Proceedings of the
ACM SIGKDD
Workshop on
Visual
Analytics and
Knowledge
Discovery

VAKD '09

www.hiit.fi/vakd09

A full-day workshop in conjunction with
the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining
in Paris, France, on 28 June 2009.

Kai Puolamäki, editor

Otaniemi, June 2009

Proceedings of the
ACM SIGKDD Workshop on Visual Analytics and Knowledge Discovery

Kai Puolamäki, editor

ISBN 978-1-60558-670-0

Otaniemi, June 2009

ACM SIGKDD Workshop on Visual Analytics and Knowledge Discovery: Integrating Automated Analysis with Interactive Exploration

General Chairs

Kai Puolamäki & Heikki Mannila (Helsinki Institute for Information Technology HIIT)
Alessio Bertone & Silvia Miksch (Danube University Krems)

Challenge Chairs

Mark A. Whiting & Jean Scholtz (Pacific Northwest National Laboratory)

Program Committee

Fosca Giannotti & Dino Pedreschi & Salvatore Rinzivillo (University of Pisa)
Georges Grinstein (University of Massachusetts Lowell)
Otto Huisman (International Institute of Geo-Information Science and Earth Observation)
Daniel A. Keim (University of Konstanz)
Catherine Plaisant (Human-Computer Interaction Lab, University of Maryland)
Tobias Schreck (Technische Universitaet Darmstadt)
Mike Sips (Max-Planck-Institut fuer Informatik)
Dimitrios Tzovaras (Center for Research & Technology Hellas)
Anders Ynnerman & Jimmy Johansson (Linköping University)

External Reviewers

Michele Coscia & Sami Hanhijärvi & Tim Lammarsch & Georg Pölzlbauer &
Alessandra Raffaeta & Andreas Rauber & Roberto Trasarti

Sponsors

VisMaster, a European FP7 Coordination Action Project focused on Visual Analytics
PASCAL2 – Pattern Analysis, Statistical Modelling and Computational Learning
Helsinki Institute for Information Technology HIIT
Danube University Krems, Departement of Information and Knowledge Engineering (DUK)
National Visualization and Analytics Center (NVAC)

Table of Contents

Interactive Spatio-Temporal Cluster Analysis of VAST Challenge 2008 Datasets

Gennady Andrienko Natalia Andrienko
Fraunhofer Institute IAIS (Intelligent Analysis and Information Systems)
Schloss Birlinghoven, Sankt Augustin
D-53757, Germany
+49 2241 142486
{gennady.andrienko, natalia.andrienko}@iais.fraunhofer.de

ABSTRACT
We describe a visual analytics method supporting the analysis of two different types of spatio-temporal data, point events and trajectories of moving agents. The method combines clustering with interactive visual displays, in particular, map and space-time cube. We demonstrate the use of the method by applying it to two datasets from the VAST Challenge 2008: evacuation traces (trajectories of people movement) and landings and interdictions of migrant boats (point events).

Categories and Subject Descriptors
H.1.2 [User/Machine Systems]: Human information processing – Visual Analytics; I.6.9 [Visualization]: information visualization.

Keywords
Spatio-temporal data, movement data, trajectory, movement patterns, movement behavior, point events, clustering, visual analytics, exploratory data analysis, visualization.

1. INTRODUCTION
Clustering, i.e. discovery and interpretation of groups of objects having similar properties and/or behaviors, is one of the most common operations in exploration and analysis of various kinds of data. Clustering is particularly useful in exploring and analyzing large amounts of data since it allows an analyst to consider groups of objects rather than individual objects, which are too numerous. However, clustering is not a standalone method of analysis whose outcomes can be immediately used for whatever purposes (e.g. decision making). An essential part of the analysis is interpretation of the clusters by a human analyst; only in this way they acquire meaning and value. To enable the interpretation, the results of clustering need to be appropriately presented to the analyst. Visual and interactive techniques play here a key role.

In clustering, objects are often treated as points in multi-dimensional space of properties. However, this approach may be inadequate for structurally complex objects, such as trajectories of moving entities and other kinds of spatio-temporal data. Thus, trajectories are characterized by a number of non-trivial and heterogeneous properties including the geometric shape of the path, its position in space, the life span, and the dynamics, i.e. the way in which the spatial location, speed, direction and other point-related attributes of the movement change over time. Each of these diverse properties needs to be handled in its own way.

There are two main approaches to clustering complex data: (i) defining ad hoc notions of clustering and devising clustering algorithms tailored to the specific data type; and (ii) applying generic notions of clustering and generic clustering algorithms by defining a specific distance function, which measures the similarity between data items. In the second case, the specifics of the data are completely encapsulated in the distance function.

In our research, we pursue the second approach. We use a generic density-based clustering algorithm OPTICS [5], which belongs to the DBSCAN [6] family. Advantages of these methods are tolerance to noise and capability to discover arbitrarily shaped clusters. A brief description of OPTICS is given in [11]. We use an implementation of OPTICS that allows different distance functions to be applied. We have developed a library of distance functions oriented to trajectories and to point events.

2. DISTANCE FUNCTIONS
The clustering tool has three parameters: the spatial distance threshold *maxD*, the minimum number of neighbors of a core object *MinNbs*, and the distance function *F*. The second parameter requires some explanation. Neighbors of an object are such objects whose distances to this object are below the distance threshold *maxD*. A core object is an object located in a dense region, i.e. inside some cluster. The parameter *MinNbs* defines the desired density inside a cluster. Additionally to these, some of the distance functions have their own parameters.

As we argue in [11], it would not be reasonable to create a single distance function for trajectories that accounts for all their diverse properties. On the one hand, not all characteristics of trajectories may be simultaneously relevant in practical analysis tasks. On the other hand, clusters produced by means of such a universal function would be very difficult to interpret. A more reasonable approach is to give the analyst a set of relatively simple distance functions dealing with different properties of trajectories and provide the possibility to combine them in the process of analysis.

We suggest and instrumentally support a step-wise analytical procedure called “progressive clustering”. The main idea is that a simple distance function with a clear meaning and principle of

VAKD'09, June 28, 2009, Paris, France.

work can be applied on each step, which leads to easily interpretable outcomes. However, successive application of several different functions enables sophisticated analyses through gradual refinement of earlier obtained results.

Our distance functions for trajectories are described in [2] and [11]. Here we briefly describe the functions we have used in analyzing the VAST Challenge data [8]. The function *"common destination"* computes the distance in space between the ending points of two trajectories. This is the distance on the Earth surface if the positions are specified in geographical coordinates (latitudes and longitudes) or the Euclidean distance otherwise. The family of functions *"check points"* computes the distances in space between the starting points of two trajectories, between the ending points, and between one or more intermediate check points, and returns the average of the distances. The functions differ in the way of choosing the check points:

- *k points by time*: the user-specified number of intermediate points *k* are selected so as to keep the time intervals between them approximately constant;
- *k points by distance*: *k* points are selected so as to keep the spatial distances between them approximately constant;
- *time steps*: the user specifies the desired temporal distance between the check points;
- *distance steps*: the user specifies the desired spatial distance between the check points.

For point events, we have two distance functions. The first one returns the distance in space between the positions of the events. The second function, spatio-temporal distance, computes the distance in space and time. For this purpose, it asks the user for an additional parameter: the temporal distance threshold *maxT*, which is assumed to be equivalent to the spatial distance threshold *maxD*. The function finds the spatial distance *d* between the positions of two events and the temporal distance *t* between the times of their occurrence. Then it proportionally transforms *t* into an equivalent spatial distance *d′* and combines *d* and *d′* in a single distance according to the formula of the Euclidean distance.

3. MINI-CHALLENGE "EVACUATION TRACES"

Clustering is especially helpful in analyzing large datasets. The dataset for the mini-challenge "Evacuation traces" is quite small as it contains only 82 trajectories. Cluster analysis is not really necessary for answering the questions of the mini-challenge. However, it can aptly complement purely visual and interactive techniques, as will be shown below, and the same or similar procedure will be applicable and effective in case of a much larger dataset. We shall not describe the whole analysis of the dataset and finding answers to all questions but only demonstrate the use of the clustering techniques. A report about a complete analysis (done mostly with the use of other methods) is available at http://vac.nist.gov/2008/entries/andrienkoevac/index.htm; see also a summary in [3].

3.1 Clustering by "common fate"

The first question we try to answer concerns the fates of the people who were in the building before the explosion and could be affected by the incident: who managed to leave the building and who did not? To answer this question, we cluster the trajectories using the distance function "common destination". After a few experiments with the distance threshold *maxD*, we obtain easily interpretable clusters, which are presented in Figures 1-3. The trajectories are represented by lines; the small hollow squares mark the starting points and the bigger filled squares mark the ending points. In Figure 1, there are four clusters of trajectories that evidently belong to people who managed to leave the building: the ending positions of the trajectories can be interpreted as being at the exits. The two clusters shown in Figure 2 consist of trajectories ending inside the building; hence, the people did not manage to evacuate because they were affected by the incident. In Figure 3, there are five trajectories that do not fit in any cluster. These trajectories need to be considered in detail: the terrorist or terrorists may be among the people who left these traces.

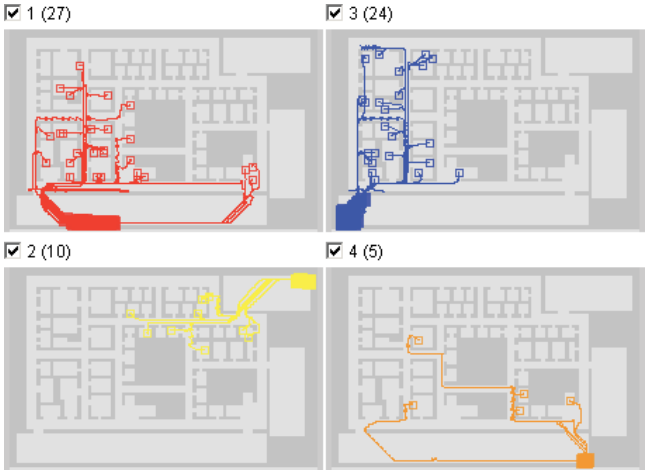


Figure 1. The clusters of the trajectories of the people who evidently managed to leave the building.

Figure 2. The clusters of the trajectories of the possible casualties.

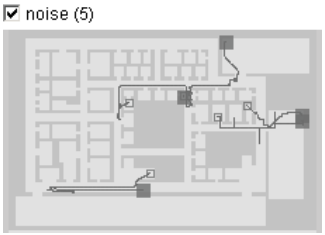


Figure 3. The trajectories that do not belong to any cluster.

The clusters can be very conveniently used for dynamic filtering of the trajectories: the checkboxes above the images of the clusters hide or expose their members. Thus, we can select the

clusters corresponding to the possible casualties and find out, with the help of the space-time cube [9][10] (Figure 4), that the people whose trajectories belong to cluster 5 (violet) stopped moving significantly earlier than the people from the second group (cluster 6, green). This means that the former group of people was closer to the place of the explosion than the latter group.

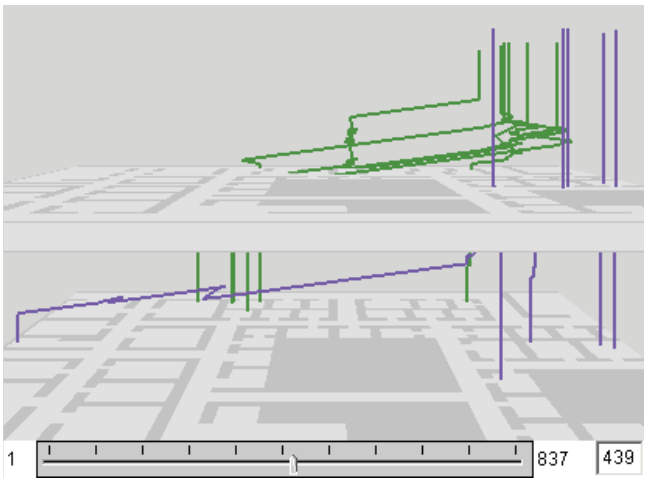


Figure 4. The space-time cube shows the trajectories of the possible casualties. The position of the movable horizontal plane corresponds to the time moment after which there was no movement in the "violet" cluster.

Now we can select the group of people corresponding to the "noise" (Figure 3) and explore their behaviors looking, in particular, whether they visited the areas where the identified casualties stopped moving. We shall not describe this analysis here. The result is that we identify a person who visited the probable area of the explosion before the explosion occurred (Ramon Katalanow), a person who never moved or, possibly, left his RFID tag in his original place (Francisco Salter), a possible casualty who stopped moving later than the others (Olive Palmer), and a person who was close to the "green" group when they stopped moving (Cecil Dennison).

3.2 Clustering by similar routes

Now we would like to check whether any of the people who left the building had extraordinary routes of the movement, which may indicate their possible participation in the incident. As in the previous case, we want to use clustering for the separation of "normal" routes from peculiar ones: the former will be grouped in clusters and the latter will be marked as noise. In our library of distance functions, we have a function "route similarity" [2][11], which measures the correspondence between the geometric shapes of two trajectories and the closeness of their spatial positions. This function appears suitable for our purposes. However, it does not find any clusters in this particular dataset. The reason is a very high fluctuation of the positions in the trajectories, illustrated in Figure 5. According to the "route similarity" function, the two trajectories shown in Figure 5 are very distant from each other, although they appear very similar if the fluctuations are ignored. Hence, we need to use a distance function less sensitive to fluctuations.

The family of distance functions "check points" can work in this case: if the number of check points is small, the impact of the fluctuations is also small. The functions "*k* points by time" and "time steps" do not suit well to our purposes: they are sensitive to the differences in the starting moments and the velocities of the movement whereas we want to consider only the routes. The function "distance steps" is not a good choice either: it is hard to select a suitable step because of a large variation of the lengths of the trajectories (from 0.5 to 189). The remaining function "*k* points by distance" works adequately. We find out that the results of the clustering do not substantially change when we vary the number of the intermediate check points (parameter *k*) in the range from 5 to 25.

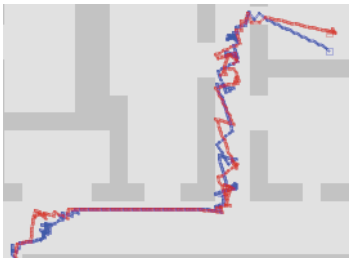


Figure 5. The fluctuations of the positions in the trajectories.

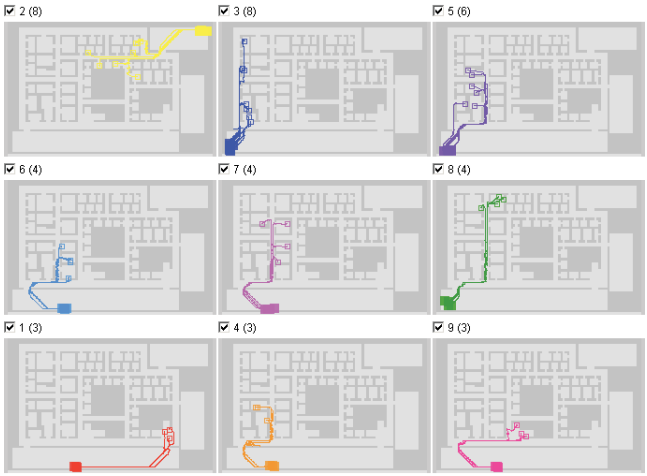


Figure 6. The trajectories of the people who left the building (see Figure 1) have been clustered according to the routes.

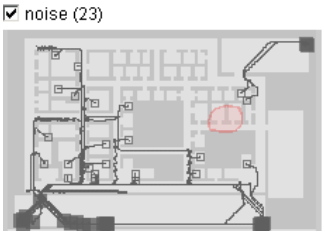


Figure 7. The trajectories not fitting in any cluster. The pink spot marks the identified area of the explosion.

Figure 6 presents the clusters discovered among the trajectories of the people who left the building (Figure 1) with the use of the distance function "*k* points by distance" where *k*=15. Figure 7 shows the remaining 23 trajectories, which have not been put in clusters. We can say that the clusters correspond to normal, logical routes of the movement. The remaining trajectories with peculiar routes need to be additionally examined. However, there is no need in a detailed examination of each trajectory. It is

sufficient to have a close look at the trajectories of the people who either visited the place of the explosion or interacted with some of the suspects or the victims. As can be seen in Figure 7, none of the uncommon trajectories passes the identified place of the explosion. Hence, we may focus on finding and examining possible interactions between the people who had these trajectories and the possible victims or suspects, whose trajectories are shown in Figures 2 and 3.

We shall not describe the further analysis in detail. In brief, we applied our computational tool for finding indications of probable interactions, i.e. cases of spatial proximity of moving agents. We found that only three of 23 people might have interactions with some of the victims or suspects. One of them was in the same room as Cecil Dennison (one of the suspects) till moment 262, when the latter left the room. The other two people might have interacted with Olive Palmer, a possible victim who stopped moving later than the other victims (Figure 8). In a case of a real investigation, it would be reasonable to interrogate these three persons.

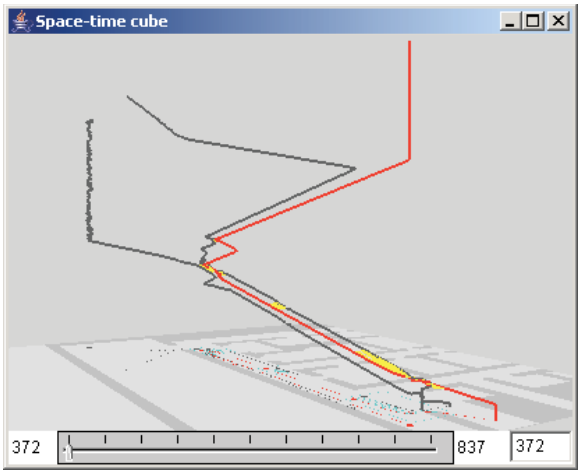


Figure 8. Yellow marks the probable interactions between one of the possible casualties, whose trajectory is in red, and two other people.

Hence, in the mini-challenge "Evacuation traces", the density-based clustering of trajectories was useful for two purposes. First, we divided people into groups according to their fates. Two of the groups were interpreted as probable casualties, the others as survivors. Second, we separated normal movement behaviors from peculiar ones. Such separation is possible owing to the specific feature of the density-based clustering, which does not put an object in a cluster if it is not sufficiently similar to others. The flexibility of the clustering tool allows us to choose distance functions according to the goals of the analysis. As will be seen in the next section, the same clustering tool is applicable to a different type of data provided that a suitable distance function is used.

4. MINI-CHALLENGE "MIGRANT BOATS"

The dataset for this mini-challenge consists of 917 records about landings and interdictions of migrant boats with the spatial positions (geographical coordinates) and times of the landing or interdiction events. The time span of the dataset is three years from the beginning of 2005 till the end of 2007. Among the questions of the mini-challenge, there are questions about the choice of the landing sites over the three years and about the geographic patterns of the interdictions over the three years. These questions may be answered with the help of clustering: using an appropriate distance function, we can discover spatio-temporal clusters of events, in particular, landings or interdictions in the same or close places shortly one after another.

4.1 Spatio-temporal clusters of landings

From the whole set of records, we select only the records about the landings. There are 441 such records. We apply the clustering tool with the distance function "spatio-temporal distance" described in Section 2. With 50 km as the spatial threshold and 21 days as the temporal threshold, we obtain the clusters shown in Figure 9 on a map and in a space-time cube (the use of space-time cube for visual exploration of event data is described in [4] and [7]). The scatterplot in Figure 10 aptly complements these two views. The horizontal and vertical dimensions of the plot represent the time and the latitude of the landings, respectively.

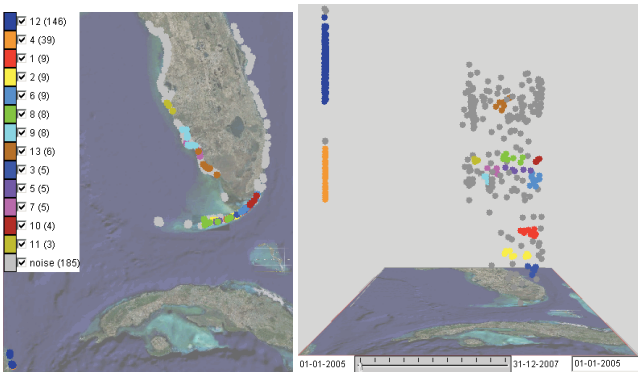


Figure 9. Spatio-temporal clusters of landings on a map (left) and in a space-time cube (right).

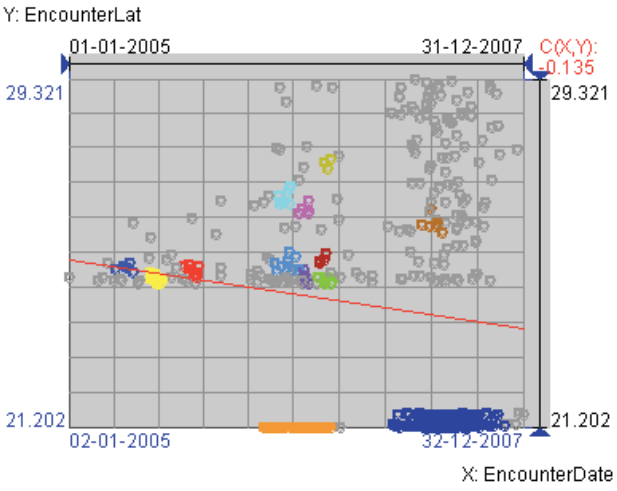


Figure 10. The clusters of landings shown on a scatterplot.

There are two big spatio-temporal clusters of landings located at the coast of Mexico. In the space-time cube, these two clusters appear as vertically aligned dots colored in orange and dark blue. In the scatterplot, the corresponding dots are aligned horizontally.

The temporal extent of the orange cluster, which consists of 39 landings, is from April 15 till September 22, 2006. The dark blue cluster consists of 146 landings, which occurred during the period from February 21 till November 18, 2007. Hence, both the number of landings at the Mexican coast and the duration of the period of active migration significantly increased from 2006 to 2007. As can be seen from the space-time cube and the scatterplot, there were no landings in this area before April 2006.

The spatio-temporal clusters of landings at the coast of Florida and nearby islands are much smaller. In 2005, there were 3 clusters of landings, shown in blue, yellow, and red (5, 9, and 9 landings, respectively); all of them occurred on the islands of the Florida Keys archipelago. In 2006, there were 4 clusters of landings on the Florida Keys islands (light blue, violet, green, and dark red; 26 events in total) and 3 clusters of landings on the western coast of Florida (light cyan, pink, and dark yellow; 16 events in total). In 2007 there was only one spatio-temporal cluster consisting of 6 landings. It is shown in brown; the landings occurred on the western coast of Florida. This may mean that the migrants changed the strategy and avoided repeated landings in the same areas in favor of more distributed targets. This may also mean that repeated attempts to reach the same place were intercepted by the coast guards.

4.2 Spatial clusters of landings

Another kind of analysis can be done by means of spatial clustering of the landing events irrespective of the time. For this purpose, we apply the distance function "spatial distance".

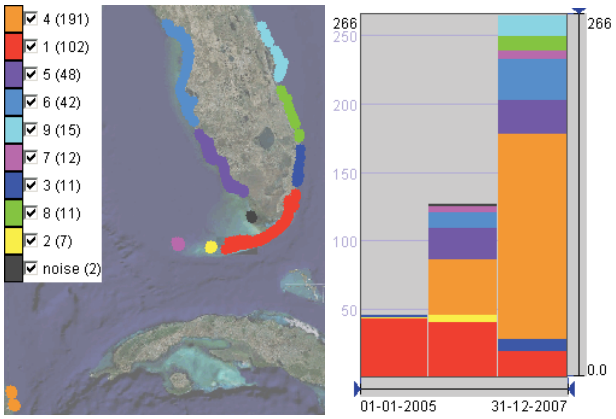


Figure 11. Left: spatial clusters of landings. Right: the distribution of the landings by years.

With the distance threshold 25km, we obtain the spatial clusters of landings demonstrated in Figure 11 left. The temporal histogram in Figure 11 right shows us how the destinations of the migrants changed over the three years. The bars of the histogram correspond to the years; they are divided into colored segments proportionally to the numbers of landings from the corresponding clusters. We can see that almost all landings in 2005 occurred on the Florida Keys archipelago (red cluster). In 2006, additional destinations appear: at the Mexican coast (orange), on the western coast of Florida (violet, light blue, and dark gray), and at the western end of Florida Keys (pink and yellow). In 2007, the number of landings on Florida Keys significantly decreases while the number of landings in Mexico dramatically increases. Besides, there is an eastern trend: many migrants land on the eastern coast of Florida, which did not occur in the previous years.

4.3 Clustering of the interdictions

Now we shall apply clustering to the interdiction events. In Figure 12, we see the spatio-temporal clusters discovered with the use of the distance function "spatio-temporal distance" (*maxD*=50 km; *maxT*=21 days). In Figure 13, we can see how the clusters and the remaining interdiction events ("noise") are distributed over the three years from 2005 to 2007. The temporal histogram in Figure 14 left shows us the sizes of the clusters and "noise" by years.

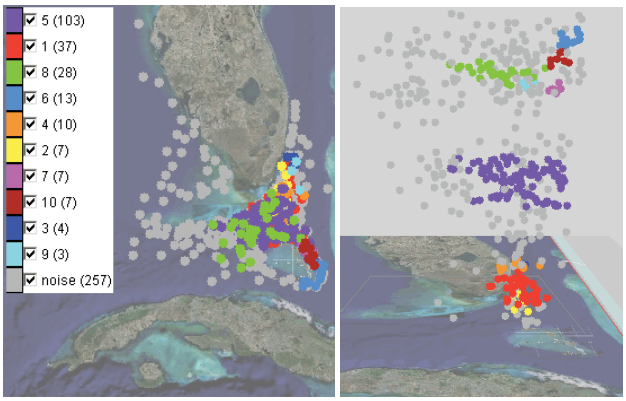


Figure 12. Spatio-temporal clusters of interdictions on a map (left) and in a space-time cube (right).

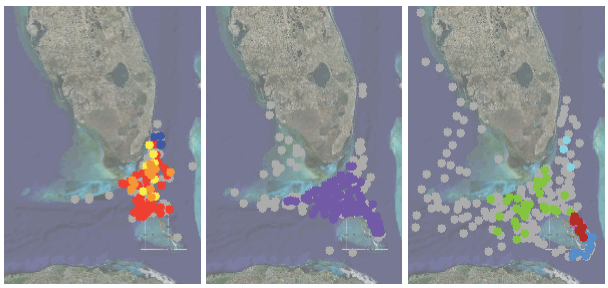


Figure 13. Spatio-temporal clusters of interdictions by years.

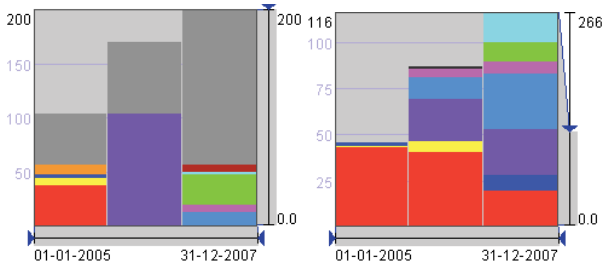


Figure 14. Left: the sizes of the clusters of the interdictions and the "noise" by years. Right: the landings in Florida and on nearby islands in the same years.

The spatio-temporal clusters of interdictions are generally larger than the spatio-temporal clusters of landings (Figure 9), except for the landings in Mexico. This refers not only to the number of events in a cluster but also to its spatial and temporal extent. The larger clusters mean that the interdiction events are spatially and temporally denser than the landing events. The highest spatio-temporal density of the interdictions is reached in 2006, when a

single cluster (violet) includes 103 out of 170 events, i.e. over 60%. Like in 2005, the events are concentrated in the area between Florida Keys and Isla Del Sueño, the origin of the migrant trips; however, the spatial extent is larger in 2006. In 2007, the spatial spreading of the interdictions further increases while the spatio-temporal density of the events decreases. This is signified by the larger number of smaller clusters; the largest cluster (light green) is smaller and looser than the largest clusters in the previous years. The ratio between the number of events in the clusters and the size of the "noise" (58 to 142) is much smaller in 2007 than in 2006 (103 to 67) and 2005 (58 to 48).

When we compare these observations with the observations concerning the landings (Sections 4.1 and 4.2), we can conclude that the strategy of the migrants changed over the three years: the migrants diversified their destinations and, evidently, the routes. This, apparently, made the coast guards extend the area of patrolling. Probably, the migrants hoped that the change of the strategy would make them harder to catch and thereby increase the success rate. If we compare the number of landings in Florida and on the nearby islands (visualized on a histogram in Figure 14 right) with the number of interdictions by years, we may conclude that the success rate, indeed, steadily increased over the three years. The ratio between the number of landings and the number of interdictions was 46:106 (0.43) in 2005, 88:170 (0.51) in 2006, and 116:200 (0.58) in 2007. In 2006 and 2007 there were also 41 and 150 landings and no interdictions in Mexico.

For the landing events, we used spatial clustering irrespective of the time, which produced meaningful spatial clusters (Section 4.2). However, this method of clustering does not work well for the interdictions: due to the high spatial density of the events, most of them are united in a single very large cluster. This does not give us new opportunities for the analysis.

Hence, in the mini-challenge "Migrant boats", the density-based clustering helped us to detect compact groups of events in space and time, to asses the spatio-temporal density of the events and its change over time, and to divide events into groups according to their spatial positions in order to examine the changes in the spatial distribution of the events over time.

5. CONCLUSION

Clustering in combination with interactive visual displays is a powerful instrument of data analysis, in particular, when the data are large and/or complex. Many clustering methods require the data to be represented as points in a multi-dimensional space of properties (in other terms, by feature vectors). However, for complex data with multiple heterogeneous properties there may be no adequate representation by feature vectors. An example of such a complex data type is trajectories of moving objects, characterized by the origin and destination, length, temporal extent, duration, geometrical shape, spatial orientation, dynamics (distribution of the speeds along the way), and, possibly, variation of other attributes during the movement.

A possible approach to the clustering of complex data types is the use of a generic clustering algorithm with a type-specific distance function, which properly accounts for the relevant properties depending on their nature. We have demonstrated this approach by applying the same clustering algorithm to two datasets of different types, trajectories of moving objects and point events distributed in space and time. We have also demonstrated that different distance functions oriented to the same type of data may be useful for different analysis tasks.

The clustering tool we use implements a density-based clustering algorithm, which does not strive to put each object in some cluster but finds compact groups of close (similar) objects and leaves the other objects ungrouped. In this way, it not only discovers frequent patterns (combinations of properties) but also enables the analyst to examine the variation of the data density (in terms of close properties) throughout the dataset. In the paper, we have demonstrated how the features of the algorithm are exploited in the analysis.

The VAST Challenge datasets [8] we have used in this paper are quite small; they could be effectively analyzed without the use of clustering. For larger datasets, clustering gives more significant advantages. Our clustering-based visual analytics tools work well with about 5,000 trajectories, i.e. the reaction time is appropriate for an interactive analysis. Clustering of 10,000 trajectories is possible but requires some patience.

Currently we continue our research related to clustering in two major directions. First, we extend the approach to other types of spatio-temporal data, in particular, interactions between moving objects (mentioned in Section 3.2). In the future, we shall also extend it to spatially referenced time series data. Second, we look for ways to increase the scalability of clustering with respect to the size of the data. Thus, we have recently devised a visual analytics method for extracting clusters from a dataset not fitting in the computer main memory [1].

6. ACKNOWLEDGMENTS

The work has been done partly within the EU-funded research project GeoPKDD – Geographic Privacy-aware Knowledge Discovery and Delivery (IST-6FP-014915; http://www.geopkdd.eu) and partly within the research project ViAMoD – Visual Spatiotemporal Pattern Analysis of Movement and Event Data, which is funded by DFG – Deutsche Forschungsgemeinschaft (German Research Foundation) within the Priority Research Programme "Scalable Visual Analytics" (SPP 1335).

The work on interactive cluster analysis of trajectories was done together with our GeoPKDD partners from the University of Pisa, Italy. We are grateful to them for the cooperation and specially thank Salvatore Rinzivillo for the implementation of the clustering algorithm OPTICS in the way allowing the use of different distance functions.

7. REFERENCES

[1] Andrienko, G., Andrienko, N., Rinzivillo, S., Nanni, M., Pedreschi, D., Giannotti, F. 2009. Interactive Visual Clustering of Large Collections of Trajectories. *VAST 2009* (submitted).

[2] Andrienko, G., Andrienko, N., and Wrobel, S. 2007. Visual Analytics Tools for Analysis of Movement Data. *ACM SIGKDD Explorations*, 9(2): 38-46.

[3] Andrienko, N., and Andrienko, G. 2008. Evacuation Trace Mini Challenge Award: Tool Integration. Analysis of

Movements with Geospatial Visual Analytics Toolkit. *Proc. VAST 2008*, IEEE Computer Society Press, 205-206.

[4] Andrienko, N., Andrienko, G., and Gatalsky, P. 2003. Exploratory Spatio-Temporal Visualization: an Analytical Review. *Journal of Visual Languages and Computing*, 14 (6), 503-541

[5] Ankerst, M., Breunig, M., Kriegel, H.-P., and Sander, J. 1999. OPTICS: Ordering points to identify the clustering structure. In *Proc. ACM SIGMOD* 1999, 49–60.

[6] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. ACM KDD* 1996, 226-231.

[7] Gatalsky, P., Andrienko, N., and Andrienko, G. 2004. Interactive Analysis of Event Data using Space-Time Cube. In Banissi, E. et al. (Eds.) *Proc. IV 2004 - 8th International Conference on Information Visualization*, July 2004, London, UK, 145-152

[8] Grinstein, G, Plaisant, C., O'connell, T., Laskowski, S. Scholtz, J., Whiting, M. VAST 2008 Challenge: Introducing Mini-Challenges, Proceedings of IEEE Symposium on Visual Analytics Science and Technology (2008)

[9] Hägerstrand, T. 1970. What about people in regional science? In: *Papers of the Regional Science Association*, 24, 7-21.

[10] Kraak, M.-J. 2003. The space-time cube revisited from a geovisualization perspective, in: *Proc. 21st International Cartographic Conference*, Durban, South Africa, August 2003, 1988-1995.

[11] Rinzivillo, S., Pedreschi, D., Nanni, M., Giannotti, F., Andrienko, N., and Andrienko, G. 2008. Visually–driven analysis of movement data by progressive clustering, *Information Visualization*, 7(3/4), 2008, 225-239.

Surveying the complementary role of automatic data analysis and visualization in knowledge discovery

Enrico Bertini
Université de Fribourg
Bd de Pérolles 90
Fribourg, Switzerland
enrico.bertini@unifr.ch

Denis Lalanne
Université de Fribourg
Bd de Pérolles 90
Fribourg, Switzerland
denis.lalanne@unifr.ch

ABSTRACT
The aim of this work is to survey and reflect on the various ways to integrate visualization and data mining techniques toward a mixed-initiative knowledge discovery taking the best of human and machine capabilities. Following a bottom-up bibliographic research approach, the article categorizes the observed techniques in classes, highlighting current trends, gaps, and potential future directions for research. In particular it looks at strengths and weaknesses of information visualization and data mining, and for which purposes researchers in infovis use data mining techniques and reversely how researchers in data mining employ infovis techniques. The article further uses this information to analyze the discovery process by comparing the analysis steps from the perspective of information visualization and data mining. The comparison permits to bring to light new perspectives on how mining and visualization can best employ human and machine skills.

Categories and Subject Descriptors
H.5.2 [**User Interfaces**]: Graphical user interfaces (GUI). H.1.2 [**User/Machine Systems**]: Human information processing. H.2.8 [**Database applications**]: Data mining.

General Terms
Survey, Human Factors, Human-Machine Interaction.

Keywords
Visualization, Data Mining, Visual Data Mining, Knowledge Discovery, Visual Analytics.

1. INTRODUCTION
While information visualization (infovis) targets the visual representation of large-scale data collections to help people understand and analyze information, data mining, on the other hand, aims at extracting hidden patterns and models from data, automatically or semi-automatically.

VAKD'09, June 28, 2009, Paris, France.

In its most extreme representation, infovis can be seen as a human-centered approach to knowledge discovery, whereas data mining is generally purely machine-driven, using computational tools to extract automatically models or patterns out of data, to devise information and ultimately knowledge.

Interactive Machine Learning [1][2] is an area of research where the integration of human and machine capabilities is advocated, beyond scope of visual data analysis, as a way to build better computational models out of data. It suggests and promotes an approach where the user can interactively influence the decisions taken by learning algorithms and make refinements where needed.

Visual analytics is an outgrowth of infovis and focuses on analytical reasoning facilitated by interactive visual interfaces [3]. Often, it is presented as being the combination of infovis techniques with data mining capabilities to make it more powerful and interactive. According to Keim et al., visual analytics is more than just visualization and can rather be seen as an integrated approach combining visualization, human factors and data analysis [4].

At the time of writing, it is not clear how this human-machine integration should happen. In our view, visual analytics should enable the collaboration between the natural abilities of humans and the powerfulness of data mining tools, thus combining in a synergetic way natural and artificial intelligences.

Despite the growing interests on this integration, however, we still lack a detailed analysis of: 1) how currently the existing techniques integrate and to what extent; 2) what other kinds of integrations might be achieved.

The purpose of this work is start shedding some light on this issue. To this end we have performed a literature review of papers from premier conferences in data mining and information visualization, extracting those in which some form of integration exists. The analysis permitted to categorize the observed techniques in classes. For each class we provide a description of the main observed patterns followed by a discussion of potential extensions we deem feasible and important to realize. The analysis is then followed by a comparison of the analytical processes as they happen in data mining and in visualization. This comparison, together with the knowledge gained in the literature review, permits to clarify some commonalities and differences between the automatic and visual approaches. We believe this kind of reasoning can help framing the problem of automatic and

interactive analysis and better understand the role of human and machine.

The paper is organized as follows. Section 2 introduces some terminology to clarify the meaning of some word that often appear when talking about automatic or interactive data analysis. Section 3 introduces the literature review and its methodology. Section 4 illustrates the result of the review. It describes the observed patterns and the potential enhancements we suggest. Section 5 dissects commonalities and differences between the analysis processes in data mining and visualization. Finally, Section 6 discusses the limitations of this work, and thus provides ideas for its future extension, and Section 7 closes the paper with conclusions.

2. TERMINOLOGY

The common goal of information visualization and data mining domains is to extract knowledge from raw data. Before going further in our inspection of this process, we thought useful to agree on the definitions of basic concepts that are commonly used in this context such as data, information, knowledge, model, pattern and hypothesis:

- *Data* refer to a collection of facts usually collected by observations, measures or experiments. Data consist of numbers, words, or images. It is generally called abstract data in infovis, since it refers to data that has no inherent spatial structure enabling further mapping to any geometry.
- A *model* in science is a physical, mathematical, or logical representation of a system of entities, phenomena, or processes. Basically a model is a simplified abstract view of the complex reality. Models are meant to augment and support humans reasoning, and further can be simulated, visualized and manipulated.
- A *pattern* is made of recurring events or objects that repeat in a predictable manner. The most basic patterns are based on repetition and periodicity.
- A *hypothesis* consists either of a suggested explanation for an observable phenomenon or of a reasoned proposal predicting a possible causal correlation among multiple phenomena. The scientific method requires that one can test a scientific hypothesis. A hypothesis is never to be stated as a question, but always as a statement with an explanation following it.
- *Information*, in its earliest historical meaning, corresponds to the act of informing, or to the act of giving form or shape to the mind, according to the Oxford English Dictionary. Inform itself comes (via French) from the Latin verb "informare", to give form to, to form an idea of.
- *Knowledge* is the "justified true belief" according to Plato. According to the Oxford English Dictionary, knowledge can be defined as (i) expertise, and skills acquired by a person through experience or education; (ii) what is known in a particular field or in total; or (iii) awareness or familiarity gained by experience of a fact or situation.

In the context of *knowledge discovery*, we believe these concepts can be linked as follow: Data are the lowest level of abstraction; researchers often speak about *raw data* to emphasize this fact. From data, models and patterns can be extracted, either automatically using data mining techniques or by humans using their conceptual, perceptual or visual skills respectively. The use of human intuition to come up with observations about the data is generally called insight, i.e., the act or outcome of grasping the inward or hidden nature of things or of perceiving in an intuitive manner. Patterns and models are not necessarily linked, even though some authors consider them as synonyms. One way to distinguish these two concepts is the following: patterns are directly attached to data or a sub-set of data; whereas models are more conceptual and are extra information that cannot necessarily be observed visually in the data. Further, the observation of some patterns can result in a model and inversely, the simulation of a model can result in a pattern. Hypotheses are derived from models and patterns. A validated hypothesis becomes information that can be communicated. Finally, information reaches the solid state of knowledge when it is crystallized, i.e., it reaches the most compact description possible for a set of data relative to some task without removing information critical to its execution.

3. LITERATURE REVIEW

We started our analysis with a literature review in order to ground our reasoning on observed facts and limit the degree of subjectivity. We followed a mixed approach in which bottom-up and top-down analyses have been mixed to let the data speak for themselves and suggest new ideas or use the literature to investigate our assumptions or formulated hypotheses.

We included in the literature papers from major conferences in information visualization, data mining, knowledge discovery and visual analytics. In the current state of our analysis the papers have been selected from the *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, *IEEE International Conference on Data Mining (ICDM)* and the *IEEE Symposium on Information Visualization (InfoVis)*. We selected infovis candidate papers searching in the IEEE Explore library using keywords like: "data mining", "clustering", "classification", etc. Reversely, in data mining conferences we looked for the keywords like: "visualization", "interaction", etc. Manual skimming followed paper extraction. The final set of papers retained counts 55 items. Table 1 shows the distribution of the retained papers according to the paper source and the classification of papers presented below.

SOURCE	NUM. OF PAPERS	VIS	V++	M++	VM
KDD	23	3	7	9	4
ICDM	16	2	5	5	4
INFOVIS	16	1	9	5	0

Table 1 - Distrubution of the final list of retained papers according to source (conference) and paper type.

The whole list of reviewed papers with attached notes and categories can be found at the following address: http://diuf.unifr.ch/people/bertinie/ivdm-review.

4. PAPER CATEGORIES

We used various dimensions in order to classify the chosen papers: the knowledge discovery step it supports, whether it is interactive or not, the major mining and visualization techniques used, etc. In particular, in regards to the aim of this paper, we classified the paper according to four major categories indicating which approach drives the research:

- **Pure Visualization (VIS)** contains techniques based exclusively on visualization without any type of algorithmic support;
- **Computationally enhanced Visualization (V++)** contains techniques which are fundamentally visual but contain some form of automatic computation to support the visualization;
- **Visually enhanced Mining (M++)** contains techniques in which automatic data mining algorithms are the primary data analysis means and visualization provides support in understanding and validating the result;
- **Integrated Visualization and Mining (VM)** contains techniques in which visualization and mining are integrated in a way that it's not possible to distinguish a predominant role of any of the two in the process.

Since the focus of this paper is on how visualization and mining can cooperate in knowledge discovery, in the following we will not take into account the VIS category of pure visualization techniques.

4.1 Enhanced Visualization (V++)

This category pertains to techniques in which *visualization* is the primary data analysis means and automatic computation (that is the "++" in the name) provides additional features to make the tool more effective. In other words, when the "++" part is removed the technique becomes a "pure" visualization technique.

4.1.1 Observed enhancements with mining

As illustrated by black boxes on figure 1, the techniques collected in our literature review can be organized around three main patterns (Projection, Data Reduction, Pattern Disclosure) that represent different benefits brought by automatic computation to the information visualization process. Interestingly, as one can notice, the three patterns occur at the beginning of the knowledge discovery process:

- **Projection**. Automatic analysis methods often take place in the inner workings of visualization, by creating a mapping between data items and their graphical objects' position on the screen. The most traditional type of this method is Multidimensional Scaling (MDS), but in the literature it is possible to find many variations and alternatives. They all share the idea that the position assumed by a data point on the screen is not the result of a direct and fixed mapping rule between some data dimensions and screen coordinates but rather on a more complex computation that takes into account all data dimensions and cases. Ward refers to this kind of placement techniques in [5] as "Derived Data Placement Strategies" in his glyph placement taxonomy.
- **Data Reduction**. Data reduction is another area where computation can support visualization. Visualization has very well known scalability problems that limit the number of data cases or dimensions that can be shown at once. Automatic methods can reduce data complexity, with controlled information loss, and at the same time allow for a more efficient use of screen space. Pattern matching techniques can replace data overviews with visualizations of selected data cases that match a user-defined query. Sampling can reduce the number of data cases with controlled information loss. Feature selection can reduce the number of data dimensions by retaining subsets that carry the large majority of useful information contained in the data (and thus are most likely to show interesting patterns).
- **Pattern Disclosure**. In several visualization techniques the effectiveness with which useful patterns can be extracted depends on how the visualization is configured. Automatic methods can help configure the visualization in a way that useful patterns more easily emerge from the screen. Axes-reordering in parallel coordinates is one instance of such case [6]. Similarly, in visualizations where the degrees of freedom in visual configuration are limited, pattern detection algorithms can help make some visual patterns more prominent and thus readily visible. For instance, Vizster [7] organizes the nodes of a social network graph in automatically detected clusters enclosed within colored areas. Johansson et al. in [8] describe an enhanced version of Parallel Coordinates where clustering and a series of user-controlled transfer functions help the user reveal complex structures that would be hard, if not impossible, to capture otherwise.

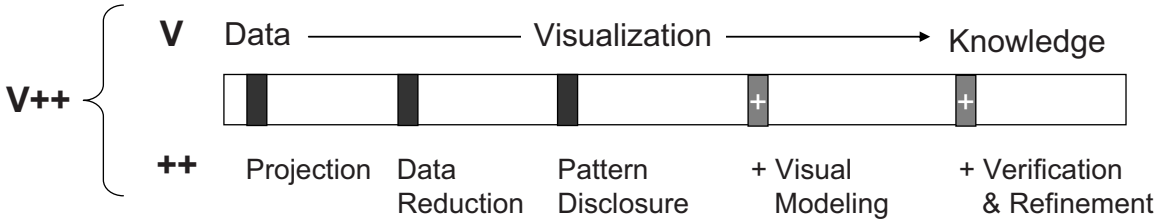


Figure 1 – Computationally enhanced Visualization (V++) benefit from mining techniques to improve information visualization standard process. Black boxes represent enhancements found in the literature survey; grey boxes (with "+") are extra benefits that could bring mining to visualization.

4.1.2 Other potential enhancements

All the automatic data analysis methods described above share the common goal of helping the user more easily extract information from the visualization. But, if we take into account the broader picture of data analysis and analytical reasoning, we see that automatic techniques could also be employed to go beyond simple pattern detection, and intervene at later stages of the knowledge discovery process, as illustrated in figure 1 (grey boxes with "+"). Here we list some of the function we deem important:

- **Visual Model Building.** One limitation of current visualization systems is their inability to go beyond simple pattern detection and frame the problem around a scheme. Ideally, the user should be able to find connections among the extracted patterns to build higher level hypotheses and complex models. This is another area where data mining has an advantage over visualization in that in the large majority of the existing methods a specific conceptual model is inherent in the technique. *Classification* and *regression* imply a functional model: any instantiation of the set of predictive variables returns a predicted target value. *Clustering* implies a grouping model, where data is aggregated in groups of items that share similar properties. *Rules* imply an inductive model where if-then associations are used. This kind of mental scaffold is absent in visualization, nonetheless there's no inherent reason why future systems might not be provided with visual modeling tools that permit, on the one hand to keep the level of flexibility of visualization tools, on the other hand to structure the visualization around a specific model building paradigm. Two rare examples of systems that go towards this direction are PaintingClass [9] and the Perception Bases Classification (PBC) system [10] in which classification can be carried out interactively by means of purely visual systems.
- **Verification and Refinement.** One notable feature of automatic data mining methods over data visualization is its ability to communicate not only patterns and models but also the level of trust a user can assign to the extracted knowledge. Similar functions are usually not present in standard visualization tools and surprisingly little research as been carried out towards this direction so far. Automatic algorithms could be run on extracted patterns to help the user assess their quality once they are detected. To date, the only systems we are aware of where a similar idea has been implemented are [11][12], where respectively data abstraction quality is measured and progressive automatic refinement of visual clusters is performed.

  Another related area of investigation is the use of the traditional split in *training data* and *test data* used in supervised learning as a novel paradigm to use in data visualization. There is no reason in principle not to use the same technique in information visualization to allow for verification of extracted patterns. Some few studies on sampling for data visualization slightly touch this issue [13][14] but none of them focuses on the use of sampling or data segmentation for verification purposes.

Worthy of special remark is also the almost complete absence of predictive modeling in visualization, as highlighted by Amar and Stasko in their analysis of "analytic gaps" in information visualization [15]. While it is fairly simple to isolate data segments and spot correlations, even in multidimensional spaces, current information visualization tools lack the right affordances and interactive tools to structure a problem around prediction. Questions like: "which data dimensions have the highest predictive power?", "what combination of data values are needed to obtain a target result?" are not commonly in the scope of traditional visualization tools.

4.2 Enhanced Mining (M++)

This category pertains to techniques in which *data mining* is the primary data analysis means and visualization (that is the "++" in the name) provides an advanced interactive interface to present the results. In other words, when the "++" part is removed it becomes a "pure" data mining technique.

4.2.1 Observed enhancements with visualization

As illustrated by black boxes on figure 2, the techniques collected in our literature review can be organized around two major patterns (Model presentation and pattern exploration & filtering) that represent different benefits brought by visualization to data mining. Interestingly, reversely to the previous category (V++), the two patterns occur at the end of the knowledge discovery process:

- **Model Presentation.** Visualization is used to facilitate the interpretation of the model extracted by the mining technique. According to the method used, the ease with which the model is interpreted can vary. Some models naturally lend themselves to visual abstraction (e.g., dendrogram in hierarchical clustering) whereas some others require more sophisticated designs (e.g., neural networks or

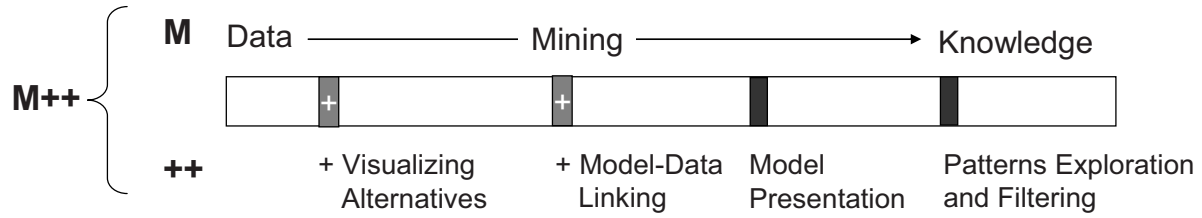


Figure 2 – Visually enhanced Mining (M++): benefits of visualization over data mining standard process. Black boxes represent potential enhancements found in the literature; grey boxes (with "+") are extra benefits that could bring visualization to mining.

support vector machines). Beyond interpretation, visualization also works as a way to visually convey the level of trust a user can assign to the model or parts of it. Interactions associated to the visualization permits to "play" with the model allowing for deeper understanding of the model and its underlying data.

