set of points S. S is an* elementary surface *iff f is homeomorphic.*

Definition 5. *(Surface) A* surface *is a connected set of points iff the neighbourhood of any point of the surface is an elementary surface.*

Next, we establish a relationship between the border of a cluster and a surface. A border is a set of points: $\partial C = [C] \backslash C^{\circ}$ where $[C]$ contains the limit points of C and C° contains the inner points of C. We show that ∂C is a surface by giving a parametrisation function that maps a disk D^2 into ∂C.

Theorem 1. *(Implicit function theorem) Suppose $f : R^n \times R^m \rightarrow R^m$ is differentiable in an open set around (u, v) and $f(u, v) = 0$. Let M be the $m \times m$ matrix given by*

$$M = \left(\frac{\partial f_i(u)}{\partial x_{n+j}}\right) \quad 1 \leq i, j \leq m.$$

If $\det M \neq 0$ then there is an open set $U \subset R^n$ that contains u and an open set V that contains v, such that for each $r \in U$ there exists $s \in V$ and $f(r, s) = 0$. If we define $g : U \rightarrow V$ as $g(r) = s$, then g is differentiable.

The implicit function theorem is a classical result and ensures the existence of a cluster boundary parametrisation. A proof can be found for example in [6].

Lemma 1. *Let f be a probability density function which has continuous derivative ($f \in C^1$), and C be a cluster for a set of maxima M and level noise ξ. Let grad $f(x) \neq 0, x \in \partial C$. Then ∂C is a surface.*

Proof. Notice that $\partial C = \{x \in D : f(x) = \xi\}$. Let $(a, b, c) \in \partial C$. Since grad$f \neq 0$ there is at least one coordinate x_i such that $\partial f / \partial x_i \neq 0$ at point (a, b, c). Then the implicit function theorem with $m = 1$ and $v = x_i$ proofs the lemma.

5 Algorithms

This section gives algorithms to compute nested surfaces: `Surface_GridPoints` and `Surface_GridLines`. Starting from the raw data, the first step is the estimation

of the PDF. We scan the (sample of the) data set twice to estimate the PDF. The first scan is used to calculate the estimation parameters (cf. Equation (2)).

The second scan is used to compute the actual PDF estimation. We use the Epanechninikov kernel [1], which is equal to 0 outside the area $t_1^2 + t_2^2 + t_3^2 \leq 1$. Thus, only observations that fall into the area $\{(t_1, t_2, t_3) : (t_1 - x)^2 + (t_2 - y)^2 + (t_3 - z)^2 \leq h^2\}$ influence the estimated PDF value at point (x, y, z).

Algorithm: PDF_Estimation

Input:
  Database with n observations: $(X[i], Y[i], Z[i]),\ i = 1, \ldots, n$
  Number of grid points in each dimension: g

Output:
  Data cube with PDF values on grid points: PDF

Body:
  1. Initialize PDF
  2. Calculate estimation parameters according to Equation (2)
  3. FOR $i = 1$ TO n DO
     3.1. Determine the set of PDF points $\mathcal{A}_g$ that are
          influenced by the data point $(X[i], Y[i], Z[i])$
     3.2. FOR EACH $(k, l, m) \in \mathcal{A}_g$ DO

$$PDF[k,l,m] = PDF[k,l,m] + K(\tfrac{k - X_i}{h}, \tfrac{l - Y_i}{h}, \tfrac{m - Z_i}{h})$$

The Surface_GridPoints algorithm calculates the border $B = \partial\{(x, y, z) : f(x, y, z) \geq \alpha\}$. The basic idea of the algorithm is to scan the PDF and compare each value against its neighbours: if the value is greater than α and there exists a point in the neighborhood that is less than α then the value is added to B.

Algorithm: Surface_GridPoints

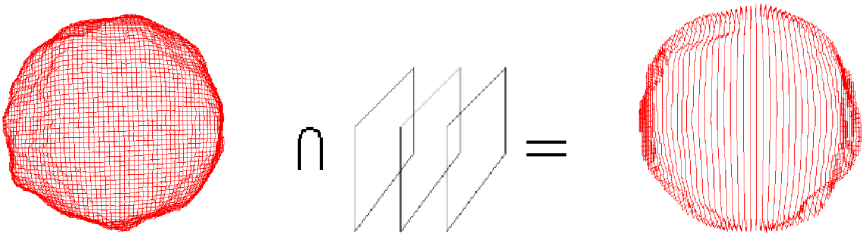
Input:
  Number of grid lines per dimension: g
  Data cube with PDF grid point values: PDF
  Density level: α

Output:
  Surface grid: B

Body:
  1. FUNCTION IsBorderPoint(PDF,i,j,k)
  2. RETURN ($PDF[i,j,k] \geq \alpha$) AND ($PDF[i',j',k'] < \alpha$)
          for some $(i', j', k') \in (i + h_1, j + h_2, k + h_3)$
          where $h_1, h_2, h_3 = -1, 0, 1,\ |h_1| + |h_2| + |h_3| = 1$
  3. END FUNCTION

  4. $B = \emptyset$
  5. FOR $i, j, k = 1$ TO g DO
     5.1 IF IsBorderPoint(PDF, i, j, k) THEN $B = B \cup PDF[i, j, k]$

The `Surface_GridLines` algorithm extends the `Surface_GridPoints` algorithm. The main idea of the algorithm is to draw contour curves on the surface. These curves, in turn, are calculated by intersecting a surface with cutting planes parallel to the XY, ZY, and ZX planes (cf. Figure 3).

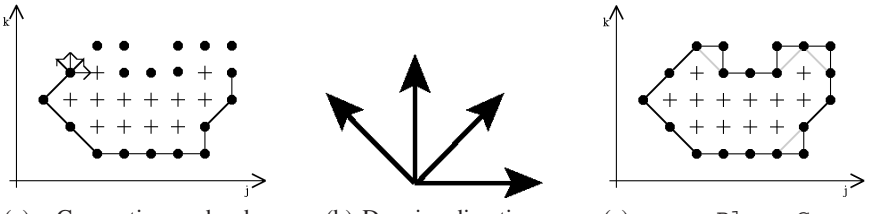


(a) Surface (b) Intersecting planes (c) Vertical Grid-Lines-Surface

Fig. 3. Grid-Line-Surface

The idea of plane curve's calculation is presented in Figure 3. We scan the PDF values with a condition $i = i_0$ for ZY planes, $j = j_0$ for ZX planes, and $k = k_0$ for XY planes.

Figure 4(a) shows a cutting plane. Border points are shown as filled circles, inner cluster points as plus signs, and outer cluster points are not shown in the picture. The algorithm connects the border points to form a polygon curve.



(a) Connecting border points (b) Drawing directions (c) `Plane_Curve` method's output. Not valid directions are drawn in grey

Fig. 4. Curve computation in intercepting plane

For each PDF border point we are looking for PDF border points in the directions presented in Figure 4(b). Note, that we scan PDF from left to right, from bottom to top. Therefore, there is no need to draw lines to the bottom and to the left.

We make vertical and horizontal connections between border points. For diagonal we make additional checks. We do not draw diagonal line if there are two lines in its neighborhood (cf. Figure 4(c)). With this we avoid squares with crossing diagonals

inside. The Individual steps of the ZY plain curve calculation are presented in the `ZY_Plane_Curve` algorithm.

Algorithm: ZY_Plane_Curve

```
Input:
  ZY plane number: i_0
  Data cube with PDF grid point values: PDF
Output:
  polygonal contour line on ZY plane at level i_0: C = C_{i_0}^{ZY}
Body:
  1. C = ∅, i = i_0
  2. FOR j,k = 1 TO g DO
     2.1 IF IsBorderPoint(PDF,i,j,k) THEN
       IF IsBorderPoint(PDF,i,j+1,k) THEN
         C = C ∪ line(i,j,k, i,j+1,k)
       IF IsBorderPoint(PDF,i,j,k+1) THEN
         C = C ∪ line(i,j,k, i,j,k+1)
       IF IsBorderPoint(PDF,i,j-1,k+1) AND
          ¬IsBorderPoint(PDF,i,j-1,k) AND
          ¬IsBorderPoint(PDF,i,j,k+1) THEN
            C = C ∪ line(i,j,k, i,j-1,k+1)
       IF IsBorderPoint(PDF,i,j+1,k+1) AND
          ¬IsBorderPoint(PDF,i,j+1,k) AND
          ¬IsBorderPoint(PDF,i,j,k+1) THEN
            C = C ∪ line(i,j,k, i,j-1,k+1)
```

Algorithm: Surface_GridLines

```
Input:
  Data cube with PDF grid point values: PDF
Output:
  Contour lines on the surface: C
Body:
  1. C = ∅
  2. FOR i = 1 TO g DO C = C ∪ ZY_PlaneCurve(PDF,i)
  3. FOR j = 1 TO g DO C = C ∪ ZX_PlaneCurve(PDF,j)
  4. FOR k = 1 TO g DO C = C ∪ XY_PlaneCurve(PDF,k)
```

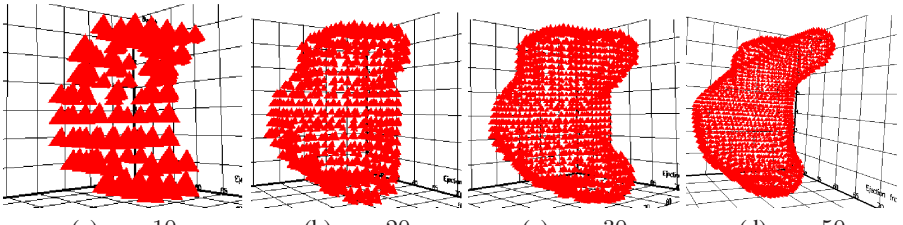
Note, that in order to include a 3D picture into the chapter we have to project it into 2D. We use the `Surface_GridPoints` method to illustrate surfaces on a 2D devices while we use the `Surface_GridLines` method to illustrate surfaces in immersive 3D environments.

6 Evaluation

This section evaluates the quality of the algorithms numerically and visually. The experiments were calculated on the Pentium III 1GHz PC computer with 512MB of main memory running GNU/Linux OS with the gcc compiler.

6.1 Quality of the Surfaces

First, we evaluate the surface quality with respect to the number of grid lines. We use the three-dimensional scatter plot in Figure 1(a) and a single level surface for $\alpha = 1/10m$. In order to get a fair visual comparison of the influence of g on the quality of the surface we let the size of tetrahedra depend on g. It is chosen so that the tetrahedras visually are always near each other. Figures 5(a) and 5(b) show that $g = 10$ and $g = 20$ are not enough for a nice surface. There are too few tetrahedras at the ends of the spiral curve. As g reaches 30 the picture becomes detailed enough.



(a) $g = 10$ (b) $g = 20$ (c) $g = 30$ (d) $g = 50$

Fig. 5. Cluster Surface for $\alpha = 1/10m$ for Varying Values of g

To quantitatively measure the quality of the surfaces we use Equation (4). It quantifies the average error we make at any point (i, j).

$$AE_S = \frac{1}{g^2 \max_{x,y,z} \hat{f}_g(x,y,z)} \sum_{i,j=1}^{g} |\hat{s}_g(i,j) - s(i,j)|, \tag{4}$$

s is the parametrisation function that maps the open unit disk to $\partial C_\alpha = \partial\{(x,y,z) : f(x,y,z) \geq \alpha\}$. Since s is usually unknown we replace it with $\hat{s}_{\bar{g}}$ with a large value for $\bar{g}$:

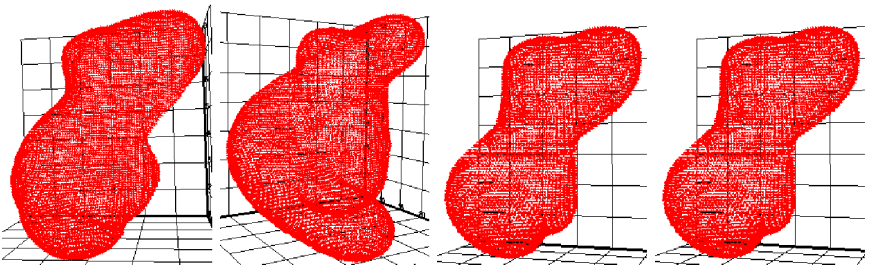
$$EAE_S = \frac{1}{g^2 \max_{x,y,z} \hat{f}_g(x,y,z)} \sum_{i,j=1}^{g} |\hat{s}_g(i,j) - s_{\bar{g}}(i,j)|, \tag{5}$$

Table 1 gives the numbers for EAE_S with $\bar{g} = 100$. The result shows that the error is very low. It is below 1% if the number of grid lines is greater than 30.

Figure 6 presents the the impact of the size of the database sample on the surface quality. The figures show that $n = 10'000$ is sufficient for a nice surface. Note that Figure 6(b) is shown from a different prospective. This perspective emphasizes the unevenness of the vertical line.

Table 1. The EAE_S Error for Different Number of Grid Lines

α	$g = 10$	$g = 30$	$g = 50$
1/10m	0.0289	0.0083	0.0045
5/10m	0.0249	0.0071	0.0038
9/10m	0.0069	0.0011	0.0005



(a) $n = 1'000$ (b) $n = 5'000$ (c) $n = 10'000$ (d) $n = 100'000$

Fig. 6. Cluster Surface for $\alpha = 1/10m$ and Varying Values of n

6.2 Space and Time Complexities

With the number of dimensions fixed at three the number of grid lines g and the number of observations n has the biggest impact on the computation time. We use the dataset from Figure 1(a) to measure the time to compute the surfaces.

Table 2 shows the times in seconds to calculate the surfaces from the raw data. A detailed analysis of the runtime reveals that the vast amount of the time is spent for the PDF estimation. Less than 1 second is needed to calculate a surface. Thus, to improve the performance it is possible to pre-compute and store PDFs. Table 3 shows that the size of the PDF is small and not usually relevant when compared to the size of the original database.

Table 2. Computation Time for Different Number of Grid Lines and Data Points

n	$g = 10$	$g = 30$	$g = 50$
1'000	<1	2	9
5'000	<1	6	24
10'000	<1	8	34
100'000	3	37	130
1'000'000	24	164	547

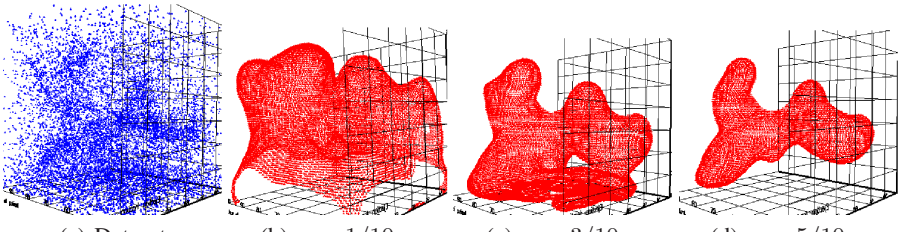
Table 3. Size of PDF in KB

g	10	30	50	100
Size	4	108	500	4'000

7 Experiments

This section illustrates our methods on an artificial data set (cf. Figure 7(a)). We show nested surface grids and offer an interpretation. Note that the visual information in the printed images is somewhat limited as three dimensions have to be projected into two. Also nested surfaces have to be shown in figures side-by-side. The reader may download and install the 3DVDM system to experiment with the surfaces.

The data set contains three data structures: 1) points, which are spread around randomly generated polygonal line, 2) a structure defined in terms of a simulated random variable: (uniform(0,1), uniform(0,1), exp(1)), and 3) uniform noise in the data set.



(a) Dataset (b) $\alpha = 1/10m$ (c) $\alpha = 3/10m$ (d) $\alpha = 5/10m$

Fig. 7. Artificial Dataset

It is hard to understand the structure from the scatter plot (cf. Figure 7(a)). The nested surfaces in Figures 7(b) to 7(d) emphasize and clarify the structure.

8 Summary and Future Work

In this chapter we defined and evaluated nested surfaces for the purpose of visual data mining. Since humans perceive surfaces much easier than individual observations, our approach to data mining gives the ability to investigate the structure easily. In addition, surfaces clarify very dense and sparse (and the combination of both) regions of the data set. That gives an ability to detect arbitrary shaped structures in a data set.

The surface calculation is based on an estimated PDF which makes our method independent of the data. The PDF estimation is implemented as a three-dimensional cube. We presented empirical results that show that the space and time complexity is reasonable. It is possible to compute surfaces on the fly during data explorations. Real time interaction can be achieved by precomputing and storing small density estimates.

In the future we will refine our methods to find curves and 2-D structures in a data set. It would also be interesting to experiment with the display of visually advanced surfaces that use transparency, light and shading. Finally, we also want to develop a new data structure that enables interactive computation of the density surfaces.

Acknowledgments

This work is supported in part by the Danish research council through grant 5051-01-004. We greatly appreciate the comments of the 3DVDM project members and

our partners from Nykredit. We thank the VRCN for the opportunity to work with immersive visualizations.

References

1. Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Chapman & Hall, London (1986)
2. Keim, D.A., Kriegel, H.-P.: Visualization Techniques for Mining Large Databases: A Comparison. Transactions on Knowledge and Data Engineering, Special Issue on Data Mining 8(6), 923–938 (1996)
3. Scot, D.W.: Multivariate Density Estimation. Wiley & Sons, New York (1992)
4. Wegman, E.J., Luo, Q.: Visualizing Densities. Technical Report Report No. 100, Center for Computational Statistics, George Mason University (1994)
5. van den Eijkel, G.C., Van der Lubbe, J.C.A., Backer, E.: A Modulated Parzen-Windows Approach for Probability Density Estimation. IDA (1997)
6. Bredon, G.E.: Topology and Geometry. Springer, Heidelberg (1995)
7. Shen, H., Johnson, C.: Sweeping Simplicies: A Fast Isosurface Extraction Algorithm for Unstructured Grids (1995)
8. Wilhelms, J., Van Gelder, A.: Octrees for Faster Isosurface Generation. ACM Transactions on Graphics 11(3), 201–227 (1992)
9. Devroy, L., Gyorfi, L.: Nonparametric Density Estimation. Jhon Wiley & Sons, Chichester (1984)
10. Farmen, M., Marron, J.S.: An Assesment of Finite Sample Performace of Adaptive Methods inDensity Estimation. Computational Statistics and Data Analysis (1998)
11. Jones, M.C., Wand, M.P.: Kernel Smoothing. Chapman & Hall, London (1985)
12. Lorensen, W., Cline, H.: Marchine cubes: A high resolution 3d surface construction algorithm (1987)

Visual Mining of Association Rules

Dario Bruzzese[1] and Cristina Davino[2]

[1] Dipartimento di Scienze Mediche Preventive, Università di Napoli Federico II
Via S. Pansini, 5 - 80131 Napoli, Italy
dbruzzes@unina.it
[2] Dipartimento di Studi sullo Sviluppo Economico, Università di Macerata
Piazza Oberdan, 3 - 62100 Macerata, Italy
cdavino@unimc.it

Abstract. Association Rules are one of the most widespread data mining tools because they can be easily mined, even from very huge database, and they provide valuable information for many application fields such as marketing, credit scoring, business, etc. The counterpart is that a massive effort is required (due to the large number of rules usually mined) in order to make actionable the retained knowledge. In this framework vizualization tools become essential to have a deep insight into the association structures and interactive features have to be exploited for highlighting the most relevant and meaningful rules.

