A Visual and Interactive Data Exploration Method for Large Data Sets and Clustering

David Da Costa[1,2] and Gilles Venturini[1]

[1] Laboratoire d'Informatique de l'Université de Tours, France
{david.dacosta,venturini}@univ-tours.fr
[2] Cohesium, France
ddacosta@cohesium.com

Abstract. We present in this paper a new method for the visual exploration of large data sets with up to one million of objects. We highlight some limitations of the existing visual methods in this context. Our approach is based on previous systems like Vibe, Sqwid or Radviz which have been used in information retrieval: several data called points of interest (POIs) are placed on a circle. The remaining large amount of data is displayed within the circle at locations which depend on the similarity between the data and the POIs. Several interactions with the user are possible and ease the exploration of the data. We highlight the visual and computational properties of this representation: it displays the similarities between data in a linear time, it allows the user to explore the data set and to obtain useful information. We show how it can be applied to standard 'small' databases, either benchmarks or real world data. Then we provide results on several large, real or artificial, data sets with up to one million data. We describe then both the successes and limits of our method.

1 Introduction

Visualization techniques have been studied since several years now and try to provide useful techniques in the data mining process [1]. Among the contributions of visual techniques to data mining (DM) and knowledge discovery in databases (KDD), one may mention the ability to easily analyze data or knowledge through a quickly understandable visualization, or the ability to allow the user formulate (or reformulate) queries in a graphical and interactive way.

One of the limitations and drawbacks often outlined for visual data mining (VDM) techniques is the handling of large data sets. When confronted to a large amount of data, many standard methods fail to produce high quality results in a reasonable amount of time. In the context of clustering for instance, it is known that ascending hierarchical clustering is an efficient method, but limited to small data sets, and therefore other work has been developed to overcome this limitation (Cure [2], Birch [3]). The same remark can be made with VDM techniques: the complexity of the visualization process is often too important to handle large data sets. So one of the most important aim and obtained result of this work is to show that VDM techniques based on points of interest may handle data sets with up to 1 million data in a reasonnable time and with useful interaction techniques.

R. Alhajj et al. (Eds.): ADMA 2007, LNAI 4632, pp. 553–561, 2007.

More precisely, we detail in this paper a new generic method for visualizing and exploring symbolic and numerical data (see the initial description in [4]). Our visual representation is an adaptation of the methods that use points of interest in the context of information retrieval in textual data [5]. Our objectives, in addition to those pursued by VDM techniques, are: 1) to be able to represent all types of data (many VDM techniques consider numerical data only), 2) to provide the domain expert with information and knowledge on the similarities between data and to allow him or her to perform interactive actions, 3) to display the data in linear time (a necessary condition in order to use dynamic display and to analyze very large data sets), 4) to require the shortest possible training time for potential users who are not considered as experts in data analysis.

The remaining of this paper is organized as follows: section 2 describes the state of the art in VDM with large data sets where we highlight the current limitations of traditional approaches. We also present in this section the existing methods based on points of interests. In section 3, we detail our approach: the selection of points of interest, the placement of the data, the interaction with the visualization (change of points of interest, zoom, queries). In section 4, we describe first the properties of our method. We provide typical results obtained on benchmark data as well as results on a real world application. Then we apply our method to large databases and provide computational results. Section 5 concludes on this work and proposes several perspectives.

2 State of the Art

2.1 Visual Data Mining Techniques and Large Databases

We review in this section the properties, possible behavior and limitations of existing VDM techniques when dealing with large databases.

There are several known methods to visualize and interact with data in a VDM process. One can mention "historical" methods such as Chernoff' faces [6]. This method represents each data in the form of a face, and it relies on the fact that the human brain easily analyzes the resemblances and differences between faces. However one may mention that occlusions and overlap between the displayed data is critical in this method when dealing with large data sets. The same comment can be made with parallel coordinates which is not well suited when the dimension of the data is high [7] [8]. One may mention also the "scatter plots" [9] which make it possible to obtain multiple views. Multiple views are however difficult to use when several thousands points must be considered.

Other more recent work is better suited for analyzing millions of data, like the pixel-oriented methods. A famous example is VisDB [10] which represents each data with a colored pixel which indicates the relevance of an article with respect to the user query for instance. Other examples have been studied like in the Tree-Map representation where hierarchical data can be displayed (with up to one million data) [11].

