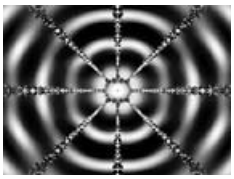
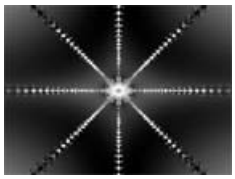


VDM@ECML/PKDD2002



Proceedings
International Workshop on
Visual Data Mining

19th August, 2002, Helsinki, Finland

Edited by
Simeon J. Simoff, Monique Noirhomme-Fraiture and
Michael H. Böhlen

in conjunction with 13th European Conference on Machine Learning (ECML'02) and 6th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'02), 19-23 August 2002, Helsinki, Finland

Proceedings of the 2nd International Workshop on Visual Data Mining - VDM@ECML/PKDD'2002, in conjunction with 13th European Conference on Machine Learning (ECML'02) and 6th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'02), 19-23 August 2002, Helsinki, Finland
S. J. Simoff, M. Noirhomme-Fraiture and M. H. Böhlen (eds)

Workshop Web Site:
http://www-staff.it.uts.edu.au/~simeon/vdm_pkdd2002/

Foreword

Visual data mining is a collection of interactive methods for knowledge discovery from data, which integrates human perceptual capabilities to spot patterns, trends, relationships and exceptions with the capabilities of the modern digital computing to characterise data structures and display data. The underlying technology builds on visual and analytical processes developed in various disciplines including data mining, information visualisation, and statistical learning from data with algorithmic extensions that handle very large, multidimensional, multivariate data sets. The growing research and development in visual data mining offers machine learning and data mining communities complementary means of analysis that can assist in uncovering patterns and trends that are likely to be missed with other non-visual methods. Consequently, the machine learning and data mining communities have recognised the significance of this area. The first edition of the workshop took place at the ECML/PKDD conference in Freiburg, following a similar workshop at the ACM KDD conference in San Francisco.

The first edition of the workshop offered to the ECML/PKDD2001 participants a mixture of presentations on state-of-art methods and techniques, with controversial research issues and applications. A report about this workshop has been published in SIGKDD Explorations 3 (2), pp. 78-81. The presentations were grouped in three streams: Methodologies for Visual Data Mining; Applications of Visual Data Mining; and Support for Visual Data Mining. The workshop included also two invited presentations – one from Erik Granum, the head of the Laboratory of Computer Vision and Media Technology, Aalborg University (on the behalf of the 3DVDM group), who presented an overview of the unique interdisciplinary 3DVDM group, its current projects and research opportunities there; and one from Monique Noirhomme-Fraiture, who demonstrated 2D and 3D visualisation support for visual data mining of symbolic data. The workshop brought together a number of cross-disciplinary researchers, who were pleased with the event and there was consensus about the necessity of turning it into an annual meeting, where researchers, both from the academia and industry can exchange and compare both relatively mature and green house theories, methodologies, algorithms and frameworks in the emerging field of visual data mining. Meantime

a discussion list on visual data mining (vdm@it.uts.edu.au) was setup to facilitate this exchange.

This workshop has been initiated and organised in response to this interest. Being a second edition, the workshop this year is aiming to create a stimulating atmosphere for open discussions of the cross-disciplinary theoretical foundations, frameworks, interactive methods, algorithms and environments for visual data processing and analysis; novel applications and utilisation of other perceptual channels. Consequently, the papers selected for presentation at the workshop are grouped in the following sessions: Visual Data Pre-processing; Visual Environments for Data Mining and Analysis; Interactive Visual Data Mining Algorithms and Applications; and 'Perceptual' Data Mining. The works selected for presentation at this workshop form more cohesive body of work, which indicates that the field has made a step forward towards achieving some level of maturity.

We would like to thank all those who submitted their work to the workshop. As part of the ECML/PKDD joint conference series, the workshop follows a rigid blind peer-review process. All papers were extensively reviewed by at least three referees drawn from the program committee. Special thanks go to them. Once again, we would like to thank all those, who supported this year's efforts on all stages – from the development and submission of the workshop proposal to the preparation of the final program and proceedings.

Simeon J. Simoff
Monique Noirhomme-Fraiture
Michael H. Böhlen

Workshop co-Chairs

July 2002

Workshop Chairs

Simeon J. Simoff

Monique Noirhomme-Fraiture

Michael H. Böhlen

Program Committee

Michael Ankerst	Boeing, USA
James L. Alty	Loughborough University, UK
Katy Börner	Indiana University, USA
Alberto Del Bimbo	Universitá degli Studi di Firenze, Italy
Maria F. Costabile	Universita' di Bari, Italy
Alex Duffy	University of Strathclyde, UK
Erik Granum	Aalborg University, Denmark
Markus Hegland	Australian National University, Australia
Maolin Huang	University of Technology Sydney, Australia
Alfred Inselberg	Multidimensional Graph Ltd, Israel
Carlo Lauro	University of Naples, Italy
Donato Malerba	Università degli Studi, Italy
Carl H. Smith	University of Maryland, USA
Michael Schroeder	City University, UK
Bruce Thomas	University of South Australia, Australia

Program for VDM@ECML/PKDD2002 Workshop

Monday, 19 August 2002, Helsinki, Finland

9:00 - 9:10 Opening and Welcome

9:10 - 10:30 Session 1 – Visual Data Pre-processing

- 09:10 - 09:50 CLUSTERING BY ORDERING DENSITY-BASED SUBSPACES
  Kan Liu, Dongru Zhou, Xiaozheng Zhou
- 09:50 - 10:30 CAN HIERARCHICAL CLUSTERING IMPROVE THE EFFICIENCY OF NON-LINEAR DIMENSION REDUCTION WITH SPRING EMBEDDING
  Michael Schroeder and George Katopodis

10:30 - 11:00 Coffee break

11:00 - 13:00 Session 2 - Visual Environments for Data Mining and Analysis

- 11:00 - 11:40 A VISUAL DATA MINING ENVIRONMENT
  Stephen Kimani, Tiziana Catarci and Giuseppe Santucci
- 11:40 - 12:20 A POST-PROCESSING ENVIRONMENT FOR BROWSING LARGE SETS OF ASSOCIATION RULES
  Alipio Jorge, João Poças and Paulo Azevedo
- 12:20 - 13:00 VISUAL POST-ANALYSIS OF ASSOCIATION RULES
  Dario Bruzzese and Cristina Davino

13:00 - 14:30 Lunch

14:30 - 16:00 Session 3 – Interactive Visual Data Mining Algorithms and Applications

- 14:30 - 15:00 COOPERATION BETWEEN AUTOMATIC ALGORITHMS, INTERACTIVE ALGORITHMS AND VISUALIZATION TOOLS FOR VISUAL DATA MINING
  François Poulet
- 15:00 - 15:30 DEFINING LIKE-MINDED AGENTS WITH THE AID OF VISUALIZATION
  Penny Noy and Michael Schroeder
- 15:30 - 16:00 VISUAL DATA MINING OF CLINICAL DATABASES: AN APPLICATION TO THE HEMODIALYTIC TREATMENT BASED ON 3D INTERACTIVE BAR CHARTS
  Luca Chittaro, Carlo Combi and Giampaolo Trapasso

16:00 - 16:30 Coffee break

16:30 - 17:30 Session 4 – 'Perceptual' Data Mining

- 16:30 - 17:00 SONIFICATION OF TIME DEPENDENT DATA
  Monique Noirhomme-Fraiture, Olivier Schöller, Christophe Demoulin, Simeon J. Simoff
- 17:00 - 17:30 A SURPRISE FROM THE 3DVDM GROUP
  Michael H. Böhlen

17:30 - 18:00 Discussion "Where Visual Data Mining is Heading" and Closure

Table of Contents

Clustering by Ordering Density-Based Subspaces

Kan LIU*, Dongru ZHOU, Xiaozheng ZHOU

School of Computer, Wuhan University
430072 Wuhan, China
*lk2000@public.wh.hb.cn

Abstract. Finding clusters on the basis of density distribution is a traditional approach to discover clusters with arbitrary shape. Some density-based clustering algorithms such as DBSCAN, OPTICS, DENCLUE, and CLIQUE etc have been explored in recent researches. This paper presents a new approach which is based on the ordered subspace to find clusters. The key idea is to sort the subspaces according to their density, and set a new cluster for the maximal subspace of the subspace list. Since the number of the subspaces is much less than that of the data, very large databases with high-dimensional data sets can be processed with high efficiency. We also present a new method to project high-dimensional data, and then some results of clustering with visualization are demonstrated in this paper.

Keywords: Clustering, Density-based, Data visualization

1 Introduction

The process of grouping the physical data or abstract data based on their similarity is called clustering. Clustering is an important analysis method in data mining, which could help people to better understand and observe the natural classification or structure of the data. Clustering algorithms are used to automatically classify data items into the relative, meaningful clusters. After clustering, the items within any cluster are highly relevant and the items across different clusters are lowly relevant. The factors listed below are always being considered when evaluating a clustering algorithm.

- Scalability: A good clustering algorithm can deal with large datasets including up to millions data items.
- Discovery of clusters with arbitrary shape: A cluster may have an arbitrary shape. A clustering algorithm should not only apply to the regular clusters.
- Minimum parameters input: It is a heavy burden for the users to input those important parameters. In the meantime, this brings more trouble in getting good quality clustering.

- Insensitive to order of input records: Inputting data in different order should not lead to different results.
- High-dimensionality: A dataset may include many attributes, clustering data in high-dimensional space is highly demanded in many applications.

Current clustering algorithms can be broadly classified into two categories: hierarchical and partitional. Hierarchical algorithms, such as BIRCH [7], CURE [8] and CHAMELEON [9] etc, decompose a dataset into several levels of nested partitions. They start by placing each object in its own cluster and then merge these atomic clusters into larger and larger clusters until all objects are in a single cluster. Or they reverse the process by starting with all objects in a cluster and subdividing into smaller pieces. Partitional algorithms, such as CLARA [5] and CLARANS [6], partition the objects based on a clustering criterion. The popular methods, K-means and K-medoid, use the cluster average, or the closest object to the cluster center, to represent a cluster.

New clustering algorithms have been proposed in recent researches [4]. For example, DBSCAN, OPTICS and CLIQUE are based on density; STING, WAVE CLUSTER are based on grid; and COBWEB, SOM are based on model.

2 Related Work

The main idea of density-based approaches is to find regions of high-density and low-density, with high-density regions being separated from low-density regions. These approaches can make it easy to discover arbitrary clusters. A common way is to divide the high-dimensional space into density-based grid units. Units containing relatively high densities are the cluster centers and the boundaries between clusters fall in the regions of low-density units. For example, the CLIQUE [1] algorithm processes dense units level-by-level. It first determines 1-dimensional dense units by making a pass over the data. Having determined (k-1)-dimensional dense units, the candidates for k-dimensional units are determined. A cluster is a maximal set of connected dense units in k-dimensions. This algorithm automatically finds subspaces of the highest dimensionality such that high-density clusters exist in those subspaces, but the accuracy of the clustering result may be degraded at the expense of simplicity of the method.

The alternative way is to calculate parameter ‘directly density-reachable’ or ‘reachability-distance’ of the object. For example, the DBSCAN [3] aims at discovering clusters of arbitrary shape based on the formal notion of density-reachability for k-dimensional points. OPTICS [2] solves the problem of DBSCAN, which only computes a single level clustering. OPTICS produces a special order of the database with respect to its density-based clustering structure, and according to the order of the reachability-distance of each object, it can quickly reach the high-density region. OPTICS is good for both automatic and interactive cluster analysis, including finding intrinsic clustering structure, and is not limited to one global parameter setting. But it is infeasible to apply it in its current form to a database containing several million high-dimensional objects. In this paper, we propose an integrative

clustering method which is to partition the data space based on the density and grid-based techniques and realize the visualization of the clustered result.

3 Clustering by Ordering Dense Units

3.1 Basic statement

The density-based technique is adopted in our approach, in which the data space is partitioned into non-overlapping rectangular units and the density distribution will be deduced by calculating the data volume of each rectangular unit.

Suppose D is a n-dimensional dataset with m items: $D=\{X_1, X_2, \ldots, X_i, \ldots, X_m\}$, where $X_i=(X_{i1}, X_{i2}, \ldots, X_{ij}, \ldots, X_{in})$, ($i \leq m$), and $X_{ij}$is the value of the j th attribute of X_i. If each dimension of the dataset is equally divided into t parts, then all the items in the dataset fall into k units: $U=\{U_1, U_2, \ldots, U_i, \ldots, U_k\}$ ($k \leq t^n$), where $U_i=(U_{i1}, U_{i2}, \ldots, U_{ij}, \ldots, U_{in})$ is the vector of each equally divided attribute. Two units U_1 and U_2 are defined to be adjacent only when any attribute of one unit is adjacent to that of the other unit: $| U_{1j} - U_{2j} | = 1$, $U_{1s} = U_{2s}$, ($j, s \leq k, j \neq s$). We define density peaks as those units whose densities are larger than those of the adjacent units; similarly, we define density valleys as the units whose densities are lower than those of the adjacent ones.

3.2 Algorithm

When high-dimensional space is divided into k equal subspaces (units), the density peaks are regarded as the clusters centers. So the key process is how to find the density peaks. In our approach, CODBU (Clustering by Ordering Density-Based Units), the units with densities greater than threshold are ranked by the density value. The change from density peaks to density valleys is expressed by hierarchical level. The density peak is positioned in the first level of the cluster, the adjacent units are in the second level, and finally, the density valley is positioned in the last level of the cluster. First, we calculate the density values of each unit, and then rank the units whose densities are greater than threshold value. The largest-density unit will be analysed first. Each unit is compared with its adjacent units (neighbours) in density, if its density is greater than that of any other adjacent unit, it is considered as a density peak and then be set as the first level, a new cluster is emerging. If its density is less than one of adjacent units, then this unit will be grouped into the cluster of the adjacent unit; if its value is less than many of adjacent units, then this unit will be grouped into the cluster of the lowest-level unit. Fig.1 describes the process of cluster, in which only 2 basic parameters are required: the number of subdivisions for each dimension and the density threshold value. These two values are entered manually according to the size of the dataset and the required accuracy of the result.

```
CODBU (MinDen, t)
BEGIN
    int cluster_no = 0;
    int k = 0;
    Divide each attribute into t equal parts, initialize U;
    Read data x from dataset
    If (x ∈ Uj)  Uj.density++;
    For all Uj
            if (Uj.density >= MinDen){
            U.addElement(Uj);
            Uj.cluster_no=0;
            Uj.layer_no=0;
            k++; }
      Quicksort ( U );                        // sort in the descending order
      for (j=0; j<k; j++){
        for all neighbors of Uj
          if (neighbor.density > Uj.density)
            if (Uj.layer_no = 0) {          // group into the high-density unit cluster
              Uj.cluster_no = neighbor.cluster_no;
              Uj.layer_no = neighbor.layer_no+1;}
            else                            // group into the low-level unit cluster
              if (Uj.layer_no > neighbor.layer_no+1){
                Uj.cluster_no=neighbor.cluster_no;
                Uj.layer_no=neighbor.layer_no+1;}
        if (Uj.layer_no=0){                 // form a new cluster
          cluster_no++;
          Uj.cluster_no=cluster_no;
          Uj.layer_no=1;}
    }
END.
```

Fig.1 CODBU: Clustering by Ordering Density-Based Units

Figure 2 shows a simple 2-dimension data set. The sequence number of each unit is shown in the unit, and ' * ' stands for the spread points among them. Now sort all the squares whose density values are larger than 3, the result is (sorted by density): 4 (11), 5 (9), 8 (8), 10 (8), 1 (7), 9 (7), 11 (7), 3 (6), 6 (5), 12 (5), 7 (4), 2 (3). Based on the above result, we can find 2 clusters (sorted by level):

C1={4 (1), 5 (2), 1 (2), 10 (2), 3 (2), 6 (3), 11 (3)};

C2={8 (1), 9 (2), 12 (2), 7 (2), 2 (2)}.

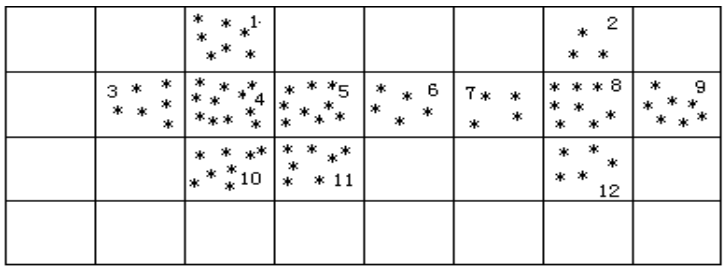


Fig.2 Two clusters based on the density values of units

We can see that, although the density of unit 5 is larger than that of unit 8, as it is adjacent to unit 4 which has higher density, unit 5 is still grouped into the first cluster. Since the density value of unit 8 is greater than that of any adjacent unit, it forms a new cluster. The density of unit 7 is lower than those of unit 6 and 8, but the unit 8 has a lower level than unit 6, so unit 7 is grouped into the second cluster.

3.3 Algorithm analysis

The dataset is only scanned once in our approach. Suppose k is the number of subspaces whose densities are greater than threshold value, the time complexity of applying quick sort is O (k*log k). By building up a search tree, the time complexity of comparing the density value of each unit with those of its adjacent units is O(nk), and n is the number of dimensions. Since the total number of units is much less than that of the data items in the dataset, the time complexity is decreased dramatically. Furthermore, the analysis of the data space based on the density order can better reflect various clusters than the traditional approach in which the data items are simply grouped together if the densities are greater than a set threshold. Therefore our approach can cluster high-dimensional data space more quickly and accurately. In addition, the quality of the clustered result will not be influenced by the shape of the high-dimensional clusters or the order of the data input, and the parameters are easily set up and modified, so all the criteria mentioned in the Section 1 have been met.

4 Visualization

Clustering high-dimensional datasets is used to help users to better understand the data structure and relationships. Visualization techniques play a very important role in displaying data and clustered results, making it more clear and reliable. The visualization techniques for high-dimensional dataset [10] [11] can be divided into two types. One type, such as “parallel coordinates” [12], is to divide the 2-d plane into several parts, each part representing an attribute. The other type is to reduce the dimensions, which is implemented through giving weights to the attributes of n-dimensional data according to the relative importance and then combine them linearly. We present a new method to project high-dimensional data, which uses stimulation spectrum to project high-dimensional data on a 3-d space.

The natural color is the summation of energy distribution in each wavelength in the range of visible spectrum. This energy distribution is called stimulation spectrum $\Phi(\lambda)$. Every stimulation spectrum can be transferred to a point in RGB color space. The quantitative relationship between stimulation spectrum $\Phi(\lambda)$ and RGB color coordinate are listed in the following formula:

$$R = k * \sum_{\lambda} \Phi(\lambda) * r(\lambda) * \Delta\lambda$$
$$G = k * \sum_{\lambda} \Phi(\lambda) * g(\lambda) * \Delta\lambda \quad (4.1)$$
$$B = k * \sum_{\lambda} \Phi(\lambda) * b(\lambda) * \Delta\lambda$$

In this formula, k is the ratio. λ refers to wavelength of visible spectrum, ranging from 400_{nm} to 700_{nm}. r(λ), g(λ) and b(λ) stand for the spectrum tristimulus functions of red, green and blue, and the value of r(λ), g(λ) and b(λ) at every 5nm is measured by CIE, which is already known. Fig.3 is the spectrum tristimulus functions graph of CIE 1931 standard colorimetric system.

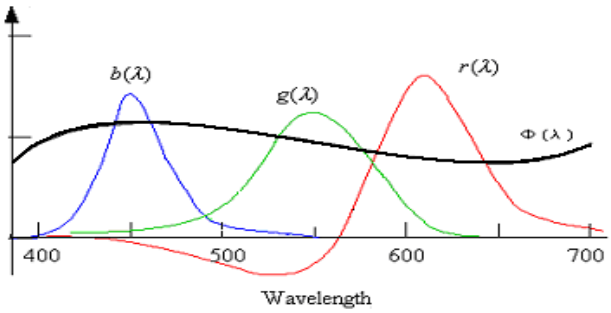


Fig. 3 Spectrum tristimulus functions graph

Each data item $X_i = (X_{i1}, X_{i2}, \ldots, X_{ij}, \ldots, X_{in})$ in high-dimensional space can be viewed as a stimulation spectrum, and spreads equally in the range of visible spectrum with the wavelength from 400_{nm} to 700_{nm}. Here X_i can be regarded as the function of λ, and the change of the attribute values corresponds to that of the spectrum tristimulus. For example, $X_i(400) = X_{i1}$, …, $X_i(700) = X_{in}$. We can work out the 3-d coordinate of data X_i in projection space according to formula (4.2), and through adjusting the value of k can make projection space not only the RGB color space, but also any 3-d space**.**

$$x = k * \sum_{\lambda} \mathrm{Xi}(\lambda) * r(\lambda) * \Delta\lambda$$
$$y = k * \sum_{\lambda} \mathrm{Xi}(\lambda) * g(\lambda) * \Delta\lambda \quad (4.2)$$
$$z = k * \sum_{\lambda} \mathrm{Xi}(\lambda) * b(\lambda) * \Delta\lambda$$

We can see from the above that the process of projecting the data items as stimulation spectrums can also be viewed as a kind of weight linear combination of n-dimensional data through spectrum tristimulus functions r(λ), g(λ) and b(λ). Taking advantage of spectrum tristimulus functions to convert high-dimensional data can completely project the data in projection space. This is because the fundamental

function of r(λ), g(λ) and b(λ) is to project stimulation spectrums in color space, so it can well reflect the feature of the original data. From fig.3, we can see that the n attributes can be divided into 3 parts, and b(λ) corresponds to the attributes of the former part of the data while g(λ) and r(λ) mainly correspond to the middle and the last part of the data attributes. In this way, the projection result of a data item will be described by all of the 3 coordinate values. On the other hand, because the spectrum tristimulus functions cover the equal area, the ranges of the coordinate axes in projection space are equal, and the data will not be over-concentrated around some coordinate axes.

5 Experiments

A 6-dimensional dataset containing 400 points was used in our CODBU testing experiment (it is a car dataset from http://stat.cmu.edu/datasets/). The attributes in the data set are: fuel economy in miles per U.S. gallon, number of cylinders in the engine, engine displacement in cubic inches, output of the engine in horsepower, 0 to 60 mph acceleration, and vehicle weight in U.S. pounds. Each attribute was divided into 5 parts, so there were 5^6=15,625 units and the 400 points scattered in 53 units. The threshold was set up as 1 which means all the points were processed. 7 clusters were obtained through linking the associated units.

Two visualization techniques were explored in displaying the result: parallel coordinates and the spectrum tristimulus functions projection. Fig.4 shows the result using parallel coordinates, in which different clusters are represented by different colors but the characteristics of the clusters are not obvious.

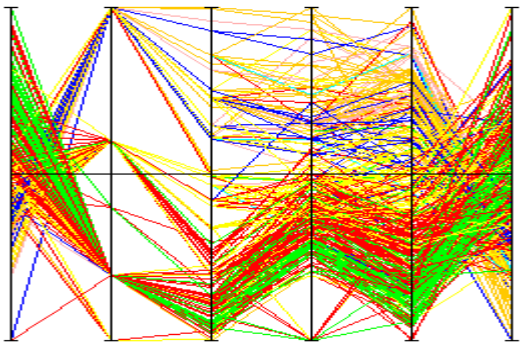


Fig. 4 Visualization of the clusters using parallel coordinates

Fig.5 shows the result using the spectrum tristimulus functions projection. The side length of the projection space is 200, Δλ=5nm, and r(λ), g(λ), b(λ) are given by CIE. In fig.5 (a), the values of the densities are reflected by color: the darker the color, the higher the density. 3 clusters are found. Fig.5 (b) displays the result of clustering using our algorithm; clusters are represented by different colors. 7 clusters are obtained based on the previous 3 big clusters. This is because that there are 7 density peaks being discovered. The clusters in white are removed due to only one data point

included. Fig.5 (c) uses symbols (such as *, +, o, ^, etc.) instead of colors to display the clusters of the data.

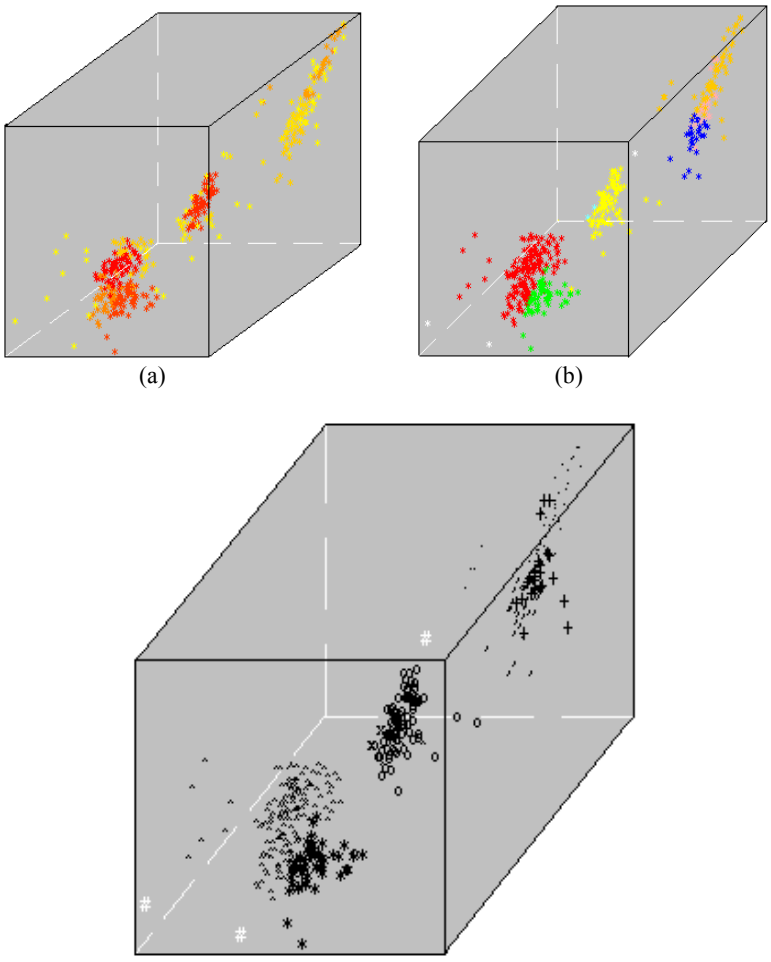


Fig. 5 Visualization of the clusters using the spectrum tristimulus functions projection. Shades in (a) reflect the different density, and different colors in (b) represent the different clusters, while in (c) the different clusters are represented by the symbols instead of the colors.

6 Conclusions

The paper presents a new clustering approach by sorting density-based units. The basic idea is to rank the units in high-dimensional data space according to the values of the density, and start from the highest density unit to search for the density peaks in order to discover clusters. The experimental results are very promising. Clusters extend from the density peaks to density valleys and this will not be affected by the

shape of data items. Arbitrary clusters can be obtained through our approach. We also propose the method of projecting high-dimensional data on the basis of the spectrum tristimulus functions, by which the data and its distribution can be displayed in the 3-dimensional space. The combination of the data mining and the visualization techniques makes it easier for users to observe and understand data, and make better use of the data to do prediction and decision.

References

1. R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan: *Automatic Subspace Clustering of High dimensional Data for Data Mining Applications.* Proc. ACM SIGMOD'98 Int. Conf. on Management of Data, Seattle, WA (1998) .94~105
2. M. Ankerst, M.M. Breunig, H. Kriegel, J. Sander: *OPTICS: Ordering Points To Identify the Clustreing Structure.* Proc. ACM SIGMOD'99 Int. Conf. on Management of Data, Philadelphia, PA (1999)
3. M. Ester, H.P. Kriegel, J. Sander, X. Xu: *A density-based algorithm for discovering clusters in large spatial databases.* Proc. 1996 Int. Conf. Knowledge Discovery and Data Mining (KDD'96), Portland, OR, Aug (1996) 226-231
4. J.W. Han, M. Damber: *Data Mining: Concepts and Techniques.* Morgan Kaufmann Publishers (2001)
5. L. Kaufman, P.J. Rousseeuw: *Finding Groups in Data: An Introduction to Cluster Analysis*. New York, John Wiley & Sons (1990)
6. R. Ng, J. Han: *Efficient and effective clustering method for spatial data mining*. Proc. Int. Conf. Very Large Data Bases (VLDB'94) (1994) 144-155
7. T. Zhang, R. Ramakrishnan, M. Livny: *BIRCH: An efficient data clustering method for very large databases*. Proc. ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'96), Montreal, Canada, June (1996) 103-114
8. S. Guha, R. Rastogi, K. Shim: *Cure: An efficient clustering algorithm for large databases*. Proc. ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'98), Seattle, WA, June (1998) 73-84
9. G. Karypis, E. H. Han, V. Kumar: CHAMELEON: *A hierarchical clustering algorithm using dynamic modeling*. COMPUTER, 32 (1999) 68-75
10. I. Herman, G. MelancËon, M.S. Marshall: *Graph Visualization and Navigation in Information Visualization: A Survey*. IEEE Transactions on Visualization and Computer Graphics, Vol.6, No.1, January-March (2000) .24~43
11. A. Buja, D. Cook, D. Swayne: *Interactive High-Dimensional Data Visualization*. Journal of Computational and Graphical Statistics, No.5 (1996) 78-99.
12. H. Siirtola: *Direct Manipulation of Parallel Coordinates.* Proc. of the IEEE International Conference on Information Visualization (IV2000), London (2000) 373-378

Can hierarchical clustering improve the efficiency of non-linear dimension reduction with spring embedding?

Michael Schroeder and George Katopodis
Department of Computing, City University
Northampton Square, London EC1V 0HB, UK
{msch,dc707}@soi.city.ac.uk

July 4, 2002

Abstract

In visual datamining proximity data, which encodes the relationship between some entities as a distance, is often available. This proximity data is inherently high-dimensional and can be mapped into a low-dimensional 2D or 3D target space such that the points in the target space adhere to the specified distances. The target space can then be visualised as scatterplot.

Force-directed graph drawing such as spring embedding can be used for this purpose and produce non-linear mappings. Spring embedding starts with a random layout, which is continually improved. This process is time-consuming since the basic algorithms perform in $O(n^3)$, where n is the number of entities.

In this paper, we consider hierarchical clustering, which performs in $O(n^2)$ as pre-processing for spring embedding. The clustering information is used to generate an approximate initial layout for the spring embedding algorithm. We compare this clustering-based initial layout with a number of others regarding their complexity, speed-up, and convergence behaviour.

Keywords: Dimension reduction, multi-dimensional scaling, force-directed graph drawing, spring embedding, hierarchical clustering

1 Introduction

Recently there has been increased interest in visual datamining, which marries the three areas of information visualisation, datamining, and human-computer interaction. Visual datamining is a process in which data is analysed, processed, and then visualised. Often the data defines pair-wise distances between the entities under consideration. For such proximity data, there are a number of analysis techniques, such as hierarchical clustering and multi-dimensional scaling. Hierarchical clustering [Gor81, Eve78, Kru77, DEKM98, Web99] aims to cluster similar objects together. As the clustering is hierarchical, this processing technique is naturally visualized in the next phase as a tree. Multi-dimensional scaling (see e.g. [Gor81, Eve78, Kru77, MKB79, Web99]), on the other hand, maps inherently high-dimensional proximity data into a 2D or 3D target space, such that distances between the entities in the target space reflect their distance as given in the proximity data. The target space can then be visualised as a scatterplot.

Proximity data can be seen as a label, fully connected, undirected graph and as a result graph drawing techniques can be useful for multi-dimensional scaling. One such graph drawing method is spring embedding [Kru64, QB79, KK89, FR91,

AAH94, RD96, Tun99, BETT99]. Spring embedding is a local optimisation technique, which starts with an initial (usually random) layout and then iteratively improves this layout by viewing edges between nodes as springs, thus leading to attractive and repulsive forces based on the desired distance between the nodes. Nodes move in the direction of this force until a state of minimal energy is reached.

In this paper, we investigate the hypothesis whether hierarchical clustering can used as pre-processing to improve the efficiency and quality of spring embedding. The paper is orgranised as follows: First, we give a motivating example from biology and show dendrograms resulting from hierarchical clustering and a scatterplot resulting from spring embedding produced with our Space Explorer tool [SGvHN01]. Next, we give an overview over hierarchical clustering, multi-dimensional scaling, and spring embedding. In section 3, set the scene to test our hypothesis by introducing six approaches to an initial layout. In section 4, we carry out experiments and discuss the results. Next, we discuss in section 5 another highly important parameter for an efficient layout besides the initial layout. Finally, we conclude with some comparison and conclusions.

2 Visual datamining

Before we present a short summary of hierarchical clustering and a more detailed background of spring embedding, let us consider an example.

2.1 Motivating Example

Advances in molecular and structural biology have resulted in a great output of biological data. The advent of DNA chip technology [dIB97] allows to measure systematically the level of expression of several thousands of genes. Such gene expression can represented as table (see Fig. 1). We selected an example from an analysis of the cell cycle in yeast [ESBB98]. The authors selected 800 genes whose level of expression fluctuates periodically during the cell cycle. Each gene is characterized by a series of expression measurements taken at successive time intervals. The alpha cell experiment contains 18 successive measurements on the same cell population. We are thus dealing with multivariate analysis, with a data matrix of 800 rows (entries) and 18 columns (variables). Next we translate the multivariate data to proximity data. Many distance metrics can be used to this end.

Definition 1 *Let $x, y \in \mathcal{R}^n$. Then*

$$d_a(x,y) = acos(\frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}})$$

is called angular separation and

$$d_e(x,y) = \sqrt{\sum_{i=1}(x_i - y_i)^2}$$

os called Euclidean distance.

For this example, we will use angular separation as it is a good choice to compare similar behaviour of genes, rather than their absolute expression levels.

How can we analyse the above data? Hierarchical clustering and spring embedding are two techniques to visualuie the data as dendrograms and scatterplot respectively.

ORF	α_0	α_7	α_{14}	...
YER150W	0.41	1.47	1.8	...
YGR146C	0.78	0.37	-0.09	...
YDR461W	2.36	2.35	2.3	...
...				