1 Introduction

A visual data mining approach should complement the data mining techniques because the data visualization allows to understand the process and the models being used. The visualization step is essential when data mining is performed through Association Rules (AR) [2] because of the presence of too many associations where to detect the really relevant implications. It is a matter of fact that even if pruning methods allow to reduce the huge number of mined rules, the resulting subset is often too large for a textual inspection.

Many graphical tools have been proposed in literature (such as [5],[12], [19], [25],[31], [35], [37]) and some of them are implemented in data mining software systems (such as [1], [7], [15], [22], [24], [26], [27], [32], [36]). They make use of classical and basic representations strengthened by interactive features to easily explore the rules. In the following, the main association rules visualizations are discussed trying to highlight their characteristics, their limits and their explanatory features. Most of them represent rules through their characteristics measures (support and confidence) and through the list of involved items and they rarely let to compare items or rules. In this framework two approaches based on Factorial Methods [3] and on Parallel Coordinates [17] are investigated and enhanced.

The analysis of the AR visualizations will be illustrated through a real data set taken from UCI Machine Learning Repository [22].

S.J. Simoff et al. (Eds.): Visual Data Mining, LNCS 4404, pp. 103–122, 2008.

2 Some Issues about Association Rules

AR allow to find frequent patterns and associations in large databases characterized by the presence of a set of transactions, where each transaction is a subset of items. Many field of applications (marketing, credit scoring, business, etc.) require to resort to this data mining tool in order to solve typical problems such as the evaluation of the products assortment, the analysis and the prediction of purchase behaviour of the consumers.

Denoting with $I = i_1, i_2, \ldots, i_m$ a set of m items (e.g. all products bought by a group of customers) and with $T = t_1, t_2, \ldots, t_n$ a set of n transactions (e.g. all products in a customer's basket), an Association Rule R can be expressed in the form $A \rightarrow C$, where both A and C are subset of I such that $A \cap C = \emptyset$. The subset A is the set of antecedent items, also named left hand side (LHS) or body of the rule while C is the set of consequent items, also named right hand side (RHS) or head of the rule (in the following we will denote generic itemsets with capital letters and single items with small letters). The one-to-one rules where both the subset A and the subset C contain only one item $(x \rightarrow y)$ is the simplest association that can be mined but more complex associations can be extracted: many-to-one $(x, ..., y \rightarrow z)$, one-to-many $(x \rightarrow y, ..., z)$ and many-to-many $(x, ..., y \rightarrow z, ..., w)$.

Each rule R is characterized by two measures: the support and the confidence.

The *support* of R can be defined as:

$$S_R = \frac{n_R}{n} \tag{1}$$

where n_R is the number of transactions in T holding $A \cup C$ and it measures the proportion of transactions in T containing both A and C independently from the possible dependence of C from A. In a probabilistic approach the Support is an estimate of the probability of observing in a transaction the items belonging to both the antecedent and the consequence of the rule. The Support can be also referred to a generic itemset if the proportion of transactions sharing the itemset is considered.

The *confidence* of R can be defined as:

$$C_R = \frac{n_R}{n_A} \tag{2}$$

where n_A is the number of transactions in T holding the itemset A. The confidence measures the strength of the implication described by the rule (it is an estimate of the conditional probability of the consequence given the antecedent).

Nowadays, the considerable advances in the computational field allow to analyze many transactions in real time and to easily discover a number of rules that often exceeds the number of transactions. The main drawback of Association Rules is thus the huge number of extracted rules that cannot be manually inspected by the user and the existence of trivial or meaningless associations that are usually mined due to the exhaustive nature of the extraction algorithms. Graphical tools and pruning methods are the main approaches used to face these problems.

Many software tools for the visualization of Association Rules have been proposed in literature. They are limited to visualize only one-to-one rules or many-to-one rules. However the number of displayed rules is so huge that many of them overlap.

AR miners have been sensible to this problem since the introduction of AR in the data mining framework as the abundant literature on pruning methods shows([18], [30], [34]). The first approaches to face the problem of the huge number of discovered AR and of their relevance for the user were based on interestingness measures both subjective and objective. While the former require user domain knowledge and they obviously depend on the user who examines the patterns, the latter force the user to fix a suitable threshold for them. For instance, minimum support and minimum confidence values are usually fixed by the user before mining association rules. Unproper choices of these values may cause many drawbacks: if they are set very low a huge number of rules (some of which being meaningless) will be found. On the contrary, if they are set very high, trivial rules will be found [34]. Moreover, using only confidence and support based thresholds doesn't allow to take into account the strength and the statistical significance of the associations. In this framework, automatic procedure based on statistical tests have been proposed ([6], [14], [21], [23], [33]) even if not all the necessary assumptions are satisfied. The issue is to exploit the theoretical reference framework of the hypothesis tests in order to derive practical criteria, namely score functions [8], to prefer some rules to others. In order to overcome the problems of some of the Association Rules graphical tools, it can be advisable to apply them on the pruned subset of rules.

3 A Real Data Set Application

The data used to discuss the different approaches to Association Rules visualization is taken from UCI Machine Learning Repository [22]. It deals with 101 animals described by 15 boolean attributes (*hair*, *feathers*, *eggs*, *milk*, *airborne*, *aquatic*, *predator*, *toothed*, *backbone*, *breathes*, *venomous*, *fins*, *tail*, *domestic*, *catsize*) and a numeric one (*legs*) which has been categorized in four boolean attributes (0, 2, 4, >4). The zoo data set is a typical machine learning set but it can be useful to highlight the main drawbacks of Association Rules because there is a strong relationship among the animals attributes. Moreover the topic is not a technical one and the rules can be easily understood. The data can be associated to a set of 101 transactions where each transaction is a sub-set of features belonging to an animal.

The number of rules discovered applying the mining process to the set of 101 animals is 3728. This huge number is obtained considering only many-to-one rules at most of the fifth order[1] and fixing a minimum support equal to 0.05

[1] The generated rules contain at most four items in the antecedent and one item in the consequence.

and a minimum confidence equal to 0.5. On the set of mined rules, a sequence of three statistical tests [2] are performed in order to evaluate if each rule significantly satisfies the user specified minimum confidence [21] and support [23] thresholds and if the presence of the itemset in the consequence of each rule depends on the presence on the itemset in the antecedent [6]. At each step, the rules are ranked according to the corresponding test statistics and a subset of rules is obtained pruning the rules out of a suitable threshold. This subset becomes the rules input set for the next step.

The main results of the pruning phase (Table 1) show that after the first step, 18% of the original rules were pruned without influencing significantly the support and confidence ranges. The second step allows to prune a lot of rules (33% of the rules survived at the second step) with not significant support values. Using also the third step as a pruning tool, 1447 final rules remain where the minimum confidence is equal to 0.6 and the minimum support is equal to 0.1. It is worth to notice that the subset of final rules is still very big for a manual inspection and it requires visualization tools to be analysed.

Table 1. Information about the pruning process

	Before Pruning	After Step 1	After Step 2	After Step 3
Nr. of rules	3728	3066	2049	1447
Percentual variation in the nr. of rules		-18%	-33%	-29%
Nr. of involved antecedent items	19	18	18	16
Nr. of involved consequent items	16	16	16	16
Nr. of 2^{nd} order rules	117	83	82	50
Nr. of 3^{rd} order rules	602	471	374	252
Nr. of 4^{th} order rules	1363	1118	758	539
Nr. of 5^{th} order rules	1646	1394	835	606
Minimum Confidence	0.5	0.6	0.6	0.6
Maximum Confidence	1.00	1.00	1.00	1.00
Minimum Support	0.05	0.06	0.1	0.1
Maximum Support	0.73	0.73	0.73	0.73

4 Visualizing Association Rules

Many visualization tools have been introduced in literature and/or implemented in data mining software systems. They differ with respect to the type of represented rules (one-to-one, many-to-one, etc.), to the number of associations that can be visualized, to the type of visualized information (items or measures characterizing the rules), to the number of dimensions (2-D or 3-D) and to the possibility to interact with the graph.

[2] p-values less than 0.01 were considered significant.

4.1 Rule Table

The most immediate Association Rules visualization method is a table (Figure 1) where each row represents a rule and each rule is divided into various parts allocated in different columns of the table. The advantage of this approach is the ability to sort the results by the column of interest. Its main limitation is the close resemblance to the original row textual form so that the user can inspect only few rules without having a global view of all the information.

	A	B	C	D	E	F	G
1	Antecedent items				Consequence	Confidence	Support
2	Breathes	Toothed			Backbone	1.00	0.47
3	Backbone	Milk	Toothed		Breathes	1.00	0.40
4	Breathes	Milk	Toothed		Backbone	1.00	0.40
5	0 Legs	Backbone			Tail	0.95	0.18
6	Backbone	Hair	Milk		Breathes	1.00	0.39
7	Breathes	Hair	Milk		Backbone	1.00	0.39
8	Backbone	Breathes	Hair	Toothed	Milk	1.00	0.38
9	0 Legs	Catsize			Tail	0.86	0.06
10	0 Legs	Predator			Eggs	0.76	0.13
11	Eggs	Fins	Predator	Toothed	Tail	1.00	0.09
12	Predator	Tail	Toothed	Venomous	Eggs	0.67	0.02
13	Tail				Toothed	0.69	0.51
14	>4 Legs	Eggs			Breathes	0.67	0.08
15	>4 Legs	Hairborne			Hair	0.67	0.04
16	0 Legs	Aquatic			Backbone	0.94	0.17
17	2 Legs	Aquatic	Eggs		Hairborne	0.83	0.05
18	2 Legs	Aquatic	Tail		Eggs	0.86	0.06

Fig. 1. Rule Table

4.2 Two-Dimensional Matrix

The rules are displayed in a bar diagram where the consequent items are on one axis and the antecedent items on the other axis. The height and the color of the bars are used to represent support and confidence. This visualization approach can be used only in case of one-to-one rules. In figure 2 a subset (50) of the rules extracted on the zoo data set is displayed in a 2-D matrix.

The matrix of associations rules proposed by [14] represent a crushed version of the two dimensional matrix where colors are used to indicate the confidence level while the tone of the colors represents the support.

It is a matter of fact that second order rules are usually pruned (see Table 1) because they represent trivial information. Some softwares like Statistica [29] and Enterprise Miner [27] try without success to overcome this drawback by grouping the items belonging to the antecedent of a rule and by plotting the new unit against the consequence but this strategy is not successful especially when a huge number of rules containing many items in the antecedent is visualized.

4.3 3-D Visualization

The visualization technique proposed by Wong et al. [35] tries to solve the 2-D visualization problems by visualizing many-to-one relationships. The rows of a matrix floor represent the items and the columns represent the rules. Bars with different heights are used to distinguish the consequence and the antecedent of

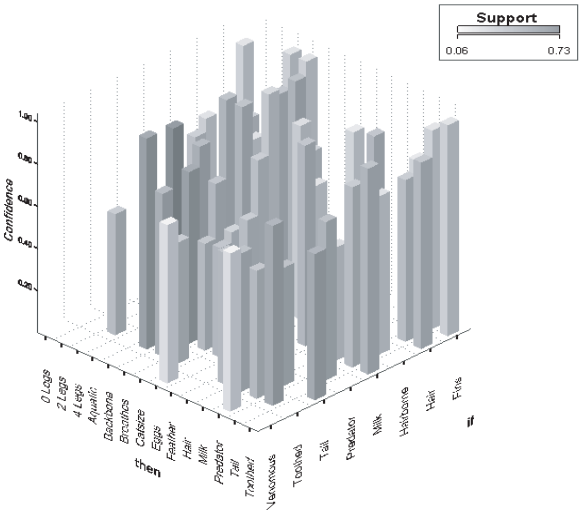


Fig. 2. 2-D matrix representation

each rule. At the far end of the matrix, bars proportional to the confidence and the support measures are represented. The 3-D visualization doesn't impose any limit on the number of items in the antecedent and in the consequence. It allows to analyse the distribution of the association rules and of each item. The 3D view is clear because the support and confidence values are shown at the end of the matrix and in general there is no need for animation.

The visualization proposed by Wong et al. improved 2-D matrix but it still had some problems: the antecedent and consequent items could overlap because they have different positions on the y-axis and the number of displayed rules is limited by the width of matrix floor. In figure 3, 50 rules of different order are plotted using cones instead of bars to partially avoid items overlapping.

4.4 Association Rules Networks

In IBM Intelligent Miner [15] a network representation of AR is provided where each node represents an item and the edges represent the associations. Different colors and width of the arrows are used to represent the confidence and the support. When many rules with many items are represented, the direct graph is not easy to understand because of the superimposition of the edges with the nodes.

Figure 4 shows the visualization of the rules presented in Table 2. If another rule such as $\{$*0 Legs*, *Predator* $\rightarrow$ *Toothed*$\}$ is added to the graph representation, the overlapping among the edges would confuse too much the visualization.

In figure 5 a different network representation is shown [29]. The network displays a subset of 15 rules obtained by setting a maximum order of three, minimum support equal to 50% and minimum confidence equal to 70%. The support

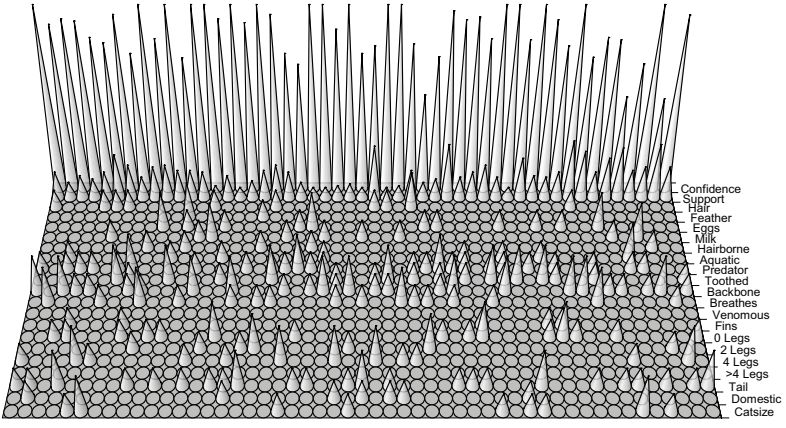


Fig. 3. 3-D matrix representation

Table 2. The rules displayed on the Direct Graph

Antecedent		Consequence	Conf.	Sup.
0 legs		Backbone	0.82	0.18
Aquatic		Backbone	0.80	0.29
0 Legs	Aquatic	Backbone	0.90	0.16
0 Legs		Toothed	0.82	0.18
Backbone		Toothed	0.73	0.6
Backbone		Predator	0.56	0.46

values for the antecedent and consequence of each association rule are indicated by the sizes and colours of each circle. The thickness of each line indicates the confidence value while the sizes and colours of the circles in the center, above the Implies label, indicate the support of each rule. Hence, in figure 5 the strongest support value was found for the one-to-one rules involving the items Tail and backbone. The visualization doesn't allow to identify the most interesting rules especially for the rules with an order greater than 2. The 3D version of the Association Rules Network adds a vertical z - axis to represent the confidence values but as the 2D version, it can be useful only in case of a very small set of rules.

4.5 The TwoKey Plot

The TwoKey plot [31] represents the rules according to their confidence and support values. In such a plot, each rule is a point in a 2-D space where the x-axis and the y-axis ranges respectively from the minimum to the maximum values of the supports and of the confidences and different colors are used to highlight the order of the rules. Many interactive features can facilitate the exploration of the rules such as the selection of a region of the plane where confidence and

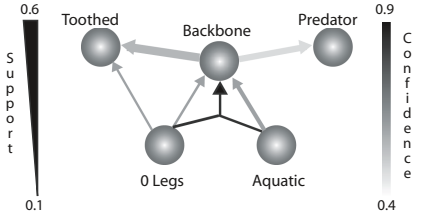


Fig. 4. Direct graph representation

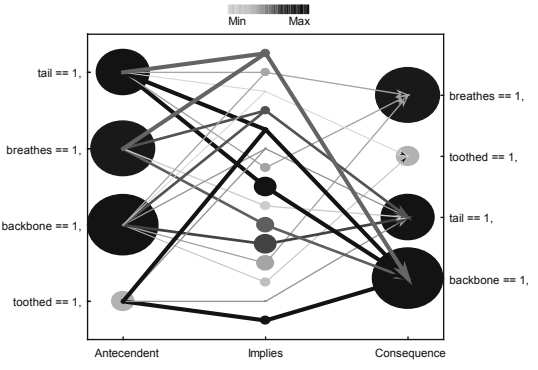


Fig. 5. Association rules network

support are above a user defined threshold or the linking with children, parents and neighbours of a rule. The TwoKey plot can be linked with other displays (barchart of the level, barcharts of the items belonging to sets of rules, mosaic plots [11]).

In Figure 6 a TwoKey plot of one thousand rules extracted from the zoo dataset is shown; an immediate and global overview of the displayed set of rules is provided and it is easy to identify privileged subsets of rules lying in particular regions (for example high confidence rules lining up the top of the graph). The analysis of the items present in the displayed rules necessarily requires to have recourse to the rule table representation which suffers the previously mentioned problems or to a different visualization involving the items.

4.6 Double-Decker Plot

Mosaic plots ([10], [11]) and their variant called Double-Decker ([12], [13], [14]) plots provide a visualization for single association rules but also for all its the related rules. They were introduced to visualize each element of a multivariate contingency table as a tile (or bin) in the plot and they have been adapted to visualize all the attributes involved in a rule by drawing a bar chart for the consequence item and using linking highlighting for the antecedent items. In Figure 7 the double decker plot of the rule *Predator* & *Venomous* & *4 legs* →

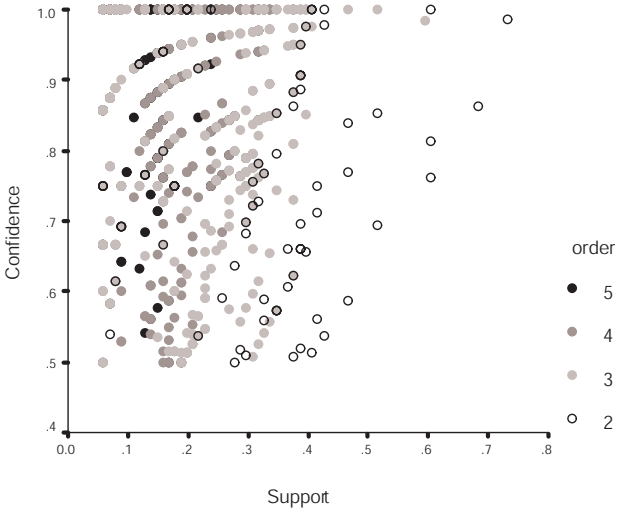


Fig. 6. The TwoKey Plot

Toothed is shown. Each row of the plot corresponds to one item, each gray shade represents one value of this item, the support is the area of highlighting in a bin, the confidence is the proportion of highlighted area in a bin with respect to the total area of the bin. The main drawback of Double Decker plot lies in the possibility to represent one rule at a time or at least all the rules generated from the different combinations of the items belonging to a given rule. In order to have the possibility to represent simultaneously many rules, Hofmann and Wilhelm ([14]) proposed the matrix of Association Rules with and without additional highlighting but only one-to-one rules are taken into consideration.

