Visualization of large data sets using MDS combined with LVQ.

Antoine Naud and Włodzisław Duch

Department of Informatics, Nicholas Copernicus University,
Grudziądzka 5, 87-100 Toruń, Poland.
www.phys.uni.torun.pl/kmk

Abstract. A common task in data mining is the visualization of multivariate objects using various methods, allowing human observers to perceive subtle inter-relations in the dataset. Multidimensional scaling (MDS) is a well known technique used for this purpose, but it due to its computational complexity there are limitations on the number of objects that can be displayed. Combining MDS with a clustering method as Learning Vector Quantization allows to obtain displays of large databases that preserve both high accuracy of clustering methods and good visualization properties.

1 Introduction

Visualization helps in data exploration by providing the user an idea about clusters in data and their mutual relations, shows outliers and allows for rough classification of data. A very popular way to visualize data is based on self-organizing topographic feature maps (SOM) [1], known also as the Kohonen networks. SOM is so popular because the method may be applied to very large databases and provides clusterization as well as visualization at the same time. Unfortunately SOM networks are among the worst classifiers, as found in empirical studies [2]. There are many better clustering methods than SOM, for example methods based on dendrograms. We have compared SOM visualization using simplexes (up to dimension 20), hypercubes and points on the hypersphere, with a method that preserves correct topographical relations among multidimensional objects [3,4]. These results clearly show that minimization of a quantitative measures for the distortions of original data topography, i.e. the multidimensional scaling technique [5,6], generates maps that preserve both local and global features in a better way than SOM does.

Unfortunately MDS requires difficult minimization of cost function that is based on distances among the represented multidimensional objects. For N objects $N(N-1)/2$ distances enter the cost function, and if maps are 2-dimensional they have $2N-3$ adaptive parameters (position (x,y) of an arbitrary point and the rotation angle, or y coordinate of a second point, may be fixed).

The MDS-based interactive software developed by one of us [7] performs a mapping of multivariate data from a high-dimensional space (D-dimensional space, $D \gg 3$) to data points in a lower d-dimensional space ($d \ll D$). Usually $d = 2$ in order to allow visualization by scatterplots, but higher d values may be useful for dimensionality reduction. The MDS dimensionality reduction is such that similar (or close, in the sense of some distance measure in the D-dimensional space) multivariate objects are mapped on representative points close to each other in the d-dimensional representation space. The mapping should preserve topography of data vectors. Linear projections are often not able to preserve topography; such mappings have to be non-linear. Topography preservation may be unreachable in the whole feature space if the analyzed multivariate data is intrinsically high dimensional and cannot be imbedded in a lower dimensional space without much distortion.

Combination of this algorithm with clusterization methods have several advantages. First, hierachical clusterization methods or any other clusterization may be used. Second, information about the cluster centers may be displayed on the map and combined with MDS information, giving more details than dendrograms typically used to display clusters. Third, multi-level mapping, starting from large clusters and creating smaller ones, may help to find a better MDS solution since initially smaller number of points are mapped. Fourth, it may be applied to large datasets, and thus be competitive to SOM, providing good clusterization and visualization.

The process that combines clusterization with dimensionality reduction is described below. We have used Learning Vector Quantization (LVQ) for clusterization [1] but any other algorithm can be used. Since it is not clear how accurate are the maps created in this way, we have compared them to the original, fully optimized maps in the third section. A short discussion concludes this paper.

2 Combining MDS with LVQ

A direct visualization by MDS of large datasets may be unpracticable due to the limitation on the number of simultaneously mapped objects. Various approaches have been proposed to tackle this problem. Our approach is similar to the so-called *frame method* [8]. It proceeds as follows: First build (by some heuristics) a smaller dataset (called the basis B), designed as a first approximation of the original large dataset. Then map the Basis using standard MDS. Finally add to the mapped Basis all the remaining cases from the original dataset. Let us denote D the dataset to be visualized and N_D the number of cases in this dataset. Let B denote the basis and N_B be the size of the basis, $N_t = N_m + N_f$ is the total number of points considered during the mapping, $F = \{P_i, i = 1, N_f\}$ the set of known data points and by $M = \{P_i, i = 1, N_m\}$ the set of new data points. In Chang and Lee [8] the basis B is build by selecting a subset of points from D, as a result the final display highly depends on which points are selected. We prefer to use a clustering technique on the dataset in order to obtain N_B cluster centers that more uniformly represent the data. For the final step, we use *relative MDS* [13], where each new data is individually added to the mapped Basis using an algorithm that is much less time consuming than the full Stress minimization. The general process proposed for mapping large datasets is shown in figure 1.