- **Patterns Exploration and Filtering.** Some mining methods generate complex and numerous patterns which are difficult to summarize in a compact representation; notably association rules. In this case visualization often adopts techniques similar to plain data visualization and the patterns are managed like raw data. Visualization here helps gaining and overview of the distribution of these patterns and to make sense of their nature. Interactive filtering and direct manipulation tools have a prominent role in that finding the interesting pattern out of numerous uninteresting is the key goal.

4.2.2 Other potential enhancements

Visualization applied to data mining output, as shown above, provides great benefits in terms of model interpretation and trust-building. We believe that visualization, however, can provide additional benefits that have not been fully addressed so far, and enable users to intervene in early stages of the knowledge discovery process, as illustrated in figure 2 (grey boxes with "+"):

- **Visualizing Alternatives.** One of the characteristic features of data mining is the capability of generating different results and models by manipulating a limited set of parameters. This is common to all methods and can be seen as both an advantage and a limitation. It is an advantage in that the necessary flexibility is given to create alternatives and adapt to different analytic goals. But, it is also a big limitation in that setting the parameters of a mining algorithm is often perceived by the user as an "esoteric" activity in which the relation between actions and results is blurred. Even more problematic, when alternative models are constructed it is extremely complicated to compare them in the space of a single user interface. Visualization in our opinion has the power to bridge this gap by: 1) providing means to more directly represent the connection between the parameters and the results; 2) allow for visualization structures that permit the comparison of alternative results. This last point is particularly interesting in that visualization has the power to provide the right tools to compare alternative visual abstractions, as demonstrated for instance by the success of the systems presented at the InfoVis 2003 contest on Pair Wise Comparison of Trees [16]. One system in our literature review partially supports this kind of comparison by generating different alternative results of a subspace clustering algorithm [17]. The user can see the results obtained through the variation of various parameters and choose the most interesting one among the set of available results.
- **Model-Data Linking.** The models that mining algorithms create out of data are higher level data abstractions that permits to summarize complex relations out of large data. If from the one hand these abstractions facilitate data analysis and reduce the complexity of the original problem space, from the other hand the abstraction process often makes it difficult to interpret the observed relations in terms of the original data space. Most systems in our literature survey provide model representation, but very rarely they permit to drill down to the data level to link an observed relation to its underlying data. In some cases such a lack of connection between model and data can create relevant limitations in model understanding and trust building and visualization seems to be the right tool to bridge this gap. One notable example is data clustering. Besides the large provision of visual and interactive techniques to represent clustering results it is very rare to find systems where the linkage between extracted clusters and data instances is made explicit by the visualization. And this is somewhat surprising in that the goal of data clustering is not only to partition data in a set of homogeneous groups but also, and potentially more important, to characterize them in a way that their content can be described in terms of few data dimensions and values. A better connection between model and raw data is then useful also to spot relevant outliers, which can often triggers new analyses and lines of thought. Without such a capability the analyst is forced to base his reasoning only on abstractions, thus limiting the opportunities for serendipitous discoveries and trust building.

4.3 Integrated Visualization & Mining (VM)

This category combines visualization and mining approaches. None of them predominate the other and ideally they are combined in a synergic way. In the literature we found two kinds of integration strategies that we describe below. Following their description we speculate on a mixed-initiative approach to the KDD process.

4.3.1 Integration strategies

There are two extreme approached to integrate mining and visualization, as described below:

- **White-Box Integration.** In this kind of integration the human and the machine cooperate *during* the model building process in a way that intermediary steps in the algorithm can be visualized and decisions can be taken by the user on how to direct the model building process. This kind of systems is quite rare. There are examples of cooperative construction of classification trees, like the one presented in [18], where the user steers the construction process and at any stage can ask the computer to make one step in his or her place like splitting a node or expanding a sub-tree. This kind of systems shows the highest degree of collaboration between the user and the machine and goes beyond the creation accurate models. They help building trust and understanding, because the whole process is visible, and also they permit to directly exploit the user's domain knowledge in the model construction process.
- **Black-Box Integration (feedback loop).** Integration between mining and visualization can also happen indirectly using the algorithm as a black box, but giving the user the possibility to "play" with parameters setting in a tight visual loop environment where changes in the parameters are automatically reflected in the visualization. In this way the connection between parameters and model, even if not explicit, could be intuitively understood. Alternatively, the same integration can be obtained in a sort of "relevance

feedback" fashion, where the system generates a set of alternative solutions and the user instructs the system on which are the most interesting ones and gives hints on how to generate a new set.

4.3.2 A mixed-initiative KDD process

Having analyzed a wide spectrum of integrations between automatic and interactive methods, we believe that one of the most interesting and promising direction for future research is to achieve a full mixed-initiative KDD process where the human and the machine can cooperate on the same level.

Humans and machines are complementary, and visualization and data mining should make use of the specificities of each. Humans are intuitive and have good skills at interpretation according to the context and their domain knowledge. They are good at getting the "big picture" and at performing high level reasoning towards knowledge. Machine on the other side are fast and reliable at computing data, and they do not make errors.

In the early 90's already, Colgan & Spence et al. had already the vision to use visualization to enhance human-machine collaboration in electronic circuit design through the cockpit of their Coco system. Their approach highlighted the need for an effective interface to blend the complementary capabilities of the human designer and computer algorithms [22, 23]. More recently, Pu & Lalanne proposed a mixed-initiative system to support problem solving via algorithm visualization and visual trade-off analysis [20, 21]. Through visual interaction, the Comind system enables designers to select and control the solving algorithm they want to use, i.e. they can visualize it while it is processing the data, stop it at anytime and modify the problem definition or select another mining or solving algorithm. Finally they can select the visualization techniques they want to view the results, while still being able to tune parameters. In the context of sequential pattern detection for text mining, [19] proposes to combine computational and statistical efforts through data mining with the human participation through visualization for the ultimate goal of knowledge discovery. In their application, visualization helps humans quickly obtain an overall structural view of patterns and complementary, data mining provides accurate support information for all patterns.

Table 2 summarizes the major complementary strengths of human and machine in the knowledge discovery process, derived from our literature review.

Human	Machine
Select strategies	Project & Reduce data
Observe, derive knowledge	Select optimal solution, best configuration
Interpretation, explanation	Build models
Measure interestingness	Extract patterns, models
Generating hypothesis	Verification

Table 2 – Complementary strengths of human and machine in the knowledge discovery process.

Figure 3 is the result of the benefits brought by visualization and mining independently to the knowledge discovery process as described in section 4.1 and 4.2 respectively. It is inline with the complementary strengths brought by humans through visualization and by machines through data mining. For example, while humans are good at choosing modeling strategies through visualization, the machine is good at computing large amount of data for projecting and reducing data. Further, while machines can disclose and highlight all the patterns found automatically over the data, human can explore them and keep only the most interesting ones, according to their knowledge of the data set and its associated domain. Later on, human and machine can collaborate to build models, either coming from mining models or alternatively derived by humans through their perceptive and cognitive systems. At this stage visualization techniques can be particularly useful to bridge the gap between data and the extracted models. Finally, data mining techniques can be useful to support the validation of observed model or knowledge that humans can ultimately refine through interaction.

To date, the only system that comes closer to the idea of a mixed-initiative KDD process is the one we mentioned above in White-Box Integration [18], where a decision tree can be constructed by alternating steps of human-based decisions and machine-based algorithmic steps.

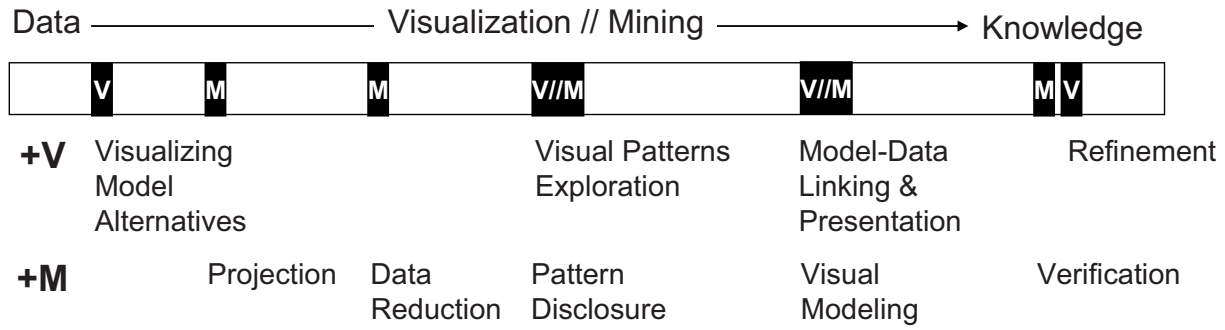


Figure 3 – Integrated visualization and Mining (VM): towards a white box for the full KDD process with benefits coming from visualization (+V) and benefits from mining (+M).

5. ANALYZING THE ANALYSIS PROCESS

Both visualization and data mining are alternative methods to transform data into knowledge. Having said that, a legitimate question remains: are they just different recipe that work in the same manner or do they differ in any substantial manner? We believe that posing this question is becoming of increasing importance as we attempt to get the most out of the two and create successful integrations like the one advocated in Visual Analytics.

Here we provide reflections on this subject, based on an initial schematization of the analysis process in data mining and visualization, highlighting notable differences and commonalities between them.

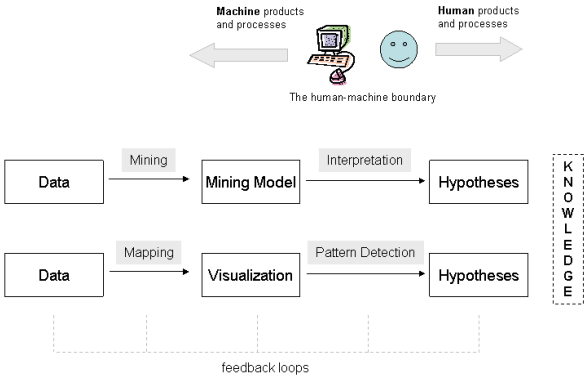


Figure 4 - Comparison between mining and visualization analytics processes.

5.1 Processes versus products

Looking at Figure 4 we can see that both in visualization and mining we have products (boxes) and processes (arrows). What is interesting to note, at least from the terminological point of view, is that *visualization* and *data mining* are not on the same level. More precisely, the word "visualization" is often intended as the *product* of the visual mapping between data and a visual representation; the word "mining", on the other hand, commonly refers to the *process* that transforms data into a data mining model. This distinction is important because in Visual Data Mining and Visual Analytics often mining and visualization are considered as alternatives. Even more important is to acknowledge the fact that in data mining there are necessarily always some tasks performed by the human and, likewise, in information visualization there are always some tasks performed by the machine. The machine, in particular, is responsible of the *mining process*, in data mining, and of the *visual mapping process*, in visualization. Moreover, the mining process produces a *mining model*, whereas the visual mapping process produces a *visualization*.

If we adopt this perspective it is easy to see for instance how in visual mapping and mining process similar human tasks are involved, like the definition of an appropriate schema (visual or functional) that fits the user's mental model and goal. Similarly, we realize that in terms of perceptive and cognitive processes it is the comparison of the activities that go from visualization to hypothesis generation, in visualization, and from mining model to hypothesis generation, in mining that matters. We believe that a deeper analysis and comparison of what happens at this stage, where the human interfaces with the machine, might lead to relevant advancements in Visual Analytics.

5.2 Mental models and problem instantiation

Again, comparing the two processes in Figure 4, it is interesting to note a key difference between them. In visualization the formation of a mental model and its formalization happen "in sequence" when the mapping has already been performed and the data is already visualized. In other terms, the visualization by itself is a vehicle to aid the formation of a mental schema. In data mining instead the human has to first *mentally* formulate a mental schema in a way that it can fit with one of the existing input-output mappings provided by data mining.

A clarifying example comes from the comparison of how knowledge building happens on Parallel Coordinates visualization or a Decision Tree algorithm. In the first case, the user most probably approaches the problem with a limited formalization of the problem space and an opportunistic approach. Usually he or she just wants to look at the data and see what's there. Moreover, the kind of extracted patterns can cover a quite broad range of data models, e.g., correlations among two or more dimensions, groupings (clusters), outliers, etc. In the case of decision trees, the user has to first formulate the problem in terms of a definite mental schema that matches the particular input-output mapping enforces by the technique. Specifically, the data will be transformed in a series of IF-THEN rules that segment the input space in groups characterized by their relations. For any additional data record, once the model is built, the model will provide a specific output (label). It is worth to note in this example that some of the conclusions to which the user might end up in one case might easily overlap with those extracted from the other. The question of how these processes compare, when and how it is more preferable to use one or another, or if a synergy between the two can be found is in our opinion one of the central issues to study in Visual Analytics.

5.3 The Human-Machine Interface

Another important aspect illustrated in figure 4 is that in both processes there is a stage in which necessarily the human has to acquire some information from the machine, that is, what we called the *human boundary*.

In traditional data mining, systems are not without an interface, they just provide simple and minimalistic interfaces like results organized in tabular data. Visualization systems on the other hand provide visually rich and highly interactive tools for data exploration.

More importantly, in data visualization the interface has the primary goal to let the user *detect* and correctly extract relevant patterns from the screen. In data mining the interface has the primary goal to let the user *understand* the model produced by the machine and its relation to data. From the visualization design point of view it is important to recognize this difference and acknowledge that not necessarily what we have learned from data visualization is enough to build effective model visualizations.

Model visualization seems to be a more complex task, where we are confronted with novel design challenges like: finding effective metaphors to represent the model, finding ways to represent the

model in relation to the data and vice versa, and finding convenient interaction methods to manipulate the model. Further research is still needed to advance towards this direction.

5.3.1 The feedback loop

So far we have only discussed one direction of the human-machine interface, that is, from the machine to the human. The opposite direction is often neglected but it is equally important because it permits to close the feedback loop. It is in fact the possibility to iterate over alternate phases of human perception and understanding of the current state and human actions to change this state and devise alternatives that fuel the discovery and learning process.

On a higher level this is also supported by the Sensemaking Theory that describes how people make sense of information. As Pirolli and Card note in [19], the process revolves around "one set of activities that cycle around finding information and another that cycles around making sense of the information".

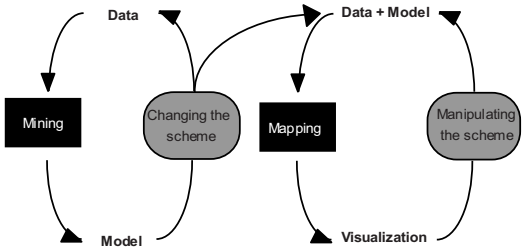


Figure 5 – The feedback loop in Knowledge Discovery. The grey boxes represents the two major stages at which humans can intervene.

5.3.2 User intervention levels

In our literature review, almost half of the papers do not propose means for users to interact with the system and as such intervene on the knowledge discovery process. In the 55 papers reviewed, the major interaction techniques found can be grouped in two major categories depending on the knowledge discovery step at which users can intervene, i.e. pre or post model interventions, to change the scheme or manipulate it respectively as illustrated on figure 5:

- **Changing the scheme.** Both in visualization and in data mining at any stage the user can decide to change the schema. In visualization changing the schema means changing the visual mapping in a way that data can be seen under a new perspective. In data mining it means reframing the problem so that it is represented under a new model, as an example, moving the analysis from the generation of rules to finding data clusters. This kind of activities is often neglected and yet it is very important because as the user's mental model changes the tools must adapt in a way to reflect this change. The goodness of a data analysis system should be measured also in terms of this flexibility. This need of reframing problems under different schemes uncover a relevant gap in current tools; especially those found in information visualization. One of the biggest challenges is yet to find an appropriate visualization for the task at hand. Despite numerous efforts towards this direction, especially at the early stage of information visualization (e.g., in Jock MacKinlay's work [20]), current tools offer very limited support. Automatic or semi-automatic methods should be employed to help users find appropriate visual mappings or yet suggest possible alternatives.
- **Manipulating and tuning the scheme.** Another option the user has to create alternatives is to change parameters within the context of a given scheme. In visualization this comprises interactions like: dynamic filtering, axes reordering, zoom & pan, etc. In data mining it involves some form of parameter tuning, as when using different distance functions or number of desired groups in data clustering. This last function is of special interest in that visualization can be a powerful means to help users tune up their mining models. As we have already discussed in Section 4.2.2 in "Visualizing Alternatives", the use of powerful visualization and interaction schemes could greatly improve the state of current tools. Of special interest is the study of efficient techniques that permit to understand how a model changes when one or more parameters change. In current tools it is almost impossible to achieve this level of interaction. Not only the large majority of parameters are difficult to interpret but also the user is forced to go through a series of "blind" trial-and-error steps where the user changes some parameters, waits for the construction of the new model, evaluates the result and iterates over until he or she is satisfied.

6. LIMITATIONS AND FUTURE WORK

Despite our effort to produce a meaningful literature survey and to extract useful indication out of it, we believe it is important to highlight and acknowledge some limitations of this work.

The literature we have analyzed, though useful, is far from being complete. We decided to use a number of papers that could be analyzed in a relative short time (by the two authors) and at the same time capture most of the relevant trends.

As a consequence we decided not to draw any statistics out of our study. The literature contains some hand-made categorizations that could have been used to further categorize the techniques and depict some general trends out of it. We postpone this task to a later version of our work, where the number and kind of collected papers will provide us with a more solid base on which to draw relevant statistics.

Finally, it's important to take into account that a large part of this paper is the product of subjective indications stemming from what we believed worth to extract from the literature. Nonetheless, we believe that our analysis and guidelines can highlight hidden patterns and stimulate further research on important issues in this cross-disciplinary topic.

We plan to advance this work after having received sufficient feedback from the community. Specifically, we want to extend the literature, further categorize the techniques, and draw some general statistics on research trends that could help suggesting additional future research directions.

7. CONCLUSIONS

We have presented a literature review on the role of visualization and data mining in the knowledge discovery process. From the review we have generated a series of classes through which we

have categorized the collected papers: the knowledge discovery step it supports, whether it is interactive or not, the major mining and visualization techniques used, etc. In particular, in regards to the aim of this paper, we classified the paper according to three major categories indicating which approach drives the knowledge discovery: computationally enhanced visualization systems, visually enhanced data mining systems, and integrated visual and mining systems.

This categorization highlights some observed patterns and suggests potential extensions which are not present in the considered literature. For instance, in order to enhance the standard visualization process, we believe data mining techniques could support visual model building to go beyond simple pattern detection. Further, mining techniques could be also used to verify and assess the quality of patterns detected by users. Reversely, visualization could enhance the data mining process to visualize modeling alternatives, and to understand modeling results through a better model-data linking and presentation.

In addition to these suggestions, the article provides a series of higher level reflections on the analysis process as it happens in visualization and data mining. These reflections suggest new perspective on the role of visualization and mining in the data analysis process and potential areas of investigation towards a better integration of both techniques. In particular, this preliminary study suggests improving the human machine interaction through a better consideration of the feedback loop so that users can intervene at different levels of the knowledge discovery process, to change and manipulate the scheme respectively.

8. REFERENCES

[1] J.A. Fails and J. Olsen, "Interactive machine learning," *IUI '03: Proceedings of the 8th international conference on Intelligent user interfaces*, New York, NY, USA: ACM, 2003, pp. 39–45.

[2] M. Ware, E. Frank, G. Holmes, M. Hall, and I.H. Witten, "Interactive machine learning: letting users build classifiers," *International Journal of Human Computer Studies*, vol. 55, 2001, pp. 281–292.

[3] J.J. Thomas and K.A. Cook, *Illuminating the path: The research and development agenda for visual analytics*, IEEE, 2005.

[4] D.A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler, "Visual analytics: Scope and challenges," *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics*, Springer, 2008, pp. 76–90.

[5] M.O. Ward, "A taxonomy of glyph placement strategies for multidimensional data visualization," *Information Visualization*, vol. 1, 2002, pp. 194–210.

[6] W. Peng, M.O. Ward, and E.A. Rundensteiner, "Clutter reduction in multi-dimensional data visualization using dimension reordering," *IEEE Symposium on Information Visualization, 2004. INFOVIS 2004*, pp. 89–96.

[7] J. Heer and D. Boyd, "Vizster: Visualizing online social networks," *Proceedings of the 2005 IEEE Symposium on Information Visualization*, 2005, pp. 33–40.

[8] J. Johansson, P. Ljung, M. Jern, and M. Cooper, "Revealing Structure within Clustered Parallel Coordinates Displays," *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, IEEE Computer Society, 2005, p. 17.

[9] S.T. Teoh and K. Ma, "PaintingClass: interactive construction, visualization and exploration of decision trees," *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, Washington, D.C.: ACM, 2003, pp. 667-672.

[10] M. Ankerst, C. Elsen, M. Ester, and H. Kriegel, "Visual classification: an interactive approach to decision tree construction," *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 1999, pp. 392-396.

[11] Q. Cui and J. Yang, "Measuring Data Abstraction Quality in Multiresolution Visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, 2006, pp. 709-716.

[12] D. Yang, Z. Xie, E.A. Rundensteiner, and M.O. Ward, "Managing discoveries in the visual analytics process," *SIGKDD Explor. Newsl.*, vol. 9, 2007, pp. 22-29.

[13] G. Ellis and A. Dix, "Density control through random sampling: an architectural perspective," *Information Visualisation, IV 2002.*, 2002, pp. 82–90.

[14] E. Bertini and G. Santucci, "Give chance a chance: modeling density to enhance scatter plot quality through random data sampling," *Information Visualization*, vol. 5, 2006, pp. 95–110.

[15] R.A. Amar, "Knowledge Precepts for Design and Evaluation of Information Visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, 2005, pp. 432-442.

[16] C. Plaisant, J. Fekete, and G. Grinstein, "Promoting Insight-Based Evaluation of Visualizations: From Contest to Benchmark Repository," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 14, 2008, pp. 120-134.

[17] E. Müller, I. Assent, R. Krieger, T. Jansen, and T. Seidl, "Morpheus: interactive exploration of subspace clustering," *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2008, pp. 1089-1092.

[18] M. Ankerst, M. Ester, and H. Kriegel, "Towards an effective cooperation of the user and the computer for classification," *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2000, pp. 179-188.

[19] P. Pirolli and S. Card, "The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis," *Proceedings of International Conference on Intelligence Analysis*, 2005.

[20] J. Mackinlay, "Automating the design of graphical presentations of relational information," *ACM Transactions on Graphics*, vol. 5, 1986.

Visual Exploration of Categorical and Mixed Data Sets

Sara Johansson
Visual Information Technology and Applications
Department of Science and Engineering
Linköping University, Sweden
sara.johansson@itn.liu.se

ABSTRACT

For categorical data there does not exist any similarity measure which is as straight forward and general as the numerical distance between numerical items. Due to this it is often difficult to analyse data sets including categorical variables or a combination of categorical and numerical variables (mixed data sets). Quantification of categorical variables enables analysis using commonly used visual representations and analysis techniques for numerical data. This paper presents a tool for exploratory analysis of categorical and mixed data, which uses a quantification process introduced in [16]. The application enables analysis of mixed data sets by providing an environment for exploratory analysis using common visual representations in multiple coordinated views and algorithmic analysis that facilitates detection of potentially interesting patterns within combinations of categorical and numerical variables. The effectiveness of the quantification process and of the features of the application is demonstrated through a case scenario.

Categories and Subject Descriptors

I.3.6 [**Computer Graphics**]: Methodology and Techniques—*interaction techniques*; I.5 [**Pattern Recognition**]: Miscellaneous

1. INTRODUCTION

In many research and application areas data sets including categorical variables or a combination of categorical and numerical variables (mixed data sets) are nothing unusual. Although several similarity measures exist that can be used for categorical data, such as the Jaccard coefficient [26], overlap and Goodall similarity [4], these are usually not as straight forward and general as similarities within numerical variables. Due to this, categorical data is often more difficult to visualize and analyse. Moreover, many commonly used visualization techniques, such as parallel coordinates [14, 28] and scatter plots, have been developed for visualization of numerical data and are hence based on the numerical similarity or distance between data items. Several visualization techniques developed for categorical data exist, but their effectiveness is often highly dependent on the structure of the data and on the analysis task. Moreover, they are often unable to represent mixed data sets including a combination of categorical and numerical variables.

One approach to overcome the difficulties involved in visualization of categorical and mixed data sets is to quantify the categorical data, representing the categories with numerical values, which enables analysis using the more general methods developed for numerical data. To avoid misleading the analyst into drawing incorrect conclusions when employing this approach, it is crucial to find a quantification that preserves the relationships within the data set.

In [16] an interactive quantification process was presented, which utilises the efficiency of algorithmic data analysis as well as making use of the knowledge of domain experts. This process identifies numerical representations that preserve relationships within the data and enables modification based on the analysis task and user knowledge.

This paper presents MiDAVisT (Mixed Data Analysis Visualization Tool), which is an application that has evolved from the process and interactive environment presented in [16]. MiDAVisT is an interactive tool for analysis of categorical and mixed data sets that provides a combined algorithmic and user controlled quantification of categorical variables, enabling analysis using both algorithmic methods and visual representations developed for purely numerical data sets. MiDAVisT also provides an interactive environment for visual and exploratory analysis where commonly used visual representations for numerical data are provided and combined with common algorithmic analysis methods to facilitate detection of patterns and relationships between categorical and numerical variables. The effectiveness and usefulness of the application is demonstrated through a case scenario where relationships within and between categorical and numerical variables in a mixed data set are identified using the features provided in MiDAVisT. The main contributions of this paper can be summarised as:

- An interactive application for user controlled quantification and analysis of data sets including a combination of categorical and numerical variables, enabling analysis based on relationships within all variables of the data set.

VAKD'09, June 28, 2009, Paris, France.

- An exploratory environment including multiple coordinated views, where visual representations and algorithmic analysis methods developed for numerical data are provided for exploration and pattern detection in mixed data sets.

The paper is organised as follows. Section 2 presents related research. In section 3 the quantification process is described and in section 4 the visual exploration environment of MiDAVisT is presented. Section 5 contains a case scenario that demonstrates the features and effectiveness of MiDAVisT. This is followed by conclusions and future work in section 6.

2. RELATED WORK

Several visualization techniques exist that are designed specifically for visualization of categorical data. Some examples being fourfold displays [9], where the cell frequencies of two-by-two tables are represented by quarter circles, mosaic displays and mosaic matrices [7, 8, 9], which represent multi-way tables with tiles whose sizes are proportional to the cell frequencies. Parallel sets [18] is a visual representation with a layout similar to parallel coordinates [14, 28] where the categories of a categorical variable are represented with a set of boxes whose sizes are proportional to the category frequency. In parallel sets the numerical variables of mixed data sets are represented by separating numerical values into bins. Further one example where the layout of parallel coordinates is used for categorical data visualization is presented in [13], where parallel coordinates are extended to avoid data overlay, meaning data items being concealed by other data items. This is achieved by spreading the lines over additional axes and by sorting the lines according to what categories they belong to in the adjacent axes.

These techniques all attend to visualization of categorical and mixed data, and are hence related to the approach presented in this paper. However they all suggest single visual representations, whereas the application presented in this paper focuses on quantification of categorical data and analysis using common methods and visual representations for numerical data, hence providing a more general and diverse environment for visual analysis.

A range of similarity measures exist for measuring the similarity between individual categorical data items. The most simple one being the overlap similarity [4] which assigns a similarity value of 1 if two items match for a variable and 0 if they do not match. Although straight forward, the overlap similarity has a major drawback in that all matches and all mismatches are treated as equal. The Jaccard coefficient [26] is a similarity measure for binary data which is also based on category matching, but only considers matching of ones for binary items, whereas matching of zeros is ignored, this makes the Jaccard coefficient a suitable similarity measure for sparse data. None of these similarity measures are, however, suitable for the application presented in this paper unless modifications are made, since categorical variables are, in general, not binary and since both measures only consider whether items match or not and do not take any other properties of similarity into consideration.

Another approach to similarity in categorical data is to use data-driven similarity measures, such as the Goodall, Occurrence Frequency, and Smirnov similarity measures [4]. For these measures the frequency distribution of variables is taken into account when measuring similarity and, as a result, the behaviour of the measures is directly dependent on the structures in the data set, and is hence not as general as numerical similarity.

Several approaches to quantification of categorical data have been previously presented. In [19] a technique for ordering of categorical data is introduced, where clusters of categories are formed based on domain semantics and the categories are ordered in a way that minimises the distances within the clusters. In [24] categorical data is quantified based on the association of categories in a categorical space. The quantification is achieved using Correspondence Analysis (CA) [12], as described in detail in section 3.1. In [21] this technique is incorporated into a framework for mapping of diverse data types.

CA has been used in different ways in visualization. In [7, 8, 9] it is used to reorder the categories in mosaic displays, and [12] presents a number of ways to visualize the result of CA using scatter plot techniques, such as CA Maps where CA is used to position the categories in a plot, and CA Bi-plots where each row and column of a table is displayed as a point. In [12] CA is also suggested as a technique for quantification of categorical data in order to apply statistical techniques that require numerical data.

The quantification approach of MiDAVisT is based on the quantification process presented in [16]. This process is similar to the approach presented in [24], but extends it by incorporating the relationships of numerical variables into the quantification process and by utilising the domain knowledge of expert users, as described in detail in section 3.

In addition to this MiDAVisT also provides an interactive environment for visual exploration by combining algorithmic analysis and multiple coordinated views. Using multiple coordinated views is a well established concept [2, 5, 23] that has been successfully used to overcome the difficulty of presenting large amounts of data in one screen while simultaneously making it possible to find detailed structures. A number of tool-kits and applications exist that can be used for visual exploration using multiple coordinated views combined with algorithm analysis, some examples are XmdvTool [27], GAV [15], InfoVis Toolkit [6] and the Hierarchical Clustering Explorer [25]. MiDAVisT has been implemented using the GAV framework.

3. INTERACTIVE QUANTIFICATION

This section briefly describes an interactive process for quantification based on algorithmic analysis and knowledge of domain experts, which was introduced in [16]. MiDAVisT employs this approach for quantification of categorical data, as well as for identification of similarities and relationships between categories. The data set used to demonstrate the process is an automobile data set containing 205 data items and including 6 categorical and 8 numerical variables [1].

When performing quantification of categorical data it is of high importance to find numerical representations that preserve the existing relationships within the data set. The quantification process described in this section uses CA to identify similarities between categories, however any algorithm able to identify similarities between categories could be used.

3.1 Correspondence Analysis

CA is a method for analysis of frequency tables, where each cell represent the frequency of a combination of categories [11, 12, 24]. An example frequency table for two categorical variables (eye colour and hair colour) is shown in table 1. For larger numbers of categorical variables, tables that contain all possible category combinations within the data set are used.

CA identifies similarities between the cells of the frequency table, and can be seen as a special case of Principal Components Analysis [17]. An example of similarity between categories can be seen in table 1 where the *brown* and *hazel* rows follow a similar pattern with highest frequency for *brown* hair and lowest frequency for *blond* hair. Hence *brown* and *hazel* can be considered as similar. The row representing *blue* eyes is less similar to *brown* and *hazel*, with high frequencies for both *brown* and *blond* hair.

Initially CA computes a correspondence matrix, $\mathbf{P} = \frac{\mathbf{N}}{\mathbf{n}}$ where $\mathbf{N}$ is the frequency table and n is the grand total of the table. $\mathbf{P}$ is normalised and centred (equation 1) using $\vec{r}$ and $\vec{c}$, which are vectors containing the row and column sums respectively, and $\mathbf{D_r}$ and $\mathbf{D_c}$, which are diagonal matrices with $\vec{r}$ and $\vec{c}$ as diagonals.

$$\mathbf{S} = \mathbf{D_r}^{-1/2}(\mathbf{P} - \vec{r}\vec{c}^{T})\mathbf{D_c}^{-1/2} \quad (1)$$

CA identifies independent dimensions in the frequency table by applying Singular Value Decomposition (SVD), $\mathbf{S} = \mathbf{U\Sigma V^T}$ where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices and $\mathbf{\Sigma}$ is a diagonal matrix where the diagonal values are singular values of $\mathbf{S}$ [10]. The first independent dimension explains most of the variance within the table, and the variance explained decreases with every succeeding dimension. Principal axes, $\mathbf{F}$, of the table rows are extracted as in equation 2.

$$\mathbf{F} = \mathbf{D_r}^{-1/2}\mathbf{U\Sigma} \quad (2)$$

Based on the theory of optimal scaling [12] the first principal axis of $\mathbf{F}$, which is related to the first independent dimension in SVD, can be used as numeric representations of the row categories in the frequency table. In this way a quantification is performed based on the relationships of all categories within the data set. Table 2 displays the first principal axis when CA has been applied to table 1. As can be seen the *brown* and *hazel* categories are represented by values close to each other, suggesting that they are similar, whereas *blue* is represented by a value further away, indicating that it is less similar to *brown* and *hazel*.

Table 1: A frequency table for two variables, eye colour (rows) and hair colour (columns). Each cell in the table represents the frequency of a combination of categories.

	Black	*Brown*	*Red*	*Blond*
Brown	11	20	4	1
Blue	3	14	3	16
Hazel	3	9	3	2
Green	1	5	2	3

Table 2: The first principal axis when CA has been performed on table 1. The values of this axis are used as numeric representations of the row categories.

	First principal axis
Brown	-0.5103
Blue	0.5516
Hazel	-0.2098
Green	0.1891

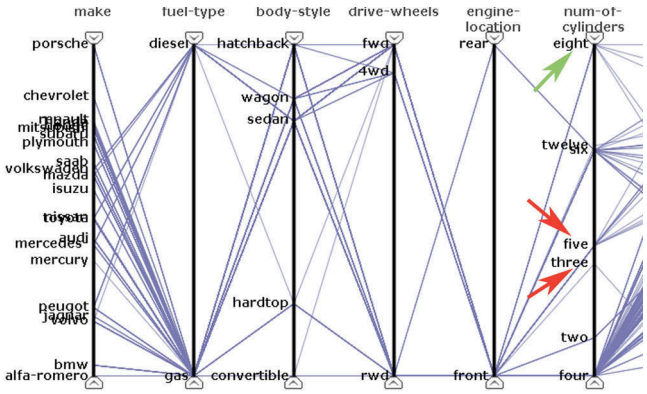


Figure 1: The suggested quantification of the categorical variables subsequent to correspondence analysis, displayed using parallel coordinates. The categorical variables are positioned to the left in the display and each category is represented by its name. The red arrows point out the categories *five* and *three* in the *number-of-cylinders* variable, which are considered similar to each other. A category that is considered less similar to them (*eight*) is pointed out by a green arrow.

3.2 Categorisation of Numerical Data

CA is performed on frequency tables where each cell represents the frequency of a combination of two categories, hence enabling a quantification based on the relationships within the categorical variables. If performing CA on a data set containing both categorical and numerical variables, the numerical variables need to be incorporated into the frequency table. This categorisation must be done in a way that preserves the existing numerical relationships. By this a quantification is performed that is based on relationships within both categorical and numerical variables.

The quantification process used in MiDAVisT provides two methods for categorisation of numerical data. One being a manual categorisation, where the user is allowed to interactively divide the data items into a number of categories guided by a visual display, and the other being an algorithmic categorisation using K-means clustering [20]. Using the K-means categorisation the relationships within the numerical variables are preserved, and hence the distance information within the numerical variables influences the quantification.

3.3 Interactive Modification

The quantification achieved through categorisation of numerical data followed by CA is presented to the user using

parallel coordinates, as shown in figure 1. This is a suggestion of how the categories can be quantified, based on the relationships within the data set, and also a presentation of similarities between categories. In figure 1 for instance, the categories *five* and *three* in the *number-of-cylinders* variable (pointed out with red arrows) are considered similar since they are positioned close together, whereas *eight* (pointed out with a green arrow) is considered less similar to *five* and *three* since it is positioned further away from them.

Although the algorithmic quantification is efficient, a domain expert may possess knowledge about the data and of the analysis task that the algorithm is unable to detect. To make use of this domain knowledge MiDAVisT provides possibilities for the user to modify the result of the quantification. The modifications include a manual reordering, which is performed by dragging and dropping categories within an interactive display, as well as a category weighting where the user assigns weight values to combinations of categories to indicate if they are to be more or less similar to each other, followed by CA re-computation as described in detail in [16].

In addition to this MiDAVisT also provides the possibility of undoing the modifications made by the user. This provides interactivity and exploratory freedom to the user by allowing analysis of different modifications without demanding a re-quantification to return to the quantification originally suggested by the algorithm.

3.4 Category Merging

For categorical variables with high cardinality, difficulties may arise if different categories are represented with numerical values close to each other. Since one category may conceal others this can cause a cluttered display. To avoid this MiDAVisT provide possibilities to merge several categories into one.

The merging can be performed based on the distance between the quantified categories, using a distance threshold which is interactively controlled by the user. This results in a merging where highly similar categories are merged into one representative category. Another approach available is a manual grouping where the user interactively selects a number of categories that are to be merged into one. In addition to this MiDAVisT provides possibilities of splitting any merged group of variables into its original categories throughout the analysis process. Figure 2 displays the scatter plot and figure 3 the graphical user interface (GUI) used for merging of categories. The glyphs of the scatter plot represent the categories of a selected variable, the x-axis represents the first principal axis achieved through CA, and the y-axis represents the second principal axis. In the GUI a threshold is set to identify categories that are close enough to be merged into one category, the suggested groups are represented by colour in the scatter plot, where the same colour is used for all categories within a group. One group containing three categories is selected and highlighted in black in figure 2, and the names of the categories are displayed within the GUI (figure 3).

4. VISUAL EXPLORATION

The quantification process described in section 3 not only provides a quantification of categorical variables which is based on relationships within the whole data set and on the

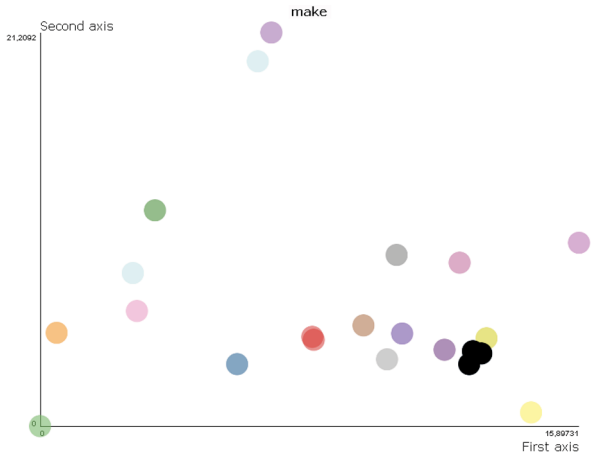


Figure 2: The scatter plot of the category merging interface, which displays the distribution of categories within a selected variable along the first and second principal axis resulting from correspondence analysis. In this example three categories (highlighted in black) are selected to be merged into one.

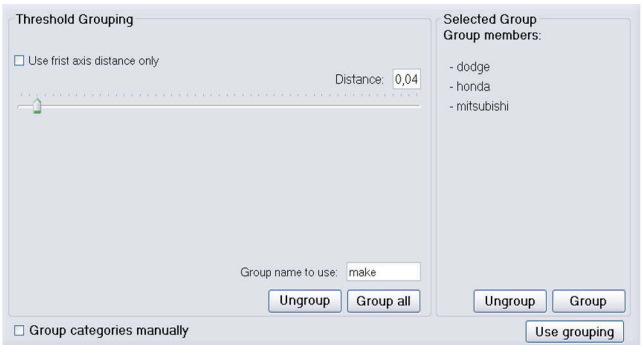


Figure 3: The graphical user interface (GUI) for merging categories. Merging can be either fully manual or guided by a similarity threshold. The names of the three categories selected in figure 2 are displayed in the right part of the GUI.

knowledge of an expert user. Through the visual representations used, it also provides an understanding of relationships and similarities between categories. Furthermore, any data set that has been quantified in MiDAVisT can be saved and re-opened at a later time, preserving all information on quantification and modifications made by the user.

In addition to the quantification MiDAVisT also provides an interactive environment for visual exploration of categorical and mixed data sets using multiple coordinated views. Within this environment the data set is treated as a numerical data set and three common visual representations for numerical multivariate data are used. Figure 4 shows the environment where a scatter plot matrix [3] is positioned in the top left view, a table lens [22] in the top right view and parallel coordinates [14, 28] in the bottom view. Within the table lens and parallel coordinates the category names are presented in addition to the numerical representations. The

Figure 4: The multiple view environment for visual exploration of categorical or mixed data sets. In the top left view is a scatter matrix displaying variable pair correlation using green and purple colour. In the top right view is a table lens where the rows are ordered according to the *make* variable. The bottom view displays the data using parallel coordinates. The purple slider to the right of the parallel coordinates is used to control the transparency of the parallel coordinates lines. Colouring, selection and highlighting is coordinated between the views. In this example all Porsches are highlighted using red colour.

views are coordinated so that any selection or highlighting of items in one view is immediately reflected in the others.

To facilitate detection of structures and relationships in the data three different colour schemes are available within MiDAVisT. The default colour scheme uses a single colour to represent the data items that are not highlighted, as shown in figure 4. Using this colour scheme, individual items that are selected and highlighted in red are easily perceived in the parallel coordinates and table lens, enabling an understanding of the behaviour of individual items. The second colour scheme facilitates understanding of the relationships of categories, by supplying one individual colour for each category of a selected categorical variable, as shown in the top view of figure 5 where colouring is done according to the categories of the *number-of-cylinders* variable and parallel coordinates are used to display the colour schemes.

The third colour scheme, shown in the bottom view of figure 5, is used to emphasise cluster structures within the data set. This colouring is achieved by performing K-means clustering on the whole data set after the categorical variables have been quantified. Each cluster is assigned a unique colour and the data items are coloured according to their cluster membership.

In MiDAVisT the understanding of relationships between variables is facilitated through visual representation of correlation values. The Pearson correlation coefficient, r, is computed for every pair of variables according to equation 3, where N is the total number of data items and $\vec{x}_j$ and $\vec{x}_k$ are variables where $j, k = 1, ..., M$ and M is the total number of variables in the data set.

$$r(\vec{x}_j, \vec{x}_k) = \frac{N \sum_{i=1}^{N} x_{i,j} x_{i,k} - \sum_{i=1}^{N} x_{i,j} \sum_{i=1}^{N} x_{i,k}}{(N \sum_{i=1}^{N} x_{i,j}^2 - (\sum_{i=1}^{N} x_{i,j})^2)(N \sum_{i=1}^{N} x_{i,k}^2 - (\sum_{i=1}^{N} x_{i,k})^2)} \quad (3)$$

The correlation values are represented by coloured cells in the top left half of the scatter matrix, as shown in figure 4, where each cell represent the correlation of a variable pair. Positive correlation is represented by purple and negative correlation by green. Strong correlations are represented by darker colour than weak correlation.

5. CASE SCENARIO

This section describes how a fictional person, a veterinarian named Cate, can use MiDAVisT to analyse a mixed data set. The data set used in the case scenario is a slightly re-

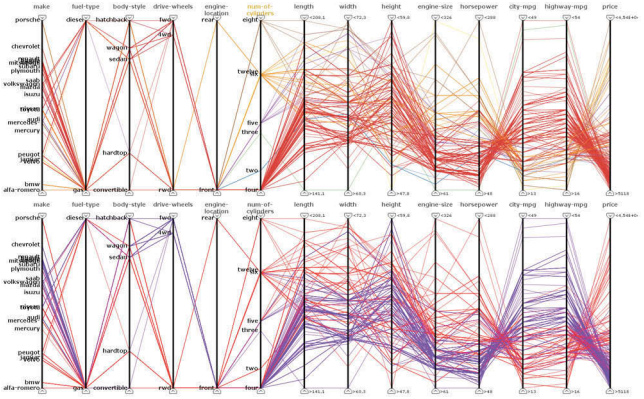


Figure 5: Two of the colour schemes available in MiDAVisT. The top view displays a colouring according to the *number-of-cylinders* variable, where each category of this variable is assigned a unique colour. The bottom view displays a cluster colouring, where colours are assigned according to cluster belonging when K-means clustering has been performed on the whole data set.

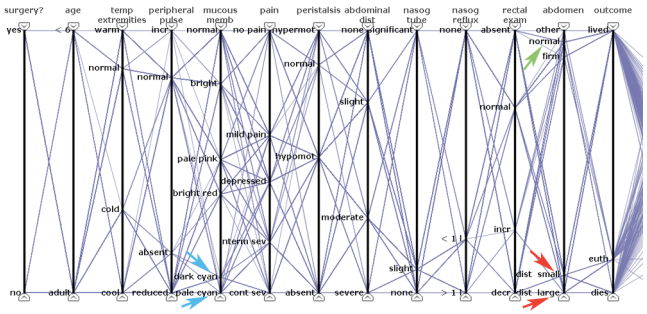


Figure 6: The suggested quantification of the horse colic data set. Categories positioned close to each other, such as *distended small intestines* and *distended large intestines* (red arrows), indicate that these horses have similar symptoms in general, whereas horses belonging to a category positioned further away, such as *normal* (green arrow), have symptoms that are less similar to the previous categories. The blue arrows point out the horses that have *dark cyanotic* or *bright cyanotic* coloured *mucous membranes*, which also are positioned close together, indicating similar symptoms.

duced version of the horse colic data set in [1], including 13 categorical variables and 5 numerical. All conclusions drawn in the scenario that require domain knowledge are based on additional information included with the data set.

Cate is about to analyse a data set of 300 horses that have been treated for colic. The data contains information on different symptoms such as body temperature, pulse, abdomen shape and pain, as well as additional information on whether surgery was performed and on the outcome of the treatment. Cate has previously been introduced to some interactive tools for data visualization and exploration through a friend, and is hence familiar with common visual representations.

Cate starts the analysis by loading the data set into MiDAVisT, and decides to use the clustering approach to categorise the numerical variables, since she is mainly interested in relationships between symptoms that are based on all variables in the data set. MiDAVisT automatically performs K-means clustering followed by correspondence analysis, as described in section 3, and within a few seconds a quantification suggestion is presented using parallel coordinates (figure 6). Within this display Cate can for instance see that the *distended small intestines* and *distended large intestines* categories (pointed out with red arrows) are positioned close together, indicating that most of these horses had similar symptoms, whereas the horses that have a *normal* abdomen (pointed out with a green arrow) mostly have less similar symptoms to the horses with *distended small intestines* and *distended large intestines*, as indicated by it being positioned further away. Furthermore she can see that the quantification indicates that horses having *dark cyanotic* or *bright cyanotic* coloured *mucous membranes* (pointed out with bright blue arrows) have similar symptoms, which agrees with her knowledge that both these colours are indicators of serious circulatory compromises.

In general Cate considers that the suggested quantification and similarities agrees with her previous knowledge and experience. However, she finds the ordering of the *temperature of extremities* (third axis from left in figure 6) slightly illogical, since normal temperature is positioned closer to cold than to cool. Due to this she decides to manually change the ordering of this variable by dragging the cool category and dropping it on top of the cold category. This immediately swaps the positions of the two categories, as shown in figure 7.

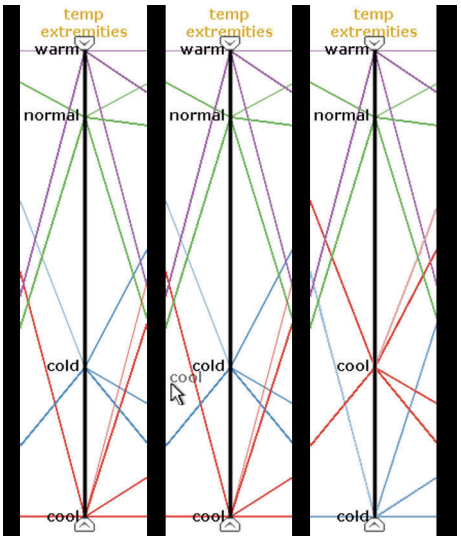


Figure 7: Manual modification of category positions. The left figure displays the suggested quantification. In the centre figure the user has dragged the *cool* category on top of the *cold* category. When dropping a category on top of another they swap positions, as shown in the right figure.

Figure 8: The multiple views environment when coloured according to the categories of the *outcome* variable, green represent *lived*, blue represent *euthanized* and red represent *dies*. In the table lens the rows are ordered according to the *abdominal distension* variable, with *none* as top category, followed by *slight* and *moderate*, and with *severe* as thin lines below. The empty lines at the bottom are missing values that represent horses where this variable was not recorded.

Since no variables contain large numbers of categories, Cate does not find it useful to merge any categories, and since she is now satisfied with the quantification she opens the visual exploration environment within MiDAVisT (figure 8). From her experience and knowledge Cate is aware that the abdominal distension is an important symptom, and based on this she decides to examine the relationship between the *abdominal distension* variable and the *outcome* variable. Figure 8 shows the multiple views environment when colouring is done according to the *outcome* variable, where green corresponds to horses that survived, blue to horses that were euthanized and red to horses that died from the colic. The rows of the table lens are ordered according to *abdominal distension* and, as can be seen, the two top categories in the table lens (which represent the *none* and *slight* categories) are mainly coloured green, whereas the lower categories, representing *moderate* and *severe* abdominal distension contain more red and blue. This indicates that there is a relationship between the *abdominal distension* of a horse and the *outcome* of the treatment.