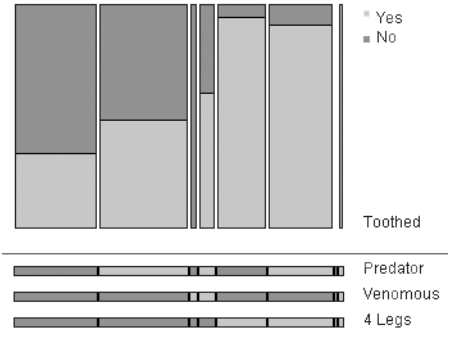


Fig. 7. Double-Decker Plot

4.7 Parallel Coordinates

Parallel coordinates, introduced by Inselberg in 1981 [16], represent a very useful graphical tool to visualize high dimensional data-sets in a two-dimensional space. They appear as a set of vertical axes where each axis describes a dimension of the domain and each case is represented by a line joining its values on the parallel axes.

Parallel coordinates have been used to visualize AR by several authors ([5], [19][3], [37]). The approach proposed by Yang starts from arranging items by groups on a number of parallel axes equal to the maximum order of the rules. A rule is represented as a polyline joining the items in the antecedent followed by an arrow connecting another polyline for the items in the consequence. The items arrangement on each axis should ensure that polylines of itemsets of different groups never intersect with each other. It is a matter of fact that such representation becomes infeasible in case of hundreds or even tens of items and it is not coherent with the original framework of parallel coordinates dealing with quantitative variables.

In figure 8 a parallel coordinate plot of 50 rules of different order is shown. It is evident that in such a case it is not possible to identify disjoint groups of items so that there is an overlapping of the polylines.

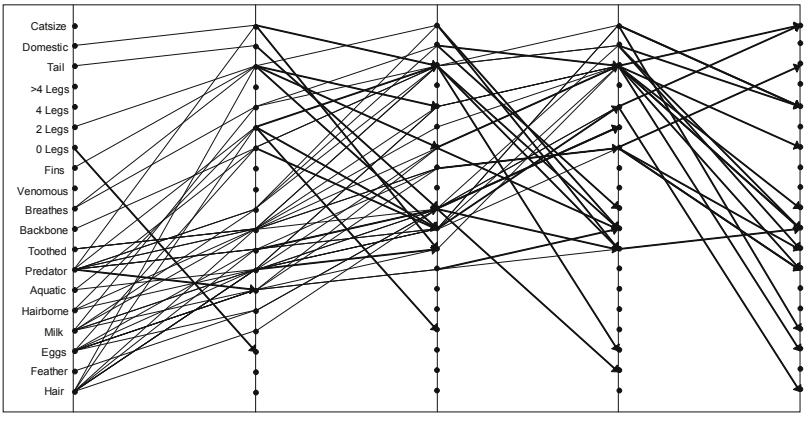


Fig. 8. The parallel coordinate plot proposed by Yang

In the visualization proposed by Bruzzese et al. [5] each antecedent item is a dimension of the graph and it spans according to the utility provided to each rule. The utility of an item i in the antecedent of a rule R is measured by an index called *Item Utility* (IU) based on the comparison between the confidence of the rule with or without item i. Considering the rule $x, y \rightarrow z$ the Item Utility of the y item was defined as follows:

[3] The representation proposed by Kopanakis et al. is not described because it is limited to quantitative AR.

$$IU_y = \frac{C_{x,y\to z} - C_{x\to z}}{max(C_{x\to z}; C_{x,y\to z})} \tag{3}$$

It is a matter of fact that the transactions holding both the x and y items, still contain some transactions sharing the y item. In order to manage the spurious presence of the y item in the rule $x \to z$, it is more appropriate to compare the confidence of the rule $R_1 = x, y \to z$ with the confidence of the rule $R_2 = x, \neg y \to z$ where $\neg y$ denotes the absence of the y item in a transaction. An enhanced *item utility*, called *NIU*, is proposed as follows:

$$NIU_i = \frac{C_R - C_{R(\neg i)}}{max(C_R; C_{R(\neg i)})} \tag{4}$$

where i is a generic antecedent item of the rule R and $R_(\neg i)$ represents the rule R free of the i item.

It results that the *NIU* stresses the importance or the uselessness of an antecedent item with respect to the *IU* as is shown in figure 9 where a graphical comparison among the two indexes is given. Considering each square as a transaction, in figure 9a the confidence of the rule $x, y \to z$ is equal to 1; in 9b the confidence of the rule $x \to z$ is equal to $\frac{2}{3}$ showing that the y item is useful as the confidence decreases when it is not considered. Taking into account the rule $x, \neg y \to z$ (figure 9c) which has a confidence equal to 0, the importance of the y item is highlighted because there are no transactions holding x and z without holding y too.

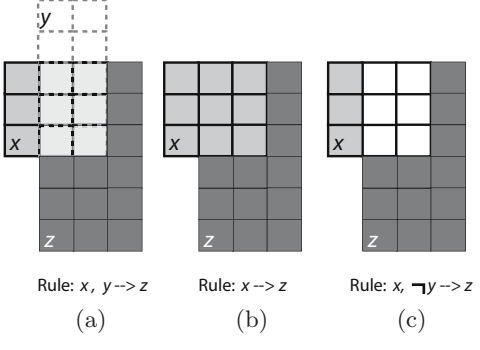


Fig. 9. A graphical comparisong beetwen the *IU* and the *NIU* for the y item

The *NIU* ranges in the interval $]-1;1]$. The value -1 is not included in the interval as it refers to rules with confidence equal to 0 which can never be mined. If $NIU_i \in]-1;0[$, the i item is harmful and the rule can be pruned as the intersection between the i item and the other antecedent items is not relevant for the prediction of the consequence. If $NIU_i \in]0;1]$, the item is useful as its interaction with the other antecedent items improves the capability to explain the consequence. The case NIU=0 refers to the presence of a redundant item as its presence in a transaction doesn't add further information. From a

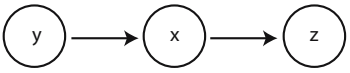


Fig. 10. The graph representation of the rule $x, y \rightarrow z$ when z and y are conditionally independent given x

probabilistic point of view the case $NIU = 0$ for the y item in the rule $x, y \rightarrow z$ means that z and y are conditionally independent given x and the rule can be represented as a directed acyclic graph (Figure 10) where the link between the item y and the item z goes through x (metaphorically x screens off z form y).

A view of the discovered rules can be obtained plotting on parallel coordinates the *NIU* of each item belonging to the antecedent of a rule.

Some of the interaction tools of parallel coordinates [17] are exploited in order to visualize, interpret and reduce the number of rules. In particular, data analysis can be facilitated by:

- selecting a subgroup of rules with one or more items below a specified *NIU* threshold in order to remove selected lines from the plot;
- identifying axes (items) with very dense positive values, given a consequence, in order to highlight items with a high explicative power;
- adding two supplementary dimensions corresponding to the support and confidence of the rules in order to remove those rules with values of these parameters below a specified threshold;
- selecting high confidence rules in order to identify sets of items involved in very strong associations;
- changing the order of the dimensions on the basis of *NIU* distributions.

In figure 11 a plot of 736 rules with a common consequence (*Toothed*) is shown. Each rule is represented as a line joining the axis corresponding to its antecedent items, to its confidence and support values. The most explicative items (*0 Legs*, *4 Legs*, *Backbone*, *Fins*, *Milk*) and the most critical items (*2 Legs*, *Eggs*) can be easily identified respectively as the ones with very dense positive or negative *NIU* values.

The empirical evaluation of the item utility must be accomplished with the assessment of its statistical significance in order to obtain an overall measure of the importance of each item in a rule. At this aim a statistical test is introduced to verify whether the difference between the two confidences is equal or greater than 0. Let C_{R_1} be the confidence of the rule $R_1 = x, y \rightarrow z$ and C_{R_2} be the confidence of the rule $R_2 = x, \neg y \rightarrow z$. The test is performed starting from the following hypothesis:

$$H_0 : C_{R_1} = C_{R_2} \qquad H_0 : C_{R_1} > C_{R_2} \tag{5}$$

Under the null hypothesis the test statistics T_{NIU}:

$$T_{NIU} = \frac{C_{R_1} - C_{R_2}}{\sqrt{C^*(1 - C^*)\left(\frac{1}{n_{x,y}} + \frac{1}{n_{x,\neg y}}\right)}} \tag{6}$$

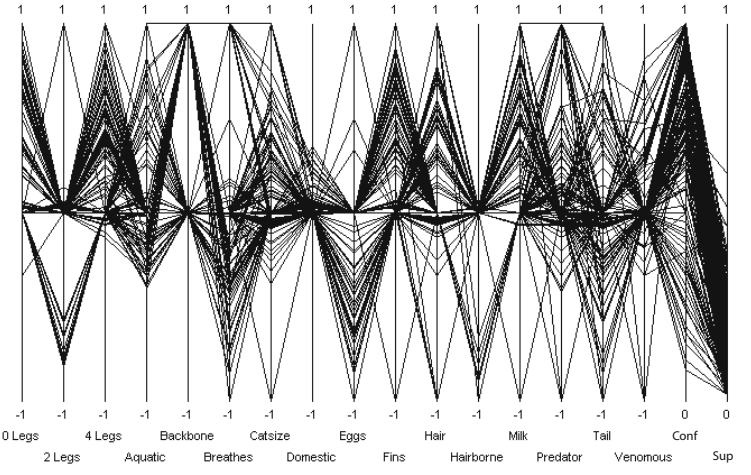


Fig. 11. The parallel coordinates plot of rules with consequence equal to *Toothed*

approximates a standard normal distribution given that $n_{x,y}$ and $n_{x,\neg y}$ are sufficiently large. The term C^* refers to the estimate of the conjoint proportion:

$$C^* = \frac{n_{x,y,z} + n_{x,\neg y,z}}{n_{x,y} + n_{x,\neg y}} \tag{7}$$

From equation 7 it follows that $C*$ measures the confidence of the rule $R^* = x \rightarrow z$.

When we deal with one-to-one rules, the test statistics T_{NIU} given in equation 6 is equal to the *Difference of Confidence (Doc)* test statistic proposed in [14] where the confidence of a rule is compared with the confidence of the rule obtained considering the same consequence but the negation of the whole antecedent set of items. The test can be used to prune those rules where at least one antecedent item has a *NIU* not significantly greater than 0 because the interaction among all the antecedent items is not relevant and a lower order rule must be retained.

Figure 12 shows a parallel plot of the 60 rules that survived the test with a significance level of 0.05. The set of rules is characterised by high confidence values and by a strong interaction among the shared items, with *NIU* values often equal to 1.

4.8 Factorial Planes

As a matter of fact, the number of extracted rules, and even the number of rules after pruning, are huge, which makes manual inspection difficult. A factorial method can be used to face this problem because it allows to synthesize the information stored in the rules and to visualize the associations structure on 2-dimensional graphs.

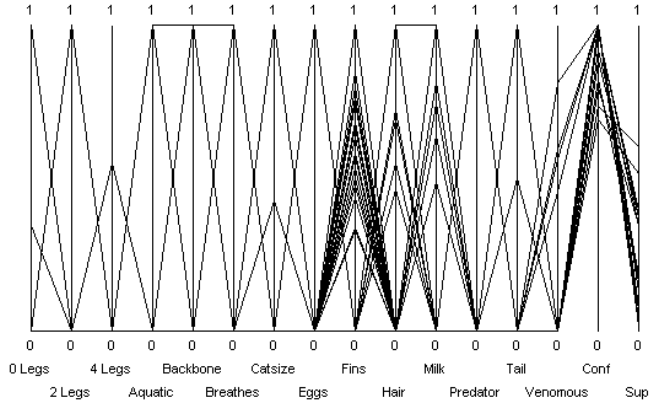


Fig. 12. The parallel coordinate plot of rules with consequence equal to *Toothed* after pruning

The rules being synthesized are stored in a data matrix where the number of n rows is equal to the number of rules and the number of columns ($p = p_{if} + p_{then}$) corresponds to the total number of different items, both in the antecedent part (p_{if}) and in the consequent part (p_{then}) of the n rules. Each rule is coded by a binary array assuming value 1 if the corresponding column item is present in the rule and value 0 otherwise. The well known confidence and support measures are also considered as added columns. The final data matrix has thus $n \times (p_{if} + p_{then} + 2)$ dimensions and it can be analysed through the Multiple Correspondence Analysis (MCA) ([3], [9]) that allows to represent the relationships among the observed variables, the similarities and differences among the rules and the interactions between them. MCA allows to reduce the number of original variables finding linear combinations of them, the so called *factors*, that minimize the deriving loss of information due to the dimensionality reduction. Different roles are assigned to the columns of the data matrix: the antecedent items are called *active* variables and they intervene directly in the analysis defining the factors; the consequent items and the support and the confidence values are called *supplementary* variables because they depend from the former and are projected later on the defined factorial planes.

Referring at the zoo data set, the rules survived to a pruning process [6] are 1447 and they involve 16 different items[4] both in the antecedent part (p_{if}) and in the consequence (p_{then}). The set of rules should thus be represented in a 16-dimensional space and the set of items in a 1447-dimensional space. In order to reduce the number of original variables through the factors, it is necessary to evaluate the loss of information deriving or the variability explained by the retained factors. According to the Benzcri approach [3] for the evaluation of the explained variability in case of MCA, in table 3, the explained variability and the cumulative variability is shown. The first two factors share more than the

[4] *Venomous*, *Domestic* and *>4 Legs* are the items removed by the pruning procedure.

Table 3. Total inertia decomposition

Factor	% of variability	Cumulative %	
1	44	44	**
2	40	84	**********************************
3	12	96	**********
4	4	100	****

80% of the total inertia and they correspond to the highest change in level in the percentage of variability.

Once the MCA is performed it is possible to represent the rules and the items on reduced dimensions subspaces: the factorial planes allowing to explain at least a user defined threshold of the total variability (in the zoo example, the first factorial plane) or a user defined factorial plane or the factorial plane best defined by a user chosen item.

Different *views* on the set of rules can be obtained exploiting the results of the MCA.

1. **Items Visualization.** A graphical representation of the antecedent and the consequent items is provided by the factorial plane where the item points have a dimension proportional to their supports and the confidence and the support are represented by oriented segments linking the origin of the axes to their projection on the plane. In Figure 13 the active and the supplementary items are visualized together with the confidence and support arrows.

   Privileged regions characterized by strong rules can be identified in case of high coordinates of the confidence and the support because their coordinates represent the correlation coefficients with the axes.

   The proximity between two antecedent items shows the presence of a set of rules sharing them while the proximity between two consequent items is related to a common causal structure. Finally, the closeness between antecedent items and consequent items highlights the presence of a set of rules with a common dependence structure.
2. **Rules Visualization.** Another view on the mined knowledge is provided by the rules representation on the factorial plane. Graphical tools and interactive features can help in the interpretation of the graph: the rules are represented by points with a dimension proportional to their confidence, the proximity among two or more rules shows the presence of a common structure of antecedent items associated to different consequences, a selected subset of rules can be inspected in a tabular format. For example in table 4 the subset of the rules selected in figure 14 is listed.

   It is worth of notice that this subset of rules is very close on the plane because they have similar antecedent structures sharing at least one item, even some rules overlap because they have exactly the same antecedent structure.

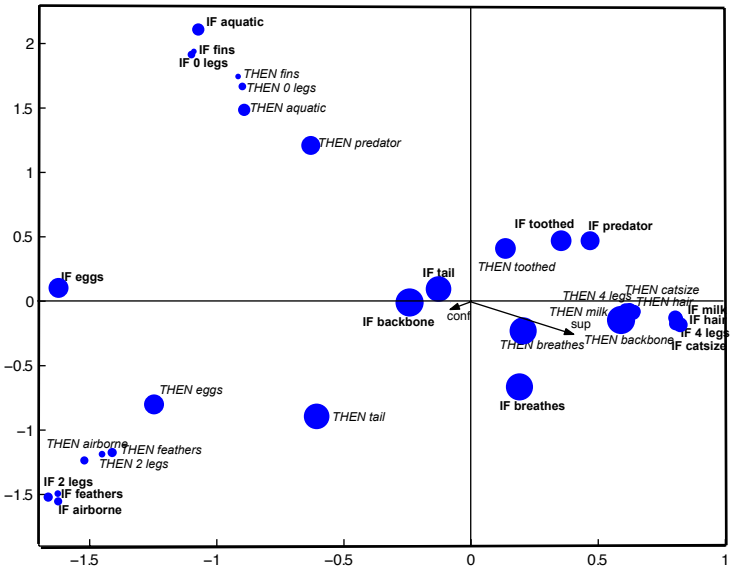


Fig. 13. The items representation

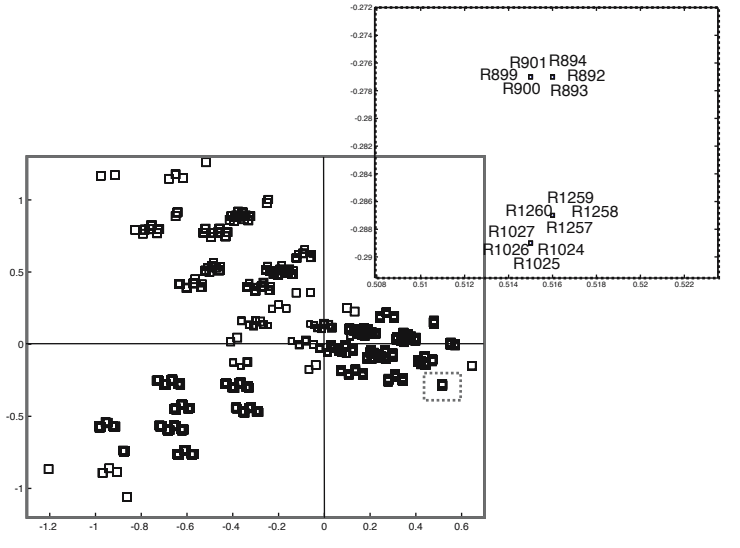


Fig. 14. The rules representation

It is possible to imagine to transform the set of overlapping rules into a higher order macro-rule obtained linking the common behaviour described by the antecedent items to the logical disjunction of the different consequent items.