$\Longrightarrow$

	150W	146C	461W	...
YER150W	0	0.3	0.6	...
YGR146C		0	0.4	...
YDR461W			0	...
...				0

Figure 1: Left: Fragment of a multivariate data table. Rows correspond to genes, and columns to experiments. The level of expression of ca. 6000 genes was measured at 28 successive time points [ESBB98]. Using for example angular separation, the multi-variate data is converted to proximity data (table on right).

	a	b	c	d
a	0	1	6	8
b		0	5	7
c			0	2
d				0

$\Longrightarrow$

	(a,b)	c	d
(a,b)	0	5	7
c		0	2
d			0

$\Longrightarrow$

	(a,b)	(c,d)
(a,b)	0	5
(c,d)		0

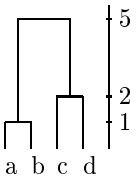


Figure 2: Hierarchical clustering using single linkage and the resulting dendrogram below.

2.2 Hierarchical Clustering

Hierarchical clustering identifies similar objects and clusters them together.

It consists in separating a set of objects into several subsets on the basis of their similarities. This problem is not simple, and various approaches have been developed to optimise the clustering in function of some criteria. The underlying idea is generally to define clusters that minimise intra-cluster variability while maximising inter-cluster distances. One particular approach is hierarchical clustering [Gor81, Eve78, Kru77, DEKM98, Web99]. Consider Figure 2. The first step is to identify the two objects with the closest relatedness. This pair builds the first cluster. The next pair is then searched, but this time a pair can be formed by joining either two objects, or an object and the cluster. The process is then repeated iteratively, by forming a cluster from the pair of closest objects/clusters, until all of them are connected forming a rooted tree. The clustering process requires to define distance not only between objects, but also between an object and a cluster or between two clusters. Several options can be used for this such as e.g. minimal, maximal or average distance (single, complete, or means/UPGMA (unweighted pair group method using arithmetic averages) [SM58] linkage respectively (see [Gor81, Eve78, Web99])). This choice has an impact on the cluster formation, and may lead to different interpretations (see e.g. [Gor81, Eve78, Kru77, Web99]). In general the above clustering algorithm runs in $O(n^2)$, where n is the number of entities. However, for the single linkage or nearest neighbour method $O(n^2)$ can be achieved [Ols95]. Applied to the above data set, one obtains the dendrogram in Fig. 3, which has ben additionally enriched by a colour map.

2.3 Spring Embedding

Before we explain how to use spring embedding as a non-linear multi-dimensional scaling technique, let us review spring embedding.

Spring embedding [Kru64, QB79, KK89, FR91, AAH94, RD96, Tun99, BETT99] uses the physical metaphor of springs (see Fig. 4). According to Hooke's law, there are attractive and repulsive forces based on the desired distance between the nodes. The Spring embedding algorithm starts with a random layout and then computes

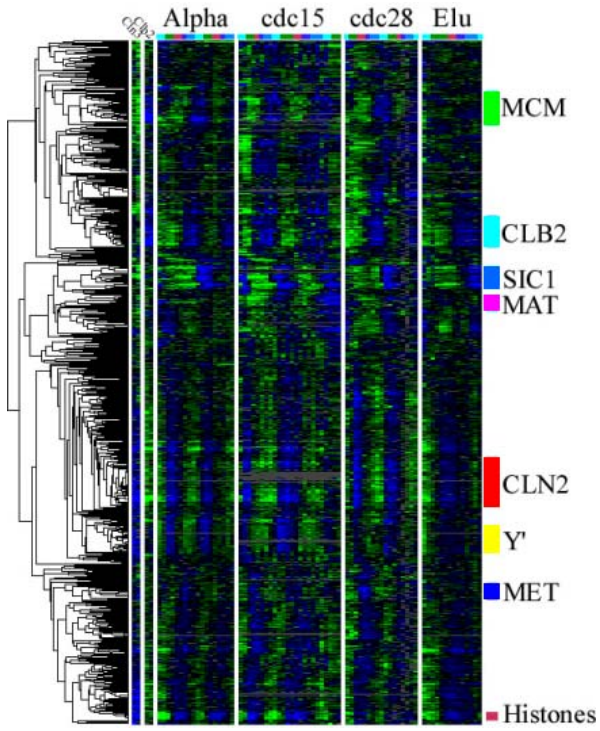


Figure 3: Dendrogram visualisation [ESBB98] for gene expression data.

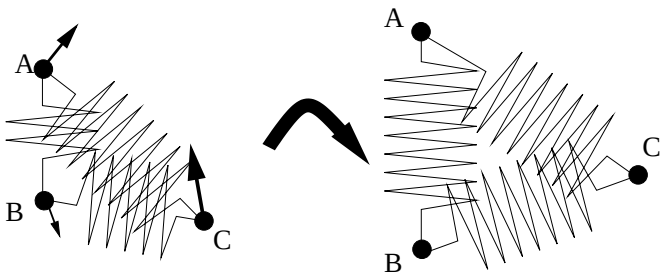


Figure 4: Multi-dimensional scaling with spring embedding. Nodes are moved into the direction of the sum of forces.

for a number of iterations the forces for each object and moves it into the direction of the overall force until a state of minimal energy is reached. Nodes are usually not moved by the full amount of the force, but are limited by a maximum amount known as temperature. In a technique known as temperature scheduling the temperature is reduced per iteration, so nodes are more and more limited in their movement.

An early example of a force directed drawing algorithm is Tutte's Barycenter method [Tut60, Tut63], in which the force acting on a node i is given as $F(i) = \sum_{i=1,i\neq j}^{n} x_i - x_j$, where x_i are the coordinates of node i. The method starts by placing a set of at least three fixed vertices in positions where spring forces cannot affect them. The rest of the vertices are free and initially located at the origin. The algorithm iterates and in each iteration a new location is computed for each free vertex.

Since then a number of different algorithms have been developed. An influential algorithm extended later in various ways is Eades' algorithm [Ead84]. Besides springs of logarithmic strength governed by Hooke's law it views nodes and non-adjacent vertices as electrically charged particles repelling each other.

Fruchterman and Reingold's FR algorithm [FR91] extends Eades' algorithm by introducing new attractive and repulsive factors as well as the above mentioned idea of temperature and temperature scheduling. Their algorithm computes all forces on all nodes, which are then moved at once. The algorithm limits the maximum displacement to a temperature, which starts at an initial value and decreases in an inverse linear fashion. They note that a better cooling schedule can dramatically change resulting layouts and reduce the number of iterations required to find the layout. To speed up their algorithm they use what they call grid-variant algorithm. In this variant the drawing area is divided into grids. The attractive forces are computed as usual, but for each vertex the repulsive force is only computed between the vertex and the nodes in its close-by cells of the grid. This method has not been very popular mainly due to its dependency on the distribution of nodes within the grid, the overhead of placing nodes in grid cells in each iteration, and the compromised layout quality that it produces in some cases. FR terminates after a maximum number of iterations is exhausted. Since different types of graphs require different number of iterations for their layouts to converge, this method of stopping may terminate too early or too late depending on the graph.

GEM [AAH94], a short form for Graph EMbedder, is a Spring algorithm based on Eades' algorithm, and is the first one that uses the history of a node's movement to choose the temperature for the current displacement of the node. Unlike FR, this algorithm moves the nodes one at a time in each iteration. The algorithm iterates until a maximum number of iterations is carried out or until the average local temperature of all nodes falls below a threshold. It also computes a likelihood that a node is oscillating or rotating and changes its temperature accordingly. GEM also starts with an incrementally computed initial layout, which is particularly benefitial for trees and grids.

Kamada and Kawai's KK algorithm [KK89] is based on graph-theoretic distances between pairs of vertices. In KK, the graph nodes are considered as particles that are all connected by springs whose ideal lengths are equal to the graph-theoretic distances between their two end-point particles multiplied by the desirable length of one edge. The goal of the algorithm is to find a balanced spring system. The major drawback of this method is its high computational cost as partial differential equations need to be solved.

Tunkelang [Tun94] has presented an incremental algorithm with three stages. In the first stage, a permutation of the nodes of a given graph is constructed from a minimal height breadth-first spanning tree of the graph. In the second stage, for each node in the order computed in the first stage, the area surrounding the already positioned neighbours of the node is sampled and examined for an approximate best

position, and the node is placed at this position. In the second stage, when the local optimisation procedure improves a vertex's position, it recursively performs the process on the already placed adjacent nodes of the vertex. Finally in the last stage, the algorithm performs the local optimisation process at every node for fine tuning.

This algorithm generates layouts that are different from those of FR, KK, and GEM [FMC96]. Specifically, it does not capture graph symmetries. In addition, the algorithm takes a quality parameter where a large value results in better quality of graph layouts. However, this quality parameter is estimated to have an exponential effect on the running-time of the algorithm [FMC96].

Tunkelang proposed a second algorithm [Tun99] based on the FR algorithm. This algorithm differs from FR in its computation of repulsive forces, in its optimization process, and in stopping the algorithm. Similar to the "grid variant" in FR, the algorithm approximates the computation of repulsive forces on a node. FR's optimization process uses force laws that in effect compute the negative gradient of an implicit objective function. However, this algorithm uses the conjugate gradient method on a non-quadratic objective function using an approximate line search. In addition, this algorithm uses the average of the square of the displacement distances of nodes for stopping the algorithm. It terminates when this value is less than 0.01.

Davidson and Harel's DH algroithm [RD96] uses simulated annealing. In each step, the layout is improved by comparing the current position of a node to one randomly selected with the neighbourhood of the node. Step by step, the temperature is reduced and the neighbourhood gets smaller. The comparison of current to randomly selected position involves weighted factors for a number of criteria such as overall stress, edge crossings, etc.

All the above algorithms reach their goals and produce aesthetically pleasing drawings. FR, KK, GEM and DH often produce similar drawings and they display symmetry. Tunkelang's first algorithm [Tun94] often yields different drawings without symmetry. DH is the most flexible but also the most time consuming while GEM and KK are very competitive in speed. Finally, FR is fast on small graphs, but slows down on graphs with more than 60 nodes [FMC96].

How can we use spring embedding to analyse the proximity data discussed earlier? I.e. how can spring embedding implement multi-dimensional scaling?

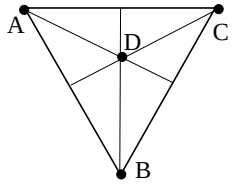
2.4 Multidimensional scaling with Spring Embedding: from distances to coordinates

Multi-dimensional scaling [Sam69, Gor81, Eve78, Kru77, MKB79, Web99] aims to find points in space which satisfy some given distances. Formally the problem can be defined as follows:

Problem: We have to define an algorithm which computes a matrix $X = (x_1, \ldots, x_n) \in \mathcal{R}^{m,n}$ for a proximity matrix $D = (d_{ij}) \in \mathcal{R}^{n,n}$ such that $d_e(x_i, x_j) = d_{ij}$, i.e. the Euclidean distance $d_e()$ between x_i and x_j is d_{ij}.

Before we show how to construct a solution for problem, let us note a limitation. Given a proximity matrix it may not be possible to find a solution at all.

Not every distance matrix can be visualized in Euclidean space. Consider e.g. the Figure below, where A, B, and C have all the same distance and D is in the middle of the triangle formed. As it stands the drawing is Euclidean (otherwise we could not draw the figure!). But now consider a distance matrix, where A, B, C are all equidistant, but the distance of A, B, and C to D is slightly less than in the drawing on the right.



Such a distance matrix satisfies all of the requirements of a metric such as triangle

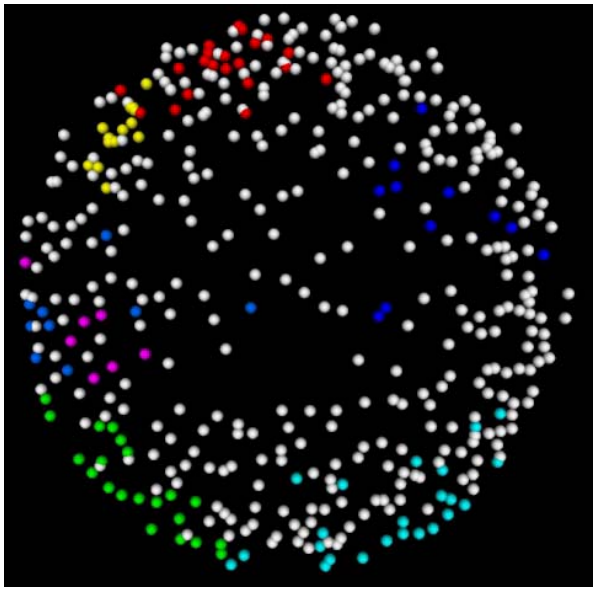


Figure 5: Scatterplot of gene expression data produced with spring embedding and anguar separation using SpaceExplorer [SGvHN01].

inequality, yet it is impossible to draw these distances in any Euclidean space.

In general, there are two approaches to multi-dimensional scaling. The mapping can be linear or non-linear. Principal coordinate analysis (PCoA) [Gow66] is an example for a linear mapping, while spring embedding is one for a non-linear mapping. How does it work?

Graph drawing techniques are applicable to proximity data, as the proximity data can be seen as specification of a complete, labeled undirected graph. There is however a difference: While graph drawing techniques as discussed above emphasise aesthetic aspects [BETT99], this is not important for proximity data as the low-dimensional drawings can rarely meet the given distances, which are inherently high-dimensional. Hence, aesthetics is not of importance, but rather to achieve a layout with minimal energy. As an example for a scatterplot of the gene expression data introdcued in 2.1 consider Fig. 5.

A typical force directed algorithm, which is used for the experiments in this paper, looks as follows:

Algorithm Given a distance matrix D.

1. Set all nodes to an initial position, i.e. $X = (x_1, \ldots, x_n) \in \mathcal{R}^{m,n}$ is populated with random values.

2. Loop: For all $1 \leq i \leq n$ compute the sum of attractive and repulsive forces $\Delta x_i = F_i^{attr} + F_i^{rep}$, where

$$\begin{aligned} F_i^{attr} &= \sum_{j=1}^{n} \frac{d_{ij} - d_e(x_i, x_j)}{3 d_e(x_i, x_j)} (x_i - x_j) \\ F_i^{rep'} &= \sum_{j=1}^{n} \frac{x_i - x_j}{d_e(x_i, x_j)} \\ F_i^{rep} &= 2F_i^{rep'} / d_e(F_i^{rep'}, 0) \end{aligned}$$

3. For all $1 \leq i \leq n, 1 \leq h \leq m$ let $x_{hi} := x_{hi} + max\{-5, min\{\Delta x_{hi}, 5\}\}$, i.e. adjust x_{hi} by Δx_{hi} but limit the change by a "temperature", 5 in this case.

The complexity of this naive algorithm is $O(m * n^2)$, where m is the number of iterations of the main loop and n is the number of nodes. Typically m will depend on n. If we choose a linear dependency then the overall algorithm performs in $O(n^3)$.

3 Improving the efficiency of Spring Embedding

Two parameters in the algorithm above influence its convergence to a solution: First, the initial layout and second, the temperature. In this section, we want to explore the former. We leave the temperature fixed and investigate how different initial layouts influence the convergence behaviour of the algorithm. Besides improving the convergence our main focus will be on the complexity of generating the initial layouts. Obviously, the pre-processing should not worsen the spring embedding algorithms performance.

We compared six different initial layouts:

- Random. A uniform distribution is used to place the nodes randomly
- Zero. All nodes are initially placed at 0.
- Circle. All nodes are initially place on a circle. Two subsequent nodes have the same angle between each other. The circle radius is the maximum distance found in the proximity data.
- Clustering. Hierarchical clustering is applied to the data and the clusters are mapped to coordinates.

  Consider the dendrogram in Figure 2. Besides displaying the clusters, the horizontal lines in the dendrogram convey the distances between nodes and between clusters. Instead of using the one dimension only, we can use the second dimension as well and use such a drawing as initial layout. Consider the algorithm in Fig. 6. The mapping from clusters to coordinates uses three parameters: the cluster root, the current position in the drawing and the current direction, which can be vertical or horizontal. We traverse the tree top-down. If the root is a leave, we can place it at the current position. Otherwise, we need the distance d between the two children clusters. Next we call the procedure recursively for each child with updated position and direction. If the direction was horizontal it is set to vertical and vice versa. Depending on the direction, the position's x or y coordinate are set to $+/-\frac{d}{2}$. As an example, consider Fig. 7.
- Spanning tree. Nodes are place on a line. We start with a randomly chosen node placed at 0. For the previous node n placed at x, the nearest neighbour m, which has not yet been placed on the line, is selected. If d is the required distance between n and m, then m is placed at $x + d$.
- PCoA. As described above, Principal Coordinate Analysis (PCoA) [Gow66] is a multi-dimensional scaling technique, which computes a linear mapping. Thus, it is a competitor to spring embedding, which computes a non-linear mapping. But of course we can combine the two and let spring embedding try to improve the linear mapping.

Figure 8 shows the complexity of these Pre-processing techniques. Theoretically, all of them are useful, since they are not worse then spring embedding itself. Practically, however, there exist faster spring embedding algorithms, so that we pre-processing should not be worse than $O(n^2)$.

```
cluster2coord(root, dir, x,y)
 if root is leave then place root at x,y
 else
   let d be distance between the root's two children
   if dir is horizontal then
     cluster2coord(left child,  vertical, x-d/2, y)
     cluster2coord(right child, vertical, x+d/2, y)
   else
     cluster2coord(left child,  horizontal, x, y-d/2)
     cluster2coord(right child, horizontal, x, y+d/2)
   fi
 fi
```

Figure 6: Algorithm to map hierarchical clusters to coordinates.

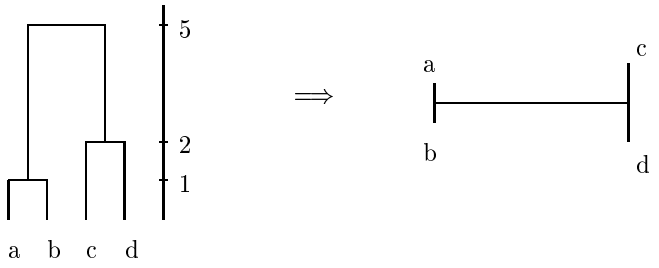


Figure 7: Example of mapping hierarchical clusters to coordinates.

Initial Layout	Random	Zero	Circle	Clustering	Spanning tree	PCoA
Complexity	$O(n)$	$O(n)$	$O(n)$	$O(n^2)$	$O(n^2)$	$O(n^3)$

Figure 8: Complexity of the pre-processing methods used, where n is the number of nodes.

4 Experiments and Results

Let us evaluate the above pre-pocessing methods: We use three data sets.

- The data in Fig. 9 contains pair-wise distances for 72 European cities. This data is not inherently high-dimensional and therefore can be laid out in 2D without large errors. Thus the stress of the spring embedding layout should converge to a low value.
- The data in Fig. 10, left contains random distances, which were derived from uniformly distributed nodes. This is an example of data, where dimension reduction is very limited, i.e. the stress of any layout in 2D will be high. The uniform distribution also means that the data has no clusters, which are the very basis for methods like the spanning tree and clustering method.
- The data in Fig. 10, right contains random data, which is hierarchically normal distributed. We decided on a number of clusters and created for each a point as cluster center using a large mean and variance. Then we recursively, created subclusters around those cluster centers. Thus the data has exactly the structure hierarchical clustering is assuming.

Consider Fig. 9 and the six initial layout described above. As argued above, the data can be laid out in 2D and hence the stress of all techniques approaches 0. However, notably, PCoA, which is linear mapping, produces a very good layout, which cannot be improved by spring embedding. The circle layout performs worst of all methods followed by the spanning tree method, which did not produce a very good initial layout. In fact, the initial random layout and the zero layout, which behave both the same, start with a 3 times better initial layout than spanning tree. Most interestingly, clustering performs very well - better than random. Thus our hypothesis that clustering can be beneficial as initial layout is supported by this data set. But let's look at others.

Consider Fig. 10, left. Since the circular layout performed so bad, it is not included in the figure. As argued above, this data set does not contain structure and thus spanning tree and clustering do not pay off. Surprisingly, the random layout performs very well and even better than PCoA. Thus, we have to refine our hypothesis: clustering can improve spring embedding, but only for data sets, where it is applicable. But let us see how good clustering can get, if the data contains hierarchical clusters.

The third data set was designed for this purpose. Consider Fig. 10, right. Clustering performs best of all methods and starts with a layout about 4 times better than the random one.

As an overall conclusion, we can say that hierarchical clustering, whose complexity is lower than spring embedding, can improve spring embedding, but only if the data it is applied to contains structure and clusters.

5 Temperature Scheduling

As explained in 3, besides the initial layout, temperature scheduling plays an important role in the convergence of the layout quality. In the above experiments, we used a fixed temperature. Let us now look into how this can be improved. Consider Fig. 11. Instead of a fixed temperature, it defines the temperature as a percentage of the force on the node. The figure shows that an optimal percentage is reached at 6%, beyond this point (e.g. at 8%), the stress oscillates. The percentage is not fixed, but as can be seen in Fig. 11, right, the oscillation threshold depends on the

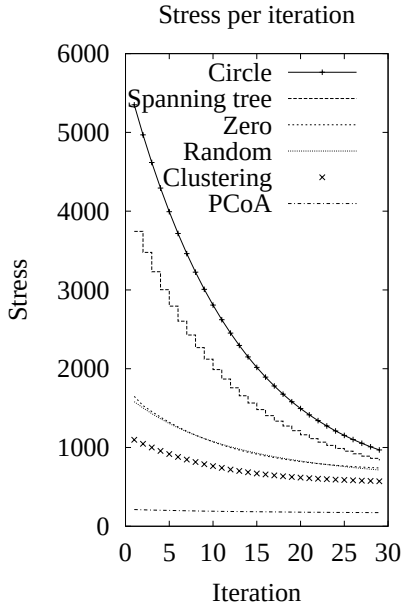


Figure 9: Stress/Iterations for different initial layout applied to pairwise distances of 72 European cities.

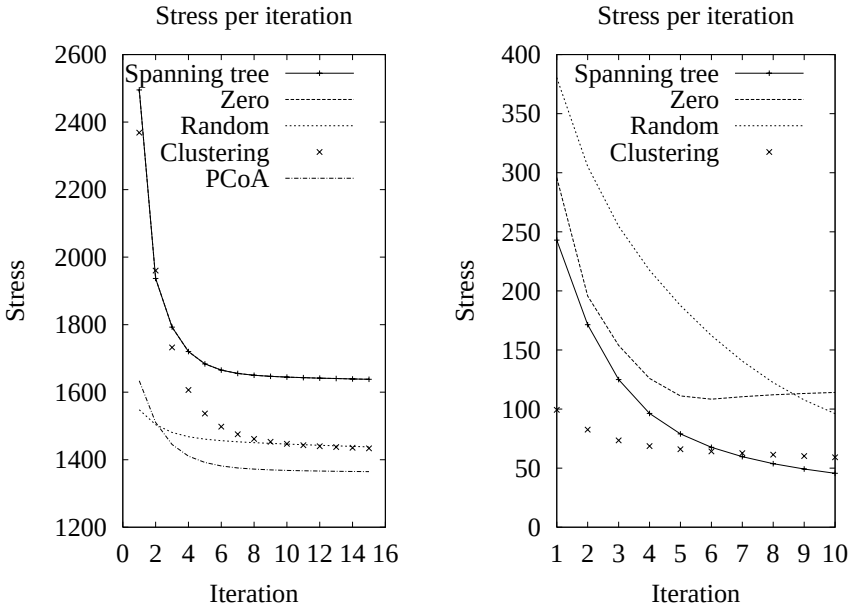


Figure 10: Stress/Iterations for different initial layouts. Left: distances calculated for nodes uniformly distributed. Right: distances for nodes hierarchically normal distributed.

number of nodes. The more nodes there are, the larger the forces acting on a node are and the smaller the force percentage used for movement of the nodes should be.

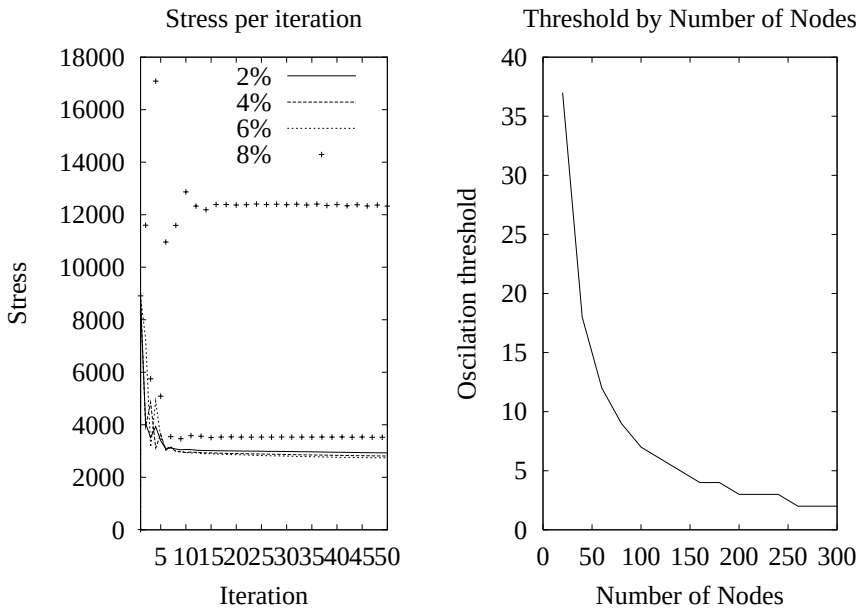


Figure 11: Left: Stress/Iteration for temperature as percentage of force. 6% leads to the best behaviour and beyond this (8%) the stress oscillates. Right: Number of nodes/oscillation threshold. Depending on the size of the data set, a different percentage is best.

Overall the temperature can highly influence the convergence behaviour of the stress function. Fig. 12 shows the stress functions for fixed temperature and for the temperature as percentage of the force. The latter performs clearly better.

6 Comparison and Conclusions

Visual datamining is a process of data preparation, processing, and visualisation. Often the data is inherently high-dimensional proximity data, which can be mapped to 2D or 3D and then visualised as scatterplot. One such mapping - a non-linear one - can be realised by force directed graph-drawing with spring embedding [Kru64, QB79, KK89, FR91, AAH94, RD96, Tun99, BETT99]. In this context of multi-dimensional scaling [Gor81, Eve78, Kru77, MKB79, Web99], spring embedding has advantages over other methods such as PCoA [Gow66] as it is a flexible anytime-algorithm which comes up with an approximation of a solution with any time limitations. But similar to PCoA, the basic spring embedding algorithm has cubic runtime in the number of nodes, if one assumes that the outer loop for the force computations depends linearly on the number of nodes. One way in which spring embedding's performance can be improved is to use pre-processing to start with an initial layout, which already approximates a solution instead of a purely random initial layout. The hypothesis we investigated in this paper is whether hierarchical clustering can be used to this end. We used different datasets and other initial layout techniques to be able to compare different approaches. We can draw the following conclusions:

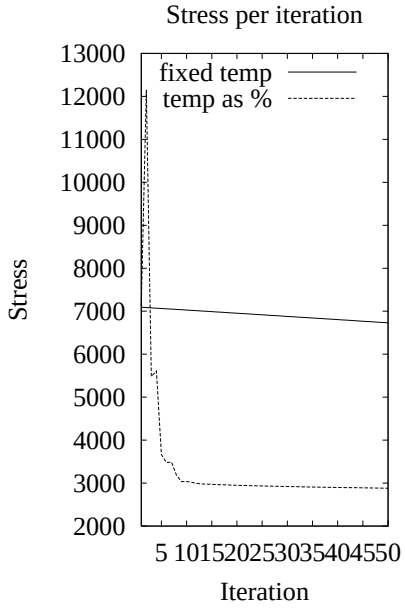


Figure 12: Improved convergence of spring embedding using temperature scheduling.

- A random initial layout performs usually very good, if temperature scheduling is used appropriately.
- The circular layout performed very poor.
- The spanning tree method performed very poor.
- Hierarchical clustering can be a very good initial layout - in fact, better than the random - if the underlying data contains clusters.
- Temperature scheduling is as important as the initial layout and a good schedule can quickly adjust any initial layout, however poor it may be.

The poor performance of the spanning tree method came as surprise and we modified it to perform better. The modification involved using the plane to layout the nodes instead of just a line as described in section 3. The results of the 2D spanning tree were an improvement, but not yet as good as the others. If one continues this approach to 3D, 4D, etc. one ultimately implements a similar transformation to PCoA. A spanning tree is also used by Tunkelang [Tun94], where the first stage of the layout consists of constructing a minimal height breadth-first spanning tree of the graph. This is related, but different from our approach. There are two other approaches to using clustering as preprocessing: Morrison, Ross, and Chalmers [MRC02] use k-means, while we use hierarchical clustering. Walshaw [Wal01] uses an iterative approach, which is very similar to ours. But there are two important differences: Both Morrison et al. and Walshaw work with typically sparse graphs, whereas we worked with complete graphs. Walshaw's approach of clustering corresponds to hierarchical clustering with single linkage. With this relationship to hierarchical clustering, one could experiment how different linkage methods influence the layout. The second difference to Walshaw is that he mainly evaluated his approach on examples, whereas we tried to go one step further and examine

examples stemming from different classes of data, such as unstructured data and hierarchically clustered data as the underlying structure of the data will greatly influence the behaviour of the algorithm.

References

[AAH94] A.Frick, A.Ludwig, and H.Mehldau. A fast adaptive layout algorithm for undirected graphs. In *Proceedings of Graph Drawing GD94*, pages 388–403. Springer, 1994.

[BETT99] G. Batista, E. Eades, R. Tamassia, and I. Tollis. *Graph Drawing: Algorithms for the visualisation of Graphs.* Prentice Hall, 1999.

[DEKM98] C. Durbin, S. Eddy, A Krough, and G. Mitchison. *Biological Sequence Analysis.* CUP, 1998.

[dIB97] J deRisi, V R Iyer, and P O Brown. Exploring the metabolic and genetic control of gene expression on a ge nomic scale. *Science*, 278:680–686, 1997.

[Ead84] P. Eades. A heuristic for graph drawing. *Congressus Numerantium*, 42:149–160, 1984.

[ESBB98] M B Eisen, P T Spellman, P O Brown, and D Botstein. Cluster analysis and display of genome-wide expression patterns. *PNAS*, 95(25):14863–14868, 1998.

[Eve78] B. S. Everitt. *Graphical techniques for multivariate data.* Heinemann Educational Books, 1978.

[FMC96] F.J.Brandenburg, M.Himsolt, and C.Rohrer. An experimental comparison of force-directed and randomized graph drawing algorithms. In *Proceedings of Graph Drawing GD95*, pages 76–87. Springer, 1996.

[FR91] T. Fruchterman and E. Reingold. Graph drawing by force-directed placement. *Software practice and experience*, 21:1129–1164, 1991.

[Gor81] A. D. Gordon. *Classification.* Chapman and Hall, 1981.

[Gow66] J. C. Gower. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53:325–38, 1966.

[KK89] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 1:7–15, 1989.

[Kru64] J. Kruskal. Multidimensional scaling by optimizing goodness to fit to non-metric hypotheses. *Psychometrika*, 29:1–27, 1964.

[Kru77] J. Kruskal. The relationship between multidimensional scaling and clustering. In *Classification and clustering.* Academic Press, 1977.

[MKB79] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate analysis.* Academic Press, 1979.

[MRC02] Alistair Morrison, Greg Ross, and Matthew Chalmers. Combining and comparing clustering and layout algorithms. University of Glasgow, 2002.

[Ols95] Clark F. Olson. Parallel algorithms for hierarchical clustering. *Parallel Computing*, 21(8):1313–1325, 1995.

[QB79] N. R. Quinn and M. A. Breuer. A force directed component placement procedure for printed cicuit boards. *IEEE Transactions on Circuits and systems*, CAS-26(6):377–388, 1979.

[RD96] R.Davidson and D.Harel. Drawing graphs nicely using simulated annealing. *ACM Transactions on Graphics*, 15:301–331, 1996.

[Sam69] J. W. Sammon. A nonlinear mapping for data analysis. *IEEE Transactions on Computers*, C(18):401–409, 1969.

[SGvHN01] Michael Schroeder, David Gilbert, Jacques van Helden, and Penny Noy. Approaches to visualisation in bioinformatics: from dendrograms to space explorer. *Information Science: An international journal*, 139(1):19–57, 2001.

[SM58] R. R. Sokal and C. D. Michener. A statistical method for evaluating systematic relationships. *University of Kansas Scientific Bulletin*, 28:1409–1438, 1958.

[Tun94] D. Tunkelang. A practical approach to drawing undirected graphs. Technical report, Carnegie Mellon University, School of Computer Science, 1994.

[Tun99] D. Tunkelang. Jiggle: Java interactive graph layout enviroment. In *Proceedings of Graph Drawing GD98*, pages 413–422. Springer, 1999.

[Tut60] W. T. Tutte. Convex representations of graphs. *Proc. of London Mathematical Society*, 10:304–320, 1960.

[Tut63] W. T. Tutte. How to draw a graph. *Proc. of London Mathematical Society*, 13:743–768, 1963.

[Wal01] C. Walshaw. A Multilevel Algorithm for Force-Directed Graph Drawing. In J. Marks, editor, *Graph Drawing, 8th Intl. Symp. GD 2000*, volume 1984 of *LNCS*, pages 171–182. Springer, Berlin, 2001.

[Web99] Andrew Webb. *Statistical pattern recognition.* Arnold, 1999.

A Visual Data Mining Environment*

Stephen Kimani, Tiziana Catarci, and Giuseppe Santucci

Universitá di Roma "La Sapienza",
Dipartimento di Informatica e Sistemistica,
Via Salaria 113, 00198 Roma, Italy
{kimani, catarci, santucci}@dis.uniroma1.it

Abstract. It cannot be overstated that the knowledge discovery process still presents formidable challenges. One of the main issues in knowledge discovery is the need for an overall framework that can support the entire discovery process. It is worth noting the role and place of visualization in such a framework. Visualization enables or triggers the user to use his/her outstanding visual and mental capabilities, thereby gaining insight and understanding of data. The foregoing points to the pivotal role that visualization can play in supporting the user throughout the entire discovery process. The work reported in this paper is part of a project aiming at designing a Data Mining system with a visual environment that supports the user in the entire process of mining knowledge.

1 Introduction

It should be acknowledged that a lot of research work has been and is being done with respect to knowledge discovery. However, much of the work concentrates on the development and optimization of data mining algorithms using techniques from other fields such as Artificial Intelligence, statistics and high performance computing, with little consideration, if any, of the other knowledge discovery phases. Consequently, corresponding tools/systems are normally difficult to integrate in the entire knowledge discovery process.

There is a major need to develop an overall framework that can support the entire knowledge discovery process[1]. The framework should accommodate and integrate all the phases seamlessly. Since it is not known *a priori* all the components the framework will be expected to support, the framework should be extensible to any new components. On the same note, the development of the framework and the components should be separated.