One difficulty when displaying categorical variables using parallel coordinates is that the lines conceal each other, making it hard to get an understanding of the category frequencies. As a veterinarian Cate finds it important to know how many of the horses survived, which is hard to tell from the parallel coordinates. Due to this Cate reorders the rows in the table lens, using an ordering according to *outcome* instead of *abdominal distension*. Figure 9 displays the re-

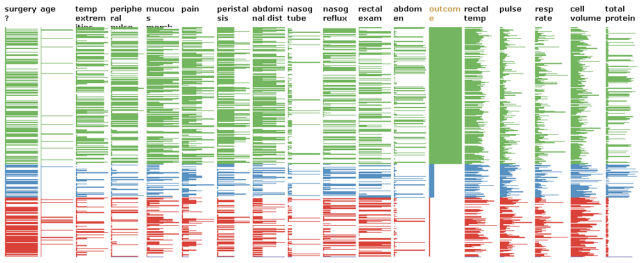


Figure 9: The table lens sorted according to *outcome*. With this ordering the category frequencies are easily perceived. More than half of the horses survived (green category), of the ones that did not survive less than half was euthanized (blue category), while the remaining died during the treatment (red category).

ordered table lens, where the frequency of a category can be told from the height of the group of rows that represent a category. In the figure the top category, representing horses that survived, is higher than the other categories, and hence most horses survived. From the other two categories it can be seen that, of the horses that did not survive, the majority were not euthanized (represented by blue).

In the scatter matrix (figure 10) Cate notices the group of purple cells in the bottom left part of the matrix. These

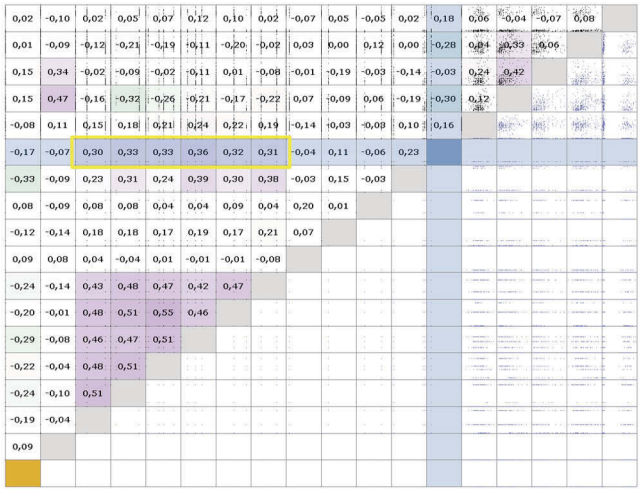


Figure 10: The scatter matrix of the quantified horse colic data set. In the bottom left part of the matrix a group of cells are coloured purple, due to high correlation between the variable pairs. The variable pairs including the *outcome* variable are highlighted in blue. The cells pointed out with a yellow rectangle are variables that have a correlation between 0.30 and 0.36 with the *outcome*.

cells represent a group of variables where all variable pairs are highly correlated. The variables are *temperature of extremities*, *peripheral pulse*, *mucous membrane colour*, *pain*, *peristalsis* and *abdominal distension*. The correlation indicates that all of these symptoms are related to each other. From the scatter matrix cells representing the *outcome* variable, which are highlighted in blue in the figure, it can be seen that the group of highly related variables are also correlated with the *outcome* variable, with correlation values ranging from 0.30 to 0.36, as pointed out by the yellow rectangle. This indicates that there is also a relationship between the previously mentioned symptoms and the outcome of the treatment, although not as strong as the relationship between the symptom variables.

The indicated relationships within the highly correlated group of variables agree with Cate's experience and domain knowledge, and she decides to continue her analysis to identify if there are any major groups of horses with similar symptoms and similar outcome. For this she decides to colour the data items using the clustering approach described in section 4. K-means clustering is applied to the whole quantified data set and the visual representations are coloured accordingly. The result, displayed using parallel coordinates, is shown in figure 11. From this it can be seen that two clusters exist, coloured red and purple, where the purple cluster mainly includes horses that survived whereas the red cluster mainly includes horses that were euthanized or died. By looking at the distribution of colours for the variables Cate is able to identify some interesting relationships that can be useful to her in her future practice and that verifies her previous experience. For instance she notices that most of the horses that did not survive had *pale pink*, *bright red*, *dark cyanotic* or *bright cyanotic* coloured *mucous membrane*, almost all of them had some *abdominal distension* and most of them also had high values for *packed cell volume*.

Cate is satisfied with what she has found so far, and decides to continue the analysis at some other time. She saves the quantification results for future analysis, and will be able to re-open the exploration environment using the same data set and quantification at a later time.

6. CONCLUSIONS AND FUTURE WORK

This paper presents MiDAVisT, an application for quantification of categorical data and exploration of data sets including both categorical and numerical variables. The quantification process used in MiDAVisT was introduced in [16] and enables a quantification based on the relationships within all variables of a mixed data set as well as utilising the knowledge of a domain expert.

MiDAVisT extends the analysis possibilities enabled by the quantification process by providing a multiple view environment for visual exploration and analysis of categorical and mixed data sets, using general and commonly used visual representations and analysis methods developed for numerical data sets. The main benefit of MiDAVisT is its ability to merge the quantification process into an interactive environment that enables versatile exploration.

The effectiveness of MiDAVisT is presented through a case scenario, where the quantification process as well as the features of the visual exploration environment are used to analyse symptoms and outcome of horses that were treated for colic. The scenario demonstrates how the quantification process can be used to identify symptoms that are closely related to each other, and how a mixed data set can be successfully explored by combining quantification and analysis methods traditionally used for purely numerical data, and how relationships between categorical and numerical variables can be identified through this.

Future work includes an evaluation of the efficiency and usefulness of the quantification process and the application together with domain experts and potential end-users. Furthermore additional analysis methods will be included in the exploration environment to further facilitate exploration and detection of patterns and relationships.

7. REFERENCES

[1] A. Asuncion and D. Newman. UCI machine learning repository. http://archive.ics.uci.edu/ml/, 2007.

[2] M. Q. W. Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the Workshop on Advanced Visual Interfaces*, pages 110–119, 2000.

[3] R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, May 1987.

[4] S. Boriah, V. Chandola, and V. Kumar. Similarity measures for categorical data: A comparative evaluation. In *Siam International Conference on Data Mining*, pages 243–254. SIAM, April 2008.

[5] A. Buja, J. A. McDonald, J. Michalak, and W. Stuetzle. Interactive data visualization using focusing and linking. In *Proc. IEEE Visualization ́91, San Diego, CA*, pages 156–153, 1991.

[6] J.-D. Fekete. The infovis toolkit. In *Proceedings of the 10th IEEE Symposium on Information Visualization*

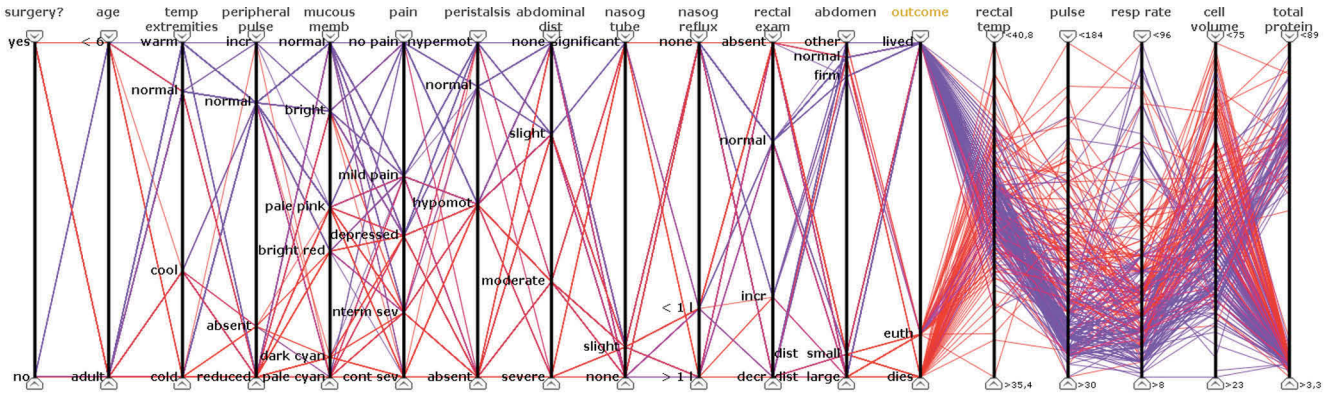


Figure 11: The parallel coordinates view in the exploration environment when the data is coloured according to a K-means clustering. As can be seen two clusters exist, coloured red and purple, where the red cluster mainly includes horses that died and the purple mainly includes horses that survived.

(InfoVis'04), pages 167–174. IEEE Press, October 2004.

[7] M. Friendly. Mosaic displays for multi-way contingency tables. *Journal of the American Statistical Association*, 89(425):190–200, 1994.

[8] M. Friendly. Extending mosaic displays: Marginal, conditional, and partial views of categorical data. *Journal of Computational and Graphical Statistics*, 8(3):373–395, 1999.

[9] M. Friendly. Visualizing categorical data: Data, stories, and pictures. In *Proceedings of the Twenty-Fifth Annual SAS Users Group International Conference*, April 2000.

[10] G. H. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *J. SIAM Numer. Anal.*, Ser. B(2):205–224, 1965.

[11] M. Greenacre. *Multiple Correspondence Analysis and Related Methods*. Chapman & Hall, 2006.

[12] M. Greenacre. *Correspondence Analysis in Practice, 2. ed*. Chapman & Hall, 2007.

[13] S. L. Havre, A. Shah, C. Posse, and B.-J. Webb-Robertson. Diverse information integration and visualization. In *Proceedings of SPIE - The International Society for Optical Engineering*, January 2006.

[14] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(4):69–91, 1985.

[15] M. Jern, S. Johansson, J. Johansson, and J. Franzén. The gav toolkit for multiple linked views. In *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization, CMV '07*, pages 85–97. IEEE Computer Society, July 2007.

[16] S. Johansson, M. Jern, and J. Johansson. Interactive quantification of categorical variables in mixed data sets. In *Proceedings of IEEE International Conference on Information Visualisation, IV08*, pages 3–10. IEEE Computer Society, July 2008.

[17] I. T. Jolliffe. *Principal Component Analysis, 2 ed.* Springer-Verlag, 2002.

[18] R. Kosara, F. Bendix, and H. Hauser. Parallel sets: Interactive exploration and visual analysis of categorical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):558–568, 2006.

[19] S. Ma and J. L. Hellerstein. Ordering categorical data to improve visualization. In *IEEE Information Visualization Symposium Late Breaking Hot Topics*, pages 15–18, 1999.

[20] B. Mirkin. *Clustering for data mining a data recovery approach*. Chapman & Hall, 2005.

[21] A. Patro, M. O. Ward, and E. A. Rundensteiner. Seamless integration of diverse data types into exploratory visualization systems. Technical report, Worcester Polytechnic Institute, 2003.

[22] R. Rao and S. K. Card. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, pages 318–322. ACM, 1994.

[23] J. C. Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization, CMV '07*, pages 61–71. IEEE Computer Society, July 2007.

[24] G. E. Rosario, E. A. Rundensteiner, D. C. Brown, M. O. Ward, and S. Huang. Mapping nominal values to numbers for effective visualization. *Information Visualization*, 3(2):80–95, 2004.

[25] J. Seo and B. Shneiderman. Interactively exploring hierarchical clustering results. *IEEE Computer*, 35(7):80–86, 2002.

[26] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*, chapter 2, page 74. Addison-Wesley, 2006.

[27] M. O. Ward. Xmdvtool: Integrating multiple methods for visualizing multivariate data. In *Proceedings of the Conference on Visualization 1994*, pages 326–333, October 1994.

[28] E. J. Wegman. Hyperdimensional data analysis using parallel coordinates. *Journal of American Statistics Association*, 85(411):664–675, 1990.

FpViz: A Visualizer for Frequent Pattern Mining

Carson Kai-Sang Leung*
Department of Computer Science
The University of Manitoba
Winnipeg, MB, Canada
kleung@cs.umanitoba.ca

Christopher L. Carmichael
Department of Computer Science
The University of Manitoba
Winnipeg, MB, Canada
umcarmi1@cs.umanitoba.ca

ABSTRACT

Over the past 15 years, numerous algorithms have been proposed for frequent pattern mining as it plays an essential role in many knowledge discovery and data mining (KDD) tasks. Most of these frequent pattern mining algorithms return the mined results in the form of textual lists containing frequent patterns showing those frequently occurring sets of items. It is well known that "a picture is worth a thousand words". The use of visual representation can enhance the user understanding of the inherent relations in a collection of frequent patterns. A few visualizers have been developed to visualize the input data or the mined results. However, most of these visualizers were not designed for visualizing the mined frequent patterns. In this paper, we develop a *visualizer for frequent pattern mining*. Such a visualizer—called *FpViz*—gives users an insight about the data, allows them to zoom in and zoom out, and provides details on demand. Moreover, FpViz is also equipped with several interactive features for effective visual support in the data analysis and KDD process for various real-life applications.

Categories and Subject Descriptors

H.1.2 [**Models and Principles**]: User/Machine Systems—*human factors*; H.2.8 [**Database Management**]: Database Applications—*data mining*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces; I.4.0 [**Image Processing and Computer Vision**]: General—*image displays*

General Terms

Algorithms; Design; Experimentation; Human factors; Management; Measurement; Performance; Reliability

Keywords

Knowledge discovery and data mining, visual analytics, visual and interactive data analysis, visual support in the knowledge discovery process, data and knowledge visualization, frequent itemsets, visual data mining

*Corresponding author: C.K.-S. Leung.

VAKD '09, June 28, 2009, Paris, France.

1. INTRODUCTION

Frequent pattern mining [1, 20, 22, 23, 25, 26] aims to search for implicit, previously unknown, and potentially useful information in the form of *frequent patterns* (i.e., frequently occurring sets of items, which are also known as *frequent itemsets*). It plays an essential role in many knowledge discovery and data mining (KDD) tasks. Examples of these KDD tasks include the mining of association rules, correlation, sequences, episodes, maximal frequent patterns, and closed frequent patterns. Hence, frequent pattern mining is in demand in various real-life applications. Mined frequent patterns can answer many questions that help users make important decisions in various real-life situations. The following are some examples:

Q1. Store managers may want to find out how frequently certain kinds of vegetables (e.g., asparagus, broccoli) are purchased *individually* and how frequently are they purchased *together*? What kinds of vegetables are frequently purchased together with eggplants (e.g., {asparagus, broccoli, eggplants, peas})?

Q2. Botanists may want to discover which features or properties associated with edible mushroom are frequently observed?

Q3. University administrators may want to know which popular elective courses (e.g., {AI, Bioinformatics, Computational Geometry}) are frequently taken together by students?

Q4. Bookstore owners may want to know which books are also bought by customers who bought a particular KDD book so that they could bundle these books together for customer convenience?

Q5. Internet providers may want to figure out what Webpages are frequently browsed by Internet users in a single session?

Q6. Service planners may want to know why the demand of some combinations of services is dropping so that they could cancel those combinations and put the resources on other demanding combinations?

Q7. Web administrators may want to find out which collection of Webpages is frequently updated by users? Which groups of users frequent update the Webpages?

Q8. Travel agencies may want to discover where are the favourite spots and when are the popular time for travel?

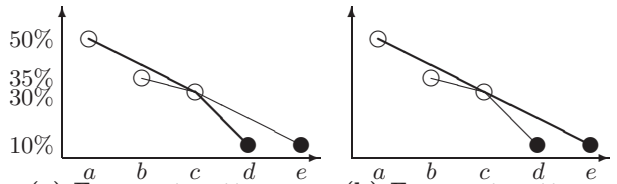
Q9. Phone service providers may want to find out where are the popular calling and receiving countries (e.g., {Canada, France}) for long-distance phone calls so that they could put these countries on their promotional package?

Q10. Security staff may want to know which parts of the building are frequently visited by employees or visitors?

To help answer the above questions in these real-life situations, numerous frequent pattern mining algorithms have been proposed over the past 15 years. However, most of the algorithms return a collection of frequent patterns in *textual form* (e.g., a very long unsorted list of frequent patterns). Consequently, users may not easily discover the knowledge and useful information that is embedded in the data.

Showing a collection of frequent patterns in *graphical form* can show the relations embedded in the data and help users understand the nature of the useful information and discovered knowledge. Hence, researchers have also considered visual analytics [8, 16, 17, 18, 21, 29, 32, 33, 37, 39] and visualization techniques [9, 13, 14, 34] to assist users in gaining insight into massive amounts of data or information. Visualization systems like Spotfire [2], VisDB [15] and Polaris [35] have been developed for visualizing data. For systems that visualize the mining results, the focus has been mainly on results such as clusters [19, 30], decision trees [3], social networks [4] or association rules [7]. However, *not* many visualizers were designed for visualizing frequent patterns.

Recently, some researchers have shown interests in visualizing frequent patterns. For example, Yang [38] developed a system that can visualize frequent patterns. However, his system was primarily designed to visualize association rules, and it does not scale very well in assisting users to immediately see certain useful information (such as exact frequencies or support) of a very large number of frequent patterns. As another example, Munzner et al. presented a visualizer called PowerSetViewer (PSV) [28], which provides users with guaranteed visibility of frequent patterns in the sense that the pixel representing a frequent pattern is guaranteed to be visible by highlighting such a pixel. However, multiple frequent patterns may be represented by the same pixel. As the third example, we previously proposed a visualization system—called FIsViz [26]—that aims to visualize frequent patterns. FIsViz represents each frequent pattern by a polyline in a two-dimensional space. The location of the polyline indicates the exact frequency of the pattern explicitly. As a result, FIsViz enables users to visualize the mined results (i.e., frequent patterns) for many real-life applications. However, in some other applications (especially, when the number of frequent patterns is huge), FIsViz may not scale very well. Users may require more effort to be able to clearly visualize frequent patterns. The problem is caused by the use of polylines for representing frequent patterns. To elaborate, the polylines can be bent and/or can cross over each other. This makes it difficult to distinguish one polyline (representing a frequent pattern) from another. For example, in Figure 1, how to distinguish the two frequent patterns $\{a, c, d\}$ & $\{b, c, e\}$ from another two patterns $\{a, c, e\}$ & $\{b, c, d\}$ if we did not use different thickness for the polylines?



(a) Frequent patterns $\{a, c, d\}$ & $\{b, c, e\}$ **(b) Frequent patterns $\{a, c, e\}$ & $\{b, c, d\}$**

Figure 1: Polylines showing $\{a, c, d\}$ & $\{b, c, e\}$ vs. polylines showing $\{a, c, e\}$ & $\{b, c, d\}$.

Hence, some natural questions to ask are: Can we design a scalable system that helps users visualize frequent patterns effectively? Can we have an alternative representation that minimizes the bend and crossover of polylines? In response to these questions, we explored an alternative representation [27], which uses two half-screens to visualize the discovered knowledge about frequent patterns: one half of the screen showing all frequent patterns and another half showing their frequencies.

In this paper, we propose a visualizer that uses a full screen for visualizing frequent patterns. The proposed visualizer enhances the KDD process by providing answers to some important business questions (e.g., Q1–Q10 above). The **key contribution** of our work is a novel interactive and scalable *frequent pattern visualizer*, called *FpViz*, which provides users with effective visual support in the data analysis and KDD process. Specifically, FpViz uses orthogonal graphs for visualizing frequent patterns. The visualizer provides users with clear and explicit depictions about frequent patterns that are embedded in the data of interest. Hence, FpViz enables users to infer (i) patterns at a glance and (ii) answers to many questions encountered in various real-life applications. It also provides interactive features for constrained mining and interactive mining. Moreover, with FpViz, users can (i) efficiently find closed or maximal itemsets and (ii) properly formulate association rules from the displayed frequent patterns.

This paper is organized as follows. Next section briefly describes related work and background. In Section 3, we introduce our FpViz and describe its design; in Section 4, we present interactive features provided by FpViz. Section 5 shows evaluation results. Finally, conclusions are presented in Section 6.

2. RELATED WORK AND BACKGROUND

Developing effective visualization systems for KDD has been the subject of many studies. This line of research can be subclassified into two general categories: (i) systems for visualizing raw data and (ii) those for visualizing data analysis or data mining results. Examples of systems in the first category include Spotfire [2], independence diagrams [5], VisDB [15], and Polaris [35]. These systems were built for visualizing data. Systems in the second category focus on *visualizing the mining results*, which include clusters [19], decision trees [3, 10], association rules [6, 12, 38], and frequent patterns [26, 27, 28, 38, 40]. Let us briefly discuss below some relevant systems for visualizing association rules or frequent patterns.

Yang [38] designed a system mainly to visualize association rules—but can also be used to visualize frequent patterns—in a two-dimensional space consisting of many vertical axes. In his system, all domain items are sorted according to their frequencies and are evenly distributed along each vertical axis. A frequent pattern consisting of k items (i.e., a k-itemset) is then represented by a curve that extends from one vertical axis to another connecting k such axes. The thickness of the curve indicates the frequency (or support) of such a frequent pattern. However, such a representation suffers from the following problems: (i) The use of thickness only shows *relative* (but not *exact*) frequency of the patterns. Comparing the thickness of curves is not easy. (ii) Since items are sorted and *evenly* distributed along the axes, users only know some items are more frequent than the others, but cannot get a sense of how these items are related to each other in terms of their exact frequencies (e.g., whether item a is twice as frequent as, or just slightly more frequent than, item b). (iii) Although Yang's system is able to show both association rules and frequent itemsets, his system does not provide users with many interactive features, which are necessary if a large graph containing many items to be displayed.

Frequent itemset visualizer (FIsViz) [26] is one of the recently developed visualizers. It was designed for visualizing frequent itemsets. It represents a k-itemset represented by a polyline that connects k nodes (where each node represents an item in the k-itemset) in a two-dimensional space. The frequency (or support) of the i-th prefix of an itemset X is indicated by the position of the i-th node in the polyline representing X. For example, when $X = \{a, c, d\}$, the frequencies of its prefixes $\{a\}$ and $\{a, c\}$ are respectively indicated by the positions of nodes a and c in the polyline. Similarly, the frequency of the itemset $X = \{a, c, d\}$ is represented by the y-position of the node d in that polyline. See Figure 1(a). With such itemset representation, slopes of different sectors of a polyline can vary. In other words, the entire polyline may not be a straight one (i.e., it may be bent). Moreover, polylines representing different itemsets may cross each others. This makes it difficult for users to distinguish one sector of a polyline from another. See Figure 1.

3. FpViz: OUR PROPOSED FREQUENT PATTERN VISUALIZER

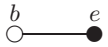
In this section, we present our proposed **F***requent* **p***attern* **Vi***sualiz***e***r* (**FpViz**). Here, FpViz is connected to a frequent pattern mining algorithm (e.g., FP-growth [11]), which finds frequent patterns from transaction database. Once frequent patterns are found, FpViz effectively displays them for the data analysis. Note that FpViz is not confined to using FP-growth for frequent pattern mining. It can use some other frequent pattern mining algorithms (e.g., , DCF [20] for constrained mining, UF-streaming [23] for stream mining, UF-growth [24] for uncertain data mining, Apriori [1]).

Like FIsViz [26], our proposed FpViz also shows frequent patterns consisting of k items (i.e., k-itemsets) in a two-dimensional space. The x-axis shows the n domain items. We allow the user to specify his preference on the ordering of these domain items. For example, the user can arrange the items in (i) non-ascending frequency order, (ii) lexicographical order, or (iii) some other orders (e.g., put those items of interest—such as promotional items—on the left and less interesting items on the right side of the x-axis) for constrained mining. The y-axis shows the frequencies of the frequent patterns.

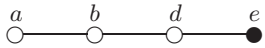
Unlike FIsViz (which represents frequent patterns as polylines), the basic representation for our proposed FpViz is an orthogonally laid out node-link diagram. According to graph aesthetics [31, 36], reducing the number of edge crossings can improve the legibility of graphs. Similarly, assigning uniform lengths to edges and minimizing bends can enhance the legibility of the node-link diagram. Since our datasets are potentially very large, a primary criterion in our design is to minimize edge crossings and bends. We, therefore, adopted an orthogonal layout mechanism that preserves edge crossings to a minimum. Bends occur only at 0° or 90° angles. As a result, FpViz minimizes crossings, facilitating legibility and visual comprehension.

3.1 Representing Frequent Patterns

Our proposed FpViz represents each frequent pattern X consisting of k items (i.e., k-itemset) by a horizontal line connecting k nodes (represented by k circles), where each node represents an item within the frequent pattern X. For example, the 2-itemset $\{b, e\}$ is represented by a horizontal line connecting two circles (where each circle represents an item), as follows:



Note that, between the two circles, one of them is filled. The filled circle (i.e., disc) represents the last item (according to the item order $\mathcal{R}$) in the frequent pattern $\{b, e\}$. As another example, the 4-itemset $\{a, b, d, e\}$ is represented by a horizontal line connecting four circles (where the last one is filled), as follows:



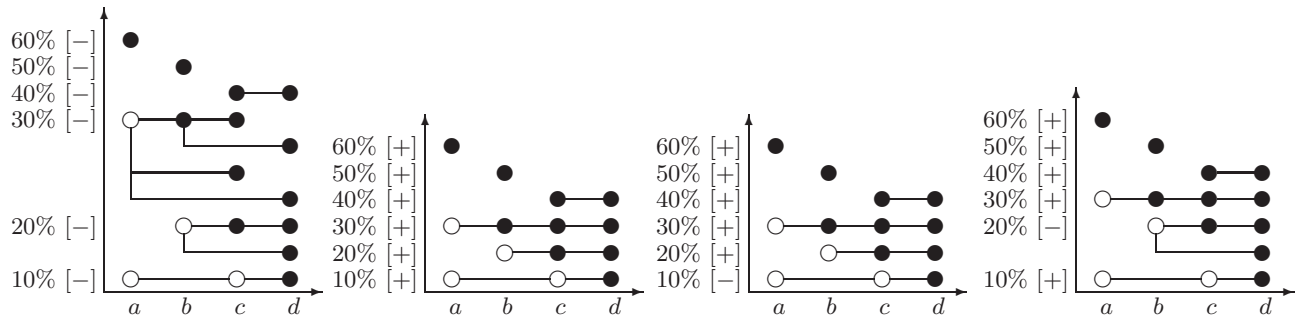
For singletons (i.e., 1-itemsets), they are represented by just filled circles (or filled diamonds for user convenience) in FpViz. For example, the singleton $\{e\}$ is represented as:

To summarize, each frequent pattern consisting of k items (i.e., k-itemset) is represented by a *horizontal line* connecting k circles, with the last circle filled.

3.2 Showing the Frequencies of Frequent Patterns

The frequency of a frequent pattern consisting of k items (which is represented by a horizontal line connecting k circles with the last circle filled) is indicated by the y-value (i.e., y-position) of the filled circle. This way of showing the frequencies work reasonable well when each frequent pattern has a distinct frequency (i.e., at most one horizontal line for each frequency value—the y-value).

However, for many real-life applications, it is not uncommon that multiple frequent patterns happen to have the same frequency. In these situations, we apply *compression*



(a) The expanded view ("expand all") **(b) The collapsed view (the default view)** **(c) An expanded view ("expand the 10% line")** **(d) An expanded view ("expand the 20% line")**

Figure 2: Expanded and collapsed views for visualizing frequent patterns with our proposed FpViz.

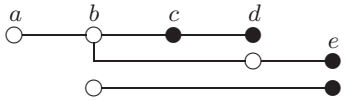
techniques to our proposed FpViz: If the two frequent patterns X and Y of the same frequency share the same prefix, then their common prefix is merged. The suffixes of X and Y are then branching out from the last item of the common prefix. For example, if frequent patterns $\{a, b, c, d\}$ and $\{a, b, d, e\}$ (which share the same prefix $\{a, b\}$) are of the same frequency, they can be represented as follows:

Here, $\{c, d\}$ and $\{d, e\}$ are two branches of the common prefix $\{a, b\}$.

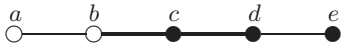
A special case of the merge occurs when a suffix of Y is branching out from the last item of X (i.e., X is a prefix of Y). In this case, the two horizontal lines representing the two frequent patterns X and Y would be merged into one line. For example, for frequent patterns $\{a, b, c\}$ and $\{a, b, c, d\}$, the former is a prefix of the latter. Hence, these two frequent patterns can be merged to form the following:

Here, the filled circle d indicates the last item of the frequent pattern $\{a, b, c, d\}$, whereas the filled circle c indicates the last item of the prefix $\{a, b, c\}$. Note that this merge helps reduce the number of horizontal lines to be drawn (i.e., reduce the amount of vertical space required for displaying the frequencies of all the frequent patterns).

When the number of mined frequent patterns is not huge, the merging of patterns with their prefixes having the same frequencies (e.g., the case for $\{a, b, c\}$ and $\{a, b, c, d\}$) reduces the amount of vertical space required. However, when the number of mined frequent patterns is huge, we may still run out of vertical space to fit all horizontal lines representing all the mined frequent patterns—even when merging is applied. Hence, we need to apply further compression technique as follows. To reduce the amount of space required in the y-direction, if multiple frequent patterns (say, m frequent patterns represented by m' horizontal lines, where $m' \leq m$) have the same frequency, they are projected or collapsed into *one horizontal line* (instead of m' lines). For instance, frequent patterns $\{a, b, c\}$, $\{a, b, c, d\}$, $\{a, b, d, e\}$ and $\{b, e\}$ are of the same frequency:



These $m = 4$ frequent patterns (represented by $m' = 3$ horizontal lines) are collapsed into one horizontal line, as shown below:



By so doing, each existing frequency value would be represented by one—and only one—horizontal line. For example, Figure 2(a) shows $m = 13$ frequent patterns represented by two disjointed filled circles for singletons $\{a\}$ & $\{b\}$ and $m' = 8$ horizontal lines for other 11 non-singleton frequent patterns. Figure 2(b) shows how these $m' = 8$ horizontal lines are collapsed into four lines by using our proposed FpViz. The resulting view shows two disjointed filled circles and four lines, which represent $m = 13$ frequent patterns having $2 + 4 = 6$ distinct frequencies.

It is important to note that (though it may not be obvious in this black-and-white version of our paper), FpViz uses the color of the circle to indicate the number of occurrences of an item within those frequent patterns of the same frequency. For example, a lighter circle a indicates that a only occurs in one frequent pattern, whereas a darker circle b indicates that b occurs more often (e.g., in four frequent patterns $\{a, b, c\}, \{a, b, c, d\}, \{a, b, d, e\}$ and $\{b, e\}$). Moreover, the thickness of the line indicates the number of horizontal lines that were collapsed into one.

3.3 Collapsing and Expanding the Horizontal Lines

Our proposed FpViz normally shows frequent patterns in the (default) *collapsed view* so as to reduce the mount vertical space required for displaying all patterns. As this collapsed view may hide some details, FpViz provides the user with the option to expand on any portion of the graph that is interesting to him by clicking the [+] button. By so doing, the user would be able to clearly get all the details.

As an example, when the user clicks the [+] button for frequency=10%, FpViz expands the horizontal line representing frequent patterns of frequency=10%. Consequently, the user obtains the expanded view as presented in Figure 2(c), which shows that such a horizontal line represents the frequent pattern $\{a, c, d\}$. Similarly, when the user clicks the [+] button for frequency=20%, FpViz expands the horizontal line representing frequent patterns of frequency=20%. The user then obtains the expanded view as presented in Figure 2(d), which shows that such a horizontal line represents three frequent patterns $\{b, c\}$, $\{b, c, d\}$ and $\{b, d\}$. Note that the user is not confined to clicking only one [+] button, he could click all six [+] buttons to obtain an expanded view as shown in Figure 2(a).

3.4 Observations

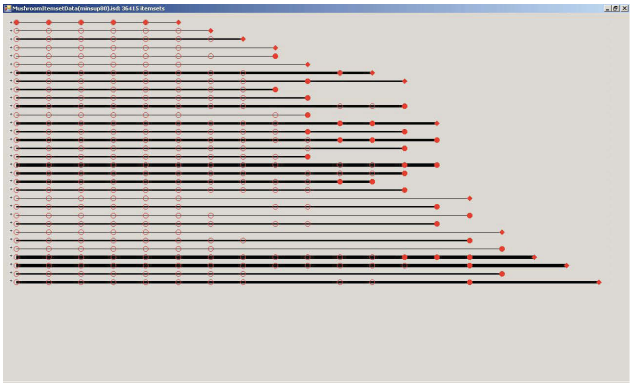
With this representation of frequent patterns and their frequencies in our proposed FpViz, users can observe the following from the default collapsed view:

1. By default, FpViz arranges the domain items in non-ascending frequency order. As a result, the most frequently occurring item (which with the highest frequency) appears on the left side and the least frequently occurring one appears on the right side. In other words, users can easily get an insight about the frequency ranking of all the domain items by walking along the x-axis. For example, we observed from Figure 2(b) that item a is the most frequent domain item, which is followed by items b and c, and item d is the least frequent domain item. (It is important to note that, users are not confined to this ordering; they can choose other ordering $\mathcal{R}$ to arrange all items in the domain.)
2. FpViz gives information about frequency distribution of items in the frequent patterns. For example, users can tell how many distinct frequency values can an item take on (in any frequent pattern) by counting the number of its y-values. For example, we observed from Figure 2(b) that item b takes on tree distinct frequency values—namely, 20%, 30% and 50%—by counting the number of circles for b.
3. The frequency of any subset of a frequent pattern X is guaranteed to be higher than or equal to that of X. Hence, the disjointed filled circle (or diamond) representing any singleton subset of X (or the horizontal line representing any non-singleton subset of X) is guaranteed to appear on or above the horizontal line representing X. For example, let us consider $\{c, d\}$, which is a subset of frequent pattern $\{a, c, d\}$. We observed from Figure 2(b) that their frequencies are 40% and 10%, respectively. In other words, the frequency of the subset ($\{c, d\}$) is higher than that of the frequent pattern $\{a, c, d\}$. Similarly, we observed that the frequency of both $\{b, c\}$ and $\{b, c, d\}$ are the same (of 20%).
4. Conversely, the frequency of any superset of a frequent pattern X is guaranteed to be lower than or equal to that of X. Hence, the horizontal line representing any superset of X is guaranteed to appear on or below the horizontal line representing X. For example, we observed from Figure 2(b) that the frequency of the superset ($\{a, c, d\}$) is lower than that of the frequent pattern $\{c, d\}$. Similarly, we observed that the frequency of both $\{b, c\}$ and $\{b, c, d\}$ are the same.
5. The highest frequency value for each item u is the frequency of the singleton $\{u\}$. For example, the highest frequency value of item b is 50%, which the frequency of the singleton $\{b\}$.
6. If a horizontal line starts with a hollow circle and follows by a filled circle, then users can reveal without requiring any expansion (i.e., without clicking any [+] button) and guarantee that a frequent pattern consisting of only two items exists with that frequency. For example, we observed from Figure 2(b) that a horizontal line at frequency=30% starts with a hollow circle for item a and follows by a filled circle for item b. We then knew, without expanding such a horizontal line, that there exists frequent pattern $\{a, b\}$ and its frequency is 30%. (This observation could be confirmed by clicking the [+] button for the 30% line.)
7. If a horizontal line involves *only two* items, users do not need to expand such a line (i.e., do not need to click the [+] button) to get the complete information about the frequent pattern consisting of only two items. The frequency of this pattern is clearly indicated by the y-position of the last item. Moreover, if such a horizontal line starts with a diamond, then this line gives additional information that the 2-itemset and its singleton prefix are of the same frequency. For example, we observed from Figure 2(b) that there exists a frequent pattern $\{c, d\}$ consisting of only two items and its frequency is 40%. Moreover, its singleton prefix $\{c\}$ is also of frequency 40%.
8. For a horizontal line representing frequency=y%, if the first circle (representing item u) is filled, then the singleton has frequency of y%. For example, the frequency of singleton $\{b\}$ is 50% and that of $\{c\}$ is 40%.
9. For a horizontal line representing frequency=y%, if the first circle (filled or hollow) represents an item u and the second circle (representing item v) is filled, then the frequency of the frequent pattern $\{u, v\}$ = y%. For example, the frequencies of $\{c, d\}$ and $\{a, b\}$ are 40% and 30%, respectively.
10. For a horizontal line that involves more than two items and does not have a filled circle in its second position in the collapsed view, FpViz provides information about the *absence* of any items from frequent patterns of that frequency. For example, since item b does not appear in the 10% line, b is guaranteed not to appear in any patterns of that frequency.

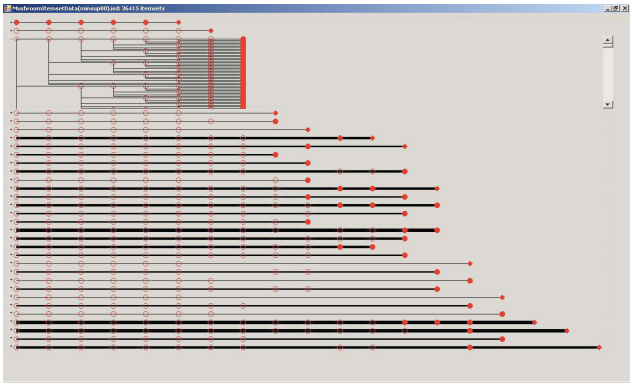
    However, one may not be able to easily determine the contents of the frequent patterns represented by this line. For example, the 10% line in Figure 2(b) could represent (i) frequent patterns $\{a, d\}$ and $\{c, d\}$ and/or (ii) frequent pattern $\{a, c, d\}$. (This explains why FpViz provides users with [+] buttons for expanding the lines to clearly obtain detailed information.)

In addition, users can also observe the following when they click one or more [+] buttons to expand horizontal lines of their interest:

11. After expanding a horizontal line, users can obtain the cardinality k of the k-itemset (which is represented by



(a) The collapsed view (the default view)



(b) An expanded view (when one horizontal line is expanded)

Figure 3: Snapshots of our proposed FpViz showing the collapsed and expanded views.

each expanded line) by counting the number of circles on such an expanded line (from the leftmost circle to a filled one). For example, the cardinality of $\{b, c\}$ is 2 because there are 2 circles from the leftmost one to the first filled one along the 20% line shown in Figure 2(d). Similarly, the cardinality of $\{b, c, d\}$ is 3 because there are 3 circles from the leftmost one to the second filled one). While users can count the number of circles to determine the cardinality of a frequent pattern, FpViz provides the feature called *query on cardinality* for user convenience. See Section 4.

12. The first portion of an expanded horizontal line represents a prefix of a frequent pattern X. If such a portion does not end with a filled circle, then the frequency of such a prefix of X is different from that of X. For example, we observed from Figure 2(c) that the frequent pattern $\{a, c, d\}$ does not have the same frequency as its prefix $\{a, c\}$ (10% vs. 30%) because the circle representing c in $\{a, c\}$ is hollow.

13. If there are more than one filled circle on an expanded horizontal line, then a frequent pattern X and at least one of its prefix have the same frequency. For example, as shown in Figure 2(d), frequent patterns $\{b, c, d\}$ and its prefix $\{b, c\}$ have the same frequency of 20%.

14. Once users observe $sup(X)$ and $sup(X \cup Y)$, they can compute the support, confidence and lift of association rule $X \Rightarrow Y$ using $sup(X \cup Y)$, $\frac{sup(X \cup Y)}{sup(X)}$ and $\frac{sup(X \cup Y)}{sup(X) \times sup(Y)}$ respectively. Moreover, if $sup(X) = sup(X \cup Y)$, then users can easily determine that $conf(X \Rightarrow Y) = 100\%$.

3.5 Discussions

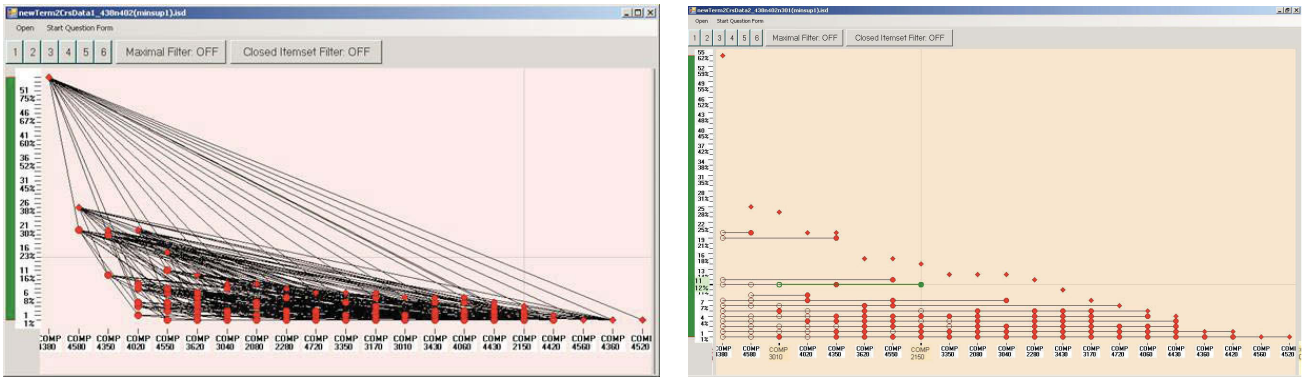
How to represent frequent patterns with a huge number of distinct frequency values? Recall that FpViz applies the compression technique for merging several horizontal lines that represent frequent patterns of the same frequency into one line. To a further extent, if the number of distinct frequencies in the mined results exceeds the number of vertical pixels (or the number of horizontal lines allocated for the space), we can apply the compression technique further. To elaborate, we not only can compress the frequent patterns of the same frequency (e.g., sup=72%) but can also compress those frequent patterns with the same frequency range (e.g., $sup \in [70\%, 75\%]$). The lesser the allocated space, the broader would be the frequency range to be used in compression.

Besides compressing/merging several horizontal lines (which represent several frequent patterns) into a single line, FpViz can also provide users with some alternative options. For example, we can allow users to pick an option to display and visualize only those *closed frequent patterns* or *maximal frequent patterns* (instead of showing all frequent patterns). A frequent pattern X is *closed* if there does not exist any proper superset of X having the *same frequency as* X, and a frequent pattern Y is *maximal* if there does not exist any proper superset of Y that is also *frequent*. By visualizing only closed or maximal frequent patterns, FpViz can reduce the number of horizontal lines that need to be shown.

How to represent frequent patterns comprising a large number of domain items? While the above dealt with the scalability issue for the y-direction (i.e., scalable for large range of frequencies), we discuss here the scalability issue for the x-direction. More specifically, what if the number of items exceeds the number of pixels allocated for the x-direction? The challenge is that the y-direction contains frequency information (which are numerical data) but the x-direction contains items (which are categorical data). In FpViz, we can apply the following two techniques. First, if items are arranged in decreasing frequency order, then we can group several adjacent items together into a "mega"-item. For example, "mega"-item$_1$ represents all the domain items having the top 10 frequencies, "mega"-item$_2$ represents all the domain items having the next 10 highest frequencies, etc. Alternatively, we can group items according to some hierarchy or taxonomy. For example, we could group items "apples", "bananas" & "cherries" into a mega-item "fruits". Similarly, we could also group "donuts" & "egg-tarts" into a mega-item "snack".

4. INTERACTIVE FEATURES OF FpViz

The above representation of FpViz allows users to get an insight about the overview distribution of raw data and the analysis results (in the default collapsed view). It also provides users with some relevant details (in the expanded view). See Figure 3 for snapshots (of these two views). In



(a) FIsViz [26] **(b) Our proposed FpViz**

Figure 4: Two visualizers showing the same set of frequent patterns mined from a student-course DB.

this section, we describe some additional interactive features of FpViz. While these features are not essential, they provide user convenience.

QUERY ON FREQUENCY. With FpViz, users can easily find all *frequent items* and/or *frequent patterns* (i.e., with frequencies exceeding the user-specified minimum frequency threshold *minsup*) by ignoring everything that lies below the "threshold line" y=*minsup* (i.e., ignoring the lower portion of the graph). To a further extent, the representation of frequent patterns in FpViz leads to effective *interactive mining*. To elaborate, with FpViz, users can see what (and how many) frequent patterns are above a certain frequency. Based on this information, users can freely adjust *minsup* by moving the slider (see the green bar on the left side in Figure 4(b))—which controls *minsup*—up and down along the y-axis to find an appropriate value for *minsup*. Moreover, FpViz also provides two related features:

(i) It allows users to interactively adjust *minsup* and automatically counts the number of patterns that satisfy *minsup*. By doing so, users can easily find TOP-N FREQUENT PATTERNS.

(ii) It also allows users to pose a RANGE QUERY ON FREQUENCY (by specifying both minimum and maximum frequency thresholds *minsup* and *maxsup*) and then shows all patterns with frequencies falling within the range [*minsup*, *maxsup*].

QUERY ON CARDINALITY. FpViz allows the user to pose a query on cardinality, and it only shows frequent patterns of the user-specified cardinality k. Moreover, FpViz also allows users to pose a RANGE QUERY ON CARDINALITY so that only those frequent patterns with cardinality k within the user-specified range $[k_{min}, k_{max}]$ are drawn.

QUERY ON FREQUENT PATTERNS. FpViz also allows users to interactively select certain items of interest (e.g., promotional items in a store) and to pose queries on frequent patterns. Examples of these queries include the following: (i) "Find all frequent patterns containing *some* of selected items"; (ii) "Find all frequent patterns containing at least *all* of the selected items"; and (iii) "Find all frequent patterns *not* containing any of the selected items".

QUERY ON RELATIONSHIPS AMONG FREQUENT PATTERNS. Recall that users can easily find the prefixes or extensions of a frequent pattern that share the same frequencies. However, sometimes prefixes or extensions (or more general case, subsets or supersets) of a frequent pattern X may have different frequencies than that of X. Hence, FpViz provides interactive features to highlight these subsets or supersets (prefixes or extensions) of a pattern of user interests.

DETAILS-ON-DEMAND. Details-on-demand consists of techniques that provide more details whenever the user requests them. The key idea is that FpViz gives users an overview of the entire dataset and then allows users to interactively select parts of the overview for which they request more details—by hovering the mouse over different parts of the display. Specifically, FpViz supports details-on-demand in the following ways:

(i) When *the mouse hovers on a segment of a horizontal line* connecting two nodes (say, representing frequent patterns x and y), FpViz shows a list of frequent patterns containing both x and y. Selecting a frequent pattern in the list instantly highlights the specific segment it is contained in, as well as both of its connecting nodes, so that users can see where the segment starts and ends.

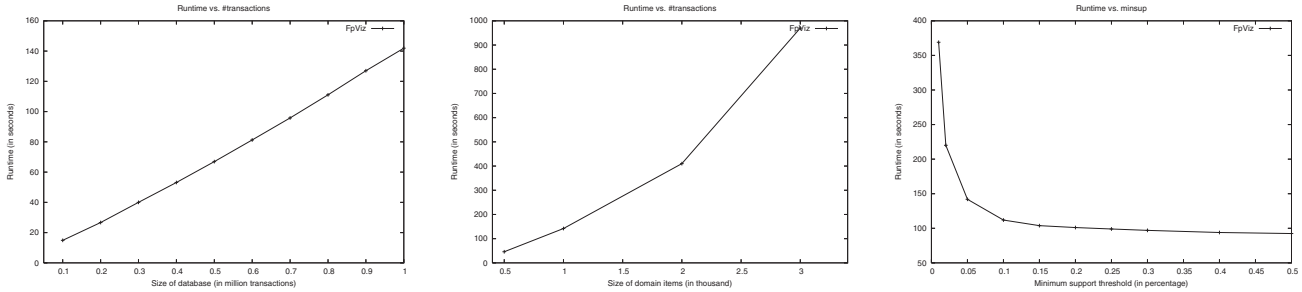
(ii) When the *mouse hovers over a node*, FpViz shows a list of all frequent patterns contained in all the line segments starting or ending at this node. Selecting a frequent pattern from the list instantly highlights the line it is contained in.

(iii) When the *mouse hovers over a pixel* in the display (even if it is not part of the graph), a small box appears showing the frequency and frequent patterns encoded by the mouse position. This is particularly useful when users need to see among the vast array of line segments what a particular point in the display refers to.

GUARANTEED VISIBILITY. Our proposed FpViz allows users to specify his preference on visualization of frequent patterns. For example, if users are interested in finding those patterns containing fruits, FpViz ensures that all corresponding horizontal lines are clearly visible.

5. EVALUATION RESULTS

In this section, we show our results on evaluating the proposed FpViz. Here, we conducted four sets of evaluation



(a) Runtime vs. sizeof(*TDB*) **(b) Runtime vs. #domain items** **(c) Runtime vs. *minsup***

Figure 5: Performance of FpViz.

tests. In the first set, we tested functionality of our FpViz by showing how it can be applicable in various scenarios or real-life applications. In the second set, we tested performance of our FpViz. In the last two sets, we evaluated readability and interactivity of FpViz.

5.1 Evaluating the Functionality of FpViz

In the first set of evaluation tests, we compared our proposed FpViz with FIsViz [26]. We considered many different real-life scenarios. For each scenario, we determined whether these systems can handle the scenarios. If so, we examined how these systems display the mined results. The evaluation results show that our FpViz was as effective as FIsViz in all these scenarios. A few samples of these scenarios are shown below.

Q1(a) What kinds of vegetables are frequently purchased by customers? Frequently purchased vegetables are patterns with high frequency. With FIsViz, as polylines representing itemsets may cross each other, users may *not* be able to easily see the itemsets of high frequency if they are in the dense or clustered area of the display. In contrast, our FpViz shows all frequent patterns by horizontal lines, which are easily visible and never cross any other horizontal lines. Let us compare Figure 4(a) with Figure 4(b). The former was a snapshot of FIsViz and the latter was a snapshot of FpViz. These two snapshots both show the same set of frequent patterns.

Q1(b) How frequently are these vegetables purchased individually and how frequently are purchased together? Depending on the density of the display in FIsViz, the frequencies of some itemsets may *not* be too visible if they are in the dense or clustered areas of the display. In contrast, users can easily obtain the frequencies of patterns from our FpViz because there is no line crossing.

Q1(c) What kinds of vegetables that are frequently purchased together with eggplants? Both FIsViz and FpViz provide users with a feature of handling queries on frequent patterns containing some specific items (in this scenario, eggplants).

We observed from all scenarios (including the above three samples) that our proposed FpViz either retained the existing features of FIsViz (e.g., for Q1(c)) or provided additional improvements over FIsViz (e.g., for Q1(a) & (b)).