Table 4. Description of a subset of overlapping rules

Rule	Antecedent	Consequence	Conf.	Sup.
899	Hair & Milk & Breathes & Catsize	Toothed	0.97	0.29
900	Hair & Milk & Breathes & Catsize	Backbone	1.00	0.30
901	Hair & Milk & Breathes & Catsize	4 legs	0.83	0.25
892	Hair & Milk & Breathes & 4 legs	Toothed	0.97	0.30
893	Hair & Milk & Breathes & 4 legs	Backbone	1.00	0.31
894	Hair & Milk & Breathes & 4 legs	Tail	0.90	0.28
1257	Milk & Breathes & 4 legs & Catsize	Hair	1.00	0.25
1258	Milk & Breathes & 4 legs & Catsize	Toothed	0.96	0.24
1259	Milk & Breathes & 4 legs & Catsize	Backbone	1.00	0.25
1260	Milk & Breathes & 4 legs & Catsize	Tail	0.92	0.23
1024	Hair & Breathes & 4 legs & Catsize	Milk	1.00	0.25
1025	Hair & Breathes & 4 legs & Catsize	Toothed	0.96	0.24
1026	Hair & Breathes & 4 legs & Catsize	Backbone	1.00	0.25
1027	Hair & Breathes & 4 legs & Catsize	Tail	0.92	0.23

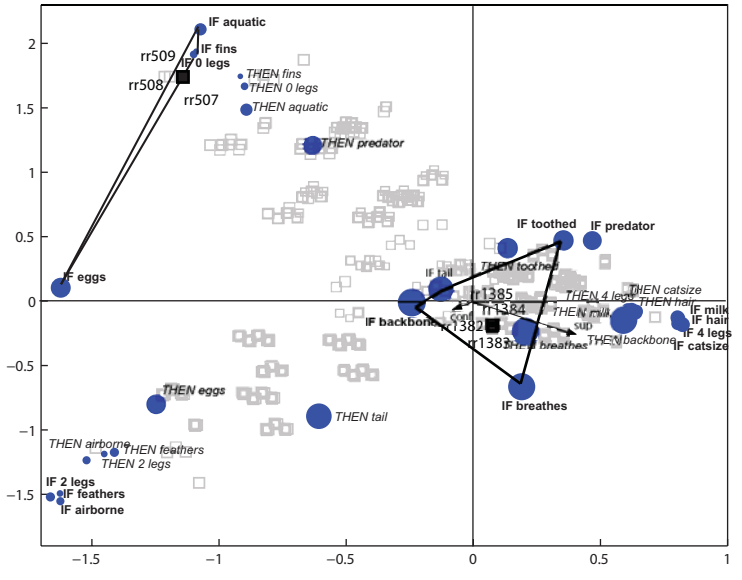


Fig. 15. The Conjoint representation

3. **Conjoint Visualization.** The factorial planes features allow to visualize simultaneously the items and the rules. In the conjoint representation, aside from a scale factor, each rule is surrounded by the antecedent items it holds and vice versa each item is surrounded by the rules sharing it. By linking two or more active items it is possible to highlight all the rules that contain at least one of the selected items in the antecent. For example in figure 15

two groups of rules have been closed inside the polygons joining the items they share in the antecedent.

5 Concluding Remarks

Association Rules Visualization is emerging as a crucial step in a data mining process in order to profitably use the extracted knowledge. In this paper the main approaches used to face this problem have been discussed. It rises that, up to day, a compromise have to be done between the quantity of information (in terms of number of rules) that could be visualized and the depth of insight that can be reached. This suggests that there is not a winning visualization but their strength lies in the possibility to exploit the synergic power deriving from their conjoint use. Moreover, it is advisable a stronger interaction among the visualization tools and the data mining process that should incorporate each other.

Acknowledgements. The paper was financially supported by University of Macerata grant (2004) *Metodi Statistici Multivariati per l'Analisi e la Valutazione delle Performance in Campo Economico e Aziendale.*

References

1. Advanced Visual Systems (AVS), OpenViz. `http://www.avs.com/software/`
2. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in Large Databases. In: Proceedings of the 1993 ACM SIGMOD Conference, Washington DC, USA, pp. 207–216 (May 1993)
3. Benzècri, J.-P.: L'Analyse des Donnèes, Dunod, Paris (1973)
4. Bruzzese, D., Buono, P.: Combining Visual Techniques for Association Rules Exploration. In: Proceedings of the International Conference Advances Visual Interfaces, Gallipoli, Italy, May 25-28 (2004)
5. Bruzzese, D., Davino, C., Vistocco, D.: Parallel Coordinates for Interactive Exploration of Association Rules. In: Proceedings of the 10th International Conference on Human - Computer Interaction, Creta, Greece, June 22-27. Lawrence Erlbaum, Mahwah (2003)
6. Bruzzese, D., Davino, C.: Significant Knowledge Extraction from Association Rules. In: Electronic Proceedings of the International Conference Knowledge Extraction and Modeling Workshop, Anacapri, Italy, September 4-6 (2006)
7. Clementine, Suite from SPSS, `http://www.spss.com/Clementine/`
8. Glymour, C., Madigan, D., Pregibon, D., Smyth, P.: Statistical Inference and Data Mining. Communications of the ACM (1996)
9. Greenacre, M.: Correspondence Analysis in Practice. Academic Press, London (1993)
10. Hartigan, J., Kleiner, B.: Mosaics for contingency tables. In: Proceedings of the 13th Symposium on the interface, pp. 268–273 (1981)
11. Hofmann, H.: Exploring categorical data: interactive mosaic plots. Metrika 51(1), 11–26 (2000)

12. Hofmann, H., Siebes, A., Wilhelm, A.: Visualizing Association Rules with Interactive Mosaic Plots. In: Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 227–235 (2000)
13. Hofmann, H., Wilhelm, A.: Validation of Association Rules by Interactive Mosaic Plots. In: Bethlehem, J.G., van der Heijden, P.G.M. (eds.) Compstat 2000 - Proceedings in Computational Statistics, pp. 499–504. Physica-Verlag, Heidelberg (2000)
14. Hofmann, H., Wilhelm, A.: Visual Comparison of Association Rules. Computational Statistics 16, 399–416 (2001)
15. IBM Intelligent Miner for Data, http://www.software.ibm.com/data/intelli-mine
16. Inselberg, A.: N-dimensional Graphics, part I - Lines and Hyperplanes, in IBM LASC Tech. Rep. G320-2711, 140 pages. IBM LA Scientific Center (1981)
17. Inselberg, A.: Visual Data Mining with Parallel Coordinates. Computational Statistics 13(1), 47–64 (1998)
18. Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., Verkamo, A.I.: Finding interesting rules from large sets of discovered association rules. In: Proceedings of the Third International Conference on Information and Knowledge Management CIKM 1994, pp. 401–407 (1994)
19. Kopanakis, I., Theodoulidis, B.: Visual Data Mining & Modeling Techniques. In: 4th International Conference on Knowledge Discovery and Data Mining (2001)
20. Liu, B., Hsu, W., Ma, Y.: Pruning and Summarizing the Discovered Associations. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 1999), San Diego, CA, USA, August 15-18 (1999)
21. Liu, B., Ma, Y., Lee, R.: Analyzing the Interestingness of Association Rules from Temporal Dimensions. In: International Conference on Data Mining, CA (2001)
22. Machine learning, http://www1.ics.uci.edu/~mlearn/MLSummary.html
23. Megiddo, N., Srikant, R.: Discovering Predictive Association Rules. In: Knowledge Discovery and Data Mining (KDD 1998), pp. 274–278 (1998)
24. Miner3D, Miner3D Excel, http://www.miner3d.com/m3Dxl/
25. Ong, K.-H., Ong, K.-L., Ng, W.-K., Lim, E.-P.: CrystalClear: active Visualization of Association Rules. In: Proc. of the Int. workshop on Active Mining, Japan (2002)
26. Purple Insight Mineset, http://www.purpleinsight.com
27. Sas Enterprise Miner, http://www.sas.com/technologies/analytics/datamining/miner
28. Shah, D., Lakshmanan, L.V.S., Ramamritham, K., Sudarshan, S.: Interestingness and Pruning of Mined Patterns. In: Workshop Notes of the 1999 ACM SIGMOD Research Issues in Data Mining and Knowledge Discovery (1999)
29. Statistica, http://www.statsoft.com
30. Toivonen, H., Klemettinen, M., Ronkainen, P., Hatonen, K., Mannila, H.: Pruning and grouping of discovered association rules. In: Workshop Notes of the ECML-95 Workshop on Statistics, Machine Learning, and Knowledge Discovery in Databases, Heraklion, Greece, April 1995, pp. 47–52 (1995)
31. Unwin, A., Hofmann, H., Bernt, K.: The TwoKey Plot for Multiple Association Rules Control. In: Siebes, A., De Raedt, L. (eds.) PKDD 2001. LNCS (LNAI), vol. 2168. Springer, Heidelberg (2001)
32. VisualMine, http://www.visualmine.com/
33. Webb, G.I.: Preliminary Investigations into Statistically Valid Exploratory Rule Discovery. In: Proceedings of the Australasian Data Mining Workshop, Sydney (2003)

34. Weber, I.: On Pruning Strategies for Discovery of Generalized and Quantitative Association Rules. In: Proceedings of Knowledge Discovery and Data Mining Workshop, Singapore (1998)
35. Wong, P.C., Whitney, P., Thomas, J.: Visualizing Association Rules for Text Mining. In: Wills, G., Keim, D. (eds.) Proceedings of IEEE Information Visualization 1999. IEEE CS Press, Los Alamitos (1999)
36. XGvis: A System for Multidimensional Scaling and Graph Layout in any Dimension, `http://www.research.att.com/areas/stat/xgobi/`
37. Yang, L.: Visualizing Frequent Itemsets, Association Rules and Sequential Patterns in Parallel Coordinates. In: Kumar, V., Gavrilova, M.L., Tan, C.J.K., L'Ecuyer, P. (eds.) ICCSA 2003. LNCS, vol. 2667, pp. 21–30. Springer, Heidelberg (2003)

Interactive Decision Tree Construction for Interval and Taxonomical Data

François Poulet[1] and Thanh-Nghi Do[2]

[1] IRISA-Texmex
Université de Rennes I
Campus Universitaire de Beaulieu
35042 Rennes Cedex, France
francois.poulet@irisa.fr
[2] Equipe InSitu
INRIA Futurs, LRI, Bat.490
Université Paris Sud
91405 Orsay Cedex, France
thanh-nghi.do@lri.fr

Abstract. Visual data-mining strategy lies in tightly coupling the visualizations and analytical processes into one data-mining tool that takes advantage of the assets from multiple sources. This paper presents two graphical interactive decision tree construction algorithms able to deal either with (usual) continuous data or with interval and taxonomical data. They are the extensions of two existing algorithms: CIAD [17] and PBC [3]. Both CIAD and PBC algorithms can be used in an interactive or cooperative mode (with an automatic algorithm to find the best split of the current tree node). We have modified the corresponding help mechanisms to allow them to deal with interval-valued attributes. Some of the results obtained on interval-valued and taxonomical data sets are presented with the methods we have used to create these data sets.

1 Introduction

Knowledge Discovery in Databases (or KDD) can be defined [10] as the non-trivial process of identifying patterns in the data that are valid, novel, potentially useful and understandable. In most existing data mining tools, visualization is only used during two particular steps of the data mining process: in the first step to view the original data, and in the last step to view the final results. Between these two steps, an automatic algorithm is used to perform the data-mining task (for example decision trees like CART [8] or C4.5 [19]). The user has only to tune some parameters before running the algorithm and waiting for its results.

Some new methods have recently appeared [22], [15], [24], trying to involve more significantly the user in the data mining process and using more intensively the visualization [9], [20], this new kind of approach is called visual data mining. In this paper we present some methods we have developed, which integrate automatic algorithms, interactive algorithms and visualization methods. These methods are two interactive classification algorithms. The classification algorithms use both human

S.J. Simoff et al. (Eds.): Visual Data Mining, LNCS 4404, pp. 123–135, 2008.

pattern recognition facilities and computer calculus power to perform an efficient user-centered classification. This paper is organized as follows.

In section 2 we briefly describe some existing interactive decision tree algorithms and then we focus on the two algorithms we will use for interval-valued data and taxonomical data. The first one is an interactive decision tree algorithm called CIAD (Interactive Decision Tree Construction) using support vector machine (SVM) and the second is PBC (Perception Based Classifier).

In section 3 we present the interval-valued data: how they can be sorted, what graphical representation can be used and how we perform the graphical classification of these data with our decision tree algorithms.

The section 4 presents the same information as section 3 but concerning the taxonomical data. Then we present some of the results we have obtained in section 5 before the conclusion and future work.

2 Interactive Decision Tree Construction

Some new user-centered manual (i.e. interactive or non-automatic) algorithms inducing decision trees have appeared recently: Perception Based Classification (PBC) [4], Decision Tree Visualization (DTViz) [12], [21] or CIAD [16]. All of them try to involve the user more intensively in the data-mining process. They are intended to be used by a domain expert and not the usual statistician or data-analysis expert. This new kind of approach has the following advantages:

- the quality of the results is improved by the use of human pattern recognition capabilities,
- using the domain knowledge during the whole process (and not only for the interpretation of the results) allows a guided search for patterns,
- the confidence in the results is improved, the KDD process is not just a "black box" giving more or less comprehensible results.

The technical part of these algorithms are somewhat different: PBC and DTViz use an univariate decision tree by choosing split points on numeric attributes in an interactive visualization. They use a bar visualization of the data: within a bar, the attribute values are sorted and mapped to pixels in a line-by-line fashion according to their order. Each attribute is visualized in an independent bar (cf. fig.1). The first step is to sort the pairs ($attr_i$, class) according to attribute values, and then to map to lines colored according to class values. When the data set number of items is too large, each pair ($attr_i$, class) of the data set is represented with a pixel instead of a line. Once all the bars have been created, the interactive algorithm can start. The classification algorithm performs univariate splits and allows binary splits as well as n-ary splits.

Only PBC and CIAD provide the user with an automatic algorithm to help him choose the best split in a given tree node. The other algorithms can only be run in a 100% manual interactive way.

CIAD is a bivariate decision tree using line drawing in a set of two-dimensional matrices (like scatter plot matrices [9]). The first step of the algorithm is the creation of a set of $(n-1)^2/2$ two-dimensional matrices (n being the number of attributes). These

matrices are the two dimensional projections of all possible pairs of attributes, the color of the point corresponds to the class value. This is a very effective way to graphically discover relationships between two quantitative attributes. One particular matrix can be selected and displayed in a larger size in the bottom right of the view (as shown in figure 2 using the Segment data set from the UCI repository [6], it is made of 19 continuous attributes, 7 classes and 2310 instances). Then the user can start the interactive decision tree construction by drawing a line in the selected matrix and performing thus a binary, univariate or bi-variate split in the current node of the tree. The strategy used to find the best split is the following. We try to find a split giving the largest pure partition, the splitting line (parallel to the axis or oblique) is interactively drawn on the screen with the mouse. The pure partition is then removed from all the projections. If a single split is not enough to get a pure partition, each half-space created by the first split will be treated alternately in a recursive way (the alternate half-space is hidden during the current one's treatment).

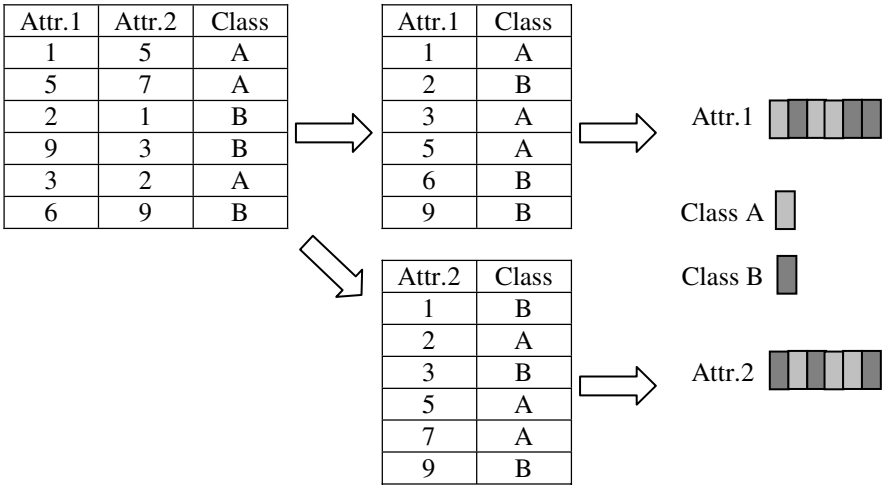


Fig. 1. Creation of the visualization bars with PBC

At each step of the classification, some additional information can be provided to the user like the size of the resulting nodes, the quality of the split (purity of the resulting partition) or overall purity. Some other interactions are available to help the user: it is possible to hide, show or highlight one class, one element or a group of elements.

A help mechanism is also provided to the user. It can be used to optimize the location of the line drawn (the line becomes the best separating line) or to automatically find the best separating line for the current tree node or for the whole tree construction. They are based on a support vector machine algorithm, modified to find the best separating line (in two dimension) instead of the best separating hyperplane (in n-1 dimension for a n-dimensional dataset).

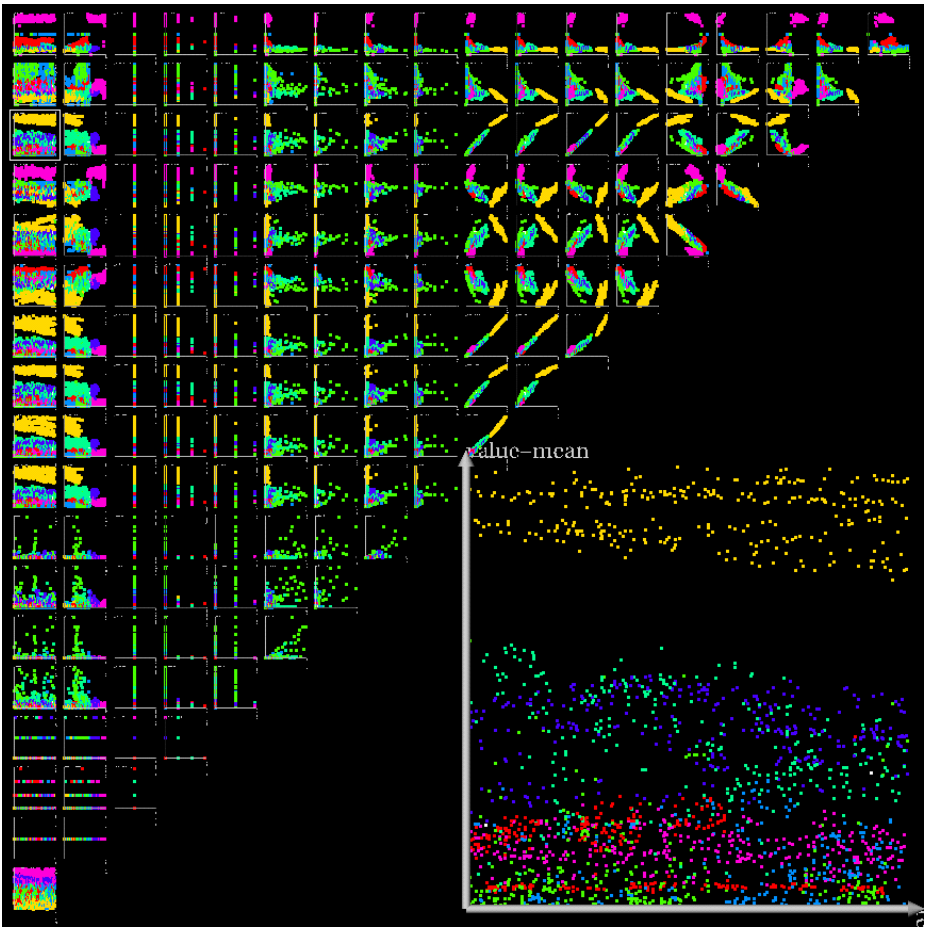


Fig. 2. The Segment data set displayed with CIAD

3 Interval Data

Decision trees usually deal with qualitative or quantitative values. Here we are interested in interval-valued data. This kind of data is often used in polls (for example for income or age). We only consider the particular case of finite intervals.