One of the interesting issues related to the ongoing discussion is the role of visualization in the knowledge discovery process. Visualization is a very effective means of enabling the user to use his/her outstanding perceptual capabilities to recognize and understand data. Traditionally, visualization has been placed at the beginning and at the end of the knowledge discovery process. Instead, visualization has its place in all the phases of the knowledge discovery process. This

* This work is supported by the MIUR project D2I: *Integration, Warehousing, and Mining of Heterogeneous Sources (http://www.dis.uniroma1.it/~lembo/D2I)*

puts visualization, and therefore the user, at the center of the entire knowledge discovery process. This is a major step toward developing user-centered data mining systems.

This paper describes a Data Mining system with a visual environment aiming at supporting the user throughout the entire data mining process. The visual data mining environment employs a wide range of novel and intuitive visual strategies toward realizing the foregoing aim.

The rest of the paper is organized as follows: section 2 focuses on related research work. Section 3 describes the architecture of the proposed Data Mining system. A detailed description of the visual data mining environment is presented in section 4. Section 5 highlights efforts aimed at defining a mapping between the abstract data mining engine and the visual interface. Work on usability studies is presented in section 6. Future work and a conclusion are presented in section 7.

2 Related Work

In this section, a discussion of some data mining systems that offer a reasonably great and diverse number of data mining and visualization functionalities that have been proposed in the literature is given.

Clementine [8] is a product by Integral Solutions Ltd (ISL). SPSS purchased ISL on December 31, 1998. Clementine provides various data mining algorithms to support various techniques including; clustering, association rules, sequential patterns, factor analysis, and neural networks. The product is easy to use through its visual programming interface.

The FlexiMine system [3] has been designed as a test bed for data mining research. The system is also intended to serve as a generic knowledge discovery tool. At the time of going to writing and to the best of our knowledge, FlexiMine contains algorithms for handling association rules, Bayesian networks, decision trees, and metaqueries.

Another related research effort is the QUEST project [9] at IBM Almaden Research Center. The project was intended to discover useful patterns in large databases. The research effort provides support for a notably wide range of data mining algorithms including association rules, sequential patterns, classification, and time-series clustering. IBM markets the technology using the commercial product DB2 Intelligent Miner.

MineSet [7] is a data mining system developed by Silicon Graphics Inc. MineSet supports association rules and classification models. It uses these models for carrying out prediction, scoring, segmentation, and profiling tasks. Besides its application of robust data mining algorithms, MineSet is notable for its sophisticated visualizations.

GGobi [11] is an interactive data visualization system for exploratory data analysis. The system is the result of significant redesign of its predecessor, XGobi [10]. One of the interesting new features supported by GGobi is the plug-

in functionality. Consequently, GGobi can accommodate functionalities from other applications and/or it can be deployed in other applications.

Viscovery SOMine [12] is a data mining system developed by Eudaptics. Among other data mining methods, the system supports clustering, prediction, regression and association. Viscovery SOMine provides an interactive environment in a bid to support the user in the data mining process.

Another data mining system is Decision Series [6] by Neo Vista Solutions Inc. Accrue Software Inc acquired Neo Vista in February 2000, and has used Decision Series to support analysis applications. On June 27, 2001, Accrue sold Neo Vista intellectual property to JDA Software Inc. Through the sale, JDA assumes the intellectual property of the Decision Series data mining toolset and the RDS-Assort and RDS-Profile applications. Decision Series supports clustering, association rules, and neural networks.

Each of the foregoing systems exhibits at least, either one or both of the following limitations:

- *Non-extensible framework:* The system is based on a framework that supports only some specific phases in the data mining process. Consequently, it is extremely difficult, if not impossible, to incorporate new components.
- *Non-homogeneous environment:* The system makes use of a non-uniform mining environment. The user is presented with "totally" different interface(s) across implementations of different data mining techniques.

Our Data Mining system takes on an integrated approach in terms of its framework. Moreover, the system is geared toward providing the user with a consistent, uniform and flexible interaction environment across the entire process of data mining.

3 The Proposed Data Mining System

At present, the system supports, but is not limited to: metaqueries, association rules, and clustering.

The architecture of the system comprises two primary layers: the user layer, and the data mining engine layer, as seen in Figure 1.

1. The *Parametric User Interface/User Layer* enables the user to interact with the other system components. It invokes the relevant system feature or functionality on behalf of the user. Ideally, this layer/component empowers the user to process data (and knowledge), and also to drive, guide and control the entire discovery process.
   The user component is organized around a GUI container which hosts specific GUI extension cartridges, which in turn contain the knowledge to access their respective underlying data mining components/modules in the *Data Mining Engine Layer*. In effect, the GUI container registers the specific data mining technique GUI extension, loading respective specific menu items and other commands specific to the data mining technique. For instance, the specific

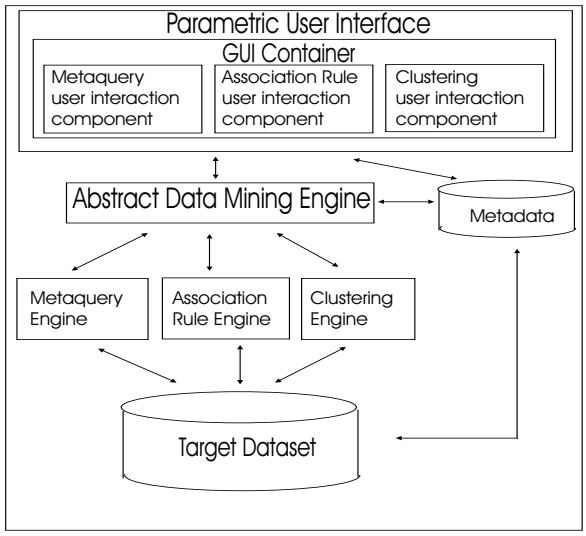


Fig. 1. System Architecture

data mining technique GUI extension for clustering has the knowledge on how to access the clustering engine and also on how to interact with the user in the acquisition of clustering input and in the visualization of clustering output.

The GUI container provides various services to the Data Mining system. These services fall under two categories: infrastructural services and end user services.

The infrastructural services that are supported include:

- Registration of extensions which implement specific interaction contracts.
- Runtime loading on the interaction environment of the features that are relevant to the active GUI extension (e.g. commands and options).
- Advertising new GUI extensions.
- Routing user commands to the active GUI extension.

As for the end user services, the Data Mining system support includes:

- Providing the user with a uniform, consistent and flexible user interface.
- Providing services whose use span across the entire data mining interaction environment (such as start, stop, save and load services).

There are various functionalities that a data mining GUI extension supports. The GUI extension carries out the specific commands that are loaded and made available to the user (loaded on the interface) by the GUI container. The extension also implements specific input and output modalities for the underlying data mining technique or algorithm(s). On the whole, modalities specific to the data mining technique may be added or may substitute some or part of the general pre-existing ones.

2. The *Abstract Data Mining Engine Layer* is completely decoupled from the *User Layer*. However, the structure of the Data Mining engine depicts parallelism with the structure of the GUI.
   The *Data Mining Engine Layer* is structured using an abstract reference model based on the following concepts:
   - A global dataset which contains the data to be mined and all the information necessary to apply and execute a data mining technique (such as metadata information).
   - A command manager which forms the interface between the Data Mining engine and the *User Layer*. On one hand, it interprets user commands originating from the user interface and on the other hand it manages any access to the internal structure of the engine. The command manager therefore serves as a two-way link between the engine and the GUI (the GUI container and the specific GUI extension). Some of the operations performed through the command manager include: defining the initial set of target data, applying some data mining algorithm, storing hypothesis and verifying hypothesis.
   - An abstract Data Mining discoverer for carrying out the discovery data mining goal. The discovery task might use inputs directly given by the user or from previous data mining results. This part of the engine must be specialized to implement some specific algorithm.
   - An abstract Data Mining verifier for carrying out the verification data mining goal. Like the abstract Data Mining discoverer, the abstract Data Mining verifier also must be specialized to implement some specific algorithm.

   The general behavior of the engine is abstract in that, it must be instantiated/specialized to specific "engines", one for every data mining technique. It should also be pointed out that a hypothesis that is discovered and/or verified by one specific "engine" can be used by another "engine". Consequently, the result of some data mining task can be used as input to another task. The instantiated "engines" are made available to the general engine framework dynamically. As a consequence, they are also made available to the user through the specific/respective GUI environment.

It is worth pointing out that there are some services that are available to every specific "engine". Such services include: metadata management, configuration savings, intersystem communication[1], data access and database connection management.

As already mentioned, the architecture supports the incorporation of new and the modification of pre-existing components. Specific extension points are defined right where such component additions or modifications occur. It is envisioned that new components will be incorporated as plug-ins[1].

In the current implementation, the *User Layer* is developed using JBuilder. It runs on Windows operating system. The command manager, which forms an

[1] Intersystem communication deals with the management of the possible interactions and data transfer of different data mining techniques in a uniform way

interface between the *Data Mining Engine Layer* and the *User Layer*, is being developed using XML DTDs, as indicated later in section 5. The specific data mining "engines" are implemented using Delphi. It should be mentioned that the "engines" correspond to specific data mining algorithms.

4 The User Interface

The visual interface is designed based on the goal of supporting the user in the entire data mining process. Moreover, the interaction environment is consistent, uniform and flexible. The interface employs various visual strategies that can effectively enable the user to exploit his/her powerful visual capabilities with a view to discovering knowledge through metarules, association rules and clustering.

Toward describing the system features, we consider a communications company that provides various services such as Web-access services, telephone services, etc. The company has a main office and a number of service centers. The main office principally deals with strategic and administrative issues. In fact, the company offers its services through the service centers. The company plans to introduce some special offers. The marketing director is expected to recommend the type of service to be featured and the customers to be targeted. Assume that the marketing head decides to mine some information using the Data Mining system. In this case, we may view him as the user of the system.

The marketing director might want to identify regions that had relatively good general service sales in the last one month. They might want to use that information further to propose some specific service and customers that might be worth consideration in the offer. The recommendation could also include another service that normally does best with the proposed service.

4.1 Identifying Regions With Good Sales: Using the Clustering Environment

Understanding how different regions have been doing can be resourceful in making marketing decisions. The task can be accomplished through the clustering environment.

As a user, one starts by specifying a target dataset. The specification relies on two intuitive interaction spaces namely the specification space and the target space. This may be seen in all the figures illustrating the visual interface (e.g. in the top-left part of Fig. 2). The specification space provides the mechanisms, tools and resources necessary for visually building the set of task relevant data. The target space holds or hosts the relations that are part of the task relevant dataset. The latter space may be envisaged as a container for the constructed target dataset. The two spaces are backed with drag and drop mechanisms and tools. Moreover, the two spaces are complementary in the manner in which they support the user. Therefore, the user operates by appropriately moving between the two components. Since the interface supports drag-and-drop mechanisms,

the user may intuitively move elements (such as relations and relation items) from one component to the other as appropriate.

In this task, the marketing executive is mainly interested in customers and services (i.e. based on relations *Customer* and *Service* or on relation *CustServ*). The company already has geographical information pertaining to customer addresses. For instance, their *loci* with respect to the main office. The user may construct a relation in which the attributes of interest are *CustID*, *CustX*, *CustY*, *CustAmt* and *ServID*.

The Data Mining system provides an interaction environment with various input widgets through which the user can specify parameters characterizing a clustering task. For instance:

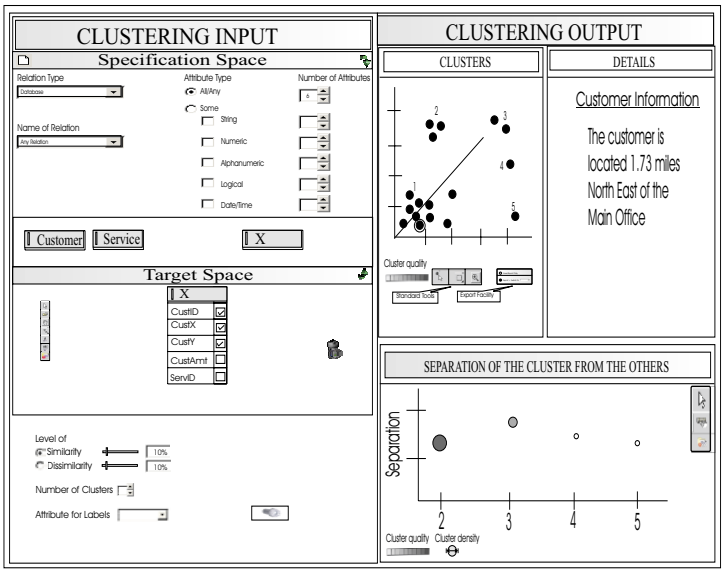


Fig. 2. The Clustering Environment

- Radio buttons for the specification of homogeneity or separation measures. Each of the two measures is presented on an ordinal scale with a slider control.
- A palette-based mechanism for specifying an attribute value.
- A "checking" mechanism for specifying attributes that will determine the partitioning of the target dataset. In Fig. 2, the attributes *CustID*, *CustX* and *CustY* have been selected ("checked").
- A combo-box for specifying an attribute to be used in labeling clusters (In Fig. 2, the labeling attribute has not been specified. In such a case, the system would default to numbering the clusters serially).

When through with the parameter specification, the marketing head may instruct the system to partition the target dataset by clicking the "torch" icon. The system performs the clustering and displays the results.

As regards clustering, the system offers two main visualization mechanisms: *Clusters + Details ("Overview + Detail")* and *Separation View*.

Clusters + Details ("Overview + Detail")
This visualization displays clusters on a scatter plot, and also presents details that correspond to a selected cluster or cluster object. The former display corresponds to an "Overview" window whereas the latter corresponds to a "Detail" window.

Cases are mapped to points on the scatter plot, with each point taking some x, y (and if appropriate z) values. The system uses two alternative encoding approaches with regard to the scatter plot display. In the first approach, objects that belong to the same cluster are optionally enclosed in a bounded region. Each such region is shaded with some grayscale level that reflects the accuracy level of the represented cluster. In the approach, outliers are drawn positioned outside the bounded regions. A currently selected cluster, cluster object, or outlier is drawn with an outline around it. The points themselves may be encoded to reflect some other aspects (e.g. by coloring). The top-right part of Fig. 2 shows a visualization in which the scatter plot is generated using this first approach and in which the user has opted to have no enclosures. The values on the x-axis correspond to *CustX*, those on the y-axis correspond to *CustY* and the z-axis values correspond to *CustID*. In the second approach, objects that belong to the same cluster are drawn using the same color saturation level. The color saturation level also represents the accuracy of the represented cluster. In this approach, outliers are all drawn using one color. The outlier color is reserved for that purpose and therefore is not used to map any other aspect. The system determines the approach to use based on the distribution of cluster objects. The "Detail" window effects the exposition of a selected cluster or point. The interface relies on a system-driven mechanism which determines an appropriate presentation style for the details of the selected entity.

Separation View
Measures of accuracy are useful in many ways (e.g. for interpretability and evaluation purposes). The system currently supports a display based on a separation function. The visualization is a graph depicting how far the selected cluster (or the cluster containing the currently selected point) or outlier is from the other clusters and outliers e.g. the bottom-right part of Fig. 2 shows how far the cluster containing the outlined object is from the other clusters and outliers. The value of separation is mapped to the y-axis. A circle encodes a cluster or an outlier. The circles are arranged along the x-axis. The density of the cluster or outlier is bound to the size of the circle. The grayscale level of a circle represents the quality of the represented cluster or outlier.

From the *Clusters + Details* and *Separation View* visualizations, regions that are close to the main office depict a lot of sales. With regard to the anticipated offer, a marketing strategy might put a lot of emphasis on people and service centers that are close to the main office.

The marketing executive might want to gain more knowledge from those interesting regions. For instance he might want to identify some specific service and customers within those particular regions. The task would entail establishing data relationships. The analysis can be done through the metaquery environment.

However, it is important to observe that the task is based on some particular subset of data which is not equivalent to the currently defined set of target data. In other words, the user intends to use some output from one task (clustering) as an input to another task (metaquerying).

The interface enables the user to select points or clusters of interest through the use of the *Standard Tools* toolbox. The marketing director may then turn to the *Export Facility*. The resource would enable him to specify whether he would want to just save the specified output or to save and switch to another task with that output as the input to the new task. In the latter case, the system switches to the new environment with the output appearing in the Target Space.

In the ongoing example, the relation *ClustOutl* in Figure 3 represents the clustering output that has become metaquerying input.

4.2 Establishing Data Relationships: Using the Metaquery Environment

The marketing director will need to analyze the relationships that exist among services, customers, and centers. The analysis would help him to determine the service to feature and potential customers. The metaquery environment can be helpful in carrying out the analysis. Such relationships can be mined by exploiting the relations $CustCent$, $CustServ$, and $ServCent$. Assume that the user is interested in the following attributes: $CustCent.CustID$, $CustCent.CentID$, $CustServ.CustID$, $CustServ.ServID$, $ServCent.ServID$ and $ServCent.CentID$. Therefore the marketing director needs to specify the three relations with the foregoing attribute constraints toward constructing the target dataset. It should be mentioned that the metaquery analysis will be restricted to only the tuples contained in the data that was "imported" from the clustering task (tuples in the relation *ClustOutl*), which is already in the target space.

In the environment, the user may define links/"joins"[2] manually (*Manual Joins*) or have the system automatically do that (*Automatic Joins*). Assume that the marketing director chooses the latter option. The system links the attributes as follows:

– $CustCent.CustID$ with $CustServ.CustID$

[2] As far as our designing of metaqueries is concerned, "joins" do not refer to table joins. A "join" refers to a link between attributes that is aimed at generating a consequent pattern

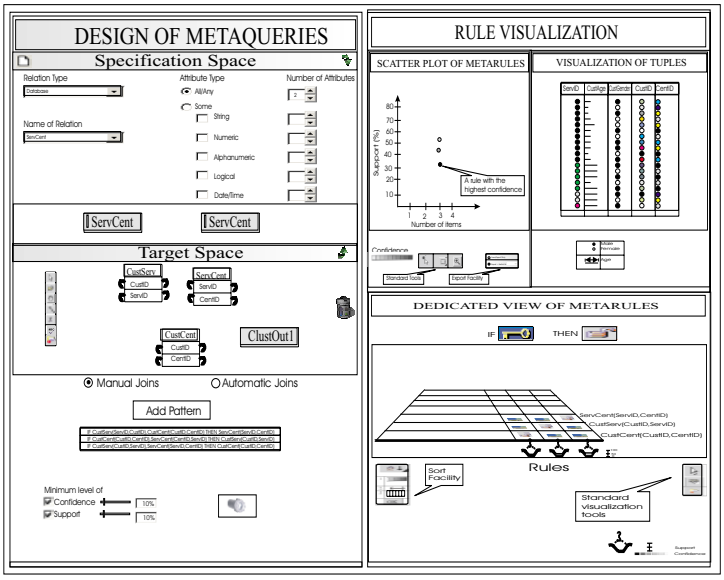


Fig. 3. The Metaquery Environment

– $CustCent.CentID$ with $ServCent.CentID$
– $ServCent.ServID$ with $CustServ.ServID$

Letting X be a representation for $CustID$, Y for $CentID$, and Z for $ServID$, and allowing reordering of attributes, the system generates the following transitive "combinations" (which are actually metaqueries):

1. $CustCent(X,Y), CustServ(X,Z), ServCent(Z,Y)$
2. $CustServ(X,Z), CustCent(X,Y), ServCent(Y,Z)$
3. $ServCent(Z,Y), CustServ(Z,X), CustCent(X,Y)$

The system puts the patterns in a pool as seen in Figure 3. The director may also specify confidence and support values by using sliders or text-boxes. He may then instruct the system to search for specific rules from the target dataset that correspond to the metapatterns in the pool and that satisfy the specified parameters, by clicking the "torch" icon.

The Data Mining system provides various visualizations for the search results. For any rule, the aspects that are of principal interest include: measures of interestingness, relationship between the head and body, and details about the items participating in the rule. The system provides two main visualizations for the search results *Rules + Tuples* (Overview + Detail) and *Dedicated View.*

Rules + Tuples (Overview + Detail)
This visualization displays all the rules from the search operation, and also

presents tuples that correspond to some selected rule(s). The rules are displayed using a scatter plot. The interface invokes a system-driven mechanism which chooses an appropriate presentation style for the tuples.

The scatter plot may be envisaged as the "Overview" window and the tuples display as the "Detail" window. On the scatter plot, a rule is mapped to a point, the confidence of a rule to grayscale, the support of a rule to the y-axis, and the number of items in a rule to the x-axis. Consider that the metaquery search based on the ongoing example returns the following results:

1. $CustCent(CustID, CentID) \leftarrow$
   $CustServ(CustID, ServID), ServCent(ServID, CentID)$
   Support = 33.33% and Confidence = 100%
2. $CustServ(CustID, ServID) \leftarrow$
   $CustCent(CustID, CentID), ServCent(CentID, ServID)$
   Support = 44.44% and Confidence = 75%
3. $ServCent(ServID, CentID) \leftarrow$
   $CustServ(ServID, CustID), CustCent(CustID, CentID)$
   Support = 55.56% and Confidence = 60%

In Fig. 3, there is a *Rules + Tuples* visualization of the foregoing results. In the visualization, the marketing director has selected the rule with the highest confidence for exposition. The tuples window depicts some interesting trends. The service represented by black circles had the highest demand. The marketing head may interact with the display, for instance by using the exposition tools or by pointing the circle(s), thereby getting to know that the interesting service is *WWW-Access*. The display also depicts that, virtually all the customers who requested the *WWW-Access* service are young.

Consequently, the marketing director may wish to consider *WWW-Access* service for the anticipated offer. Moreover, he has fairly substantial information regarding the customers to target: those living *close* to the main office and who are of *young* age.

Dedicated View
Like the foregoing visualization, the *Dedicated View* enables the user to visualize all the rules from the search operation. However, in this case, rules are visualized in a more elaborate manner. The *Dedicated View* displays: the confidence and support values of each rule, the relationship between the head and the body of each rule, and the individual items/components that make up each rule. The visualization uses a simple 2D floor with a perspective view. The floor has rows and columns. A rule is represented by a column on the floor. The rule is made up of the components which have entries in the column. The rows represent the items (such as attributes). Associated with each column/rule is a "bucket". The gray value of the contents of the "bucket" represents the confidence value of the rule. The level of the contents of the "bucket" is bound to the support value of the rule. The handle of the "bucket" can be used to select the corresponding rule. Rule items that form the antecedent are each represented using a "key" icon,

whereas those that form the consequent are each represented using a "padlock" icon. The visualization can be seen in the bottom-right part of Figure 3.

4.3 Market-Basket Analysis: Using the Association Rule Environment

The marketing director might also intend to find out another service that is frequently requested every time *WWW-Access* service is requested. Such knowledge would be instrumental in making some marketing decisions. For instance in designing advertisements that capture the two products. The analysis can be realized by switching to the association rule environment. It is worth mentioning that if the user would be interested in switching to the the new environment with the previous output as the new input, he could use the *Export Facility* that was described at the end of section 4.1. One of the distinct features in the

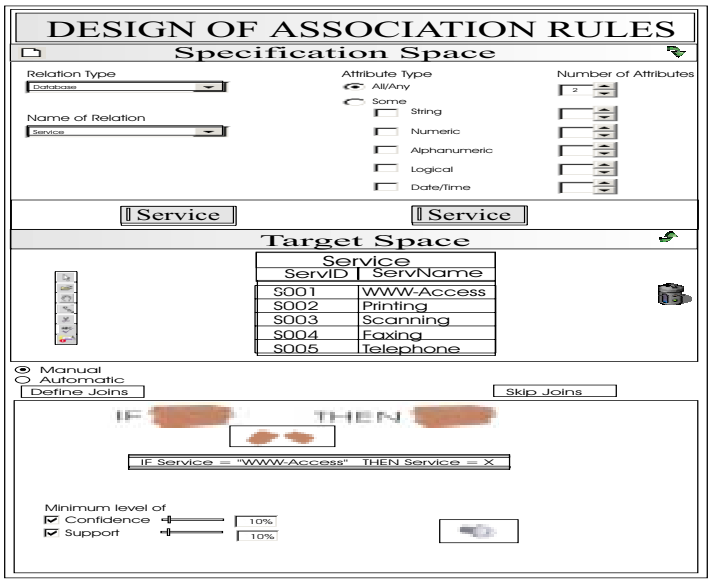


Fig. 4. Basket-based Construction of Association Rules

association rule environment is the provision of "market baskets". It is interesting to note that the interface allows the marketing director to formulate the quest without having to understand the transaction details. For this task, it is enough for him to just have the *Service* relation and constrain it to attributes *ServID* and *ServName*. The resultant relation is seen in the target data space of Fig. 4. Toward specifying the structure of the association rule of interest, the marketing director would drag and drop the tuple *Service = "WWW-Access"*

into the first "basket". He may leave the second "basket" empty as a generic service entry that the system will later instantiate with various relevant service entries. Figure 4 shows part of the association rule environment. In the figure, the marketing director already has put the item into the *IF* "basket" and has emptied the "baskets" into the pool. The user may specify threshold levels and then instruct the system to carry out a search based on the foregoing inputs. The system returns association rules that satisfy the specified parameters.

The association rules are visualized using the same mechanisms that are used for visualizing metarules. By observing the association rule having outstanding measures of interestingness in the visualization, the marketing director may be able to determine the best service to associate with *WWW-Access*.

5 Mapping

Defining a mapping between the abstract components of a system and the corresponding visual ones is beneficial in many ways. For instance, such a definition facilitates data exchange, process automation, data storage and capturing of semantics. It is on the basis of the foregoing understanding that we have embarked on an effort to develop a mapping language for the Data Mining system[2]. At the moment, we have realized some preliminary definitions for metaqueries and clustering.

With regard to metaqueries, we have developed an initial version of MIF (Metarules Interchange Format). MIF may be envisaged as a two-way link for exchanging metarule-based information between any applications that deal with metarules. In the context of our Data Mining system, MIF would facilitate communication between the Visual Interface and the metaquery engine, Metaquery Evaluation Engine (MEE). The Visual Interface generates XML input documents, written in MIFIn format. A MIFIn document specifies the inputs (or contents of a request) for a metaquery task. The MEE produces XML output documents, written in MIFOut format. A MIFOut document contains the answer to a metaquery request. The document can be displayed by the Visual Interface.

The Data Mining Group has recently developed an industrial XML-based standard for the exchange of results between mining applications named PMML, an acronym for *Predictive Model Markup Language*[4]. PMML 2.0 is almost entirely satisfactory with regard to the description of output from a clustering task. However, the specification has no sufficient provision for the description of input to a clustering task. We propose an XML DTD that specifies input to a clustering task. As for output, we have carried out an evaluation of PMML 2.0 and consequently defined an update that supports clustering output description.

6 Usability

To determine the usability of interfaces, it is necessary to subject them to rigorous evaluation tests. It should be pointed out that our system primarily targets

users who are acquainted with data mining. This user audience is specialized and it may be reasonable to consider them as expert users. It would arguably be easier to design an interface for a specific type of users than for a mixed audience. Nonetheless, the need to carry out usability tests remains. As a way of getting started, we carried out usability heuristics. The term "usability heuristics" refers to a more informal evaluation where the interface is assessed in terms of more generic features. This informal evaluation presents reasonably concise and generic principles that apply to virtually any kind of user interface. In the following discussion, we analyze how some of the principles have been applied in the design of the Data Mining system.

- The interface dialogue should be simple and natural. Moreover, the interface design should be based on the user's language/terms. In general, there should be an effective mapping between the interface and the user's conceptual model. In our system, the interface primarily uses data mining terms. It is worth recalling that our target audience comprises users who are conversant with data mining concepts. Furthermore, the provision of "hooks" and "chains" for linking attributes, "baskets" for designing association rules, "drag and drop" mechanisms, "buckets" for measures of interestingness, and "keys" and "padlocks" for antecedents and consequents are part of getting effective mappings between the interface and the user's conceptual model.
- The interface should shift the user's mental workload from the cognitive processes to the perceptual processes. Our Data Mining interface supplies various mechanisms to support the shift. For practically all inputs, the user does not have to supply the units of measurement. Moreover, the system offers interaction controls (e.g. sliders) for helping the user get familiar with the range of valid values and also for helping him/her input within the range. Furthermore, visually presenting query parameters (e.g. data relations) minimizes the possibility of making mistakes while formulating a query.
- There should be consistent usage and placement of interface design elements. Consistency builds confidence in using the system and also facilitates exploratory learning of the system. In our interface, the same information is presented in the same location on all the screens.
- The system should provide continuous and valuable feedback. One of the mechanisms our Data Mining system uses to provide feedback is realized when the user puts some item into the "baskets" or empties the "baskets". The "baskets" respond to reflect the insertion or the removal.
- There is a need to provide shortcuts especially for frequently used operations. In the Data Mining interface, there are various shortcut mechanisms, for instance double clicking and single key press commands.
- There are many situations that could potentially lead to errors. Adopting an interface design that prevents error situations from occurring would be of great benefit. In fact, the need for error prevention mechanisms arises before (but does not eliminate) the need to provide valuable error messages. Our Data Mining interface offers mechanisms to prevent invalid inputs (e.g. specification by selection, specification through sliders). It also provides some

status indicators (e.g. when an item is put in a "basket", the status of the "basket" changes to indicate containment).

Moreover, on informal user tests, we performed some informal user tests on a previous version of the prototype with data mining experts from the universities of Bologna [13] and Calabria [14]. We got encouraging results from the tests and even suggestions on how to improve the interface. For instance, the data mining experts suggested that the interface should provide an optional interaction environment specifically designed for the expert user and still leave the user with the freedom to switch between the two.

At present, we are designing formal usability experiments for the current version of the prototype. Consequent results would be instrumental in determining further relevant interface improvements and modifications.

7 Future Work and Conclusion

There are plans to incorporate an optional visual environment designed for the expert user. Based on the understanding that similarity queries can significantly improve the interactivity of a data mining process, we are planning to incorporate such support. Moreover, there are plans to incorporate visualization systems/tools into the system for the exploration of data mining results. One of the visualization systems that we are intending to use is the DARE system [15] [16]. However, at present, the incorporation has not been done yet. Work on formalization and usability studies is still going on. At the moment, there is a partial prototype of the Data Mining system. The complete implementation of the same is underway. In this paper, the need for a framework that supports the entire discovery process has been discussed. The paper has also highlighted the pivotal role that visualization plays in such a framework. A Data Mining system with a visual environment that is aimed at supporting the user in the pursuit of knowledge has been described.

References

1. Fayyad, U., Grinstein, G. G., Wierse, A. (eds.): Information Visualization in Data Mining and Knowledge Discovery. Morgan Kaufmann Publishers (2002)
2. Catarci, T., Ciaccia, P., Curci, V., Kimani, S., Ianni, G., Lodi, S., Palopoli, L., Patella, M., Santucci, G., Sartori, C.: Visual Data Mining System Architecture. Technical Report D3.R2 - D2I, Integration, Warehousing, and Mining of Heterogeneous Data Sources, Italian MIUR Project, `http://www.dis.uniroma1.it/~lembo/D2I` (2001)
3. Domshlak, C., Gershkovich, D., Gudes, E., Liusternik, N., Meisels, A., Rosen, T., Shimony, S. E.: FlexiMine - A Flexible Platform for KDD Research and Application Construction. Technical Report FC9804, BenGurion University (1998)
4. The Data Mining Group: PMML 2.0 - Predictive Model Markup Language `http://www.dmg.org/pmmlspecs_v2/pmml_v2_0.html`
5. SGI `http://www.sgi.com`

6. Accrue Software Inc. http://www.accrue.com
7. SGI http://www.sgi.com
8. SPSS: Clementince http://www.spss.com/clementine
9. IBM: Quest http://www.almaden.ibm.com/cs/quest
10. Swayne, D. F., Cook, Buja, A.: XGobi - Interactive dynamic graphics in the X Window System with a link to S, Proceedings of the Section on Statistical Graphics. American Statistical Association, 1992.
11. GGobi Data Visualization System http://www.ggobi.org
12. Eudaptics Software gmbh http://www.eudaptics.com/technology/index.html
13. http://www-db.deis.unibo.it
14. http://www.deis.unical.it
15. Catarci, T., Santucci, G., Costabile, M. F., Cruz, I. F.: Foundations of the DARE system for Drawing Adequate Representations, Proceedings of the International Symposium on Database Applications in Non-Traditional Environments. IEEE Press, 1999.
16. Catarci, T., Santucci, G.: The prototype of the DARE System, Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data. ACM Press.
17. Baldonado, M., Chang, C.-C.K., Gravano, L., Paepcke, A.: The Stanford Digital Library Metadata Architecture. Int. J. Digit. Libr. **1** (1997) 108–121

A Post-processing Environment for Browsing Large Sets of Association Rules [1]

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

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

Abstract. Association rule engines typically output a very large set of rules. Despite the fact that association rules are regarded as highly comprehensible and useful for data mining and decision support in fields such as marketing, retail, medicine, demographics, among others, lengthy outputs may discourage users from using the technique. In this paper we propose a post-processing methodology and tool for browsing/ visualizing large sets of association rules. The method is based on a set of operators that transform sets of rules into sets of rules, allowing focusing on interesting regions of the rule space. Each set of rules can be then depicted with different graphical representations. The tool is web-based and uses SVG. The input set of association rules is given in PMML.

Keywords: Data mining, association rules, post processing, decision support, visualization.

1 Introduction

Association Rule (AR) discovery (Agrawal et al. 96) is many times used, for decision support, in data mining applications like market basket analysis, marketing, retail, study of census data, analysis of medical data, among others. This type of knowledge discovery is 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 possibly not measurable (like "what characterizes a good client?",

[1] 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.

"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).

Due to the data characterization objectives of the association rule discovery task, AR 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 of such an algorithm 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 (sometimes also called rule mining) facilities for inspecting the set of produced rules.

In this paper 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 for predictive models, PMML (Data Mining Group). The complete set of rules can then be browsed by applying rule set 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 through a number of techniques.

This paper is organized as follows: we start by introducing the basic notions related to association rule discovery, and association rule space. We then describe PEAR, the post processing environment for association rules and its implementation. We describe the set of operators in more detail, show one example of the application of PEAR, compare with related work and conclude, also suggesting the next steps of our work.

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 the presence of 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 number of transactions that contain A. Each rule represents a pattern captured in a dataset. The support of the rule is the commonness of that pattern, while the confidence measures its predictive ability.