5.2 Evaluating the Performance of FpViz

In the performance test, we used (i) several IBM synthetic datasets [1], (ii) some real-life databases (e.g., mushroom dataset) from UC Irvine Machine Learning Depository, (iii) some CNN documents, and (iv) a student-course database for our university. The results produced are consistent. In the first experiment, we varied the size of the database *TDB*. We measured the time both for mining frequent patterns (by using FP-growth [11]) and for constructing the display layout (by using our proposed FpViz). The results showed that the runtime (which includes CPU and I/Os) increased linearly with the number of transactions in the database. See Figure 5(a). In the second experiment, we varied the number of items in the domain. The results showed that the runtime increased when the number of domain items increased. See Figure 5(b). In the third experiment, we varied the user-defined frequency threshold. When the threshold increased, the number of patterns that satisfy the threshold (i.e., frequent patterns to be displayed) decreased, which in turn led to a decrease in runtime. See Figure 5(c).

5.3 Evaluating the Readability of the Mined Results Shown by FpViz

To assess the effectiveness of conveying frequent pattern relationships, we carried out a user evaluation with FpViz. The evaluation was primarily case-based, within which several types of users were required to solve many different questions based on the visualizations of a given dataset (e.g., a database containing information about courses taken by students (see Figure 4(b)). Therefore, the scenario was that users need to identify a set of relationships and make decisions based on their observations.

We recruited 24 participants and separated them into two groups: (i) those who have data mining background and (ii) those who do not. None of the participants (regardless which of the two groups) was exposed to any form of visualization for frequent patterns—including our proposed FpViz.

To test the expressiveness of our visualization, we formulated two types of questions: multiple choices and those open-end ones that require participants to perform some level of analytical reasoning with the visualization. Sample questions include the following:

1. Which course is most frequently taken (i.e., course with highest enrolment)?
2. Which course is the next/second most frequently taken (i.e., course with second highest enrolment)?
3. Which two courses are most frequently taken?
4. What course is least frequently taken (i.e., course with lowest enrolment)?
5. What three courses are taken together by exactly four students?
6. How many students are taking COMP 4350 together with 4380?
7. What three courses are taken together by four students?
8. How would you use this chart to make any changes in terms of how 3rd and 4th year courses are offered and/or distributed?
9. If you were to reduce the offerings of 4th year courses, which ones would you select? and why (i.e., how did the diagrams lead to your conclusions)?
10. If you were to schedule exams of these courses, which pair of courses would you avoid scheduling on the same day?

We began the evaluation by presenting our FpViz and asking the participants to explore it at their own will. We did not give them any information regarding what the symbols and representations meant in the visualization. We first questioned them on what they were able to identify. Evaluation results showed that all the participants were able to identify the basic meaning behind the representations (e.g., that frequency was assigned to the y-axis, and courses to the x-axis). Participants were also able to identify the most frequently taken courses, without having us to tell them the answer.

Afterwards, we gave the participants detailed information on how to read the graphs and what the various lines and circles meant. This information was then followed by a set of questions that queried into the participants' ability to simply read the graph, with a set of close-ended questions. Evaluation results showed that a majority of the participants were able to correctly answer most—and some even correctly answer all—of the questions. Statistically, the average accuracy rate was above 83% and three participants obtained an accuracy rate of 100%.

Several interesting observations were found at the post-evaluation interview. For example, the participants told us that, while we gave the participants a multiple choice list to help them identify the answers, they did not use the multiple choice questions to guide their selection. Instead, their curiosity in understanding the visualizations led them to answer the questions by looking at the graphs first, and then confirming their answer with one of the choices provided. Moreover, the results were similar in both participant groups (with or without data mining background). This shows that the easy readability of FpViz. This also suggested that, with very little training, participants felt comfortable to use the visualizations. Furthermore, they were able to quickly assimilate the representations, to the point at which they were able to answer all questions adequately.

Finally, we asked the participants a set of open-ended questions. Each participant completed the evaluation separately (i.e., no discussion among the participants). The results showed that the participants were able to make the best use of visualization for answering these questions.

5.4 Evaluating the Benefits of the Interactive Features Provided by FpViz

We also evaluated the effectiveness of the interactivity of our FpViz on various datasets mentioned above. We divided a set of 24 participants into two groups. The first group performed some tasks using *only* the essential features (i.e., without using any interactive feature described in Section 4), and then used both the essential and the interactive features to answer some similar but not identical tasks. The second group did so in the reverse order (i.e., with interactive features and then without interactive features). By so doing, we avoid measuring the unwanted effect of learning (i.e., participants may learn from the first set of tasks). We observed the following from the results: (i) The participants were able to correctly answer all the questions using the interactive features. (ii) Most participants were able to do so without using the interactive features, but required much longer time. On average, participants took about 5 minutes to complete all the questions when using the interactive features, but took longer than 12 minutes to complete all the questions when not using any interactive feature. This indicated that, while interactive features are not essential, they provide convenience to users (and saved their time). Hence, it is beneficial to use the interactive features provided by FpViz.

6. CONCLUSIONS

Many frequent pattern mining algorithms return a collection of the data analysis results in the form of a textual list of frequent patterns. This list can be very long and difficult to comprehend. Since "a picture is worth a thousand words", it is desirable to have visualization systems. However, many existing visualization systems were not designed to show frequent patterns. To improve this situation, we proposed and developed a powerful *frequent pattern visualizer* called *FpViz*, which provides users with explicit and easily-visible information among the frequent patterns. Specifically, FpViz represents frequent patterns as horizontal lines in a two-dimensional graph. If multiple patterns have the same frequency, the corresponding lines representing these patterns are collapsed into one line. With this compression technique, FpViz allows the user to expand some portion (or all) of the "collapsed" mining results for data or result exploration. Moreover, FpViz also provides users with additional nice interactive features. Evaluation results showed the effectiveness of FpViz in terms of functionality, performance, readability, and interactivity. FpViz provides users with visual support for visual analytics as well as knowledge discovery and data mining (KDD).

7. ACKNOWLEDGMENTS

This project is partially supported by Natural Sciences and Engineering Research Council of Canada (NSERC) in the form of research grants.

8. REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proc. VLDB 1994*, pp. 487–499.

[2] C. Ahlberg. Spotfire: an information exploration environment. *SIGMOD Record*, **25**(4), pp. 25–29, 1996.

[3] M. Ankerst et al. Visual classification: an interactive approach to decision tree construction. In *Proc. KDD 1999*, pp. 392–396.

[4] P. Appan et al. Summarization and visualization of communication patterns in a large-scale social network. In *Proc. PAKDD 2006*, pp. 371–379.

[5] S. Berchtold et al. Independence diagrams: a technique for visual data mining. In *Proc. KDD 1998*, pp. 139–143.

[6] J. Blanchard et al. Interactive visual exploration of association rules with rule-focusing methodology. *KAIS*, **13**(1), pp. 43–75, 2007.

[7] C. Brunk et al. MineSet: an integrated system for data mining. In *Proc. KDD 1997*, pp. 135–138.

[8] S.-M. Chan et al. Maintaining interactivity while exploring massive time series. In *Proc. IEEE VAST 2008*, pp. 59–66.

[9] C.H. Chih and D.S. Parker. The persuasive phase of visualization. In *Proc. KDD 2008*, pp. 884–892.

[10] J. Han and N. Cercone. RuleViz: a model for visualizing knowledge discovery process. In *Proc. KDD 2000*, pp. 244–253.

[11] J. Han et al. Mining frequent patterns without candidate generation: a frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, **8**(1), pp. 53–87, 2004.

[12] H. Hofmann et al. Visualizing association rules with interactive mosaic plots. In *Proc. KDD 2000*, pp. 227-235.

[13] T. Iwata et al. Probabilistic latent semantic visualization: topic model for visualizing documents. In *Proc. KDD 2008*, pp. 363–371.

[14] D.A. Keim. Information visualization and visual data mining. *IEEE TVCG*, **8**(1), pp. 1–8, 2002.

[15] D.A. Keim and H.-P. Kriegel. Visualization techniques for mining large databases: a comparison. *IEEE TKDE*, **8**(6), pp. 923–938, 1996.

[16] D.A. Keim and D. Oelke. Literature fingerprinting: a new method for visual literary analysis. In *Proc. IEEE VAST 2007*, pp. 115–122.

[17] D.A. Keim and J. Schneidewind (eds.). Special issue on visual analytics. *SIGKDD Explorations*, **9**(2), 2007.

[18] D.A. Keim et al. Monitoring network traffic with radial traffic analyzer. In *Proc. IEEE VAST 2006*, pp. 123–128.

[19] Y. Koren and D. Harel. A two-way visualization method for clustered data. In *Proc. KDD 2003*, pp. 589–594.

[20] L.V.S. Lakshmanan, C.K.-S. Leung, and R.T. Ng. Efficient dynamic mining of constrained frequent sets. *ACM TODS*, **28**(4), pp. 337–389, 2003.

[21] H. Lam et al. Session viewer: visual exploratory analysis of web session logs. In *Proc. IEEE VAST 2007*, pp. 147–154.

[22] C. K.-S. Leung. Frequent itemset mining with constraints. To appear in *Encyclopedia of Database Systems*, Springer, 2009.

[23] C.K.-S. Leung and B. Hao. Mining of frequent itemsets from streams of uncertain data. In *Proc. IEEE ICDE 2009*, pp. 1663–1670.

[24] C.K.-S. Leung et al. A tree-based approach for frequent pattern mining from uncertain data. In *Proc. PAKDD 2008*, pp. 653–661.

[25] C.K.-S. Leung et al. CanTree: a tree structure for efficient incremental mining of frequent patterns. In *Proc. IEEE ICDM 2005*, pp. 274–281

[26] C.K.-S. Leung et al. FIsViz: a frequent itemset visualizer. In *Proc. PAKDD 2008*, pp. 644–652.

[27] C.K.-S. Leung et al. WiFIsViz: effective visualization of frequent itemsets. In *Proc. IEEE ICDM 2008*, pp. 875–880.

[28] T. Munzner et al. Visual mining of power sets with large alphabets. Technical report UBC CS TR-2005-25, Dept. of Computer Science, UBC, Canada, 2005.

[29] D. Oelke et al. Visual evaluation of text features for document summarization and analysis. In *Proc. IEEE VAST 2008*, pp. 75–82.

[30] G. Pölzlbauer et al. A vector field visualization technique for self-organizing maps. In *Proc. PAKDD 2005*, pp. 399–409.

[31] H.C. Purchase et al. Validating graph drawing aesthetics. In *Proc. GD 1995*, pp. 435–446.

[32] J. Scholtz. Beyond usability: evaluation aspects of visual analytic environments. In *Proc. IEEE VAST 2006*, pp. 145–150.

[33] T. Schreck et al. Visual cluster analysis of trajectory data with interactive Kohonen Maps. In *Proc. IEEE VAST 2008*, pp. 3–10.

[34] R. Spence. *Information Visualization: Design for Interaction - 2e*. Prentice Hall, 2007.

[35] C. Stolte et al. Query, analysis, and visualization of hierarchically structured data using Polaris. In *Proc. KDD 2002*, pp. 112–122.

[36] C. Ware et al. Cognitive measurements of graph aesthetics. *Information Visualization*, **1**(2), pp. 103-110, 2002.

[37] P.C. Wong and J. Thomas. Visual analytics. *IEEE CG&A*, **24**(5), pp. 20–21, 2004.

[38] L. Yang. Pruning and visualizing generalized association rules in parallel coordinates. *IEEE TKDE*, **17**(1), pp. 60–70, 2005.

[39] X. Yang et al. A visual-analytic toolkit for dynamic interaction graphs. In *Proc. KDD 2008*, pp. 1016–1024.

[40] J. Yuan et al. From frequent itemsets to semantically meaningful visual patterns. In *Proc. KDD 2007*, pp. 864–873.

Multiple Coordinated Views Supporting Visual Analytics

Bianchi Serique Meiguins
Universidade Federal do Pará
Belém – Pará - Brazil
bianchi@ufpa.br

Aruanda Simões Gonçalves Meiguins
Centro Universitário do Pará
Belém – Pará - Brazil
aruanda@redeinformatica.com.br

ABSTRACT
This paper proposes the use of multiple coordinated views to support visual analytics. The Information Visualization tool PRISMA includes the implementation of three coordinated visualization techniques (treemap, parallel coordinates and scatterplot) and other auxiliary charts. The coordination is supported by filter, brushing, visual representation and other mechanisms. The mini –challenge 4 (Evacuation Traces) from IEEE VAST 2008 Challenge was used as a case study on this paper.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *Interaction styles (e.g., commands, menus, forms, direct manipulation).*

General Terms
Design, Human Factors

Keywords
Information Visualization, Multiple Coordinated Views.

1. INTRODUCTION
Information Visualization (IV) aims to analyze data quickly, interactively and intuitively. IV takes advantage of human cognitive capacity to extract information from data using visual representations. Datasets grow in volume and diversity constantly – or sometimes exponentially. IV techniques are therefore challenged to cope with larger and larger datasets in terms of data representation, interaction and performance. Multiple coordinated views have been more frequently supported in IV tools since a single view of the dataset may be unable to identify potentially interesting relationships.

An Information Visualization tool should minimally support the following features: overview, zoom, filters and details-on-demand [1] [2] [3].

Additionally, the tool should provide interactive mechanisms that allow the user to easily and efficiently manipulate graphical representations of the data in order to better understand characteristics and relationships of the dataset. [3] [4] [5].

Multiple coordinated views, when not excessive in number, may considerably improve the quality of user perceptions of a given dataset. This approach allows the user to perform correlation of different views of the same dataset [9].

Coordination ensures that changes made in one data view are propagated to all other views in order to keep the analyzed data consistent between the views. Coordination options include data filters, visual attributes and sorting criteria, among other mechanisms [10].

2. PRISMA
PRISMA is an information visualization tool based on multiple coordinated views to explore multidimensional datasets using treemap, scatterplot and parallel coordinates as its main interactive techniques [6].

The main characteristics of PRISMA are:

- PRISMA is an extensible, portable and easy to maintain Java-based tool.
- The graphic interface is automatically customized to data types and to the range of data values in each dataset.
- The filter components are adapted to the characteristics of each dataset.
- Coordination is supported in filter, color, shape, size and details-on-demand components.
- Provides support to many different data sources, such as relational databases, XML files and pre-formatted text files.
- Pie, bar and line charts and automatically-generated reports are additional coordinated features.

The tool implements three information visualization techniques, each favoring a different kind of data analysis: parallel coordinates, treemap and scatterplot [7] [8]. PRISMA allows the user to analyze data either individual or simultaneously in all views.

PRISMA stores user interaction history including manipulation and configuration data and allows the user to save the state of an explored visualization.

VAKD'09, June 28, 2009, Paris, France.

3. A CASE STUDY: EVACUATION TRACES

The mini-challenge "Evacuation Traces" from IEEE VAST 2008 Challenge [11] describes the evacuation of employees and visitors from a hospital building after a bomb explosion. Everyone in the building wore a RFID badge that enabled their location to be recorded during the time of the incident. In order to help the police department during the investigations, five main questions should be answered:

- Where was the device set off?
- Identify potential suspects and/or witnesses to the event.
- Identify any suspects and/or witnesses who managed to escape the building.
- Identify any casualties.
- Describe the evacuation.

The dataset included spatial information related to the movement of each person in the building and a representation of the building itself (solid and open space mappings).

The following sections will answer the questions indicated in questions using PRISMA screenshots to support each analysis.

3.1 The explosion Time Question

The time the bomb was set off was not included among the questions. It may be identified by the simultaneous movement of people in the building, indicating panic. As presented in Figure 1, up to time 373 very few persons were moving, including the following Person IDs : 29 (red), 56 (blue), 44 (green), 21(black) and 28 (orange).

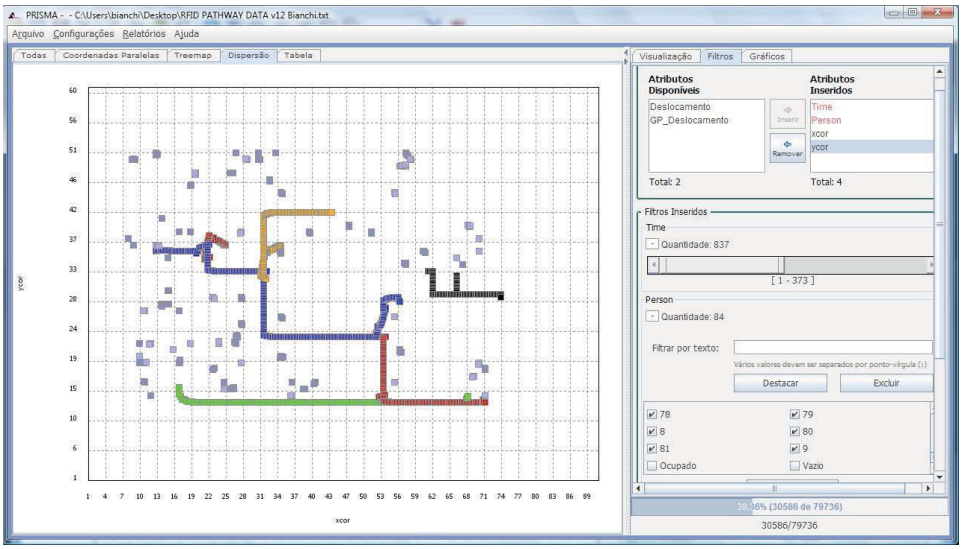


Figure 1. Movement before time 373

After the moment represented in Figure 1, many people moved simultaneously, indicating this was the time the bomb was set off (Figure 2)

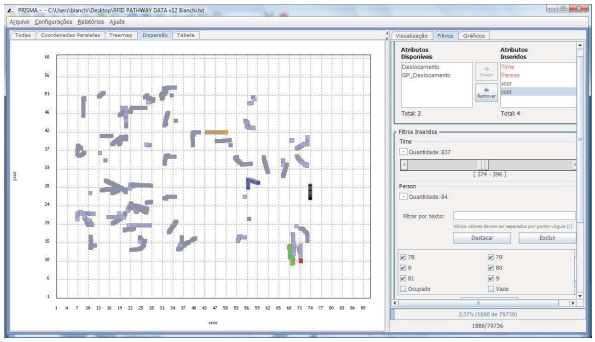


Figure 2. Movement after time 374

3.2 The Casualties Question

In order to identify potential casualties we highlighted the persons who stopped moving soon after the explosion or did not move too far. The total distance between the initial and final points was calculated for each person and included as one of the dataset attributes. Two distance groups were then assigned: less than 10, and 10 or more distance units.

The parallel coordinate technique was used to indentify individuals with the least absolute distance values. Alternatively, the treemap technique was used to select only the items that moved less than 10 distance units. The brushing technique then allowed the analysis of three different groups on the scatterplot view.

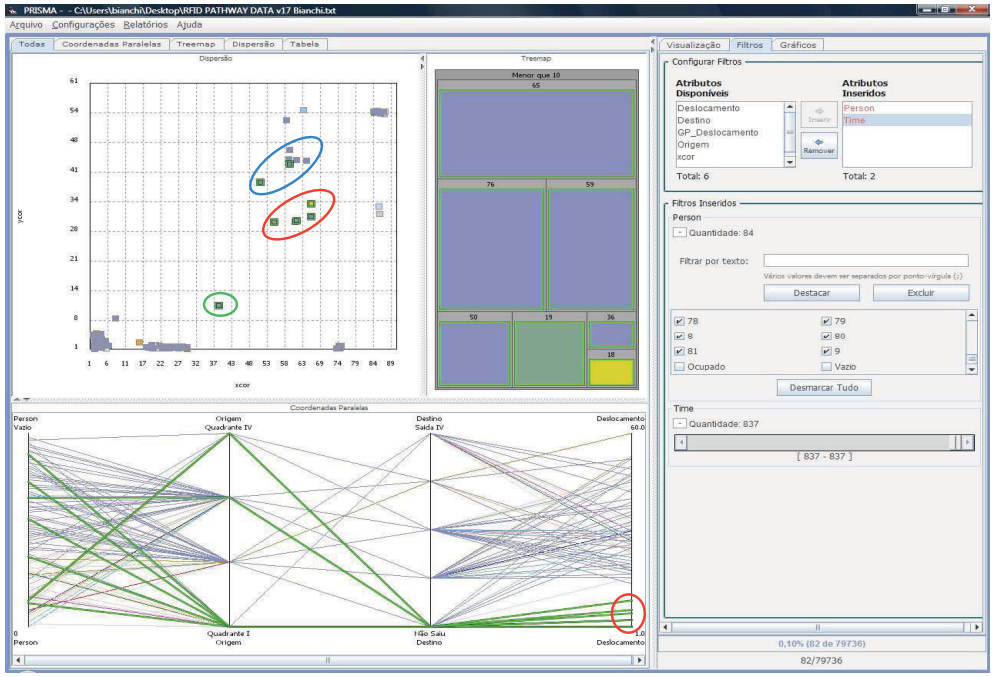


Figure 3. Three distance groups

Analyzing the blue group (Person IDs 65 and 36), the first moved up to time 600, a considerable time after the explosion. Person 36 does not move before or after the explosion. In the green group, person 59 moves during the whole period of captured information and the total distance is not very large returning to a point close to the initial point. The red group (19, 76, 5 and 18) moves very little after the explosion. When we analyzed the red group area, an additional person was identified: number 56 (yellow item in Figure 4). This person moved a long way before the explosion and was next to the red group on the moment the bomb was set off.

Therefore possible casualties are represented by Person IDs 18, 19, 50, 56 and 76.

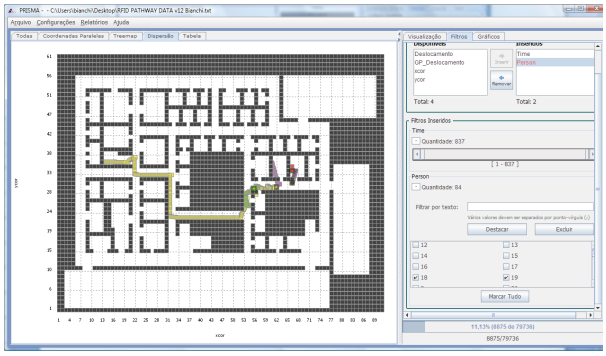


Figure 4. Possible Casualties

3.3 The Suspect Question

Potential suspects or witnesses are those who move next to the casualties location before the explosion and left this perimeter quickly afterwards. Figure 5 presents the preliminary suspects: 29 (purple), 56 (among the casualties), 44 (yellow), 21 (red) and 28 (pink). Person 21 was the only one that had been to the probable bomb location area previously to the explosion. The probable bomb location area is where people moved very little after the explosion and is highlighted in red in Figures 3 and 5. After leaving this area, Person 21 seems to be unable to find the way out. This person therefore presents the most suspect behavior.

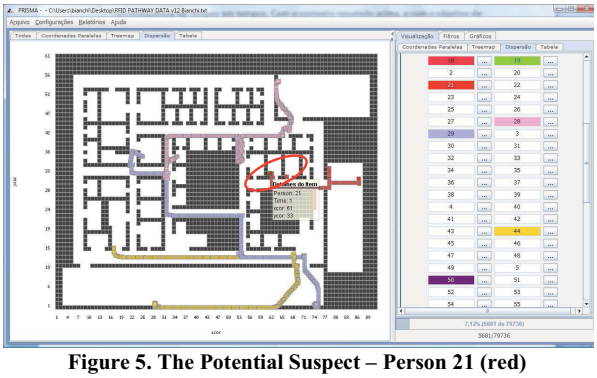


Figure 5. The Potential Suspect – Person 21 (red)

The potential witnesses list included Person IDs 80, 28, and 1. Person ID 1 (blue) was probably the main witness, since he/she crossed the main suspect (21 – red). Person ID 80 (yellow) was in the room next to the victims and Person ID 28 (purple) moved

next to the casualties' area (in white) and came back. The witness behavior was filtered in Figure 6.

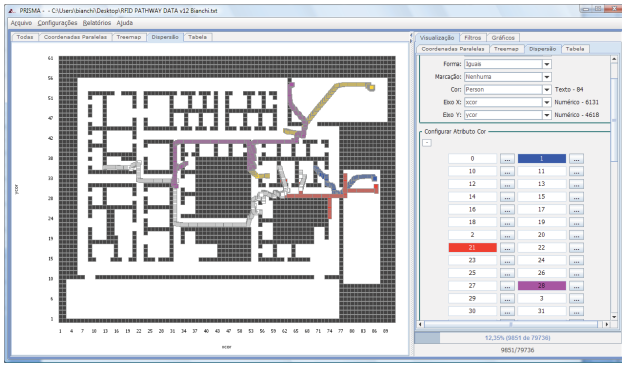


Figure 6. Main Witnesses

Therefore, the main suspect is 21 and the main witnesses are 1, 80 and 28.

3.4 The Bomb Location Question

Considering as the main suspect Person ID 21 and that the victim that moved the less after the explosion was Person ID 18 we assumed the suspect entered the room where Person 18 was and set the bomb off on that location. The coordinates are 61x33, the farthest position of the suspect into the room.

3.5 The Escaping Witnesses Question

The main exit locations are highlighted by a red circle in Figure 7, which also included the complete movement of each person in the building. Neither suspect 21 nor witness 1 have passed these locations so only witnesses 28 and 80 did leave the building.

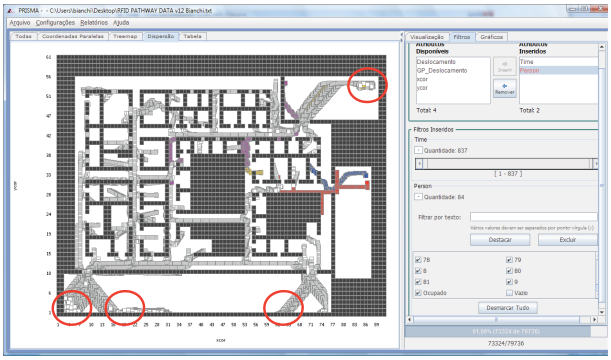


Figure 7. Escaping Witnesses

4. DESCRIBING THE EVACUATION

Using coordination between the visualization techniques, filter and brushing, it was possible to describe the evacuation process.

Additional information was generated for a better analysis of the building evacuation process. One of the generated attribute was the original quadrant occupied by each person (origin attribute). The building was split into four sub-areas as illustrated in Figure 8. Another generated attribute was the destination, which indicated the exit used by each person to leave the building. The four possible building exits were identified by the number of people that eventually moved to the same position.

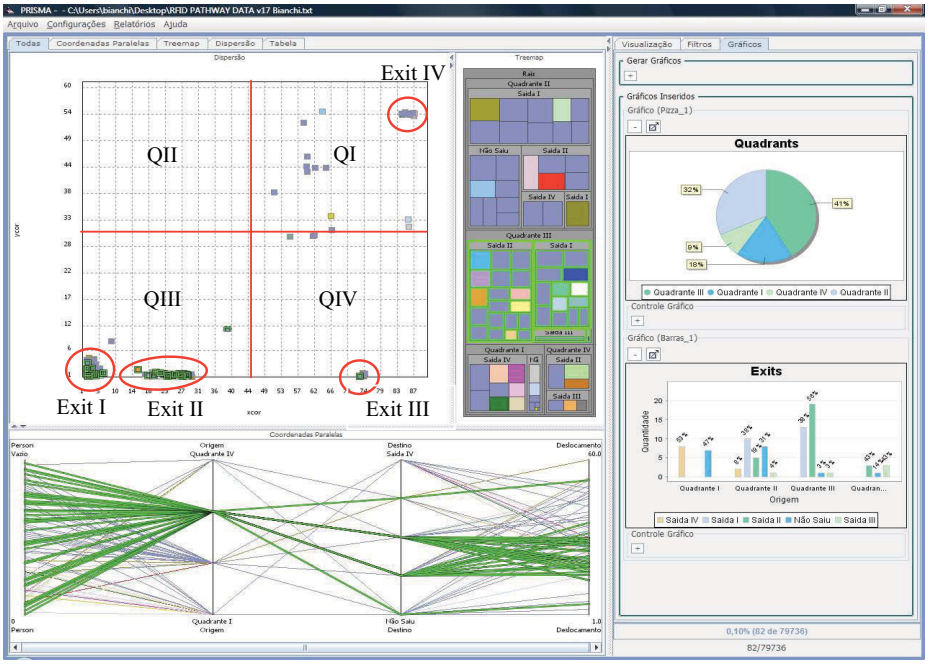


Figure 8. Use of coordination to analyze the evacuation routes.

Using the additional information, it was possible to realize, for example, that 96% of those originally in quadrant III used exits I and II and more than 50% of those in quadrant II used these same exits, with absolute distance from medium to low.

The evacuation was concentrated on the south and west areas of the building (Figure 9). The two main escape routes were the halls identified by green arrows in Figure 9. At some point the right corridor was crowded and some people changed directions and moved to the adjacent hall on the left. People located at the center of the building followed the evacuation route indicated by the blue arrow. People on the north of the building were directed to a northeast exit represented by the red arrow.

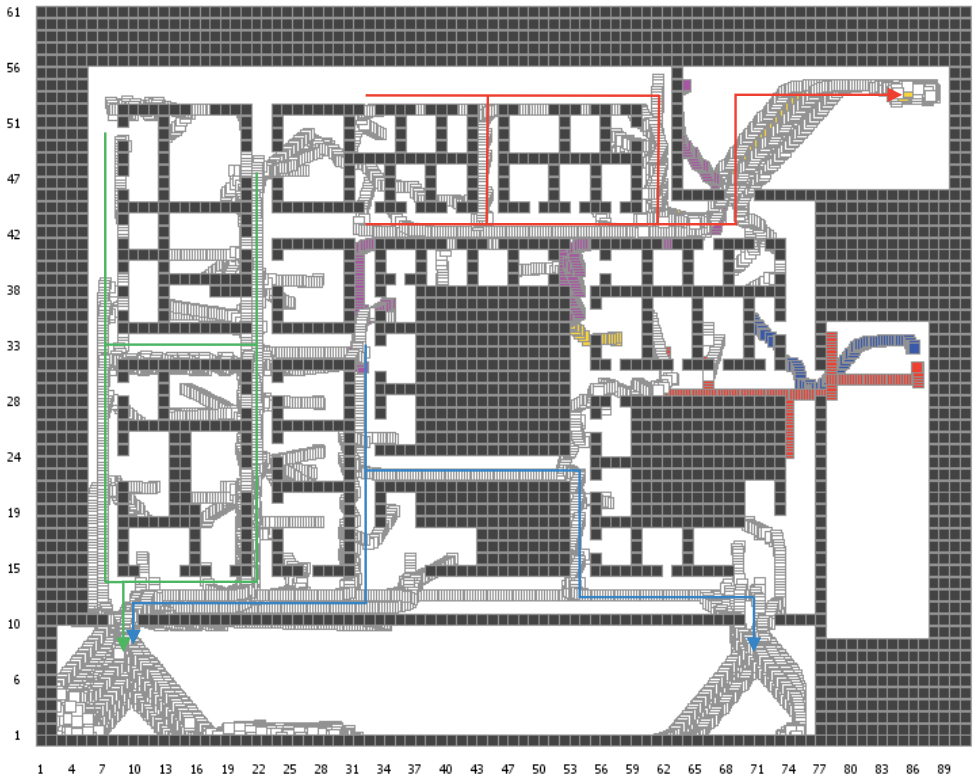


Figure 9. Evacuation Routes

5. FINAL REMARKS

The use of PRISMA to solve mini-challenge 4 focused on the scatterplot technique. It was necessary to perform some preprocessing and to create some new columns in original dataset. The main new information for the analysis was the distance calculation which helped to use the main coordination features of the tool.

The task was facilitated fundamentally by coordination, the use of filter and color mechanisms. The use of the interval filter to analyze temporal information was also very important. It was possible to analyze simple animations which were very helpful to understand people behavior and the evacuation routes.

As future work we believe that the coordination features may be enhanced by the development of new visualization techniques.

6. REFERENCES

[1] Berry, B.; Smith, J.; Wahid, S. Visualizing Case Studies. Technical Report TR-04-12, Virginia Tech, 2003.

[2] Carr, D. A. Guidelines for Designing Information Visualization Applications. Proceedings of ECUE'99. Stockholm, Sweden. December 1999.

[3] Keim, D. A. Information Visualization and Visual Data Mining. IEEE Transactions On Visualization And Computer Graphics, January-March 2002.

[4] Oliveira, M. C. F; Levkowitz, H. From Visual Data Exploration to Visual Data Mining: A Survey. IEEE Transactions on Visualization and Computer Graphics, vol. 9, no. 3, pp. 378-393, July-September 2003.

[5] Spence, R. Information Visualization: Design for Interaction. Barcelona: Acm Press. Second Edition, 2007.

[6] Godinho, I; Meiguins, B.;Gonçalves, A.; Carmo, C.; Garcia, M.;Almeida, L.; Lourenço, R. PRISMA – A Multidimensional Information Visualization Tool Using Multiple Coordinated Views. Proceedings of the 11th International Conference on Information Visualization, pp. 23-32. Zurich, 2007.

[7] Spence, R. Information Visualization. Addison Wesley - ACM Press, 2001. 459 p.

[8] Card, S. K.; Mackinlay, J. D.; Shneiderman, B. Readings in Information Visualization—Using Vision to Think. Morgan Kaufmann, 1999.

[9] Baldonado, M. Q. W.; Woodruff, A.; Kuchinsky, A. Guidelines for using multiple views in information visualization. Proceedings of the working conference on Advanced Visual Interfaces, pp. 110 – 119. Palermo. Italy. 2000.

[10] Pillat, R. M.; Freitas, C. D. S. Coordinating Views in the InfoVis Toolkit. Proceedings of Advanced Visual Interface. pp. 496-499. Venezia, Italy. 2006.

[11] Grinstein, G.; Plaisant, C.; O'connell, T.; Laskowski, S.; Scholtz, J.; Whiting, M. VAST 2008 Challenge: Introducing Mini-Challenges. Proceedings of IEEE Symposium on Visual Analytics Science and Technology (2008).

Exploration and Visualization of OLAP Cubes with Statistical Tests

Carlos Ordonez
University of Houston
Dept. of Computer Science
Houston, TX 77204, USA

Zhibo Chen
University of Houston
Dept. of Computer Science
Houston, TX 77204, USA

ABSTRACT

In On-Line Analytical Processing (OLAP), users explore a database cube with roll-up and drill-down operations in order to find interesting results. Most approaches rely on simple aggregations and value comparisons in order to validate findings. In this work, we propose to combine OLAP dimension lattice traversal and statistical tests to discover significant metric differences between highly similar groups. A parametric statistical test allows pair-wise comparison of neighboring cells in cuboids, providing statistical evidence about the validity of findings. We introduce a two-dimensional checkerboard visualization of the cube that allows interactive exploration to understand significant measure differences between two cuboids differing in one dimension along with associated image data. Our system is tightly integrated into a relational DBMS, by dynamically generating SQL code, which incorporates several optimizations to efficiently explore the cube, to visualize discovered cell pairs and to view associated images. We present an experimental evaluation with medical data sets focusing on finding significant relationships between risk factors and disease.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*; I.3.6 [**Computer Graphics**]: Methodology and Techniques—*Interaction techniques*

General Terms

Algorithms, Human Factors, Performance

Keywords

Parametric test, cube, visualization

1. INTRODUCTION

In a modern data mining environment users have a wide spectrum of options to analyze a data set going from simple queries to building machine learning and statistical models. On-Line Analytical Processing (OLAP) [5, 11] is an important application of exploratory database analysis that is complementary to such approaches. In an OLAP database users generally explore a large fact table with aggregations performed at multiple granularity levels trying to find interesting results. In OLAP most computations return simple univariate statistics such as sums, row counts, means and standard deviations. On the other hand, data mining [7, 11], statistical [13] and machine learning [17] techniques generally build models of varied complexity on a data set, depending on problem requirements. A comprehensive family of statistical tests sit somewhere in the middle between univariate statistical analysis and complex statistical models. Our tool shows parametric statistical tests are a promising technique to explore OLAP cubes.

Statistical tests [22] exhibit several advantages over statistical and machine learning models. They have simple and weak assumptions about the probability distribution behind the data set. In our case, such distribution generally comes in the form of a normal (Gaussian) distribution, or a closely related probability distribution function. Statistical tests use mathematically simple equations that can be efficiently evaluated with SQL queries because they generally do not require vector or matrix manipulation. Statistical tests can produce statistically reliable results with both large data sets and small data sets, whereas many data mining and machine learning techniques require large data sets in order to find significant results. It is important to notice that small data sets may appear even when working with large databases, due to analyzing a database at coarse aggregation levels (e.g. grouping by store) or when the distribution behind some selection attribute is skewed (e.g. zipf). On the other hand, compared to standard exploratory OLAP analysis, statistical tests provide more evidence that a finding is indeed significant, going beyond simple comparisons or getting variance proportions. Nevertheless, statistical tests generally require many trial and error runs before a plausible finding is made and each run requires varying parameters or selecting subsets of the data set. With such motivation in mind, our tool automates the process of exploring cuboids from a low dimensional cube trying to find significant measure differences supported by statistical tests. Since the lattice behind the cube dimensions represents a combinatorial search space the problem is computationally challenging; several optimizations are incorporated to make the exhaustive comparison process faster. Our current application is in medical databases.

VAKD'09, June 28, Paris, France.

This is an outline of the rest of the article. Section 2 introduces basic definitions for OLAP databases and statistical tests. Section 3 explains how our tool automatically applies statistical tests on all cuboids from a cube. Section 4 presents an experimental evaluation of visualization and database optimizations with medical data sets. Related work is discussed in Section 5. The conclusions are presented in Section 6.

2. DEFINITIONS

Let F be a fact table with n records having d cube dimensions [11], $D = \{D_1, \ldots, D_d\}$, a set of e measure [11] attributes $A = \{A_1, A_2, \ldots, A_e\}$ and an additional set of f image attributes $I = \{I_1, \ldots, I_f\}$. This set I is required only for visualization. The data structure representing all subsets of dimensions and their containment is called the dimension lattice [11]. Due to their simplicity and wide application in the medical domain we restrict dimensions to be binary. The set of image attributes represents a single image, where each attribute can be a pixel, an image segment or an image region. In OLAP processing, the basic idea is to compute aggregations (sum(), count()) on measures A_i by subsets of dimensions (i.e. cuboids or cuboids) G s.t. $G \subseteq D$, effectively performing aggregations at different granularity levels. The set of all potential aggregations at a certain level is called a cuboid and one specific group is called a cell. In our case, aggregations are used to derive univariate statistics such as μ, σ, which in turn are the basic elements in the equations of a parametric statistical test, introduced in Section 3.

Example

In Figure 1 we present an example of a cube having three dimensions D_1, D_2, D_3. Each face represents a 2-dimensional cuboid. As can be seen, there exist two sets of cell pairs within one cuboid that differ in exactly one dimension. The difference in fill pattern is indicating there is a significant difference on a measure attribute.

3. APPLYING STATISTICAL TESTS ON OLAP CUBES

This section introduces our main technical contributions. We explain how to apply statistical tests to explore OLAP cubes. We propose an algorithm that explores the cube at several dimension granularities to compare highly similar groups. Since we consider data sets with alphanumeric and imaging attributes, our algorithm performs processing of image attributes in order to provide visualization. We introduce optimizations to generate efficient SQL code. We discuss the application of our research in medical databases.

3.1 Statistical Tests

Instead of looking for unusual patterns in cuboids like previous work [8, 10, 9], we propose to use a statistical test to compare pairs of cells in cuboids, providing a more reliable discovery. Most existing work relies on simple comparisons and multi-level aggregations to find interesting findings. Instead, in our approach we exploit a parametric statistical test comparing populations means [22]. Such approach provides the following advantages:

- Two large groups of any size can be compared. Two groups with very different number of elements can be compared (e.g. a large and a small group).
- The means comparison takes into account data variance, which measures measure overlap between the corresponding subpopulations.
- In the case of OLAP, dimensions can be used to focus on highly similar groups, differing in a few dimensions.
- It represents a natural extension of OLAP computations since it relies on distributive aggregations [9].
- Measures are assumed to have a normal distribution, which is applicable in most cases.

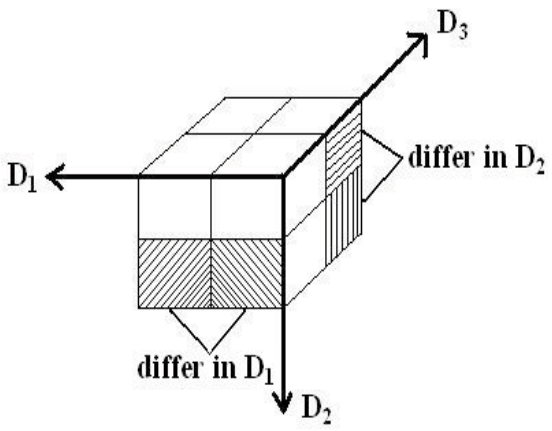


Figure 1: Discovering group pairs with significant measure differences.

We now describe the parametric statistical test in more formal terms. We use a statistical comparison test of the means μ_1, μ_2 from two data subsets (populations), where the size of each data subset is N_1, N_2. Each data subset is assumed to be an independent sample. In this case the null hypothesis H_0 states that $\mu_1 = \mu_2$ and the goal is to find cells where H_0 can be rejected (deemed false) with high confidence $1 - p$, where p generally takes the following thresholds, $p \in \{0.01, 0.05, 0.10\}$. Therefore, the so-called alternative hypothesis H_1 asserts $\mu_1 \neq \mu_2$. When H_0 can be rejected the test will return the significance level p; such outcome will allow us to provide strong statistical evidence supporting $H_1 : \mu_1 \neq \mu_2$. Otherwise, when $p > 0.1$ there does not exist a significant means difference. We use a two-tailed test which allows finding a significant difference on both tails of the Gaussian distribution. The statistical test relies on Equation 1 to compute a random variable z with pdf $N(0, 1)$:

$$z = \frac{|\mu_1 - \mu_2|}{\sqrt{\sigma_1^2/N_1 + \sigma_2^2/N_2}}, \quad (1)$$

where μ_i, σ_i correspond to the estimated mean and standard deviation from group $1, 2$, respectively. When both groups are large the z value just needs to be compared with $z_{p/2}$ in the cumulative probability table for $N(0,1)$ (e.g. $z_{0.495}$) When either group is small or both groups are small the statistical test requires computing the degrees of freedom as

$$df = N_1 + N_2 - 2 \quad (2)$$

and then looking up z on the t-student distribution table according to df. This implies that either $N_1 > 1$ or $N_2 > 1$ because $df \geq 1$. If one group is much larger than the other one then there exists a row for $df = \infty$. For instance, it is possible to compare a singleton set with a large set of records.

Applying the statistical test on cubes

Applying the statistical test has two goals: (1) finding significant differences between two groups in a cuboid on at least one measure. Finding two or more significant measure differences is desirable, but rare. (2) When there exists a significant difference we focus on groups that differ in one dimension, which can explain cause-effect. Even though dimensions are considered independent the aggregation automatically groups records with correlated dimensions together. Therefore, if a high correlation exists in binary dimensions it will be automatically considered. With respect to the first goal, when applying a statistical test a significant difference can only be supported by a small p-value which takes into account both the means and the standard deviation of the distributions. The smaller the p-value the more likely the difference between both groups is significant. It is expected that many differences will not be significant, making the search problem expensive. Regarding the second goal, we are interested in finding significant differences in highly similar groups because that helps explain which specific dimension "triggers" a significant change on the cuboid measure. For instance, finding a significant measure difference, between two highly dissimilar groups, makes causal explanation difficult, since such difference may be attributed to two or more dimensions. Nevertheless, such less significant findings can be stored on additional tiers of group pairs.

3.2 Algorithm

We introduce an algorithm that integrates cube exploration, statistical tests and visualization. Our algorithm has the following goals: (1) exploring all cuboids from F when d is medium or low. Otherwise, when d is large, exploring all cuboids based on k dimensions selected by the user s.t. $k < d$. (2) running the statistical test for every pair. (3) selecting significant pairs differing in δ cube dimensions. (4) interactive visual exploration of the cube together with statistically significant results. (5) efficient visualization of image data associated with each group.

Our algorithm basically computes the entire cube, exploring the entire dimension lattice and then applies statistical tests for every pair. The algorithm assumes a low d or alternatively low k, binary dimensions, which is common in medical databases. Our tool applies a top-down approach exploring all cuboids from a cube, working level-wise.

The algorithm input and output is as follows:

- Input parameters: maximum p-value, δ threshold of maximum # of different dimensions (generally $\delta = 1$).
- Output: a table C containing all cell pairs differing in δ dimensions.

The algorithm steps are the following:

1. Precompute cube with d dimensions on all e measures getting groups at the finest granularity level.
2. Traverse the lattice. Compute sufficient statistics N, L, Q for every group in the dimension lattice.
3. Create group pairs with groups differing in at most δ dimensions.
4. Compute subpopulation parameters $\mu_1, \sigma_1, \mu_2, \sigma_2$, based on n, L, Q.
5. Compute a statistical test for every cell pair on the same level of aggregation.
6. Select pairs having significance value $< p$ with at most δ different dimensions. Categorize results into tiers having 1,2,3 dimensions differences.

When d is small or medium (e.g. our medical data sets), the table F is first aggregated at the finest granular level as given by $D_1, \ldots, D_d$ and then the cube exploration proceeds with subsets of dimensions having $d-1, d-2, \ldots, 1$ dimensions. Otherwise, we assume the user selects k dimensions s.t. $k < d$. Cube pre-computation automatically eliminates groups with zero records (empty groups). When d is large the user specifies a small subset of k dimensions that can be analyzed interactively. Given the combinatorial number of dimension subsets and hence pairs, it is infeasible to explore all of them. Given one level of aggregation the algorithm performs a pair-wise test-based comparison of all group pairs. Such comparison is made on each of the measure attributes. The algorithm tries a list of increasing p-values, in order to find the smallest p value at which H_0 can be rejected. If $p > 0.1$ then the test shows there is little evidence $\mu_1 \neq \mu_2$.

3.3 Exploration and Visualization

After the cube has been built, either with the d or k dimensions, the user can explore it and visualize results. Our prototype allows visualizing pre-processed images through summarization, sampling or visualizing the entire set for a group. We are currently exploring the visualization of individual raw images, when images are not uniform or are bigger. For the remainder of the article we will assume F has pre-processed images at a low resolution. In the case of medical databases image attributes are generally standardized, allowing uniform techniques for visualization. That is, image attributes have been already pre-processed when they are stored in the DBMS.

Our prototype can do the following:

1. Visualizing the cube and all cuboids in a 2D representation.
2. Highlighting significant pairs of cells indicating their different dimension and equal dimensions.

3. Isolating the measure attribute where the significant difference was found.

4. Visualizing associated image data based on image summaries or randomly selected samples (to speedup visualization)

5. Interactively navigating the cube, allowing the user to jump from one cell to another cell.

The 2D representation of the cube follows a checkerboard display similar to a Karnaugh map (as used in digital design), where cells differing in one dimension are visually linked. Each cell has a fill pattern determined by the 1s and 0s determined by the specific aggregation group it represents. Image summaries provide an accurate representation on what an average record looks like; this is enabled by image standardization. Samples are needed to inspect a few images coming from a large group since it would be infeasible to manually inspect a large number of images. When the user switches from one pair to another pair the program dynamically retrieves and displays the corresponding average, sample or "all" images, based on user's selection. In general, our prototype retrieves records and imaging attributes from tables in the DBMS.

3.4 Optimizations

Our tool generates standard SQL code and it can connect to any relational DBMS through the JDBC interface. After exploring a cube the tool produces an output table with the most important group pairs where mean differences are significant, indicating which dimensions are equal and which specific dimension is different.

Our algorithm was implemented by dynamically generating SQL queries to build the d-dimensional cube, traverse the lattice, form pairs and compute the statistical test. We found several issues in trying to optimize SQL queries. (1) It is not possible to pre-define a general-purposes primary index for the cube because d may vary and the corresponding columns will be different, given different fact tables F. Instead the cube table has either a simple primary key to uniquely identify groups (cube cells) or a primary index on the dimensions. (2) A search for a specific cell requires handling nulls and "All" separately. In particular they were coded as negative integer values (i.e. codes different from 0 and 1). (3) Traversing the entire dimension lattice is the most time-consuming stage, especially for subsets of dimensions around $d/2$. (4) OLAP cubes combined with statistical tests are not an "association rule" problem. (5) When visualization is required, it may be necessary to inspect individual record images to get a more concrete idea about statistical findings or when images have not been pre-processed. Therefore, image sampling is required when N is large for some group.

Computing statistical tests on cubes although it seems a similar problem to association rules [11], it is different because there is no "minimum support" threshold and binary dimensions are not equivalent to items [2]. This is because both 1 and 0 are used to get groups, whereas association rule algorithms generally consider only 1s. Each group may have any combination of 0s and 1s in the dimensions.

We introduce the following optimizations: (1) All aggregations are stored on the same table. This table contains all cuboids at different aggregation granularities. The table has a secondary index on all dimensions coding "All" and "null" separately. (2) Search for a specific cell is indexed. The secondary index allows efficient retrieval of cell pairs and associated image data in one indexed search per group (i.e. two indexed searches per pair). (3) All measure and image attributes are uniformly manipulated with sufficient statistics N, L, Q, to be explained below. We are not currently interested in finding significant differences in image regions, but sufficient statistics open that possibility. (4) When there are image attributes we introduce two optimizations. Image attributes are aggregated per cell for visualization. Each cell has its "average" image. That is, OLAP computations are extended to image attributes. Individual images can be visualized performing sampling from a specific group from F, or all images can be visualized when the group (cell) has a small N. Image retrieval is done in a single query, retrieving all image information for display purposes. Once images are retrieved they are held in main memory for visualization.

Contrary to common intuition, when d is small a "bottom-up" level-wise algorithm is not used. Since dimensions are assumed to be independent and the statistical test relies on averages rather than counts (frequencies), downward closure ("association rule"-like) optimizations are not applicable. That is, we cannot explore $k-1$-dimensional cuboids in order to prune out k-dimensional cuboids. In fact, a significant means difference might be found between cuboids having very different N_1, N_2. This leads to an exhaustive search process, computationally expensive for a high-d cube.