3.1 Ordering Interval Data

To be able to use this new kind of data with PBC, we need to define an order on these data. There are mainly three different orders we can use [14]: according to the minimum values, the maximum values or the mean values. Let us consider two interval data: $I_1=[l_1,r_1]$ (mean=m_1) and $I_2=[l_2,r_2]$ (mean=m_2).

If the data are sorted according to the minimum values, then:
if $l_1= l_2$, then $I_1 < I_2 <=> r_1< r_2$; if $l_1\neq l_2$, then $I_1 < I_2 <=> l_1< l_2$.

If the data are sorted according to the maximum values, then:
if $r_1 = r_2$, then $I_1 < I_2 <=> l_1 < l_2$; if $r_1 \neq r_2$, then $I_1 < I_2 <=> r_1 < r_2$.

And finally, if the data are sorted according to the mean values, then $I_1 < I_2 <=> m_1 < m_2$.

We can choose any of these three functions to create the bar in the first step of the PBC algorithm in order to sort the data according to the values of the current attribute.

3.2 Graphical Representation of Interval Data

In order to use interval data with CIAD+, we must find what kind of graphical representation can be used in the scatter plot matrices for two interval attributes and for one interval attribute with a continuous one. In the latter case, a segment (colored according to the class) is an obvious solution.

To represent two interval attributes in a scatter plot matrix, we need a two dimensional graphical primitive allowing us to map two different values on its two dimensions, the color being the class. Among the possible choices, there are a rectangle, an ellipse, a diamond, a segment or a cross as shown in figure 3. To avoid occlusion, we must use the outline of the rectangle, the diamond and the ellipse.

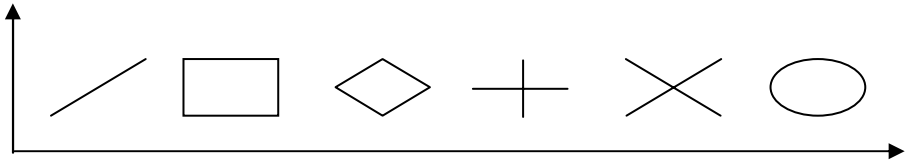


Fig. 3. How to visualize interval x interval data in 2 dimensions?

The rectangle and the diamond will introduce some bias when two rectangles (diamonds) are overlapping, it is impossible to know if there are two or three rectangles (diamonds) drawn as shown in figure 4.

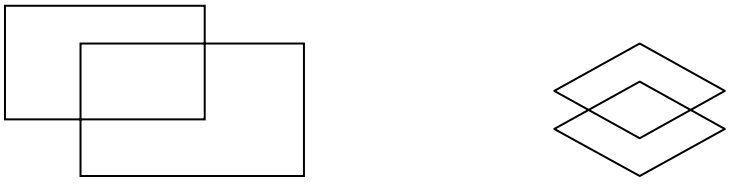


Fig. 4. Are there three or two rectangles and diamonds?

And this can become considerably more complicated if we increase the number of overlapping rectangles or diamonds. For example, in the figure 5, we have drawn 3 rectangles and three diamonds, but it is possible to see between 3 and 6 diamonds and between 3 and at least 19 rectangles!

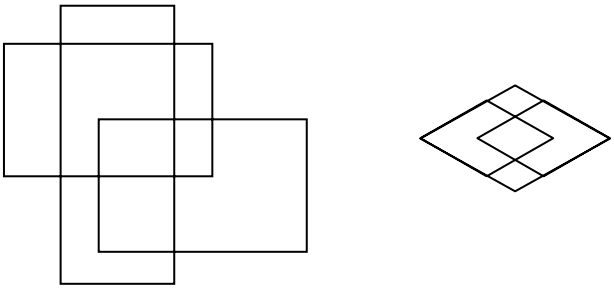


Fig. 5. Three rectangles and three diamonds

So we cannot use rectangles or diamonds; the ellipse, segment and cross do not have the same drawbacks. Concerning the segments, they have another kind of disadvantage: they are drawn from the minimum to the maximum of the two intervals, so they are all in the first quadrant (or third one). When using such a representation, people always try first to find a separating line in the same quadrant, even if a larger pure partition exists but requires a cut in the second (or fourth) quadrant. That is why we have rejected this choice. The only remaining graphical representations are the ellipses and the crosses. The cross being of a lower cost to display, it is the graphical primitive we have chosen.

3.3 Classifying Interval Data with PBC

We have explained how the interval data can be sorted in section 3.1. This method is used in the first step of the PBC algorithm to create the bar charts. Once this task has been performed for each attribute, the classification algorithm is exactly the same as for continuous data (when it is used in its 100% manual mode).

3.4 Classifying Interval Data with CIAD

As explained in section 2, the first step of CIAD is to display a set of two-dimensional matrices being the two-dimensional projections of all possible pairs of attributes, the color corresponding to the class value. This first step will be the same for the interval data, but using crosses instead of points. Once all the matrices have been drawn, the algorithm is exactly the same as the continuous version. We try to find the best pure partition, etc.

In order to keep the same help mechanism we need to adapt the SVM algorithm for dealing with interval-valued data.

3.5 Interval SVM Algorithm

We need to construct linear kernel function for dealing with interval datasets. Let us consider a linear binary classification task with *m* data points in the n-dimensional input space R^n, represented by the mxn matrix *A*, having corresponding labels ±1, denoted by the mxm diagonal matrix *D* of ±1. For this problem, the SVM try to find

the best separating plane, i.e. furthest from both class +1 and class -1. It can simply maximize the distance or margin between the support planes for each class. Any point falling on the wrong side of its supporting plane is considered to be an error. Therefore, the SVM algorithm needs to simultaneously maximize the margin and minimize the error. The proximal SVM classifier proposed by [11] expresses the training in terms of solving a set of linear equations of (w,b) instead of quadratic program (1).

$$[w_1\ w_2\ ..\ w_n\ b]^T = (I/\nu + E^TE)^{-1}E^TDe \quad (1)$$

where E = [A -e].

Our investigation aims at using this PSVM algorithm to classify interval-valued datasets. We show how PSVM can deal with interval-valued data.

Suppose we have two intervals represented by low and high values: $I_1 = [l_1, h_1]$ and $I_2 = [l_2, h_2]$, we use the operational definitions [2] to get the following interval arithmetic:

$I_1 + I_2 = [l_1+l_2, h_1+h_2]$
$I_1 - I_2 = [l_1-h_2, h_1-l_2]$
$I_1 \times I_2 = [\min\{l_1l_2, l_1h_2, h_1l_2, h_1h_2\}, \max\{l_1l_2, l_1h_2, h_1l_2, h_1h_2\}]$
$1/I_1 = [1/h_1, 1/l_1]$ if $l_1 > 0$ or $h_1 < 0$
$I_1/I_2 = I_1 \times 1/I_2$

We use these definitions to solve the set of equations (1) with interval-valued data. Note that $[w_1\ w_2\ ..\ w_n\ b]$ is an interval-valued vector, so we use the mean vector to get the final plane.

Here, we use interval-valued version of Support Vector Machine algorithm in the help mechanism to classify interval-valued data.

4 Taxonomical Data

A taxonomical variable [7] can be defined as a mapping of the original data on a set of ordered values. It is equivalent to a structured or hierarchical variable. For example, a geographical description can be made with the town or with the county or the country. The taxonomical variable describing the location will use any level of the description (town, county or country). In the data set we can find items with a location given by a town name and other ones with a county or country name. From the hierarchical description, we get a set of ordered values by using a tree traversal (either depth-first or width-first). Let us show the results on a very simple example of geographical location. The location is defined by the binary tree described in figure 6. The leaves correspond to town, and the upper levels to county and country.

In the data set, the location attribute can take any value of this tree (except the root value). An example of such a data set is given in table 1, the *a priori* class has two possible values: 1 and 2. The two columns on the left correspond to the original data, the two columns in the middle are the same data set sorted according to a depth-first traversal of the tree, and the two columns on the right are the same data set sorted according to a width-first traversal of the tree.

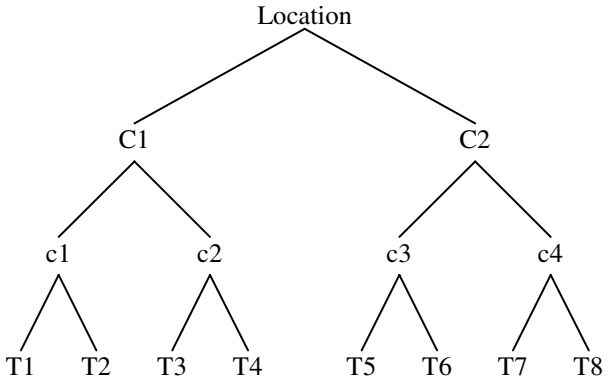


Fig. 6. Hierarchical description of the location

Table 1. An example of taxonomical data set

Location	Class	Location (depth-1st)	Class	Location (width-1st)	Class
T1	1	T1	1	C1	2
T2	2	c1	2	c1	2
T3	1	T2	2	c3	2
T3	1	C1	2	T1	1
c1	2	T3	1	T2	2
C1	2	T3	1	T3	1
T5	1	T5	1	T3	1
c3	2	c3	2	T5	1
T7	1	T7	1	T7	1

4.1 Graphical Representation of a Taxonomical Variable

Once the data have been sorted (whatever the tree traversal is), a taxonomical variable can be seen as an interval variable. When the variable is not a leaf of the tree (for example C1 or c3 in figure 6), it is graphically equivalent to the interval made of all the leaves of the corresponding sub-tree (C1=[T1,T4] and c3=[T5,T6]). In a two dimensional representation, we will use exactly the same graphical primitive as for the interval data: a cross for (taxonomical x taxonomical) or (taxonomical x interval) representation and a segment for (taxonomical x continuous) representation.

4.2 Interactive Taxonomical Data Classification

Here again, the way PBC is used is exactly the same as for interval or continuous data (when it is used in 100% manual mode). There is an order for the taxonomical data, it is used in the first step of the PBC algorithm, to sort the data according to the attribute value.

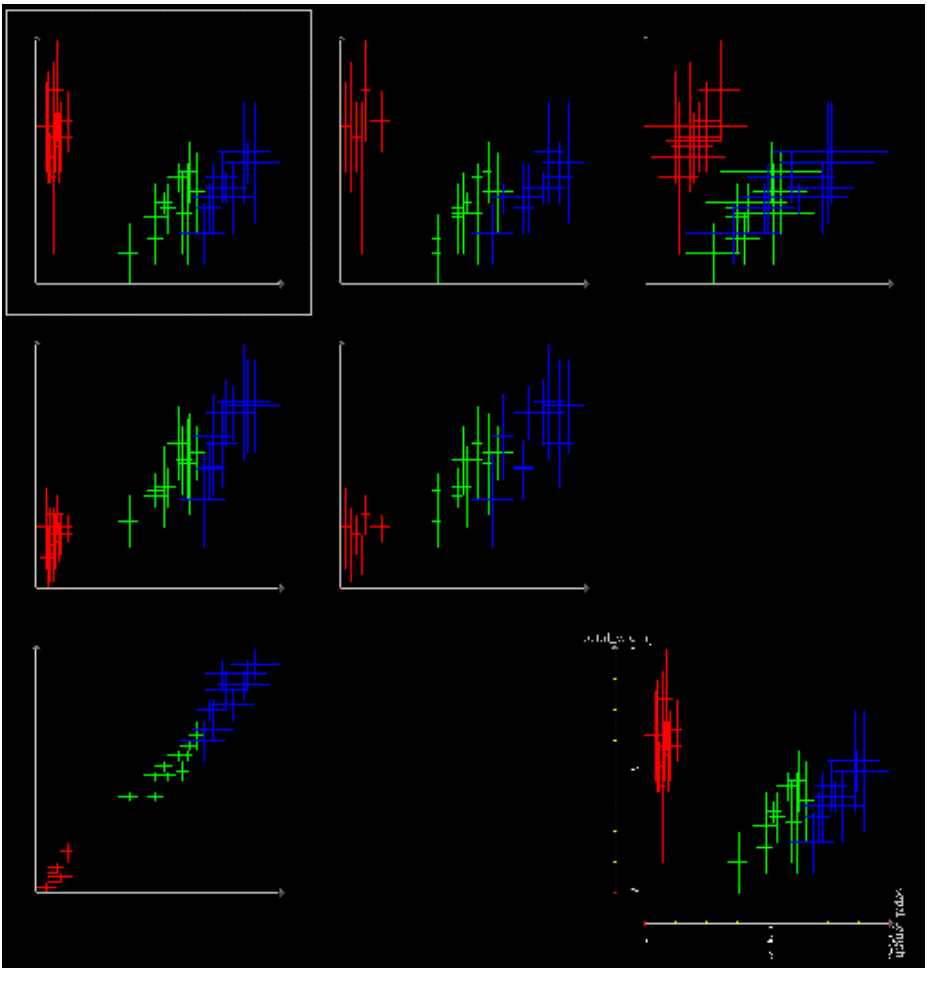


Fig. 7. Interval-valued version of the iris data set

CIAD is also used the same way as for interval data. The help mechanisms can be used if we consider the interval corresponding to the taxonomical data treated.

5 Some Results

First of all, we must underline that as far as we know, there is no other decision tree algorithm able to deal with interval and taxonomical data and there are no available interval-valued data sets in existing machine learning repositories. We present in this section some of the results we have obtained and we start with the description of the data sets we have created. We have used existing data sets with continuous variables to create the interval-valued data sets. The first data set used is the well-known iris data set from the UCI Machine Learning Repository [6]. First, a new attribute has

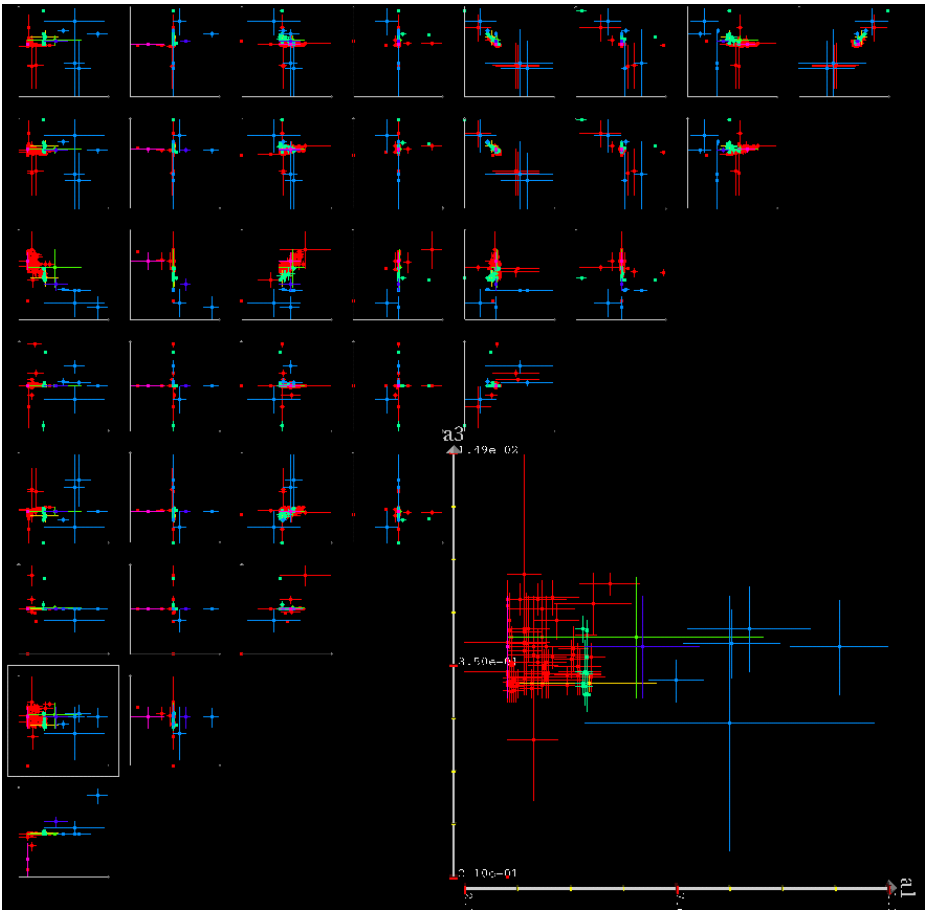


Fig. 8. Interval version of the shuttle data set

been added to data set: the petal surface. Then the data set has been sorted according to this new attribute (it nearly perfectly sorts the iris types) and we have computed (for each attribute) the minimum and the maximum values of each group of five consecutive items. And so we obtain a data set made of four interval-valued attributes and 30 items (10 for each class).

The resulting display with the CIAD set of 2D scatter plot matrices is shown in figure 7. The original data set has one class linearly separable from the other two, and the other two not linearly separable from each other. The interval data have three linearly separable classes. Some of the data set items are represented with a segment because the minimum and maximum have the same value according to the second attribute used in the matrix.