In general, it sounds difficult to try to visualize the dimensions of the data when dealing with large data sets (with many data and many dimensions). Therefore, we have concentrated our efforts on methods that are based on a similarity measure between the data. One possible method is to display directly the similarity matrix [12], but we argue that this method is not as intuitive as the one we propose in this paper.

2.2 Motivations for Using Methods Based on Points of Interest

One can notice that in general many VDM methods easily handle numerical data, whereas the representation of symbolic data seems to be much more difficult. The reason for this is the following one: defining a mapping between numerical attributes and visual attributes such as the position, length, orientation, seems rather "straightforward", while for symbolic values the standard visual representations are less adapted. Moreover, many VDM techniques try to visualize the dimensions of the data, and are thus limited to data sets with a small number of dimensions (or may require the use of attribute selection algorithms), like in the parallel coordinates for instance [7] [8].

Another point to consider is the learning time and adaptation required from the user which can be prohibitive. For instance, with the parallel coordinates, detecting a correlation between attributes is possible but requires considerable learning in order to detect the specific visual pattern.

We are thus interested in this paper with known techniques based on points of interest, but which have not really been used yet in symbolic/numerical data visualization. Among these methods, we can mention systems like VIBE [5], SQWID [13], Radviz [14] or Radial [15], which were used as methods for exploring a set of documents (in general, the results of a search engine).

3 Our Method

3.1 Basic Principles of Visualization

We consider a data set compound of n data $D_1, ..., D_n$ and a similarity matrix "Sim" between these data. $Sim(i, j)$ denotes the similarity between D_i and D_j. If $Sim(i, j) = 1$ then D_i and D_j data are identical, and if $Sim(i, j) = 0$ then they are completely different. This matrix is thus symmetrical and its diagonal is filled with 1s. It is the only assumption that we make about the data, and from this point of view, this allows us to be independent of data types (we may represent numerical/symbolic values, but also texts, images, etc), and of data dimensions. The similarities summarize all the information.

The basic principles of our method are the followings (more complex extensions will be explained in the next sections): initially, we will consider that the POIs are a subset of these data, and are denoted by $D_1, ..., D_k$. We display these k data on a circle, at equal distances.

If D_i is identically similar to all of the POIs, it will be displayed in the middle of the disc. On the opposite, if it is completely similar to one POI and completely different from the others, its position will be confounded with that POI. If its similarity is biased toward certain POIs, then it will tend to approach these POIs.

More generally, our method is such that two data close to one another in the initial representation will thus be also close with respect to POIs, and they will thus be close in 2D space. The visualized space thus becomes a space of distances between selected points (POIs) and the data. It is in this manner that this method can represent a set of data of any type. On the other hand, the reciprocity of this property is not true all the time : two data which are close in the 2D space are not necessarily close in the original space (all the points at equal distance of two POIs in the initial space form a mediating

line and are thus displayed at the same 2D location). It will be necessary to use other methods to remove these ambiguities (see the next sections).

3.2 Cutting Down the Complexity

In general, when one wants to display the similarities between n data, the associated complexity is often in $\frac{n \times n+1}{2}$: if one wants to correlate the displayed distance with the similarity between data, it requires the computation of the similarity between all couples of data. In our method, the computation of similarities is limited to $(n-k) \times k$ where k is less than 10 in general, but the 2D display highlights the similarities between all data but without computing them: data which are similar to each others will be located at the same distances to POIs, and will thus be close to each others in the 2D representation. Once those similarities are computed, the display of the data is immediate (linear time). This property is fundamental in order to allow quick interactions and to display very large databases (see section 4).

3.3 Interaction

To be really efficient, the visualization must be interactive and must make it possible to dynamically refine the display and to answer to users graphic requests. In a visualization with POIs, the user can ask for the following information: what is this data (or this POI), how to enlarge this part of the visualization (zoom without loss of context), how to change POI (to remove some, to add some, to change their order, and possibly to define POIs which are not necessarily some data of the initial database).

When the mouse is on a point, we thus indicate what is this point. The user may also select a group of points in order to perform interactive clustering, i.e. labeling the data (see section 4.2).