The tool uses the following optimizations for efficient statistical test computation: (1) F is aggregated at the finest granular level producing a cube table C and further coarser-grained aggregations are computed from C. This step is extremely important to eliminate empty cells in the cube. Only groups (cells) with $N > 0$ participate in further processing. (2) Sufficient statistics are computed on each cell so that univariate statistics can be derived in one pass. Such statistics include count(*), sum(A_i) and sum(A_i^2) for a measure column A_i. More specifically, $N_i, L_i = \sum A_i, Q_i = \sum A_i^2$ for each A_i. This idea is also applied to image data: $N_i^I, L_i^I = \sum I_i, Q_i^I = \sum I_i^2$ for I_i. Then $\mu_1, \mu_2, \sigma_1, \sigma_2$ are easily derived from sufficient statistics. (3) In general, for large groups it is required to compare the test statistic against a given z value using the normal $N(0,1)$ distribution, which generally requires looking up a value in a small table. We introduce a simple optimization, finding the significance of the test statistic without visiting such table with a CASE statement based on the specific p-values commonly used in the medical domain ($p \in \{0.01, 0.05, 0.10\}$). That is, finding $z_{p/2}$ for the two-tailed test is done in main memory. On the other hand, when the group is small (say < 30) we perform an indexed search on the t-student distribution using df as the search key. (4) Depending on input parameters, the SQL code will compare only groups having up to δ dimension differences, being $\delta = 1$ the default. This means groups in a pair must differ in up to δ dimensions in order to be compared.

3.5 Medical Application: Finding Differences between Similar Groups

Table 1 shows actual significant findings on a medical data set, explained in more detail in Section 4. Each row represents the comparison between two patient groups, differing in one dimension indicated by "0/1". Cube dimensions are

D_1 FamHist	D_2 Diab	D_3 Gender	D_4 HighChol	D_5 highBP	N_1	N_2	A_1 LAD	A_2 RCA
0	All	All	1	0/1	35	23	$p > 0.1$	$p \in [0.01 - 0.05]$
0/1	All	All	1	0	35	26	$p > 0.1$	$p < 0.01$
All	0/1	All	All	All	47	157	$p < 0.01$	$p < 0.01$

Table 1: Medical database: group pairs with significant measure differences.

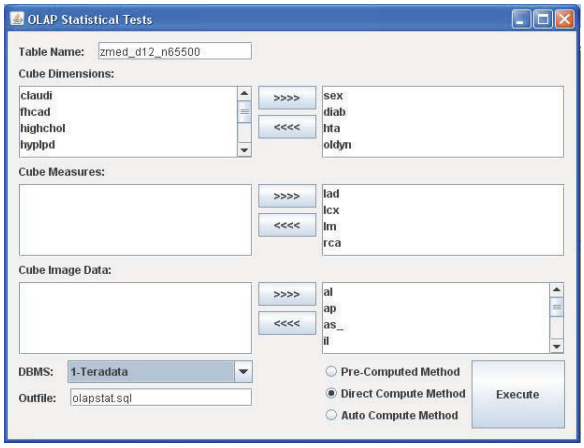


Figure 2: OLAP statistical test GUI.

well-known risk factors for heart disease including family history of heart disease, diabetes, gender, high cholesterol and high blood pressure. For a given group, each matching dimension will be 0, indicating absence of a risk factor, 1 indicating presence of a risk factor, or "All" indicating such dimension was ignored in the aggregation. When p is small it indicates two highly similar groups of patients, differing in exactly one risk factor have a high likelihood of having a different artery health (medical measurement of artery narrowing in this case).

3.6 GUI for Visualization and Exploration

We are currently applying our prototype on a medical data set having heart images, where such images are normalized. The images are pre-processed to get 32 regions, describing an 8×4 2-dimensional map of the heart muscle. For medical interpretation these 32 regions are condensed into 9 regions that are in the range [-1,1]. A value close to -1 indicates a healthy region, whereas numbers closer to +1 indicate a severe degree of disease, with 0 being neutral. These numbers allow visualizing a spectrum of patients' health in color. This scale also allows a uniform visualization technique for heart images of diverse patients.

Our tool GUI and visualization aids is illustrated in Figure 2, where the medical doctor can select cube dimensions, cube measures and image attributes. The output from the tool is shown in Figure 3 (for an average image per group) and Figure 4 (with a sample of heart images per group). The tool allows 2D visualization for the dimension lattice on the left panel for a specific dimensionality k, where $2 \leq k \leq d$. This 2D visualization is based on a checker board display, where red means "1" (risk factor present) and blue means "0" (risk factor absent). Since this checkerboard is based on a set of k selected binary dimensions there are up to $\binom{d}{k}$ cells. A pair of cells linked by the green lines means they differ in one dimension, which highlights a specific risk factor triggering heart disease. The left upper part indicates dimensions which are equal, while the right panel indicates the specific dimension that is different for each group (diabetes in this case). For heart image attributes, on the right two windows the medical doctor can visualize for a group with many records a summary of all its images, or alternatively, a sample dynamically retrieved from the database. Otherwise, when the group is small enough all its images can be visualized (perhaps as thumbnails). In short, these two windows enable visualization of image data, with a scale going from -1 (very sick) to +1 (completely healthy).

4. EXPERIMENTAL EVALUATION

Our tool is an OLAP query generator developed in the Java language, where queries are written in ANSI SQL for maximum portability. Our tool connects to the DBMS via the standard JDBC (Java Database Connectivity) interface. We conducted our experiments on a modern DBMS. The DBMS was SQL Server running on a server with a CPU at 3.2GHz, 4GB of memory and 750GB on disk.

4.1 Data Sets

We performed experiments based on two real data sets coming from the medical domain. The first medical data set contains profiles of $n = 655$ patients and has 25 attributes containing categorical, numeric and image data. This data set was obtained from a hospital and we call it the "Heart" data set. There were medical measurements such as weight, heart rate, blood pressure and pre-existence of related diseases. Finally, the data set contains the degree of artery narrowing (stenosis) for the four heart arteries. All numeric attributes were converted to binary dimensions. There were $d = 12$ binary dimensions (e.g. gender, hypertension Y/N), $e = 4$ measures (artery disease measurement) $f = 9$ image attributes representing a standardized image of the heart. The second data set was obtained from the UCI Machine Learning repository [3] and we call it "Thyroid". The Thyroid data set contained the profiles of $n = 9,172$ patients. We transformed this data set to have $d = 10$ binary dimensions and $e = 5$ measures; this data set had no image attributes. These data sets were treated as the fact table F, defined in Section 2.

4.2 Default settings

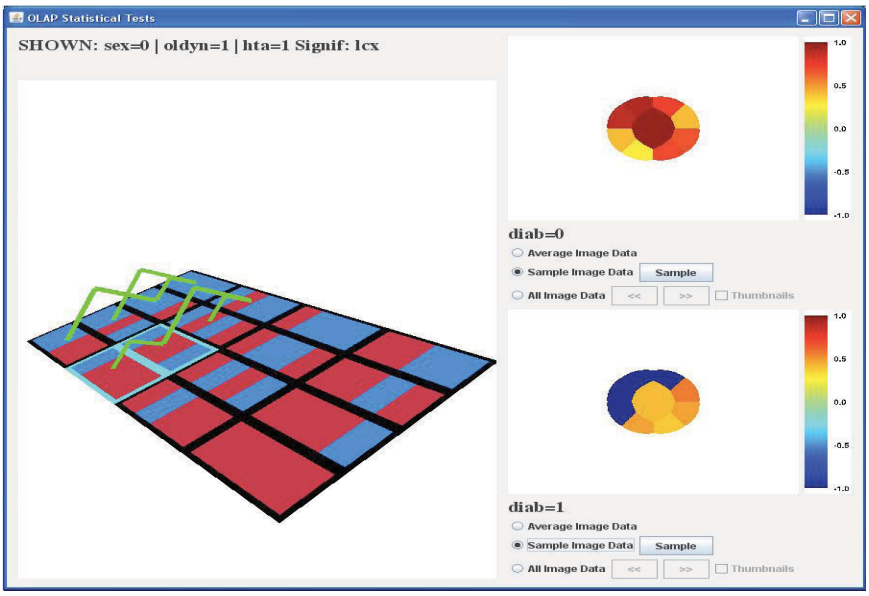


Figure 3: Cube exploration in 2D and image visualization (average image).

For our medical data sets our goal was to explore the entire dimension lattice. Therefore, we used all d dimensions. The settings for parameters were as follows. $p = 0.01$, $\delta = 1$, which can be interpreted as follows. We want to find significant measure differences, with 99% confidence, on all group pairs differing in one dimension. A group pair in the cube can have from 1 to d dimensions, out of which one will be different. It is possible, but unlikely, that a group pair has significant differences in two or more measures.

4.3 Significant Group Pairs

We provide a summary of pairs having a significant difference in at least one measure for both data sets. Table 2 summarizes significant pairs, which represent potentially valuable medical knowledge. Those pairs where $p < 0.01$ are valuable since they show a discriminating risk factor (binary dimension) causes a significant measure change. Those pairs whose p-value is between 0.05 and 0.10 are considered unimportant findings and therefore we do not show them. The most important pairs are those with a few equal dimensions (1 or 2) because they are general and involve larger groups, and those with many equal dimensions (8 for Heart data set [19], 10 for Thyroid data set [3]) because they summarize redundant subsets in the middle of the lattice, but they tend to be specific.

Table 2: Significant group pair differences.

Data Set	p-value	# equal dims	# pairs
Heart	<0.01	1	6
Heart	<0.01	2	76
Heart	<0.01	3	378
Heart	<0.01	4	974
Heart	<0.01	5	1436
Heart	<0.01	6	1287
Heart	<0.01	7	705
Heart	<0.01	8	201
Heart	<0.01	9	0
Thyroid	<0.01	1	8
Thyroid	<0.01	2	88
Thyroid	<0.01	3	406
Thyroid	<0.01	4	1068
Thyroid	<0.01	5	1780
Thyroid	<0.01	6	1966
Thyroid	<0.01	7	1441
Thyroid	<0.01	8	680
Thyroid	<0.01	9	188
Thyroid	<0.01	10	23

4.4 Image Visualization

We now discuss experiments for image visualization. These experiments illustrate the interactive response of our tool to explore the cube, isolate significant pairs and visualizing

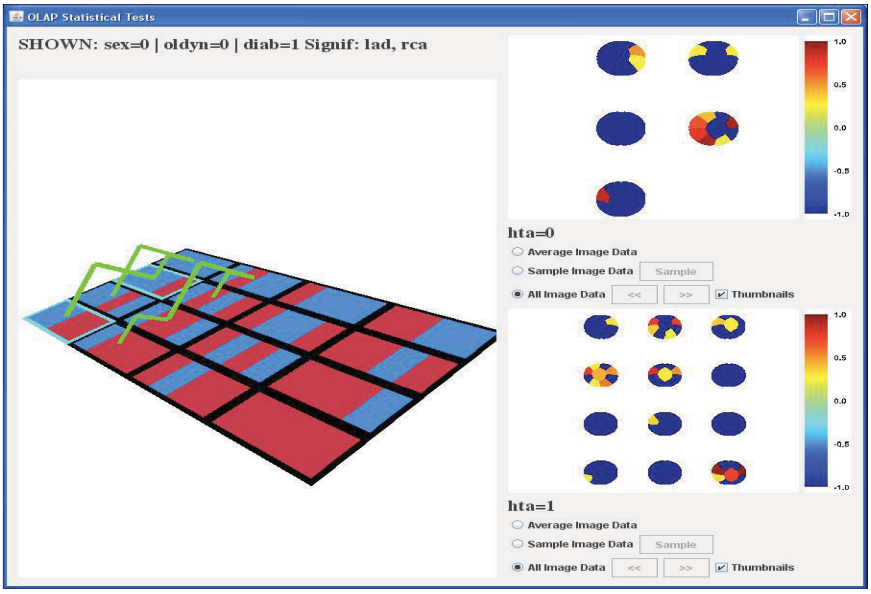


Figure 4: Cube exploration in 2D and image visualization (sample of images).

associated image data. We recomputed the cube at each d and then we measured the time to retrieve one group at each d. Based on expert opinion, we initially ranked dimensions in order of medical importance so that the most important dimension had rank one. Figure 5 shows time growth to retrieve images from the database. We compare the time to visualize the "average" image versus sampling one image from one cube cell (group); times are in milliseconds. In this case the cube is already pre-computed and we dynamically retrieve one average or one sample image for the group based on the following common medical dimensions: oldyn, sex, diab, hta, highchol. This represents a highly interesting group with specific risk factors. The left graph shows the time to retrieve images in this group varying n. In this case the time to retrieve an average image remains constant because the average image is already precomputed from sufficient statistics and the search is fully indexed for equality search. On the other hand, the time to retrieve one sample image grows with n because the group also grows in size. The plot shows time to retrieve all images as d grows, with C being recomputed at every d. We do not include the time to precompute C in the time to retrieve images. We can see the time to get the average image grows slowly as d grows showing an asymptotic behavior. On the other hand, we can see the time to get a sample image from one group gets increasingly faster because the group shrinks in size. The time to retrieve one sample should be greater than the time to retrieve the average image because the sample is dynamically obtained from one group which cannot be obtained through a fully indexed search.

In order to give a complete experimental evaluation on image visualization, we now discuss experiments including the time to retrieve all images in one cell. Table 3 shows the times to retrieve images varying n and Table 4 has the times to retrieve images varying d. We include the times to retrieve the average image from the cube, one sample image from one group and the time to retrieve all images from one group. Notice retrieving all images from a group incurs on significant overhead, but it may still be acceptable for data sets having $n < 100$k. Our image retrieval is done in a single query, retrieving all image information for display purposes. Once images are retrieved they are held in main memory for visualization.

Table 3: Image retrieval varying n (msecs).

n	Average	Sample	All
655	40	31	32
1310	40	37	41
2620	40	45	75
5240	41	46	100
10480	41	49	231
20960	42	51	487
41920	42	86	907
83840	43	146	1722

4.5 Time complexity

Our previous experiments showed the challenge for our problem is d and not n. Nevertheless, we created larger data sets replicating F several times and we measured time

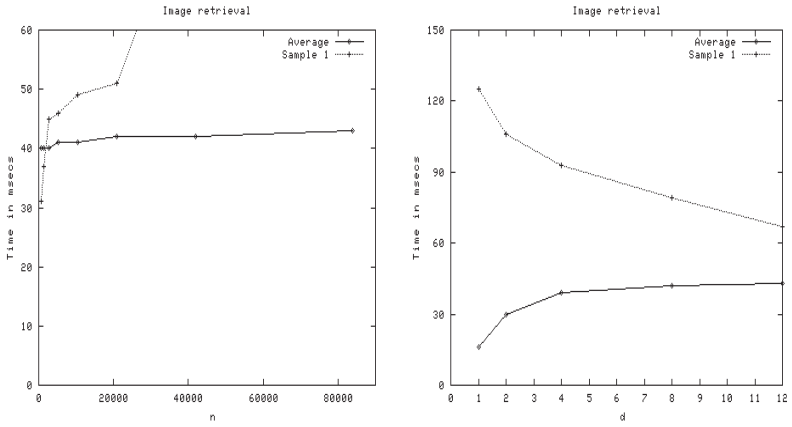


Figure 5: Time complexity for image retrieval; times in msecs.

Table 4: Image retrieval varying d (msecs).

d	Average	Sample	All
1	16	125	2100
2	30	106	1279
4	39	93	506
8	42	79	203
12	43	67	94

for the entire process. The contribution of n to total time was so small that a plot varying n was a flat line. Therefore, we omit such plot.

Figure 6 illustrates time scalability as we vary d. The entire cube, cuboids, pairs and statistical test are computed at each d. The left graph corresponds to the Heart data set and the right graph corresponds to the Thyroid data set. It is clear the time complexity grows exponentially as d increases, highlighting the expensive search process for significant pairs. However, for our medical data sets the entire lattice is explored in a matter of minutes.

4.6 Profiling Processing Steps

Table 5 gives a breakdown of the total execution time to explore the entire dimension lattice. As we can see most computations take significant time, despite F being a small data set. Traversing the dimension lattice is the slowest operation; this step requires creating an output group for each dimension combination and it is I/O bound since it requires visiting every F row. Computing the test statistic is the second slowest operation; this is mostly CPU time. Classifying pairs into tiers for final analysis comes slightly behind. The fastest operation is computing the cube at the finest granularity level to get groups having d dimensions.

Table 6 compares optimizations. For the sake of completeness, we simulated a large fact table by replicating F 1000 times. This gives two large data sets having n =655k rows and $n \approx$ 9M rows, respectively. We want to understand how much longer it takes to compute the group-by query to get the cube on d dimensions. As we can see the cube can still be efficiently computed on large fact tables. The

Table 5: Profile of cube exploration ($d = 12$ and $d = 10$, time in secs).

Step	Heart		Thyroid	
	Time	%	Time	%
Cube & NLQ	< 1	0	< 1	0
Get N, L, Q on lattice	495	31	116	34
Compute μ, σ from N, L, Q	173	11	28	8
Create group pairs based on δ	158	10	39	12
Compute test statistic	419	26	87	26
Categorize based on p-value	344	22	66	20

Step	Data set	N	Y	Impr
cube & NLQ	Heart	na	< 1	-
cube & NLQ $n \times 1000$	Heart	na	11	-
lattice NLQ from cube	Heart	1396	1366	-2%
Primary index on dims	Heart	1366	400	-70%
cube & NLQ	Thyroid	na	< 1	-
cube & NLQ $n \times 1000$	Thyroid	na	82	-
lattice NLQ from cube	Thyroid	336	216	-36%
Primary index on dims	Thyroid	216	100	-54%

Table 6: Optimizations (na=not applicable, impr=improvement).

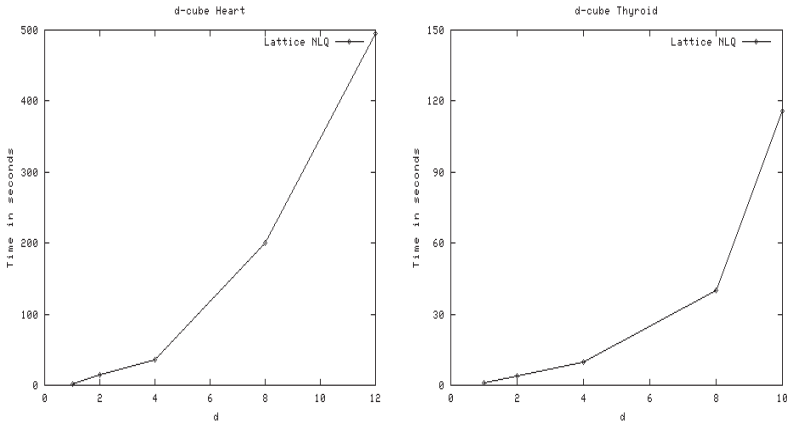


Figure 6: Time complexity to compute cube varying d.

rest of the steps remain unaffected with the same measured time. Time for computing the cube directly from the data set F is not applicable in the first two rows. Precomputing the d cube and computing N, L, Q is compared with directly computing N, L, Q from F. That is, we want to understand if it is worth it to compress the (small) data set by precomputing the cube at the finest granularity level. As we can see there is an important performance improvement for the Thyroid data set. A primary index on dimensions is compared with a simple primary key for each group. In this case we want to understand the improvement on time to search each group efficiently. As we can see the index in dimension has a significant impact on performance.

5. RELATED WORK

Cube exploration is a well researched topic. OLAP and a classification of aggregations originates in the seminal paper [9]. Methods to increase the performance of multidimensional aggregations, by combining novel data structures and precomputation at different aggregation levels, are introduced in [1, 8]. The authors of [10] puts forward the plan of creating smaller, indexed summary tables from the original large input table to speed up aggregating executions. In [21] the authors explore tools to guide the user to interesting regions in order to highlight anomalous behavior while exploring large OLAP data cubes. This is done by identifying exceptions, that is, values in cells of a data cube that are significantly different from the value anticipated, based on a statistical model. In contrast, we propose to use statistical tests to do pair-wise comparison of neighboring cells in cuboids to discover significant metric differences between similar groups. We identify such differences giving statistical evidence about the validity of findings. Reference [12] proposes computing large lattices with a greedy algorithm.

Our work is related to applying data mining in medical data sets to improve heart disease diagnosis [19, 18]. These works discuss the medical significance of pairs to detect specific risks for heart disease. Neural networks are used to predict heart response based on exercise stress and heart muscle thickening images [4]. A basic set of search constraints is introduced in [20] and experimental results stress their importance.

There has been research on visualizing OLAP cubes. Reference [15] studies the problem of improving understanding through the use of visualization. Recent work can also be found on the use of tools to visually and interactively explore OLAP warehouses. The authors for [24] explore the requirements for analyzing a spatial database with an OLAP tool. This work shows the need to apply spatial data techniques, used in geographic information systems, for OLAP exploration, in which drill up/down, pivoting, and slicing and dicing provide a complementary perspective. In contrast, our work relies on statistical tests to explore OLAP cubes and can automatically detect significant metric differences between highly similar groups. Additional visualization work was completed in [16] where the mapping of the Cube Presentation Model, a display model for OLAP screens, involves visualization techniques from the Human-Computer Interaction field. In [14], the author presents a rigorous multidimensional visualization methodology for visualizing n-dimensional geometry and its applications to visual and automatic knowledge discovery. The application of visual knowledge discovery techniques is possible by transforming the problem of searching for multivariate relations among the variables into a two-dimensional pattern recognition problem. A framework for exploration of OLAP data with user-defined dynamic hierarchical visualizations is presented in [23]. While this study emphasizes the use of visualization tools to explore data warehouses, we are proposing a tool that not only gives the user visual aids to explore the data, but also to present the user with a novel method of highlighting interesting features of the cubes by means of statistical tests. Indexing is one method of increasing performance in the searching of images and the authors in [6] proposed a multilevel index structure that can efficiently handle queries on video data.

6. CONCLUSIONS

In this article, we presented a prototype which combines OLAP cube exploration, statistical tests and visualization. Statistical tests are applied on pairs of similar groups in order to find significant measure differences caused by some

distinctive dimension. The cube is explored with automatically generated SQL code. Our tool can produce statistically reliable results on both large and small subsets. The internal aggregation algorithm incorporates several database optimizations. The cube is precomputed when the number of cube dimensions is low. Otherwise, the program works with a user-specified set of dimensions to pre-compute a lower dimensional cube. The cube incorporates a primary index on dimensions for efficient group search. We also introduced optimizations for interactive visualization of the cube, statistical test results and associated image data. The fact table is indexed for efficient image retrieval. Image attributes are uniformly treated as measures in order to get an average representative image per group through sufficient statistics or to collect sample images.

There are many research issues for future work. The combination of OLAP processing and visualization creates several research challenges. In particular, visualization of high-dimensional cubes requires novel cube data transformation techniques. In the case of medical databases image attributes can be efficiently visualized in a cell, but this could not be done for non-uniform images: dynamic image compression techniques could be required to interactively visualize them. We want to study how to further optimize pair creation and computation of statistical tests. There is a big family of statistical tests that may be applied in OLAP databases. The cube provides a natural way to summarize large databases which is more difficult if underlying records have image attributes.

Acknowledgments

We would like to thank the Emory University Hospital for providing the medical data set used in this work.

7. REFERENCES

[1] S. Agarwal, R. Agrawal, and P. Deshpande. On the computation of multidimensional aggregates. In *VLDB*, pages 506–521, 1996.

[2] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *ACM SIGMOD Conference*, pages 207–216, 1993.

[3] A. Asuncion and D.J. Newman. *UCI Machine Learning Repository*. University of California, Irvine. School of Inf. and Comp. Sci., 2007.

[4] L. Braal, N. Ezquerra, E. Schwartz, and Ernest V. Garcia. Analyzing and predicting images through a neural network approach. In *Proc. of Visualization in Biomedical Computing*, pages 253–258, 1996.

[5] S. Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. *ACM SIGMOD Record*, 26(1):65–74, 1997.

[6] L. Chen, M. Ozsu, and V. Oria. Mindex: An efficient index structure for salient-object-based queries in video databases. *Multimedia Syst.*, 10(1):56–71, 2004.

[7] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11):27–34, November 1996.

[8] Lixin Fu and J. Hammer. Cubist: a new algorithm for improving the performance of ad-hoc OLAP queries. In *DOLAP Workshop*, 2000.

[9] J. Gray, A. Bosworth, A. Layman, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab and sub-total. In *ICDE Conference*, pages 152–159, 1996.

[10] H. Gupta, V. Harinarayan, A. Rajaraman, and J.D. Ullman. Index selection for OLAP. In *IEEE ICDE Conference*, 1997.

[11] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, San Francisco, 1st edition, 2001.

[12] V. Harinarayan, A. Rajaraman, and J.D. Ullman. Implementing data cubes efficiently. In *ACM SIGMOD Conference*, pages 205–216, 1996.

[13] T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning*. Springer, New York, 1st edition, 2001.

[14] A. Inselberg. Visualization and knowledge discovery for high dimensional data. In *UIDIS*, pages 5–24, 2001.

[15] D. A. Keim, C. Panse, J. Schneidewind, M. Sips, M. C. Hao, and U. Dayal. Pushing the limit in visual data exploration: Techniques and applications. In *KI*, pages 37–51, 2003.

[16] A. S. Maniatis, P. Vassiliadis, S. Skiadopoulos, and Y. Vassiliou. Advanced visualization for OLAP. In *ACM DOLAP*, pages 9–16, New York, NY, USA, 2003. ACM Press.

[17] T.M. Mitchell. *Machine Learning*. Mac-Graw Hill, New York, 1997.

[18] C. Ordonez. Association rule discovery with the train and test approach for heart disease prediction. *IEEE Transactions on Information Technology in Biomedicine (TITB)*, 10(2):334–343, 2006.

[19] C. Ordonez, N. Ezquerra, and C.A. Santana. Constraining and summarizing association rules in medical data. *Knowledge and Information Systems (KAIS)*, 9(3):259–283, 2006.

[20] C. Ordonez, E. Omiecinski, Levien de Braal, Cesar Santana, and N. Ezquerra. Mining constrained association rules to predict heart disease. In *IEEE ICDM Conference*, pages 433–440, 2001.

[21] S. Sarawagi, R. Agrawal, and N. Megiddo. Discovery-driven exploration of OLAP data cubes. In *EDBT*, pages 168–182. Springer-Verlag, 1998.

[22] M. Triola. *Essentials of Statistics*. Addison Wesley, 2nd edition, 2005.

[23] S. Vinnik and F. Mansmann. From analysis to interactive exploration: Building visual hierarchies from OLAP cubes. In *EDBT*, pages 496–514, 2006.

[24] A. Voß, V. Hernandez, H. Voß, and S. Scheider. Interactive visual exploration of multidimensional data: Requirements for CommonGIS with OLAP. In *DEXA Workshops*, pages 883–887, 2004.

Hierarchical Difference Scatterplots

Interactive Visual Analysis of Data Cubes

Harald Piringer
VRVis Research Center
Vienna, Austria
hp@vrvis.at

Matthias Buchetics
VRVis Research Center
Vienna, Austria
buchetics@vrvis.at

Helwig Hauser
Department of Informatics
University of Bergen, Norway
Helwig.Hauser@uib.no

Eduard Gröller
Institute of Computer Graphics
and Algorithms
Vienna University of
Technology, Austria
groeller@cg.tuwien.ac.at

ABSTRACT

Data cubes as employed by On-Line Analytical Processing (OLAP) play a key role in many application domains. The analysis typically involves to compare categories of different hierarchy levels with respect to size and pivoted values. Most existing visualization methods for pivoted values, however, are limited to single hierarchy levels. The main contribution of this paper is an approach called Hierarchical Difference Scatterplot (HDS). A HDS allows for relating multiple hierarchy levels and explicitly visualizes differences between them in the context of the absolute position of pivoted values. We discuss concepts of tightly coupling HDS to other types of tree visualizations and propose the integration in a setup of multiple views, which are linked by interactive queries on the data. We evaluate our approaches by analyzing social survey data in collaboration with a domain expert.

Categories and Subject Descriptors

H.5 [**Information Interfaces and Presentation**]: Miscellaneous

General Terms

Data cubes, OLAP, focus+context visualization

Keywords

Data cubes, OLAP, focus+context visualization

1. INTRODUCTION

Data dimensions of multivariate datasets can roughly be distinguished as being either continuous or categorical. While the data of some application fields is predominantly continuous (e.g., physical quantities), many application domains have to deal with mixed data, which has many categorical as well as continuous attributes (e.g., data from Customer Relationship Management). In this case, pivot tables are widely used to summarize the values of continuous attributes with respect to a classification given by categories. On-Line Analytical Processing (OLAP) [4] uses categorical attributes, called *Dimensions*, to split the data before aggregating continuous attributes, called *Numeric Facts*. An important aspect of OLAP systems is to use large-scale overview summaries of the data as starting point for selective drill down into interesting parts of the data.

OLAP is based on the fact that categorical data is closely related to hierarchical data and selective drill down (and roll up) is thus related to navigating a hierarchy. Apart from inherently hierarchical categories (e.g., years can be subdivided into months, days, hours, etc.), dimension composition is the key approach for defining hierarchies as it allows for specializing the categories of one attribute by the categories of another one. For example, two separate attributes "sex" and "age group" can be combined to obtain a category like "female and younger than 30". In the context of information drill down, pivot tables are also hierarchically structured and often referred to as data cubes (or OLAP cubes). Interactive analysis tools for pivot tables should consequently support navigation in a way that it is up to the user to decide where to drill down and where to stay at a summary level. They should reflect this hierarchical aspect in the visualization.

Apart from the navigation within the hierarchy itself, a frequent analysis task is to compare categories within one hierarchy level and also between multiple hierarchy levels. The difference of pivoted values with respect to parent categories may characterize individual categories very well as demonstrated by common statements like "the average income in a particular region is x percent higher as compared to the entire country". A visualization approach for OLAP cubes should therefore also facilitate relating categories along the hierarchy.

Based on these considerations, this paper introduces the Hi-

VAKD'09, June 28, 2009, Paris, France.

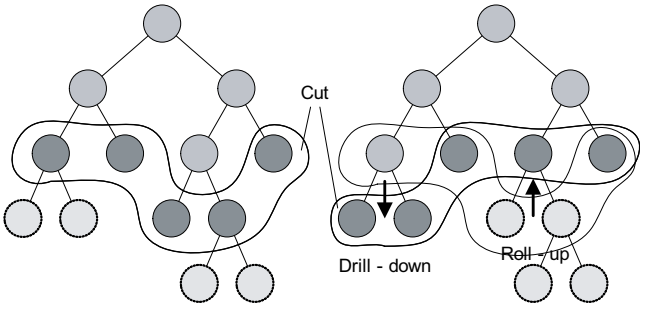


Figure 1: Navigating a hierarchy. Dark nodes represent the current state of navigation (the "cut"); nodes above the cut are contextual information and nodes below the cut are not visualized. Drill-down and roll-up operations transform the left hierarchy to the one on the right-hand side.

erarchical Difference Scatterplot (HSP) as a novel approach to the interactive visual analysis of OLAP cubes. The following list of goals and tasks guided the design of HDS:

- Relating categories to siblings and to parent categories with respect to two continuous attributes. Our consideration is that differences between pivoted values of parent and child categories provide an intuitive way of comparison. We therefore represent them explicitly.
- Integrating multiple hierarchy levels into a single visualization in order to analyze hierarchy levels in the context of the other levels.
- Supporting local drill-down and roll-up (see Fig. 1). Unlike other hierarchical visualizations, it is an essential aspect of HDS to provide different levels of detail for various parts of the data instead of representing the entire hierarchy as such. This is in accordance with drill-down tasks in huge OLAP cubes, which also often emphasize depth rather than breadth.
- Supporting a setup of multiple linked views in order to dynamically integrate results of arbitrary queries as defined by the user in linked visualizations (e.g., a certain cluster of customers of a sales dataset as selected in parallel coordinates).

Our clear focus is on supporting specific OLAP tasks by a combination of visualization and interaction. It is explicitly not the goal of HDS to be superior to existing tree visualizations with respect to providing visually pleasing still images of huge hierarchies as a whole. For tasks where this is required, we discuss, how other types of hierarchical visualizations can be tightly coupled to HDS. As one of many potential application scenarios, we evaluate our approach by analyzing a real-world social survey regarding national identity. The analysis has been conducted in collaboration with a social scientist. We also provide a discussion of analysis tasks as supported by HDS, limitations, and a motivation for visualizing differences explicitly.

2. RELATED WORK

Pivot tables have long been used to summarize values of continuous attributes with respect to a classification given by categories. Flat pivot tables can be visualized using common techniques for multivariate, quantitative data. The Gapminder Trendalyzer [6], for example, maps two aggregated indicators of countries to the axes of a time-dependent scatterplot and shows the population, i.e., the size of the category, by the area of according circles.

The concept of pivoting data is also important for databases, where the predominant Structured Query Language (SQL), for example, offers the "GROUP BY" clause of "SELECT" statements for this purpose. However, as Gray et al. [7] point out, SQL statements have limitations with respect to drill-down and roll-up operations. Therefore they propose to treat multidimensional databases as n-dimensional data cubes, which have widely been adopted by On-Line Analytical Processing (OLAP) [4]. OLAP supports drill-down operations by splitting single categories with respect to additional dimensions.

While most OLAP front-ends only offer selected business graphics, Polaris [18] uses a formal algebra as specification of pivot tables and their visual representation. The user can incrementally construct complex queries by intuitive manipulations of this algebra. The layout is based on small-multiple displays of information [21]. Stolte et al. [19] also describe an extension to the algebra for rich hierarchical structures. Polaris is a very intuitive and highly effective approach for analyzing data cubes, as shown by the success of its commercial version Tableau [1]. However, Polaris displays a single level of detail (i.e., hierarchy level) and thus does not support comparisons between different levels of detail. The authors of Polaris also describe design patterns for adapting visualizations of data cubes on multiple scales [20]. This work deals with transitions between level of details while still showing a single level of detail at a time. It has been mentioned as future work to communicate parent-child relationships and to deal with non-uniform branching factors.

The current version 4.1 of Tableau [1], however, does support comparisons between hierarchy levels using sub-totals and grand-totals, which are displayed in additional rows and columns. As the main drawback of this approach, comparisons require the user to look at multiple places on the screen in a successive manner. This makes comparisons difficult as will be discussed in more detail in section 6. This problem is inherent for approaches that rely on showing absolute values in a side-by-side manner. Therefore, visualizing differences explicitly was a main consideration in the design of HDS.

Hierarchical Parallel Coordinates [5] categorize a dataset by clustering before using this classification for multi-resolution analysis of aggregated values. This approach draws information of different levels of the hierarchy into one visualization. However, unlike HDS as introduced in this paper, Hierarchical Parallel Coordinates are limited to comparing results along one cut through the hierarchy, while our approach focuses on differences between levels. Sifer [16] proposes parallel trees, which also employ a parallel axes layout for aligning multiple drill downs into a data cube. The categories of all hierarchy levels are stacked on top of each other. For analysis, the user may relate one active dimension to all others by coloring parts of the boxes. This implicitly conveys the information for comparing siblings as well as child categories to parent categories. Differences are not represented explic-

itly which requires remembering one category and shifting the attention to another one for comparison. This becomes even more difficult as categories are scaled in proportion to their relative frequencies and thus their size may differ significantly. Moreover, parallel trees require categorization of continuous dimensions (i.e., facts) and do not support typical aggregations like average or sum. This severely limits their applicability to frequent OLAP tasks.

There has been very much research on the *visualization of hierarchies* and hierarchically structured data. Containment-based approaches like Tree Maps [15] are one of the most popular techniques and show the size of the hierarchy nodes very well, while depth information is occasionally harder to read. In contrast, node-link representations [2, 8] show the structure more explicitly, but most approaches do not clearly convey the size of the nodes. The rooted tree growing from top to bottom is a very common layout, but does not utilize space efficiently for large hierarchies. Centric approaches are superior in this respect as they grow outwards from the representation of the root node and thus allocate more space to more detailed levels of the hierarchy. Nodes are typically placed corresponding to their position in the hierarchy, e.g., putting nodes with equal depth on concentric circles (radial tree) [2] or enclosing each sub-tree in a bubble (balloon tree) [8]. There are many extensions and variations to these approaches: focus+context techniques to improve scalability [10], combinations of node-link representations and enclosure [25], combinations of centric layout and enclosure [24], and edge bundles for integrating relations between items into the visualization [9]. Only a few approaches derive the node placement from multi-variate properties (i.e., each node is associated with several attributes) rather than edge topology as necessary for typical OLAP tasks.

Wattenberg proposes PivotGraph [23] for analyzing multivariate graphs and he addresses OLAP by supporting drill-down and roll-up. The graph layout corresponds to a grid which is given by two categorical dimensions for the X and the Y axes, respectively, and edge thickness is determined from the number of edges being aggregated. While the basic idea of property-based node placement is similar to HDS, there are several differences. PivotGraph only supports placement based on discrete dimensions while HDS uses a node layout scheme suited for comparison of differences between continuous facts. Moreover, PivotGraph visualizes a single level of detail at a time (similar to Polaris [18]) and thus does not allow for relating nodes to their parents. After all, the intention of PivotGraph is to improve the interpretability of the graph topology for a particular level of detail, while HDS focuses on comparing aggregated facts along and across a categorical hierarchy.

Queries defined through interaction within visual representations (also known as "brushing") are a proven standard approach for the identification of selected data subsets of interest. Successful systems such as Spotfire [17] offer interactive queries as an integral technology to link multiple views. There has been little research on integrating brushed subsets in hierarchical visualization techniques. In particular, no approach explicitly characterizes brushed subsets by displaying the difference between the properties of an entire category and its selected part.

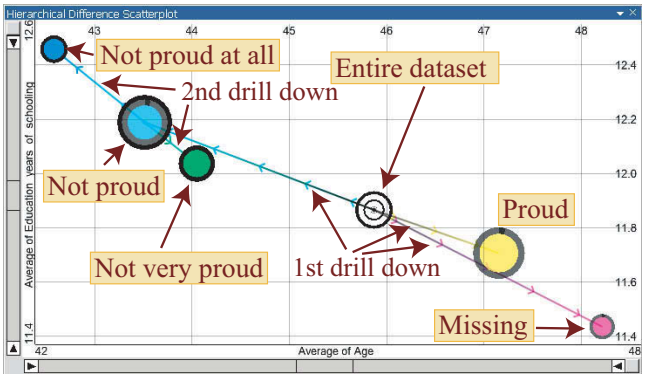


Figure 2: A simple hierarchy as conceptual example: the average age (X-axis) and the average years of schooling (Y-axis) are compared for several degrees of pride on armed forces and the entire data. Drill-down on "not proud" distinguishes "not very proud" and "not proud at all". The size of nodes shows the number of respective interviewees. The visualization reveals that pride is increasing with age and is decreasing with education.

3. HIERARCHICAL DIFFERENCE SCATTERPLOTS

This section introduces the Hierarchical Difference Scatterplot (HDS) as a novel combination of scatterplots and tree visualizations. After describing the approach itself, we provide examples of tightly coupling HDS to other hierarchical visualizations and propose techniques for linking our technique to other multivariate visualizations.

3.1 Visualization

The main idea of HDS is to layout nodes of a tree based on properties similar to a scatterplot (see Fig. 2). For parameterization, HDS require a pre-defined hierarchy, i.e., a data cube, and several properties, which are assigned to the visual attributes X-position, Y-position, size, and color. Properties may be pivoted values of continuous data attributes. An example are aggregated "measures" like the average revenue per node or other aggregates like minimum, maximum, median, sum, etc. Another possibility are inherent features of hierarchy nodes like absolute frequencies or depth. Applying data-driven glyph placement [22], the properties assigned to the X- and Y-attributes are directly mapped to the position of the visual representations of categories. In addition to X- and Y-position, the user may independently assign different properties to size and color which is comparable to Polaris [18], or use default settings. For example, size per default represents the number of raw data items for each node. Color is discussed further below.

In accordance with the idea of information drill-down, the user may increase the complexity incrementally and selectively. Initially, the entire data cube is handled as a single category and it is thus shown as one visual item. By clicking on this item, the user may drill down to the next hierarchy level that displays the respective hierarchy nodes as additional visual items. Clicking on any of these items adds its direct children and thus increases the amount of shown information locally for this particular sub-tree (see Fig. 2). As most important aspect of HDS, the visualization is not lim-

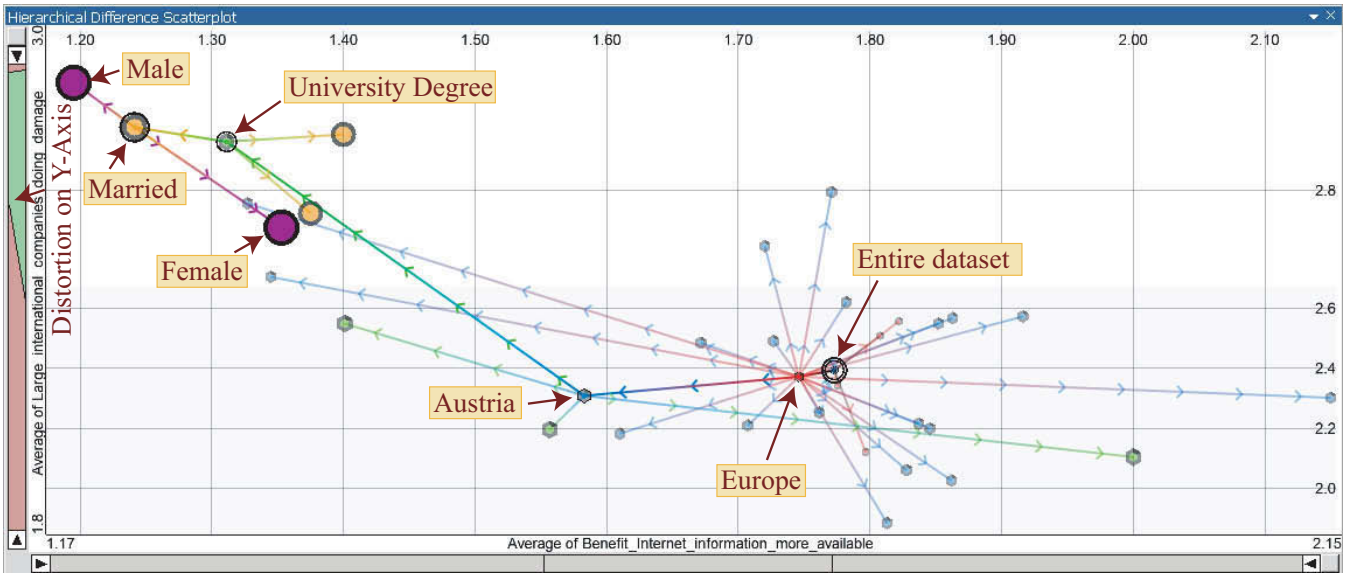


Figure 3: Example of a deep drill-down: the focus is on comparing men and women of the category path Europe – Austria – University degree – Married (i.e., five levels of the hierarchy plus the root) with respect to their attitude towards the Internet and international companies. Size, color and opacity are used to visually discriminate hierarchy levels. All siblings along the path are shown as valuable context information. Distortion is used on the Y-axis.

ited to the categories within the current state of navigation in the hierarchy (referred to as "cut", see Fig. 1), but also includes all nodes above the cut up to the root of the hierarchy. This allows for direct comparison of properties between child nodes and parent nodes as both are displayed in the same visualization and thus share the same visual context with respect to node placement. However, this necessitates concepts for discriminating levels of the hierarchy and recognizing structural relationships, which we address in multiple ways.

First and foremost, lines connect each parent to all visualized children, thus representing the topology of the hierarchy. Small arrows pointing towards the respective child indicate the direction and greatly facilitate tracing the structure of the hierarchy. In order to improve the distinction of lines in densely populated areas, connection lines smoothly blend the color of the parent to the color of the child. As interesting aspect, these directed lines could be seen as "skeleton" of the visualization, which sketches the structure of the scatterplot of non-aggregated raw data entries. Even more important in the context of OLAP, the lines explicitly visualize the difference between the properties of each category with respect to its direct parent category (or the root of the hierarchy). Both the length and the angle have semantics, namely the overall amount of difference and the ratio. Due to the 2D layout, the lines support the perception of relationships between differences on the X and the Y axis. This allows for fast identification of sub-categories deviating in the same way from their parents for multiple sub-trees.

As mentioned above, each visual attribute can be used in different ways. In particular, each attribute can be used to enhance the discrimination of hierarchy levels, where transparency can be modulated independently from color. Transparency and size-based discrimination amount to a focus + context approach. One hierarchy level C is considered to be the current one, which is drawn opaque and in full size. Opacity and size decrease for lower and higher levels N with a factor of $1/2^{|C-N|}$. The current hierarchy level is a global property of the visualization, i.e., the same depth is highlighted through all sub-trees. Drill-down and roll-up operations automatically update the current level, or the user may manually set any level as current. Expanded nodes, i.e., nodes above the cut, are highlighted by an additional opaque circle. Directed lines leading towards expanded nodes are always drawn in full opacity (see Fig. 3), which facilitates tracing individual sub-trees as generated by local drill-down.

HDS offer various modes for coloring hierarchy nodes. In addition to representing common categorical properties like size or pivoted values of an arbitrary measure as mentioned above, users may optionally also emphasize the structure of the hierarchy. Hierarchy-based coloring recursively subdivides the hue circle in a similar way as described for the Interring [24]. The segment of the hue circle assigned to each node is proportional to the number of leaf-nodes in the sub-tree and the hue in the middle of the segment is applied to the node itself. Color is a particularly important issue when coupling different tree-visualizations, as it supports the visual matching of hierarchy nodes (see Section 3.2).

With an increasing number of displayed nodes, the extents and the density of the visualization may vary significantly. Restricting the displayed value range in a similar manner as in Spotfire [17] and also supported by our approach has the disadvantage that users may lose the overview. The entire hierarchy is not visible any more. As alternative, spatial distortion has proven useful to provide focus plus context for areas where nodes with similar properties lie close together. Applying a piecewise linear visual transfer function [3], the user may smoothly magnify any contiguous sub-interval of

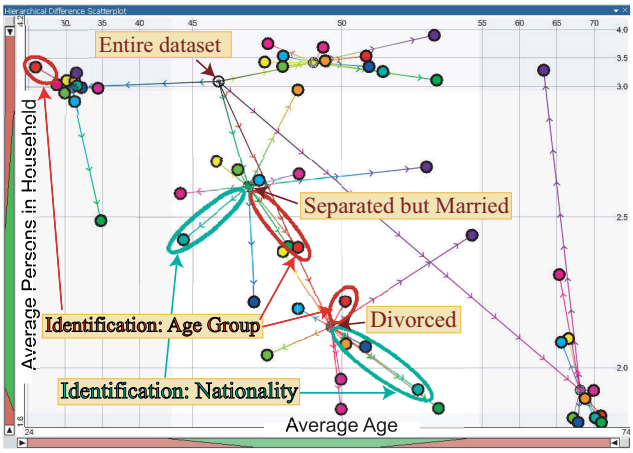


Figure 4: Comparing multiple sub-trees: interviewees are distinguished by their marital status and most important identification (in this order). Each class is characterized by its average age and the average number of persons in the household. While most identification nodes deviate roughly in the same direction for all marital status nodes, some interesting exceptions, like "Nationality", show contrary behaviour for different nodes. Color is derived from the category name. Spatial distortion is applied on both axes to focus on divorced and separated but married interviewees.

the displayed value range. The factor is chosen separately for the X- and Y-axis (see Fig. 3 and 4). The reason for using a piecewise linear function instead of using non-linear distortion (e.g., fish-eye distortion [10]) is that differences between nodes remain comparable as long as all involved nodes are inside the focus. This can easily be ensured by the user.

3.2 Coupling Tree Visualizations

Arguably, no single visualization approach perfectly covers all aspects of hierarchical data. The clear focus of HDS is on supporting the interactive analysis of data cubes in the context of OLAP. By displaying multiple pivoted values (or other properties) and the differences to parent levels at the same time, HDS visualize comparatively much information per node. Due to the data-centric layout, however, HDS do not perfectly scale to the visualization of both depth and breadth of large hierarchies at the same time (i.e., the hierarchy as a whole). This is due to well-known graph-drawing problems like a potentially high number of crossing edges. However, as discussed in section 6, this is not a limitation with respect to analyzing large real-world data cubes, because the user may increase the complexity incrementally and selectively by drilling down to interesting details while staying at a coarse level for less interesting sub-trees (or even hiding them).

Still, aspects conveyed not so well by HDS might be interesting. We therefore briefly discuss concepts of tightly coupling HDS to other approaches for visualizing hierarchies in order to combine their benefits when analyzing the same hierarchy. As an example, we have implemented a layout similar to parallel trees [16] as used by Sifer to analyze OLAP data. This layout is related to ArcTrees [11], which we refer to as hierarchical bargrams since we do not show any arcs. In hierarchical bargrams, a horizontal bar representing 100% of the displayed data is subdivided in proportion to the relative frequencies of the categories in the first level of the hierarchy. The obtained boxes are recursively split in proportion to the relative frequencies of their sub-categories. This generates bars nested inside the representation of their parent-category. Each bar displays the name of the respective node (see Fig. 5).

We have identified the following attributes for tightly coupling HDS to other kinds of tree visualizations.

- **State of navigation** The user may perform drill-down and roll-up operations in any visualization, which consistently updates all views. In the hierarchical bargrams, the recursion stops at the current cut, which is also conceivable for most other types of tree visualizations (like treemaps).
- **Color** As discussed above, HDS offer multiple ways for using color. Applying consistent coloring of nodes to all visualizations greatly facilitates the visual matching between them. In our case, the bars in the bargrams are drawn in the same color as the nodes in the HDS. Deriving the color from the position of nodes in the HDS (e.g., by mapping the position on the X-axis or the difference from the root to color) enhances the matching even more. Coupling by color is possible for almost all types of tree visualizations.
- **Order** Many tree visualizations have a degree of freedom in which order siblings are represented. This freedom can be used to roughly maintain proximities between nodes throughout all visualizations. The hierarchical bargrams, for example, optionally order sibling nodes with respect to their position on the X- or Y-axis in the HDS.
- **Selection** Interaction is generally very powerful for linking visualizations. We provide different types of selection: (1) based on dedicated mark up interactions (e.g., by drawing a rubber band or actively clicking on an item) (2) temporarily hovering over visual items, which highlights the node or sub-tree beneath the mouse cursor throughout all visualizations. This has turned out to be very intuitive and fast for matching nodes as no mouse clicks are needed.

