Lecture Notes in Computer Science Vol. 6113, pp. 445-452, 2010.

Triangular Visualization

Tomasz Maszczyk and Włodzisław Duch

Department of Informatics, Nicolaus Copernicus University, Toruń, Poland
tmaszczyk@is.umk.pl;Google:W.Duch
http://www.is.umk.pl

Abstract. The *TriVis* algorithm for visualization of multidimensional data proximities in two dimensions is presented. The algorithm preserves maximum number of exact distances, has simple interpretation, and unlike multidimensional scaling (MDS) does not require costly minimization. It may also provide an excellent starting point significantly reducing the number of required iterations in MDS.

1 Introduction

Almost all datasets in real applications have many input variables that may be inter-related in subtle ways. Such datasets can be analyzed using conventional methods based on statistic, providing a numerical indication of the contribution of each feature to a specific category. Frequently exploratory data analysis is more informative when visual analysis and pattern recognition is done rather than direct analysis of numerical data. The visual interpretation of datasets is limited by human perception to two or three-dimensions. Projection methods and non-linear mapping methods that show interesting aspects of multidimensional data are therefore highly desirable [1].

Methods that represent topographical proximity of data are of great importance. They are usually variants of multidimensional scaling (MDS) techniques [2]. Here a simpler and more effective approach called *TriVis* is introduced, based on the growing structures of triangles that preserve exactly as many distances as possible. Mappings obtained in this way are easy to understand, and may also be used for MDS initialization. Other methods do not seem to find significantly better mappings. In the next section a few linear and non-linear visualization methods are described, and *TriVis* visualization based on sequential construction of triangles is introduced. Illustrative examples for several datasets are presented in section 3. Conclusions are given in the final section.

2 Visualization algorithms

First a short description of two most commonly used methods, principal component analysis (PCA) [1] and multidimensional scaling (MDS) [2], is given. After this, our *TriViz* approach is presented. Comparison of these 3 methods is presented in the next section.

2.1 Principal component analysis

PCA is a linear projection method that finds orthogonal combinations of input features $\mathbf{X} = \{x_1, x_2, ..., x_N\}$ preserving most variation in the data. Principal component directions $\mathbf{P}_i$ result from diagonalization of data covariance matrix [3]. They can be ordered from most to the least important, according to the size of the corresponding eigenvalues. Directions with maximum variability of data points guarantee minimal loss of information when position of points are recreated from their low-dimensional projections. Visualization can be done taking the first two principal components and projecting the data into the space defined by these components, $y_{ij} = \mathbf{P}_i \cdot \mathbf{X}_j$. PCA complexity is dominated by covariance matrix diagonalization, for two highest eigenvalues it is at least $O(N^2)$. For some data distributions this algorithm shows informative structures.

Kernelized version of the standard PCA may be easily formulated [4], finding directions of maximum variance for vectors mapped to an extended space. This space is not constructed in an explicit way, the only condition is that the kernel mapping $K(\mathbf{X}, \mathbf{X}')$ of the original vectors should be a scalar product $\Phi(\mathbf{X}) \cdot \Phi(\mathbf{X}')$ in an extended space $\Phi(\mathbf{X})$. This enables interesting visualization of data, although interpretation is rather difficult.

2.2 Multidimensional scaling

MDS is perhaps the most popular non-linear technique of proximity visualization. The main idea how to decrease dimensionality while preserving original distances in high-dimensional space has been rediscovered several times [5–7] and is done either by minimization of specific cost functions [2] or by solving a system of cubic equations [7]. MDS methods need only similarities between objects as inputs, so explicit vector representation of objects is not necessary. Qualitative information about pairwise similarities is sufficient for non-metric MDS, but here only quantitative evaluation of similarity calculated by numerical functions is used. MDS methods differ by their cost functions, optimization algorithms, the type of similarity functions and the use of feature weighting. There are many measures of topographical distortions due to the reduction of dimensionality, most of them weighted variants of the simple stress function [2]:

$$S(\mathbf{d}) = \sum_{i>j}^{n} W_{ij} \left(D_{ij} - d_{ij}\right)^2 \tag{1}$$

where d_{ij} are distances (dissimilarities) in the target (low-dimensional) space, and D_{ij} are distances in the input space. Weights $W_{ij} = 1$ for simple stress function, or to reduce effect of large distances $W_{ij} = 1/D_{ij}$ or $W_{ij} = 1/D_{ij}^2$ are used. The sum runs over all pairs of objects and thus contributes $O(n^2)$ terms. In the k-dimensional target space there are $kn - k$ parameters for minimization. For visualization purposes the dimension of the target space is $k = 1, 2$ or 3 (usually $k = 2$).

