

Layout Based Visualization Techniques for Multi Dimensional Data

Wim de Leeuw

Robert van Liere

Center for Mathematics and Computer Science, CWI

Amsterdam, the Netherlands

{wimc,robertl}@cwi.nl

October 27, 2000

Abstract

In this paper we present an overview for the interactive visualization of structural information in tabular multidimensional data. We first provide an overview of methods for layout of high dimensional data. Then we discuss a number of interactive visualization methods that can be used to present the structure of these high dimensional spaces. We discuss the use of the described methods in three applications.

Keywords: Information Visualization, Multi Dimensional Scaling, Kohonen Maps,

1 Introduction

A large class of data can be characterized by tables. This table is a matrix of attribute variables in one dimension and the outcome of specific cases in the other. Discovery and understanding of the structure in this type of data is a crucial part of science and business. Structure can take on many forms: clusters, regular patterns, outliers etc.

Two approaches in the analysis of data can be distinguished. In the first approach the data in the elements are aggregated into some new information in the other approach the data elements are laid out on a two or threedimensional space in some way. Of the first type are methods such as principal component analysis, k-means and hierarchical clustering algorithms. The second method will be described in more detail later as it is the focus of work in this paper. Both methods can serve as a basis for visualization.

Interactive visualization can be an aid in the process of discovery and understanding. The process of transforming the data tables into an image can be considered as a pipe line consisting of a number of steps; the so called visualization pipeline. By manipulation of the steps the user can interact with the system. For data tables the following steps can be discerned in the pipeline (see Figure 1): matrix generation, layout, and mapping. Manipulation is an important ingredient of visualization which can affect each of these steps.

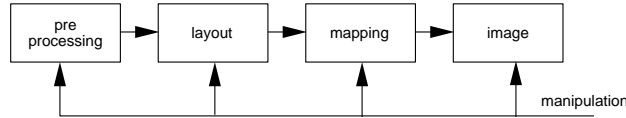


Figure 1: The steps for visualization table data

A distance matrix or adjacency matrix is generated by defining a metric by which the similarity or dissimilarity between cases in the table can be determined. Depending on the data type in the table, numeric, boolean or textual, many different metrics exist to calculate this difference. Using this distance matrix a layout of the samples is determined. This layout can be considered as a mapping from a high dimensional space to a lower dimensional space. Depending on the application the layout space can have one, two or three dimensions.

This layout forms the basis for the visualization of the data set. A distinction can be made between discrete mapping and continuous mapping. In case of discrete mapping each sample is represented by a separate icon. Visualization of attributes of the samples can be realized via color or shape mappings. In continuous mapping the underlying data samples are not explicitly represented but some form of aggregation is applied such that the overall properties of the set are reflected in the visualization.

Manipulation of the visualization allows the user to investigate different aspects of the data. In all but the smallest data sets it is impossible to present all information contained in the data automatically in a single image. Therefore the user should be able to manipulate the parameters in the visualization pipeline in a meaningful and understandable way to expose certain aspects of the information in the data.

Others have done work on frameworks for information visualization [1].

The paper is structured as follows.....

2 Methods

As described the visualization of sets of objects consists of three steps: layout, visualization and manipulation. This section presents an overview of methods which can be used to for each of these steps. The input of the pipeline is a collection data samples and some way to calculate distances between samples. Depending on the nature of the data (numerical, textual or boolean) many different distance metrics can be used.

2.1 Layout

In the layout stage a position is determined for each object in the set. These positions are chosen such that the structural properties of the set are conserved. This step can be regarded as a projection from the multi dimensional data space to a two dimensional or three dimensional Euclidean space. We assume a two dimensional projection space here, however all presented methods are in theory equally capable of producing three

dimensional layouts. The methods for layout can be classified based on the representation of the mapping function, for some methods the mapping function used for a particular set of data is not explicitly formulated. This means that if a new data element is added the method does not have the ability to position it in the existing layout. Other criteria to look at the method is dependence of ordering of the input or initial status of the algorithms. Some algorithms deliver the same result independent of ordering and initial position others can be influenced by the order.