3.3 Integrating Selected Subsets

The previous section discussed tightly coupling HDS to other tree visualizations. This section describes the integration of subsets as defined by brushing arbitrary multivariate visualizations like parallel coordinates. It also applies to linking multiple instances of HDS visualizing different hierarchies. Linking views by interactive queries has established itself as important concept, because different sub-tasks of a complex analysis typically require different types of visualization. For example, the user may want to identify multidimensional clusters in parallel coordinates, and immediately relate each cluster to a hierarchy as visualized by HDS.

In a linked setup, each type of visualization typically highlights the subset of selected entries in an appropriate way. In HDS, the integration is based on the fact that the selection state is categorical too. Each row in the underlying

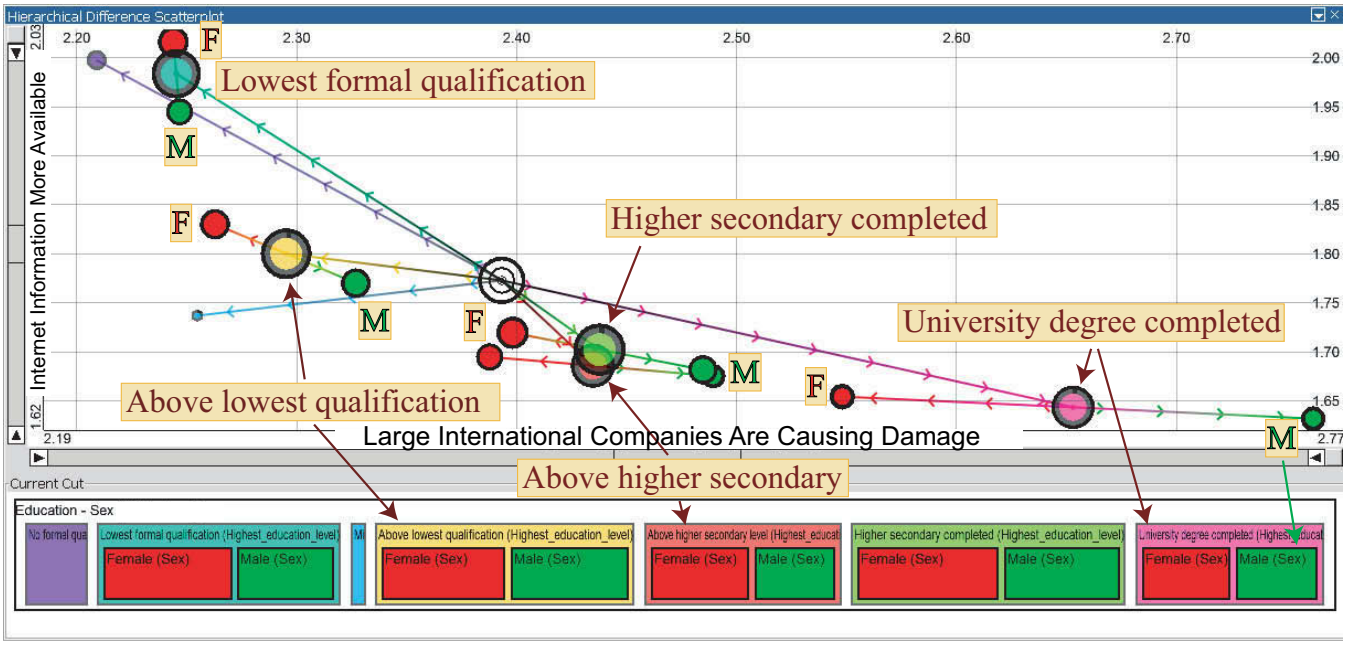


Figure 5: Tightly coupling HDS to hierarchical bargrams for displaying frequencies and names of hierarchy nodes. Several education levels, partly split into male (green) and female (red), are compared with respect to the average attitude towards international companies (X-axis, hue) and benefits of the Internet (Y-axis, saturation). The nonlinear relationship between the questions and the influence of education and sex are clearly visible.

non-aggregated main data table is either selected or not at any point in time with respect to a particular query. Employing the concept of dimension composition, a selection thus refines any hierarchy node X into "X and selected" and "X and not selected". This allows for visualizing selections similar to normal child nodes.

For each node X of the cut, the aggregations of the selected part of X (unless empty) are computed and visualized at the respective position in the plot (see Fig. 6). As for actual sub-categories, a line connecting the representations of the entire category X and its selected part explicitly represents the difference between both with respect to pivoted values. In order to discriminate multiple selections, the border of selection nodes is drawn in the color of the respective query, while this part is black for actual nodes of the hierarchy. Immediately updating the visualization at each modification of the selection implicitly generates an animation of change similar to moving the time slider of the Gapminder Trendalyzer [6]. In our case it concerns general variation on arbitrary data dimensions. The modification speed of each node representation reflects the gradient of change with respect to the selection criterion. As recently discussed by Robertson et al. [14], it also reveals overall trends, e.g., all selection nodes move from left to right, and makes outliers discernable, which move in a contrary direction.

4. IMPLEMENTATION AND USER INTERFACE

HDS have been implemented in the context of VISPLORE, an application framework for visually supported knowledge discovery in large and high-dimensional datasets. VISPLORE supports the analysis of datasets with millions of entries

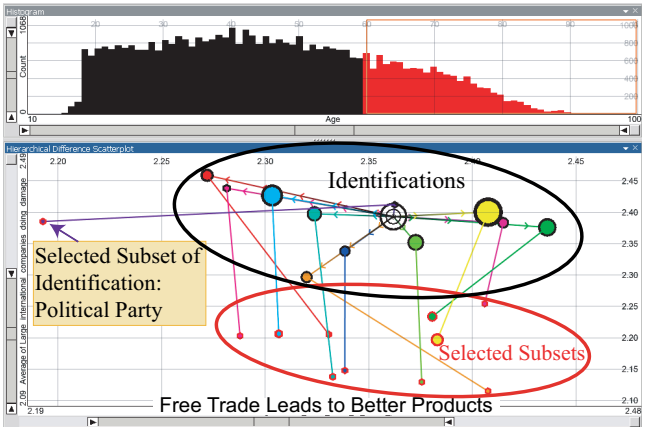


Figure 6: Integrating queries: interviewees older than 60 years, as brushed in a histogram, are highlighted for each category regarding most important identification with the average attitude towards free trade (X) and damage done by international companies (Y) assigned to the axes of the HDS. The visualization shows that elderly people tend to have an over-proportionally negative attitude towards international companies, while the attitude towards free trade is in most cases independent of age. The category "political party" is an exception, though, as the acceptance of free trade is higher for elderly people and – unlike for the other categories – more significant than the difference regarding international companies.

and hundreds of dimensions at interactive rates on consumer hardware. This has a major impact on the design of all views (including HDS) and necessitates advanced software techniques like multithreading. VISPLORE also supports missing values and requires all views to do so.

VISPLORE currently provides more than 10 different visualizations, which are partly standard (e.g., 2D and 3D scatter plots, parallel coordinates, histograms, etc.) and partly specific to certain application tasks [12]. A key aspect of VISPLORE is to discriminate multiple queries, which are defined by composite brushing and are highlighted by all views in a linked way. All components also offer convenience functionality like undo/redo and a consistent way to arrange controls like data dimensions of the current dataset. In particular, the user may at any time specify new hierarchies of arbitrary complexity by dimension composition or by combining categories. Data dimensions and hierarchies can easily be assigned to views, which is the way how the axes and the displayed hierarchy of HDS are parameterized.

Making the user interface easy-to-use was also an essential design aspect of HDS. The user may perform drill-down and roll-up operations by just clicking on a visual representation, or may hide entire sub-trees. Tool tips provide details-on-demand showing the name, the size, and the aggregated values for the node beneath the mouse cursor. In order to highlight subsets of the data in linked views, the user can brush nodes by either clicking on them or dragging a rubber band. Dedicated widgets next to the X- and Y-axis offer all functionality related to adapting the displayed value range and the spatial distortion.

5. CASE STUDY AND EVALUATION

We now discuss the evaluation of our approach by the interactive visual analysis of a large survey, which we did together with a sociologist. The analysis of opinion polls is an important topic, where too little attention has been devoted to. HDS are designed to be generally applicable to data cubes of any kind, e.g., business data as a typical application of OLAP, and are not limited to opinion poll data. The sociologist had rich experience with the analysis of surveys, but had used static statistical software and had never used interactive visualizations before.

The survey was conducted by the International Social Survey Programme (ISSP) [13] in 33 countries between February 2003 and January 2005 with 44.170 respondents in total. Disregarding country-specific and thus incomparable questions, the dataset consists of 104 predominantly categorical attributes. The attributes are partly demographic questions and partly concern the attitude towards national consciousness, identity, and pride. The answers to most questions comprise 4 or 5 levels, e.g., very proud, somewhat proud, not very proud, not proud at all. This allows for both treating them as categories as well as computing meaningful aggregations, e.g., the average accordance to a statement. The dataset contains missing values, which represent an own category for categorical attributes. Missing values are disregarded when aggregating a continuous attribute.

Before analyzing the questions regarding attitude and pride, the sociologist first wanted to gain an overview about characteristics of various demographic categories, figures of the survey, and potential relationships between them. HDS facilitate this task, as it is fast to visualize simple pivot tables like the average number of persons in a household per country and they also quickly provide the size of each category. Within a few minutes, the expert could look at dozens of combinations, partly confirming expected facts, e.g., the average age of widowed people is 22 years higher than the average of the dataset. Partly, this basic analysis already revealed unexpected features like a significant variance in the average age of interviewees throughout the countries (which must be taken into account for subsequent conclusions).

Already for such flat pivot tables, the sociologist appreciated very much being shown the average of the entire dataset as visual reference. The reason is that this reference is not affected by categories of different size - a common problem when trying to determine the centre in a purely visual manner (e.g., by assuming the centre of the image as centre of the data, which is typically misleading). As criticism regarding our implementation, the expert said that he lacked labels next to the nodes, although he admitted that tooltips partly compensate for that. We suggested using coupled bargrams as legend and deriving the order of the nodes from the X-position in the HDS. He made use of them for cases where only a few nodes are simultaneously shown, while they turned out to be of limited scalability for more complex hierarchies.

After analyzing cross tabulations between categories (a frequent task in sociology) in another visualization of our framework, the expert returned to HDS in order to characterize categories in the context of other categories. For example, he was interested whether different categories concerning identification have a similar distribution of age for different marital status categories (see Fig. 4). Showing multiple hierarchy levels simultaneously and explicitly representing the difference between them turned out to significantly help answering this and other comparatively complex questions. Within a short time, the sociologist identified multiple interesting and unexpected facts in the data. Comparing the difference vectors of the red dots in Figure 4, for example, reveals that for all categories related to marital status, the subset specifying "age group" as most important identification tends to be older than the average of the entire category. However, singles are a remarkable exception, as the "age group" subcategory of singles is the youngest of all. Assigning the same color to related sub-categories (e.g., red to all "age group" sub-categories) greatly facilitates such comparisons between different sub-trees. As the visualizations became more complex, the sociologist used distortion increasingly often and found it a convenient way to clarify relationships for densely populated areas.

As the next step of the analysis, the expert was interested in results concerning attitude and pride. Figure 5, for example, shows that people with a positive attitude towards the Internet turned out to be less sceptical towards large international companies. It further reveals a strong influence of the education level. For drill-down scenarios involving more hierarchy levels, the sociologist liked that he could focus on particular categories but still see the rest as context information, as illustrated by figure 3. While focussing

on Austrian interviewees with a university degree, still all other education levels are shown for Austria, all other European countries, and all continents. The centre of the entire dataset is given as well. The expert considered such deep local drill-downs a key advantage of HDS. Analyzing the difference between two levels is of course also possible by visualizing this derived information in simple scatterplots. Relating four or five levels at a time, however, would generate numerous derived data dimensions, which are hard to analyze intuitively without HDS.

The sociologist needed some time to familiarize with the idea of specifying ad-hoc categories by brushing linked visualizations. He eventually embraced this approach and used queries as defined in linked visualizations frequently for two types of tasks:

- **Motion** Due to the immediate update, changing the query in one view generates an animation in HDS. Figure 6 shows an example, where interviewees are selected by age in a histogram. Moving the interval from young towards old makes the selected parts of most identification classes in the HDS wander from top to bottom, indicating more scepticism towards international companies for elderly people. It also reveals interesting contrary trends for "political party" and "ethnic background" regarding the attitude towards free trade in dependence of age.
- **Highlighting** When comparing multiple sub-trees, a convenient way of identifying related categories throughout all shown extracted branches is by brushing this particular category in a linked view. For example, instead of assigning the same color to related subcategories in figure 4, it would also be possible to highlight all "age group" nodes by selecting the category "age group" in another view, e.g., in a second instance of HDS. Using the "Superfocus", the sociologist could identify many different categories in a short time. This was particularly useful when color was needed otherwise - for example to discriminate hierarchy levels as in figure 3.

Although we can only describe a small part of our analysis here, this application has demonstrated how HDS facilitate and speed up the interactive analysis of data cubes. As result of our evaluation, the sociologist particularly liked being shown the centre of the data as reference and being able to analyze multiple levels of the hierarchy in the context of each other. Despite tooltips and coupled hierarchy visualizations, his most important criticism concerned the lack of labels, which we will address in future work.

6. DISCUSSION AND FUTURE WORK

The main idea of HDS is to support the interactive visual analysis of data cubes. Selective drill down ensures that users can increase the amount of detail incrementally for sub-trees of interest. This is an important aspect regarding the scalability of HDS. As for all approaches relying on pivot tables, the speed for aggregating data is the most significant limitation with respect to the number of underlying data rows. Aggregating data is generally fast even for millions of data rows and may even make use of explicit optimizations for data cubes in data warehouses. Therefore, HDS scale well for data sets consisting of multiple millions (and even billions) of underlying data records, which makes them applicable to real world data cubes.

A relevant question concerns the amount of detail (i.e., how many hierarchy levels and how many nodes), that can be shown before the visualization suffers from cluttering. An answer depends on the purpose. Generally speaking, HDS are suitable for:

- comparisons *along the hierarchy*. The main intention is to relate a few particular nodes to their direct and indirect parent nodes. Such comparisons involve local drill downs of numerous hierarchy levels while typically little information is shown per level (see Fig. 3 for an example). In this case, the most interesting information is the path to the root node (i.e., the properties of the entire data cube). Siblings provide rather context information and it is often even tolerable to hide siblings for certain hierarchy levels. In this case, comparing more than ten hierarchy levels is possible.
- comparisons *across one hierarchy level*. The focus is on the position of siblings relative to each other and to common parent nodes (see Fig. 4 for an example). Much information is shown for a single hierarchy level while little information – if any – is typically shown for other levels. In this case, HDS resemble non-hierarchical scatter plots, but may still convey additional information (e.g., the properties of the entire data cube as one additional item). In this case, comparing a few hundred categories is possible.

As a consequence of displaying much information per node (i.e., two pivoted properties and topology), HDS are limited with respect to showing both depth and breadth of large hierarchies simultaneously. Showing large hierarchies in their entirety was not a design goal of HDS and it is not necessary for many tasks. As discussed in section 2, most tree visualizations convey the topology but disregard multi-variate attributes. Most approaches for OLAP, on the other hand, consider multiple attributes but are limited to displaying a single hierarchy level.

Tableau [1] optionally displays multiple hierarchy levels using sub totals and grand totals which are added as additional rows or columns. However, comparisons require looking at multiple places on the screen in a successive manner. Generally speaking, comparisons become increasingly difficult and less precise with increasing visual distance and number of visualizations involved in the comparison. For example, while detecting even minor differences in the height of two adjacent bars of a bar chart is easily possible, comparing the position of points of multiple non-adjacent scatterplot panes is difficult and coarse. The reason is that the user is forced to "remember" one pane while shifting his focus to another – potentially distant – pane (which might even involve scrolling the entire visualization). Fig. 7 illustrates this aspect. Although three panes (as shown in the lower half) is quite a small number, precise comparisons are particularly difficult with respect to the position on the X-axis. The figure also shows that using a single row makes comparisons with respect to height much easier, because the same vertical reference is given for all items. Using multiple rows (e.g., by assigning identification to rows instead of us-

Figure 7: Comparing hierarchy levels using HDS (upper half) and using multiple scatterplots in Tableau (lower half). The average age (X axis) and the average number of education years (Y axis) are shown for groups having different most important identifications (color), which are further subdivided by sex. The same colors are used for corresponding identifications in both halves. In HDS, "male" is drawn green and "female" red, while multiple panes are used below. Comparing especially the horizontal position of items is difficult across columns, while even minor differences are clearly conveyed by HDS.

ing color) would severely compromise comparability of the Y-position as well. This problem is inherent for approaches that do not explicitly visualize the difference between items but rely on showing multiple visualizations in a side-by-side manner. Drawing items in a single scatterplot does explicitly visualize the difference between them, as this difference is directly proportional to their distance. This was a main consideration in the design of HDS.

As future work, we will address the issue of labelling nodes as mentioned by the sociologist. The challenge is to add labels in a scalable way without compromising readability. Furthermore, we intend to examine the effect of varying the shape of node representations on the interpretability of the visualization.

7. CONCLUSION

The analysis of data cubes is a key issue in many application domains. It involves navigating a potentially large hierarchy as well as comparing nodes within one or between multiple hierarchy levels with respect to properties like size and pivoted values. Particularly the difference between hierarchy levels is important information, which is not adequately represented by existing visualization techniques. Therefore, this paper introduced Hierarchical Difference Scatterplots (HDS) as an interactive approach to analyze multiple hierarchy levels in the context of each other and to emphasize differences between them. Visualizing both the topology and two pivoted values per node, HDS display much information at a time. For many tasks, this means an added value as compared to alternative approaches. For example, analyzing differences between hierarchy levels using non-hierarchical scatterplots requires the user to look at multiple views (i.e., positions of the screen) in a successive manner. HDS display the difference between categories explicitly within one visualization, which makes comparisons more intuitive and more precise.

A key idea of HDS is to allow for incrementally and selectively increasing the amount of detail using local drill-down. This ensures that the proposed concept of HDS is reasonably applicable to data cubes of any size. HDS em-

ploy several focus plus context approaches involving transparency, size, and distortion in order to ensure interpretability also for a significant number of displayed nodes. As other tree-visualizations are superior with respect to providing a pleasant layout of the entire topology or showing frequencies, we discussed concepts of tightly coupling HDS to other tree visualizations. Moreover, we discussed linking arbitrary other visualizations by user-defined queries to HDS. This allows for analyzing properties of ad hoc categories, it reveals trends through animations when changing queries, and it may also be used to highlight particular nodes. We described an evaluation of our approach by analyzing a large survey, which revealed numerous interesting and non-trivial aspects within a short time.

8. ACKNOWLEDGMENTS

This work was done at the VRVis Research Center in Vienna, Austria, in the scope of the projects MUMOVI, ENGVIS, and AVISOM (Nr. 818060). Thanks go to Florian Spendlingwimmer for the sociological support with the case study and to Martin Brunnhuber for important parts of the implementation. Additional thanks go to Wolfgang Berger, Philipp Muigg, and Helmut Doleisch for help in preparing this paper. Finally, we want to thank our company partner AVL List GmbH for co-financing the application framework.

9. REFERENCES

[1] Tableau software. `http://www.tableausoftware.com`.

[2] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall PTR, 1998.

[3] S. Card, J. Mackinlay, and B. Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann Publishers Inc., 1999.

[4] E. F. Codd. Providing OLAP (On-line Analytical Processing) to User-Analysts, 1993.

[5] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *VIS '99: Proceedings of the 1999 IEEE Conf. on Visualization*, pages 43–508, 24-29 Oct. 1999.

[6] Gapminder Foundation. Gapminder. `http://www.gapminder.org/`.

[7] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. *Data Min. Knowl. Discov.*, 1(1):29–53, 1997.

[8] I. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, 2000.

[9] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748, 2006.

[10] J. Lamping, R. Rao, and P. Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 401–408, 1995.

[11] P. Neumann, S. Schlechtweg, and M. Carpendale. Arctrees: Visualizing relations in hierarchical data. In *Proceedings of Eurographics, IEEE VGTC Symposium on Visualization*, pages 53–60, 2005.

[12] H. Piringer, W. Berger, and H. Hauser. Quantifying and comparing features in high-dimensional datasets. *International Conference on Information Visualisation (IV08)*, 0:240–245, 2008.

[13] I. S. S. Programme. National Identity II. `http://zacat.gesis.org`, 2003.

[14] G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko. Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1325–1332, 2008.

[15] B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Trans. Graph.*, 11(1):92–99, 1992.

[16] M. Sifer. User interfaces for the exploration of hierarchical multi-dimensional data. In *VAST '06: Proceedings of the 2006 IEEE Symposium On Visual Analytics Science And Technology*, pages 175–182, 2006.

[17] Spotfire Inc. Spotfire. `http://spotfire.com/`.

[18] C. Stolte, D. Tang, and P. Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):52–65, 2002.

[19] C. Stolte, D. Tang, and P. Hanrahan. Query, analysis, and visualization of hierarchically structured data using polaris. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 112–122, New York, NY, USA, 2002. ACM Press.

[20] C. Stolte, D. Tang, and P. Hanrahan. Multiscale visualization using data cubes. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 9(2):176–187, 2003.

[21] E. R. Tufte. *The visual display of quantitative information*. Graphics Press, Cheshire, CT, USA, 1986.

[22] M. O. Ward. A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization*, 1(3/4):194–210, 2002.

[23] M. Wattenberg. Visual exploration of multivariate graphs. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 811–819. ACM, 2006.

[24] J. Yang, M. O. Ward, and E. A. Rundensteiner. Interring: An interactive tool for visually navigating and manipulating hierarchical structures. In *INFOVIS '02: Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, pages 77 – 84, 2002.

[25] S. Zhao, M. J. McGuffin, and M. H. Chignell. Elastic hierarchies: Combining treemaps and node-link diagrams. In *INFOVIS '05: Proceedings of the IEEE Symposium on Information Visualization*, pages 57–64. IEEE Computer Society, 2005.

Visual Analysis of Documents with Semantic Graphs

Delia Rusu, Blaž Fortuna, Dunja Mladenić, Marko Grobelnik, Ruben Sipoš
Department of Knowledge Technologies
Jožef Stefan Institute, Ljubljana, Slovenia
{delia.rusu, blaz.fortuna, dunja.mladenic, marko.grobelnik, ruben.sipos}@ijs.si

ABSTRACT

In this paper, we present a technique for visual analysis of documents based on the semantic representation of text in the form of a directed graph, referred to as *semantic graph*. This approach can aid data mining tasks, such as exploratory data analysis, data description and summarization. In order to derive the semantic graph, we take advantage of natural language processing, and carry out a series of operations comprising a pipeline, as follows. Firstly, named entities are identified and co-reference resolution is performed; moreover, pronominal anaphors are resolved for a subset of pronouns. Secondly, subject – predicate – object triplets are automatically extracted from the Penn Treebank parse tree obtained for each sentence in the document. The triplets are further enhanced by linking them to their corresponding co-referenced named entity, as well as attaching the associated WordNet synset, where available. Thus we obtain a semantic directed graph composed of connected triplets. The document's semantic graph is a starting point for automatically generating the document summary. The model for summary generation is obtained by machine learning, where the features are extracted from the semantic graph structure and content. The summary also has an associated semantic representation. The size of the semantic graph, as well as the summary length can be manually adjusted for an enhanced visual analysis. We also show how to employ the proposed technique for the Visual Analytics challenge.

Categories and Subject Descriptors

I.2.7 [**Natural Language Processing**]: *Text analysis*.

General Terms

Algorithms, Design.

Keywords

Natural language processing, text mining, document visualization, semantic graph, triplet, summarization.

VAKD'09, June 28, 2009, Paris, France.

1. INTRODUCTION

Visual Analytics incorporates, among others, knowledge discovery, data analysis, visualization and data management. The goal of this research field is to derive insight from dynamic, massive, ambiguous and often conflicting data [6]. Providing visual and interactive data analysis is a key topic in Visual Analytics. Data mining tasks, such as data description, exploratory data analysis and summarization can be aided with such visualizations. Visual exploration and analysis of documents enables users to get an overview of the data, without the need to entirely read it. The document overview offers a straightforward data visualization by listing the main facts, linking them in a way that is meaningful for the user, as well as providing a document summary.

In order to respond to this challenging task, we present a document visualization technique based on semantic graphs derived from subject – predicate – object triplets using natural language processing. This technique can be applied for providing documents and their associated summary with a graphical description that enables visual analysis at the document level. The triplets are automatically extracted from the Penn Treebank [10] parse tree which was generated for each sentence in the document. They are further processed by assigning their co-referenced named entity, by solving pronominal anaphors for a subset of pronouns and by attaching their corresponding WordNet [3] synset. Finally, the semantic graph is built by merging the enhanced triplets.

Moreover, this semantic representation is not only useful for visualizing the document, but it also plays an important part in deriving the document summary (as proposed in [7, 12]). This is obtained by classifying sentences from the initial text, where the features are extracted from the document and its semantic graph. The size of the semantic graph, as well as the summary length are not fixed, and this characteristic improves visual analysis. Furthermore, the document summary is also provided with a semantic graphical representation.

There are several tools dedicated to document corpus visualization, which are helpful in data analysis. Some are focused on a particular kind of data, such as news collections [5], other are developed for general text either based solely on the text of the documents in the corpus [4], or ontology-based, taking advantage of, for example, an ontology representing the users' knowledge or interests [14]. While these approaches explore and analyze a collection of documents as a whole, providing the overall picture of the text corpus, we perform a more in depth visual exploration and analysis of a single document.

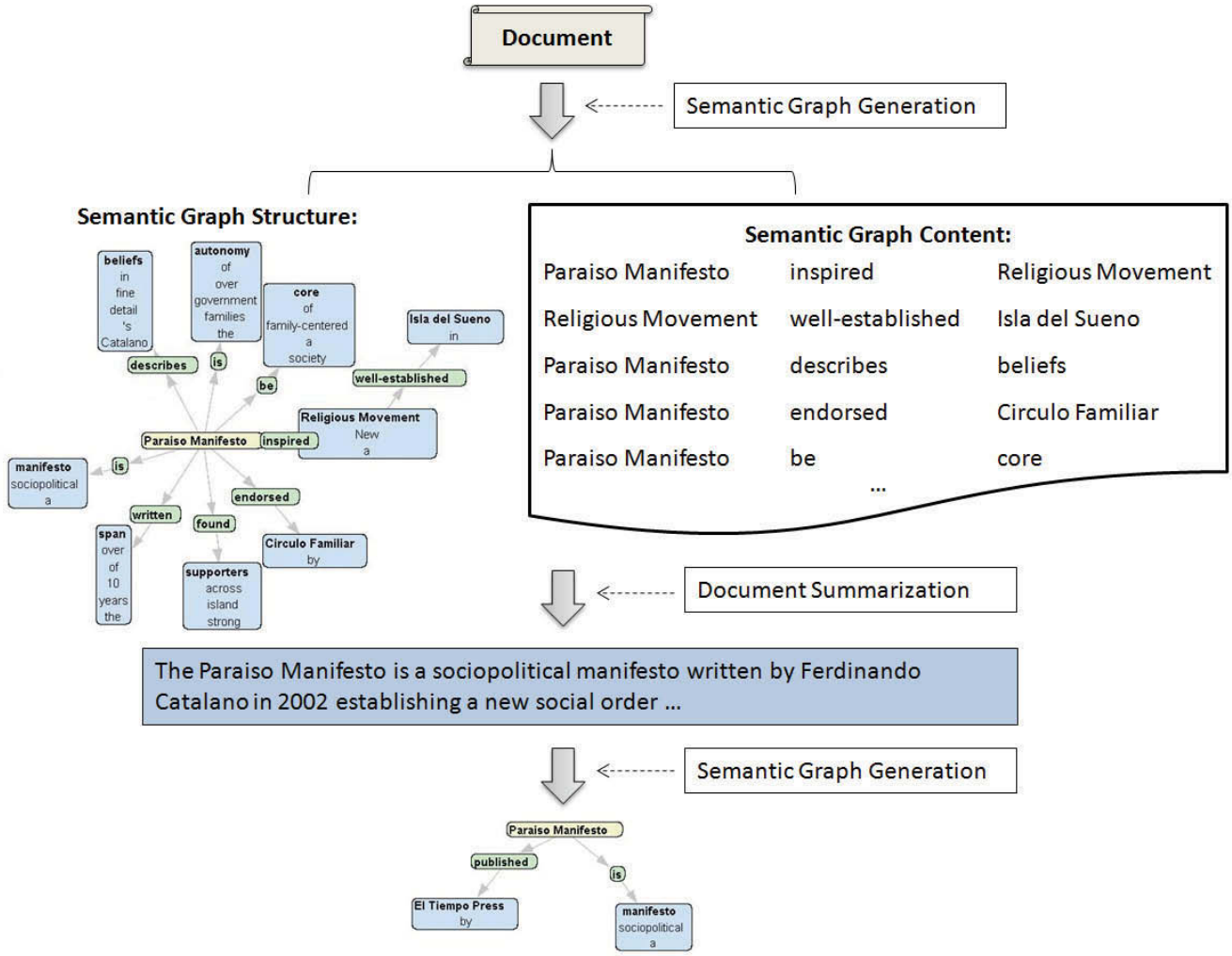


Figure 1. The document analysis process.

Moreover, we detail the main facts and connect them in a semantic structure, as well as provide a visual description for the document summary. Other visualization tools focus on tracking of story evolution: evolutionary theme patterns discovery, summary and exploration. The work described in [13] takes advantage of graphs to represent the development of a story; these graphs consist of elements of a co-occurence network, disregarding synonymy relationships among the elements. In contrast to this approach, we construct a semantic graph where the building blocks are enhanced triplets linked to WordNet synsets. These triplets are much more than co-occuring entities, they are considered the core of the sentence, salient enough to carry the sentence message. They are connected taking into account the synonymy relationship among them.

Previous work related to visualizing a single document has focused on highlighting named entities, facts and events in the given text, or on using the human created structure in lexical databases for revealing concepts within a document. Our purpose is to further analyze the concepts in a text fragment, determine the connections among them and visually represent this in the form of a semantic graph, either of the entire document or of its summary.

The Calais Document Viewer[1] creates semantic metadata for the user submitted text, in the form of named entities, facts and events, which are highlighted and navigable; the RDF output can also be viewed and captured for analysis in other tools. DocuBurst [1] presents the concepts within a document in a radial, space filling tree structure, using WordNet's IS-A hyponymy relationship. In the case of our system, named entities and facts or concepts represent the starting point; they are further refined in order to enable the construction of a semantic description of the document in the form of a semantic directed graph. The nodes are the subject and object triplet elements, and the link between them is determined by the predicate. The initial document, its associated facts and semantic graph are then employed to automatically generate a summary, which can also be visualized in the form of a graph.

[1] Calais url: http://www.opencalais.com/

The paper is organized as follows. We start with an overview of the document visualization process in Section 2, continuing with a description of semantic graphs in Section 3, document summaries in Section 4 and an application of the described technique to the Visual Analytics challenge in Section 5. The paper concludes with several remarks.

2. DOCUMENT VISUAL ANALYSIS

The document visualization process is described in Figure 1, where an example document from the Visual Analytics Challenge[2] is used.

It starts with the original document, which is further processed and refined in order to obtain the set of subject – predicate – object triplets as well as its associated semantic graph. Next, the semantic graph structure and content serve as input for the document summarizer, which automatically generates a summary of sentences from the text. The approach considered for summarization is sentence extraction. This summary can also be visualized in the same way as the original document, by associating it with a semantic description.

3. SEMANTIC GRAPHS

The *semantic graph* corresponds to a visual representation of a document's semantic structure. As proposed in [7] a document can be described by its associated semantic graph, thus providing an overview of its content. The graph is obtained after processing the input document and passing it through a series of sequential operations composing a pipeline (see Figure 2):

- *Text preprocessing*: splitting the original document into sentences;
- *Named entity extraction*, followed by named entity co-reference resolution and pronominal anaphora resolution;
- *Triplet extraction* based on a Penn Treebank parser;
- *Triplet enhancement* by linking triplets to named entities and semantic normalization via assigning each triplet its WordNet synset;
- *Triplet merger into a semantic graph* of the document.

In what follows, we are going to further detail the aforementioned pipeline components as proposed in [12].

3.1 Named Entity Extraction, Co-reference and Anaphora Resolution

The term *named entities* refers to names of people, locations or organizations, yielding semantic information from the input text. For named entity recognition we consider GATE (General Architecture for Text Engineering)[3]; it was used as a toolkit for natural language processing. For people we also store their gender, whereas for locations we differentiate between names of cities and of countries, respectively. This enables co-reference resolution, which implies identifying terms that refer to the same entity. It is achieved through consolidating named entities, using text analysis and matching methods.

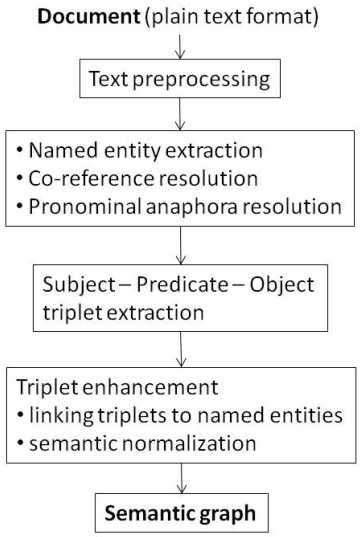


Figure 2. The semantic graph generation pipeline.

We match entities where one surface form is completely included in the other (for example "Ferdinando Catalano" and "Catalano"), one surface form is the abbreviation of the other (for example "ISWC" and "International Semantic Web Conference"), or there is a combination of the two situations described above (for example "F. Catalano" and "Ferdinando Catalano").

Figure 3 represents an excerpt of a document with two annotated named entities and their corresponding co-reference (we eliminate stop words when resolving co-references – for example in the case of "EBay Inc", "Inc" will be eliminated, as it is a stop word).

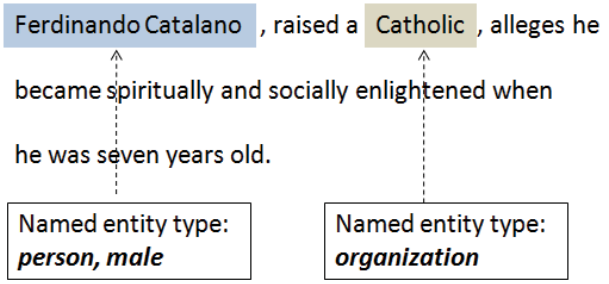


Figure 3. A document excerpt with two annotated named entities (a person and an organization).

We highlight the named entities found within the document, distinguishing between the three different entity types: people, locations and organizations, as illustrated in Figure 3.

Moreover, we resolve anaphors for a subset of pronouns: {*I, he, she, it, they*}, and their objective, reflexive and possessive forms, as well as the relative pronoun *who*. For solving this task, we take advantage of the co-referenced named entities and try to identify, for each pronoun belonging to the considered subset, its corresponding named entity. In the previous example (see Figure

[2] IEEE VAST 2008 Challenge url: http://www.cs.umd.edu/hcil/VASTchallenge08/tasks.html

[3] GATE url: http://gate.ac.uk/

3), the person named entity ("Ferdinando Catalano") would be a good candidate to replace the pronoun "he".

The pronominal anaphora resolution heuristic can be described as follows. We start by identifying pronouns in the given document, and search for each pronoun possible candidates that could replace it. The candidates receive scores, based on a series of antecedent indicators (or preferences) [12]: givenness, lexical reiteration, referential distance, indicating verbs and collocation pattern preference. The candidate with the highest score is selected as the pronoun replacement.

3.2 Triplet Extraction

We envisage the "core" of a sentence as a *triplet* consisting of the *subject*, *predicate* and *object* elements and assume that it contains enough information to express the message of a sentence. The usefulness of triplets resides in the fact that it is much easier to process them instead of dealing with very complex sentences as a whole.

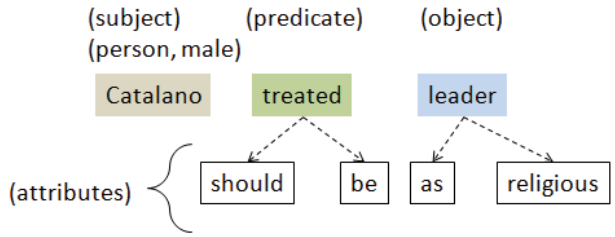


Figure 4. A triplet (Catalano – treated – leader) extracted from the sentence "*Followers claim the Paraiso Manifesto has inspired a New Religious Movement and Catalano should be treated as a religious leader.*".

Triplets are extracted from each sentence independently, without taking text outside of the sentence into account. We apply the algorithm for obtaining triplets from a Penn Treebank parser output described in [11], and employ the statistical Stanford Parser[4]. The extraction is performed based on pure syntactic analysis of sentences. The rules are built by hand, and use the shape of the parse tree to decide which triplets to extract. Figure 4 shows a triplet (*Catalano – treated – leader*) extracted from the sentence *"Followers claim the Paraiso Manifesto has inspired a New Religious Movement and Catalano should be treated as a religious leader."*. Aside from the main triplet elements (subject, predicate, object), the image also depicts the predicate and object attributes (*should*, *be* and *as religious*) – these are the words which are linked to the predicate and object in the parse tree.

As in the case of named entities, triplets are also highlighted differently, according to the triplet element type: subject, predicate or object. This convention is kept in the next phase of the pipeline, when building the semantic graph. Therefore, the triplet elements are much easier to identify within the graph structure.

3.3 Triplet Enhancement and Semantic Graph Generation

The semantic graph is utilized in order to represent the document's semantic structure. Our approach is based on the research presented in [7] and further developed in [12]. While in [7] semantic graph generation was relying on the proprietary NLPWin linguistic tool [2] for deep syntactic analysis and pronominal reference resolution, we take advantage of the co-referenced named entities as well as the triplets extracted from the Penn Treebank parse tree and derive rules for pronominal anaphora resolution and graph generation. For generating the graph, triplets are first linked to their associated named entity (if appropriate). Furthermore, they are assigned their corresponding WordNet synset. This is a mandatory step, preceding the semantic graph generation, as it enables us to merge triplet elements which belong to the same WordNet synset, and thus share a similar meaning. Hence we augment the compactness of the graphical representation, and enable various triplets to be linked based on a synonymy relationship. We obtain a directed semantic graph, the direction being from the subject node to the object node, and the connecting link (or relation) is represented by the predicate.

Figure 5 presents a semantic sub-graph of a text excerpt. Semantic graph visualization was achieved through adapting the Prefuse visualization toolkit[5] in a Java applet embedded in the web page. The graph layout is a dynamic force-directed one, yielding a spring graph, scalable to several hundred nodes.

The semantic graph generation system components were evaluated by comparing their output with the one of similar systems, as described in [12]. The evaluation was performed on a subset of the Reuters RCV1 [8] data set. For co-reference resolution, the comparison was made with GATE's co-reference resolver; our co-reference module performed about 13% better than GATE. In the case of anaphora resolution, we compared the outcome of our system with two baselines that considered the closest named entity as a pronoun replacement: one baseline also took gender information into account, whereas the other did not. We obtained good results, particularly in the case of the masculine pronoun *he.*

4. DOCUMENT SUMMARY

The document summary is a means of retrieving a more synthetic text by extracting sentences from the original document. This is automatically obtained starting from the initial document and its corresponding semantic representation. The technique involves training a linear SVM classifier to determine those triplets that are useful for extracting sentences which will later compose the summary. The features employed for learning are associated with the triplet elements and obtained from the document content (linguistic and document attributes) and from the graph structure (graph attributes) [12]. Table 1 lists several examples of features that were used for learning. All in all, there are 69 features distributed among the triplet elements: 26 for the subject, 11 for the predicate, 26 for the object, and 6 sentence attributes (associated to the sentence that generated the triplet).

[4] Stanford Parser url: http://nlp.stanford.edu/software/lex-parser.shtml

[5] Prefuse url: http://prefuse.org/

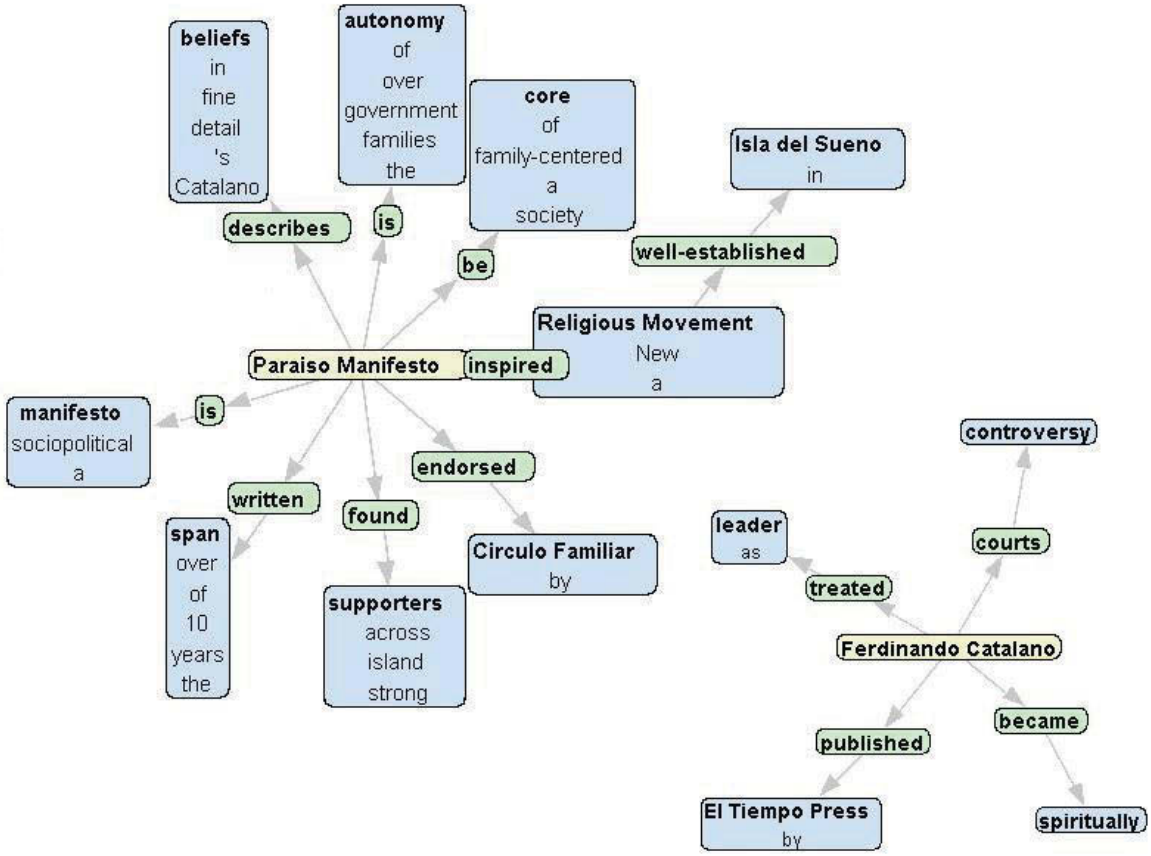


Figure 5. A semantic sub-graph of the "Paraiso Manifesto" Wikipedia article provided by the Visual Analytics Challenge.

Table 1. Examples of features used for learning.

Feature categories	Examples
Linguistic	• the triplet type: subject, predicate or object, • the Penn Treebank tag, • the depth of the linguistic node extracted from the Penn Treebank parse tree, • the part of speech tag;
Document	• the location of the sentence within the document, • the triplet location within the sentence, • the frequency of the triplet element, • the number of named entities in the sentence, • the similarity of the sentence with the centroid (the central words of the document);
Graph	• hub and authority weights, • page rank, • node degree, • the size of the weakly connected component the triplet element belongs to;

For training the linear SVM model and for the evaluation of the document summary, we utilize the DUC (Document Understanding Conferences)[6] datasets from 2002 and 2007, respectively, and compare the results with the ones obtained in the 2007 update task, as described in [12]. Thus we can compare the performance of our system with similar summarization applications that participated in the DUC 2007 challenge, for example, that generate compressed versions of source sentences as summary candidates and use weighted features of these candidates to construct summaries [9], or that learn a log-linear sentence ranking model by maximizing three metrics of sentence goodness [15].

The DUC datasets contain news articles from various sources like Financial Times, Wall Street Journal, Associated Press and Xinhua News Agency. The 2002 dataset comprises 300 newspaper articles on 30 different topics and for each article we have a 100 word human written abstract. The DUC 2007 dataset comprises 250 articles for the update task and 1125 articles for

[6] DUC url: http://duc.nist.gov/

the main task, part of the AQUAINT dataset[7]; the articles are grouped in clusters and 4 NIST assessors manually create summaries (of 100 or 250 words) for the documents in the clusters. As training data we used the DUC 2002 articles, as well as the DUC 2007 main task articles, while the DUC 2007 update task articles were used for testing. We extracted triplets from the training and test data, and learned which triplets appear in the summaries. If we order the classified triplets by the confidence weights of their class we obtain a ranked list of triplets. In order to build the summary of a document, we trace back the sentences from which the triplets were extracted.

The summarization process, described in Figure 6, starts with the document semantic graph. The three types of features abovementioned are then retrieved. Further, the triplets are classified with the linear SVM, and then the sentences from which the triplets are extracted are identified, thus obtaining the document summary. The summary length is interactively determined by the user, for an enhanced visual analysis. As sentences have an associated SVM score (the one of the triplets extracted from the sentence), the summary will be composed of those sentences that received the highest score. In order to keep the summary readable, we maintain the same sentence ordering that appears in the original text. Because the training data was formed of newspaper articles, we expect the results to generalize to other news corpora, as well as Wikipedia articles.

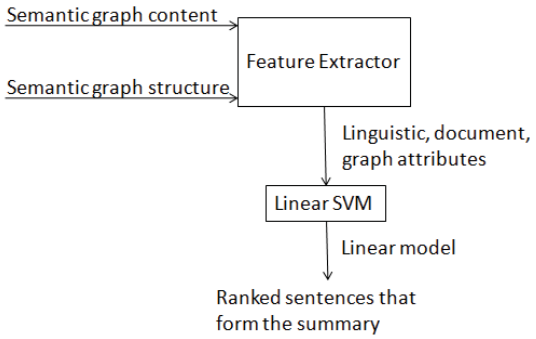


Figure 6. The summarization process.

5. VISUAL ANALYTICS CHALLENGE

The Visual Analytics challenge proposes a set of 4 mini-challenges, which combined form an overall challenge, concerning a fictitious, controversial socio-political movement. The datasets provided to solve the challenges are synthetic: a blend of computer- and hand-generated data. As a starting point, the organizers offer background material for both the overall challenge and the individual challenges in the form of a Wikipedia article page accompanied by discussions related to the article content.

[7] AQUAINT url: http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2008T25

The visual analysis techniques that we have described throughout the paper are very useful in exploring the documents provided as a starting point, that is, the Wikipedia article describing the "Paraiso Manifesto", as well as the discussion page. By applying the pipeline components to these documents, we get an insight on the main issues mentioned in the text. The list of facts for the two documents, their associated semantic graphs and document summaries offer a good starting point for data analysis.

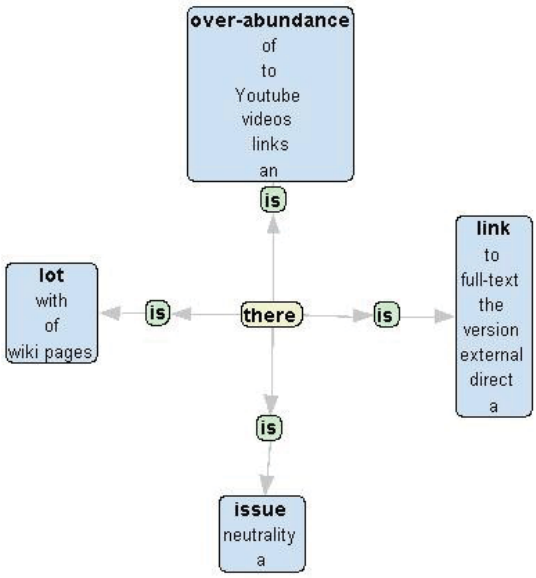


Figure 7. A semantic sub-graph generated from the Wikipedia Discussion page, under "POV Pushing".

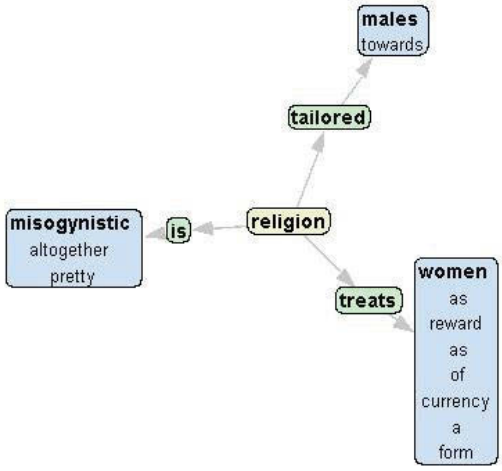


Figure 8. A semantic sub-graph generated from the Wikipedia Discussion page, under "Distinctive Doctrines".

We have shown a sub-graph obtained from the Wikipedia article describing the "Paraiso Manifesto" movement (see Figure 5). The other sub-graph is centered on the "Ferdinando Catalano"

node (not shown entirely in the figure). We have also generated semantic graphs for the Wikipedia discussion page on the "Paraiso Manifesto". We also show two sub-graphs obtained by processing the text under "POV Pushing" and "Distinctive Doctrines" (Figure 7 and Figure 8 respectively). Figure 9 shows an example of two document summaries, generated based on the Wikipedia Discussion page. The first two-sentence long summary was obtained from the "POV Pushing" sub-section, whereas the latter summary corresponds to the "Distinctive Doctrines" sub-section.

Thus we can use our system as a first step in solving either the overall challenge or the sub-challenges, by analyzing the documents provided as background material.

Wikipedia Discussions Page, under "POV Pushing"
2 sentence-long summary:

> There also is an over-abundance of links to Youtube videos produced by Catalano's Pirate Radio programs, this is unsavoury.
> Disciples of any religious group will lie and twist the truth to keep up appearances and that is what is happening on this Wiki page dedicated to Catalano.

Wikipedia Discussions Page, under "Distinctive Doctrines"
2 sentence-long summary:

> The religion treats women as a form of currency or a reward, not giving them any say in who they will marry, where they will live, and how many children they will have.
> However, the women believe they are given to the men but what the men do is none of their business.

Figure 9. Two document summaries obtained by processing parts of the Wikipedia discussion page.