MDS cost functions are not easy to minimize, with multiple local minima representing different mappings. Initial configuration is either selected randomly or is based on PCA projection. Dissimilar objects are represented by points that are far apart, and similar objects are represented by points that are close, showing clusters in the data. Orientation of axes in the MDS mapping is arbitrary, and the values of coordinates do not have any meaning, as only relative distances are preserved. Kohonen networks [8] are also a popular tool combining clusterization with visualization, but they do not minimize directly any measure of topographical distortion for visualization, therefore their visualization is not as good as that provided by MDS.

2.3 Triangular visualization

TriVis algorithm creates representation of data points in two-dimensional space that exactly preserves as many distances between points as it is possible. Distances between any 3 vectors forming a triangle may always be correctly reproduced; a new point is iteratively added relatively to one side of existing triangle, forming a new triangle that exactly preserves two original distances. There are many possibilities of adding such points in relation to the existing triangle sides. To preserve the overall cluster structure 3 furthest points are selected for the start (an alternative is to use centers of 3 clusters), and the new point is choosen to minimize the MDS stress function $S(\mathbf{d}) = \sum_{i>j}^{n} \left(D_{ij} - d_{ij}\right)^2$. This gives mapping that preserves exactly $2n-3$ out of $n(n-1)/2$ original distances, minimizing overall stress.

Algorithm 1

```
Find three farthest vectors and mark them (preserving original distances) as points of the initial triangle.
Mark segments (pairs of points) forming triangle sides as available.
for $i = 1$ to $n - 3$ do
    Find the segment AB for which vector $\mathbf{X}_i$ added as the point C=C($\mathbf{X}_i$) forms a triangle ABC preserving two original distances —AC— and —BC—, and gives the smallest increase of the stress $S_i = \sum_{j=1}^{m} \left(D_{ij} - d_{ij}\right)^2$.
    Delete the AB segment from the list of available segments, and add to it segments AC and BC.
end for
```

Complexity of this algorithm grows like $O(n^2)$, but MDS has to perform minimization over positions of these points while *TriVis* simply calculates positions. To speed up visualization process this algorithm could be applied first to K vectors selected as the nearest points to the centers of K clusters (for example using the K-means or dendrogram clusterization). Plotting these points should preserve the overall structure of the data, and applying *TriVis* to points within each cluster decomposes the problem into K smaller $O(n_k^2)$ problems. For large number of vectors the use of jitter technique may be sufficient, plotting the vectors that belong to one specific cluster near the center of this cluster, with dispersion equal to the mean distance between these points and the center.

To measure what can be gained by full MDS minimization *TriVis* mapping should be used as a starting configuration for MDS. This should provide much lower stress at the beginning reducing the number of iterations.

3 Illustrative examples

The usefulness of the *TriVis* sequential visualization method has been evaluated on four datasets downloaded from the UCI Machine Learning Repository [9] and from [10]. A summary of these datasets is presented in Tab. 1; their short description follows:

1. **Iris** the most popular dataset, it contains 3 classes of 50 instances each, where each class refers to a type of the Iris flowers.
2. **Heart** disease dataset consists of 270 samples, each described by 13 attributes, 150 cases belongs to group "absence" and 120 to "presence of heart disease".
3. **Wine** wine data are the results of a chemical analysis of wines, grown in the same region in Italy, but derived from three different cultivars. 13 features characterizing each wine are provided, the data contains 178 examples.
4. **Leukemia:** microarray gene expressions for two types of leukemia (ALL and AML), with a total of 47 ALL and 25 AML samples measured with 7129 probes [10]. Visualization is based on 100 best features from simple feature ranking using FDA index [1].

For each dataset two-dimensional mappings have been created using PCA, *TriVis*, MDS starting from random configuration and MDS starting from *TriVis* configuration (Figs. 1-4). Visualization is sensitive to feature selection and weighting and frequently linear projections discovered through supervised learning may be more informative [11, 12]. Since our goal here is not the best visualization but rather comparison of *TriVis* algorithm with PCA and MDS methods all features have been used.