The most common algorithm for discovering AR from a dataset D is APRIORI (Agrawal et al. 96). 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 (Savasere et al.), DIC (Brin et al.) or SAMPLING (Toivonen), but that should produce exactly (in the case of SAMPLING it can be approximately) the same set of rules since the exact set of rules to produce is determined by the problem definition and the data.

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, corresponding each lattice to one particular itemset that appears as the antecedent, or to one itemset that occurs as a consequent. For example, the rule $\{a,b,c\} \rightarrow \{d,e\}$, belongs to two lattices: the one of the rules with antecedent $\{a,b,c\}$, structured by the generality relation over the consequent, and the lattice of rules with $\{d,e\}$ as a consequent, structured by the generality relation over the antecedents of the rules.

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

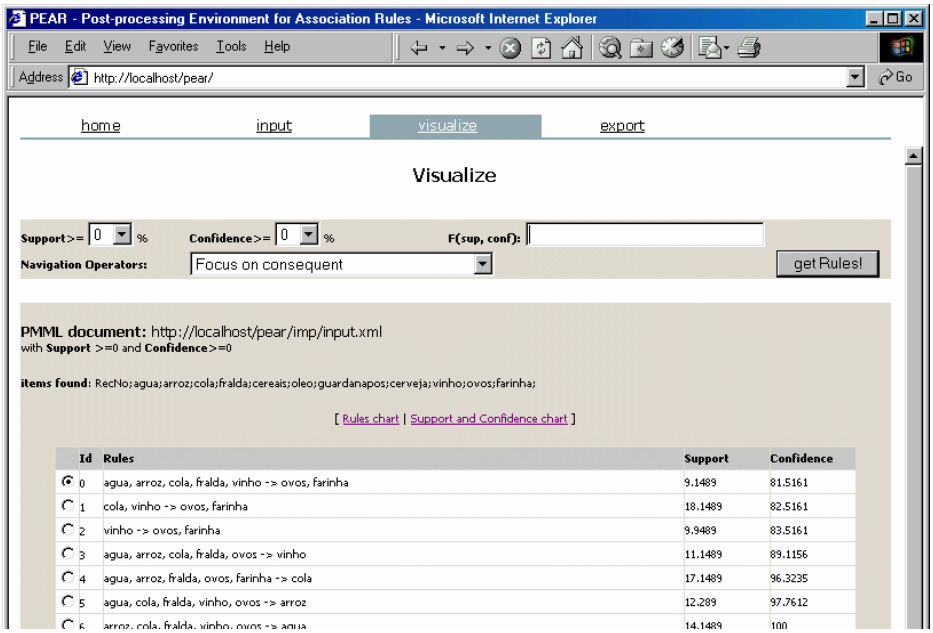


Figure 1: PEAR screen showing some rules.

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 Rules). 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 PEAR 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 1). From this page the user can go to other pages containing ordered lists of rules with support and confidence.

To move from page (set of rules) to page, the user applies restrictions and operators. The restrictions can be done on the minimum confidence, minimum support, or on functions of the support and confidence of the itemsets in the rule. Operators can be selected from a list. If it is a $\{Rule\} \rightarrow \{Sets\ of\ Rules\}$ operator, the input rule must also be selected.

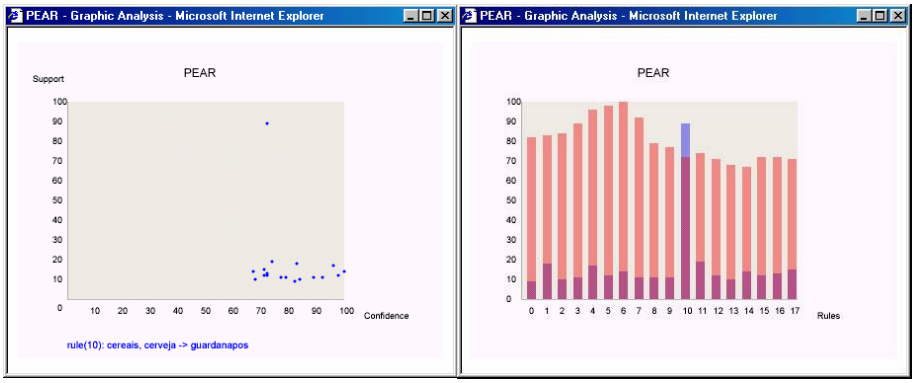


Figure 2: PEAR plotting support x confidence points for a subset of rules, and showing a multi-bar histogram.

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 plot and Confidence / support histograms (Figure 2). The produced charts are interactive and indicate the rule that corresponds to the point under the mouse.

4 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 interesting operators may transform one set of rules into another. In the following we describe the operators of the former class.

Antecedent generalization (AntG)

$AntG(A \rightarrow B) = \{A' \rightarrow B \mid A'$ is obtained by deleting one or more atoms in $A\}$

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. The support and confidence lines of the resulting set of rules allow the visual identification of items to prune in the antecedent. In terms of the antecedent lattice, it gives all the rules below the current one with the same consequent.

Antecedent least general generalization (AntLGG)

$AntLGG(A \rightarrow B) = \{A' \rightarrow B \mid A'$ 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)

$ConsG(A \rightarrow B) = \{A \rightarrow B' \mid B'$ is obtained by deleting atoms in $B\}$

Consequent least general generalization (ConsLGG)

$ConsLGG(A \rightarrow B) = \{A \rightarrow B' \mid B'$ 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)

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

This produces rules with lower support but higher confidence than the current one.

Antecedent least specific specialization (AntLSS)

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

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

Consequent specialization (ConsS)

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

Consequent least specific specialization (ConsLSS)

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

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

Focus on antecedent (FAnt)

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

Gives all the rules with exactly the same antecedent. $FAnt(R) = AntG(R) \cup AntS(R)$.

Focus on consequent (FCons)

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

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

5 The Index Page

Our methodology is based on the philosophy of web browsing, page by page following hyperlinks. The operators implement the hyperlinks between two pages. To start browsing, the user needs an index page. This should include a subset of the rules that summarize the whole set. 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. A candidate for such a set could be the, for example, 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.

6 One Example

We now describe how the method being proposed can be applied to browse through a set of association rules. The domain considered is the analysis of downloads done from the site of the Portuguese National Institute of Statistics (INE). This site (www.ine.pt/infoline) functions like 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. Although we show here how rules at the highest level of the products taxonomy, a similar study could be carried out for lower levels.

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

Figure 3: First page (index)

The rules in Figure 3 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
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

Figure 4: Applying the operator ConsG (consequent generalization).

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.

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 (Silbershatz & Tuzhilin).

7 Implementation

To develop this web environment we chose a Microsoft platform, due to the development background of the team, and also because of the possibilities offered in terms of XML development. This option does not compromise our goal of having a browser-free tool. Currently, all PEAR's features are supported in both Netscape and Internet Explorer. In the following sections we describe the main technologies involved in PEAR. The interactions are summarized in Figure 5.

7.1 Microsoft Internet Information Server

We use the Microsoft Internet Information Server (IIS) as PEAR's http server to run the Active Server Pages (ASP) for server-side programming, allowing database and XML manipulation and form data submitted by the user. PEAR also runs offline with no limitation under Microsoft Personal WebServer (Windows 95/98/2000/Me) or under Microsoft Peer Web Services (Windows NT Workstation). This means it can be installed in any PC with a Microsoft system in it (Windows 98/Me/NT/2000/XP).

7.2 Active Server Pages and VbScript

Active Server Pages (ASP) (Microsoft) are dynamic and interactive web pages processed on the server-side, thus useful to manipulate data submitted by users (for in-

stance, selecting a set of association rules given certain restrictions by the users) as well as manipulating database requests.

An ASP page integrates HTML tags with script commands. These scripts can be either VbScript or Jscript (JavaScript similar). Microsoft JScript is an open implementation of Netscape's JavaScript which are both compliant with the European Computer Manufacturing Association's ECMAScript Language Specification (ECMA-262 standard[2]) When the page is downloaded, these scripts are executed on the Active Server Page environment thus producing the final HTML code to the requesting browser.

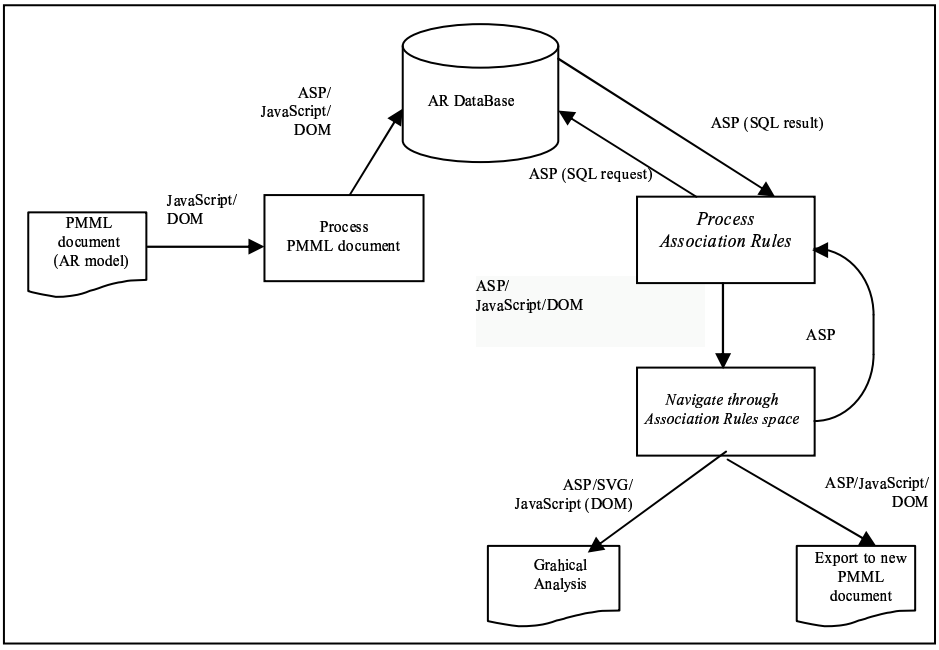


Figure 5: General architecture of PEAR.

PEAR uses VbScript (in Active Server Pages) to process the set of association rules represented in a PMML document (using Document Object Model), to allow the user to browse through it, and to store the rules in a relational database. VbScript is used at the server-side only.

7.3 JavaScript

JavaScript, (Microsoft Web Site) is used for data manipulation on the client-side, for its portability. We try to use only commands that are compliant with ECMAScript. This way PEAR may rub under Netscape and Internet Explorer. With JavaScript we create and manipulate PMML documents or SVG (both XML documents) using

[2] ECMA is an international industry association founded in 1961 and dedicated to the standardization of information and communication systems.

Document Object Model. JavaScript is also important for data validation and for interaction with the user (event handling).

7.4 Document Object Model

The Document Object Model (DOM) is a tree structure-based application program interface (API) for HTML and XML documents issued as a W3C Recommendation in October 1998 (W3C DOM Level 1 specification). It is used to process and manipulate an XML document by accessing its internal structure. DOM represents an XML document as a tree. Its nodes are elements, text, and so on. DOM makes it convenient for application programs to traverse the tree and access the contents of the tree. The Document Object Model provides a standard programming model for working with XML.

PEAR uses the Microsoft XML parser provided in Microsoft Internet Explorer 5 (and above), which implements the W3C DOM specification. With this parser we can easily access and manipulate the internal tree-structure of an XML document. In particular, 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 (SVG documents).

7.5 Scalable Vector Graphics

Scalable Vector Graphics (SVG) is an XML-based language that specifies and defines vector graphics that can be visualized by a web browser. [W3C Recommendation]. «…defines the features and syntax for SVG, a language for describing two-dimensional vector and mixed vector/raster graphics in XML». So, using SVG is very similar to working with any other normal XML document. An SVG document must also follow the DTD (Data Type Definition) that specifies the graphic elements that can be produced.

Again, we can manipulate SVG graphics with VbScript or JavaScript (using Document Object Model). With SVG, it is easy to produce a data visualization and even make it interactive (controlling keyboard or mouse events). PEAR gets data from PMML and presents it using VbScript and SVG graphics.

7.6 Database and SQL

We use a relational database (Microsoft Access) to store the PMML model and take advantage of using Structured Query Language (SQL) to obtain sets of association rules. Compared to using DOM directly to manipulate the original PMML document, SQL provides a faster and easier access. In PEAR, all database connections and requests are done with Active Server Pages on the server side.

7.7 Representing Associations Rules with PMML

Predictive Model Markup Language (PMML) is an XML-based language. A PMML document provides a non-procedural definition of fully trained data mining models with sufficient information for an application to deploy them. It provides a way for

people to share models between different applications. Like any XML document, also a PMML document must follow a Data Type Definition (DTD) that defines the entities and attributes for documenting a specific data mining model. For instance, there is one DTD to specify a Regression model; another DTD to represent a Naive Bayes model; other to define an AR model and so on. Any AR model written in PMML by different entities must follow the same AR specific DTD.

A model described using PMML has the following structure:

1) A header,
2) A data schema,
3) A data mining schema,
4) A predictive model schema,
5) Definitions for predictive models,
6) Definitions for ensembles of models,
7) Rules for selecting and combining models and ensembles of models,
8) Rules for exception handling.

Component (5) is required. The other components are optional.

The main reasons that drove the formulation of the PMML for predictive models were that it must be universal, extensible, portable and human readable. It allows users to develop models within one vendor's application, and use other vendors' applications to visualize, analyze, evaluate or otherwise use the models. Previously, this was virtually impossible, but with PMML, the exchange of models between compliant applications now will be seamless. At this moment, only a few data mining tools and applications allows to export their models to PMML, but is urgent to implement it in other software tools to satisfy dramatically increasing requirements for statistical and data mining models in business systems.

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.

8 Related Work

There is some work on the visualization and summarization of association rules. In this section we refer to selected work on theme.

The system DS-WEB (Ma et al.) uses the same sort of approach as the one we propose here. In common, DS-WEB and PEAR have 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, however, 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.

VizWiz is the non-official name for a PMML interactive model visualizer implemented in Java (Wettshereck). It graphically displays, not only association rules, but

many other data mining models. The philosophy of WizWiz for displaying AR relies on the presentation of the list of rules, allowing the user to set the minimal support and confidence through very intuitive gauges. VizWiz also accompanies the display of each rule by color bars representing support and confidence. This visualizer can be used directly in a web browser as a java plug-in.

(Lent et al 97) 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 paper we describe a rule post processing environment that allows the user to browse the rule space, 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 summarize 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 AR rules by an user is an implicit form of providing background knowledge, that can be later used, for example, in selecting rules for a a classifier made out of a subset of rules.
- PEAR presents an open philosophy by reading the set of rules as a PMML model.

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.
- The entry point (the index page) is still relatively weak.
- The visualization techniques offered are very limited.

Future work:

- Develop metrics to measure the gains of this approach.

- Develop mechanisms that allow the incorporation of user defined visualizations and rule selection criteria, such as for example, the combination of primitive operators.
- Evaluate the current implementation against other alternatives such as java, as well as an alternative to client-server, such as plug-in.
- Investigate and implement other visual representations of subsets of rules.
- Allow the definition of rule selection criteria based on the support and confidence of the rule, its antecedent and its consequent.

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. Brin, S., Motwani, R., Ullman, J. D. and 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
3. Data Mining Group (PMML development), http://www.dmg.org/
4. ECMA-262 standard http://www.ecma.ch/ecma1/STAND/ECMA-262.HTM
5. Lent, B., Swami, A., Widom, J.: Clustering Association Rules, in Alex Gray, Per-Åke Larson (Eds.): *Proc. of the Thirteenth International Conference on Data Engineering, ICDE 97* Birmingham U.K. IEEE Computer Society 1997
6. Ma, Yiming, Liu, Bing, Wong, Kian (2000), Web for Data Mininig: Organizing and Interpreting the Discovered Rules Using the Web, School, *SIGKDD Explorations*, ACM SIGKDD, Volume 2, Issue 1, July 2000.
7. Microsoft Web Site (ASP) http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbegvb/html/activeserverpages.asp
8. Microsoft Web Site (Descriptions of Java, JScript, and JavaScript) http://support.microsoft.com/default.aspx?scid=kb;EN-US;q154585
9. Savasere, A., Omiecinski, E. and Navathe, S., An efficient algorithm for mining association rules in large databases . *Proc. of 21st Intl. Conf. on Very Large Databases (VLDB)*, 1995.
10. Silberschatz, A. and Tuzhilin, A., On subjective measures of interestingness in knowledge discovery. *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, 1995, 275-281. http://citeseer.nj.nec.com/silberschatz95subjective.html
11. Toivonen, H., Sampling large databases for association rules. *Proc. of 22nd Intl. Conf. on Very Large Databases (VLDB)*, 1996. http://citeseer.nj.nec.com/toivonen96sampling.html
12. W3C DOM Level 1 specification http://www.w3.org/DOM/
13. W3C, Scalable Vector Graphics (SVG) 1.0 Specification, W3C Recommendation, September 2001, http://www.w3.org/TR/SVG/
14. Wettshereck, D., A KDDSE-independent PMML Visualizer, in *Proc. of IDDM-02, workshop on Integration aspects of Decision Support and Data Mining*, (Eds.) Bohanec, M., Mladenic, D., Lavrac, N., associated to the conferences ECML/PKDD 02, Helsinki, Finland, 2002.

Visual Post-Analysis of Association Rules[1]

Dario Bruzzese and Cristina Davino

Department of Mathematics and Statistics
University of Naples Federico II
Via Cintia Monte S. Angelo
I-80126 Naples, Italy
{dbruzzes, cdavino}@unina.it

Abstract. Association Rules (AR) represent a consolidated tool in Data Mining applications as they are able to discover regularities in large data sets. The information mined by the rules is very often difficult to exploit because of the presence of too many associations where to detect the really relevant logical implications. In this framework, by combining methodological and graphical pruning techniques, AR post-analysis tools are proposed. The methodological techniques will ensure the statistical significance of the AR which were not pruned while the graphical ones will provide interactive and powerful visualization tools.

1 Introduction

The post analysis is one of the most crucial steps in the knowledge extraction process, mainly when data mining is performed through Association Rules (AR) [1]. Even if mining rules is a quite simple task, their analysis and interpretation is often difficult due to the huge number of rules that can not be manually inspected.

The main approaches used to face this problem are graphical tools and pruning methods. AR visualization tools proposed in literature [5] [9] [14] allow to obtain a global view of all the discovered rules but they are still lacking because a huge number of rules is displayed and most of them are uninteresting. On the other hand, pruning methods [8] [12] [13] [10] [11] allow to kill the redundant rules but the pruned set still requires graphical tools to be interpreted.

In order to exploit the synergic power of both the visualizing and pruning phase, we propose two different strategies to help the user analyzing and interpreting AR.

The first strategy, "Display and Prune", gives priority to the visualization phase that becomes an interactive tool for pruning unuseful and redundant rules. In the second strategy, "Prune and Display", the capability of Factorial Methods [2] to synthesize and to visualize multidimensional patterns is exploited on those rules survived to a pruning process, which is based on statistical tests.

[1] This research was partially supported by "Data Mining e Analisi Simbolica" COFIN2000 grant (Prof. C. Lauro).

The two strategies will be applied on a real data set provided by RAI regarding the Italian national television channel preferences.

The outline of the reminder of this paper is as follows. In section 2 we introduce Association Rules and some basic notations. Section 3 formalizes the proposed visual post-analysis strategies from the methodological and application point of view. Some concluding remarks and further developments are summarized in section 4.

2 Association Rules: some basic notations

Let $I = i_1, i_2, \ldots, i_m$ be a set of items, called literals, (e.g. all products bought by a group of customers), and $T = t_1, t_2, \ldots, t_n$ be a set of n transactions, where each transaction t_i is a subset of I (e.g. all products in a customer's basket).

An association rule R is an implication of the form: $A \rightarrow C$, where $A \subset I$ is the set of the antecedent items of the rule and $C \subset I$ is the set of the consequent items of the rule such that $A \cap C = \emptyset$. Each rule of the form $A \rightarrow C$ is characterized by two ratios:

- $Support : S_R = \frac{n_R}{n}$ where n_R is the number of transactions in T holding $A \cup C$;
- $Confidence : C_R = \frac{n_R}{n_A}$ where n_A is the number of transactions in T holding the antecedent items A.

The *Support* measures the proportion of transactions in T containing both A and C and it is not related to the possible dependence of C from A. On the other hand, the *Confidence* aims at measuring the strength of the logical implication described by the rule and it refers to the conditional probability of the consequent given the antecedent. The concept of Support can be also referred to a generic item set if the proportion of transactions sharing the item set is considered.

Usually minimum support and minimum confidence values are fixed by the user before mining association rules. These values are generally user-dependent and unproper choices 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 [13]. Another problem is the strength of the associations being commonly evaluated only by means of the confidence values and no assessment of the statistical significance of these values is made. Furthermore, even if the visualization of the rules has to be framed in a multidimensional context, most of the solutions proposed in literature force their representation in two-dimensional grids without considering the interaction among them.

3 Visual post-analysis of Association Rules

The visual post-analysis approach proposed in this paper aims at exploiting the synergic power of both the visualizing and the pruning phases in order to improve

the profitability of the discovered association rules. The approach is structured into two concurrent strategies where the graphical and the pruning phases have different priorities.

The first strategy, "Display and Prune", is based on the use of parallel coordinates in order to visualize the discovered rules; each item is a dimension of the graph and it spans according to the utility it provides to the rule. The previous utility is defined by an index called Item Utility (IU) able to take into account the confidence of the rule with or without each item. The user can visually inspect the rules and prune those ones whose items are below a specified IU threshold.

In the second strategy, "Prune and Display", a pruning method [4] based on three statistical tests is followed by the introduction of Factorial Methods in order to synthesize the information stored in the rules and to represent themselves, the items and their interactions on 2-D graphs [3].

The two strategies will be applied on a real dataset provided by RAI (Italian public television) regarding the channel preferences of the official users panel (9965 units) collected by the RAI in order to study television customers behaviors. The preferences refer to 39 different typologies of television programs (variety show, cartoons, musical, sports time, religious time, etc.) and each user is described by the sub-set of genres he has watched for more than five minutes per day during a week (a threshold equal to five minutes allows to avoid users that frequently change channel). In appendix, the list of the considered genres is reported.

The aim of the application is to explain the television customers behaviour through the discovery of the logical associations among the various typologies of television programs in the set of 9965 transactions. Considering only those rules with at most three items in the antecedent and one item in the consequent, fixing very low support (0.01) and confidence values (0.01), a huge number of rules (35783) is obtained.

3.1 The "Display and Prune" strategy

One of the input parameters of association rules mining algorithms is represented by the number of different items in the association, defined as the order of the rule. It can happen that not all the antecedent items give a real contribute to the rule confidence value but if those items are drawn away and a lower order rule is considered, the rule confidence can improve. Viceversa, increasing the order of the rule by adding an item in the antecedent part can significantly enhance the confidence value if the added item is very useful in explaining the consequence. The two previous cases are graphically represented in figure 1 where each rectangle describes an item and its surface is proportional to the item support. From figure 1(a) it results that the presence of item B is relevant because in case of its absence (1(c)) the confidence value reduces while from figure 1(b) it results that by drawing away B from the association, the different interactions among the items cause a rise in the confidence value.

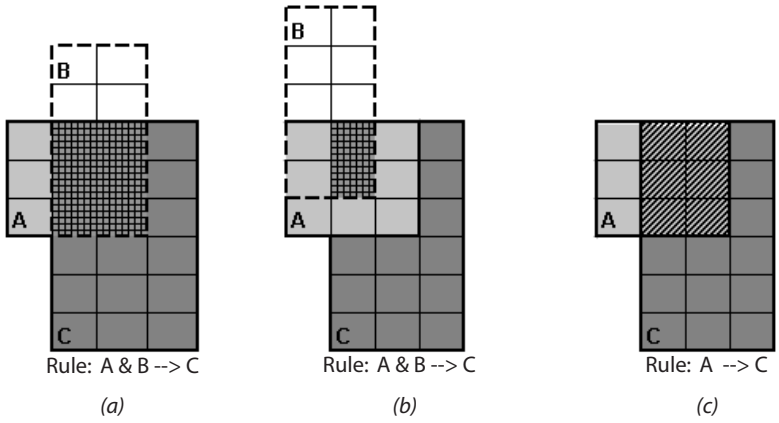


Fig. 1. A graphical representation of the item B utility in the rule $A \ \& \ B \rightarrow C$

In order to measure the real utility of an item i in the premise of a rule R, we introduce an index called *Item Utility* (IU) based on the comparison between the confidence of the rule with or without item i:

$$IU_i = \frac{C_R - C_{R(-i)}}{max(C_R; C_{R(-i)})} \tag{1}$$

If $IU_i \in \ ]-1;0[$, the item is not useful but dangerous and the rule can be pruned as a lower order rule is the real association; the case $IU = 0$ refers to the presence of a redundant item because $A \subseteq B$, the confidences of the rules $A\&B \rightarrow C$ and $A \rightarrow C$ are equal and the real association is described by the rule $A \rightarrow B$; if $IU_i \in \ ]0;1[$, the item is very useful in the rule as it improves the capability of the antecedent to explain the consequence.

The discovered association rules are visualized plotting on parallel coordinates the IU of each item belonging to the antecedent of a rule. Parallel coordinates, introduced by Inselberg in 1981 [6], represent a very useful graphical tool fort the visualization of 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. In the proposed parallel visualization of association rules , each antecedent item is a dimension of the graph and it spans according to the utility provided to each rule.

Some of the interaction tools of parallel coordinates [7] 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 belove a specified IU threshold in order to remove selected lines from the plot;

- identifying axes (items) with very dense positive values, given a consequent, 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 IU distributions.

The "Display and Prune" strategy has been applied to the rules discovered on the RAI dataset. In figure 2, a coordinate plot of the utilities of the items in the 1652 rules sharing the same consequent (FLC: series) is provided. The lines corresponding to rules with the item INP (parliamentary news), that has very dense negative IU values, are highlighted in black. This set of 96 rules can be pruned because lower order but more explicative rules will remain.

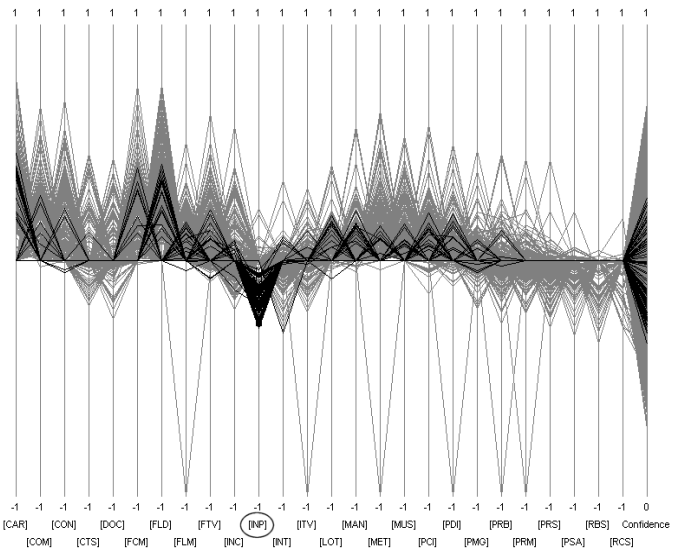


Fig. 2. A parallel coordinate plot of the rules with the consequent equal to FLC.

In figure 3, a subset of 1084 rules with the consequent equal to FLC and with the confidence greater than 0.5 (the choice of this threshold will be justified in Section 4.2) is shown. It is worth of noticing that a huge number of rules with one or more items having negative values for IU still survive thus confirming the guess that high confidence does not ensure the explicative power of each item involved in the rule. A further consideration regards items with high IU values that are still present after the confidence filter and representing very important dimensions in strong associations.

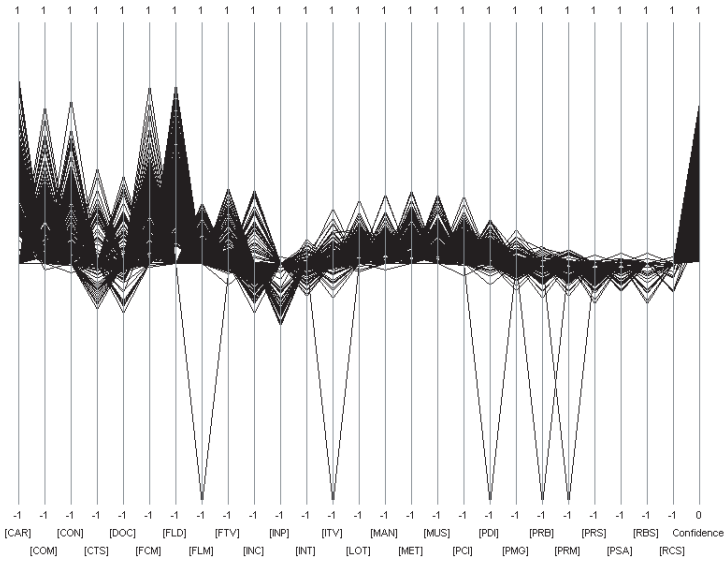


Fig. 3. Plot of the rules with the consequent equal to FLC and $C_R > 0.5$.

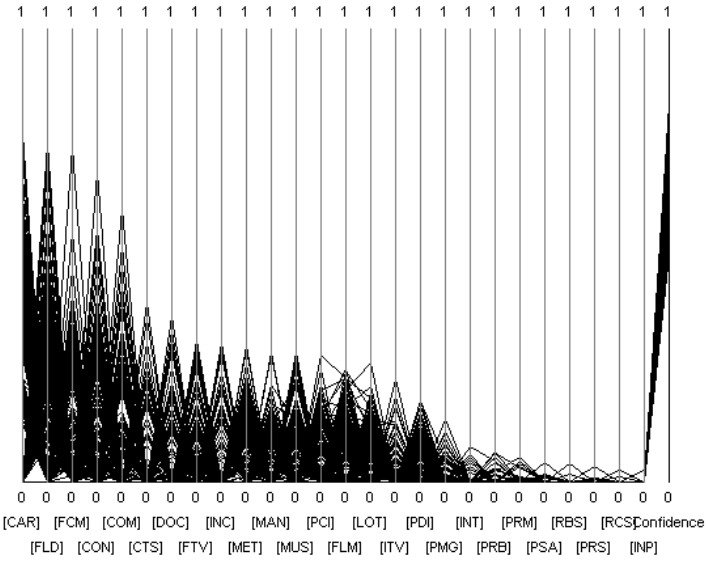


Fig. 4. Plot of the rules survived to the confidence and IU pruning.

Finally, the 463 rules with confidence greater than 0.5 and positive IU for each item in the antecedent are plotted in figure 4. The possibility of reordering the dimensions of the parallel graph on the basis of IU values allows to visually and immediately capture the most relevant items for the analyzed consequent and to face one of the main limits of these graphs which is represented by the arbitrary variables ordering.

3.2 The "Prune and Display" strategy

The "Prune and Display" strategy follows the PEV (**P**runing, **E**xploring, **V**isualizing) approach proposed by the authors in [3] that aims firstly at allowing the user finding automatically a reduced subset of rules, secondly at capturing the most relevant structure inside the pruned set and, finally, at providing graphical tools able to represent the rules, the items and their interactions on 2-D graphs .

AR obtained by a mining process with very low support and confidence values are sequentially pruned using three statistical tests performed on the significance of the consequence, of the antecedent and of the confidence. At each step, the rules are ranked according to the corresponding test statistic and a subset of rules is obtained by pruning the rules out of a suitable threshold. This subset becomes the rules input set for the next step.

The three tests are described in the following.

A test on the significance of the consequence.
Rules whose high confidence is only related to the presence of very frequent items are pruned through a test performed comparing the rule confidence with the support of the consequent part of the rule. This test allows to evaluate how much the antecedent part is able to explain the variability of the consequent part.

A test on the significance of the antecedent.
A chi-2 test is performed for each group of rules given the same antecedent part in order to evaluate how significant the logical implication is and to avoid meaningless associations. The test is based on the comparison between the support of each rule of the group and the theoretical support in case of casual associations with the given antecedent part.

A test on the significance of the confidence.
A test on the significance of the confidence is necessary in order to evaluate how strong, from a statistical point of view, the logical implication between A and C is. The test proposed in [4] is based on the comparison between the rule support and the support of the antecedent part of the rule that means evaluating how much the confidence is far from an strong implication. A "soft" version of this test is now introduced by comparing the rule support with the proportion of transactions sharing the antecedent items but not the consequent. This comparison allows to find rules with a confidence not significantly far from a "soft" implication ($C_R = 0.5$).
The proposed "soft" test is based on the following hypothesis:

$$H_0 : S_R = S_A - S_R \qquad H_1 : S_R > S_A - S_R$$

deriving from the assumption that the association between A and C, measured by the support of the rule (S_R), should be at least equal to the association between A and $\bar{C}$ measured by $S_A - S_R$. After simple algebraic manipulations, the test hypothesis becomes as follows:

$$H_0 : C_R = 0.5 \qquad H_1 : C_R > 0.5$$

and, under H_0, C_R is a binomial random variable that can be approximated with a normal distribution:

$$C_R \sim N\left(0.5; \sqrt{\frac{0.5 \cdot (1 - 0.5)}{n_A}}\right)$$

if n_A is sufficiently large.
It follows that the test statistic $V_{ConfSoft}$ is a standardized normal random variable:

$$V_{ConfSoft} = \frac{C_R - 0.5}{\sqrt{\frac{0.5 \cdot (1-0.5)}{n_A}}} \sim N(0; 1).$$

The results of the pruning phase applied to the 35783 rules mined from the RAI data set are summarized in the following table.

Table 1. Information about the pruning process.

	Before Pruning	After Step 1	After Step 2	After Step 3
Number of rules	6901	32872	10649	1562
Minimum Confidence	0.06	0.09	0.09	0.53
Maximum Confidence	1	0.87	0.85	0.85
Minimum Support	0.01	0.01	0.01	0.01
Maximum Support	0.15	0.15	0.15	0.15

After the first step, about 10% of the rules are pruned without influencing significatively the support and confidence ranges. The second step allows to prune a huge number of rules (22232) as it works on groups of rules with the same antecedent items. Using the "soft" test in the third step, 1562 rules survive with a minimum confidence equal to 0.53 compared to the 8536 associations with a confidence greater than 0.53 in the original set of mined rules. This gap is due to the nature of the used pruning approach that allows to kill also rules whose confidence, although high, does not ensure a real logical implication.