Relative MDS mapping differs from standard MDS in this respect that during minimization of the topography distortion measure (called “Stress") only the points from M set are allowed to move while the points from F set are kept fixed. This is

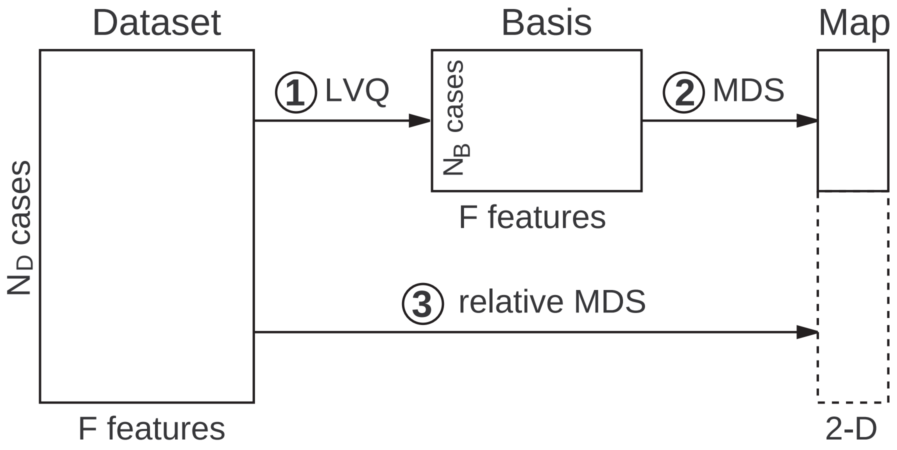


Fig. 1. Process used for the visualization of large datasets: ① – cluster the data using LVQ to form the so-called Basis, ② – Map the Basis using standard metric MDS, ③ – Add the remaining Dataset cases to the mapped Basis using relative MDS.

achieved by modifying the Stress function so that it sums only over the distances that change during mapping, i.e. the distances between the added and the basis points, and the added points interpoint distances. The original Stress function:

$$S(\mathbf{x}) = \sum_{ij}^{N_D} w_{ij} \cdot \left(\hat{d}_{ij} - d_{ij}\right)^2 \tag{1}$$

is redefined as:

$$S_r(\mathbf{x}) = \sum_{i=1}^{N_D} \sum_{j=1}^{N_B} w_{ij} \cdot \left(\hat{d}_{ij} - d_{ij}\right)^2 \tag{2}$$

In relative mapping, the order in which the data are added does not influence the final result because each case is added independently from the other cases. Relative mapping can also be seen as an alternative to methods designed for the purpose of giving generalization capability to Sammon mapping, such as the ANN Sammon mapping [9], Neuroscale [10] or incremental scaling [11].

3 Visualization of the `Satimage` dataset

It is interesting to see where a new data points “falls” among known cases, discover the class of its neighbors (classified or labeled), and to get an insight on how a classifier would evaluate this new data. `Satimage` dataset comes from the Landsat MSS imagery, it consists of four digital images of the same scene in different spectral bands. This dataset is often used to perform comparison of classifiers, and it is part of the UCI repository [12]. The dataset is made of $N_D = 4435$ cases with 36 numerical features in the range 0 to 255. Each case stands for one pixel in the image, the pixels are attributed to 6 classes standing for different soil types photographed.

The visualization of the N_D cases has been performed by clustering the data into N_B cluster centers. As we are interested in determining the optimal size for the basis, we performed a series of mappings with different values for N_B and compared the resulting displays visually and by means of the final Stress. The following values were chosen: $N_B = 100, 300, 500, 700, 1000, 1200, 1300$, this last value is the maximum reachable with 256 MB of RAM. The final Stress values obtained are presented in **Fig.** 2. As can be expected, the final Stress value decreases when the Basis size increases. The curve should converge towards the lowest Stress obtained when mapping all N_D cases in one MDS run.