In Multi Dimensional Scaling (MDS) [2], also known as Sammon's Mapping the idea is to approximate the original distance relations between samples in the layout as good as possible. Formally, if d_{ij} is the distance between sample x_i and x_j and r_i is the position of x_i the minimum of the equation

$$E = \sum_i \sum_{j>i} \frac{(d_{ij} - ||r_i - r_j||)^2}{d_{ij}} \quad (1)$$

is calculated. Usually some iterative process is used for the minimization.

By Self Organizing Maps (SOM) or Kohonen Maps[3] the layout of data samples is generated by a trained neural network. The the neural network consists of a two dimensional arrangement of nodes (neurons). This network is trained using a collection of training samples. During training of the network the reference vector associated which each node are iteratively improved such that a good distribution of the data samples over space is achieved. The trained network represents a non-linear mapping from n-dimensional space to the two dimensional node array. After training the response to an input samples produces localized signal in the network.

Generative Topographic Mapping (GTM) [4] is a technique in which an explicit topographic mapping function between the input data and the mapping space is found. The idea is to use a function, which maps a distribution in mapping space into the original data space. This is combined with a Gaussian noise model such that a sample mapped through the has a random position with a Gaussian distribution. A so called EM algorithm is used to find a combination of the 2D distribution function and mapping function which gives the optimal representation of the original data.

The methods described can be divided into two categories. The first category contains methods in which the position in the mapping space are explicitly taken into account in the layout process for example MDS. In the second category this is not the case for example Self Organizing maps.

2.2 Visualization

This section deals with the presentation of the generated layout to the user. How the information is presented to the user depends on:

- The structure of the underlying data.
- The size of the set
- The question which must be answered.
-

A global distinction can be made in discrete and continuous presentation. In the first case the individual samples of the data are presented as separate items. In the second case an aggregate of the data is presented in which the individual data samples might not be visible but the structure of the data set remains intact.

If the data set is small icons can be used to show the individual samples. Attribute data can be encoded in the color and shape of the icon. If the number of samples grows this method becomes less suitable due to cluttering.

A continuous representation is useful for the presentation of global structure in large data sets.

2.3 Manipulation

3 Applications

3.1 City distances

The idea is to construct a map based on a distance table. The distance table used in this case was a table found on an old road map of the Netherlands and presents the road distances between 39 towns in the Netherlands. Based on this data a map was constructed using the mass spring system the towns were modeled as nodes and the distance matrix was filled with the found distances. The road distances are larger or equal to the the straight distances.

The mass spring system was used to find the best fit of the towns on a plane. (See figure 2 left) This actually corresponds pretty well with the actual map of the Netherlands (See figure 2 right). The biggest aberration are the towns in the south west. An explanation can be found in the fact that a large detour is needed to reach these places from almost all other places due to the water around these places.

The visualization can also be considered as a distance map of the Netherlands it shows which places, although geometrically are close are far away from a driving perspective.

3.2 Image feature spaces

Image classification is a field in which the goal is to allow images to be retrieved from data repositories subject to a user defined query. To be able to process these queries the images are classified based on a collection of attributes, such as color, texture and object shape. The usefulness of the attributes for the query system is an important question for the developers of image retrieval systems. Our goal was the development of a system in which feature developers can experiment with features on wide varieties of image sets.

The input of our system is a single attribute vector consisting of the individual attributes. Because the individual features usually are vectors a user controllable scaling is applied to the weight of the attributes in the attribute vector. The distance matrix used for the layout of the images was defined by the Euclidean distance between the feature vectors. This way of presentation gives a global overview of the structure of the feature space as well as similarity relations among images. In addition, the method