The 1562 survived associations still represent a huge number of rules to be manually inspected by the user so that performing a factorial method can allow to synthesize the information and to visualize the associations structure on 2-dimensional graphs. A Multiple Correspondence Analysis (MCA) [2] is applied

to an $n \times (p+2)$ data matrix where n represents the number of rules survived to the pruning steps and p corresponds to the total number of different items both in the antecedent part and in the consequent part of the n rules; support and confidence values are also considered. 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. Different roles are assigned to the $p+2$ variables in the MCA: the antecedent items are the *active* variables (variables that intervene directly in the analysis and define the factorial planes) while the consequent items and the support and the confidence values are the *supplementary* variables (variables depending by the former and projected later on the defined factorial planes). This choice is related to the logical link that exists between the two components of a rule where the antecedent part represents the logical premise of the consequent part.

Once the MCA is performed, it is possible to represent either the factorial planes allowing to explain at least a user defined threshold of the total variability either a user defined factorial plane or the factorial plane best defined by a user chosen item.

One of the possible views offered by MCA allows to graphically represent the association structures among the antecedent and the consequent items.

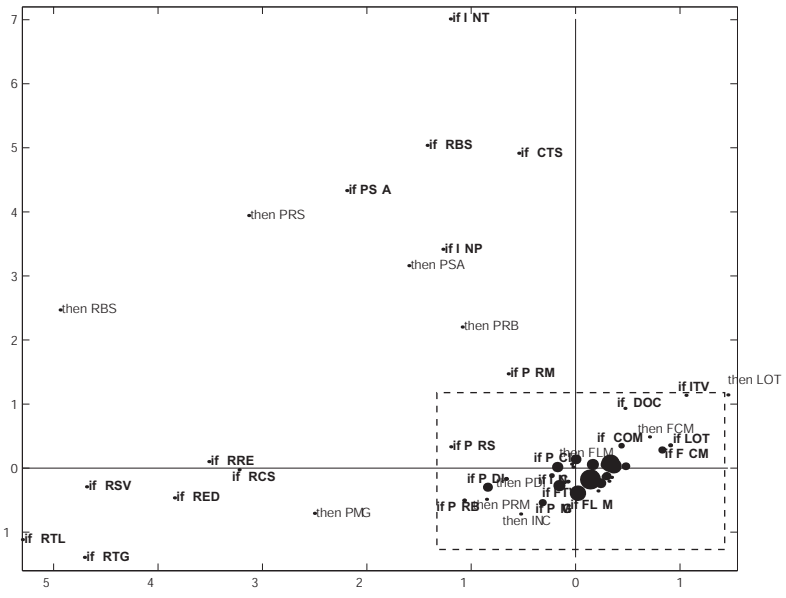


Fig. 5. The Item Representation.

In figure 5 the antecedent and the consequent items are plotted on the first factorial plane with a dimension proportional to the number of rules sharing them and in figure 6 a zoom of the selected area is shown. The support and

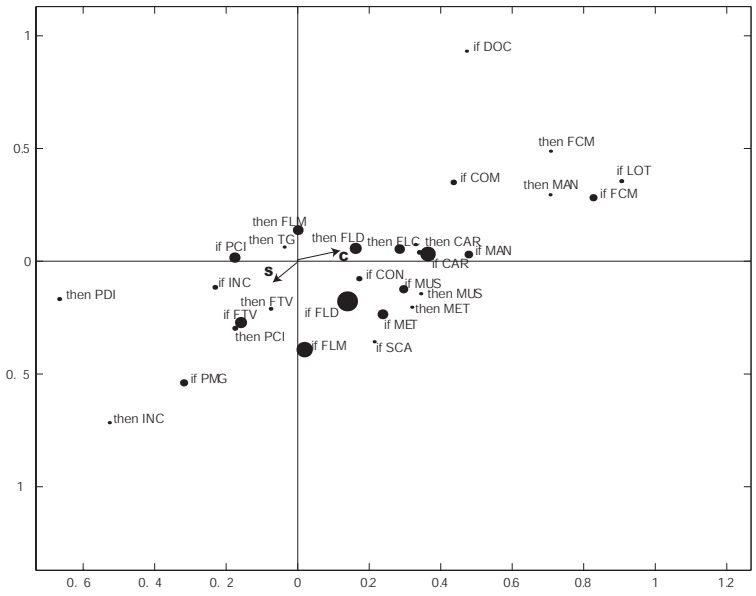


Fig. 6. A zoom of the Item Representation.

the confidence measures can be represented by oriented segments linking the origin of the axes to their projection on the plane as their coordinates are the correlation coefficients with the axes. The previous expedient allows to identify privileged regions in the plane with high supports and confidences: in figure 5 the first quadrant contains items associated to high confidence but low support rules. 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. For example, the consequent item FLC results very close to antecedent items such as *Cartoons* (CAR), *Comic programme* (COM), *Musical* (MUS), etc thus identifying genres that contribute to explain the same consequent. This result confirms the interpretation of figure 4 where the aforementioned items showed very dense IU positive values.

Once the common structure has been grasped in the item visualization, it is possible to come back to the associations by plotting, on the factorial plane, the rules with a dimension proportional to their confidence. In this representation, the proximity among two or more rules gives evidence to the presence of a common structure of the antecedent items associated to different consequences and it allows to change the set of close rules into a higher order macro-rule by linking the common antecedent items to the logical disjunction of the different consequent items.

4 Concluding remarks

In this paper the Association Rules post-processing problem is faced in order to both guarantee statistical significance of mined implications and to make them usable and interpretable through interactive graphical tools. The priority given to the methodological pruning or to the graphical representation depends on the data analyst requirements and it leads to the choice between the "Display and Prune" and the "Prune and Display" strategies.

The modular feature of the proposed approaches gives the possibility to differently combine them, e.g. to plot on parallel coordinates the rules sub-set survived to the statistical pruning or to synthesize the rules with high IU items by MCA.

The calculation of the measures involved in the two post-analysis tools (IU values and test statistics values) has very low computational costs as it is based on the use of information already computed during the mining process.

Further developments will regard the statistical evaluation of the IU measure to introduce an objective threshold in order to identify the items characterized by the worse or best IU values and the analysis of the validation and visualization issues in case of Classification Rules.

References

1. Agrawal, R., Imielinski, T. & Swami, A.: Mining Association Rules between Sets of Items in Large Databases, Proceedings of the 1993 ACM SIGMOD Conference, May, Washington DC, USA, (1993) 207–216.
2. Benzècri, J.-P.: L'Analyse des Donnèes, Dunod, Paris (1973).
3. Bruzzese, D. & Davino, C.: Pruning, Exploring and Visualizing Association Rules, Statistica Applicata, vol. 12, n. 4, (2000) 461–472 .
4. Bruzzese, D. & Davino, C.: Statistical Pruning of Discovered Association Rules, Computational Statistics, vol. 16 (2001) 387–398.
5. 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, Physica-Verlag, Heidelberg (2000) 499–504.
6. Inselberg, A.: N-dimensional Graphics, part I - Lines and Hyperplanes, in IBM LASC Tech. Rep. G320-2711, 140 pages. IBM LA Scientific Center (1981).
7. Inselberg, A.: Visual Data Mining with Parallel Coordinates, Computational Statistics, vol. 13, n.1, (1998) 47–64.
8. Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., & Verkamo, A.I.: Finding interesting rules from large sets of discovered association rules, Proceedings of the Third International Conference on Information and Knowledge Management CIKM-94, (1994) 401–407.
9. Liu, B., Hsu, W. & Ma, Y.: Pruning and Summarizing the Discovered Associations, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-99), August 15-18, San Diego, CA, USA (1999).
10. Liu, B., Hsu, W., Wang, K. & Chen, S.: Visually Aided Exploration Interesting Association Rules. Proceedings of the Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD-99). Springer Eds., April 26-28, Beijing (1999).

11. Shah, D., Lakshmanan, L.V.S., Ramamritham, K. & Sudarshan S.: Interestingness and Pruning of Mined Patterns, Workshop Notes of the 999 ACM SIGMOD Research Issues in Data Mining and Knowledge Discovery (1999).
12. Toivonen, H., Klemettinen, M., Ronkainen, P. , Hatonen, K. & Mannila, H.: Pruning and grouping of discovered association rules. Workshop Notes of the ECML-95 Workshop on Statistics, Machine Learning, and Knowledge Discovery in Databases, Heraklion, Greece, April 1995 (1995) 47–52.
13. Weber, I.: On Pruning Strategies for Discovery of Generalized and Quantitative Association Rules. Proceedings of Knowledge Discovery and Data Mining Workshop, Singapore (1998).
14. Wong, P.C., Whitney, P., Thomas, J.: Visualizing Association Rules for Text Mining. In Wills, G., Keim, D. (eds.): Proceedings of IEEE Information Visualization '99, IEEE CS Press, Los Alamitos, CA (1999).

Appendix: Television genres legend

Table 2. TV genres.

CAR: cartoons	COM: comic programme	CON: concert
CTS: customs and society	DOC: documentary	FCM: short
FLC: series	FLD: film with discussion	FLM: film
FTV: TV film	INC: survey	INP: parliamentary news
INT: suspension	ITV: break	LOT: lottery
MAN: demonstrations	MET: weather-forecast	MUS: musical
PCI: showing	PDI: discussion	PMG: montage
PRB: children programme	PRM: advertising	PRS: coming shortly
PSA: drama	RBS: sport programme	RCS: school programme
RED: editorial	RRE: religious magazine	RSV: documentary programme
RTG: tv news programme	RTL: current affairs	RUB: programme
SCA: Science	SCN: serial	TG: news
TLD: teaching programme	TLF: TV film	TQZ: quiz show

Cooperation between automatic algorithms, interactive algorithms and visualization tools for Visual Data Mining

François Poulet

ESIEA Recherche
38, rue des Docteurs Calmette et Guérin
Parc Universitaire de Laval-Changé
53000 Laval, France
poulet@esiea-ouest.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 strengths from multiple sources. This paper presents concrete cooperation between automatic algorithms, interactive algorithms and visualization tools. The first kind of cooperation is an interactive decision tree algorithm called CIAD+. It allows the user to be helped by an automatic algorithm based on a support vector machine (SVM) to optimize the interactive split performed at the current tree node or to compute the best split in an automatic mode. This algorithm is then modified to perform an unsupervised task, the resulting clustering algorithm has the same kind of help mechanism based on another automatic algorithm (the k-means). The last effective cooperation is a visualization algorithm used to explain the results of SVM algorithm. This visualization tool is also used to view the successive planes computed by the incremental SVM algorithm.

1 Introduction

Knowledge Discovery in Databases (or KDD) can be defined [1] 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. The user has only to tune some parameters before running his algorithm and wait for its results.

Some new methods have recently appeared [2], [3], [4], trying to involve more significantly the user in the data mining process and using more intensively the visualization [5], [6], this new kind of approach is called visual data mining. In this paper we present some tools we have developed, which integrate automatic algorithms, interactive algorithms and visualization tools. These tools are two interactive classification algorithms and a visualization tool created to show the

results of an automatic algorithm. The classification algorithms use both human 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 present our new interactive algorithms, the first one is an interactive decision tree algorithm called CIAD+ (Interactive Decision Tree Construction) using support vector machine (SVM) and the second is derived from the first one and performs unsupervised classification (clustering).

In section 3 we present a graphical tool used to explain the results of support vector machine algorithms. These algorithms are known to be efficient but they are used as "black boxes", there is no explanation of their results. Our visualization tool graphically explains the results of the SVM algorithm. The implemented SVM algorithm can modify an existing linear classifier by both retiring old data and adding new data. We visualize the successive separating planes computed by this algorithm.

Section 4 concludes the paper and lists some 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) [7], Decision Tree Visualization (DTViz) [8], [9] or CIAD [10]. 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 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.

CIAD is described in the next section and, in section 2.2, we present a new version of CIAD (called CIAD+) with a help tool added to the interactive algorithm allowing the user to perform an automatic computation of the best bivariate split.

[9] uses a two dimensional polygon or more precisely, an open-sided polygon (i.e. a polyline) in a two dimensional matrix. It is interactively drawn in the matrix. The display is made of one 2D matrix and one-dimensional bar graphs (like in PBC).

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

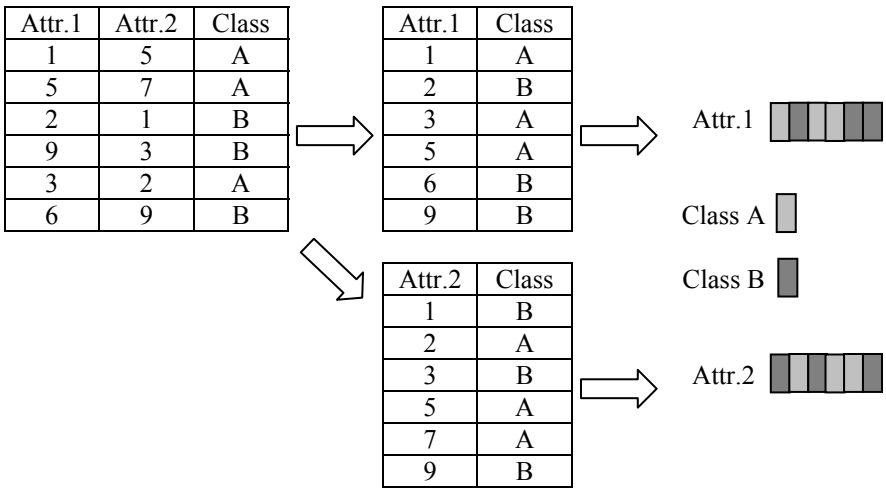


Fig. 1. Creation of the visualization bars with PBC

2.1 CIAD

CIAD uses a bivariate decision tree using line drawing in a set of two-dimensional matrices (like scatter plot matrices [11]). 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 [12], 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).

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 / highlight one class, one element or a group of elements.

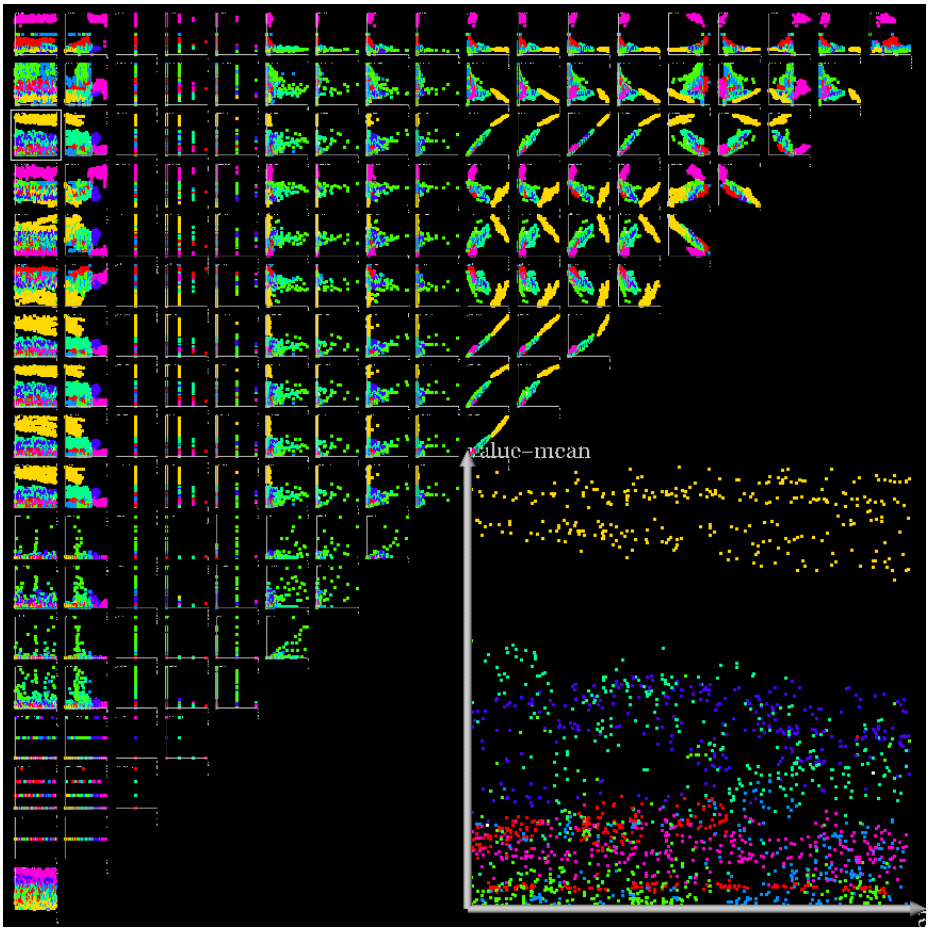


Fig. 2. The Segment data set displayed with CIAD

2.2 CIAD+

The first version of the CIAD algorithm was only an interactive algorithm. No help was available for the user, and sometimes, it was difficult to find the best pure partition in the set of two-dimensional matrices. We have decided to provide such help. Our first intention was to use a modified OC1 (Oblique Classifier 1) algorithm [13]: OC1 performs real oblique cuts (we have a real n-dimensional hyperplane with n-dimensional data) and in CIAD, the cuts are only "oblique" in two dimensions. The plane coefficients are null in all the other dimensions. We have made another choice:

we use a support vector machine (SVM). This algorithm is equivalent to OC1 in its simplest use, and will allow us to benefit from all its other possibilities for further developments.

2.2.1 The SVM algorithms

The SVM algorithms are kernel-based classification methods. They can be seen as a geometric problem: to find the best separating plane of a two classes data set. A lot of methods can be used to find this best plane, and a lot of algorithms have been published. A review of the different algorithms can be found in [14]. They are used in a wide range of real-world applications such as text categorization, hand-written character recognition, image classification or bioinformatics. We briefly describe here the basis of the algorithm, from the geometrical point of view.

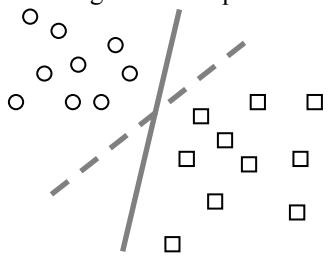


Fig. 3. Two possible separating planes

The aim of the SVM algorithm is to find the best separating plane between the n-dimensional elements of two classes. There are two different cases according to the nature of the data: they are linearly separable or not.

In the first case, the data are linearly separable i.e. there exists a plane that correctly classifies all the points in the two sets. But there are infinitely many separating planes as shown in Fig.3. Geometrically, the best plane is defined as being furthest from both classes (i.e. small perturbations of any point would not introduce misclassification errors). The problem is to construct such a plane. It has been shown in [15] that this problem is equivalent to finding the convex hull (i.e. the smallest convex set containing the points) of each class, and then to finding the nearest two points (one from each convex hull); the best plane bisects these closest points.

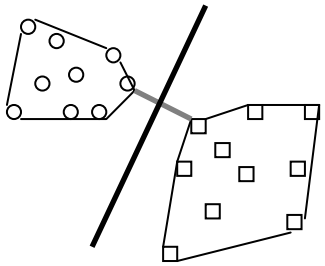


Fig. 4. The best separating plane bisects the closest points

In the second case, the data are not linearly separable (i.e. the intersection of the two convex hulls is not empty). There is no clear definition of what is the "best" plane. The solution is to create a misclassification error, and to try to minimize this error.

2.2.2 SVM in CIAD+

We use two different SVM algorithms in CIAD+. The first one is the geometric version described in section 2.2.1 for the linearly separable case. The convex hulls are computed in two dimensions with the quick hull algorithm [16], then the two closest points of the convex hulls are computed with the rotating calipers algorithm [17].

For the linearly inseparable case, a lot of solutions have been developed and compared, we have chosen one of these algorithms: the RLP (Robust Linear Programming) algorithm [18] because it is the best one when the data are not linearly separable.

The RLP algorithm will compute the separating plane wx=γ minimizing the average violations:

$$\frac{1}{m}\sum_{i=1}^{m}(-A_i w+\gamma+1)_+ + \frac{1}{k}\sum_{i=1}^{k}(B_i w-\gamma+1)_+ \qquad (\mathbf{1})$$

of points of *A* lying on the wrong size of the plane wx=γ+1, and of points of *B* lying on the wrong side of the plane wx=γ-1 as shown in Fig.5.

This algorithm computes the n-dimensional hyperplane, we have modified it to compute the best two-dimensional plane.

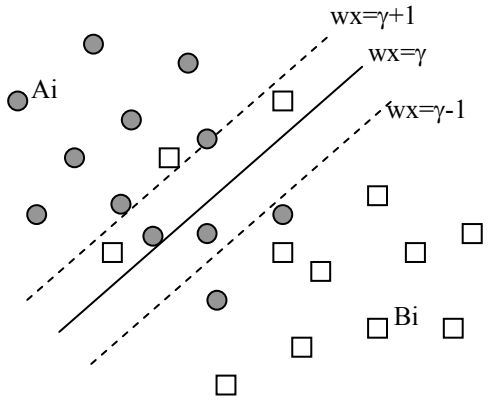


Fig. 5. Optimal plane for linearly inseparable data

SVMs in CIAD+ are used to help the user. The first kind of help is when the user draws interactively the separating line on the screen with a pure partition on one side, the optional help optimizes the line position to reach the best line position (furthest from both groups) with the computation of the closest points of the convex hulls. The second kind of help is the same case except there is no pure partition, the best line position is computed with the RLP algorithm in the two dimensions corresponding to the selected matrix. The last kind of help is used when the user cannot find a separating plane, the help algorithm has to compute the best separating plane among all the ones corresponding to the projections along pairs of attributes. So we compute all the separating lines in the 2D projections and we keep the best one.

This help mechanism can be turned on / off by the user. It slightly improves the accuracy of the results on the training sets (this result may be more significant according to the kind of user), more on the test set, and it considerably reduces the time needed to perform the classification and increases the ease of use. An example is shown on figure 6, the left part is the original line drawn interactively by the user on the screen and the right part shows the transformed line (the best separating plane computed with the convex hulls).

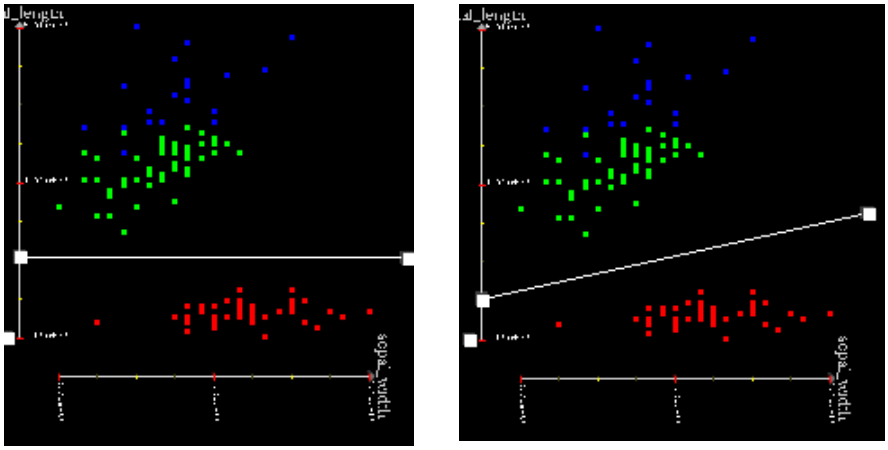


Fig. 6. An example of the automatic best separating plane on iris data set

2.3 Clustering

The interactive algorithm described in the previous section can also be used for unsupervised classification. The computation of the convex hulls and the nearest points can be computed with or without the class information. The proposed algorithm can perform either usual decision tree (supervised classification) or clustering (unsupervised classification). This kind of approach allows the user to perform clustering of the dataset easily using its pattern recognition capabilities and avoiding the usual complexity of the other algorithms. The same kind of help as for decision tree construction is provided to the user: the separating line drawn in a 2D projection can be optimized to be the furthest from the two clusters.

But there is one difference with the decision tree construction algorithm, when the user does not perceive clearly a separating line, this line can be computed automatically (with a modified SVM algorithm). This SVM algorithm cannot be used without the class information, so we have chosen a k-means algorithm. All possible partitions into two clusters are searched for in each matrix and the best one is kept. Then we compute the convex hulls and nearest points to find the best separating line in the same way as for decision tree construction.

As shown in [19], this kind of algorithm may not be very efficient for high dimensional data sets because axis-parallel projections may lead to an important loss of information. This restriction exists anyway because of the graphical representation we use (we cannot display a very large quantity of scatter plot matrices). On the other hand, the results are more comprehensible because we only use one attribute on each axis and not a linear combination of various number of attributes for the axes.

2.4 Some results of interactive algorithms

The characteristics of the different algorithms are summarized in table 1. Some results of interactive algorithms compared to automatic ones have been presented by their authors. To summarize them, we can say their results concerning efficiency, are generally at least as good as automatic decision tree algorithms such as CART (Classification And Regression Trees) [20], C4.5 [21], OC1 [13], SLIQ (Supervised Learning In Quest) [22] or LTree (Linear Tree) [23]. The main difference between the two kinds of algorithms is the tree size. Most of the time, interactive algorithms have smaller tree sizes. This tree size reduction can significantly increase the result comprehensibility. Furthermore, as the user is involved in the tree construction, his confidence in the model is increased too. This may be a little less significant for the LTree algorithm because of the open-sided polygon used in the classification: they are easy to understand during the visual construction step of the tree, but without this information, the resulting equations may be not so easy to understand.

	Vis. technique	split	+	-
Ware	bar graphs + 1 scatter plot	poly-line binary	large dataset tree size	result comprehensibility plot of qual. x qual. attr.
PBC	bar graphs	univariate n-ary	large datasets help	loss of information
DTViz	bar graphs	univariate n-ary	large datasets	loss of information
CIAD+	set of scatter plot	bivariate binary	tree size help	plot of qual. x qual. attr.

Table 1. Comparison of interactive decision tree algorithms

If we compare the interactive algorithms, we can say PBC and DTViz are particularly interesting for large data sets (because of the pixelisation technique used), but their pixelisation technique introduces some bias in the data: for example two classes very far one from the other have the same representation as the same two classes very near one to one other. We lose the distance information in this kind of representation. Ware and CIAD+ will provide smaller trees because they can use bivariate splits, but their kind of visualization tool (scatterplot matrices) are not at all suitable for the display of two qualitative attributes (a lot of points have the same projection). Only two algorithms provide the user with a help, PBC and CIAD+, this is a significant advantage of these two algorithms because during the decision tree

construction, there is often a least one particular step where the best split is difficult to visually detect.

The kind of cooperation between automatic and manual algorithms in PBC and CIAD+ shows the interest of mixing the human pattern recognition facilities and the computer processing power. The human pattern recognition facilities reduce the cost of the computation of the best separating plane, and the computer processing power can be used at low cost (for a single step, instead of the whole process) when the human pattern recognition fails.

3 Visualization of SVM results

Another kind of cooperation is between automatic algorithms and visualization tools used to show the results. As described in the previous section, SVM are today widely used because they give high quality results, but they are used as a "black box". They give high quality results, but there is no explanation of these results.

One paper [24] talks about SVM results visualization: they use projection-based tour [25] method to visualize the results. They use a visualization of the distribution of the data predicted class (by the way of histograms), a visualization of the data and the support vectors in 2D projection, and examine the weights of the plane coordinates to find the most important attributes for the classification.

The authors recognize that their approach is very "labor intensive for the analyst. It cannot be automated because it relies heavily on the analyst's visual skills and patience for watching rotations and manually adjusting projection coefficients."

3.1 Visualization of the SVM separating plane

Our approach is to visualize all the intersections of the 2D planes (of the scatter plot matrices) with the separating plane computed by the SVM algorithm. We have chosen to use the incremental SVM algorithm from [26]. This algorithm gives the coefficient values of the separating hyperplane and the accuracy of the algorithm. We visualize the intersection of this hyperplane with the 2D scatter plot matrices, i.e. a line in each matrix (as shown in Figure 7 with the diabetes data set, from the UCI repository).

As we can see on figure 7, the resulting lines do not necessarily separate the two classes, the hyperplane does separate the data (the accuracy of the incremental SVM is 77,8% on this dataset), but not its "2D projections". It is only an approximate interpretation of the results.

All 2D representations of one n-dimensional feature will lead to a loose part of the information like the lines we get here or the support vectors displayed in [24]. This kind of representation seems more comprehensible than the support vectors, but it can only be used with a linear kernel function.

For large data sets, it is possible to only display the separating plane and not the data. The SVM algorithm used is able to classify 1 billion points, and such a quantity of points cannot be displayed in a reasonable time (furthermore this kind of representation is not at all suitable for such data set size).

For other kinds of kernel functions (not linear), this method cannot be used.

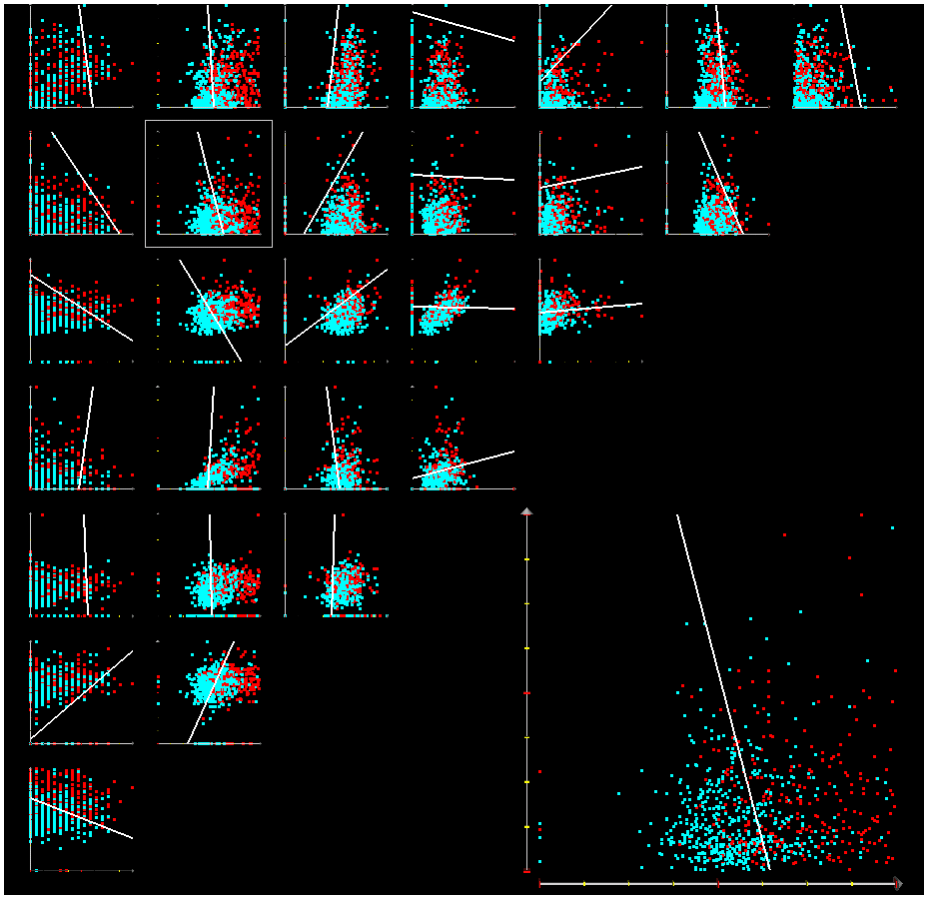


Fig. 7. Visualization of the separating hyperplane of the diabetes dataset.

3.2 Visualization of the evolution of SVM separating planes

A very interesting feature of the incremental support vector machine we have used is its capability to modify an existing linear classifier by both withdrawing old data and adding new data. We use our visualization tool to show the successive modifications of the separating plane projections. The first plane is calculated (and projected on the 2D matrices) and then blocks of data are successively added and withdrawn. The modification of the plane is computed and the corresponding projections are displayed in the matrices.

This kind of visualization tool is very powerful to examine the variations of the separating plane according to the data evolution. Even if the projections used lose some amount of information, we know what the attributes involved in the

modification of the separating plane are. The evolution of the n-dimensional plane is very difficult to show in another way. The authors of the paper describing the algorithm measure the difference between planes by calculating the angle between their normals. These angle values are then displayed like circle radii.

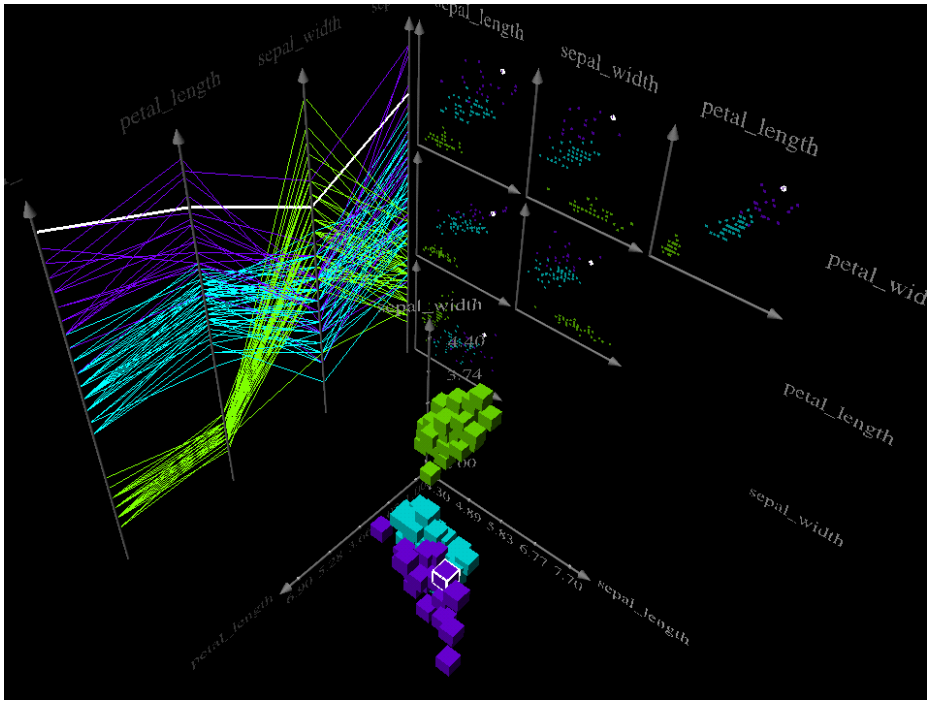


Fig. 8. Three linked tools in the FullView environment

4. 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 [27], where each tool can be linked to other tools and be added or removed as needed. Figure 8 shows an example with a set of 2D scatter plot matrices, a 3D matrix and parallel coordinates. The element selected in the 3D matrix appeared selected too in the set of scatter plot matrices and in the parallel coordinates. 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 and a visualization tool to explain the results of an automatic SVM algorithm. 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 visualization tool was created to help the user in understanding the results of an automatic SVM algorithm. These SVM algorithms are more and more frequently used and give efficient results in various applications, but they are used as "black-boxes". Our tool gives an approximate but informative graphical interpretation of these results.