The second data set we have created is an interval-valued version of the shuttle data set (which also comes from the UCI Machine Learning repository). This data set is made of nine continuous attributes, 43500 items and seven classes. Four of them

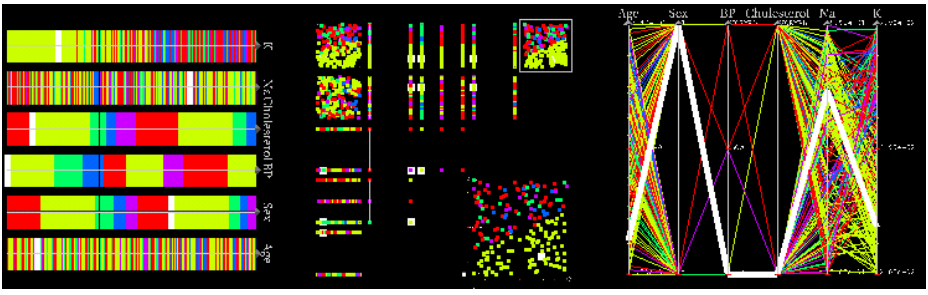


Fig. 9. Three linked representation of the drug data set

have very few items and for these four classes we have only created one interval-valued item with the average value plus and minus the standard deviation for each attribute. The three remaining classes have more items, we wanted to have nearly one hundred interval-valued items, so we created a number of interval-valued items in proportion to the original number of continuous items. These items have been computed by a clustering algorithm (k-means) to get similar continuous elements in an interval-valued one (then the average value plus and minus the standard deviation are computed in each cluster for all the attributes). The same method has been used for both the training set and the test set. The graphical display of the shuttle-interval training set is shown in figure 8.

Once these data are displayed, we perform the interactive creation of the decision tree. The accuracy obtained on the training set is 100% with a 10-leaf tree (99.7% on the test set with a ten-fold cross-validation). On the continuous data set, the accuracy obtained with CIAD was 99.9% on the test set with a tree size of nine leaves. As we can see, the tree size of the interval-valued data set is larger than the continuous one and the accuracy is lower. This is because the interval-valued data set has only one hundred elements compared to the 43500 elements of the original data set. One misclassified element has an accuracy loss of 1% on the training set and 3% on the test set in the first case and 0.002% (and 0.006%) in the later one. To get a similar accuracy, the interval-valued tree needs more splits (and more leaves) to avoid any misclassification when the continuous version allow tens of misclassification errors while keeping the accuracy greater than 99.9%.

To evaluate the accuracy on the test set, our decision tree algorithm gives as output the source code of a C program we only need to compile and run it to compute the accuracy on the test set. We had not enough time to manage other large datasets, we are working on a software program being able to automatically create an interval-valued data set from a continuous one.

6 Conclusion and Future Work

Before concluding, some words about the implementation. All these tools have been developed using C/C++ and three open source libraries: OpenGL, Open-Motif and Open-Inventor. OpenGL is used to easily manipulate 3D objects, Open-Motif for the graphical user interface (menus, dialogs, buttons, etc.) and Open-Inventor to manage

the 3D scene. These tools are included in a 3D environment, described in [18], where each tool can be linked to other tools and be added or removed as needed. Figure 9 shows an example with CIAD set of 2D scatter plot matrices, PBC bar charts and parallel coordinates [13]. The element selected in a bar chart appeared selected too in the set of scatter plot matrices and in the parallel coordinates (in bold white). The software program can be run on any platform using X-Window, it only needs to be compiled with a standard C++ compiler. Currently, the software program is developed on SGI O2 and PCs with Linux.

In this paper we have presented two new interactive classification tools able to deal with interval-valued and taxonomical data. The classification tools are intended to involve the user in the whole classification task in order to:

- take into account the domain knowledge,
- improve the result comprehensibility, and the confidence in the results (because the user has taken part in the model construction),
- exploit human capabilities in graphical analysis and pattern recognition.

The possibility to deal with interval-valued data is a convenient way to overcome one of the most important drawbacks of interactive classification methods, the limit of the dataset size. Here, we deal with a higher-level representation of the data. This allows us to treat potentially very large dataset because we do not deal with the original data, but this higher-level representation of the data. Furthermore, the possibility to deal with taxonomical data also allows us to simultaneously deal with different higher-level data representations.

A forthcoming improvement will be to try to use the same kind of abstraction method with some other classification or data-mining algorithms (here we were in the particular case of supervised classification with decision trees).

References

1. Aggarwal, C.: Towards Effective and Interpretable Data Mining by Visual Interaction. SIKDD Explorations 3(2), 11–22, http://www.acm.org/sigkdd/explorations/
2. Alefeld, G., Herzberger, J.: Introduction to Interval Computations. Academic Press, New York (1983)
3. Ankerst, M., Elsen, C., Ester, M., Kriegel, H.-P.: Perception-Based Classification, in Informatica. An International Journal of Computing and Informatics 23(4), 493–499 (1999)
4. Ankerst, M.: Visual Data Mining, PhD Thesis, Ludwig Maximilians University of Munich (2000)
5. Ankerst, M., Ester, M., Kriegel, H.-P.: Toward an Effective Cooperation of the Computer and the User for Classification. In: Proc. of KDD 2001, pp. 179–188 (2001)
6. Blake, C., Merz, C.: UCI Repository of machine learning databases, University of California, Department of Information and Computer Science, Irvine, CA (1998), http://www.ics.uci.edu/~mlearn/MLRepository.html
7. Bock, H.H., Diday, E.: Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data. Springer, Berlin (2000)
8. Breiman, L., Friedman, J., Olsen, R., Stone, C.: Classification and Regression Trees, Wadsworth (1984)

9. Chambers, J., Cleveland, W., Kleiner, B., Tukey, P.: Graphical Methods for Data Analysis. Wadsworth (1983)
10. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.): Advances in Knowledge Discovery and Data Mining. AAAI Press, Menlo Park (1996)
11. Fung, G., Mangasarian, O.: Proximal Support Vector Machine Classifiers. In: Proc. of the 7th ACM SIGKDD, Int. Conf. on KDD 2001, San Francisco, USA, pp. 77–86 (2001)
12. Han, J., Cercone, N.J.: Interactive Construction of Decision Trees. In: Cheung, D., Williams, G.J., Li, Q. (eds.) PAKDD 2001. LNCS (LNAI), vol. 2035, pp. 575–580. Springer, Heidelberg (2001)
13. Inselberg, A., Avidan, T.: Classification and Visualization for High-Dimensional Data. In: Proc. of KDD 2000, pp. 370–374 (2000)
14. Mballo, C., Gioia, F., Diday, E.: Qualitative Coding of an Interval-Valued Variable. In: Proc. of the 35th Conference of the French Statistical Society, Lyon, France (in french) (June 2003)
15. Poulet, F.: Visualization in data mining and knowledge discovery. In: Lenca, P. (ed.) Proc. of HCP 1999, 10th Mini Euro Conference Human Centered Processes, Brest, pp. 183–192 (1999)
16. Poulet, F.: CIAD: Interactive Decision Tree Construction. In: Proc. of 8th Conf. of the French Classification Society, Pointe-à-Pitre, pp. 275–282 (2001) (in French)
17. Poulet, F.: Cooperation Between Automatic Algorithms, Interactive Algorithms and Visualization Tools for Visual Data Mining. In: Proc. of VDM@ECML/PKDD 2002, International Workshop on Visual Data Mining, Helsinki, Finland, pp. 67–80 (2002)
18. Poulet, F.: FullView: A Visual Data-Mining Environment. International Journal of Image and Graphics 2(1), 127–144 (2002)
19. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan-Kaufman Publishers, San Francisco (1993)
20. Schneiderman, B.: Inventing Discovery Tools: Combining Information Visualization with Data Mining. Information Visualization 1(1), 5–12 (2002)
21. Ware, M., Franck, E., Holmes, G., Hall, M., Witten, I.: Interactive Machine Learning: Letting Users Build Classifiers. International Journal of Human-Computer Studies 55, 281–292 (2001)
22. Wong, P.: Visual Data Mining. IEEE Computer Graphics and Applications 19(5), 20–21 (1999)

Visual Methods for Examining SVM Classifiers

Doina Caragea[1], Dianne Cook[2], Hadley Wickham[2], and Vasant Honavar[3]

[1] Dept. of Computing and Information Sciences, Kansas State University, Manhattan, KS 66502, USA
[2] Dept. of Statistics, Iowa State University, Ames, IA 50011, USA
[3] Dept. of Computer Science, Iowa State University, Ames, IA 50011, USA

Abstract. Support vector machines (SVM) offer a theoretically well-founded approach to automated learning of pattern classifiers. They have been proven to give highly accurate results in complex classification problems, for example, gene expression analysis. The SVM algorithm is also quite intuitive with a few inputs to vary in the fitting process and several outputs that are interesting to study. For many data mining tasks (e.g., cancer prediction) finding classifiers with good predictive accuracy is important, but understanding the classifier is equally important. By studying the classifier outputs we may be able to produce a simpler classifier, learn which variables are the important discriminators between classes, and find the samples that are problematic to the classification. Visual methods for exploratory data analysis can help us to study the outputs and complement automated classification algorithms in data mining. We present the use of tour-based methods to plot aspects of the SVM classifier. This approach provides insights about the cluster structure in the data, the nature of boundaries between clusters, and problematic outliers. Furthermore, tours can be used to assess the variable importance. We show how visual methods can be used as a complement to cross-validation methods in order to find good SVM input parameters for a particular data set.

1 Introduction

The availability of large amounts of data in many application domains (e.g., bioinformatics or medical informatics) offers unprecedented opportunities for knowledge discovery in such domains. The classification community has focused primarily on building accurate predictive models from the available data. Highly accurate algorithms that can be used for complex classification problems have been designed. Although the predictive accuracy is an important measure of the quality of a classification model, for many data mining tasks understanding the model is as important as the accuracy of the model itself. Finding the role different variables play in classification provides an analyst with a deeper understanding of the domain. For example, in medical informatics applications, such an understanding can lead to more effective screening, preventive measures and therapies.

The SVM algorithm [39] is one of the most effective machine learning algorithms for many complex binary classification problems (e.g., cancerous or

S.J. Simoff et al. (Eds.): Visual Data Mining, LNCS 4404, pp. 136–153, 2008.

normal cell prediction based on gene expression data [8]). In the simplest case, SVM algorithm finds a hyperplane that maximizes the margin of separation between classes. This hyperplane is defined by a subset of examples, called support vectors, which "mark" the boundary between classes. However, understanding the results and extracting useful information about class structure, such as what variables are most important for separation, is difficult. SVM is mostly used as a black box technique.

The SVM algorithm searches for "gaps" between clusters in the data, which is similar to how we cluster data using visual methods. Thus, SVM classifiers are particularly attractive to explore using visual methods. In this paper, we use dynamic visual methods, called tours [4,14,13], to explore SVM classifiers. Tours provide mechanisms for displaying continuous sequences of low-dimensional linear projections of data in high-dimensional Euclidean spaces. They are generated by constructing an orthonormal basis that represents a linear subspace. Tour-based methods are most appropriate for data that contain continuous real-valued variables. They are useful for understanding patterns, both linear and non-linear, in multi-dimensional data. However, because tours are defined as projections (analogous to an object shadow) rather than slices, some non-linear structures may be difficult to detect. Tours are also limited to applications where the number of variables is less than 20 because otherwise the space is too large to randomly explore within a reasonable amount of time. Hence, when we have more than 20 variables, it is important to perform some dimensionality reduction prior to applying tour methods. In classification problems, tours allow us to explore the class structure of the data, and see the way clusters are separated (linearly or not) and the shape of the clusters.

Visualization of the data in the training stage of building a classifier can provide guidance in choosing variables and input parameters for the SVM algorithm. We plot support vectors, classification boundaries, and outliers in high-dimensional spaces and show how such plots can be used to assess variable importance with respect to SVM, to complement cross-validation methods for finding good SVM input parameters and to study the stability of the SVM classifiers with respect to sampling.

Effective application of machine learning algorithms, SVM included, often requires careful choice of variables, samples and input parameters in order to arrive at a satisfactory solution. Hence, a human analyst is invaluable during the training phase of building a classifier. The training stage can be laborious and time-intensive, but once a classifier is built it can repeatedly be used on large volumes of data. Therefore, it is valuable to take the time to explore alternative variable, samples, parameter settings, plot the data, meticulously probe the data, to generate accurate and comprehensible classifiers.

Our analysis is conducted on a particular data problem, SAGE gene expression data [6], where the task is to classify cells into cancerous cells or normal cells based on the gene expression levels.

The rest of the paper is organized as follows: The **Methods** section describes the algorithms for SVM and tours, and also the aspects that we study to

understand and explore the SVM model; The **Application** section illustrates how tour methods can be used to examine SVM classifiers, using a SAGE gene expression data set; The **Summary and Discussion** section summarizes our methods, describes their strengths and limitations, and presents some related work.

2 Methods

2.1 Support Vector Machines

The SVM algorithm [39] is a binary classification method that takes as input labeled data from two classes and outputs a model (a.k.a., classifier) for classifying new unlabeled data into one of those two classes. SVM can generate linear and non-linear models.

Let $E = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_l, y_l)\}$, where $\mathbf{x}_i \in R^p$ and $y_i \in \{-1, 1\}$ be a set of training examples. Suppose the training data is *linearly separable*. Then it is possible to find a hyperplane that partitions the p-dimensional pattern space into two half-spaces R^+ and R^-. The set of such hyperplanes (the solution space) is given by $\{\mathbf{x}|\mathbf{x} \cdot \mathbf{w} + b = 0\}$, where $\mathbf{x}$ is the p-dimensional data vector and $\mathbf{w}$ is the normal to the separating hyperplane.

SVM selects among the hyperplanes that correctly classify the training set, the one that minimizes $\|\mathbf{w}\|^2$ (subject to the constraints $y_i(\mathbf{x_i} \cdot \mathbf{w} + b) \leq 1$), which is the same as the hyperplane for which the *margin* of separation between the two classes, measured along a line perpendicular to the hyperplane, is maximized.

The algorithm assigns a weight α_i to each input point $\mathbf{x}_i$. Most of these weights are equal to zero. The points having non-zero weight are called *support vectors*. The separating hyperplane is defined as a weighted sum of support vectors. Thus, $\mathbf{w} = \sum_{i=1}^{l}(\alpha_i y_i)\mathbf{x_i} = \sum_{i=1}^{s}(\alpha_i y_i)\mathbf{x_i}$, where s is the number of support vectors, y_i is the known class for example $\mathbf{x}_i$, and α_i are the support vector coefficients that maximize the margin of separation between the two classes. The classification for a new unlabeled example can be obtained from $f_{\mathbf{w},b}(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b) = \text{sign}(\sum_{i=1}^{l} \alpha_i y_i (\mathbf{x} \cdot \mathbf{x_i}) + b)$.

If the goal of the classification problem is to find a linear classifier for a non-separable training set (e.g., when data is noisy and the classes overlap), a set of *slack variables*, ξ_i, is introduced to allow for the possibility of examples violating the constraints $y_i(\mathbf{x_i} \cdot \mathbf{w} + b) \leq 1$. In this case the margin is maximized, paying a penalty proportional to the cost C of constraint violation, i.e., $C\sum_{i=1}^{l}\xi_i$. The decision function is similar to the one for the linearly separable problem. However, in this case, the set of support vectors consists of *bounded support vectors* (if they take the maximum possible value, C) and *unbounded* (*real*) *support vectors* (if their absolute value is smaller than C).

If the training examples are not linearly separable, the SVM works by mapping the training set into a higher dimensional *feature* space, where the data becomes linearly separable, using an appropriate kernel function k.

The SVM algorithm has several input parameters that can be varied (e.g., cost, C, tolerance in the termination criterion, ϵ, kernel function, k) and several

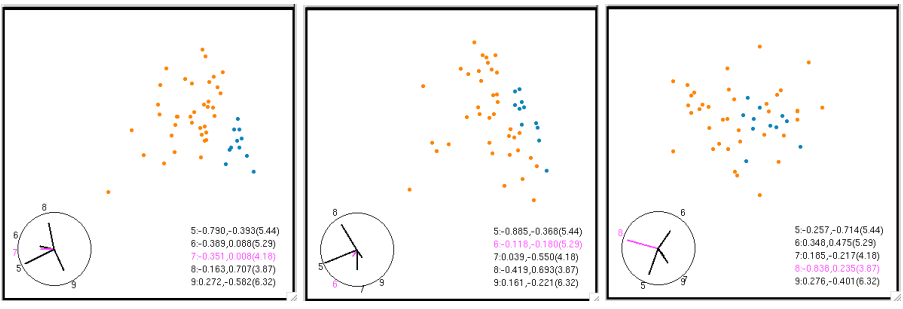


Fig. 1. Three projections from a tour of a 5-dimensional data set (variables denoted V_5,V_6,V_7,V_8 and V_9) where the two groups are separable. The left plot shows a projection where the groups are well-separated, and the plot at right shows a projection where they are not separated. The magnitude of the projection coefficients indicate variable importance, the larger the coefficient - in the direction of separation - the more important the variable. For example, the left plot shows $\frac{-0.790}{5.44}V_5 - \frac{0.389}{5.29}V_6 - \frac{0.351}{4.18}V_7 - \frac{0.163}{3.87}V_8 + \frac{0.272}{6.32}V_9$ horizontally, and $\frac{-0.393}{5.44}V_5 + \frac{0.088}{5.29}V_6 + \frac{0.008}{4.18}V_7 + \frac{0.707}{3.87}V_8 + \frac{-0.582}{6.32}V_9$ vertically.

outputs that can be studied to assess the resulting model (e.g., support vectors, separating hyperplane, variables that are important for the separation).

In this paper, we use the SVM implementation available in the R [32] package, called `e1071` [17]. The SVM implementation in `e1071` is based on the LIBSVM implementation [12]. We use this implementation because the R language allows us to quickly calculate other diagnostic quantities and to link these numbers to graphics packages.

2.2 Tours Methods for Visualization

Tours [4,41,42,9,16] display linear combinations (projections) of variables, $\mathbf{x}'\mathbf{A}$ where $\mathbf{A}$ is a $p \times d(< p)$-dimensional projection matrix. The columns of A are orthonormal. Often $d = 2$ because the display space on a computer screen is 2, but it can be 1 or 3, or any value between 1 and p. The earliest form of the tour presented the data in a continuous movie-like manner, but recent developments have provided guided tours [14] and manually controlled tours [13]. Here we use a $d = 2$-dimensional manually-controlled tour to recreate the separating boundary between two groups in the data space. Figure 1 illustrates the tour approach.

We use the tour methods available in the data visualization software ggobi [37], and the R [32] package Rggobi [38] that makes ggobi functionality accessible from R.

2.3 SVM and Tours

Understanding the classifier in relation to a particular data requires an analyst to examine the suitability of the method on the data, adjust the performance of the method (e.g., by appropriate choice of parameters) and adjust the data

(e.g., by appropriate choice of variables used for building the classifier) as necessary to obtain the best results on the problem at hand. Tour methods can be used as exploratory tools to examine the outputs of SVM classifiers.

There are several outputs that we can examine when exploring SVM results using tours: support vectors, boundaries between classes, outliers, among others. The support vectors specify the classifier generated by the SVM algorithm. First, we observe their location in the data space and examine their importance (position) relative to the other data points. We expect to see the unbounded support vectors from each group roughly indicating the margin between the groups in some projection. The bounded support vectors should lie inside this margin.