Title	#Features	#Samples	#Samples per class			Source
Iris	4	150	50 "Setosa"	50 "Virginica"	50 "Versicolor"	[9]
Heart	13	303	164 "absence"	139 "presence"		[9]
Wine	13	178	59 "C_1"	71 "C_2"	48 "C_3"	[9]
Leukemia	100	72	47 "ALL"	25 "AML"		[10]

Table 1. Summary of datasets used for illustrations

Mappings of both Iris and Cleveland Heart datasets are rather similar for all 4 techniques (selecting only relevant features will show a better separation between classes); PCA shows a bit more overlap and MDS optimization of *TriVis* configuration does not provide more information than the initial configuration.

Wine dataset does not map well with PCA and different classes are somehow better separated using MDS with *TriVis* initialization. This is a good example showing that using *TriVis* configuration as the start for MDS leads to faster and better convergence.

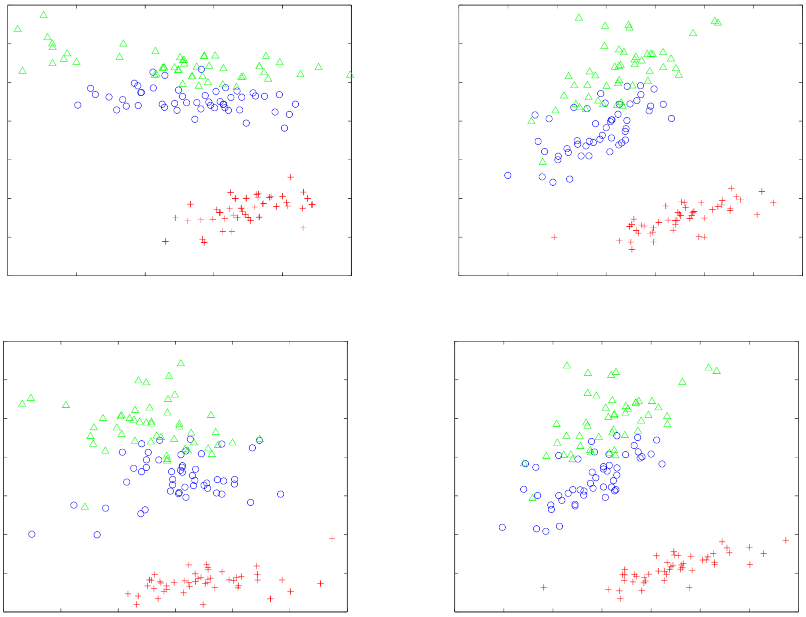

Fig. 1. Iris dataset, top row: PCA and *TriVis*, bottom row: typical (randomly initialized) MDS and MDS initialized by *TriVis*.

Leukemia shows good separation using *TriVis* projection (Fig. 4), providing a bit more interesting projection than other methods.

To show the influence of *TriVis* initialization on MDS comparison of the convergence starting from random, PCA and *TriVis* configurations is presented in Fig. 5. *TriVis* initialization clearly works in the best way leading to a convergence in a few iterations and achieving the lowest stress values. This type of initialization may prevent MDS from getting stuck in poor local minimum.

4 Conclusions

In exploratory data analysis PCA and MDS are the most popular methods for data visualization. Visualization based on proximity helps to understand the structure of the data, to see the outliers and to place interesting cases in their most similar context, it may also help to understand what black box classifiers really do [13, 14]. In safety-critical areas visual interpretation may be crucial for acceptance of proposed methods of data analysis.

The *TriVis* algorithm presented in this paper has several advantages: it enables visualization of proximities in two dimensions, preserves maximum number of exact distances reducing distortions of others, has simple interpretation, allows for simple assesment of various similarity functions and feature selection and weighting techniques, it may unfold various manifolds [15] (hypersurfaces embedded in high-dimensional spaces). For large datasets it may be coupled with