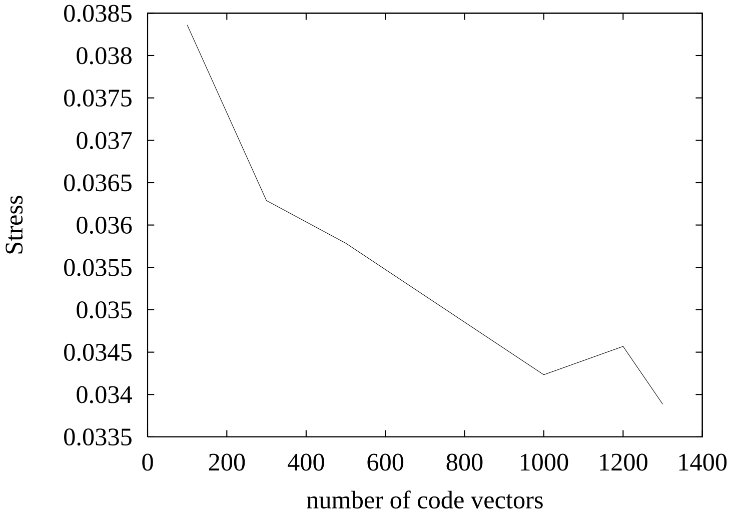


Fig. 2. Final Stress values obtained after relative mapping of the dataset cases added to a Basis of 100, 300, 500, 1000, 1200 and 1300 LVQ code vectors.

The whole mapped `Satimage` dataset is shown in figure 3. It can be seen that a Basis size of $N_B = 500$ cluster centers is enough to get quite good display because the obtained configuration does not change much for higher sizes. This optimal size should also be observed on the curve of **Fig.** 2, where the “speed" of Stress decrease slows down.