As far as the POIs are concerned, we have represented the main possible interactions: first of all, it is possible to remove a POI. This is done very simply by dragging a POI inside the disc. This POI takes its place back within the data. The view is dynamically recomputed. A dynamic and progressive transition is performed so that the user can follow the change of representation. He then has the possibility to cancel its action, which causes to put the POI back on the circle. It is also possible to choose a data and to define it as a POI. For this purpose, the user drags the data on the circle. If the data is placed on a POI, it replaces this POI, and if it is placed between two POIs, it is inserted between them. The length of the arcs between POIs is kept constant.These functionalities are very important since they allow the user to redefine at will the representation.

Two zooming techniques have been implemented, one being based on a standard hyperbolic transformation of the display, and the second one being a distortion of the similarity function.

The hyperbolic zoom is triggered when the user clicks on a point: it places the selected data on the center of the disc, it increases the area centered on this data and pushes back the other data toward the edges of the visualization. The distortion is calculated using a hyperbolic function. This zoom makes it possible to enlarge the display while preserving the context of the data. It uses Cartesian coordinates but with we are currently studying a new version with polar coordinates.

The second zoom performs a kind of thresholding on the similarities, reducing the similarities below that threshold and increasing the similarities above this threshold. The resulting visual effect is that data are attracted by their most similar sets of POIs. One observes the creation of "straight lines" in the representation and the "crushing" of the data toward the POIs. The data that remain in the center are those which are the less attracted by the POIs. This zoom thus makes it possible to remove ambiguities.

3.4 Extensions

Several simple extensions can be added to this initial representation.

First of all, the initial choice of the POIs must be carried out. Initially, we consider that if the data are supervised (a class label is available), then we take the first representative of each class as initial POIs. There will thus be as many POIs as classes in the first visualization suggested to the user. If the data are not supervised (no data labels exist), we choose the k random data. Other automatic choices are possible (and certainly more judicious), and we try here to suggest initial choices that the user will be able to interactively and dynamically modify according to what is displayed.

A second extension is motivated by considering the order of the POIs: if a great number of data are attracted by two POIs, then it is desirable that these POIs are close to each others on the circle. A critical situation would consist in placing these POIs in a diametrically opposed way, which would generate unreadable visualizations (many data in the center). We propose an interactive solution to this problem: when the user adds a POI in the representation, he may add this POI at the location indicated by its mouse click (drag and drop), but our tool may also insert this POI at the best possible location in the circle. Given a new POI D_{k+1}, this POI is inserted in the ordered list $D_1, ..., D_k$ between its two most similar POIs. In this way, this should avoid the critical situation mentioned.

Then we have changed the constant distance between POIs to a variable one. The length of the arc between two POIs is a linear function of the similarity between them. In this way, POIs which are similar to each others are represented at close locations on the circle.

Lastly, it is possible to generalize POIs from specific data to any point in the space of data representation, and even more generally to knowledge which represent hypothesis to be checked by the user. Thus, one can represent "ideal" data, not really existing, and according to which the user would like to position the real data. Also, it would be possible to represent for example a decision rule as a POI, and to place the data according to their matching with this rule. This functionality offers many perspectives by visualizing not only data but also knowledge.

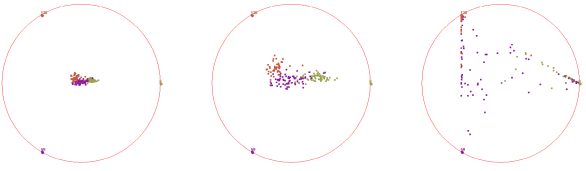
4 Results

4.1 Basic Tests and Illustrations

We illustrate the functionalities of our tool on standard databases. For this purpose, we have selected several databases with known characteristics from the Machine Learning Repository [16].

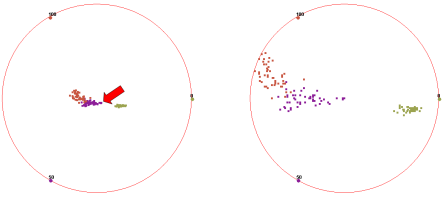
We present in figure 1 the effects of the zoom based on similarity distortion. Similarly, we have represented in figure 2 the use of the hyperbolic zoom.