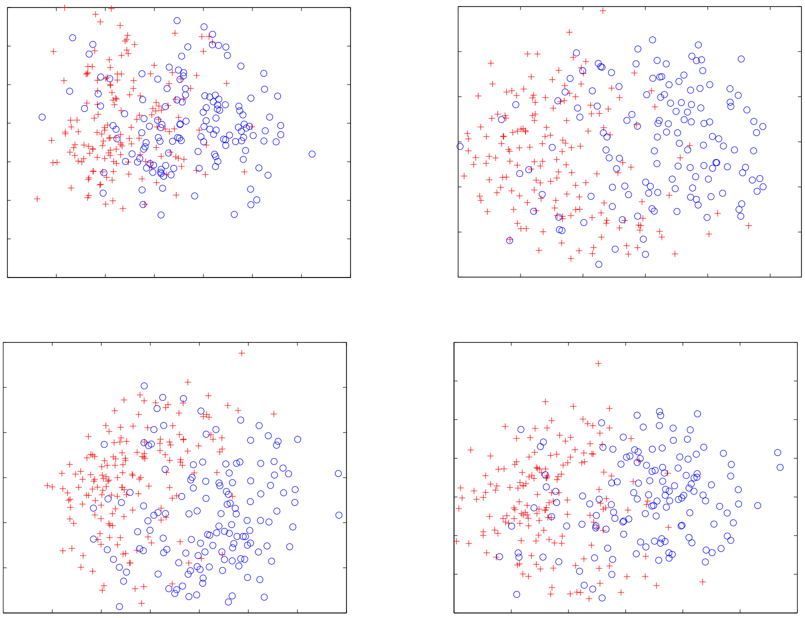

Fig. 2. Cleveland Heart dataset, top row: PCA and *TriVis*, bottom row: typical MDS and MDS initialized by *TriVis*.

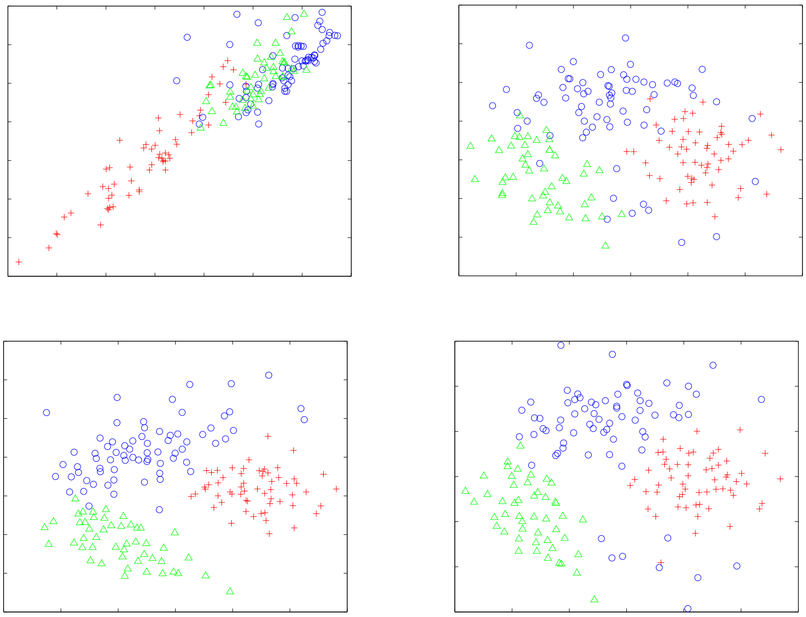

Fig. 3. Wine dataset, top row: PCA and *TriVis*, bottom row: typical MDS and MDS initialized by *TriVis*.

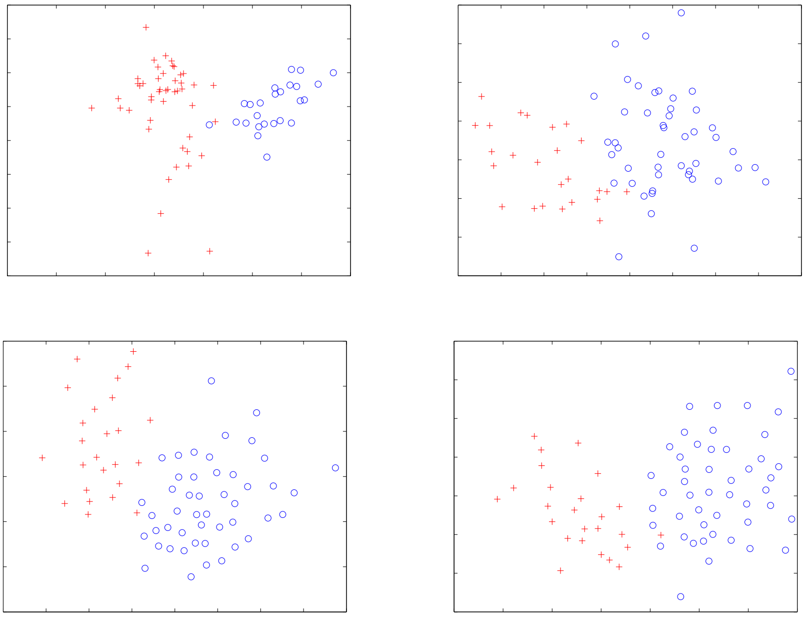

Fig. 4. Leukemia dataset, top row: PCA and *TriVis*, bottom row: typical MDS and MDS initialized by *TriVis*.