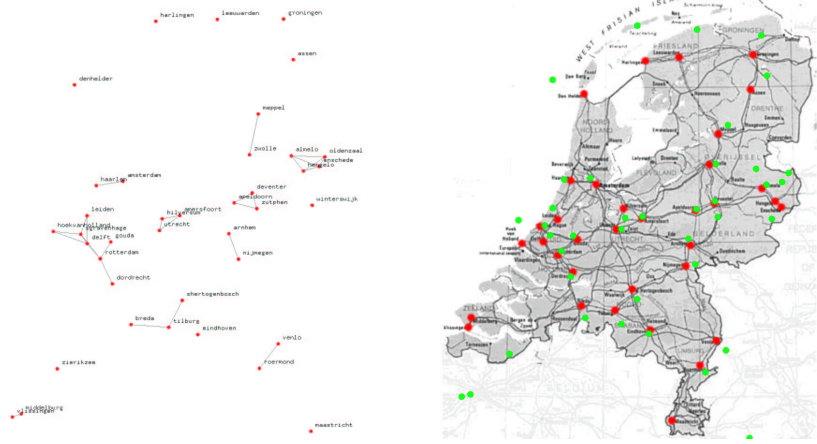


Figure 2: City distance visualization. Left: the found layout of towns based on the road distances. Right: the solution overlaid on a map of the Netherlands

allows a user to interactively scale each dimension of the feature space. In this way the user can explore the relation between features.

We applied our methods to a synthetic test set of 3276 images. The test set consisted of 36 groups of images with distinct hue values. Each group had 91 textures of varying frequency and orientation. For each image, 6 feature vectors were computed: a 1 four-dimensional Gabor feature vector for texture analysis and 5 distinct color-based features vectors. The color-based features vectors including a hue histogram, a hue histogram of the center region of the image, and 3 hue transition histograms. For transition histograms, the hue is first dithered to 16 bins; then the histogram of the 256 resulting combinations is recorded. As a pre-processing step, the images were segmented into 32, 128, and 256 tiles, and each tile was replaced by its dominant hue. The dimensionality of the feature space spanned by the 6 features vectors is 804.

Figure 3 shows a snapshot of the user interface. The left panel shows the graph view: an arrangement of the graph in the visualization space. Small dots are used to represent vertices. Grey lines represent edges between points with distances below the threshold distance T . Edges also provide additional feedback on the state and progression of the layout algorithm. For example, very long edges will indicate that the layout algorithm has not reached an equilibrium. Some selected vertices are annotated with a thumbnail image. The right panel shows the splat field, which is color encoded from white (low density values) to black (high density values). In Figure 3, the mass spring algorithm has reached an equilibrium. Users can drag vertices to other positions, after which the mass spring algorithm will compute a new equilibrium. Animation is used to display each step of the mass spring system evolving to an equilibrium. In this way, the user can study how an arrangement evolves towards another.

The graph provides a 2D view in which the images are displayed according to their mutual dissimilarities and similar images are clustered. A problem with the graph view

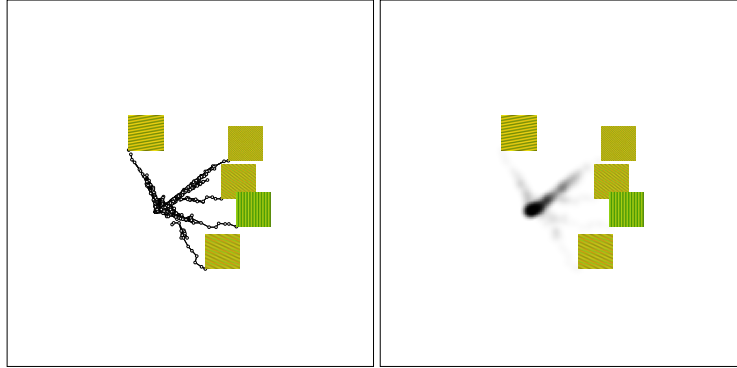


Figure 3: Two views of a graph arrangement for the test set. The left panel shows graph view with vertices and edges. The right panel shows the splat field. Some vertices are annotated with a thumbnail image.

is the potential cluttering, making it difficult to estimate density of vertices in dense regions. The splat field provides a 2D view of a continuous density field. Colors are used to show which areas have a high density of vertices. In this way, the user can see in a glance which images are similar.