6. CONCLUSIONS

In this paper we presented a document visualization technique based on semantic graphs. We showed that this technique can be applied not only to the original document, but also to its automatically generated summary. Each of the system components were detailed, starting with the semantic graph generation pipeline composed of named entity recognition, triplet extraction and enhancement, semantic graph construction, and concluding with the document summarization process. The runtime of our approach mainly depends on the document size and sentence complexity. The main bottleneck is represented by sentence parsing, which we intend to overcome by using a faster parser.

Regarding future improvements, we aim at extending the system by adding several components such as a more sophisticated named entity recognizer module, and a new triplet extraction module. To further refine the document overview through semantic graphs and summaries, we intend to integrate external resources that would enhance the semantic representation, as well as the document summary.

Based on the feedback obtained from several users we can conclude that the presented document visualization using semantic graphs is promising and can be helpful for the user. However, more experiments evaluating the usefulness of the proposed visualization approach are needed for firm conclusions.

7. ACKNOWLEDGMENTS

This work was supported by the Slovenian Research Agency and the IST Programme of the EC SMART (IST-033917) and PASCAL2 (IST-NoE-216886).

8. REFERENCES

[1] Collins, C. 2006. DocuBurst: Document Content Visualization Using Language Structure. In Proceedings of IEEE Symposium on Information Visualization, Poster Session. Baltimore.

[2] Corston-Oliver, S.H. and Dolan, B. 1999. Less is more: eliminating index terms from subordinate clauses. In Proceedings of the 37th Conference on Association for Computational Linguistics, College Park, Maryland.

[3] Fellbaum, Ch. 1998. WordNet: An Electronic Lexical Database. MIT Press.

[4] Fortuna, B., Grobelnik, M and Mladenić, D. 2005. Visualization of Text Document Corpus. Informatica Journal 29, pp. 270-277.

[5] Grobelnik, M. and Mladenić, D. 2004. Visualization of news articles. Informatica Journal 28, pp. 375-380.

[6] Keim, D. A., Mansmann, F., Schneidewind J., and Ziegler, H. 2006. Challenges in visual data analysis. In Proceedings of IEEE Interntional Conference on Information Visualization, pages 9 -16.

[7] Leskovec, J., Grobelnik, M. and Milic-Frayling, N. 2004. Learning Sub-structures of Document Semantic Graphs for Document Summarization. Workshop on Link Analysis and Group Detection (LinkKDD) at KDD 2004 (Seattle, USA, August 22 – 25, 2004).

[8] Lewis, D. D., Yang, Y., Rose, T. G., Li, F. 2004. RCV1: A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research, Vol. 5.

[9] Madnani, N., Zajic, D., Dorr, B., Ayan, N. F. and Lin, J. 2007. Multiple Alternative Sentence Compressions for Automatic Text Summarization. In Proceedings of the Document Understanding Conference (DUC).

[10] Marcus, M. P., Santorini, B.and Marcinkiewicz, M. A. 1993. Building a large annotated corpus of English: the Penn Treebank. Computational Linguistics, Volume 19.

[11] Rusu, D., Dali, L., Fortuna, B., Grobelnik, M. and Mladenić, D. 2007. Triplet Extraction from Sentences. In Proceedings of the 10th International Multiconference "Information Society - IS 2007" (Ljubljana, Slovenia, October 8 – 12, 2007). 218 – 222.

[12] Rusu, D., Fortuna, B., Grobelnik, M. and Mladenić, D. 2009. Semantic Graphs Derived From Triplets With Application In Document Summarization. Informatica Journal.

[13] Subašić, I. and Berendt, B. 2008. Web Mining for Understanding Stories through Graph Visualisation. In Proceedings of the International Conference on Data Mining (ICDM), pp. 570-579.

[14] Thai, V, Handschuh, S. and Decker, S. 2008. IVEA: An information visualization tool for personalized exploratory document collection analysis. In Proceedings of the European Semantic Web Conference (ESWC), pp. 139-153.

[15] Toutanova, K., Brockett, C., Gamon, M., Jagarlamudi, J., Suzuki, H. and Vanderwende, L. 2007. The PYTHY Summarization System: Microsoft Research at DUC2007. In Proceedings of the Document Understanding Conference (DUC).

Algebraic Visual Analysis: The Catalano Phone Call Data Set Case Study

Anna A. Shaverdian
Department of EECS
University of Michigan
annaas@umich.edu

Hao Zhou
Department of Statistics
University of Michigan
zhouhao@umich.edu

George Michailidis
Department of Statistics
University of Michigan
gmichail@umich.edu

H. V. Jagadish
Department of EECS
University of Michigan
jag@umich.edu

ABSTRACT

While many clever techniques have been proposed for visual analysis, most of these are "one of" and it is not easy to see how to combine multiple techniques. We propose an algebraic model capable of representing a large class of visual analysis operations on graph data. We demonstrate the value of this model by showing how it can simulate the analyses performed by several groups on the Catalano family cell phone call record data set as part of the VAST 2008 challenge.

1. INTRODUCTION

As visual analytics has gained importance as a field, there have been many impressive systems constructed and many clever techniques invented to support visual analysis of large data sets. From an application perspective, the ultimate measure of any technique or system has to be how effective it is in the context for which it is designed – does it support the derivation of the desired analytical results. While such a holistic measure may be the ultimate objective, from an engineering perspective, it is useful to break this down into pieces. Perhaps there are aspects of multiple systems that are each superior in their own way – how can we maximize learning from other systems and integration of novel techniques from multiple projects.

To enable this sort of integration, we propose an algebra for visual analysis, with a small number of fundamental operations. The design of specific systems can then be viewed as supporting specific expressions in this algebra. We can mix and match ideas from multiple projects by manipulating these algebraic expressions. Furthermore, we can devise new analysis path by making (often small) changes to these algebraic expressions that are harder to devise at the system level without the algebraic abstraction.

The set of operations in the algebra depends very much on the type of data to be analyzed. We restrict our attention to data that is naturally represented as a graph, with attributes on nodes and on edges. We describe our data model in Sec. 2.

Given a very large graph, the primary impediment to its visual analysis is size. There are two major ways in which size can be reduced, selection (retain only nodes/edges that satisfy a specified predicate) and aggregation (merge nodes/edges that are in some equivalence class). In Sec. 3, we develop an algebra that formally specifies these operators, and a few additional required "housekeeping" operators.

In Sec. 4, we demonstrate the value of this algebra by showing how it can represent the analyses performed by several researchers on one of the problems in the VAST 2008 challenge [4, 14]. Additional analyses enabled in our algebraic framework are illustrated in Sec. 5.

VAKD'09, June 28, 2009, Paris, France.

2. MODEL

We start by defining the appropriate data structure: let $\{\mathcal{D}(t) = [G(t), X(t)]\}_{t\in\mathcal{T}}$ denote a collection of graphs and their attributes, indexed by a finite set $\mathcal{T}$; in the case of the motivating application the index set corresponds to time and $\mathcal{T} = \{1, 2, \cdots, 10\}$. Further, $G(t) = (V(t), E(t), A(t))$ is the observed graph at time t, with node set $V(t)$, edge set $E(t)$ and weighted adjacency matrix $A(t)$. The associated attribute structure $X(t)$ is comprised of three components: $X(t) = [X_V(t),\ X_E(t),\ X_G(t)]$, where $X_V(t)$ contains node attributes (e.g. in- and out-degree in the motivating application), $X_E(t)$ edge attributes (e.g. edge betweenness) and $X_G(t)$ graph attributes (e.g. diameter). It should be noted that node attributes can be either intrinsic or computed. For example, geographic location or educational level of the nodes correspond to intrinsic attributes, whereas degree or clustering coefficient are computed ones. The same applies to edge and graph attributes.

In order to obtain computed attributes, it is assumed that there exists a collection of functions $\mathcal{F} = \{\mathcal{F}_X, \mathcal{F}_G\}$, relevant to the visual analytics problem at hand. Functions in class F_X are used for computing quantities of interest from the intrinsic attributes. Such functions are defined as follows: $f \in \mathcal{F}_X : \mathbb{R}^{|V|} \to \mathbb{R}^q$, with $1 \le q \le |V|$, where $|\cdot|$ denotes the cardinality of the underlying set. Examples of such functions include sorting an attribute, where $q = |V|$, calculating the max or the min of the attribute $q = 1$, quantizing (binning) the attribute in which q corresponds to the number of prespecified bins, etc.

3. ALGEBRA

We start by introducing the main operators in the algebra, which include aggregation $\mathcal{A}$ and selection $\mathcal{S}$. All operators in this algebra

take a set of graphs as input and produce a set of graphs as output. Therefore, the algebra is closed with respect to the operators we define, and compositions of primitive operators can be used to construct compound operators (or expressions) in this algebra. In our analysis of cases studies (see Section 4) we argue that *most* visual analytics tasks can be captured by expressions in this algebra.

Set Selection: Given a collection of graphs and their attributes, $\mathcal{D}$, a set selection applied to it, based on a predicate α, is written as $\sigma_\alpha(\mathcal{D}) = \{G \in \mathcal{D} | \alpha(G) = TRUE\}$. Observe that the selection predicate is evaluated independently for each element graph of the set, and the entire graph is either retained or discarded depending on the truth value of this predicate.

Element Selection: This is a basic filter based on a selection predicate that is applied to individual components of each element graph in the given collection of graphs. The cardinality of the collection remains unchanged, but each element in the collection is potentially reduced to a smaller graph. Recall that a graph may have a set of attributes $X = [X_V,\ X_E,\ X_G]$. An element selection $\mathcal{S}$ takes as argument a predicate on either X_V or X_E, and accordingly selects either nodes (and incident edges) or edges, respectively, in each graph, if it satisfies the specific predicate. Notice that the predicate is evaluated on the entire data structure $\mathcal{D}$. Formally, we have $\mathcal{D}' = \mathcal{S}(\mathcal{D}|X_i = \tau)$, where $\mathcal{D}$ denotes the input data structure on which the selection operator is applied to, $X_i = \tau$ the generic predicate and the value that it is evaluated at, and finally $\mathcal{D}'$ the output data structure.

An example of an application of the selection operator is using the computed node degree for the Catalano phone call network as the underlying predicate, setting a high threshold in order to get the subgraph of most active members of the movement.

Set Aggregation: We can union the sets of nodes and aggregate the sets of edges in each partition ,$\mathcal{D}_i$, of $\mathcal{D}$ after we have partitioned it using a grouping function of your choice. Given a collection of graphs, $\mathcal{D}_i$, a set aggregation applied to it, based on a predicate β, is written as $\varphi_\beta(\mathcal{D}_i) = \{\bigcup_i \mathcal{D}_i | \beta(\mathcal{D}_i) = TRUE\}$ for some $i \in \{1, \cdots, n\}$. The aggregation predicate is evaluated independently for each set, and the entire set $\mathcal{D}_i$ is either contained or discarded in the aggregated set depending on the truth value of this predicate.

An example related to the Catalano network is aggregating the daily data structures $\mathcal{D}_i,\ i = 1, 2, \cdots, 10$ to a couple of them covering the periods of days 1-7 and 8-10, respectively (for a justification see Section 4). Another example would be to cluster similar nodes according to some of their characteristics.

Element Aggregation: This operator includes summations, counts and averages. In addition, we allow sampling as a form of aggregation that returns a subset of elements sampled according to the specified mechanism. Formally, we have $\mathcal{D}' = \mathcal{A}(\mathcal{D}|Z)$, where $\mathcal{D},\ \mathcal{D}'$ are defined as above and Z is either an attribute or the graph itself that is being aggregated. The following is the element aggregation operator: $\mathcal{D}' = \mathcal{A}(\mathcal{D}|Z) = \bigcup_i \{X_i \in \mathcal{D} | X_i = Z\}$, where $X_i = Z$ is a generic predicate.

Graph Partitioning: This operator is the "inverse" of aggregation with disjoin subsets. Each graph element of $\mathcal{D}$ is partitioned into multiple subgraphs based on the value of appropriate predicates,γ, defined or computed on its nodes and edges. $\mathcal{P}_\gamma : \mathcal{D} \rightarrow \{\mathcal{D}_1, \cdots, \mathcal{D}_n\}$, where $\mathcal{D}_i \bigcup \mathcal{D}_j = \emptyset$ for all $i \neq j$. γ is the partitioning predicate. It evaluates independently for each element of the set and assign it to its own subclass.

An example related to the Catalano network is using the cell phone tower locations to split the network into three disjoint sets for tracking the geographical movements.

In addition, for accomplishing the visual analytic task we introduce a visualization operator $\mathcal{V}$ whose role is to provide a visual representation of the underlying data structure of interest. The visualization operator can take various forms, including different ways of laying-out graphs [8], e.g. force-directed, hierarchical, hyperbolic, and also presenting the attribute data (e.g. histograms for numerical attributes, bar-graphs for categorical ones, etc...). In addition, it is assumed that the visual operators can be composed and thus produce multiple and possibly linked displays. Note that $\mathcal{V}$ is not an operator in the algebra, in that it does not have the closure property – it is a special operator, applied last, and used to create visual presentations. Its output is not a collection of graphs.

In order to accomplish the required visual analytic task, we need to apply multiple operators in sequence. In the presentation of the case studies, several such sequences are introduced and analyzed. Specifically, the final finding will usually be the results of a sequence of selection and aggregation operators; formally, the data structure $\mathcal{D}^\star$ from which the finding is obtained is given by $\mathcal{D}^\star = \mathcal{S}(\mathcal{S}(\mathcal{A}(...\mathcal{A}(D|Z))))$.

4. ANALYSIS CASE STUDY

We analyze the workflows of the Cell Phone Mini-Challenge from VAST 2008 [4]. This challenge requires analysis on a set of 400 unique cell phone call records over a ten-day period to learn the Catalano social network structure. The data set includes 9834 phone records with the following fields: calling phone identifier, receiving phone identifier, date and time, duration, and the cell tower of the call origin. A map is also provided to show the rough locations of the cell towers throughout the island region. The purpose of the challenge is to identify the Catalano/Vidro social network at day ten and to characterize the social structure changes throughout the time period. The first part of the challenge requires identifying Ferdinando Catalano, Estaban Catalano, David Vidro, Juan Vidro, and Jorge Vidro. Along with the data, the challenge provides a lead that Ferdinando Catalano is identifier 200. Also Ferdinando calls his brother, Estaban, most frequently. Finally, we know that David Vidro coordinates high-level activities and communications within the network.

Most competition submissions interpreted the challenge information as a static graph where nodes represent people and directed edges represent a call transaction. Competition entries also translated the challenge clues into a graph interpretation. For example, to find Estaban, a common method is to search within identifier 200's neighborhood for the node ***X*** with the most number of edges between 200 and ***X***.

If we preprocess the data set to convert it from a multi-edge to a simple-edge graph by merging common directed edges between nodes and use a force directed layout on Cytoscape [1], we produce the following hairball network shown Figure 1. This graph displays the entire ten day period data set with node 200 and its neighborhood nodes magnified and colored black. We also computed common metrics of the data set in Table 1. The entire time period data set forms one connected component.

From this presentation of the network, it is impossible to complete the challenge tasks. There are too many nodes and edges to even grasp any of the attribute properties such as time, cell tower location, or duration of the call. Of the 22 competition entries, we analyze and interpret into our framework the workflows of two winners: MobiVis and GeoTemporalNet.

4.1 Case Study 1: MobiVis Entry

The first workflow we analyze through our framework is from the MobiVis entry [11]. MobiVis is a visual analytics system for social and spatial information. For social networks, MobiVis uses graphs where nodes represent entities and edges represent a rela-

Table 1: Data Set Network Properties

Measurement	Value
Number of Nodes	400
Number of Edges	9834
Clustering Coefficient	0.02
Connected Components	1
Network Diameter	12
Characteristic Path Length	4.832

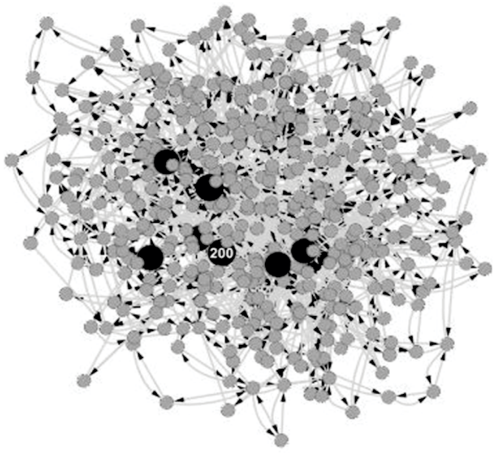


Figure 1: Hairball of entire dataset with node 200 and its neighbors colored black.

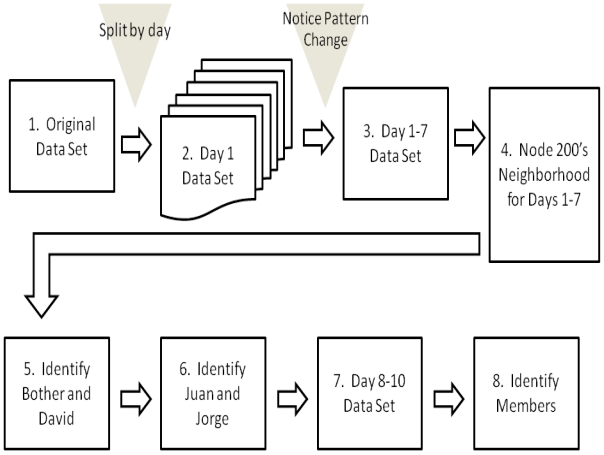


Figure 2: Workflow.

tion or association. MobiVis filters on attributes and time to help understand large data sets. We trace the data manipulation steps used in MobiVis' solution to the challenge. At stages which involve human intuition and decisions based on visual interpretation of a display, we formalize the method into our framework. Figure 2 shows the high level workflow. For each stage we provide the analysis and algebraic representation of the data set and operations. The algorithm overview below shows the algebraic expressions that correspond to the workflow in Figure 2. In the following section we describe each of these steps more thoroughly by incorporating the MobiVis analysis.

Algorithm 4.1: WORKFLOW($Dataset$)

1: Read in the original dataset.
$D^0 = \{G = (V, E), X_E = (X^{date}, X^{duration}, X^{tower})\}$

2: Compute node count.
$X_G^{Vcount} : \sum_i 1_{\{V_i \in V\}}$
$D^0 = \{G = (V, E), X_E, X_G^{Vcount}\}$

3: Split dataset into ten days.
$\mathcal{P}_\gamma : D^0 \rightarrow D^1$
$\gamma = date$
$D^0,\ D^1 = \{D_1^1, ..., D_{10}^1\}$

4: Element select node 200 subgraph for each day.
$D^2 = S(D^1 | X_i = \tau_{200})$
$\tau_{200} =$ One of the neighbors of 200
$D^0,\ D^1,\ D^2 = \{D_1^2, ..., D_{10}^2\}$

5: Set aggregation on days 1 - 7.
$D^3 = \varphi_\beta(D_i^1) = \{\bigcup_{i=1}^7 D_i^1 \mid \beta(D_i^1) = TRUE\}$
$\beta =$ days 1- 7
or element selection based on days 1-7
$D^3 = S(D^0 \mid X_i = \tau_{1-7})$
$\tau_{1-7} =$ days 1- 7
$D^0,\ D^1,\ D^2,\ D^3$

6: Element select node 200 subgraph for days 1-7
$D^4 = S(D^3 \mid X_i = \tau_{200})$
$D^0,\ D^1,\ D^2,\ D^3,\ D^4$

7: Identify members
$Estaban = \max_i X_{symmfreq}[i]$
$David = \max_i X_{freq}[i] = \arg\max_i \{X_{in} + X_{out}\}[i]$
...

8: Repeat identification for days 8-10 data set.

At the first stage, the original data set is given by $D^0 = \{G = (V, E), X_E = (X^{date}, X^{duration}, X^{tower})\}$, where the nodes represent the identifiers in the call records and edges represent call records between identifiers. X^{date}, $X^{duration}$, and X^{tower} are intrinsic edge attributes, directly inserted from the original data set. The choice between defining attributes as node versus edge attributes depends on the application. For example, the duration of a call is associated with a phone call transaction; therefore, duration is more appropriate as an edge attribute.

As we discussed earlier, it is impossible to answer the challenge questions when we view the whole time period data set at once. The MobiVis entry also sees this difficulty when they display the original data set and provide a node count. To perform these actions in our framework we call Display(G) using a suitable layout. We can also compute a network attribute,X_G, to represent the node count. We compute this by first aggregating the number of nodes: $\sum_i 1_{\{V_i \in V\}}$. Next we store this as a network attribute X_G^{Vcount}.

Based on the challenge hint, the MobiVis entry examines node 200's properties and interactions separately for each day. Since the original data set represents a ten day time period, a per day analysis is a logical choice for further examination. To perform a split on the original data set, we use partition operator $\mathcal{P}_\gamma : D^0 \rightarrow D^1 = \{D_1^1, ..., D_{10}^1\}$ based on the selection predicate γ = date.

Now we have the following collection of new data sets $\{D_1^1, ..., D_{10}^1\}$. After creating new data sets, we must decide whether to re-compute attributes. Intrinsic attributes carry over; for example, the duration of a call record does not change if the call record is placed in a different data set. However, the number of nodes in each data set changes. So we recompute the network node count attribute, $X_{G_i^1}^{Vcount}$ for each new data set created.

MobiVis filters all nodes except for the neighborhood of node 200 in their per day examination. For a similar effect, we select the subgraph for node 200's neighborhood on each day's data set. Again we use the element selection operator with predicates $D^2 = S(D^1|X_i = \tau_{200})$, where $\tau_{200} = \{$One of the neighbors of 200$\}$, and produce the following set of graphs: $D^2 = \{D_1^2, ..., D_{10}^2\}$. In Figure 3 we display each of these neighborhood subgraphs. We see that node 200 is active until day 8. On day 8, there is no communication and afterwards its call pattern changes. The MobiVis entry realizes this conclusion by observing the total duration of calls per day for node 200. In our framework, we produce this set of total duration values by aggregating the duration of calls on the adjacent edges to node 200 for each day data set and displaying the results. This leads to the next stage in the workflow: consider the first seven days and last three days separately.

At stage 3 of the flowchart, the MobiVis entry analyzes the first seven days. Storing all previous data sets allows us now to create a data set of call records between days 1 and 7. In our algebra we can either merge days 1 through 7 data sets by set aggregation: $D^3 = \varphi_\beta(D_i^1) = \{\bigcup_{i=1}^7 D_i^1 \mid \beta(D_i^1) = TRUE\}$. Or we can select the first 7 days from the original data set by element selection:$D^3 = S(D^0 \mid X_i = \tau_{1-7})$ where τ_{1-7} = days 1 - 7 . Both methods result in the following data set of days 1 through 7, D_{1-7}. Again, MobiVis zooms into the neighborhood of node 200. We select the subgraph of node 200 in the days 1 through 7 data set, D^4.

In stage 5 of the workflow, MobiVis reaches a human readable display of the graph and begins identifying the members of the Catalano/Vidro network. Their identification at this point is done by visually inspecting the graph. Assuming that 200 is Ferdinando, they identify the brother by selecting the node in 200's neighborhood with maximum symmetric frequency. The symmetric frequency for vertex i is defined as follows:

$$X_{1\text{-}7}^{symmfreq} = |(X_{1-7})_{in}[i] - (X_{1-7})_{out}[i]| \text{ where,}$$

$$X_{in}[i] = \sum_{j,j\neq i} (A_{1-7})_{ji}$$

$$X_{out}[i] = \sum_{j,j\neq i} (A_{1-7})_{ij}$$

The maximum symmetric frequency is defined as a function on

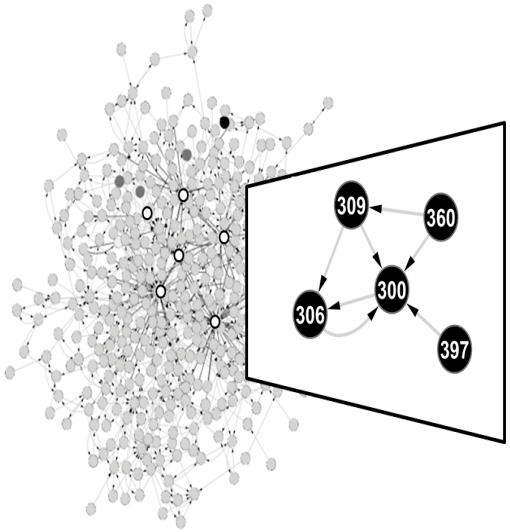


Figure 4: Days 8-10 network and Catalano/Vidro subnetwork. The days 1-7 social network identified are the dark grey nodes. The white nodes represent the new social network with their adjacent edges darkened.

$X^{\text{symm freq}}$, where $\mathcal{F}_{X^{\text{symm freq}}} : \mathbb{R}^{400} \rightarrow \mathbb{R}^1$, for calculating the max of the attribute. This results in node 5 identified as Estaban, Ferdinando's brother. The challenge clue states that David is a highly active member of the network. Therefore, MobiVis identifies David by selecting the node with the maximum frequency:

$$\arg\max_i X_{freq}[i] = \arg\max_i \{X_{in} + X_{out}\}[i].$$

Hence, node 1 is David. Once David and Estaban are identified, the remaining two neighbors of node 200 are identified as Juan and Jorge. There is not enough information to distinguish between the two. In our framework, we find these nodes by selecting the next two maximum frequency nodes in 200's neighborhood subgraph: D^4. Juan and Jorge are identified as nodes 2 and 3.

4.1.1 MobiVis Days 8-10 Analysis

To determine the social structure after day 8, the MobiVis entry examines the data set of days 8 through 10 merged. This data set is created similar to days 1 through 7 data set computed earlier. The MobiVis entry hypothesizes that the elite members switch phones after day 8. They support this hypothesis by searching for a subgraph in the days 8 through 10 network which resembles the 200 neighborhood found in days 1 through 7. The difficulty here is we cannot rely on the lead that node 200 is Ferdinando. To identify this subgraph, MobiVis visually inspects the days 8-10 data set to remove nodes with low frequency of communication until they identify a neighborhood similar to the subgraph of node 200 during the first seven days. In our framework, we rank edges by frequency of communication, and display the graph. Given this display, selections to remove lower frequency nodes can be iteratively performed until we find a similar network. Figure 4 shows the subgraph MobiVis finds on days 8 through 10, which they guess is the new Catalano/Vidro network. The nodes in this subgraph are 309, 306, 360, 397, and 300. Visually they map the members of this subgraph to the ones from days 1-7. We translate their visual mapping to our algebra.

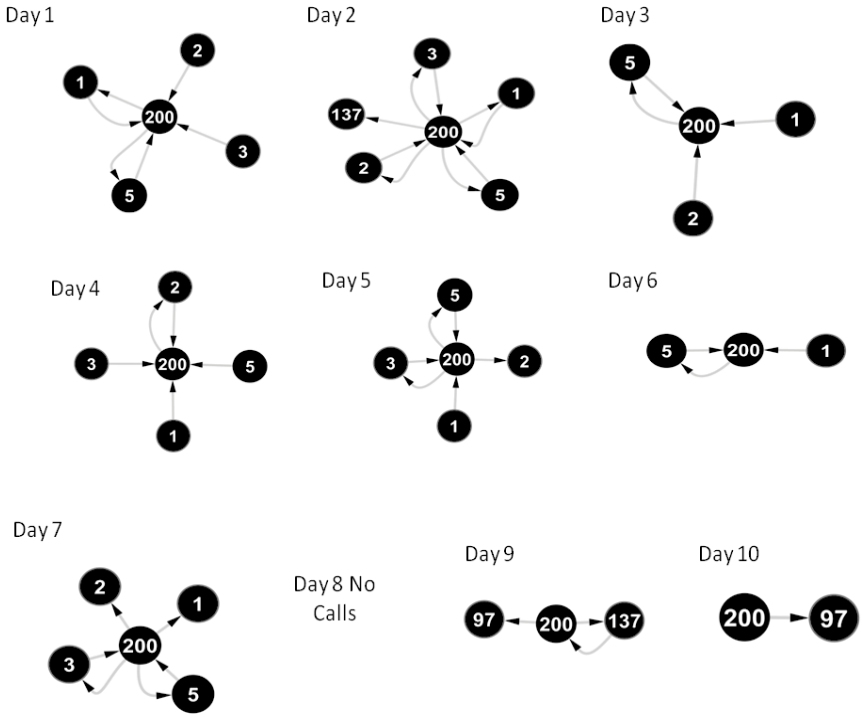


Figure 3: Node 200's neighborhood through the time period.

From this subgraph MobiVis identifies David by visually selecting the node with the highest frequency of communication. This is equivalent to a max frequency operation,$\arg\max_i\{X_{1-7}\}_{freq}[i]$, resulting in node 309. To find Ferdinando and his brother they find the two nodes in this network with high symmetric frequency. Ferdinando is identified as the node with connections to Juan and Jorge which are mapped to 397 and 301 from visual inspection. This method to determine the new subgraph in the days 8-10 data set might be time consuming if the data set is noisy or contains several possible subgraphs. We notice that David in days 1-7 has the highest degree in the network. If we simply select the node in the merged days 8-10 data set with the maximum degree, we can narrow the choices for possible subgraphs. After computing maximum degree, $\arg\max_i \sum_{j,j\neq i}\{A_{1-7}\}_{ij} + \{A_{1-7}\}_{ji}$, we find David is node 309. On the days 1-7 data set, a selection of node 309 results in null. This result might support the hypothesis that the phones were indeed replaced after day 7.

In the days 1-7 data set, Estaban and David have the most number of common neighbors. Again if we assume that the social structure during days 1-7 remains similar for the new set of phones, we can identify Estaban by selecting the node with the most number of common neighbors with David. This computation is done by creating a new node attribute: common neighbors with node 309. The common neighbor computation is the intersection of nodes in two rows of the adjacency matrix. For each node we store its common neighbors with the fixed node 309. We perform a maximum aggregation operator on this attribute $\arg\max_i\{X_{8-10}\}_{\text{common neigh. 309}}[i]$. Estaban is node 306.

Now we can find Ferdinando since we know Estaban is his highest interactivity neighbor. The final mapping produced is David: 309, Estaban: 306, Ferdinando: 300, and Juan and Jorge: 397 and 360. The algebra in coordination with the display helps support our hypothesis and decisions at each stage. While the MobiVis entry does a visual inspection to support their identification mapping, providing the exact operation helps trace the stages in the workflow.

4.2 Case Study 2: GeoTemporalNet Entry

After the competition deadline, VAST never released correct answers for the challenges. The reason for not publishing the answers is to allow open interpretation of the data sets. In the above analysis, we see the interpretation of the data set given an analyst uses MobiVis. However, other winning entries delivered different conclusions. The differences in tools and interpretation of the challenge hints lead to unique results for the same data set.

The challenge hints must be translated from a word sentence to a network property. There was a different degree of open interpretation for these hints. For example, the first hint that 200 might be Ferdinando has a direct network interpretation: identify node 200 in the graph as Ferdinando. David's hint is that he coordinates high-level Paraiso activities and communications. This hint does not have a direct network interpretation. What are considered high-level activities? What does it look like in the graph to coordinate these activities?

We present the GeoTemporalNet entry [14], also a winner, for the cell phone mini-challenge. Instead of analyzing their workflow from the start, we describe the differences in analysis and results between GeoTemporalNet and MobiVis. As we will see, our algebraic framework provides a superset of operations used by MobiVis and GeoTemporalNet. We can use our framework as a linking language between the two tools. Therefore, again we analyze the steps, this time for GeoTemporalNet, within our framework.

The GeoTemporalNet entry used a combination of tools: JSNVA (Java Straight-line drawing Network Visual Analysis framework) and TemporalNet. JSNVA is a software framework for net-

Table 2: Identification of Catalano/Vidro Network

Member	GeoTemporalNet	MobiVis (1-7)	MobiVis (8-10)
Ferdinando	200	200	300
Estaban	5	5	306
David	0	1	309
Jorge	1/2	2/3	397/360
Juan	1/2	2/3	397/360

work visual analysis in different applications. The GeoTemporalNet group developed TemporalNet within JSNVA to show communication patterns in call graphs. They use a static graph with nodes representing people and edges representing calls for the social network. Like MobiVis, they use force-directed graphs in their layout.

The first notable analysis difference between MobiVis and GeoTemporalNet occurs at stage 5 in the workflow: the identification of David and Estaban. GeoTemporalNet finds David through a most common neighbor element selection operation with node 200. They apply this operator on the original data set, which includes the entire time period.

$$\text{Common neighbors}(i,j) = N(i) \bigcap N(j) \text{ where,}$$
$$N(i) = \text{list of neighbors for vertex i}$$
$$N(j) = \text{list of neighbors for vertex j}$$

The common neighbor operation is performed between a pair of nodes in a graph. As described earlier, common neighbor finds the intersection of two rows in the adjacency matrix. In this case, the node pair is 200 and each of the other nodes in the graph. We store a new node attribute for the number of common neighbors with node 200. Then we apply a maximum function to return the node with the most number of common neighbors. This function leads GeoTemporalNet to believe David is node 0.

$$\text{David} = \arg\max_i \text{Common neighbors}(i,0) \text{ for all } i \in V$$
$$\text{where } V \in \mathcal{D}^0$$

The challenge instructions never explicitly state that David and Ferdinando communicate directly. During the ten day period, node 0 and 200 never call each other. MobiVis does not compute common neighbors between nodes. Also, they filter nodes to observe only node 200's neighborhood subgraph. Therefore, their tool cannot identify node 0 as a possible result for David.

On the other hand, GeoTemporalNet does not filter the graph to zoom into only node 200's subgraph. In addition, they interpret the challenge hint differently. As a result, MobiVis and GeoTemporalNet provide different answers. However, as we have shown, our framework captures both methods and can arrive at both answers. The limitations of the tool on the analyst's interpretation are removed.

GeoTemporalNet identifies the other members, Estaban, Juan, and Jorge, similar to MobiVis' method. These are a series of selections for the nodes with most frequent communication with node 200. Again they perform this operation on the original data set.

The final mapping GeoTemporalNet produces is: Ferdinando: 200, David: 0, Brother: 5, Juan and Jorge: 1 and 2. Figure 5 shows the identification differences between GeoTemproalNet and MobiVis for the Catalano/Vidro network.

4.2.1 GeoTemporalNet Days 8-10 Analysis

GeoTemporalNet does similar per day analysis as MobiVis to discover node 200's calling pattern changes after day 7. However, GeoTemporalNet does not guess that the members replaced phones to explain this pattern change. In their work, they assume that a person is never assigned a new phone. Instead they hypothesize that new members entered the Pareto movement. These new members have equivalent roles as some of the high-level members identified in the problem: Estaban, Jorge, and Juan. They support this hypothesis by computing the Jaccard coefficient,$J(i,j) = \frac{|N(i) \bigcap N(j)|}{|N(i) \bigcup N(j)|}$ for any vertices i and j , of these new members with their member of equivalent societal role. The Jaccard coefficient is a vertex similarity metric to measure the structural equivalence between two nodes. This metric is simply the number of common neighbors normalized. The GeoTemporalNet entry does not state how they identify the new set of members: 309, 306, 360, and 397; or their equivalent role pair: (1,309), (5,306), (3,360), and (2, 397). However, they compute and display the Jaccard coefficient for these pairs of nodes. They state that the high Jaccard coefficient leads them to believe these pairs have equivalent roles; therefore, the later appearing nodes may be replacements for the previous nodes.

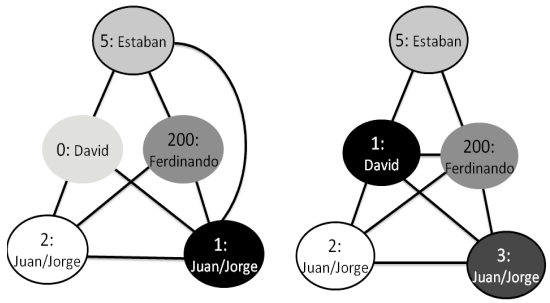


Figure 5: Mapping between MobiVis and GeoTemporalNet networks.

In our framework to produce the same support we do the following operations: First we create a Jaccard coefficient attribute for the following node pairs: (1,309), (5,306), (3,360), and (2, 397). Since we are not interested in computing the Jaccard coefficient for all node pairs in the graph, we can create a set of network attributes for these node pairs, $X_G^{\text{common neighbors}(i,j)}$ for all i and j in set $\Omega = \{(1,309),(5,306),(3,360),(2,397)\}$ where $G \in \mathcal{D}^0$.

After analyzing the MobiVis and GeoTemporalNet entries, we see two different workflows. Our algebraic framework captures both methods and is also capable of filling in the intuition gaps with algebraic operations.

5. NEW FINDINGS

Of course, the algebraic model we propose is capable of representing analyses beyond the specific examples studied in the preceding section. In this section, we present one such analysis instance, showing a new analytic finding on the Catalano data set, one that was not reported by any of the teams participating in the VAST08 challenge.

5.1 Social Structure

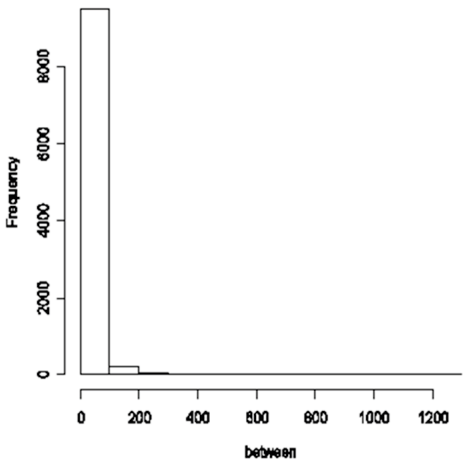


Figure 6: Edge betweenness for the original network .

On each step of the algebraical model, we compute graph related attributes for the set of graphs and denoted as X_G. Among all the computed metrics, the most important attributes for identifying the social structure are edge betweenness and clustering coefficient [2].

The edge betweenness is defined as $B_u = \frac{g(i,u,j)}{g(i,j)}$, where $g(i,u,j)$ is the number of shortest paths between vertices i and j that pass through edge u, $g(i,j)$ is the total number of shortest paths between i and j, and the sum is over all pairs i,j of distinct vertices [2, 9] .

The clustering coefficient is also known as transitivity [2, 9] , which is based on the following definition for an undirected unweighted network: $C = \frac{N_\Delta}{N_3}$, where N_Δ is the number of triangles in the network and N_3 is the number of connected triples. Therefore we have

$$N_\Delta = \sum_{k>j>i} A_{ij}A_{ik}A_{jk}$$

$$N_3 = \sum_{k>j>i} (A_{ij}A_{ik} + A_{ji}A_{jk} + A_{ki}A_{kj})$$

are the elements of the adjacency matrix A and the sum is taken over all triples of distinct vertices i, j and k. Since it assigns the same weight for each triangle in the network, it can be related to the clustering coefficient for each vertex, which captures the hierarchical structure in the network.

According to the distribution of the edge betweenness [3] shown in Figure 6, there is only a small number of edges suggesting important relationships in all the graphs. Further, almost all clustering coefficients are small and low in transitivity, for all graphs during the ten day period. The combination of these findings strongly supports the existence of a hierarchical structure within the Catalano network.

5.2 Geographical location and movement

After the discovery of a change in the social structure after day 7, we examine the geographical location of the main actors in the network, as well as their movement in the 1-7 day and 8-10 day periods. The proposed framework can easily address such issues as shown next.

Based on the map of Isla del Sueno, we decided to partition the thirty cell phone towers on the island into three groups, representing the Upper, Middle and Lower sections of the island. This can be accomplished by applying an element aggregation operator to the towers' location. Since one of the available attributes corresponds to the cell tower used by the phone call's originator, it is possible to track the caller's movement throughout the ten day period. However, due to the finding of a hierarchical network structure, we focus on the leadership group formed around Ferdinando.

We start by selecting the nodes corresponding to the leadership group for each day i. Specifically, $\tilde{D}_i^1 = S(D_i^1 \mid X_i = \tau)$, where $\tau =$ subset of $\{1, 2, 3, 5, 200, 300, 306, 309, 360, 397\}$ for day $i = 1, \cdots, 10$. Notice that for not all members of the leadership group made phone calls on every day. However, some broad patterns emerge from our analysis, as shown in Figure 7. It exhibits the geographic location of the leadership group at different days. The plots are constructed based on the bipartite graph defined by the caller's ID and the grouped (Upper, Middle, Lower) location of the cell tower employed. Further, the length of the edges is weighted by the call frequency of each node for that day; hence, shorter edges indicate a more active call pattern with regards to the tower under consideration, and longer ones a less active pattern.

It can be seen that on day 2, Ferdinando is located in the Middle section of the island, while his brother Esteban and Juan in the Upper section. This pattern holds for days 1-7. On the other hand, a more mobile pattern emerges for days 8-10 (operating under the assumption that node 300 is Ferdinando , 306 is Esteban, 309 is David, etc. For example, it can be seen that although Ferdinando remains stationary in the Middle of the island, Esteban moves from the Upper section in day 8 to the Middle section in subsequent days, while David shares his time between the Middle and Lower sections on days 8-9, but stays in the Lower one on day 10.

Figure 8 summarizes the location changes of selected members of the leadership group throughout the ten day period. The following conclusions can be reached.

- The person with IDs 200 and 300 is Ferdinando Catalano. He spends the entire ten day period in the Middle section of the island.
- Ferdinando's brother Estaban Catalano, corresponds to IDs 5 and 306. He spends most of his time in the Upper section of the island for the first eight days, while for the last two days he moves to Middle section and is co-located with his brother.
- David, a rather active member of the Catalano network, has IDs 1 and 309. He spends the first seven days in the Middle section, but starts visiting the Lower section on days 8 and 9 and stays there the entire length of day 10.
- A similar pattern to David's emerges for Juan and Jorge that exhibit more movement after day 8.

Our analysis provides additional insight on the location and movements of the leadership group of the Catalano network. It is worth noting the increased activity of several members (but not Ferdinando) over the last three days. Nevertheless, without additional information, it is hard to assess the significance of these movements regarding the activities of the network.

6. RELATED WORK

As the VAST Challenge demonstrates, there are several visual analytic tools with different capabilities for geospatial activity and behavior, text processing, and social network analyses. We focus on just a few references that particularly deal with the visual

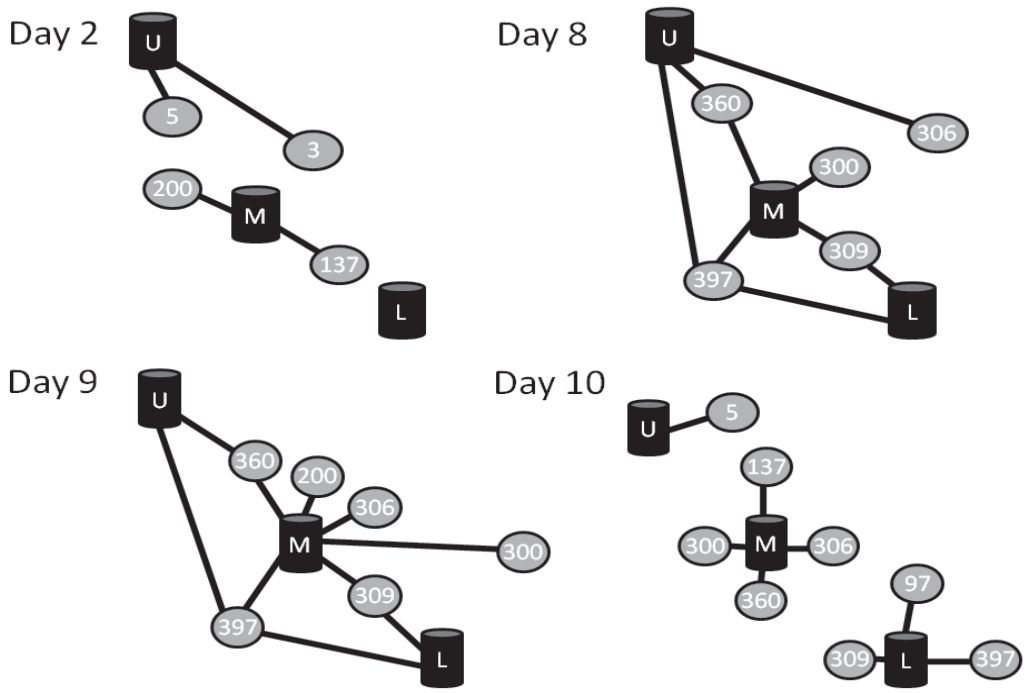


Figure 7: The location changes for elite members in Catalano family.

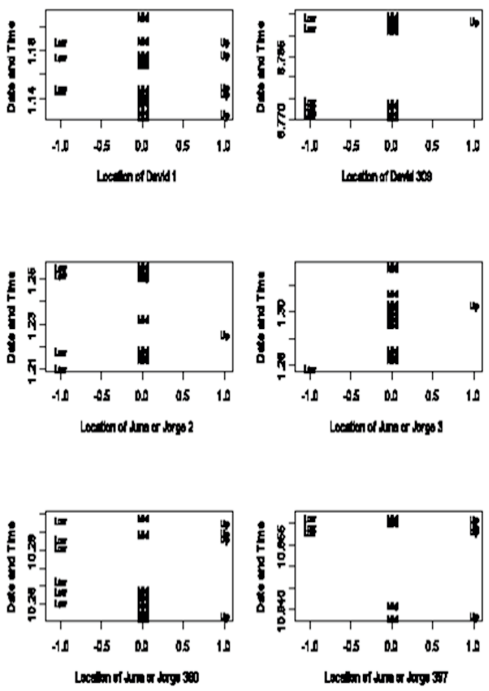


Figure 8: The individual geographical movement.

analytics of graphs. In [13], a human-centric design approach is adopted to create a tool for descriptive graphs, while in [12] information about the local topology of a graph is captured in a signature that aids exploration of graphs. In [6], interactive exploration of networks is undertaken through enhanced layouts, while in [10] semantic and structural abstractions are used for analyzing social networks. Data traffic is explored through network maps in [7] and Internet routing changes in [5]. A key point to observe is that while there are several systems that have been very effective in providing better support for visual analytics of network data in a particular application context, no one has attempted to develop a formal foundation on which to construct such systems. This is what we aim to do. Thereby, we hope to be able to support a broad range of applications rather than just one.

7. CONCLUSIONS

With our proposed algebraic model we can represent a large class of visual analytic operations on graphs, as we demonstrated through analysis of the VAST 2008 cell phone mini challenge. For future work, we plan to consider the computational issues involved in efficiently implementing our model and issues involved in incorporating this framework into a tool.

8. ACKNOWLEDGEMENTS

The research is supported in part by NSF grant numbers 0438909 and 0808824 and NIH 1-U54-DA021519.

9. REFERENCES

[1] The Cytoscape Collaboration. *Cytoscape Users Manual*. The Cytoscape Collaboration, Institute for Systems Biology and University of California San Diego, 2006.

[2] Luciano Costa, Francisco Rodrigues, Gonzalo Travieso, and Villas Boas. Characterization of complex networks. *Advances in Physics*, 56(1):167 – 242, May 2007.

[3] Gabor Csardi. *igraph*. R User Manual, CRAN, 2009.

[4] G Grinstein, C. Plaisant, S. Laskowski, T. OŠConnell, J. Scholtz, and M Whiting. Vast 2008 challenge: Introducing mini challenges. *Proceedings of IEEE Symposium*, 1(1):195 – 196, October 2008.

[5] Mohit Lad, Dan Massey, and Lixia Zhang. Visualizing internet routing changes. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1450 – 1460, November 2006.

[6] Bongshin Lee, Cynthia S. Parr, Catherine Plaisant, Benjamin B. Bederson, Vladislav D. Veksler, Wayne D. Gray, and Christopher Kotfila. Treeplus: Interactive exploration of networks with enhanced tree layouts. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1414 – 1426, November 2006.

[7] Florian Mansmann and Svetlana Vinnik. Interactive exploration of data traffic with hierarchical network maps. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1440 – 1449, November 2006.

[8] George Michailidis. *Data Visualization Through Their Graph Representations*. Springer, Berlin, 2006.

[9] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(1):167 – 256, March 2003.

[10] Zeqian Shen, Kwan-Liu Ma, and Tina Eliassi-Rad. Visual analysis of large heterogeneous social networks by semantic and structural abstraction. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1427 – 1439, November 2006.

[11] Visualization and Interface Design Innovation (VIDI) Group. Intuitive social network graphs visual analytics of cell phone data using mobivis and ontovis. *IEEE Symposium on Visual Analytics Science and Technology*, 1(1):19–24, October 2008.

[12] Pak Chung Wong, Harlan Foote, George Chin Jr, Patrick Mackey, and Ken Perrine. Graph signatures for visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1399 – 1413, November 2006.

[13] Pak Chung Wong, Harlan Foote, Patrick Mackey, Ken Perrine, and George Chin Jr. Generating graphs for visual analytics through interactive sketching. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1386 – 1398, November 2006.

[14] Qi Ye, Tian Zhu, Deyong Hu, Bin Wu, Nan Du, and Bai Wang. Exploring temporal communication in mobile call graphs. *IEEE Symposium*, 1(1):16 – 19, October 2008.

Heidi Matrix: Nearest Neighbor Driven High Dimensional Data Visualization

Soujanya Vadapalli Kamalakar Karlapalem
Centre for Data Engineering, IIIT-Hyderabad, INDIA.

ABSTRACT

Identifying patterns in large high dimensional data sets is a challenge. As the number of dimensions increases, the patterns in the data sets tend to be more prominent in the subspaces than the original dimensional space. A system to facilitate presentation of such subspace oriented patterns in high dimensional data sets is required to understand the data.

Heidi is a high dimensional data visualization system that captures and visualizes the closeness of points across various subspaces of the dimensions; thus, helping to understand the data. The core concept behind Heidi is based on prominence of patterns within the nearest neighbor relations between pairs of points across the subspaces.

Given a d-dimensional data set as input, Heidi system generates a 2-D matrix represented as a color image. This representation gives insight into (i) how the clusters are placed with respect to each other, (ii) characteristics of placement of points within a cluster in all the subspaces and (iii) characteristics of overlapping clusters in various subspaces.

A sample of results displayed and discussed in this paper illustrate how Heidi Visualization can be interpreted.

1. INTRODUCTION

Data mining aims to uncover knowledge from large data repositories. Ideally, the extracted knowledge is to be conveyed in such a format that facilitates easy human understanding of the data and its characteristics. One such format is a visual representation of the knowledge; others being textual summaries and auditory representations [4]. Data visualization and information visualization strive to present information obtained from the data in various visual representations to meet the needs of a user analyzing high dimensional data.