Second, we examine the boundaries between classes in high dimensional spaces. To do this, we generate a rectangular grid of points around the data, by choosing a number of grid points between the minimum and maximum data value of each variable. For example, with two variables, 10 grid values on each axis will give $10^2 = 100$ points on a square grid, or with four variables we would have $10^4 = 10000$ points on a 4D grid. We then compute the predicted values $\mathbf{w} \cdot \mathbf{x} + b$ for each point $\mathbf{x}$ in the grid. The points that are close to the boundary will have predicted values close to 0. For two variables the boundary is a line, for three variables the boundary is a 2D plane, for four variables the boundary is a 3D hyperplane, etc. When using linear kernels with SVM, we expect to see very clear linear boundaries between the two groups. For non-linear kernels, the boundaries will be more complex.

Third, we investigate anomalies in the data, the misclassified samples and the outliers, to get insights about how these points differ from the rest of the data. The anomalies should be isolated from their group in some way. We look at a separate test set after the classifier is built from a training data set and the projection that shows the training separation is found.

The visualization of the outputs can be explored to:

1. Assess the importance of the variables based on the best projections observed;
2. Tune SVM input parameters according to the outputs observed;
3. Study the stability of the model with respect to sampling.

Assessing variable importance. Real world data sets are described by many variables (e.g., for gene expression data there are commonly a lot more variables than examples). A classifier may be unreliable unless the sample size is several times as large as the number of variables [34]. Very often, a small number of the variables suffices to separate the classes, although the subset of variables may not be unique [27]. Variable selection is also useful before running tours, because the smaller the space the more rapidly it can be explored. There are many methods for variable selection [21,22,33,7,19,26,44] and the subsets of variables that they return can be very different. Which subset is the best? We use tours to explore and select subsets of variables than can be used to separate the data in two classes.

To do that, we first order the variables according to several criteria (e.g., PDA-PP index [26], BW index [19] and SVM variable importance [22]) and form small subsets, either by taking the best k variables according to one criterion or by combining the best variables of two or more criteria. After running SVM on the subsets formed with the variables selected, we examine the difference between results and select those subsets of variables that show the best separation between classes in some projection.

We can also assess the importance of the variables within a subset. The support vectors from each group roughly indicate a boundary between the groups in some projection. The variables contributing to the projection provide an indication of relative importance of the variables to the separation. The coefficients of the projection (elements of P) are used to examine the variable contribution.

Tuning the parameters. The performance of the classifier depends on judicious choice of various parameters of the algorithm. For SVM algorithm there are several inputs that can be varied: the cost C (e.g., $C = 1$), the tolerance ϵ of the termination criterion (e.g., $\epsilon = 0.01$), the type of kernel that is used (e.g., linear, polynomial, radial or Gaussian), and the parameters of the kernel (e.g., degree or coefficients of the polynomial kernel), etc. It is interesting to explore the effect of changing the parameters on the performance of the algorithm. Visual methods can complement cross-validation methods in the process of choosing the best parameters for a particular data set. In addition, examination of the SVM results for different parameters can help understanding better the algorithm and the resulting classifiers.

Stability of the classifier. Machine learning algorithms typically trade-off between the classification accuracy on the training data and the generalization accuracy on novel data. The generalization accuracy can be estimated using a separate test set or using bootstrap and cross-validation methods. All these methods involve sampling from the initial data set. It is useful to explore how the sampling process affects the classifier for the particular data at hand. This can be accomplished by studying the variation of the separation boundary, which can provide insights about the stability of the algorithm.

3 Application

3.1 Data Description

We use SAGE (Serial Analysis of Gene Expression) [40] data to illustrate the visual methods described in this paper. SAGE is an experimental technique that can be used to quantify gene expression and study differences between normal and cancerous cells [45]. SAGE produces tags (10-base sequences) that identify genes (mRNA). The frequencies of these tags can be seen as a measure of the gene expression levels in the sampled cells. Different from microarray technology, SAGE does not need the sequences of the set of genes to be known. This allows for the possibility that genes related to cancer, but whose sequences or functionality

have not been discovered, to be identified. However, SAGE technology is very expensive and there is not much data available.

It is believed that cancers are caused by mutations that alter the normal pattern in gene expression [45]. Genes exhibiting the greatest differences in the expression levels corresponding to normal or cancerous cells are most likely to be biologically significant. One difficulty with SAGE data, when trying to identify genes that distinguish between cancerous and normal cells, is that different samples can come from very different types of tissues (e.g., brain, breast, lung, etc.) as well as from *in vivo* and *in vitro* samples. It is known that different types of tissues are characterized by different expression patterns and they cluster together [28]. The same is believed to be true for *in vivo* and *in vitro* conditions. This makes it difficult to assert that genes whose expression levels are different in cancerous versus non-cancerous cells are indeed responsible for cancer. However, given the scarcity of the data (not too many samples from the same tissues) any findings along these directions are often interesting and potentially useful in clinical applications.

Analysis of SAGE data has received a lot of attention in the last few years. The data set used in our analysis is introduced in [6], which also provides information about the data preprocessing. It was assembled from the complete human SAGE samples (`http://www.ncbi.nlm.nih.gov/sage`) by selecting a subset of tags corresponding to a minimal transcriptome set. Our subset contains the expression values (transcripts per cell) for 822 genes found in 74 human cells. The study in [6] shows that genes with similar functionality cluster together when a strong-association-rule mining algorithm is applied.

3.2 Visualizing SVM Outputs

In this section, we show how to explore the SVM outputs using tours. Suppose that the set of important variables for the analyst is $S = \{V800, V403, V535, V596\}$. We apply SVM algorithm on this data set. The results are shown in Figure 2. The two classes are colored with different colors. The support vectors (1 in one class and 3 in the other class) have larger glyphs. The left plot shows a projection where the linear separation found by SVM can be seen. The support vectors line up against each other defining the boundary between the two classes. The coefficients of the projection are also shown. The separation between the two classes is in the top left to bottom right direction, which is a combination of most of the variables.

Using only 4 variables, it is easy to generate a grid around the data. The class of grid points can be predicted using SVM algorithm. Coloring the grid points according to the predictions allows us to see the boundary estimated by SVM. A good separation of the grid can be seen in the middle plot in Figure 2. Coloring the grid points that have predicted values close to 0 allows us to focus on the boundary between the two groups (right plot in Figure 2).

To assess the quality of the model and to find outliers, we divide the examples into training and test sets, build the model for the training data, and find the projection showing the separation between classes. We then plot the test data in

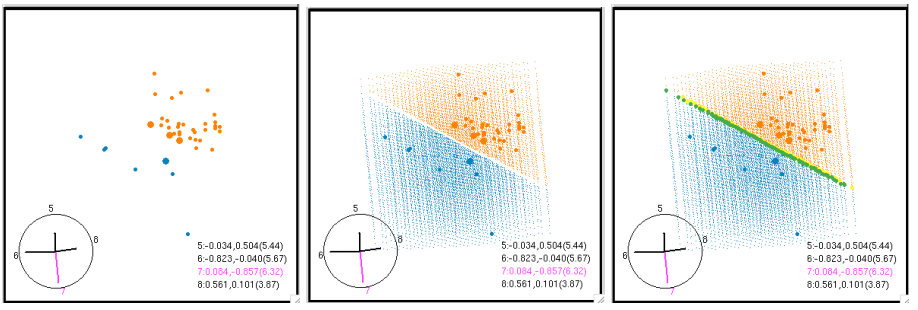


Fig. 2. SVM results for a 4-dim data set. (Left) A projection where the linear separation found by SVM can be seen. (Middle) A grid over the data colored according to the class colors. (Right) The grid points that have predicted values close to 0, define a nice linear boundary between the two groups.

the same projection to see how well it is separated, and to examine errors and outliers (Figure 3).

If we use SVM with non-linear kernels, non-linear separations can also be viewed (results presented in an expanded version of this paper [11]).

The ggobi brushing interface allows the user to shadow or show different groups of points, making it very easy to focus on different parts of the model for exploratory analysis. The ggobi main interface allows selecting different groups of variables to be shown. The ggobi identity interface allows identifying points in the plot with points in the data. See [11] for figures showing these interfaces.

The classifly R package [43] automates the process of classifier fitting and grid generation. It does this using rggobi to allow a seamless transition between classifier building in R and visualisation in GGobi. Classifly can display either the complete grid, or just the boundaries of the classification regions.

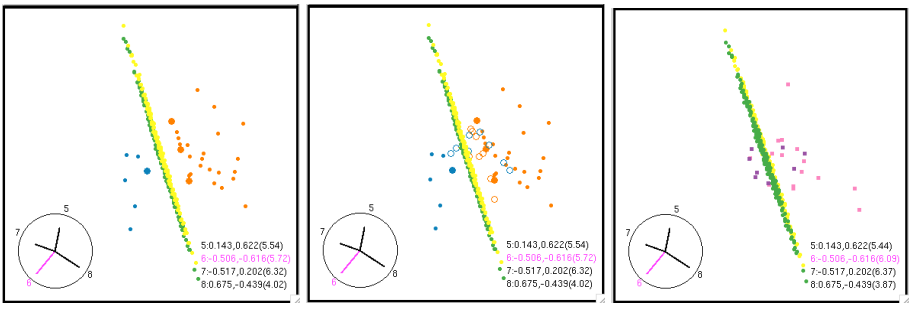


Fig. 3. A subset of the data (2/3 from each class) is used for training and the rest of the data is used for test. (Left) A projection showing the separation for the training set. (Middle) Bounded support vectors are also shown as open circles. (Right) The test set is shown with respect to the separating hyperplane found by the algorithm. We can see the errors. They belong entirely to one group.

3.3 Gene Selection

To construct reliable classifiers from SAGE data, we need to select a small set of genes. As Liu et al. [27] have shown, variable selection for gene expression data usually improves the accuracy of the classifier. Variable selection is also necessary in order to make it possible to view the data using tours. As mentioned earlier the initial set has 822 variables (genes), which makes it impossible for visual analysis.

To select small sets of genes, first, data is standardized, so that each gene has mean 0 and variance 1 across samples. Three different methods, BW index [19], PDA-PP index [26] and SVM variable importance [22], are used to order the 822 variables according to their importance with respect to the criterion used by each method. The BW index of a set of genes gives the ratio of their between-group to within-group sums of squares. The PDA-PP index represents a projection pursuit index corresponding to the penalized discriminant analysis [23]. The SVM importance of the variables is determined by running SVM iteratively several times, each time the less important variable - with the smallest w_i^2 - being eliminated. The reverse order of the eliminated variables represents the order of importance [22].

The best 40 genes based on BW index are:

V721, V113, V61, V299, V176, V406, V409, V596, V138, V488, V663, V208, V165, V403, V736, V535, V70, V803, V112, V417, **V357**, V166, V761, V119, V666, V99, V398, V769, V26, V4, V55, V277, V788, V73, V45, V800, V111, V523, V94, V693.

The best 40 genes based on PDA-PP index are:

V721, **V357**, V50, V594, V559, V119, V575, V663, V523, V298, V578, V372, V817, V6, V97, V74, V299, V287, V810, V190, V655, V768, V710, V667, V390, V331, V513, V58, V661, V581, V60, V713, V509, V463, V644, V654, V799, V759, V797, V751

The best 40 genes based on SVM variable importance are:

V390, V389, V4, V596, V54, V409, V741, V398, V725, V736, V581, V263, V817, V701, V655, V165, **V357**, V157, V545, V692, V488, V70, V714, V638, V594, V195, V713, V7, V148, V150, V138, V616, V269, **V721**, V603, V208, V517, V94, V289, V424

In general, SVM takes more time to run than methods such as BW index. Therefore, we also considered the possibility of first ordering all the 822 genes according BW index and subsequently ordering the best 40 genes found by BW index according to the SVM variable importance. The result is shown below:

V800, V403, V535, V596, **V357**, V398, V113, V73, V119, V488, V99, V138, V736, V26, V803, V112, V693, V165, V406, V788, V277, V45, V666, V176, **V721**, V663, V417, V769, V208, V111, V523, V761, V55, V166, V94, V299, V409, V70, V61, V4

The set of genes selected by each method is quite different from the others. However, there are two genes that are on the lists of all three methods: **V721** and **V357**. Surprisingly many more genes are common for the BW and SVM

gene selection methods: V596, V409, V4, V721, V138, V488, V208, V165, V736, V70, V357, V398, V94.

Given the difference in subsets of important genes, the question is: which one is the best? Not very surprisingly, different subsets of genes give comparable results in terms of classification accuracy, which makes the choice difficult. However, this is where visual methods can help. We use tours to explore different sets of genes and visualize the results of SVM algorithm on those particular subsets. This gives us an idea about how different sets of genes behave with respect to SVM algorithm.

First, to determine how many genes are needed to accurately separate the two classes, we select subsets of the 40 genes and study the variation of the error with the number of genes, using cross-validation. The initial data set is randomly divided into a training set (2/3 of all data) and a test set (1/3 of all data), then SVM is run on the training set, and the resulting model is used to classify both the training set and the test set. The errors are recorded together with the number of unbounded (real) and bounded support vectors for each run. This is repeated 100 time and the average over the measured values is calculated. Then a new variable is added and the process is repeated.

Plots for the variation in average accuracy for the training and test sets (i.e., fraction of the misclassified examples relative to the number of training and test examples, respectively), as well as for the fraction of unbounded support vectors and bounded support vectors (relative to the number of training examples) with the number of variables, are shown in Figure 4. The variables used are the best 20 SVM variables selected from the set of the best 40 BW variables. The average training error decreases with the number of variables and it gets very close to 0 when 20 variables are used. In the test error the average decreases and then starts to rise around 12 variables. There is a dip at 4 variables in both training and test error, and a plateau at 7 variables in the test error. The observed number of unbounded and bounded support vectors shows that there is a negative correlation between the two: as the number of unbounded support vector increases, the number of bounded support vectors decreases. This corresponds to our intuition: as the number of dimensions increases, more unbounded support vectors are needed to separate the data.

As the tours are easier to observe when less variables are used, we chose to look at sets of 4 variables. Although the errors for 4 variables are slightly higher than the errors obtained using more variables, the analysis should give a good picture of class separations in the data.

We tried various combinations of subsets of 4 variables formed from the lists of most important variables. Some of these subsets gave very poor accuracy, some gave reasonable good accuracy. Table 1 shows a summary of the results obtained for three subsets of variables that give good accuracy: $S1 = \{V800, V403, V535, V596\}$ (first 4 most important SVM genes from the best 40 BW genes), $S2 = \{V390, V389, V4, V596\}$ (first 4 most important SVM genes from all 822 genes) and $S3 = \{V800, V721, V357, V596\}$ (a combination of the

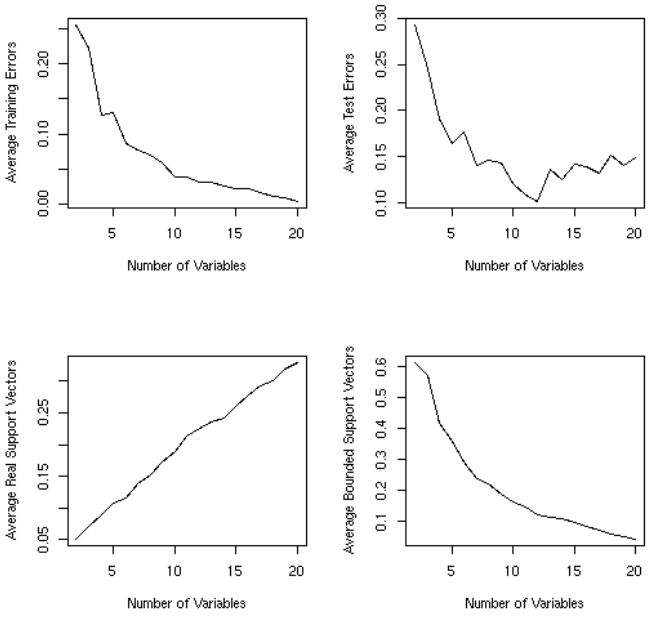


Fig. 4. (Top) The variation of the average (over 100 runs) training and test errors (i.e., fraction of the misclassified examples relative to the number of training or test examples, respectively) with the number of variables. (Bottom) The variation of the average (over 100 runs) fraction of real and bounded support vectors (relative to the number of training examples) with the number of variables.

2 important SVM genes from best 40 BW genes and 2 important genes for all three methods).

Because the variation in the errors is not significant for the three sets of variables shown in Table 1, we looked at the results of an SVM run with each of these sets of variables (all data was used as training data). The projections where the separation found by SVM can be seen are shown in Figure 5. Although we get similar accuracy for all three sets *S1,S2,S3*, there is some difference in the results. The set $S1$ has the smallest error, but $S2$ has a larger margin between the real support vectors, suggesting that $S2$ may be a better choice. By examining the coefficients of the projection that shows the best separation for $S2$, we observe that all variables contribute comparably to this projection, therefore we can not conclude that some are more important than others.

3.4 Varying SVM Input Parameters

In another set of experiments we study the dependence of the margin found by the SVM algorithm on the parameter C. We ran SVM with all the data corresponding to the set $S3 = \{800, V721, V357, V596\}$ and for each run we found a projection clearly showing the separation. Figure 6 shows the best projections

Table 1. Result summaries for three subsets of 4 variables: $S1 = \{V800, V403, V535, V596\}$, $S2 = \{V390, V389, V4, V596\}$ and $S3 = \{V800, V721, V357, V596\}$. The values are averaged over 100 runs.

Subset	Tr Err	Std Tr	Ts Err	Std Ts	RSV	Std RSVl	BSV	Std BSV
S1	0.134	0.032	0.178	0.070	0.084	0.019	0.426	0.051
S2	0.160	0.040	0.195	0.073	0.106	0.019	0.446	0.060
S3	0.166	0.032	0.217	0.063	0.092	0.016	0.426	0.049

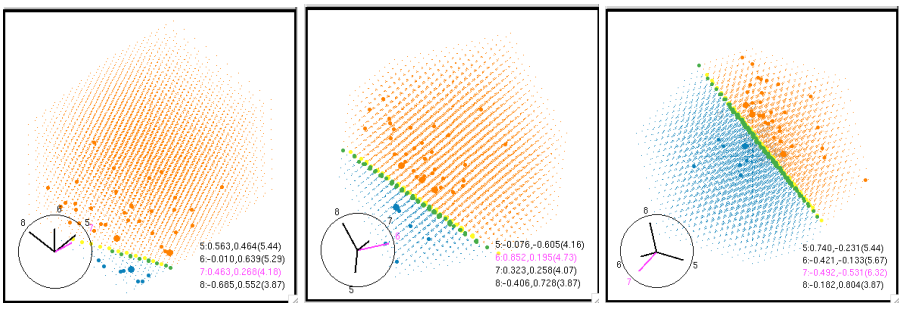


Fig. 5. Visualization of SVM results using three different subsets of the data, corresponding to three different sets of 4 variables. (Left) Gene subset $S1 = \{V800, V403, V535, V596\}$. (Middle) Gene subset $S2 = \{V390, V389, V4, V596\}$. (Right) Gene subset $S3 = \{800, V721, V357, V596\}$. Note that the subset $S2$ presents the largest separation margin, suggesting that $S2$ may be a better choice than $S1$ and $S3$.

when $C = 1$, $C = 0.7$, $C = 0.5$ and $C = 0.1$. Recall that C can be seen as the cost of making errors. Thus, the higher the C bound the less errors are allowed, corresponding to a smaller margin. As C decreases, more errors are allowed, corresponding to a larger margin. This can be seen in the plots, as you look from left ($C = 1$) to right ($C = 0.1$), the margin around the separating hyperplane increases. Which is the better solution?