3.3 Citations

The goal of this application is the analysis of the citation index of all IEEE Vis'9X papers. We show that clustering of citations leads to specific topics in visualization.

We have applied our method to the analysis of the IEEE Vis'9X citation index. The input data set are BibTeX entries of all papers in the proceedings of the IEEE Vis'9X conferences and all references to papers in this set from other papers in the set. The data set consists 599 BibTeX entries and 881 references. The graph represents papers as a vertices and references as edges.

The goals of the visualization was item to test the hypothesis that topics in visualization could be identified by only analyzing the density of the references. The motivation of this hypothesis is that papers in one topic often refer to other papers in the same topic.

The distance matrix used for the layout was the reference matrix. This matrix which has the dimension of the total number of papers in both directions and each element contains 'true' if a paper references the other.

Figure 4 shows the output of the spring mass algorithm. As can be seen, aside from the papers which are not referenced and do not reference papers, there is a single main graph which can not be partitioned without breaking edges. The number of elements in the graph is too large and cluttered to get insight into the structure of the graph.

Figure 4 shows a slightly zoomed in 2D rendition of the same splat field. One can clearly see various clusters of papers. For example, the papers in the large dark region in the middle of the image deal with flow visualization. The (smaller) region

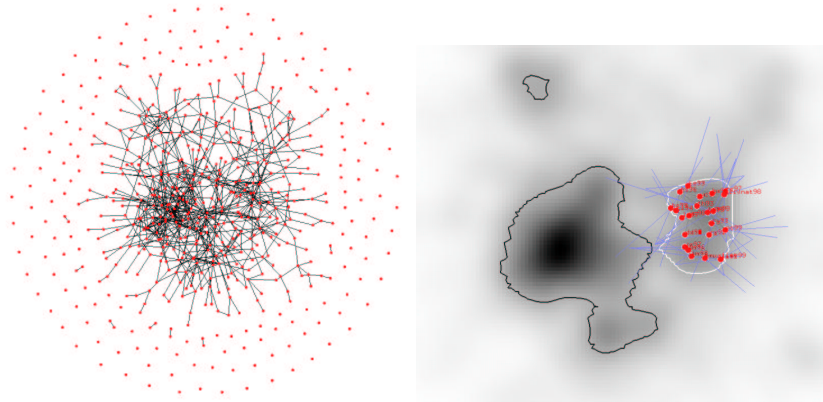


Figure 4: Left: All papers published in IEEE Vis'9X conferences, represented as discs and references between the papers represented as lines. Right: Interacting with the citation index. The influence of a group of papers is drawn with yellow (incoming) and blue (outgoing) references. Papers in the region selected by the highlighted contour on the right are shown as discs.

below are papers describing visualization systems. The region on the right are volume visualization papers. Discrete discs (red dots) and lines (yellow lines for incoming references and blue lines for outgoing references) are used for representing vertices and edges are also drawn as annotation. The region at the top left contain information visualization papers. The distance to the other peaks in the field clearly illustrates the distinction between information visualization and other data visualization topics.

Figure 4 also illustrates some interaction techniques. The contour line around the right region is used as a selection criterion. In this way all papers in an area, in this case volume visualization, can be selected. Also, the influence of a paper is shown by drawing the edges representing references to the selected papers. The user can also pick individual papers and show all information related to that individual paper.

4 Conclusion

References

- [1] M. Kreuseler N. López H. Schumann. A scalable framework for information visualization. In *Proceedings IEEE Symposium on Information Visualization 2000*, pages 27–36, 2000.
- [2] S.S. Schiffman M.L. Reynolds F.W. Young. *Introduction to Multidimensional Scaling, Theory, Methods, and Applications*. Academic Press, 1981.

- [3] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag Berlin Heidelberg New York, 1995.
- [4] C.M. Bishop M Svensén C.K.I. Williams. GTM: A principled alternative to the self-organizing map. *Advances in Neural Information Processing Systems*, 9:354–363, 1997.