Also, the output of data clustering is a set of groups of points (highly dissimilar groups of highly similar points); and existing algorithms usually focus on obtaining these groups but not on presenting these results in a comprehensive manner. Added to that, identifying and conveying high dimensional data information is non-trivial and is a major problem continued to be addressed by researchers for past few decades.

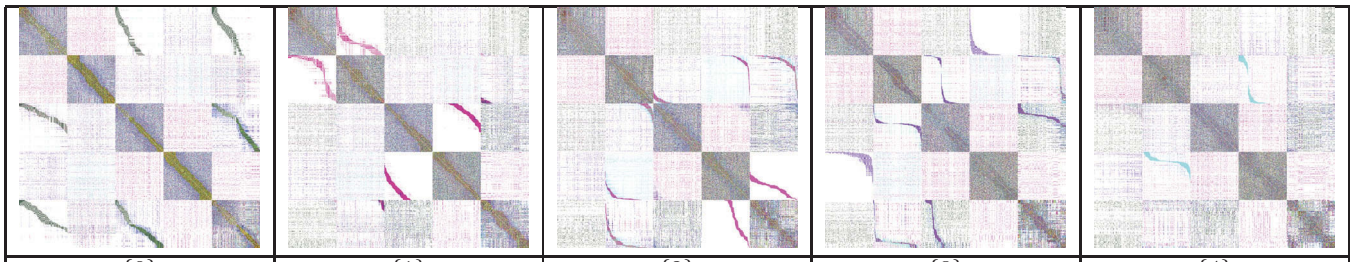
In this paper, we present Heidi Matrix which is a two dimensional representation of high dimensional data based on the locality information between every pair of points in the data set across all subspaces. The 2-D representation is later used to generate a multi-color image that facilitates user to understand various aspects of the data. We also present the Heidi system and highlight its features.

Heidi is a system to process higher dimensional data sets and extract the closeness of points across subspaces and present this closeness through a visual two-dimensional representation. This representation acts more like an X-ray of the data set, giving prominence to relevant subspaces. The core concept behind Heidi is building a matrix (called Heidi[1] matrix) based on the prominence of patterns within the nearest neighbor relations between pairs of points across the subspaces. k nearest neighbors (referred to as kNN in the rest of the paper) is used between every pair of points to identify subspaces in which they are "close". It enables the user to visualize and reason about the closeness of data points across all the subspaces, by displaying how the proximities of points change in various subspaces. Heidi has been designed to meet the needs of a user analyzing high dimensional data.

Heidi is a system to:
1. Generate two dimensional representation of higher dimensional data sets.
2. Present concise representation and display of kNN relationships among points in all the subspaces.
3. Present spatial overlap among clusters in various subspaces.
4. Present all the above information for analysis in a single Heidi image.
5. Aid the user in interpreting the details of clusters like number of clusters, size of the cluster, position of the cluster with respect to the other clusters in the data set.

VAKD'09, June 28, 2009, Paris, France.

[1]The word 'Heidi' rhymes with Hi-D like 5-D, 2-D; hence, we named the system 'Heidi', pronounced as *high d.*



{0}	{1}	{2}	{3}	{4}

Table 1: Heidi images for 5-D dataset with points ordered in various 1-D subspaces

Organization of the paper: Section 2 presents the Heidi Matrix Approach, Section 3 presents Heidi visualization and implementation aspects, Section 4 gives experimental results, Section 5 covers the related work, and Section 6 provides conclusions and future work.

2. HEIDI MATRIX

2.1 Definitions

Given a d-dimensional data set $\mathcal{X}$ ($n = |\mathcal{X}|$), the aim is to generate a single visualization that stores comprehensive information related to close-ness of points in various subspaces and clusters in various subspaces. The subspaces are non-empty subsets of the dimensions of data set $\mathcal{X}$.

Let $\mathcal{X}$ represent the data set; $\mathcal{D}$, the set of dimensions; and $\mathcal{P}$, the set of all possible subspaces. $\mathcal{P} = \mathcal{P}^{\mathcal{D}} - \phi$ ($\mathcal{P}^{\mathcal{D}}$ denotes the powerset including ϕ).

For a subspace $\mathcal{S}$, let $d_{\mathcal{S}}(p, q)$ denote the distance between two points p and q such that only the dimensions present in subspace $\mathcal{S}$ are considered. The distance function used in this paper is Euclidean distance; other l_p-norm distances could also be used.

$$d_{\mathcal{S}}(p, q) = \sqrt{\sum_{s \in \mathcal{S}} (p_s - q_s)^2}$$

k-Nearest Neighbors (kNN): The k-th nearest neighbor of a point p is the point that is in the k-th place when all the distances from p to other points in the data set are sorted in ascending order. Let the k-th nearest neighbor distance of the point p in a subspace $\mathcal{S}$ be represented as $kd_{\mathcal{S}}(p)$.

The k-nearest neighbors of point p in a subspace $\mathcal{S}$ are defined as:

$$k\text{NN}_{\mathcal{S}}(p) = \{q | d_{\mathcal{S}}(p, q) \leq kd_{\mathcal{S}}(p)\}$$

$k\text{NN}_{\mathcal{S}}(p)$ forms a set of all points that are at most k-th nearest to point p in subspace $\mathcal{S}$. kNN relationships are used to define the closeness between a pair of points.

Why kNNs?
The values of the global distances between all points in high-dimensional data are not so useful. Global distances in high dimensional spaces are not as effective as distances within subspaces [1]. Also, in the present circumstances, where information about various subspaces is linked to each point-pair, a relationship that is comparable across all the subspaces is needed. The kNN relations are comparable across the subspaces, i.e., the kNN set of subspace S_1 can be compared with that of another subspace S_2. By doing so, we are only comparing the relative proximity (1st nearest of p in S_1 vs. 1st nearest of p in S_2).

2.2 Heidi Approach

Heidi Matrix: Let $\mathcal{H}$ denote the Heidi Matrix in which a cell of position (p, q) is represented as $\mathcal{H}_{pq}$. Each cell is a bit vector. The bits correspond to the subspaces present in $\mathcal{P}$. The length of the bit vector is $|\mathcal{P}|$, i.e., $2^d - 1$. The subspaces are ordered such that the least significant bits in the bit vector (bits positioned towards the right) correspond to lower subspaces (less number of dimensions in subspaces), while the most significant bits (positioned towards the left) correspond to higher subspaces (more number of dimensions in subspaces). For example, consider a three dimensional data set and a bit vector 0101101 for a point pair (p, q). It implies that p considers q as its kNN in subspaces {1,2}, {0,1}, {2} and {0}.

Each element $\mathcal{H}_{pq}$ corresponding to a pair (p, q) in the Heidi matrix $\mathcal{H}$ generated, reflects on the kNN relationship between those two points in all the subspaces. For example, a point p can consider q to be k-nearest in only one subspace thus having only one bit corresponding to that subspace as 1; while another point r considers s to be k-nearest in two subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$, having the two corresponding bits set to one, the rest set to zero. Once the Heidi matrix is obtained, the bit vectors are used to generate color codes for visualization.

Heidi Hierarchy: The basic unit of abstraction in Heidi matrix is a bit. A bit (p, q, S_i) is set to 1, if p is "close" to q in subspace S_i. The set of bits corresponding to all possible subspaces together constitute the bit vector which is the cell of the matrix. Groups of adjacent cells belonging to a single cluster either along the rows or columns form the blocks. Each block B_{ij}, ($1 \leq i, j \leq c$, c = number of clusters) corresponding to the "closeness" relations existing between the points belonging to cluster C_i and cluster C_j. Each block B_{ij} constitutes a set of $|C_i| \times |C_j|$ cells ($|C_i|$ and $|C_j|$ are number of points in the clusters C_i and C_j respectively), each cell of the matrix (p, q) representing if q is a kNN of p across various subspaces.

Heidi Blocks: When the points are ordered cluster-wise along the rows and columns of the Heidi matrix, the matrix is partitioned into $c \times c$ (c representing the number of clusters) number of blocks of adjacent cells. Each block represents the "closeness" relations between
(i) two different clusters; a set of points belonging to one cluster along the rows and another set of points belonging to another cluster along the columns, or
(ii) points in one cluster; same order of cluster points along the rows and columns of the block.
The blocks along the diagonal of the Heidi matrix have the points along both the rows and the columns belonging to a single cluster. The block of cells corresponding to a particular cluster is along the diagonal of the heidi matrix, while the blocks corresponding to two different clusters are away from the diagonal.

2.3 Heidi Matrix Computation

The step-by-step approach of computing Heidi Matrix is given below:

1. **Computing $\mathcal{P}$ (all possible subspaces)**: Given $\mathcal{D}$, $\mathcal{P}$ is computed. The total number of subspaces are $2^d - 1$.

2. **Computing kNNs**: For each subspace $\mathcal{S} \in \mathcal{P}$, kNNs are computed for every point $p \in \mathcal{X}$. The distance function $d_{\mathcal{S}}(p, q)$ uses only the dimensions present in $\mathcal{S}$.

3. **Computing $\mathcal{H}$**: An $n \times n$ matrix of bit vectors is created. Each bit vector is of length $2^d - 1$ and all the bits are initialized to 0.

   For every point $p \in \mathcal{X}$, its $k\text{NN}_{\mathcal{S}}(p)$ are obtained for all $\mathcal{S} \in \mathcal{P}$ (already computed in an earlier step). The bit-vector $\mathcal{H}_{pq}$ is retrieved for the point-pair (p, $q \in k\text{NN}_{\mathcal{S}}(p)$). The bit corresponding to the subspace $\mathcal{S}$ is set to 1.

2.4 Coloring Heidi Matrix

With the Heidi Matrix computed, the integer value of bit vector is used to obtain a unique color for every possible set of subspaces. Heidi uses the RGB color model; each component (i.e., R, G, B) requires 1 byte to store the color intensity value. Altogether, requiring 24 bits to represent the color.

The whole color range is spread across the subspace lattice. The total number of sets of subspaces in this lattice is 2^{2^d-1}. It has been observed that the number of unique bit-vectors in the Heidi Matrix is far less than 2^{2^d-1}. Assigning a unique color to each unique bit-vector is the key idea of this technique.

The total number of possible colors using the RGB model is $256 \times 256 \times 256 = 16777216$. The whole color range is divided into the number of unique bit vectors and the color at each division corresponds to one of the unique bit vectors.

This technique can handle data sets of larger dimensionality within the range of 5-10. For much higher dimensionality, the color resolution needs to be increased. In PPM file format, each color component can take a maximum value of 65536 (i.e., 16 bits for R, G, B individually) and with this color resolution, this technique can compute Heidi for data sets with dimensionality < 50 (approximately).

Maximum number of colors
For each unique bit vector, a unique color is associated with it (computed by earlier mentioned coloring techniques). The length of the bit vector being $2^d - 1$, the various possibilities of the bit vector turn out to be $2^{(2^d-1)}$. The bit vector is a set of subspaces represented in the form of a bit-array. So, the number of possible unique bit vectors is the powerset of the set of subspaces ($=2^{(2^d-1)}$); very exponential in terms of d.

It has been observed that the actual number of k-nn pairs in a set of subspaces in a data set is a lot less than this maximum number. In the worst-case, the maximum number of colors is $n \times n$ or $2^{(2^d-1)}$, which ever is lesser of the two. Since the data sets dealt with have some clustering structure in them, the number of colors is far less than that.

But even a small percentage of such large number is large for the number of colors that a human eye can comfortably perceive at once. In order to reduce the number of colors used in Heidi visualization, the top-m frequent sets of subspaces are identified and colors are assigned to pairs that are close in these sets of subspaces accordingly. The remaining set of subspaces are not displayed for the obvious reason that they did not qualify in terms of various criteria like number of point-pairs that are close in a particular set of subspaces and the quality of the set of subspaces computed from SURFING technique [5].

2.5 Grouping and Ordering Points

The order of the data points along the rows and columns of the Heidi Matrix play a significant role in visualization; giving better visualization and understanding of the internal structure of the data. With no specific purposeful point-order (say randomized), the image will not have any valuable information to offer. Grouping the points based on the clusters or kNNs (when no clustering results are given) gives better visualization and understanding of the internal structure of the data set.

Heidi relies on pre-computed clustering results (given by the user). When these are not given, Heidi uses the results obtained from the Stability-based RECORD [9] clustering algorithm over the data set in grouping the points cluster-wise. The RECORD algorithm is used because it does not need any parameters. Heidi can also incorporate the results of any other clustering algorithms like DBSCAN [7], PreDeCon [8] and PROCLUS [2], provided the parameters are set properly by the user. Heidi images over various data sets are generated using RECORD clustering results to group the points. These images are shown in Section 4.

Given a set of clusters and noise points, the order is created by first placing all the points belonging to cluster 1 together, followed by cluster 2 and so on. After all the clusters are covered, the noise points are appended to the list. This list is

then used to re-order the points within each cluster group in Heidi Matrix accordingly. The first point along the row and column refers to the first point in this ordered-list, the i-th point along the row and column refers to the i-th point in the order. Heidi uses kNNs and distance between the points in a cluster and the respective cluster center in obtaining two different point-orderings. The two strategies employed within each cluster are:

kNN ordering
Points are ordered based on the corresponding k-nearest neighbors of the point. This order is easily obtained by executing depth-first traversal on the kNN sets of the points. The traversal is started from the closest point to the origin.

Cluster spiral ordering
For each cluster, the cluster points are internally ordered based on their distance to the center of the cluster. The center of the cluster is the average of all points in the cluster. The value of each dimension of the center is set as the arithmetic mean of the values of the same dimension across all the points in the cluster. The first point in this order, is the point closest to the center and towards the end of the ordering, the points are farthest to the cluster center. So, the beginning cell of any block B_{ij} (C_i along the rows, C_j along the columns), say (p, q) indicates that point p is closest to the center of cluster C_i and q is closest to the center of the cluster C_j.

To gain additional insight into the cluster overlap in lower dimensional subspaces, the above mentioned ordering techniques also generate various point-orders by computing distances based on the indicated (by the user) subspaces and getting the point-order pertaining to a particular subspace. This is illustrated in the following example and also in various experimental results.

Example: As an example, consider a 5 dimensional (represented as 5-D) data set of size 1000 points and having 5 clusters. The data set is generated using SynDECA [10]. Table 1 displays a set of five Heidi images for the same with points ordered in various subspaces (mentioned below each image). The point-ordering technique used in this example is cluster spiral ordering. It can be observed that the images have 5 × 5 blocks, cluster blocks along the diagonal have prominent colors than compared to the rest of blocks. Such prominence indicates closeness of points within a cluster. Blocks away from the diagonal have lesser color and presence of single color strips indicate overlap of clusters (by overlap we refer to the geometrical overlap). When points are ordered based on a single-dimensional subspace, the corresponding Heidi image gives more insight on overlap of clusters along that particular subspace.

2.6 Composite Heidi Visualization

Each ordering based on a subspace gives a different Heidi image (as shown in Table 1). Instead of observing various Heidi images to check possible overlaps between clusters, Heidi creates a composite Heidi image by considering single color patterns that occur in a block across all the Heidi images which are ordered based on single-dimensional subspaces. Such a composite Heidi image gives a higher level information about the overlap of points within a cluster and points between clusters across all the dimensions.

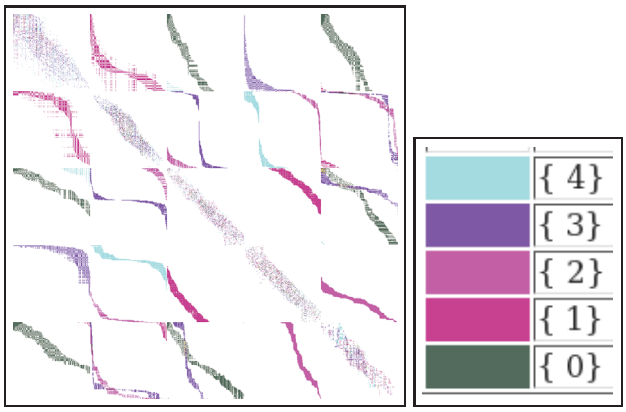


Figure 1: Composite Heidi Image of 5-D data set, with color legend

Example: Heidi's composite visualization of the 5-D data set is shown in Figure 1. The Heidi image has 5 × 5 blocks, blocks along the diagonal have a uniform diagonal multi-color strip indicating clusters. The blocks away from the diagonal have various color strips; these strips indicate the overlap of clusters in various dimensions. Numbering the blocks 1 to 5 from left to right and top to bottom; observe blocks at positions (1,3) and (1,5). These blocks have a single green color strip. The green color corresponds to subspace {0} (see the color legend provided next to the Heidi image), implying that cluster pairs (C1, C3) and (C1, C5) overlap along dimension {0}. Multiple color strips in a single block indicate overlaps between a pair of clusters across more than one dimension. Observe blocks at positions (3, 2) and (5, 3). Both these blocks have two different color strips. Block (3, 2) has strips with corresponding colors of subspaces {3} and {2}; indicating that cluster pair (C3, C2) overlap along dimensions {3} and {2}. Similarly, block (5, 3) indicates overlap between cluster pair (C5, C3) in subspaces {0} and {3}. Notice that the composite Heidi image lists out only a few colors, colors corresponding to prominent subspaces; leading to a Heidi image that only displays color strips, eliminating noise (colors corresponding to lesser prominent subspaces).

3. HEIDI VISUALIZATION

Given a d-dimensional data set of size n points and corresponding set of clusters (obtained from any clustering algorithm) as input, Heidi generates a visualization that is a multi-colored, block segmented image; colors representing the various subspaces and blocks represent the various clusters. Summarizing, each element (p, q) in the Heidi matrix is a bit vector and is represented as a pixel in the Heidi visualization. The bit vector for a pair (p, q) stores closeness information if p considers q as one among p's kNNs in a particular subspace. The integer value of the bit vector is then used to obtain a unique color for the pixel, indicating the closeness of points p, q in the corresponding relevant set of subspaces. The data points are placed along the rows and columns in a particular *order* computed based on the earlier mentioned ordering techniques. The ordering of points along the rows and columns, and the organization of colors

in the Heidi matrix help in identifying and analyzing the Heidi blocks (i.e., cluster blocks). The length of the Heidi block along the row and the column indicates the sizes of the clusters, whose points are ordered along the rows and columns of the block. A row or a column in the Heidi matrix captures the nearest neighbor information between a point and other points across various subspaces.

Different colors in a single cluster block: If a Heidi block has different colors, it signifies that the set of corresponding cluster points have different sets of kNNs in various subspaces. Hence, when there are subspace clusters present, the blocks obtain different colors due to the definition of these clusters in different subspaces. A color legend is provided to the user which indexes the color to its corresponding subspace. A sample color legend for a data set is given in Figure 2. Below each color is the set of subspaces it represents.

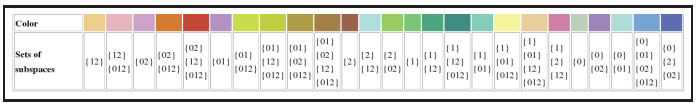


Figure 2: Color legend for the example

3.1 Interpreting the visualization

Consider an example data set shown in Figure 3(a); size of the data set is 2000 points and dimensionality = 2 (say $\mathcal{D}$={0, 1}). It can be observed that the data set has two well-separated clusters with no noise points. The cluster on the top is labeled as C1 and the other as C2. Both the clusters are of the same size (1000 points). The clusters overlap in subspace {0}.

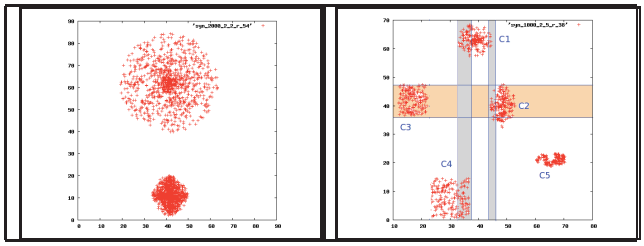


Figure 3: (a) Two clusters and (b) Five clusters

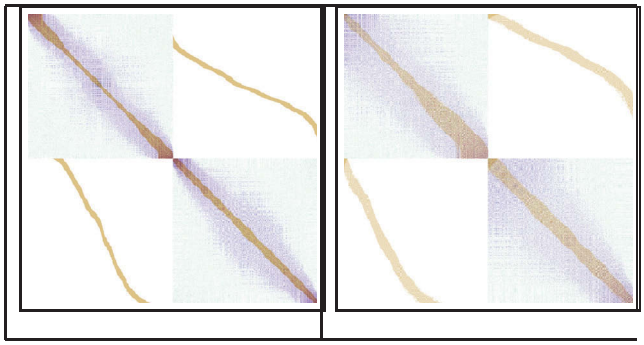


Figure 4: (a) kNN ordering, (b) Cluster spiral ordering

Figures 4 (a) and (b) display the Heidi images obtained by different point-orderings.

kNN point-ordering
The points are ordered in 4(a) using the kNN technique. Points belonging to cluster C1 are followed by the points belonging to cluster C2, hence the cluster blocks B_{11}, B_{12}, B_{21} and B_{22} (by numbering the blocks 1 to 2 along the rows and columns in the figure) are easily identifiable in the Heidi image.

Cluster blocks B_{11} and B_{22} have the points that belong to the cluster ordered along the rows and columns of the block. Patterns occurring in these blocks reflect upon the closeness between points belonging to a cluster. Observing the B_{11} and B_{22} cluster blocks in Figure 4(a), it can be noticed that the blocks have predominantly three colors, sky blue, violet and brown. The partial color legend is as follows: brown ($\{0\}$), skyblue ($\{1\}$) and violet ($\{0, 1\}$). There is little prominence of color corresponding to other combinations of subspaces in the figure.

The kNN point-ordering technique is depth first traversal of kNN relations starting from the point closest to origin, any two sequentially occurring points in the order will be close to each other, not just in terms of distance values, but also in their kNN sets across various subspaces. A continuous pattern in brown color running along the diagonal of the cluster block, as a narrow strip. A slightly broader violet strip around the brown strip indicates those point-pairs which fail to be kNN in the $\{0, 1\}$ space, but share kNN relations in $\{1\}$ space.

Cluster blocks B_{12} and B_{21} have different clusters and the patterns that occur in these blocks help in understanding the relative placement of the clusters and their overlap in other subspaces. Blocks B_{12} and B_{21} (any B_{ij} and B_{ji}) are almost similar and they are transpose of each other (with a few exceptions, since kNN relations are not symmetric). A brown narrow strip of color running along the blocks can be observed. Observations and conclusions on B_{21} (or B_{12}) are as follows: (i) the strip spans across all the columns of the block (horizontally). Hence, it covers all the points belonging to C2.
(ii) the strip spans the rows of the block partially (vertically); implying that it covers only a subset of cluster C1.
(iii) the color of the strip is brown, and the corresponding subspace is $\{x\}$.
(iv) All points in cluster C2 participate with a subset of cluster C1 in kNN relations in subspace $\{0\}$. This indicates that cluster C2 and cluster C1 overlap in the $\{0\}$ subspace. A subset of C1 sharing kNN relations with the whole cluster indicate that the span of cluster C1 in subspace $\{0\}$ is larger than that of cluster C2 and it encloses the span of cluster C2. The span i.e., the maximum and minimum values of 0 for cluster C1 is [15, 65] (roughly). The span of C2 is [30, 50] which is a part of [15, 65].

Cluster spiral point-ordering
The Heidi Matrix in Figure 4 (b), the points are ordered using the cluster spiral point-ordering technique. In the cluster spiral approach, the top-leftmost beginning point in a cluster block is the point closest to the centre of the cluster represented along the rows of the block. In Figure 4(b), the brown strip in cluster blocks B_{12} and B_{21} start at the top-leftmost point, implying that the centre of cluster C2 is

close to the centre of cluster C1 in the subspace $\{x\}$. If the beginning of the color strip does not start at the beginning of the block, it implies that the center of cluster placed along the rows is is distant to the center of the cluster placed along the columns.

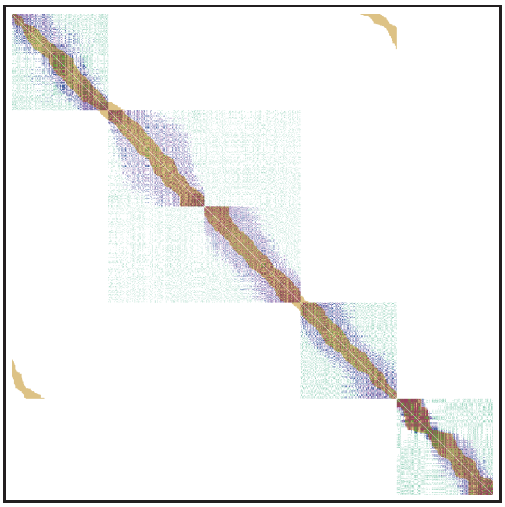


Figure 5: Heidi Image using kNN point-ordering of Dataset 3(b)

Figure 3(b) displays another 2-D data set, having five clusters. There are three cluster overlaps in subspaces {0} and {1} as indicated by the red and blue color bands in the data set figure. The corresponding Heidi Matrix obtained from kNN point ordering based on subspace {0} is displayed in Figure 5. Numbering the cluster blocks from 1 to 5 along the rows and columns, blocks B_{12} and B_{21} have a small brown patch, near the point where blocks B_{11} and B_{22} meet. This indicates that cluster C_1 and C_2 overlap along {0} subspace. Similarity, blocks B_{14} and B_{41} have a brown color strip, again indicating a cluster overlap along {0} subspace. Blocks B_{23} and B_{32} have the whole block filled with skyblue color, indicating that the clusters $C2$ and $C3$ completely overlap along subspace {1}.

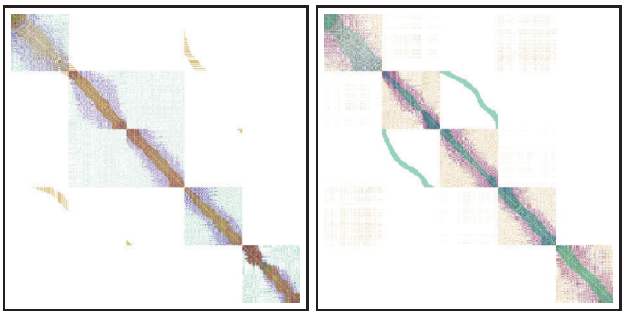


Figure 6: (a), (b): Heidi Images using Cluster Spiral of Dataset 3(b)

The Heidi Visualization obtained from cluster-spiral ordering based on subspaces {0} and {1} are displayed in Figures 6(a) and 6(b). The corresponding composite Heidi image of these two subspaces is shown in Figure 7, along the color legend of the subspaces.

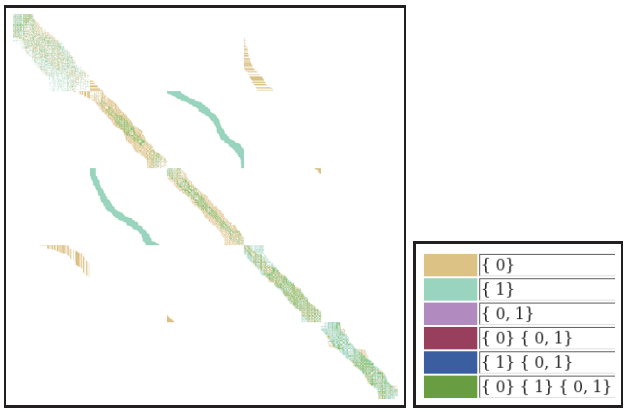


Figure 7: Composite Heidi Image of Dataset 3(b)

3.2 Reading Heidi Images

1. Within the same cluster, color arrangement along the diagonal of the cluster indicates the shape of the cluster.

2. In the inter-cluster blocks (like B_{ij}, $i \neq j$), presence of color strips indicate overlap of clusters in a particular set of subspaces.

3. The spread of the color strip along the rows and columns of the block indicates if the clusters overlap completely (i.e., all points in C1 share kNN relations with all points in C2) or partially (i.e., a few points in C1 share kNN relations with all or few points in C2 or vice-versa).

4. In the cluster spiral approach, the start of the color strip conveys the relative position of a cluster's centre with respect to that of the other cluster.

3.3 Reading Composite Heidi Images

1. In the inter-cluster blocks (like B_{ij}, $i \neq j$), presence of a color strip indicates overlap of clusters the corresponding set of subspaces. Multiple color strips imply overlaps in different sets of subspaces.

2. The spread of the color strip along the rows and columns of the block indicates if the clusters overlap completely (i.e., all points in C1 share kNN relations with all points in C2) or partially (i.e., a few points in C1 share kNN relations with all or few points in C2 or vice-versa).

3.4 Implementation Details

The system is developed in GNU C/C++ on a Linux (Fedora Core 9) system. All the images are generated in PPM image file format (*.ppm* files) which are later converted to JPEG image file format (*.jpg* files). All tests were run on a LINUX (Fedora Core 9) system featuring a 1.77GHz processor and 2GB RAM.

Complexity: The complexity of computing the Heidi Matrix is $\mathcal{O}(2^d \times n \times n)$. Space complexity is also the same. But as d > 8 and n>2k, memory allocation for the Heidi Matrix is an issue. The time taken to compute the Heidi matrix for the examples shown in the paper ranges between 1-5 minutes. A Heidi image for a particular subspace is obtained

in a few seconds time (< 1 minute). As the dimensionality increases, the computation time also increases. For example, for the 50-D data set, Heidi took nearly 16 minutes to compute the Heidi matrix and generate all the Heidi images and composite Heidi image (shown in Figure 9).

Efficiency add-ons In order to speed up Heidi visualization, one solution would be to reduce the number of subspaces to be considered. The user could either mention the "interesting" subspaces based on a priori domain knowledge or techniques like SURFING [5] could be used to reduce the number of subspaces. Reducing number of subspaces, also helps reduce the memory requirements to store Heidi Matrix. Heidi uses Radix trees to store the bit-strings uniquely and also to serve as an index structure (easy access to bit vectors through sorting).

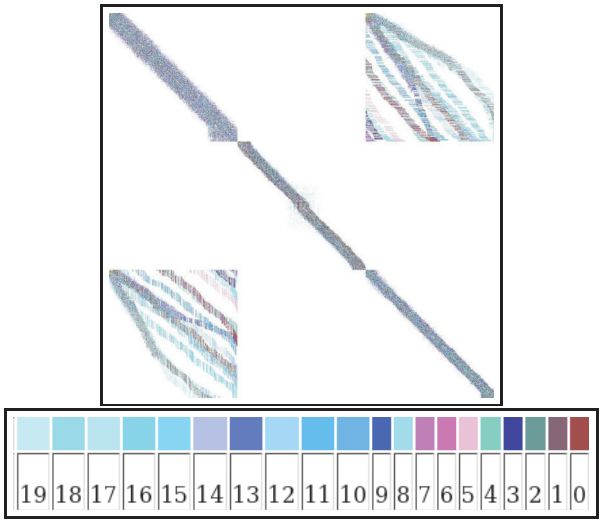


Figure 8: Composite Heidi Image of 20-D data set

4. EXPERIMENTAL RESULTS

4.1 High-dimensional data examples

Composite Heidi images on a 20-D data set (1500 points, 3 clusters) and a 50-D data set (1500 points, 3 clusters) are shown in Figures 8, 9. Of the three clusters, the first and the third cluster overlap in certain subspaces, while the second cluster does not overlap with any clusters in all subspaces. Numbering the visible cluster blocks from 1 to 3, along the rows and columns; observe the blocks at 1,3 and 3,1. When there is an overlap between clusters C_1 and C_3, the corresponding blocks have color in them. While the cluster C_2 does not overlap with the other two, hence blocks at (2,1), (1,2), (2,3) and (3,2) have no color. Corresponding color legends are given below the composite images. In Figure 8, the inter-cluster block corresponding to cluster 1 and cluster 3 (at position 1,3 in the Heidi blocks) has 15 color strips, implying that cluster 1 and cluster 3 overlap in 15 dimensions of the whole 20-D space. Similarly in Figure 9, there are multiple color strips in inter-cluster block corresponding to cluster 1 and cluster 3. Different lengths of color strips indicate the percentage of overlap between the clusters along each subspace.

A number of data sets are generated using SynDECA [10],

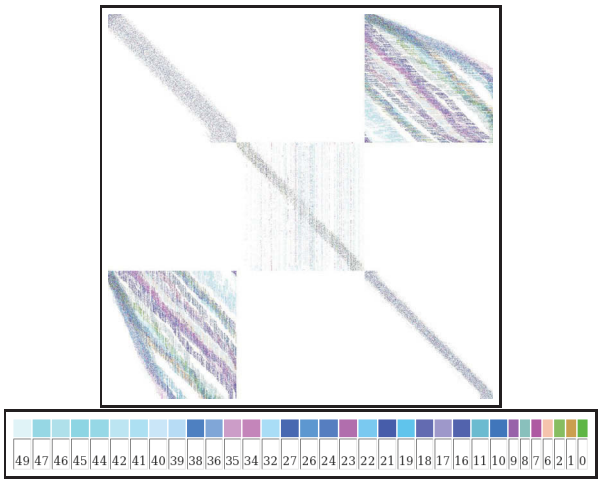


Figure 9: Composite Heidi Image of 50-D data set

with special requirements; presence of noise, presence of subspace clusters and different sized clusters. All Heidi images displayed earlier before this section are all tested on synthetic data sets with different sizes n=1000 and 2000, different number of clusters c=2, 3 and 5 and various dimensionalities d=2, 4, 10, 20, 40 and 50.

4.2 VAST08 Challenge Data: Migrant Boats Dataset

We present results of VAST Mini Challenge 2 by running Heidi over the raw data set after some pre-processing (mentioned below). Heidi is executed over the Migrant Boats (for geo-temporal analysis) data set of VAST 2008 Data Challenge (Mini challenge 2); to evaluate useful information delivered by Heidi about the data. The original data set is a mixed-attribute data set (having numerical, temporal and categorical attributes) and has the following attributes {EncounterCoordsx, EncounterCoordsy, RecordType, Passengers, USCG_Vessel, EncounterDate, RecordNotes, NumDeaths, LaunchCoords, VesselType}. A total of 917 records (hence a data size of 917 points) are available in the data set.

We modified the original data set such that categorical attributes are converted into numerical attributes, by assigning each unique value of a categorical attribute a unique integer value. A standard numerical value of '-1' is set wherever the attribute value is missing. Due to lack of time, we could not incorporate temporal attributes in Heidi; so, the temporal attributes are currently eliminated for the analysis. A set of six dimensions are finalized: {EncounterCoordx, EncounterCoordy, Passengers, USCG_Vessel, NumDeaths, VesselType}. Attributes {LaunchCoordsx, LaunchCoordsy} are also eliminated as there are many missing values. The data set is divided into two clusters; points in the original data set having 'RecordType' attribute set to 'Interdiction' into cluster Interdiction C1 and points having 'Landing' as cluster Landing C2. Note that, these clusters are not generated by any clustering algorithm, but are taken from the data with the help of a binary attribute 'RecordType'. This leads to inclusion of noise in the clusters and hence, Heidi images might have too many colors thus forming noise in the visualization. The Composite Heidi matrix tries eliminating the

problem of too many colors by only displaying prominent patterns.

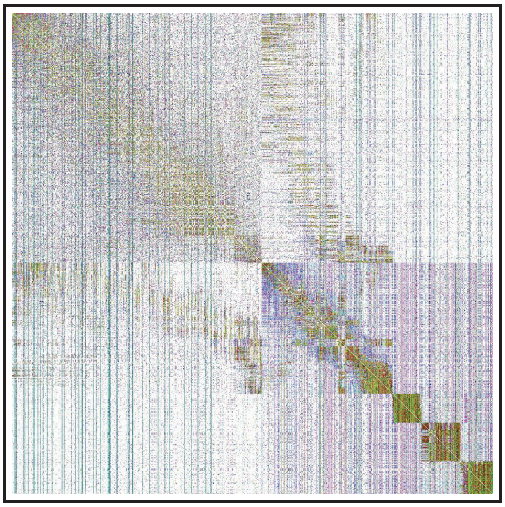


Figure 10: Heidi image of MigrantBoats data set over subspace EncounterCoords

The aim of the mini VAST 08 challenge is to use visual analytics to explain migration patterns during the years. For this, we generated a set of Heidi images for various meaningful, correlated subspaces to understand the patterns; as these images generated over selected subspaces can give more patterns than the Composite Heidi image. As the number of clusters = 2, the Heidi images predominantly have 4 color blocks.

Heidi image over EncounterCoords attributes: Figure 10 is the Heidi image of the data set obtained by ordering points based on subspace {EncounterCoordsx, EncounterCoordsy}. It is interesting to see a partial overlap between the clusters (see blocks at positions (1, 2) and (2, 1)); Interdiction C1 completely overlaps with Landing C2; implying there are a few points in Landing C2 whose landing coordinates are spatially closer to those of Interdiction C1 cluster points. There are solid blocks of green color towards the bottom of the diagonal in cluster Landing C2. This indicates well separated small groups of EncounterCoords which spatially form a dense area of close coordinates.

Heidi image over Passenger, NumDeaths attributes Figure 11 is the Heidi image obtained by ordering points based on subspace {Passengers, NumDeaths}. For the cluster blocks, there is a strong multi-color diagonal and between the clusters too, there is an overlap (the color strip).

Composite Heidi Image over 1-D and 2-D subspaces: is generated and is shown in Figure 12. The color legend of a few prominent sets of subspaces is shown in the Figure 12 due to space limitations. The Composite Heidi Image is not symmetric because of the presence of categorical attributes (in spite of mapping them to integers); ordering the points in Heidi Matrix based on subspaces of categorical attributes needs to be addressed to resolve this.

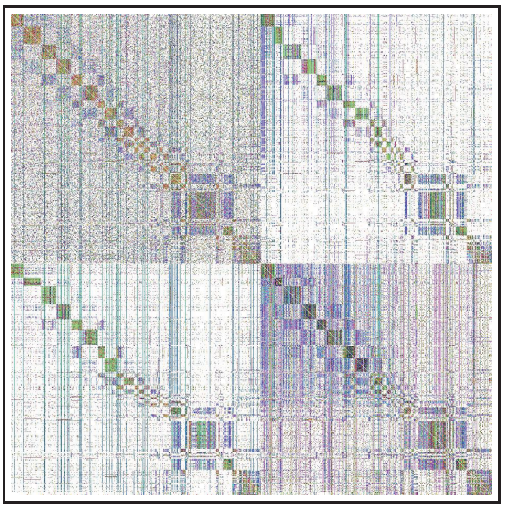


Figure 11: Heidi image of MigrantBoats data set over subspace Passengers, NumDeaths

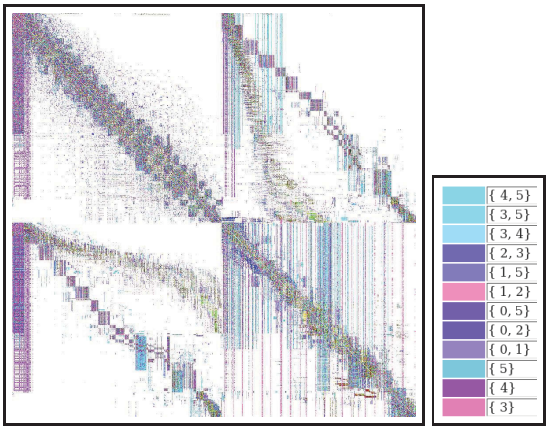


Figure 12: Composite Heidi image of MigrantBoats data, with color legend

4.3 Real-life data set: NBA data set

Heidi is used to visualize NBA players' data set. The data set contains the average match statistics and has five attributes: {Average number of minutes played per match, Average field-goals made, Average free-throws made, Average rebound per match, Average points per match}. The 5-D data set has 310 player entries. The Heidi images of this data set for various subspaces are shown in Figure 13. From the images, three clusters and noise can be observed. These images are obtained by changing the point-order along the rows and columns based on the cluster spiral ordering technique based on distances computed over various subspaces. This is done with the purpose of analyzing how informative are the attributes.

Figures 13(a,b,c,d) are computed based on the single dimensional subspaces {Avg mins}, {Avg field goals}, {Avg free throws}, {Avg rebound}. Figures 13(e,f,g,h) are computed based on the two-dimensional subspaces {Avg mins, Avg field goals}, {Avg free throws, Avg rebound}, {Avg mins, Avg free throws} and {Avg field goals, Avg rebound}. Figures 13(i,j) are based on subspaces {Avg mins, Avg field

goals, Avg free throws, Avg rebounds} and all the dimensions (original dimensional space). It can be observed that for Heidi images obtained by ordering points based on single-dimensional subspaces, the visualization helps in identifying the overlap between clusters (the color strips in all the inter-cluster blocks). Subspaces {Avg field goals} and {Avg free throws} have no clustering structure, as can be seen in the Figures 13(c) and 13(d), there are color strips between all inter-cluster blocks. That is, all the clusters overlap with each other along these two subspaces. Figures 13(a,e) though look similar, have subtle differences in the prominence of color along the diagonal. Same is the case with the pair of figures 13(e,g) and (f,h).

5. RELATED WORK

Evaluating the data set before decision-making is crucial to ensure accurate operations. Good visualization systems are needed to help the user in making timely decisions.

Early visualizations: Earlier techniques visualize data projected on 2 and 3 dimensions. The 2 or 3 dimensions are chosen based on their 'usefulness' or 'importance'. Multi-dimensional scaling and principal component analysis are other techniques where data is transformed into a lower dimensional space, which makes it easy for visualization. The problem with transformed domains is that it is not clear what the new transformed dimensions mean; there is no semantic information of these.

Gray-scaled distance matrix image: Another technique to visualize data was to visualize the distance matrix by binning the distance values in 255 bins. Each bin is associated with a gray-intensity (ranging from 0 to 255). Depending on the image format, the range could differ; for example, ppm format supports a range of 0 to 65536.

Distance matrix is an $n \times n$ matrix, each cell (p, q) representing the distance between the points p and q. The distance matrix so computed is normalized and all the values are multiplied with 255 (or the highest intensity based on the image encoding format) to obtain the corresponding gray-scale intensity. The closer the points are to each other, the lesser the distance between them. Hence, the gray intensity tends towards the extreme. The distance matrix image is informative even if the data is high-dimensional, because it is only the distance that is taken into consideration, not the individual dimension values, though it does not give any insight on the subspace structure in the data.

Heidi vs. Gray-scaled Distance Matrix: Heidi is inspired from the gray-scaled image of a distance matrix used to visualize clusters. The traditional gray-scaled distance matrix is used to represent the distances calculated using all the dimensions in the data set. Heidi uses distances computed in all subspaces and then semantically defines closeness relationship between the points using kNNs. It uses this definition to identify closer points in all the subspaces and displaces them using a (R, G, B) color model; color being used to bring out the subspace structure in the data set.

Generalized Association Plots [6]: Generalized association plots use correlation matrices to obtain a two color intensity visualization. Correlation ϕ is applied over the initial proximity matrix (the proximity matrix could be a correlation matrix to start with; attribute correlations or it could be a distance matrix of a data set), and correlation is re-applied on the new matrix and so on. A series of visualizations are obtained, as each visualization seems to form a step in a virtual hierarchical clustering algorithm based on correlation coefficient. Ordering the attributes along the rows and columns, if the proximity matrix is an attribute correlation matrix is discussed in detail, with a few ranking strategies.

The paper mentions that there is no way of displaying relationship between the points and the set of attributes; on the contrary, Heidi does effectively show the structure of points when considering various sets of attributes. The aim of Heidi differs from that of Generalized Association plots; Heidi emphasizes on subspaces (sets of attributes) and how points are structured in various subspaces.

VISA [3]: The existing literature on subspace clustering does not provide a visualization of results. *VISA* is a first step towards subspace clusters visualization, though it uses Multi-dimensional scaling across various subspaces. It is built to visualize subspace clusters and the subspace cluster hierarchy obtained from a subspace clustering algorithm developed by them. VISA only displays subspace clusters and their interaction across various subspaces as cluster-overlap (member set intersection: overlap of two clusters C1 and C2 = (C1 $\cap$ C2)) across subspaces. It does not give information if the two clusters C1 and C2 geometrically overlap in the Cartesian coordinate system. Also, it does not have any spatial relationships between points (both inter-cluster and intra-cluster) across subspaces.

The objective of these systems is different from that of Heidi's; Heidi encompasses closeness between points across all the subspaces. Heidi irrespective of the clustering results, displays the overlaps based on kNNs across subspaces. In any case, Heidi is a utility for visualizing high dimensional data based on subspaces and it could also use the results obtained from subspace clustering algorithms in visualization. For example, by using color dial and k dial, different aspects of subspace relationships and depth of the relationships can be visualized.

6. SUMMARY

Heidi Matrix displays the overlap of clusters in high dimensional data across various subspaces. Such images help us to place the clusters of high dimensional data with respect to one another. In particular, Heidi visualization system:

1. Generates two dimensional images for higher dimensional data sets.
2. Presents nearest neighbor proximity information among data points.
3. Presents spatial overlap among clusters in various subspaces.
4. Presents all this information for analysis in a single Heidi image.

Our experimental results on synthetic and real data sets show the spatial relation (in subspace) among clusters of high dimensional data sets. In our on-going work, we are addressing efficiency aspects of generating Heidi images over large higher dimensional data sets.

Acknowledgement: We would like to thank the reviewers for their helpful suggestions.

7. REFERENCES

[1] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional space. In *Lecture Notes in Computer Science*, pages 420–434, 2001.

[2] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park. Fast algorithms for projected clustering. In *Proc. ACM SIGMOD*, pages 61–72, 1999.

[3] I. Assent, R. Krieger, E. Muller, and T. Seidl. Visa: Visual subspace clustering analysis. In *SIGKDD Explorations (1)*, 2007.

[4] U. Axen and H. Edelsbrunner. Auditory morse analysis of triangulated manifolds. In *Mathematical Visualization*, pages 223–236. Springer-Verlag, 1998.

[5] C. Baumgartner, C. Plant, K. Kailing, H. P. Kriegel, and P. Kroger. Subspace selection for clustering high-dimensional data. In *Proc. ICDM*, pages 11–18, 2004.

[6] C.-H. Chen. Generalized association plots for information visualization: The applications of the convergence of iteratively formed correlation matrices. volume 12, pages 1–23. Statistica Sinica, 2002.

[7] M. Ester, H. P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. ACM SIGMOD*, pages 226–231, 1996.

[8] K. Kailing, H. P. Kriegel, and P. Kroger. Density-connected subspace clustering for high-dimensional data. In *Proc. ICDM*, 2004.

[9] S. Vadapalli, S. Valluri, and K. Karlapalem. A simple yet effective data clustering algorithm. In *Proc. ICDM*, pages 1108–1112, 2006.

[10] J. Vennam and S. Vadapalli. Syndeca: Synthetic generation of datasets to evaluate clustering algorithms. In *COMAD*, 2005.

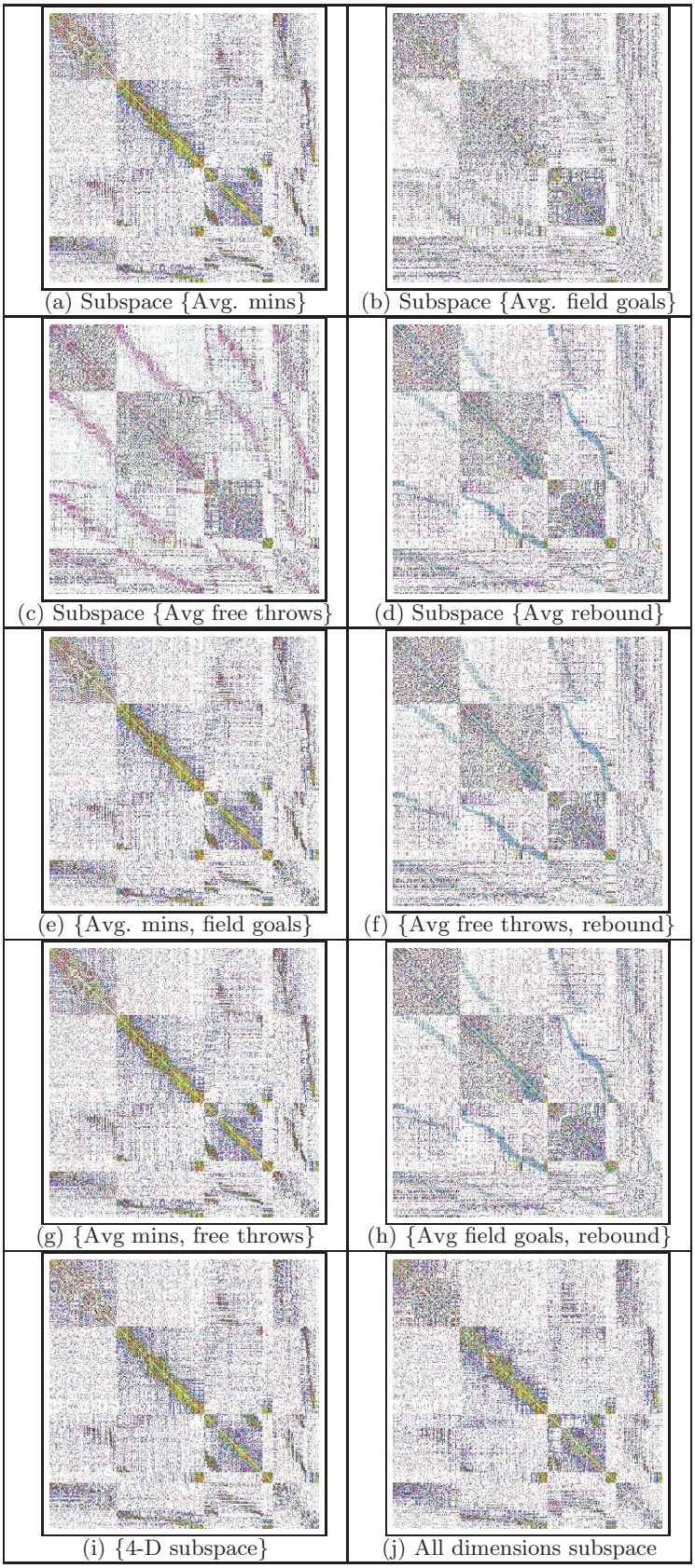


Figure 13: Heidi images of NBA data set for different Subspaces