We have represented in figure 2 the Iris data (150 data, 3 classes). In the Iris database, on may easily get correct information about the global shape of the classes (3 classes, 2 being non linearly separable). The same remark can be done for others databases. Our method is able to highlight outliers for instance, i.e. points which are far away from the others (and from more dense area).



(a) (b) (c)

Fig. 1. Visualization of Wine data without zoom (a), with the zoom based on similarity distortion (b) and with increasing zoom factor (c)



(a) (b)

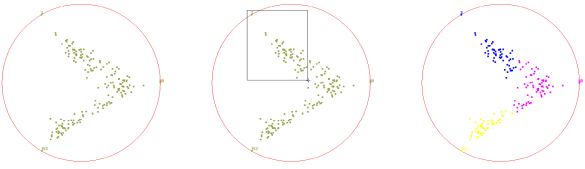
Fig. 2. Visualization of Iris data with data selection (a), and with hyperbolic zoom (b)

4.2 Interactive Clustering

In the case of an unsupervised database, our tool offers the possibility to create clusters from the 2D visualization. For this purpose, our method proposes to the user (or domain expert) a first visualization where the choice of POIs is not random (figure 3(a)). Then he has the possibility to select some displayed points and to assign them a label (figure 3(b) and (c)). We have represented this process in figure 3.

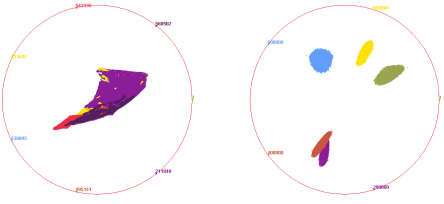
4.3 Dealing with Very Large Databases

We have applied our method to databases that contain up to one million data. The first database is the Forest CoverType data set as represented in figure 4. This data set contains a total of 581012 observations and each observation consists of 54 attributes, including 10 quantitative variables, 4 binary wilderness areas and 40 binary soil type variables. In our test, we have used all variables. There are seven forest cover type designations. We have represented first this database in figure 4.



(a) (b) (c)

Fig. 3. Visual and interactive clustering of "Wine" data



(a) (b)

Fig. 4. Visualization of the Forest CoverType data set (a) and the 1 000 000 artificial data set (b)

The second series of databases have been generated using uniform laws. Each data consists in a vector in dimension 10. Several classes are defined, and a different set of parameters of uniform laws is defined for each class. We generate in this way databases with an increasing number of data, from 10000 to 1000000 data. We have then displayed the largest database in figure 4. For this database, the total reading/displaying time is about 1 minute on a standard computer (Pentium M at 2 GHz with 1Go RAM). An interactive action such as the change of a POI takes about 2 minutes. In table 1, we have shown how the computational cost evolves with an increasing number of data.

As mention in section 3.2, our method has a linear complexity while being able to let the user perceive the $n \times n$ similarities between n data. More precisely, our method performs the following operations: the first pass consists in reading the database for normalizing numerical attributes (detecting minimum and maximum values) and for choosing the POIs (first representative of each class). Then, the second pass consists in computing the $(n-k) \times k$ similarities between the $n-k$ data and the k POIs. If d denotes the dimension of the data (i.e. the number of variables/attributes), then the overall complexity of our method is in $O((n-k) \times k \times d)$.

Table 1. Performance results on the very large data set

Name	# of data	# of attributes	# of classes	Displaying time (sec.)
GenFile_10000	10 000	10	5	0.719
GenFile_100000	100 000	10	5	4.470
GenFile_1000000	1 000 000	10	5	49.505
Forest CoverType	581 012	54	7	91.069

5 Conclusion and Perspectives

We have described in this paper a new visualization method which is inspired from the work on points of interest in the context of information retrieval. This method consists in mapping a space of similarities into a 2D representation. This is performed by positioning the data according to a selected set of points of interest. These points can be a subset of the data or hypothesis to be tested. The main characteristics of our method are the following: it may represent the $n \times n$ similarities while needing only a linear computation time, it is easy to learn and to interact with, it visually represents the relationships between the data. We have shown how standard benchmark databases can be displayed with this method, and how it can help the user to perform interactive clustering, and how it can deal with real world databases. Finally, we have shown how it may handle large databases with up to one million data on a standard computer.