A forthcoming improvement will be another kind of cooperation between SVM and visualization tools: an interactive visualization tool will be used to improve the SVM results when we have to classify more than two classes.

References

1. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy, Eds, "Advances in Knowledge Discovery and Data Mining", AAAI Press, 1996.
2. P.Wong, "Visual Data Mining", in IEEE Computer Graphics and Applications, 19(5), 20-21, 1999.
3. F. Poulet, "Visualization in data mining and knowledge discovery," in Proc. HCP'99, 10th Mini Euro Conference "Human Centered Processes" ed. P. Lenca (Brest, 1999), 183-192.
4. M.Ankerst, M.Ester, H-P.Kriegel, "Toward an Effective Cooperation of the Computer and the User for Classification" in proc. of KDD'2001, 179-188.
5. C.Aggarwal, "Towards Effective and Interpretable Data Mining by Visual Interaction", in SIKDD Explorations 3(2), 11-22, accessed from www.acm.org/sigkdd/explorations/.
6. B.Schneiderman, "Inventing Discovery Tools: Combining Information Visualization with Data Mining", in Information Visualization 1(1), 5-12, 2002.
7. M.Ankerst, "Visual Data Mining", PhD Thesis, Ludwig Maximilians University of Munich, 2000.
8. J.Han, N.Cercone, "Interactive Construction of Decision Trees" in proc. of PAKDD'2001, LNAI 2035, 575-580, 2001.
9. M.Ware, E.Franck, G.Holmes, M.Hall, I.Witten, "Interactive Machine Learning: Letting Users Build Classifiers", in International Journal of Human-Computer Studies (55), 281-292, 2001.
10. F.Poulet, "CIAD: Interactive Decision Tree Construction" in proc. of VIII Rencontres de la Société Francophone de Classification", Pointe-à-Pitre, 275-282, 2001 (in french).
11. J.Chambers, W.Cleveland, B.Kleiner, P.Tukey, "Graphical Methods for Data Analysis", Wadsworth (1983).
12. C.Blake, C.Merz, UCI Repository of machine learning databases, [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science, (1998).
13. S.Murthy, S.Kasif, S.Salzberg, "A system for induction of oblique trees", Journal of Artificial Intelligence Research 2, 1-32, 1994.

14.K-R.Muller, S.Mika, G.Ratsch, K.Tsuda, B.Scholkopf, "An Introduction to Kernel-Based Learning Algorithms" in IEEE Transactions on Neural Networks, 12(2), 181-201, 2001.
15.K.Bennett, E.Bredensteiner, "Duality and Geometry in SVM Classifiers", in proc of the Seventeenth International Conference on Machine Learning, Pat Langley Editor, Morgan Kaufmann, San Francisco, 57-64, 2000.
16.C.Barber, D.Dobkin, H.Huhdanpaa, "The Quickhull algorithm for convex hulls", in ACM Transactions On Mathematical Software, 22, 469-483, 1996.
17.G.Toussaint, "Solving geometric problems with the rotating calipers", in proc. of IEEE MELECON'83, Athens, Greece, pp. A10.02/1-4, 1983.
18.K.Bennett, O.Mangasarian, "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets", in Optimization Methods and Software, 1, 23-34, 1992.
19.C.Aggarwal, P.Yu, "Redifining Clustering for High-Dimensional Applications", in IEEE Transactions on Knowledge and Data Engineering, 14(2), 210-225, 2002.
20.L.Breiman, J.Friedman, R.Olsen, C.Stone, "Classification and Regression Trees", Wadsworth, (1984).
21.J.Quinlan, "C4.5: Programs for Machine Learning", Morgan-Kaufman Publishers, 1993.
22.M.Metha, R.Agrawal, J.Rissanen, "SLIQ: A fast scalable classifier for data mining", in proc. of the 5th International Conference on Extending Database Technology, Avignon, France, 1996, 18-32.
23.J.Gama, P.Brazdil, "Linear Tree" in Intelligent Data Analysis, 3, 1-22 (1999).
24.D.Caragea, D.Cook, V.Honavar, "Gaining Insights into Support Vector Machine Pattern Classifiers Using Projection-Based Tour Method", in proc. of KDD'2001 Workshop on Visual Data Mining.
25.D.Asimov, "The Grand Tour: A Tool for Viewing Multidimensional Data" in SIAM Journal of Scientific and Statistical Computing, 6(1):128-143, 1985.
26. G. Fung, O. Mangasarian, "Incremental Support Vector Machine Classification" in proc. of the 2nd SIAM International Conference on Data Mining, Airlington, USA, Apr.11-13, 2002.
27.F.Poulet, "FullView: A Visual Data-Mining Environment", in International Journal of Image and Graphics, 2(1), 127-144, 2002.

Defining Like-minded Agents with the Aid of Visualization

Penny Noy and Michael Schroeder

City University, London EC1V 0HB, UK
{p.a.noy, msch}@soi.city.ac.uk

Abstract. Profile carrying agents offer the opportunity of meeting like minds and increasing efficiency in many information search applications. Profiles can also increase the sophistication of relationships and interactions in multi-agent systems in general. Such profiles (feature lists) may be of the agent owner (interests) or of the information sought (specifications). At the same time there is a continuing increase of profiling data of increasing complexity becoming available.

The paper first describes various possibilities for defining profiles and selecting similarity metrics in the agent interaction context. General and domain-specific data sets are specially constructed and used to provide a concrete view of the behaviour of the metrics with visualization.

Visualization usually results in xy (or xyz) coordinates for each agent, placing them in a profile space. The use of these coordinates as a means of carrying and comparing one's profile without revealing it to other agents is proposed.

Thus this paper extends our previous work on visualizing multivariate data and proximity data in the agent scenario in two ways where visualization is not the end product: using visualization tools in the phase of similarity metric choice; proposing the use of coordinates as a profile. The paper seeks to illustrate, with these applications, an objective of visual data mining, namely the increased integration of visualization and analytic techniques.

1 Introduction

Every second that passes witnesses millions of people searching for information via a myriad of means. Increasingly such means and media employed are electronic. Many people are seeking similar information and work on similar problems. Software agents carrying our profile can meet with the representatives of others and exchange important information or provide introductions. Consider the following messages from some of my (hypothetical) personal agents:

myNetworkAgent: Good morning. Professor Blake in Helsinki has just started work on one of your main research areas.

myShoppingAssistant: Here are the fourteen houses closest to your ideal specification.

myInformationSearchAgent: Here are the four most relevant documents in your specified area 'agent matchmaking' from the survey point of view.

These agents may be providing real-time search responses or background information monitoring. The concepts are embodied in many search and classification applications. These may be ordinary searches, not explicitly involving other agents (or agents at all), but agents may assist in improving precision and speed.

Of course the above scenario is far fetched in several senses: the degree to which information is searchable (i.e. specified according to agreed ontologies); the accuracy of the metrics in matching seeker with sought; the willingness of humans to allow an agent to carry their profile (and the security issues involved); privacy of information.

The focus of this paper is the task of 'matching seeker with sought'. This may involve a user profile or a profile of a task or desired piece of information. It is in this general sense that the word profile is used here. To compare profiles a metric is needed. There are many examples of this in agent systems. For instance: Faratin et al use the maximum value distance in making negotiation trade-offs[10]; The Yenga system uses correlation to determine user interests and then a direct comparison to find a common joint interest [4]; Somlo and Howe use incremental clustering for profile maintenance in information gathering web agents based on term vectors of documents the user has shown interest in [14]; GRAPPA (Generic Request Architecture for Passive Provider Agents) is a configurable matchmaking framework which includes demand and supply profiles and has been applied to matching multidimensional profiles of job applicants to vacancies [18,17]; Ogston and Vassiliadis use minimal agents working without a facilitator to match services and providers [9]. The many possible application areas divide broadly into two areas - matchmaking (e.g. matching services to clients, people connector) and search. The general topic lies within organizational concepts in multi-agent systems and improving learning with communication [21]. The *finding* and *remembering* of appropriate, like-minded agents can be centralized, left as a diffusion process or engineered in a computational ecology sense [5,4,9].

In the agent domain, as in others, visualization is often seen as a separate application that one adds on to an application for a variety of purposes. It may be of value from this point of view, but it can do more. Visual datamining seeks, amongst other things, to give the human visual system a more central role in the knowledge discovery process, to increase the integration of visualization and datamining techniques. This can be approached in a variety of ways, taking advantage of the human visual system's pattern recognition abilities, for instance, or presenting overviews of large amounts of data in novel ways. Related to this, but with a different focus, is another way, which the work presented here seeks to illustrate: the merging of the visualization and analytic processes. Here the question of whether visualization is the end result is not so important. Two examples of this are presented here:

- Overview understanding and similarity metric choice merge: Metrics are used for layouts involving dimension reduction, but different metrics produce different layouts. A metric (or other transformation process) may be needed for layout, especially in creating an overview of large complex datsets, but how is the user made aware of the different possibilities and their implications? At the same time many applications use a similarity measure - how can designers choose appropriately?
- Viewer and computational object (an agent) merge: Our layout creates a topic space. If such topic spaces are valid (they are usually approximations), can we

use this notion of spaces (or surfaces) for software agents to use when meeting or seeking other agents? In this case the visualization concept is being used to assist in agent-orientated computation. Can this help in the pursuit of the use of visualization techniques for reasoning (diagrammatic reasoning [6])?

Our earlier work looked at proximity data and multivariate data in the agent domain and indicated possible metric choices for visualization[11,13]. From the agent point of view, the question is how can we apply this and how can we assist in the problem of metric choice for profiling and classification in the agent domain. How can we meaningfully identify like-minded agents and then put this to use?

The agent paradigm is considered by some to be a valuable way of looking at problems and therefore of general application. Our work in the agent and visualization fields seeks to use visualization to serve the agent community, but, from the agent-orientated computation point of view, suggests other uses of agent ideas within visualization.

The paper first briefly surveys visualization possibilities for different types of data matrices and presents definitions of profiles, considering also the desirability of comparing profiles without revealing them. It then looks at metric choice and suggests strategies to improve choice using visualization techniques and the designing of a classification system for evaluating the metrics. The idea of using the visualization position coordinates as a profile is introduced and an example given. The nature of the examples in this paper is illustrative and the intention is to show the merging process, where visualization is not necessarily the end product, as well as to present the two specific applications.

2 Defining Profiles

In general an agent's profile is considered to be a vector of interests and behaviours (a feature list) or a similarity measure or sets and/or combinations of these [11,13]. The purpose of our work in visualization was to find layouts (in 2D or 3D) which would satisfy (usually approximately) these data either by using mathematical transformations (effective reductions via e.g. Principal Component Analysis (PCA) or distance metrics followed by Principal Coordinates Analysis (PCoA), spring embedding [3] or Self-organizing Map (SOM)[16]) or novel representations (e.g. colour maps, hierarchical axes, 'Daisy', parallel coordinates[1,2,15]).

PCA, SOM and PCoA are described briefly here as they are used in the discussion that follows:

- Principal Components Analysis: PCA is a means by which a multivariate data table is transformed into a table of factor values, the factors being ordered by importance. The two or three most important factors, the principal components, can then be displayed in 2D or 3D space.
- Self Organizing Map: The SOM algorithm [16] is an unsupervised neural net that can be used as a method of dimension reduction for visualization. It automatically organizes entities onto a two-dimensional grid so that related entities appear close to each other.

– Principal Coordinates Analysis: PCoA is used for proximity data, finding first a multivariate matrix which satisfies the distances, then transforming this into its principal components so that the two or three most important factors can be displayed in a similar fashion to PCA.

These visual representations may provide meaningful clusters or reveal patterns from which knowledge can be gained. A key problem in this area is that different methods produce different clusters (and cluster shapes). The determination of an appropriate metric [1] is a difficult problem for which general solutions are not evident. We propose the use of constructed data in a process called *signature exploration* [8] to assist in this area. This process uses specially constructed data sets to increase the user's understanding of the behaviour of visualization algorithms applied to high dimensional data.

Two developments suggest themselves from aspects of our previous visualization work: a tool for metric choice; the use of layout coordinates as a profile.

– Tool for metric choice: Agents need to compare profiles, i.e. when they meet they need to be able to compare themselves (or their tasks) and get a measure of similarity which they can interpret. Assuming (for the moment) that they are carrying their profile with them, they will need to apply an algorithm to calculate a similarity measure by both submitting their profiles to the algorithm, either both independently of the other, or via an intermediary. In designing a specific application a decision (by the designer) needs to made about what similarity measure is appropriate. The tool for metric choice developed in application of the principle of signature exploration provides an interactive interface which can help the designer to choose the metric.
– Use the layout coordinates as a profile: For layout on the screen, the data transformation or set of similarity measures results in x/y (or x/y/z) coordinates for each entity. For complex data this usually involves a significant error (i.e. it is normally not possible to find a layout which will satisfy the similarity measurements - on the one hand - and matrix transformations and truncations to 2 or 3 attributes rely on a sharp fall off of the relevant eigenvalues, which is unusual for complex data sets - on the other hand). Nevertheless such algorithms are commonly used and thus the approximations involved are often adequate. The relevant point here is that the end result is that there is an x/y (or x/y/z) coordinate pair associated with each entity and *within the current space of possibilities* this locates their *interest position*. This suggests the possibility of them carrying a much more lightweight *position profile* with them, that also means they can compare positions without revealing profiles.
  The use xy or xyz coordinates as the profile avoids revealing the profile, but the implication is that either there must be a central entity which will do the calculation (and thus that one needs to reveal one's profile to) and then give the agent its coordinates and the bounds of the space (so that it can judge relative similarity). Also this does not deal easily with dynamic situations (i.e. reflecting changing profiles), as it would require a periodic return to base to profile updating. A possible alternative is to calculate one's own coordinates with respect to a number of reference

[1] metric is here used in a general sense to mean a means of measurement which may not result in a numerical measure directly, i.e. possibly indirectly by means of layout position derived directly from SOM or PCA

points, i.e. calculate one's proximity to the reference points and then find a position in space to satisfy this reduced set of distances.
For instance for a feature list of length 5, consisting of a set of five possible agent interest areas and interest values in the range 0 to 1 (say), the following is an indication of the bounds of the space.

	A	B	C	D	E
agent1	1	0	0	0	0
agent2	0	1	0	0	0
agent3	0	0	1	0	0
agent4	0	0	0	1	0
agent5	0	0	0	0	1

It may be unwise to base the position on a computation that satisfies the similarity measures to all of these vectors (since this increases the inaccuracy of the layout), but the agent could carry the set of coordinates for certain bounds (or other reference vectors) and profile position, having the calculations made back at base. These ideas are illustrated below.

3 Choosing a Metric - Possibilities

For specific applications different metrics are used, this means that often an applications area uses one metric only. Measures may be chosen because of time complexity issues, rather than that they provide the most accurate or appropriate measure. There is also a link between the creation of the feature list and the metric choice (i.e. the formulation of the feature list affects which metric provides the most appropriate clustering) which is a further complication. In general terms the choice of metric and creation of the feature list should correspond to the required classification, but in many situations the starting point is an unknown set of data and clustering indications are sought. There is no training set and no classification. It is likely that different classifications exist. In fact there are hidden classifications, that is to say, the user has a set of things they are interested in and they would like to have the entities (other users, documents..) classified according to these groupings. One of the purposes of the signature exploration process that is being developed is to explore the mapping to clusters (via various metrics) of features of interest to the user. Originating as part of work to increase comprehension and choice of algorithms for visualization of complex data, it does not focus on feature list construction but on metric choice for a given feature set. In the process of constructing data sets for evaluation of the different options the user creates an ad hoc classification system for assessment purposes (demonstrated below).

3.1 How to choose - metrics, feature selection and weighting

The first issue is to specify the variables to be used in describing the profile and the ways in which pairwise similarities can be derived from the matrix formed by the set of profiles.

Many different measures of pairwise similarity have been proposed [7,19]. Some are closely related to one another. Measures are usually presented that are particularly relevant for comparing objects that are described by a single type of variable. This discussion restricts itself to quantitative data type for brevity.

Quantitative variable Let x_{ik} denote the value that the kth quantitative variable takes for the ith object ($i = 1,\ldots,n;\ \ k = 1,\ldots,p$). The Minkowski metric defines a family of dissimilarity measures, indexed by the parameter λ.
Minkowski metric

$$d_{ij} = (\sum_{k=1}^{p} w_k^{\lambda} |x_{ik} - x_{jk}|^{\lambda})^{1/\lambda} \ \ (\lambda \geq 1) \tag{1}$$

where $w_i (k = 1,\ldots,p)$ are non-negative weights associated with the variables, allowing standardization and weighting of the original variables. Values of λ of 1 and 2 give the two commonly used metrics of this family.
City block

$$d_{ij} = \sum_{k=1}^{p} w_k |x_{ik} - x_{jk}| \tag{2}$$

Euclidean distance

$$d_{ij} = (\sum_{k=1}^{p} w_k^2 (x_{ik} - x_{jk})^2)^{1/2} \tag{3}$$

These measures can be standardized, for instance so that d_{ij} is bounded by 1. If $w_k = (p\mathcal{R}_k)^{-1}$, where $\mathcal{R}_k$ denotes the range of values taken by the kth variable. One could also consider $w_k = p(\max_{k=1}^{p}(\mathcal{R}_k))$, which preserves the quantitative comparison between objects. Also consider not the range, but (assuming it is relevant to consider the possible minimum value then take $w_k = p(\max_{i=1}^{n}(x_{ik}))$ or $w_k = p(\max_{k=1}^{p}(\max_{i=1}^{n}(x_{ik}))$. For example:-

	Sport	Art	Music
agent1	5	1	3
agent2	4	1	5

Without weighting this gives 1.29, with the range 0.816, with max value 0.086.

Sometimes it is the relative magnitudes of the different variables that is of interest - the behaviour across the variables rather than the absolute values. Put another way, the variables describing the object define a vector with p components and interest is in the comparison of the directions of the vectors. In the following metric the cosine of the angle between the vectors is used. Since values are between -1 and 1, the measure can be transformed to take values between 0 and 1 by defining $s'_{ij} = (1 + s_{ij})/2$.
Angular separation

$$s_{ij} = \frac{\sum_{k=1}^{p} x_{ik} x_{jk}}{(\sum_{k=1}^{p} x_{ik}^2 \sum_{l=1}^{p} x_{jl}^2)^{1/2}} \tag{4}$$

For the previous example this metric gives a value for s' of 0.0465.

Mixed variables For profiling of objects the variables will sometimes be of different types: for example a person can be described in terms of their gender (binary variable), their age (quantitative variable, their amount of interest in a subject (ordinal variable if sectioned quantitative variable is used) and their personality classification (nominal variable). A general measure is:-
General similarity coefficient

$$s_{ij} = \sum_{k=1}^{p} s_{ijk}; \;\; d_{ij} = \sum_{k=1}^{p} d_{ijk} \tag{5}$$

where s_{ijk} (and correspondingly d_{ijk}) denotes the contribution to the measure of similarity provided by the kth variable. The values of s_{ijk} and d_{ijk} can take definitions as appropriate to the variable type.

Selection (feature extraction) and standardization (normalization) Sometimes it is clear what variables should be used to describe objects. In our case, with profiling queries, documents, specifications and personal profiles, it is likely that variables have to be selected from many possibilities. Thus the process is not straightforward. The pattern recognition literature describes the appropriate specification of variables as feature extraction. It is tempting to include a large number of variables to avoid excluding anything relevant, but the addition of irrelevant variables can mask underlying structure. Whilst the choice of relevant variables is important, there is also the possibility (particularly here - multidimensional nature of profiles themselves) that there is more than one relevant classification based on different, but possible overlapping, sets of variables.

Having determined appropriate variables, there is then the question of standardizing and/or differentially weighting them, followed by the construction of measures of similarity.

One aspect to the standardization is that two variables can have very different variability across the dataset. It may or may not be desirable to retain this variability. Standardization may also be with respect to the data set under consideration or with respect to a population from which the samples are drawn. In the case of quantitative variables, standardization can be made by dividing by their standard deviation or by the range of values they take in the data set. The idea of standardization lies within the larger problem of the differential weighting of variables.

4 Choosing a Metric - Visual Exploration

To assist the process of metric choice the use of specially constructed data sets in an exploration of the algorithm behaviours is proposed in signature exploration. Thus, by examining known data we gain a concrete idea of the behaviour of the various possible metrics. We have suggested a number of possible constructed data types[8]: generic(provided by the application to illustrate the behaviour of the particular algorithm); constructed(determined by the user to illustrate the behaviours in the data that are of interest to them, for evaluation purposes this represents an ad hoc classification); query (by visualization or sql-type, based on an unknown dataset, to examine clustering

of metric in practice); landmark (to provide marker entities in the visualization); feedback (the means to enable the user to enter their assessed similarities and find or modify the appropriate metric). This paper limits itself to the first two, generic and constructed, to illustrate.

4.1 Using generic data sets

Generic data sets are those considered to illustrate the behaviour of the visualization algorithms. Simple data sets do not always give an intuitive placement after such transformations. In this examination a small matrix of 7 agents were given a randomly assigned level of interest (of 1 to 10) in 7 topics.

Agent1	9	3	4	6	5	5	5
Agent2	1	10	10	1	7	2	0
Agent3	4	1	6	8	0	5	7
Agent4	2	7	8	4	0	2	0
Agent5	3	6	4	7	1	10	6
Agent6	1	7	6	5	0	2	0
Agent7	8	1	7	1	2	5	9

Subsequently three other agents were added to illustrate (a) interests identical to agent1 but scaled, (b) agent1 with the same level of interest in each topic and three other agents as in (a), (c) two of the agents showing reverse behaviours of another two. These data sets were visualized with various distance measures (using the tool SpaceExplorer [11,13,12])and comments noted. The results illustrate the similarity in behaviour of the metrics, whilst indicating the differences obtained with the two basic types - Euclidean and Angular Separation. The measures used were Minkowski ($\lambda = 3$), City, Euclidean and Angular Separation (equations 1-4). These were followed by Principle Coordinates Analysis to find points in 2D-space that satisfied the distances. Note that the accuracy of such layouts for visualization is an issue, since it is often very low. In the case of PCoA, the eigenvalues can be examined - the sum of the values of the first three (for 3D layout) being above 70% of the sum of all the eigenvalues accounts for 70% of the variance in the data and is thus an encouraging indicator.

As an illustration of this process, figure 1 shows the three shots of City, Euclidean and Angular separation with agents a,b and c having scaled interest distribution of agent1.

4.2 User-constructed data sets

Here the user constructs data sets specific to their application, explicitly or implicitly creating a classification system with which to measure the performance of the metrics in clustering their interest feature(s). This may provide a distance matrix for comparison, or such a matrix may be obtained by an informal assessment. This could be followed by feedback analysis to obtain weightings of the feature list, but here the focus is on metric choice rather than modification.

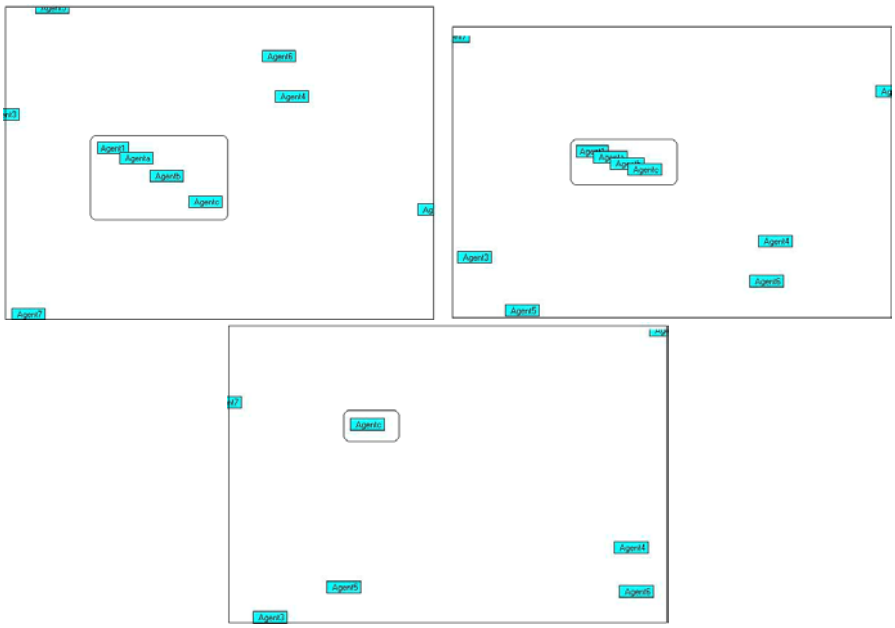


Fig. 1. City (top left), Euclidean (top right) and Angular Separation (bottom) measures followed by layout using Principal Coordinates Analysis. Agents a, b and c have scaled interest distribution of agent1. In the angular separation, agents a,b,c and 1 are located in the same position.

Step 1 - decide features of interest The first thing to consider is what features in the data one is interested in. We suppose that the aspects are: overlap of interest; intensity of interest; joint disinterest; similar pattern of interest (irrespective of subject). These elements provide a classification system with which we can construct a system to give numerical values to differences between a pair of agents' interests. Then these differences can be used to give a comparison measure for the behaviours of the various metrics. Statistical measures indicate the closeness of the match. The metrics may not correspond to the classification, even approximately. It could be that it is useful to use the classification system as the similarity measure itself and dispense with the metrics. However, in general, we are looking for a similarity metric that is not just a simple query, but something more subtle, something that reflects the multidimensional nature of the profiling data available. This corresponds to the scope that lies between the two questions:- Are you interested in sport? and Are you like me?

Also, if you are interested in sport, it may be valuable to know if you are a specialist or a generalist and in general terms what level your interest is on. Thus other similarity measures act as discriminators in this situation. Final choice of overall similarity measure may consist of additions of different similarity metrics (which may include results of specific queries) and can be arrived at in the manner of equation 5.

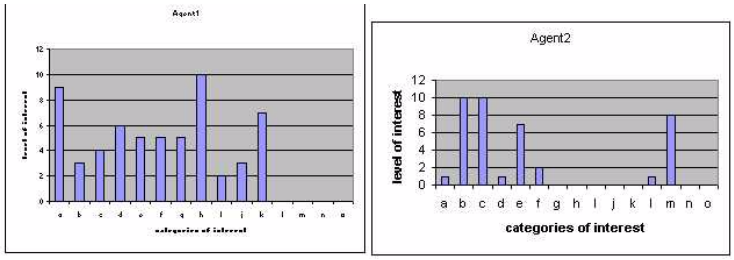


Fig. 2. Interests level against category for agents 1 and 2

The use of visualizations of data for pairs of agents can assist in the specification of features of interest. Simple diagrams such as bar and pie charts are helpful in designing a measure with which to make an informal assessment of the similarity between two agents. Figures 2, 3 are illustrative of this process and identify the features interest level and interest intensity that are used in step 2.

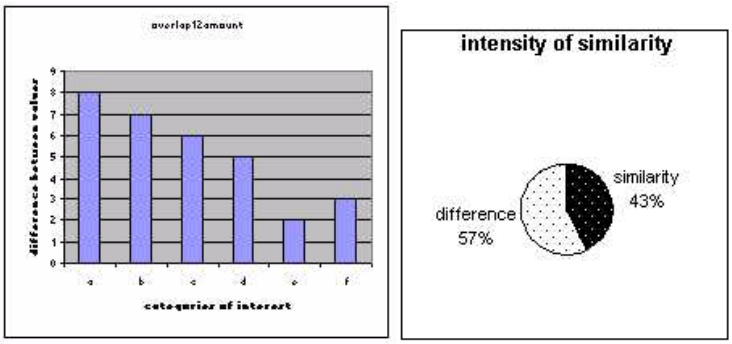


Fig. 3. Interests level difference against category for agents 1 and 2 (left) and intensity of overlap (right)

Step 2 - create a measure for the features to generate test data sets Suppose that types of agent similarity are chosen to examine e.g.: overlaps of three or more interests of high intensity; large overlaps irrespective of intensity with high common disinterest. Data is created for a representative member and edge member of desired clusters so that representative pairs of data can be created (or groups if required) to examine the metric behaviours both visually and by comparing distance matrices. To illustrate, a data set was created to produce examples covering the range of possibilities of overlap extent and intensity with respect to a reference agent's interests (as suggested by the visual

explorations of step 1). Then the metrics were examined to see how they clustered the group of similarities with number of overlap subjects $>= 3$ and intensity of overlap $>= 2/3$. One would expect the metrics to perform badly against this criterion, which is an example where a simple query would perform better (for instance, in the Yenta system, the matching between profiles is done simply on the basis of matching a single *granule*, which corresponds to a single interest, the metric is used in deriving the interest categories - if you simply want to exchange information on a subject that's ok, however such aspects as level of expertise are relevant, and finding like-minds needs greater subtlety). For metric discrimination, the distance comparison should be made by also evaluating the test criteria *for a number of other features* (such as joint disinterest and large overlaps irrespective of intensity) and combining the similarities.

Step 3 - evaluate visually and numerically The visual evaluation consists of visualizing the constructed data set and observing how well clustered the group of interest is. However, since the layout of such visualizations is an approximation (in order to satisfy the distances), and the observations not themselves measurements, evaluation by visualization is inexact. On the other hand, numerical evaluation, based on measuring differences between the estimated differences and the differences arrived at by the metric under consideration, is precise, but relies on the ability of the designer to define or estimate similarities between the data entities. For the example above this was done by awarding points according to number and intensity of topics of joint interest.

Figure 4 shows PCoA layout with City, Euclidean and Angular Separation differences, the reference agent is circled and the agents that are in my group of interest (according to the criteria in step 2) are indicated. That there is little difference between City and Euclidean indicates that it would be adequate to use city where time complexity was an issue. The three outlines traced by the points in the City and Euclidean plots correspond closely to the classification system and the group of interest is well clustered in visual terms. The Angular Separation plot does not cluster so well, misplacing three agents. The layout of the angular separation distances is actually a screenshot of a 3D representation as the layout was particularly inaccurate and needed the extra dimension to improve it (the first two eigenvalues accounted for only 38% of the variance in the data and the first three for only 48%). The inaccuracy of this layout highlights the difficulty of using visualization to assess similarity.

5 The Use of Position Instead of Vector for Profile

The pictures of information spaces as maps or terrains derived from multivariate data using self-organizing maps [16] provide us with a compelling image of the profile or topic space we are exploring. The metrics discussed above generate similar conceptual spaces when visualized. Yet this is a misleading image, since the data are high dimensional and it is impossible to represent their similarities accurately in 2 or 3D space (direct mapping methods for multivariate data, such as colour maps and parallel coordinate plots, are not included in this comment). Nevertheless, as an approximation and as a representation, an overview perhaps, of a large body of entities, it is being found useful (see e.g. [20]). Suppose we assume the validity of the layout and propose that

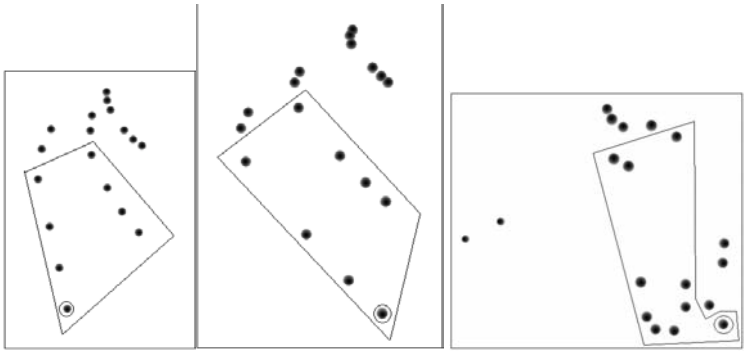


Fig. 4. Constructed data set with (left to right) PCoA and City, Euclidean and Angular differences.

the agent carries with them their xy (or xyz) coordinates and uses them as their profile. When meeting a fellow agent they can ask for the agent's xy coordinates and compute the Euclidean distance to calculate their similarity. This would be more efficient than carrying a potentially long profile vector and enable them to use their profile without revealing details or requiring encryption. Two different ways of using this idea suggest themselves - calculating back at base and on the fly.

5.1 By base calculation

The agents both have the calculations done at a base point and periodically return for updates. Here the error will be that of the layout itself and the agent would be able to have details of the mean error and variance supplied with its coordinates, so that it can take this into account. Figure 5 shows the layout after City distance and PCoA of the seven agents of randomly generated data from above. Thus, if Agent1 meets Agent2 they can compare coordinates, ((-12.30, -5.20),(23.27,-8.44)), to calculate the Euclidean distance to give them the distance they are apart in this map.

5.2 By calculation on the fly

Here the agent calculates its position with respect to a number of reference vectors (either dynamically or at an earlier point in time) and then compares with another agent's position calculated similarly. Using the seven agent random data again, the reference vectors are chosen to be agents 5,6 and 7 illustrated in the generic data section. Three reference agents are the minimum since only two will create two possible arrangements when agents 1 and 2 overlay their positions. Agents 1 and 2 separately calculate their City distances to the three reference vectors and subsequently lay out these distances with PCoA as shown in figure 6.

They now have xy coordinates, but in order to compare them they must be scaled (the Euclidean distance between 5 and 6 is used here), centered (here Agent 5 is placed

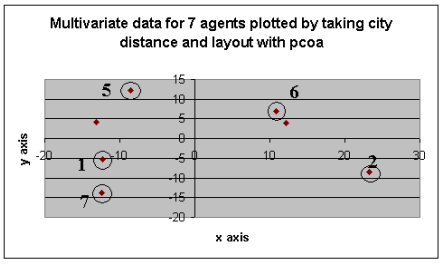


Fig. 5. Illustration of base plot, the three reference agents (5,6 and 7) and the two of interest in this measurement (1 and 2) are circled.

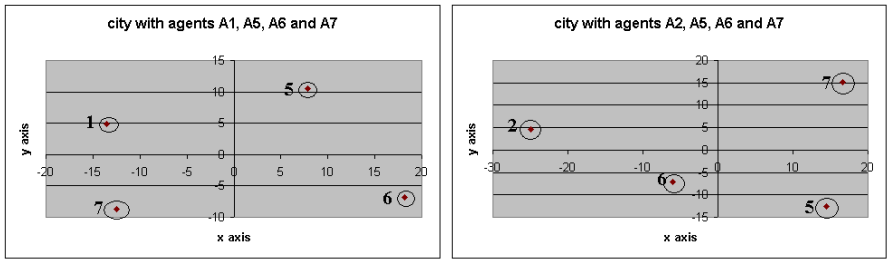


Fig. 6. Illustration of plots calculated individually by agents 1 (left) and 2 (right) with respect to the three reference agents (5,6 and 7) as circled and numbered.

at 0,0) and finally rotated to bring the agents 5,6 and 7 into position. Now the coordinates of the agent's position are in a form that they can use for comparisons. The results of the base calculation and on-the-fly calculation of the difference between agents 1 and 2 are given in the table below. (Since these are normalized with respect to the distance between agents 5 and 6, a value of 1 would indicate that they were the same distance away from each other as agents 5 and 6 are)

original city dist	base dist	on-the-fly dist
1.64	1.77	1.57
exact	8%err	-4%err

6 Conclusions and Future Work

Visual datamining seeks to increase the integration of visualization with specific datamining techniques. This paper presents two applications with this in mind.