Table 2 shows the variation of the training error (the proportion of misclassified examples relative to the number of training examples) with the parameter C. The values corresponding to the plots shown in Figure 6 are highlighted. It can be seen that the smallest error is obtained for $C = 1$, which corresponds to the plot with the smallest margin (or equivalently, the plot with the smallest number of bounded support vectors). However, based on the visual examination, we may choose to use the value $C = 0.1$, as it results in a larger margin and possibly in better generalization error.

3.5 Model Stability

With regard to the dependence of the separation on sampling, the separating hyperplane should not vary much from one training sample to another (we

Table 2. The dependence of the training error on the parameter C. The highlighted values correspond to the plots shown in Figure 6.

C	**1**	0.9	0.8	**0.7**	0.6	**0.5**	0.4	0.3	0.2	**0.1**
Error	**0.162**	0.148	0.148	**0.175**	0.189	**0.189**	0.175	0.202	0.202	**0.229**

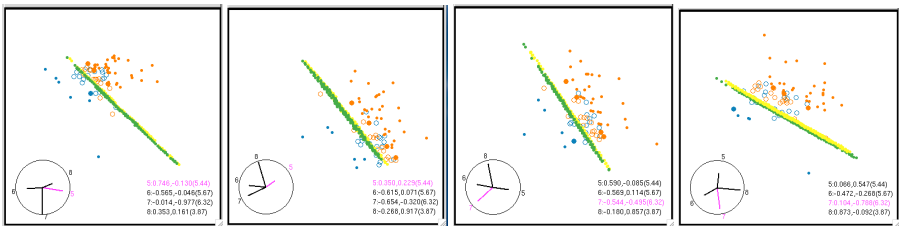


Fig. 6. Variation of the margin with the cost C. Plots corresponding to values C=1, C=0.7, C=0.5, C=0.1 are shown. As C decreases the margin increases.

might expect more variability if the data is not separable). To explore this conjecture, we ran the SVM algorithm on all examples using the variables $S3 = \{V800, V721, V357, V596\}$ and identified the bounded support vectors. Then, we removed the bounded support vectors (33 examples), obtaining a linearly separable set (containing the remaining 41). We ran SVM on samples of this set (about 9/10 points were selected for each run), found the projection showing the linear separation and kept this projection fixed for the other runs of SVM. We examined how the separation boundary between the two data sets changes. The results are shown in Figure 7. There is some variation in the separating hyperplane from sample to sample. In some samples the separating hyperplane rotated substantially from that of the first sample, as seen by the thick band of grid points.

To see how much the coefficients of the variables actually change between samples we start with the projection showing the separation for the first run, we keep this projection fixed and plot results of the second run, then slightly rotate this second view until the best projection is found for the second run. This is shown in Figure 8. The coefficients change only a tad, with those of variable 6 changing the most.

4 Summary and Discussion

4.1 Summary

We have presented visual methods that can be used in association with SVM to study many aspects of the model fitting and solution. The reason for using these methods is to gain a better understanding of the model and to better characterize the fit.

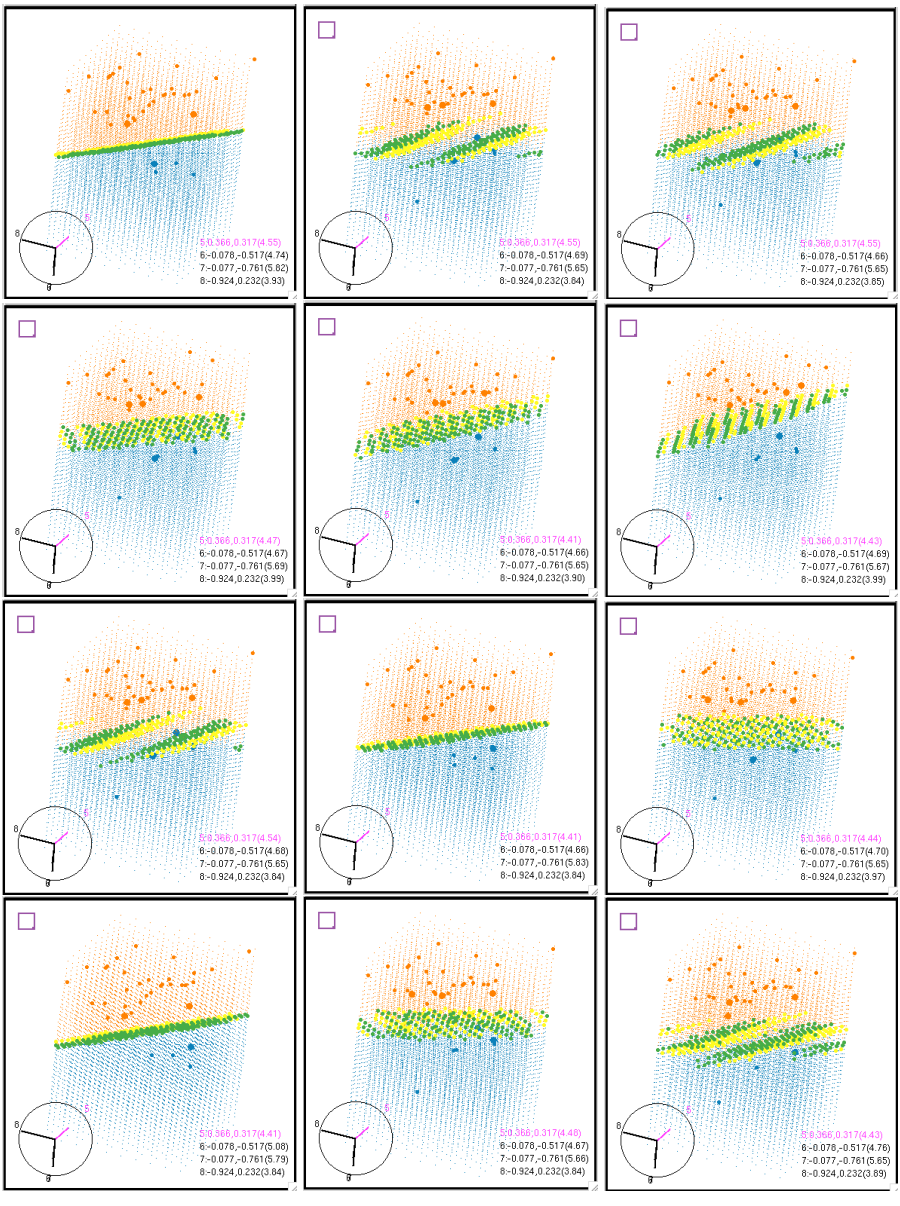


Fig. 7. We examine the variation of the separating hyperplane when sub-sampling the data. We find the projection that shows the linear separation for the first data set and we keep this projection fixed for the subsequent views. There is some variation in the separating hyperplane from sample to sample. In some samples the separating hyperplane rotated substantially from that of the first sample, as seen by the thick band of grid points.

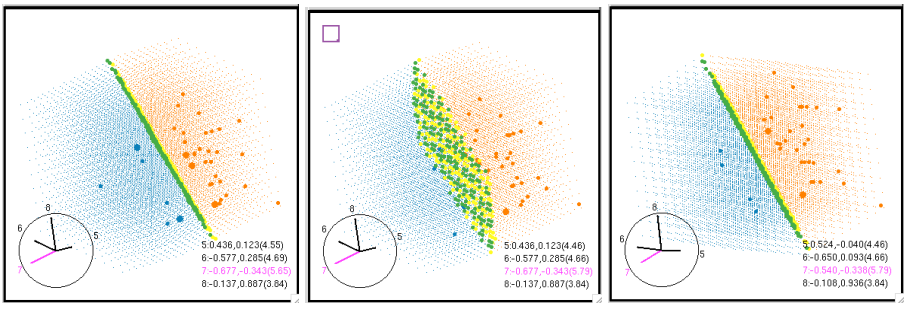


Fig. 8. Two different runs of the SVM with slightly different training data sets (9/10 points of the whole data set are selected at each run). The projection is kept fixed in the (Left) and (Middle) plots. A small rotation of the data shows the clear separation found by the second run of the SVM.

We have shown how tour methods can be used to visualize and examine the position of the support vectors found by SVM relative to the other data points and anomalies in the data. They can also be used to explore boundaries between classes in high dimensional spaces. This can be done by generating a rectangular grid around the data and computing the predicted values for each point in the grid. The values close to the boundary will have predicted values close to zero. The main hindrance to drawing the boundary is that the grid of points increases in size exponentially with the number of variables. Hence, alternative ways for showing the boundary are of interest.

We have also shown how we can use visual methods in association with variable selection methods to find sets of variables that are important with respect to the separation found by the SVM algorithm. Finally, we have shown how visual methods can be used to get insights about the stability of the model found by the algorithm and to tune the parameters of the algorithm. Therefore, these methods can be used as a complement to cross-validation methods in the training phase of the algorithm.

We have illustrated the proposed methods on a publicly available SAGE gene expression data set. The implementation of these methods will be made available to the research community as an R package.

4.2 Discussion

The importance of the visual methods in the area of knowledge discovery and data mining (KDD) is reflected by the amount of work that has combined visualization and classification methods during the last few years [35,20,1,25]. Visual methods for understanding results of several classes of classifiers have been proposed, e.g., decision tree classifiers [2,29], Bayesian classifiers [5], neural networks [36], temporal data mining [3], etc. However, there has been relatively little work on visualizing the results of SVM algorithm in high dimensional spaces, with a few notable exceptions [29,31,10,15].

Poulet [29,31] has proposed approaches to visualizing the results of SVM. Here, the quality of the separation is examined by computing the data distribution according to the distance to the separating hyperplane and displaying this distribution as a histogram. The histogram can be further linked to other visual methods such as 2-dimensional scatter plots and parallel coordinates [24] in order to perform exploratory data analysis, e.g., graphical SVM parameter tuning or dimensionality and data reduction. These methods have been implemented in a graphical data-mining environment [30]. Similar to our methods, Poulet's approaches have been applied to visualize the results of SVM algorithm applied to bio-medical data [18].

Our previous work [10] has demonstrated the synergistic use of SVM classifiers and visual methods in exploring the location of the support vectors in the data space, the SVM predicted values in relation to the explanatory variables, and the weight vectors, $\mathbf{w}$, in relation to the importance of the explanatory variables to the classification. We have also explored the use of SVM as a preprocessor for tour methods, both in terms of reducing the number of variables to enter into the tour, and in terms of reducing the number of instances to the set of support vectors (which is much smaller than the data set). Also in previous work [15], we have shown that using visual tools it is possible not only to visualize class structure in high-dimensional space, but also to use this information to tailor better classifiers for a particular problem.

References

1. Ankerst, M.: Report on the sigkdd-2002 panel the perfect data mining tool: Interactive or automated. SIGKDD Explorations 4(2) (2002)
2. Ankerst, M., Elsen, C., Ester, M., Kriegel, H.-P.: Visual classification: An interactive approach to decision tree construction. In: Proceedings of ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, San Diego, CA (1999)
3. Ankerst, M., Jones, D., Kao, A., Wang, C.: Datajewel: Tightly integrating visualization with temporal data mining. In: Proceedings of the ICDM Workshop on Visual Data Mining, Melbourne, FL (2003)
4. Asimov, D.: The Grand Tour: A Tool for Viewing Multidimensional Data. SIAM Journal of Scientific and Statistical Computing 6(1), 128–143 (1985)
5. Becker, B., Kohavi, R., Sommerfield, D.: Visualizing the simple bayesian classifier. In: Fayyad, U., Grinstein, G., Wierse, A. (eds.) Information Visualization in Data Mining and Knowledge Discovery. Morgan Kaufmann, San Francisco (2001)
6. Becquet, C., Blachon, S., Jeudy, B., Boulicaut, J., Gandrillon, O.: Strong association rule mining for large gene expression data analysis: a case study on human sage data. Genome Biology 3(12) (2002)
7. Bi, J., Bennett, K., Embrechts, M., Breneman, C., Song, M.: Dimensionality reduction via sparse support vector machines. Journal of Machine Learning Research 3 (2003)
8. Brown, M., Grundy, W., Lin, D., Christianini, N., Sugnet, C., Furey, T., Ares Jr., M., Haussler, D.: Knowledge based analysis of microarray gene expression data using support vector machines. Technical Report UCSC CRL-99-09, Computing Research Laboratory, USSC, Santa Cruz, CA. (1999)

9. Buja, A., Cook, D., Asimov, D., Hurley, C.: Computational Methods for High-Dimensional Rotations in Data Visualization. In: Rao, C.R., Wegman, E.J., Solka, J.L. (eds.) Handbook of Statistics: Data Mining and Visualization, Elsevier/North Holland (2005), http://www.elsevier.com
10. Caragea, D., Cook, D., Honavar, V.: Gaining insights into support vector machine classifiers using projection-based tour methods. In: Proceedings of the Conference on Knowledge Discovery and Data Mining (KDD), San Francisco, CA, USA (2001)
11. Caragea, D., Cook, D., Honavar, V.: Visual methods for examining support vector machines results, with applications to gene expression data analysis. Technical report, Iowa State University (2005)
12. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2001), Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
13. Cook, D., Buja, A.: Manual Controls For High-Dimensional Data Projections. Journal of Computational and Graphical Statistics 6(4), 464–480 (1997)
14. Cook, D., Buja, A., Cabrera, J., Hurley, C.: Grand Tour and Projection Pursuit. Journal of Computational and Graphical Statistics 4(3), 155–172 (1995)
15. Cook, D., Caragea, D., Honavar, V.: Visualization for classification problems, with examples using support vector machines. In: Proceedings of the COMPSTAT 2004, 16th Symposium of IASC, Prague, Czech Republic (2004)
16. Cook, D., Lee, E.-K., Buja, A., Wickham, H.: Grand Tours, Projection Pursuit Guided Tours and Manual Controls. In: Handbook of Data Visualization. Springer, New York (2006)
17. Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D., Weingessel, A.: e1071: Misc Functions of the Department of Statistics, TU Wien (2006), http://www.r-project.org
18. Do, T.-N., Poul, F.: Incremental SVM and visualization tools for bio-medical data mining. In: Proceedings of the European Workshop on Data Mining and Text Mining for Bioinformatics (2003)
19. Dudoit, S., Fridlyand, J., Speed, T.: Comparison of discrimination methods for the classification of tumors using gene expression data. Journal of the American Statistical Society 97(1) (2002)
20. Fayyad, U., Grinstein, G., Wierse, A.: Information Visualization in Data Mining and Knowledge Discovery. Morgan Kaufmann, San Francisco (2001)
21. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3 (2003)
22. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning 46, 389–422 (2002)
23. Hastie, T., Tibshirani, R., Buja, A.: Flexible discriminant analysis by optimal scoring. Journal of the American Statistical Association 89(428), 1255–1270 (1994)
24. Inselberg, A., Avidan, T.: The automated multidimensional detective. In: Proceedings of Infovis 1999, pp. 112–119 (1999)
25. Keim, D., Sips, M., Ankerst, M.: Visual data mining. In: Johnson, C., Hansen, C. (eds.) The Visualization Handbook. Academic Press, London (2005)
26. Lee, E.-K., Cook, D., Klinke, S., Lumley, T.: Projection pursuit for exploratory supervised classification. Technical Report 2004-06, Iowa State University (2004)
27. Liu, H., Li, J., Wong, L.: A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns. Genome Informatics 13 (2002)
28. Ng, R.T., Sander, J., Sleumer, M.C.: Hierarchical cluster analysis of SAGE data for cancer profiling. In: BIOKDD, pp. 65–72 (2001)

29. Poulet, F.: Cooperation between automatic algorithms, interactive algorithms and visualization tools for visual data mining. In: Proceedings of VDM@ECML/PKDD 2002, the 2nd Int. Workshop on Visual Data Mining, Helsinki, Finland (2002)
30. Poulet, F.: Full view: A visual data mining environment. International Journal of Image and Graphics 2(1), 127–143 (2002)
31. Poulet, F.: Svm and graphical algorithms: a cooperative approach. In: Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM 2004) (2004)
32. R Development Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2006); ISBN 3-900051-07-0
33. Rakotomamonjy, A.: Variable selection using svm-based criteria. Journal of Machine Learning Research 3 (2003)
34. Ripley, B.: Pattern recongnition and neural networks. Cambridge University Press, Cambridge (1996)
35. Soukup, T., Davidson, I.: Visual Data Mining: Techniques and Tools for Data Visualization and Mining. John Wiley and Sons, Inc., Chichester (2002)
36. Streeter, M.J., Ward, M.O., Alvarez, S.A.: NVIS: An interactive visualization tool for neural networks. In: Visual Data Exploration and Analysis VII, San Jose, CA, vol. 4302, pp. 234–241 (2001)
37. Swayne, D.F., Temple Lang, D., Buja, A., Cook, D.: GGobi: Evolving from XGobi into an Extensible Framework for Interactive Data Visualization. Computational Statistics & Data Analysis 43, 423–444 (2003), `http://www.ggobi.org`
38. Temple Lang, D., Swayne, D., Wickham, H., Lawrence, M.: rggobi: An Interface between R and GGobi (2006), `http://www.r-project.org`
39. Vapnik, V.: The Nature of Statistical Learning Theory (Statistics for Engineering and Information Science). Springer, New York (1999)
40. Velculescu, V., Zhang, L., Vogelstein, B., Kinzler, K.: Serial analysis of gene expression. Science 270, 484–487 (1995)
41. Wegman, E.J.: The Grand Tour in k-Dimensions. Technical Report 68, Center for Computational Statistics, George Mason University, (1991)
42. Wegman, E.J., Carr, D.B.: Statistical Graphics and Visualization. In: Rao, C.R. (ed.) Handbook of Statistics, pp. 857–958. Elsevier Science Publishers, Amsterdam (1993)
43. Wickham, H.: classifly: Classify and Explore a Data Set (2006), `http://www.r-project.org`
44. Xing, E.P., Jordan, M.I., Karp, R.M.: Feature selection for high-dimensional genomic microarray data. In: Proceedings of the Eighteenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco (2001)
45. Zhang, L., Zhou, W., Velculescu, V.E.,, S.E.K., Hruban, R.H., Hamilton, S.R., Vogelstein, B., Kinzler, K.W.: Gene expression profiles in normal and cancer cells. Science 276(5316), 1268–1272 (1997)

Author Index