One limit of our visualization are the ambiguities which occur when points have similar display coordinates, especially when large data sets are displayed. For this purpose we intend to use forces and springs methods in order to separate points that are too close. Another extension consists in using a 3D representation such in Lyberworld [17]. We have already tested this approach with up to 60000 data.

References

1. Wong, P.C., Bergeron, R.D.: 30 years of multidimensional multivariate visualization. In: Scientific Visualization — Overviews, Methodologies and Techniques, pp. 3–33. IEEE Computer Society Press, Los Alamitos, CA (1997)
2. Sudipto, G., Rajeev, R., Kyuseok, S.: CURE: an efficient clustering algorithm for large databases. In: Haas, L.M., Tiwary, A. (eds.) Proceedings ACM SIGMOD International Conference on Management of Data, Seattle, Washington, USA, pp. 73–84. ACM Press, New York (1998)
3. Tian, Z., Raghu, R., Miron, L.: Birch: An efficient data clustering method for very large databases. In: Jagadish, H.V., Mumick, I.S. (eds.) Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, Montreal, Quebec, Canada, June 4-6, 1996, pp. 103–114. ACM Press, New York (1996)
4. Costa, D.D., Venturini, G.: An interactive visualization environment for data exploration using points of interest. In: Li, X., Zaïane, O.R., Li, Z. (eds.) ADMA 2006. LNCS (LNAI), vol. 4093, pp. 416–423. Springer, Heidelberg (2006)
5. Korfhage, R.: To see, or not to see: Is that the query? In: Bookstein, A., Chiaramella, Y., Salton, G., Raghavan, V.V. (eds.) Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Chicago, Illinois, USA, October 13-16, 1991, pp. 134–141. ACM, New York 1991(special Issue of the SIGIR Forum)
6. Chernoff, H.: Using faces to represent points in *k*–dimensional spae graphically. Journal of the American Statistical Association 68, 361–368 (1973)
7. Inselberg, A.: The plane with parallel coordinates. The Visual Computer 1, 69–91 (1985)
8. Fua, Y.H., Ward, M.O., Rundensteiner, E.A.: Hierarchical parallel coordinates for exploration of large datasets. In: VISUALIZATION '99: Proceedings of the 10th IEEE Visualization 1999 Conference (VIS '99). IEEE Computer Society, Washington, DC, USA (1999)
9. Becker, R.A., Cleveland, W.S.: Brushing Scatterplots. Technometrics 29, 127–142 (1987). In: Cleveland, W.S., McGill, M.E. (eds.) Dynamic Graphics for Data Analysis. Chapman and Hall, New York 1988 (reprinted)

10. Keim, D.A., Kriegel, H.: VisDB: Database exploration using multidimensional visualization. In: Computer Graphics and Applications (1994)
11. Fekete, J., Plaisant, C.: Interactive information visualization of a million items proceedings of ieee symposium on information visualization (2002)
12. Jun Wang, B.Y., Gasser, L.: Classification visualization with shaded similarity matrices. Technical report, GSLIS University of Illinois at Urbana-Champaign (2002)
13. McCrickard, S., Kehoe, C.: Visualizing search results using sqwid. In: Proceedings of the Sixth International World Wide Web Conference (1997)
14. Hoffman, P., Grinstein, G., Pinkney, D.: Dimensional anchors: a graphic primitive for multidimensional multivariate information visualizations. In: NPIVM '99: Proceedings of the 1999 workshop on new paradigms in information visualization and manipulation in conjunction with the eighth ACM internation conference on Information and knowledge management, pp. 9–16. ACM Press, New York, NY, USA (1999)
15. Au, P., Carey, M., Sewraz, S., Guo, Y., Rüger, S.M.: New paradigms in information visualization. Research and Development in Information Retrieval, 307–309 (2000)
16. Blake, C., Merz, C.: UCI repository of machine learning databases (1998)
17. Hemmje, M., Kunkel, C., Willett, A.: Lyberworld visualization user interface supporting fulltext retrieval. In: SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, pp. 249–259. Springer, New York (1994)