Appropriate clusterings of data are sought, whilst at the same time layouts are required to present overviews. The user needs understanding of the layout algorithm to appreciate the implications of the overview, the implication of arriving at different clusterings with different algorithms needs to be understood by those seeking valid cluster-

ings and classifications. These two purposes concern the same process, but are subtly different in their focus. The first application described in this paper, illustrating the use of signature exploration in making the behaviour of the similarity metrics more concrete and assisting in similarity metric choice, is an example of the merging of understanding of overview and determining appropriate metric. Visualization of pairs of data helped in the creation of an ad hoc, user-specific, classification with which to assess the overview and thus also the metrics. An obvious next step is to use feedback to select and modify metrics and features and this is another part of the signature exploration process. Continuing work lies in further developing the interface for exploration, the data construction engine and in conducting usability tests.

Visualizations sometimes suggest the idea of a topic or similarity space - looking at a 2D or 3D scatterplot the closeness of entities is intuitively understood as similarity. Where dimension reduction is involved, considerable approximation or abstraction is required. If this is a valid procedure (in the sense of the considerable error sometimes incurred), and such diagrams are widely used without warnings given, then the idea of using location as a form of privacy protection (the transformation is a one-way function) must also hold on some level. The simple example for using position as a profile demonstrated in this paper - a potentially most useful mechanism - is encouraging, now evaluation for many different data sets is required to test its robustness, in terms of whether the original profile is fully protected and the tolerance of approximation in *locations* of the entities.

Evaluation of the position-as-profile concept points to one of our most pressing problems in visualization - how valid are our visualizations when dealing with complex data and involving approximation or abstraction? How can the level of approximation be indicated to the viewer? Correspondingly, how can a measure of confidence in the agent's location in the interest space be given to the agent? The investigation of the position-as-profile idea is the same investigation as that of the validity of layout. Thus, we begin to think in terms of transfering our picture as a viewer to the agent, so that the two can become one - a kind of viewer/agent entity. The agent thus may be a software agent or a human agent. The question now becomes, how can the boundaries or parameters of the validity be described to the viewer/agent? How can they be encoded visually and in software terms? We interchange the viewer with agent and must express what the user *sees* (or finds useful) in a form that the software agent can work with. Via the agent paradigm we may thus be helped toward creating programs that can use graphical elements to mimic our visual thinking.

6.1 Acknowledgements

This work is supported by the EPSRC and British Telecom (CASE studentship - award number 99803052).

References

1. S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Readings in Information Visualization: Using Vision To Think*. Morgan Kaufmann, 1999.
2. C. Chen. *Information Visualisation and Virtual Environments*. Springer, 1999.
3. G. di Battista, P. Eades, R. Tamassia, and I. G.Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1999.
4. L. Foner. Yenta: a multi-agent, referral-based matchmaking system. In *The First International Conference on Autonomous Agents, Marina del Rey, California*. ACM press, 1997.
5. L. N. Foner. Clustering and information sharing in an ecology of cooperating agents. In *AAAI Spring Symposium '95 on Information Gathering in Distributed, Heterogeneous Environments, Palo Alto*, 1995.
6. J. Glasgow, N. Hari Narayanan, and B. Chandrasekaran. *Diagrammatic Reasoning*. AAAI Press / The MIT Press, 1995.
7. A. D. Gordon. *Classification*. Chapman and Hall / CRC, 2nd edition edition, 1999.
8. P. Noy and M. Schroeder. Introducing signature exploration: a means to aid the comprehension and choice of visualization algorithms. In *ECML-PKDD01 Visual Data Mining Workshop*, Freiburg, Germany, Sept 2001.
9. E. Ogston and S. Vassiliadis. Matchmaking among minimal agents without a facilitator. In *Proceedings of Autonomous Agents2001*, Montreal, Canada, 2001. ACM press.
10. P.Faratin, C.Sierra, and N.R.Jennings. Using similarity criteria to make negotiation tradeoffs. In *Proc. of 4th Int. Conf. on Multi-Agent Systems ICMAS-2000*, pages 119–126, Boston, USA, 2000. IEEE Computer Society.
11. M. Schroeder. Using singular value decomposition to visualise relations within multi-agent systems. In *Proceedings of the third Conference on Autonomous Agents*, Seattle, USA, 1999. ACM Press.
12. M. Schroeder, D. Gilbert, J. van Helden, and P. Noy. Approaches to visualisation in bioinformatics: from dendrograms to Space Explorer. *Information Sciences*, 139:19–57, 2001.
13. M. Schroeder and P. Noy. Multi-agent visualization based on multivariate data. In *Proceedings of Autonomous Agents2001*, Montreal, Canada, 2001. ACM press.
14. G. Somlo and A. Howe. Incremental clustering for profile maintenance in information gathering web agents. In *Proceedings of Autonomous Agents2001*, Montreal, Canada, 2001. ACM press.
15. R. Spence. *Information Visualization*. Addison-Wesley, 2001.
16. T.Kohonen. *Self-organising maps*. Springer-Verlag, 2nd edition edition, 1997.
17. D. Veit. *Matchmaking algorithms for autonomous agent systems*. Master's thesis, Institute of Computer Science, University of Giessen, Germany, 1999.
18. D. Veit, J. Muller, M. Schneider, and B. Fiehn. Matchmaking for autonomous agents in electronic marketplaces. In *Proceedings of Autonomous Agents2001*, Montreal, Canada, 2001. ACM press.
19. A. Webb. *Statistical Pattern Recognition*. Arnold, 1999.
20. Websom. http://websom.hut.fi/websom/.
21. G. Weiss, editor. *Multiagent Systems*. MIT Press, 1999.

Visual Data Mining of Clinical Databases: an Application to the Hemodialytic Treatment based on 3D Interactive Bar Charts

Luca Chittaro[1], Carlo Combi[2], Giampaolo Trapasso[1]

[1] HCI Lab, Dept of Math and Computer Science,
University of Udine,
via delle Scienze 206, 33100 Udine, Italy
chittaro@dimi.uniud.it
[2] Department of Computer Science,
University of Verona,
strada le Grazie 15, 37134 Verona, Italy
combi@sci.univr.it

Abstract. The capability of interactively mining clinical databases is an increasingly urgent need. This paper considers a relevant medical application (i.e., hemodialysis) and proposes a system for the visualization and visual data mining (VDM) of the collections of time-series acquired during hemodialytic treatments. Our proposal adopts bar charts as the basic visualization technique (because it is very familiar for clinicians) and augments them with several interactive features, exploiting a 3D space to significantly increase both the number of time-series that can be simultaneously analyzed in a convenient way and the number of values associated with each series.

1 Introduction

The capability of interactively mining patient clinical information is an increasingly urgent need in the clinical domain, due to the continuous growth in the number of parameters that can be automatically acquired and in the size of the databases where they accumulate [5]. This is particularly critical for the success of medical research projects which generate massive databases of patient data.

Some techniques for visual data mining (VDM) of multidimensional clinical databases are illustrated in [7]. They are mainly based on 3D versions of *parallel coordinate plots*. Graphical connections between points in adjacent planes are drawn in such a way that each patient's case is visually represented by a line connecting individual points referring to it. This allows for VDM of interesting patterns (e.g., a group of patients with the same profile results in parallel lines).

A different approach is presented by [10] and is based on *tables* displaying records of the clinical database and their attributes in highly compressed format such that they fit onto the screen. Users directly manipulate the table (e.g., performing zoom and filter operations) that dynamically rearranges itself. To compress the tables, the system relies on visualization criteria such as (i) neighboring cells with identical values are combined into a larger cell, or (ii) if there is no space to display a numeric value in its cell, the value is substituted by a small horizontal line whose position indicates relative size.

This paper explores a third possibility, especially suited to clinical databases containing time-series data. Since, historically, *bar charts* are a widely adopted approach to display time-series and are a very familiar representation for clinicians, we chose them as the basis of our visual approach. Unfortunately, while a bar chart allows for an easy comparison among the data values for a single time-series, when the considered task requires to compare a *collection* of time-series (such as a monitored signal from the same patient in different sessions of the same clinical test or treatment), traditional bar charts (as other historical approaches) become unfeasible. Therefore, we augment bar charts exploiting a 3D space and adding several interactive features. A 3D space can significantly increase both the number of time-series that can be simultaneously analyzed in a convenient way and the number of values associated with each time-series, but poses well-known problems such as occlusions, 3D navigation, difficulties in comparing heights, proper use of space, and the need for effective interaction techniques to aid the user in the analysis of large datasets (e.g., highlighting interesting patterns, checking trends,…). The limited capabilities of commercial tools that generate 3D bar charts have led well-known researchers (e.g. [9]) to classify these visualizations as "chartjunk 3D". However, solutions to the problems of 3D bar charts are emerging from research: e.g., Cichlid offers temporal animation capabilities of 3D stacked bar charts [2], while ADVIZOR allows one to interactively link the 3D bar chart representation with related 2D representations, compare heights with a "water level" plane (perpendicular to the bars) and use filtering tools [6].

Alternative approaches to time-series visualization have been recently proposed, e.g. drawing the timeline along spiral structures [12] is reported to allow for an easier detection of cyclic phenomena. However, we preferred to adopt bar charts, because they were familiar to clinicians. Moreover, we did not have a focus on a specific pattern such as cycles.

In the following, we first introduce the real-world clinical context we are working in and motivate the need for VDM in that context. Then, we illustrate the system we have built and its main features. Finally, we show some examples of how our system is being applied to the clinical context.

2 Hemodialysis and Visual Data Mining

Hemodialysis is the widely used treatment for patients with acute or chronic end-stage renal failure. During an hemodialysis session, the blood passes through an extra-

corporeal circuit where metabolites (e.g., urea) are eliminated, the acid-base equilibrium is re-established, and water in excess is removed. In general, hemodialysis patients are treated 3 times a week and each session lasts about 4 hours.

Hemodialysis treatment is very costly and extremely demanding both from an organizational viewpoint [8] and from the point of view of the patient's quality-of-life. A medium-size hemodialysis center can manage up to 60 patients per day, i.e. more than 19000 hemodialytic sessions per year. Unfortunately, the number of patients that need hemodialysis is constantly increasing [12]. In this context, it is very important to be able to evaluate the quality of (i) each single hemodialysis session, (ii) all the sessions concerning the same patient, and (iii) sets of sessions concerning a specific hemodialyzer device or a specific day, for the early detection of problems in the quality of the hemodialytic treatment.

Modern hemodialyzers are able to acquire up to 50 different parameters from the patient (e.g., heart rate, blood pressure, weight loss due to lost liquids,…) and from the process (e.g., pressures in the extra-corporeal circuit, incoming blood flow,…), with a configurable sampling time whose lower bound is 1 sec. As an average example, considering only 25 parameters with a sampling time of 30 seconds, 12000 values (4*120*25) are collected in each session, and a medium-sized center collects more than 228 millions of values per year (considering 19000 provided treatments).

While the daily accumulation of huge amounts of data prompts the need for suitable techniques to detect and understand relevant patterns, hemodialysis software is more concerned with acquiring and storing data, rather than visualizing and analyzing it. Data mining applications can thus play a crucial role in this context. More specifically, *visual* data mining applications are of particular interest for three main reasons.

First, clinicians' abilities in recognizing interesting patterns are used suboptimally or not used at all in the current context. Visual mining of hemodialytic data would allow clinicians to take decisions affecting different important aspects such as therapy (personalizing the individual treatment of specific patients), management (assessing and improving the quality of care delivered by the whole hemodialysis centre), medical research (discovering relations and testing hypothesis in nephrology research).

Second, since data mining on the considered database is (at least, at initial stages) intrinsically vague for clinicians, the adoption of VDM techniques can be more promising than fully automatic techniques, because it supports clinicians in discovering structures and finding patterns by freely exploring the datasets as they see fit.

Third, the clinical context is characterized by a need for user interfaces that require minimal technical sophistication and expertise to the users, while supporting a wide variety of information intensive tasks. A proper exploitation of visual aspects and interactive techniques can greatly increase the ease of use of the provided solutions.

In summary, a clinical VDM system has to achieve two possibly conflicting goals: (i) offering powerful data analysis capabilities, while (ii) minimizing the number of concepts and functions to be learned by clinicians. In the following, we illustrate how our system attempts to achieve these two goals.

3 The Proposed Approach

The system we have built, called IPBC (*Interactive Parallel Bar Charts*) connects to the hemodialysis clinical database, produces a visualization that replaces tens of separate screens used in traditional hemodialysis systems, and extends them with a set of interactive tools that will be described in detail in this section.

Each hemodialysis session returns a time-series for each recorded clinical parameter. In IPBC, we visually represent each time-series in a bar chart format where the X axis is associated with time and the Y axis with the value (height of a bar) of the series at that time. Then, we layout the obtained bar charts side by side, using an additional axis to identify the single time-series, and we draw them in a 3D space, using an orthogonal view. It must be noted that also the additional axis has typically a temporal dimension, e.g. it is important to order the series by date of the hemodialysis session to analyze the evolution of a patient. An example is shown in Fig. 1, that illustrates a visualization of 50 time-series of 50 values each, resulting in a total of 2500 values (the axis on the right is the time axis for single sessions, while the axis on the left identifies the different time-series, ordered by date). Hereinafter, we refer to this representation as a *parallel bar chart*.

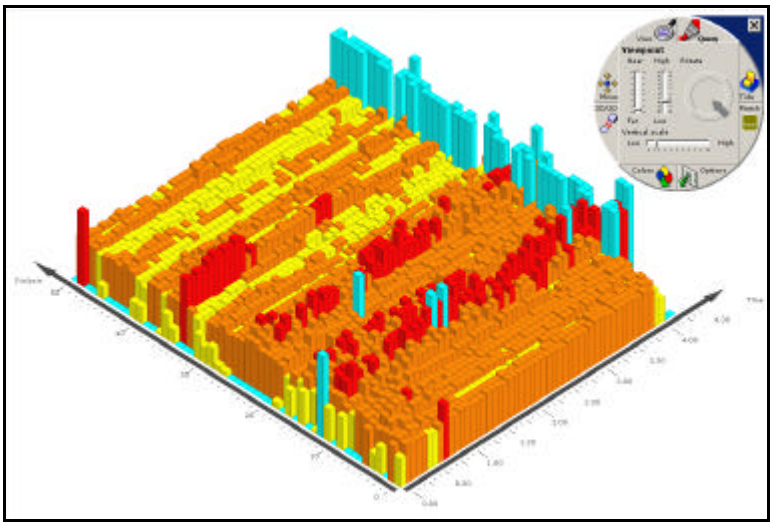


Fig. 1. A Parallel Bar Chart.

3.1. The RoundToolbar widget

In designing how the different interactive functions of IPBC should be invoked by the user, we wanted to face two different problems:

- First, one well-known limitation of many 3D visualizations is the possible waste of screen space towards the corners of the screen;

- Second, the traditional menu bar approach would require long mouse movements from the visualization to the menu bar and vice versa.

To this purpose, we designed a specific round-shaped pop-up menu (see Fig. 2), called *RoundToolbar* (RT), that appears where the user clicks with the right mouse button. The RT can be easily positioned in the unused screen corners, thus allowing a better usage of the screen space (e.g., see Fig. 1) and a reduction of the distance between the visualization and the menu. Moreover, to further improve selection time of functions with respect to a traditional menu, the organization of modes in the toolbar is inspired by Pie Menus [3]: in particular, the main modes are on the perimeter of the RT, and when a mode is selected, the center of the RT contains the corresponding tools (which are immediately reachable by the user, who can also quickly switch back from the tools to a different mode).

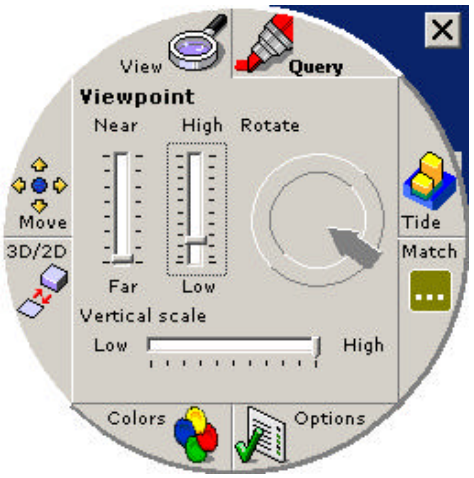


Fig. 2. Viewpoint mode.

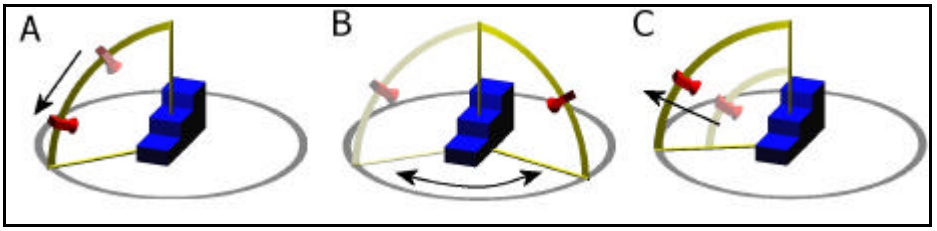


Fig. 3. Viewpoint movements: A) Low; B) Rotate; C) Far.

3.2. Changing Viewpoint

It is well-known that free navigation in a 3D space is difficult for the average user,

because (s)he has to control 6 different degrees of freedom and can follow any possible trajectory. To make 3D navigation easier, when the *Viewpoint* mode is selected in the RT (as in Fig. 2), the proposed controls for viewpoint movement (*Rotate*, *High-Low* and *Near-Far*) cause movement along limited pre-defined trajectories which can be useful to examine the visualization: in particular, Fig. 3 shows how viewpoint movement is constrained. The remaining *Vertical scale* control in the *Viewpoint* mode is used to scale the bars on the Y axis. Vertical scaling has been included in the *Viewpoint* mode, because it has been observed that when users scaled the bars, they typically changed the viewpoint as the following operation.

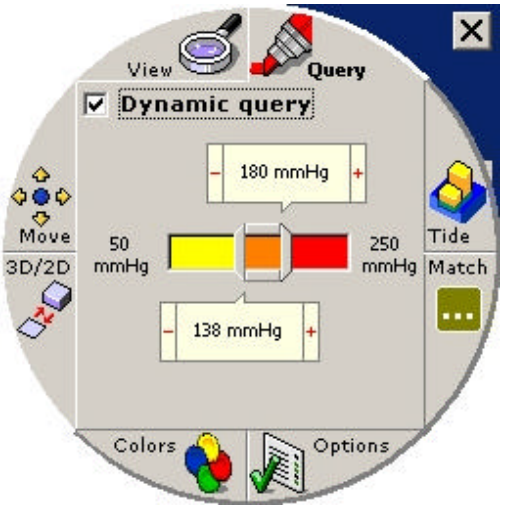


Fig. 4. Dynamic Query mode.

3.3. Dynamic Queries

IPBC uses color to classify time-series values into different ranges. In particular, at the beginning of a session, the user can define units of measure and her general *range of interest* for the values, specifying its lowest and highest value. These will be taken as the lower and upper bounds for an IPBC dynamic query control in the RT (as shown in Fig. 4) that allows the user to interactively partition the specified range into subranges of interest. Different colors are associated to the subranges and when the user moves the slider elements, colors of the affected bars in the IPBC change in real-time. Possible bars with values outside the specified general range of interest are highlighted with a proper single color. For example, Fig. 1 shows a partition that includes the three subranges corresponding to the colors shown by the slider in Fig. 4, and also some bars which are outside the user's predefined range. The color coding scheme can be personalized by the user with the *Colors* mode in the RT. The dynamic query control allows the user to:

- move the two slider elements *independently* (to change the relative size of adja-

cent subranges). For example, in Fig. 4, one has been set to 130 mmHg and the other to 180 mmHg. This can be done both by dragging the edges or (more easily) the tooltips which indicate the precise value. Plus and minus signs in the tooltips also allow for a fine tuning of the value.

- Move the two slider elements *together* by clicking and dragging the area between the two bounds. This can be particularly useful (especially when the other areas are associated to the same color), because it will result in a "spotlight" effect on the visualization: as we move the area, our attention is immediately focused on its corresponding set of bars, highlighted in the visualization.

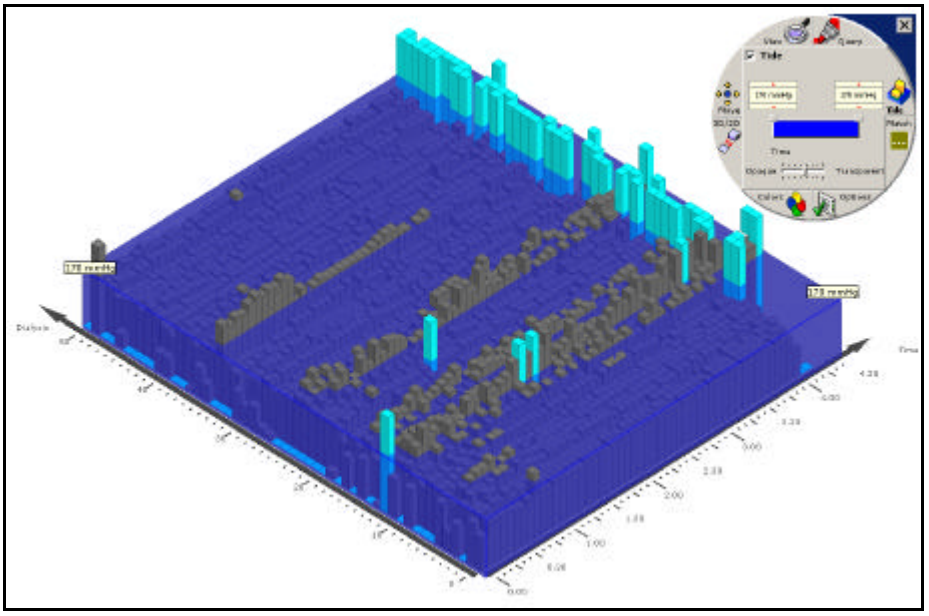


Fig. 5. Tide mode.

3.4. Comparing data with (time-varying) thresholds

A frequent need in VDM is to quickly perceive how many and which values are below or above a given threshold. This can be easily done with the previously described dynamic queries when the threshold is constant. However, the required threshold is often time-varying, e.g. one can be interested in knowing how many and which values are not consistent with an increasing or decreasing trend. For this need, IPBC offers a mode based on a tide metaphor. As it can be seen in Fig. 5, the *Tide* mode adds a semi-transparent solid to the visualization: the solid metaphorically represents a mass of water that floods the bar chart, highlighting those bars which are above the level of water. The slope of the top side of the solid can be set by moving two tooltips shown in the RT (that specify the initial and final values for the solid height), thus determin-

ing the desired linearly increasing or decreasing trend. The height of the solid can be also changed without affecting the slope by clicking and dragging the blue area in the RT. An *opaque/transparent* control allows the user to choose how much the solid should hide what is below the threshold. When the *Tide* mode is activated, all the bars in the user's range of interest are turned to a single color to allow the user to more easily perceive which bars are above or below the threshold; if multiple colors were maintained, the task would be more difficult, also because the chromatic interaction between the semitransparent surface and the parts of bars inside it adds new colors to the visualization.

The *Tide* mode can be also used to help compare sizes of bars by selecting a zero slope and changing the height of the solid (in this special case, *Tide* becomes analogous to the "water level" function of other visualization systems). Fig. 5 illustrates this latter case, while Fig. 9 shows a positive slope case.

Implementing a non-linear *Tide* would be relatively straightforward (only linear trends are anyway used by clinicians in the considered hemodialysis domain).

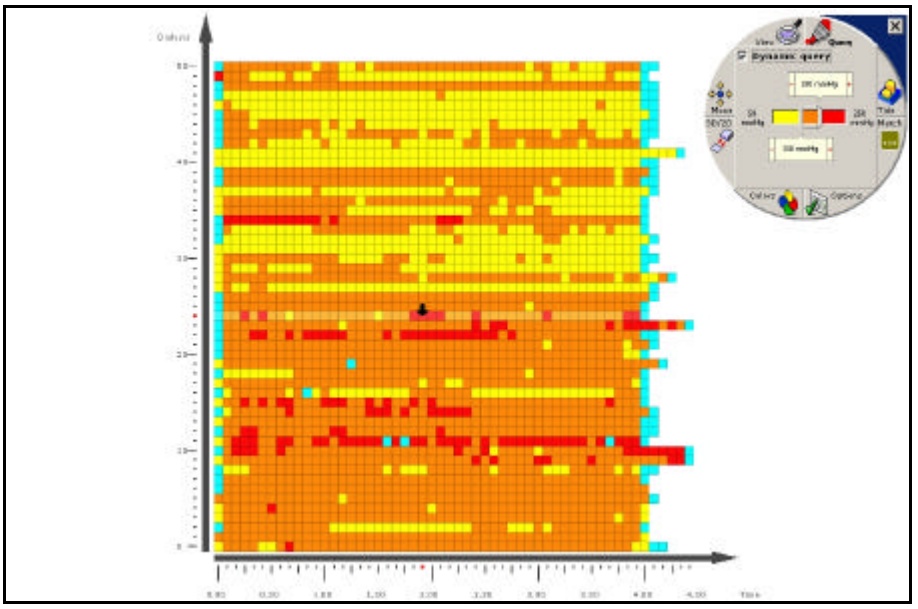


Fig. 6. Matrix Visualization.

3.5. Managing Occlusions

As any 3D visualization, IPBC can suffer from occlusion problems. To face them, the approach offers two possible solutions.

First, by clicking on the *2D/3D* label on the RT, the user can transform the parallel bar chart into a matrix format and vice versa. For example, Fig. 6 shows the same data as Fig. 1 in the matrix format. The transformation is simply obtained by automatically

moving the viewpoint over the 3D visualization (and taking it back to the previous position when the user deselects the matrix format). This can solve any occlusion problem (and the dynamic query control can still be used to affect the color of the matrix cells), but the information given by the height of the bars is lost. Transitions to matrix format and back are animated to avoid disorienting the user and allow her to keep her attention on the part of the visualization (s)he was focusing on.
Second, by directly clicking on any time-series in the 3D visualization, only the time-series which can possibly occlude the chosen one collapse into a flat representation analogous to the matrix one, as illustrated in Fig. 7.

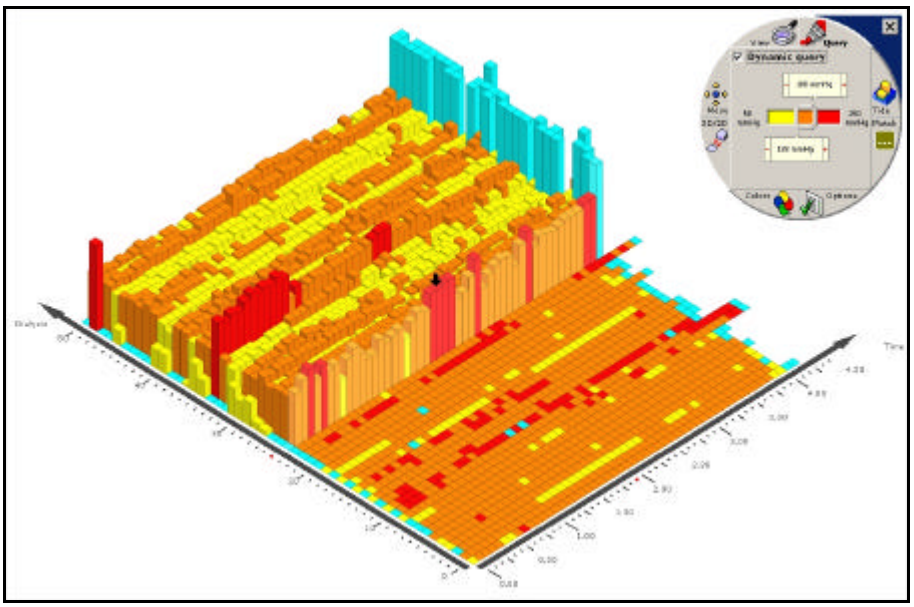


Fig. 7. Removing occlusions.

3.6. Pattern Matching

When the user notices an interesting sequence of values in one of the time-series, IPBC offers her the opportunity to automatically search for and highlight occurrences of a similar pattern in all the visualization (a detailed example will be described in Section 4.4).

The user selects her desired sequence of values in a time-series by simply dragging the mouse over it, then (s)he can specify how much precise the search should be, by indicating two tolerance values in the RT: (i) how much a single value can differ in percentage from the corresponding one in the given pattern, (ii) the maximum number (possibly zero) of values in a pattern that can violate the given percentage.

3.7. Mining Multidimensional Data

If multiple variables are associated to the considered time-series, IPBC can organize the screen into multiple windows, each one displaying a parallel bar chart for one of the variables. The visualizations in all the windows are linked together, e.g. if one selects a single time-series in one of the windows (or a specific value in a time-series), that time-series (or the corresponding value) is automatically highlighted in every other window. This (as some other features of IPBC) will be shown in more detail in the next section.

4. Mining Hemodialytic data

In the following, we will show how IPBC can be used during real clinical tasks, to help physicians evaluating the quality of the hemodialytic treatments given to single patients, on the basis of the clinical parameters acquired during the sessions. Each hemodialysis session returns a time-series for each parameter; different time-series are displayed side by side in the parallel bar chart according to date (in this case, the axis on the left chronologically orders the sessions).

The following examples are ordered according to the complexity of the related task: in particular, the first two tasks are relatively simple and are taken from the daily activity of clinicians, while the last two tasks are more complex and are performed by clinicians only in specific occasions (in the two considered examples, they are related to a detailed evaluation of the quality of care provided by nurses).

4.1. Mining patient signs data

A first task consists in analyzing patient signs, as the systolic and diastolic blood pressures and the heart rate; indeed, these parameters are important both for the health status of the patient and for the management of device settings during the hemodialytic session.

Let us consider, for example, the task of analyzing all the systolic pressures of a given patient: Fig. 8 shows a parallel bar chart (containing more than 5000 bars), representing the systolic pressure measurements (about 50 per session) during more than 100 hemodialytic sessions. In this figure, we can observe that the presence of out-of-scale values, usually related to measurement errors (e.g., the patient was moving; the measurement device was not properly operating), has been highlighted by specifying a proper range of interest (that highlights them in a suitable color) and hiding their height. In the specific situation represented in the figure, the presence of several out-of-scale values at the beginning of each session is due to the fact that nurses activate the measurement of patient's blood pressure with some delay with respect to the beginning of the session.

In the figure, the user is focusing on a specific session, avoiding occlusion problems (as described in Section 3.5). At the same time, with a dynamic query, (s)he is able

to distinguish low, normal, and high blood pressures. In this case, the clinician can observe that the systolic pressure in the chosen session, after a period of low values (yellow bars), was in the range of normal values (orange bars). While the values for the chosen session correspond to a normal state, it is easy to observe that several sessions among those in the more recent half part of the collection contain several high values (in red) for the systolic blood pressure. Thus, the clinician can conclude that in those sessions the patient had some hypertension, i.e. a clinically undesired situation.

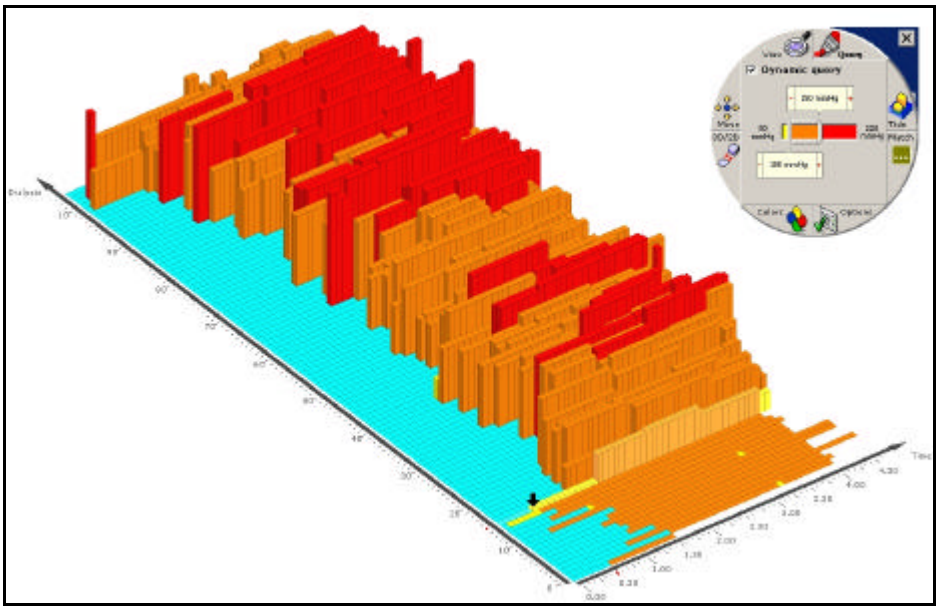


Fig. 8. Analyzing systolic blood pressures.

4.2. Mining blood volume data

Another task is related to observing the percentage of reduction of the blood volume during hemodialysis, mainly due to the removal of the water in excess. This reduction is sometimes slowed down to avoid situations in which the patient has too low blood pressures. In this case, VDM can benefit from the usage of the *Tide* mode. Fig. 9 shows an IPBC with more than 9000 bars, representing 36 hemodialytic sessions, containing about 250 values each. In this case, being the percentage of reduction of the blood volume increasing during a session, *Tide* allows the physician to distinguish those (parts of) sessions characterized by a percentage of reduction above or below the desired trend. In the figure, for example, the selected session has a first part emerging from the tide, while the last part is below. At the same time, it is possible to observe that one of the last sessions has the percentage of reduction above the tide during almost the entire session. The clinician can thus easily identify those (parts of) sessions with a satisfying reduction of the blood volume as the emerging (parts of)

sessions.