hierarchical dendrogram clusterization methods to represent with high accuracy relations between clusters. PCA does not preserve proximity information, while MDS is much more costly and does not seem to have advantages over *TriVis*. If MDS visualization is desired *TriVis* gives an excellent starting point significantly reducing the number of required iterations.

References

1. Webb, A.: Statistical Pattern Recognition. J. Wiley & Sons (2002)
2. Cox, T., Cox, M.: Multidimensional Scaling, 2nd Ed. Chapman and Hall (2001)
3. Jolliffe, I.: Principal Component Analysis. Springer-Verlag, Berlin (1986)
4. Schölkopf, B., Smola, A.: Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA (2001)
5. Torgerson, W.: Multidimensional scaling. I. Theory and method. Psychometrika **17** (1952) 401–419
6. Sammon, J.: A nonlinear mapping for data structure analysis. IEEE Transactions on Computers **C18** (1969) 401–409
7. Duch, W.: Quantitative measures for the self-organized topographical mapping. Open Systems and Information Dynamics **2** (1995) 295–302
8. Kohonen, T.: Self-organizing maps. Springer-Verlag, Heidelberg Berlin (1995)
9. Asuncion, A., Newman, D.: UCI machine learning repository. http://www.ics.uci.edu/~mlearn/MLRepository.html (2009)
10. Golub, T.: Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. Science **286** (1999) 531–537

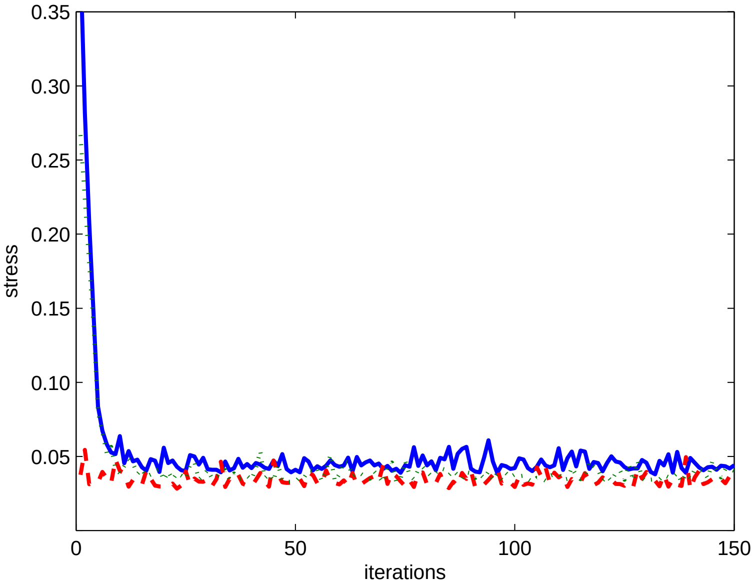

Fig. 5. Comparison of 3 types of MDS initialization (Wine dataset): solid blue line - random, dotted green line - PCA, dashed red - *TriVis*.

11. Maszczyk, T., Duch, W.: Support vector machines for visualization and dimensionality reduction. Lecture Notes in Computer Science **5163** (2008) 346–356
12. Maszczyk T, Grochowski M, D.W.: Discovering Data Structures using Meta-learning, Visualization and Constructive Neural Networks. In: Studies in Computational Intelligence Vol. 262. Springer (2010) (in print)
13. Duch, W.: Visualization of hidden node activity in neural networks: I. visualization methods. In Rutkowski, L., Siekemann, J., Tadeusiewicz, R., Zadeh, L., eds.: Lecture Notes in Artificial Intelligence. Volume 3070. Physica Verlag, Springer, Berlin, Heidelberg, New York (2004) 38–43
14. Duch, W.: Coloring black boxes: visualization of neural network decisions. In: Int. Joint Conf. on Neural Networks, Portland, Oregon. Volume I. IEEE Press (2003) 1735–1740
15. Tenenbaum, J., de Silva, V., Langford., J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290** (2000) 2319–2323