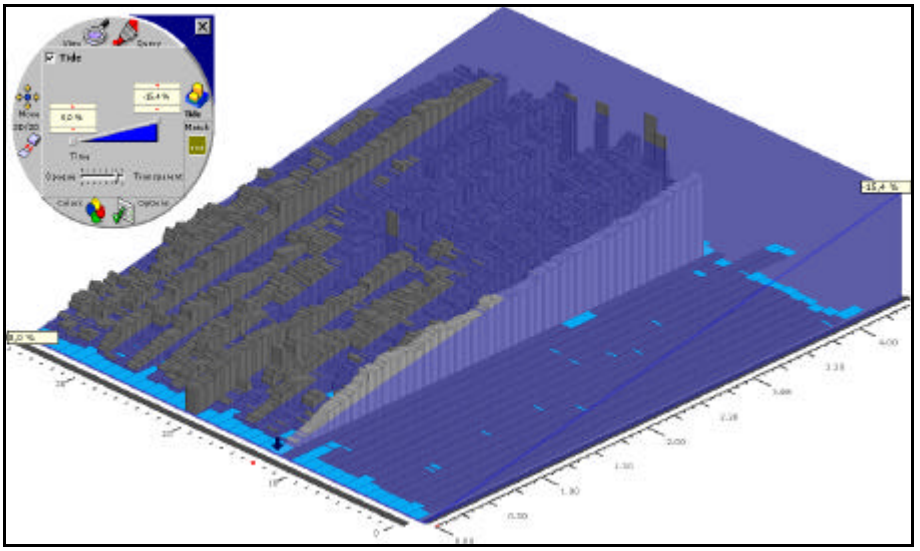


Fig. 9. Visualizing the time-varying reduction of the blood volume in the Tide mode.

4.3. Mining related clinical parameters

The next task we consider is related to the analysis of three related parameters: the systolic and diastolic blood pressures (measured on the patient) and the blood flow (QB) entering the hemodialyzer. QB is initially set by the hemodyalizer, but it can be manually set (reduced) by nurses when the patient's blood pressures are considered too low by the medical staff. It is thus interesting to visually relate QB and blood pressures, to check whether suboptimal QBs are related to low pressures. Otherwise, suboptimal values of QB would be due to human errors during the manual setting of the hemodialyzer. Fig. 10 shows the coordinated visualization of three clinical parameters for the same patient: the diastolic blood pressure (small window in the upper left part), the systolic blood pressure (small window in the lower left part), and QB (right window). The user can freely organize the visualization, switching the different charts from the smaller to the larger windows (by clicking on the arrow in the upper right part of the smaller windows). In the figure, the clinician is focusing on a session where the QB was below the prescribed value during the first two hours of hemodialysis (yellow color for QB) and (s)he has selected a specific value (the system highlights that value and the corresponding values in the other windows with black arrows). It is easy to notice that the suboptimal QB was related to low blood pressures (yellow bars in the corresponding time-series in the two small windows); then, QB was set to the correct value by nurses (see black arrow in the right window) only after blood pressures reached normal values (orange color in the corresponding charts). In this case, the

physician can conclude that the suboptimal QB has been correctly set by nurses because of the patient's hypotension.

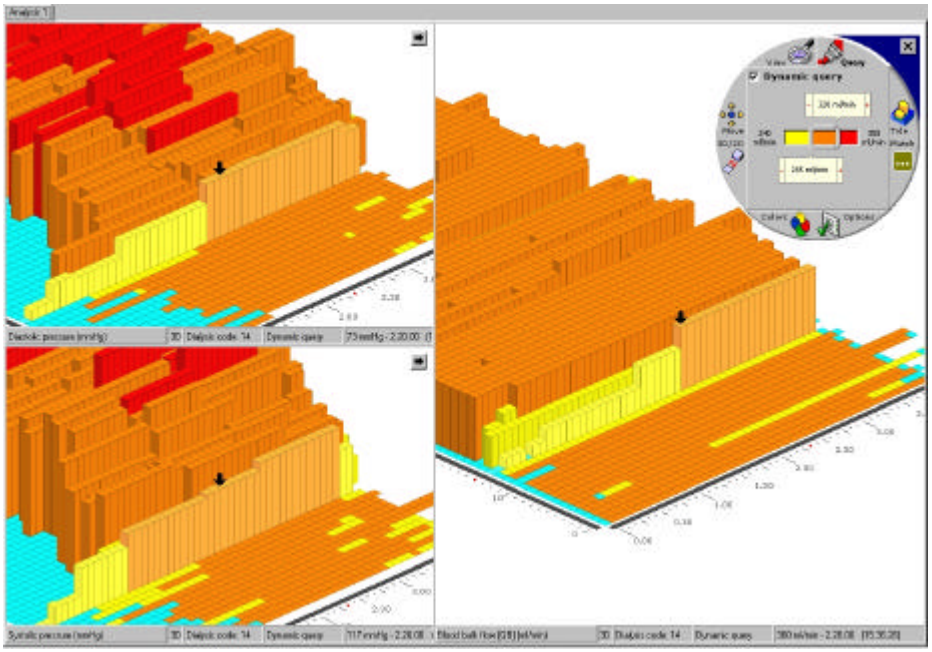


Fig. 10. Coordinated analysis of blood pressures and incoming blood flow.

4.4. Mining for similar patterns

Finally, let us consider a task concerning the analysis of QB. As previously mentioned, the value of QB can be manually set by nurses and it may happen that this value is below the optimal one, due to hypotensive episodes. Fig. 11 shows a visualization where the clinician noticed a change of QB from a lower value to the correct one in a session: this means that, after a period of suboptimal treatment, the proper setting had been entered. Therefore, the clinician asks IPBC to identify QB patterns similar to the one (s)he noticed, by indicating it with the mouse, and setting the tolerance parameters (see Section 3.6). Fig. 11 shows the selected pattern (see the area near the black arrow) and the similar patterns automatically found by IPBC (two are in the lower right part of the figure, one in the upper left part): these patterns are identified by a line of a suitable color, which highlights the contours of the first and last bar of the pattern and intersects the inner bars. To avoid possible occlusion problems in visually detecting the patterns, the physician can move the viewpoint or switch to the matrix representation, where each pattern can be easily observed.

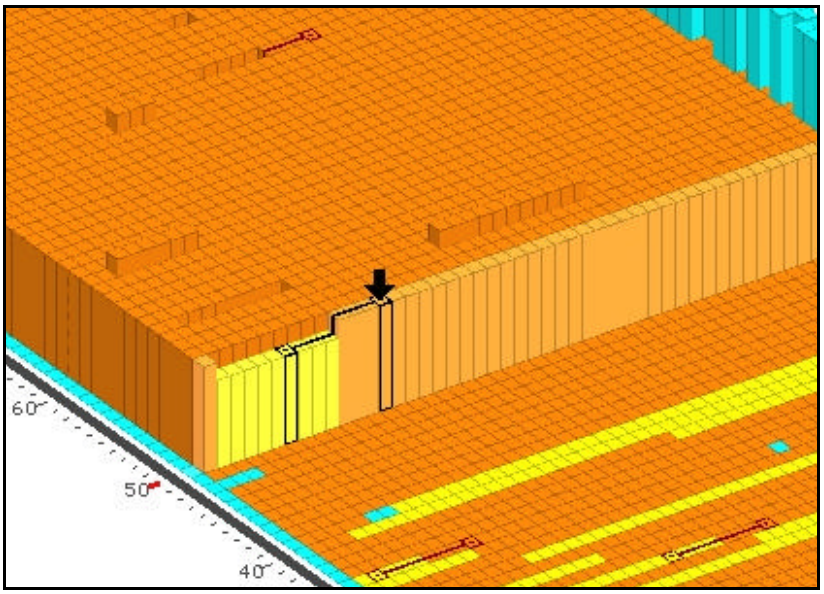


Fig. 11. Automatic Pattern Matching.

5. Conclusions and Future Work

In this paper, we described the main features of IPBC (Interactive Parallel Bar Charts), a VDM system devoted to interactively analyze collections of time-series, and showed its application to a real clinical database of hemodialytic data.

We are currently carrying out a field evaluation of IPBC with the clinical staff of the hemodialysis center at the Hospital of Mede, PV, Italy. One of the major advantages of IPBC that is emerging is that the visualization and its interactive features are very quickly learned and remembered by clinicians, the major disadvantage is that usage of screen space becomes difficult if a clinician tries to relate more than 3 collections of time-series simultaneously (Section 4.3 dealt with the analysis of 3 collections). This early feedback received from the field evaluation is helping us in identifying new research directions. Besides facing the problem of analyzing more than 3 collections in a convenient way, we aim to face another problem (that is considered very relevant by clinicians), i.e. dealing with time-series at different abstraction levels, allowing for both a fine exploration of time-series (e.g., to detect specific unusual values) and their coarse exploration (to focus on more abstract derived information). In both cases, we are working at the integration of parallel bar charts with other visualizations that can provide a synthetic view of data (e.g., the medical literature is proposing some computation methods to derive some quality indexes of the hemodialytic session from the time-series of that session). In particular, we are experimenting with Parallel Coordinate Plots, e.g. a trajectory in a plot could connect the quality indexes (typically, 5-7 values) of a session, and this high-level perspective would be linked to the much

of a session, and this high-level perspective would be linked to the much more detailed perspective of the parallel bar chart.

Acknowledgements
This work is partially supported by a MURST COFIN 2000 project ("Analysis, Information Visualization, and Visual Query in Databases for Clinical Monitoring").

References

1. Ahlberg, C., Williamson, C., Shneiderman B.: Dynamic queries for information exploration: An implementation and evaluation. Proc. of the CHI '92 Conference on Human Factors in Computing Systems, ACM Press, New York (1992) 619-626
2. Brown, J.A., McGregor, A.J., Braun H-W.: Network Performance Visualization: Insight Through Animation. Proc. of PAM2000: Passive and Active Measurement Workshop, Hamilton, New Zealand (2000) 33-41
3. Callahan, J., Hopkins, D., Weiser, M., Shneiderman, B.: An empirical comparison of pie vs. linear menus. Proc. of the CHI '88 Conference on Human Factors in Computing Systems, ACM Press, New York (1998) 95-100
4. Chittaro L. (ed.), Special issue on Information Visualization in Medicine, Artificial Intelligence in Medicine Journal, 22(2) (2001)
5. Chittaro L.: Information Visualization and its Application to Medicine, in [4] 81-88
6. Eick, S. G.: Visualizing Multi-Dimensional Data. ACM SIGGRAPH Computer Graphics, 34(1) (2000) 61-67
7. Falkman, G.: Information Visualization in Clinical Odontology: Multidimensional Analysis and Interactive Data Exploration, in [4] 133-158
8. McFarlane, P.A., Mendelssohn, D.C.: A call to arms: economic barriers to optimal hemodialysis care. Perit Dial Int 20 (2000) 7-12.
9. Shneiderman, B.: 3D or Not 3D: When and Why Does it Work?, invited talk at Web3D: 7th International Conference on 3D Web Technology, Tempe, AZ (2002)
10. Spenke, M.: Visualization and Interactive Analysis of Blood Parameters with InfoZoom, in [4] 159-172
11. Tufte, E.R.: The Visual Display of Quantitative Information, Graphics Press (1982)
12. USRDS, The United States Renal data system, http://www.usrds.org
13. Weber, M., Alexa, M., Mueller, W.: Visualizing Time-Series on Spirals. Proc. of the IEEE InfoVis Symposium, IEEE Press, Los Alamitos, CA (2001)

Sonification of time dependent data

Monique Noirhomme-Fraiture[1], Olivier Schöller[1], Christophe Demoulin[1], Simeon Simoff[2]

[1] University of Namur, Institut d'Informatique,
5000 Namur, Belgium
mno@info.fundp.ac.be

[2] University of Technology Sydney, Faculty of Information Technology,
NSW2007 Sydney, Australia
simeon@it.uts.edu.au

Abstract. This paper presents the results of experiments with sonification of 2D and 3D time dependent data. A number of sonification means for these experiments have been implemented. An Internet Web site was created where sound sequences were presented and could be evaluated by the participants in the experiment. All participants that performed the tests also needed to fill an evaluation questionnaire. The purpose of the experimentation was to determine how the sonification of two and three-dimensional graphs can support or be an alternative to visually displayed graphs. The paper concludes discussion of the results and the issues related with the experiments.

1 Introduction

Visual data mining is a part of the KDD process [1], which places an emphasis on visualisation techniques and human cognition to identify patterns in a data set. [1] identified three different scenarios for visual data mining, two of which are connected actually with the visualisation of final or intermediate results and one operates directly with visual representation of the data. The design of data visualisation techniques, in broad sense, is the formal definition of the rules for translation of data into graphics. Generally, the term 'information visualisation' has been related to the visualisation of large volumes of abstract data. The basic assumption is that large and normally incomprehensible amounts of data can be reduced to a form that can be understood and interpreted by a human through the use of visualisation techniques. The process of finding the appropriate visualisation is not a trivial one. A number of works offer some results that can be applied as guiding heuristics. For example, [2] defined the Proximity Compatibility Principles (PCP) for various visualization methods in terms of tasks, data and displays - if a task requires the integration of multiple data variables, they should be bundled in proximity in an integrated display. Based on this principle authors have concluded that 3D graphs do not have an advantage over 2D graphs for scientific visualisation (which may not necessarily hold for visual data mining).

Visual data mining relies heavily on human visual processing channel and utilises human cognition overall. The visual data mining cycles are shown in Fig. 1. In most systems, visualisation is used to represent the output of conventional data mining algorithms (the path shown in Fig. 1a). Fig. 2 shows an example of visualisation of the output of an association rule mining algorithm. In this case, visualisation assists to comprehend the output of the data mining algorithms. Fig. 1b shows the visual data mining cycle when visualisation is applied to the original or pre-processed data. In this case, the discovery of the patterns and dependencies is left to the capacity of the human visual reasoning system. The success of the exercise depends on the metaphor selected to visualise the input data [3].

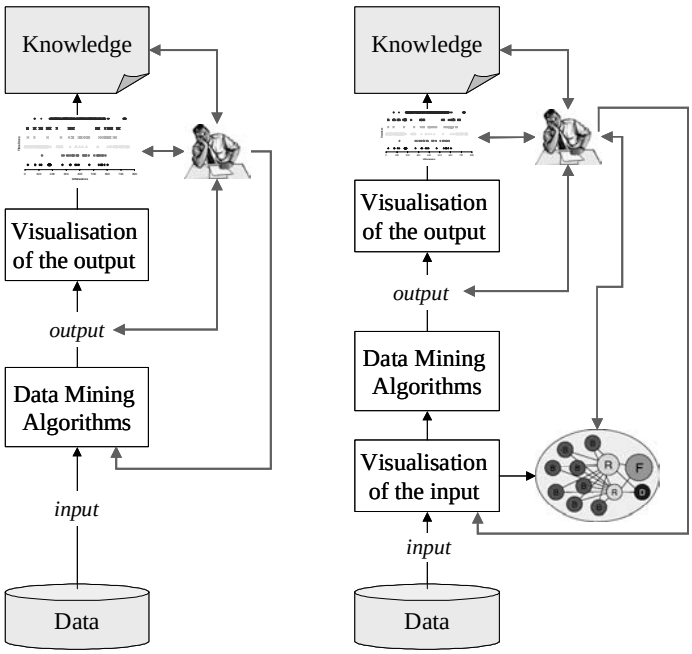


Fig. 1. Visualisation and visual data mining

Although human visual processing system remains a powerful 'tool' that can be used in data mining, there are other perceptual channels that seem to be underused. Our capability to distinguish harmonies in audio sequences (not necessarily musical ones) is one possibility to complement the visual channel. Such approach can be summarised as 'What You Hear Is What You See'. The idea of combining the visual and audio channels is illustrated in Fig. 3. The conversion of data into a sound signal is known as sonification. Similar to the application of visualisation techniques in Fig. 1b, sonification can be used both for representing the input and/or the output of the data mining algorithms.

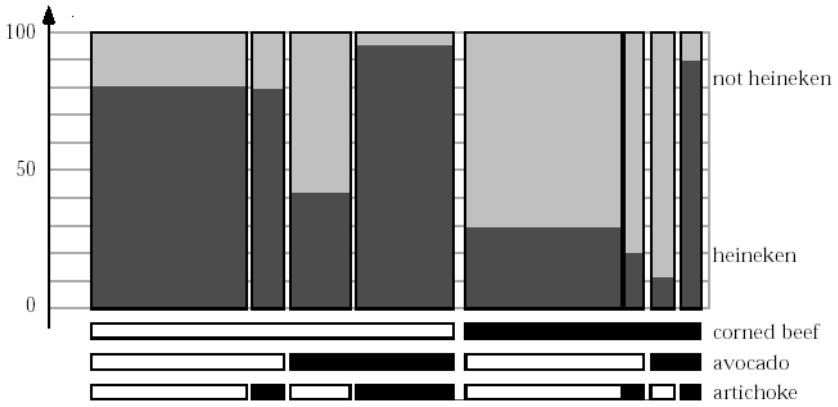


Fig. 2. Example of visualisation of the output of an association rule miner.

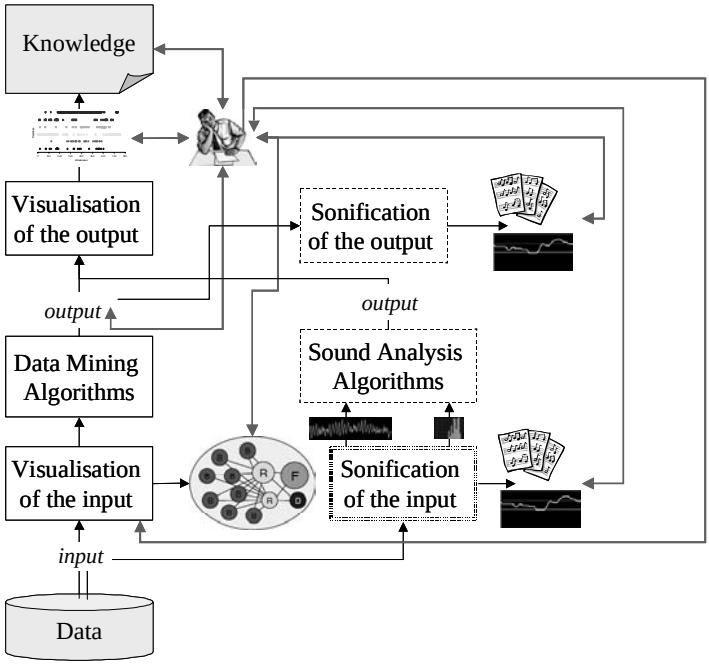


Fig. 3. Combining visual data mining and sonification

In visual data mining, sonification should be synchronised with the visualisation technique. Further, in this paper we discuss the issues connected with designing such data mining techniques and present an example of a practical implementation of combined technique. We briefly discuss the characteristics of the sound that are suitable for such approach, the actual sonification process, the design of the overall combined technique and the results of the experiments conducted with proposed technique.

2 Characteristics of sound for time dependent data representation

Several researchers have investigated the use of sound as means for data representation [4-11]. In this context, the important feature of the sound is that it has a sequential nature, having particular duration and evolving as a function of time. A sound sequence has to be heard in a given order, i.e. it is not possible to hear the end before the beginning[1]. Similarly, a time series depends on time and have the same sequential characteristics. Consequently sound provides good means to represent time series.

3 Sonification

The easiest way to transform time dependent data into sound is to map the data to frequencies by using linear as well as chromatic scale mappings. We call this process a pitch-based mapping. We compute the minimum and maximum data values from the chosen series and map this data interval into a frequency range, chosen in advance. Each value of the series is then mapped into a frequency. To avoid too large, non-realistic intervals, we first discard outliers (see below).

Another pre-treatment is the smoothing of the series. In fact, if we map all the points of a series into a sound, we will hear rather inconsistent sounds. A first treatment consists in smoothing the series by a standard mean, for example, by moving average method. After that, we map the smoothed curve into pitch. Beat drums can be used to enhance the shape of the curve (see below).

3.1 Detection of outliers

To detect statistically the values of the outliers, a confidence interval is computed at each time t., based on the normal distribution. Once a data value is detected outside the confidence interval, the corresponding time value is stored and sonified at the experiment phase.

3.2 Beat drums mapping

The rhythm of a beat drum increases with respect to the rate of growth of the curve (i.e. the first derivative).

[1] Images and drawings do not have such constraint. Strictly speaking digital sound recording can provide access to an arbitrary section of the sound fragment, even reproduce the sound in reverse order, which is beyond the scope of this paper.

3.3 Stereo panning

Variation of the stereo acoustics is introduced, for example, an increase of the volume of the right speaker and decrease of the volume of the left speaker.

3.4 3D Curve

When the time series is given at each time, not by a single value but by a function of values, we decide to "hear" at each discrete time the function. We can also choose to cut the surface at a certain level and to hear "continuously" the obtained curve as a function of time. We call these transformations respectively horizontal and vertical travelling. An example of a 3D data surface for sonification is shown in Fig. 4.

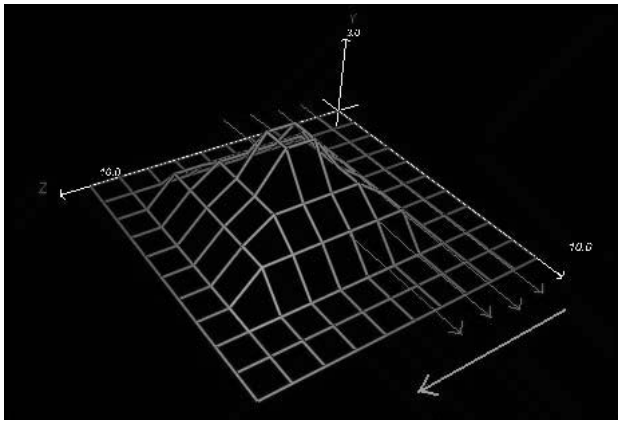


Fig. 4. Example of a surface that can be sonified

4. Prototype implementation

The prototype has been implemented in Java programming language, using the MIDI package of the Java Sound API[2] [12]. The MIDI sequence is constructed before the actual playback. When the designer starts the sonification, the whole sequence is computed. Then computed sequence is sent to the MIDI sequencer for playback.

5. Experimentation

The purpose of the experimentation is to determine how the sonification of two and three-dimensional graphs can complement or be an alternative to visually displayed

[2] API – **A**pplication **P**rogramming **I**nterface

graphs. An Internet Web site has been created, where sound sequences are presented and can be evaluated by the visitors. The site contains questionnaire that has to be filled in by visitors performing the test. The structure of the questionnaire is the following one:

A. Identification of the user: name, age, gender, title/position, e-mail address. These data are used to identify the subject and to validate the answer.

B. Ability: field of activity, musical experience (instrument played, practicing period), self-evaluation of musical level (from 'no experience' to 'expert level').

C. 2D evaluation: 2D evaluation is divided into three subtasks*:*

Part C.a: **Explanation** about the four sonification techniques used: pitch based only, beat drums, stereo and extreme values detection. Each of them is briefly described and at least one example is given.

Part C.b: **Application test**: four sequences are presented to the user. Each time precise questions are asked:

Question 1: Annual sheep population in England and Wales between 1867 and 1939 (see Fig. 5).
- Were there more sheep in 1867 then in 1939?
- In your opinion, when (which year) did the sheep population reach the minimum?

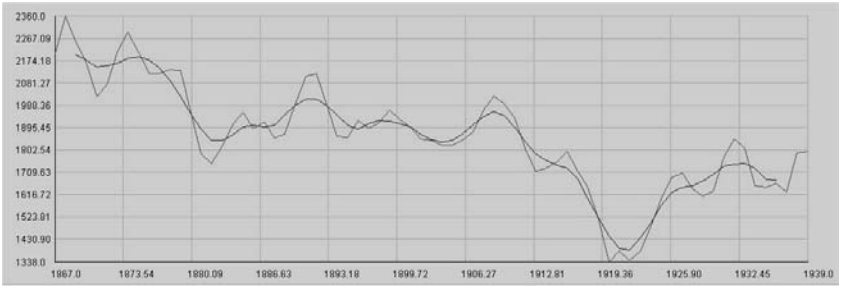


Fig. 5. Annual sheep population in England and Wales between 1867 and 1939

This question aims to evaluate if subject can perceive a global trend in the series and to understand if the relation with the time scale is done. For each sequence, beat drums and stereo mapping are added to enhance the pitch-based sonification.

Question 2 aims to identify whether extreme values are detected. Question 3 aims to identify whether seasonal trend can be detected. Question 4 is focused on trend

identification. For each question, the subject must specify the number of times he listened to the sonified data before answering the question.

Part C.c: **Subject preferences**: four other questions aim to evaluate subject preference:

1. Choice of instrument for pitch mapping. The user is asked to hear the sonification of the same data, but with different MIDI instruments for the pitch mapping. The user is asked to grade on the scale 0 to 10 the different instruments (acoustic grand, steel string guitar, violin, synthstrings 2, pan flute). The instruments proposed are very different and belong to a specific MIDI group such as piano, guitar or to string group.

2. Choice of instrument for drums mapping. Different instruments for the beat drums mapping are presented and have to be marked (Celesta, Slap Bass 1, Timpani, Tinkle Bell, Woodstock).

3. Choice of sonification technique. The subject has to give his opinion on the different mapping techniques: pitch, beat drums, stereo mapping, extreme values detection and on the sonification in general. The four levels are proposed: useless, sometimes useful, always useful and essential.

4. Open question: “Please tell us what you think about our project, our applications, this web page or anything else that comes to mind”.

D. 3D Evaluation. The same schema is used for 3D curves: explanation about the sonification method, the application test (3 questions), subject preferences. The sonification method is based on a cutting of the initial surface following some direction: vertical (see Fig. 6), horizontal (see Fig. 7) or diagonal.

Each 2D line obtained by the cutting process can be heard, similar to the 2D case. The different lines, proposed one after the other, are separated by a specific sound. Beat drums can still be added to pitch mapping.

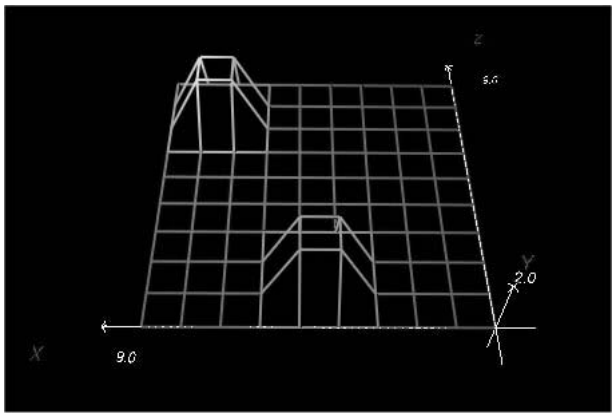


Fig. 6. Data for vertical travelling

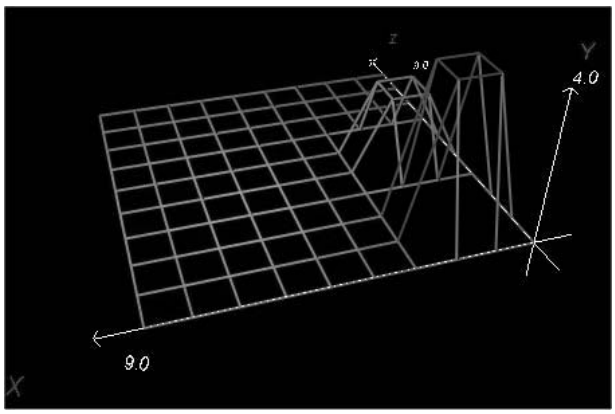


Fig. 7. Data for horizontal travelling

6 Results

Below we present the results of the experiments.

6.1 The sample

23 visitors answered the questions for the case of a 2D visualisation and 18 visitors - for the case of 3D visualisation. A large part of the sample (9) includes people working in the computer science area, who have limited musical experience or no experience at all. To see if people with musical experience get a better score, we have compared the average score in both groups. The influence of the musical and computer science background on the results is presented in Table 1 and Table **2**, respectively. The score obtained for each question is the number of good answers, normalised on a

score out of 100. The average score is the mean of the scores for the different questions.

Table 1. The influence of musical background on results

	2D sonification		3D sonification
	Av. score/100	**N**	**Av. score/100**
Musical background	76	9	40
No experience	66	14	42
		23	

The same comparison was done for computer scientist.

Table 2. The influence of computer science background on results

	2D sonification		3D sonification
	Av. score/100	**N**	**Av. score/100**
Computer science	72	11	41
Others	69	12	40
		23	

Having a musical or a computer science background gives a minor advantage in using sonification of 2D curves. The differences are not significant. We do not observe any difference for the 3D sonification case. The way 3D case has been implemented is rather complex and uses a good spatial representation. Musical or computer experience has a minor influence on the result in this case.

6.2 Results in 2D

The following questions were included in the questionnaire, targeting 2D sonification:

Qu1 Annual sheep population in England and Wales between 1867 and 1939
- 1.1 Were there more sheep in 1867 than in 1939?
- 1.2 About which year did the sheep population reach the minimum?

Qu2 Daily morning temperature of an adult woman during two months
- 2.1 Did she have fever during the period?
- 2.2 If yes, for how long did she have the fever?

Qu3 Monthly electricity production in Australia between January 1956 and August 1995
- 3.1 Is the electricity production in Australia lower in 1956 than in 1995?
- 3.2 How would you categorise the evolution of electricity production in Australia: as linear or as exponential?

3.3 Is the evolution of electricity production in Australia characterised by seasonal trend?

Qu4 Monthly Minneapolis public drunkenness intakes between January 1966 and July 1978 (151 months)

4.1 Were there more intakes in 1966 than in 1978 ?

4.2 Is the evolution of public drunkenness intakes linear?

The results are summarised in Table 3.

Table 3. Summary of the results for 2D

		Correct	Wrong	No idea
Qu1	1.1	17	5	1
	1.2	17	6	-
Qu2	2.1	23	0	0
	2.2	13	10	0
Qu3	3.1	22	1	0
	3.2	12	11	0
	3.3	16	6	1
Qu4	4.1	20	2	1
	4.2	19	3	1

6.3 Results in 3D

The following questions were included in the questionnaire, targeting 2D sonification:

Qu1 A 3D graph containing 2 bumps has been sonified. The selected mapping is the vertical travelling and the sonification starts from the bottom right corner.

- If the grid below (3 x 3) represents the graph, where are these 2 bumps located?
- Do they have the same height?

Qu2 Same kind of questions with respect to horizontal travelling.

Qu3 Same kind of questions with respect to diagonal travelling.

The results are summarised in Table 4.

Table 4. Summary of the results for 3D

		Correct		Wrong	No idea
		2 correct	*1 correct*		
Qu1	1.1 Trend	4	6	8	-
	1.2 Value	11		4	3
Qu2	2.1 Outlyers	4	6	8	0
	2.2	12		33	0
Qu3	3.1 General trend	1	11	6	-
	3.2	10		6	2

7 Discussion

There are some issues related to the design of the experiments that could have influenced the outcome of the experimentation:

- In a graphical representation, if you want to identify a particular point, you need to find the information concerning that point on each axis. In the experiment, we provided little information about the scale (for example, see Fig. 6 and Fig. 7). The wording gives the limits for the time period. The lack of scaling information could have caused some difficulty in identifying particular points or sub-periods.
- The outcomes in the case of sonification of a 3D graph are worse than in the case of 2D. It is necessary to take in account that the sonification of a 3D graphical representation is more difficult then the sonification of a 2D graphical representation. A possible reason could be that the sonification technique is based on visual representation and does not use sound properties, but surface properties, as seen in a 3 axis referential.

An important issue becomes the correspondence between visual and audio representations of the data. Consistent representations should provide audio representations that allow transitions from 3D to 2D projections in terms of corresponding sound representations.

8 Conclusions

Overall, the results of the experimentation on sonification of time dependent data leave optimism for further investigation of sound as medium for presenting information. The sound can be an effective complementary interface to the visual interface for data representation. Similar results were presented by Alty [5] for people with disabilities. On the other hand, the experimentation with sonification of surfaces in

3D space did not efficiently support the visual representation and certainly could not replace it (at least for the way it had been implemented).

In general, this experimental work contributes to the research efforts on bringing other (non-visual) channels for information and data processing. This research area, that can be labelled as 'perceptual data mining', is focused on interactive systems that support rich perceptual – visual, audio, tactile – interaction between the human and the data representation. Such systems are expected to play significant role in assisting data understanding and supporting pattern discovery process, utilising human information processing capabilities.

Acknowledgements

We thank our colleagues Anne de Baenst and Florence Collot for their help in the preparation of this paper.

References

1. Ankerst, M.: Visual Data Mining: *Ph.D. thesis.* Faculty of Mathematics and Computer Science, University of Munich, Dissertation.de, 2000.
2. Wickens, C.D. and Carswell, C.M.: The proximity compatibility principle: Its psychological foundation and relevance to display design. Human Factors. 37 (1995) 473-495
3. Simoff, S.J.: Towards the development of environments for designing visualisation support for visual data mining. Proceedings Int. Workshop on Visual Data Mining, 12th European Conference on Machine Learning and 5th European Conference on Principles and Practice of Knowledge Discovery in Databases ECML/PKDD2001. Freiburg, Germany (2001) 93-106
4. Anrijs, K.: The use of sound in 3d representations of symbolic objects. Namur, Facultés Universitaires Notre-Dame de la Paix, 1999.
5. Alty, J.L., Rigas, D. and Vickers, P.: Using music as a communication medium. Proceedings of CHI'97 (1997)
6. Brewster, S.A., Wright, P.C. and Edwards, A.D.N.: An evaluation of earcons for use in auditory human-computer interfaces. Proceedings of Inter CHI'93 (1993)
7. Conversy, S. and Beaudouin-Lafon, M.: Le son dans les applications interactives. Paris, Laboratoire de Recherche en Informatique, Université de Paris-Sud, 1995.
8. Hermann, T.: Data exploration by sonification, 1999. Available from [http://www.techfak.uni-bielefeld.de/techfak/ags/ni/projects/datamining/datamin_e.htm].

9. Kramer, G.: An introduction to auditory display, Auditory Display: Sonification, Audification, and Auditory Interfaces. Santa Fe Institute Studies in the Sciences of Complexity (1994)
10. Noirhomme-Fraiture, M.: Le son dans les interfaces IHM: Application à la représen-tation de données multivariées complexes. Actes des 2èmes Journées Multimédia. Namur (2000)
11. Sahyun, S.C.: A comparison of auditory and visual graphs for use in physics and mathematics, Oregon State University, 1999. Available from [http://www.physics.orst.edu/sahyun/thesis/].
12. Java: Javasound API programmer's guide, Sun Microsystems, 2000.

Author Index