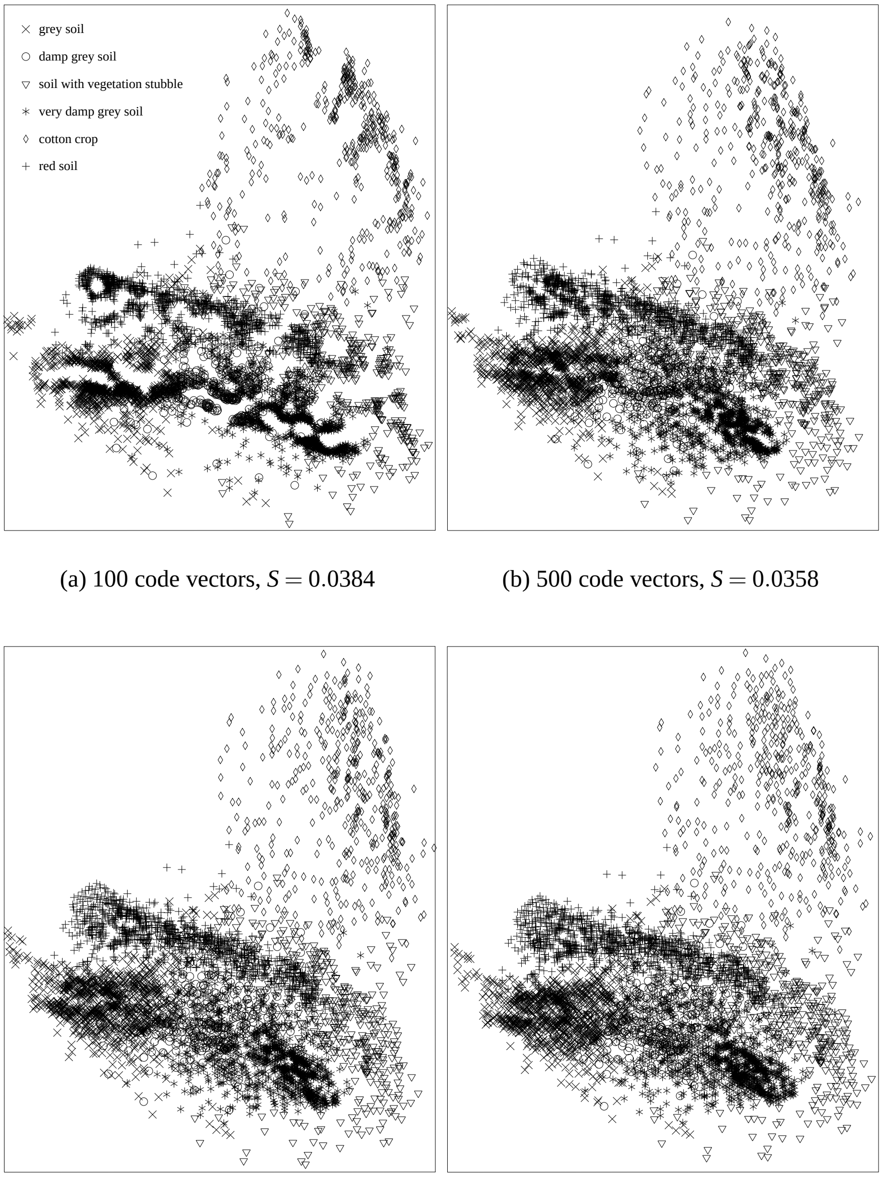
4 Discussion

The main advantage of MDS dimensionality reduction is that the topographical distortions induced can evaluated by the value of the Stress function reached during minimization. In this paper we have shown that the main disadvantage of MDS, its computational complexity, may be removed by mapping smaller number of cluster centers and adding the remining points using relative maping. Interactive zooming on interesting areas of the input space [13] allows for exploration of the neighborhood of the case under inspection. Since there may be some uncertainty in the input one may generate a number (for example 100) of vectors drawn from a Gaussian distribution centered at this vector. The class of these additional points is determined using neural or other classification systems and corresponding points added (using relative mapping) to the map. Visual inspection of such map allows estimating the probability of misclassification, proportional to the number of points from alternative classes.

Acknowledgments: Support by the Polish Committee for Scientific Research, grant 8 T11C 006 19, is gratefully acknowledged.

References

1. Kohonen T. (1995) *Self-Organizing Maps.* Springer-Verlag, Heidelberg Berlin
2. D. Michie, D. J. Spiegelhalter, and C. C. Taylor, editors. *Machine Learning, Neural and Statistical Classification.* Ellis Horwood, New York, 1994.
3. Duch W, Naud A. (1996) Multidimensional scaling and Kohonen's self-organizing maps. In *Proc. 2nd Conf. "eural Networks and their Applications", Szczyrk, Poland*, Vol. I, pp. 138–143
4. Duch W, Naud A. (1996) On global self-organizing maps. In *Proc. 4th European Symposium on Artificial Neural Networks, Brugges*, pp. 91–96
5. Kruskal J. B. (1964) Non metric multidimensional scaling : a numerical method. *Psychometrika* **29**, 115–129
6. Sammon J. W. (1969) A nonlinear mapping for data analysis. *IEEE Transactions on Computers* **5**, 401–409
7. Naud A. (2001) Neural and statistical methods for the visualization of multidimensional data. PhD Thesis, Dept. of Informatics, Nicholas Copernicus University, Poland. http://www.phys.uni.torun.pl/kmk/publications.html
8. Chang C. L, Lee R. C. T. (1973) A heuristic relaxation method for nonlinear mapping in cluster analysis. *IEEE Transactions on Systems, Man, and Cybernetics* **3**, 197–200
9. Kraaijveld M. A, Mao J, Jain A. K. (1995) A nonlinear projection method based on Kohonen's topology preserving maps. *IEEE Transactions on Neural Networks*, **6**, 548–559.
10. Tipping M. E. (1996) *Topographic mappings and feed-forward neural networks*. PhD thesis, Aston University, Birmingham, UK
11. Basalaj W. (1999) Incremental multidimensional scaling method for database visualization. In *Proceedings of Visual data Exploration and Analysis VI, SPIE*, Vol. **3643**, 149-158
12. Blake, C.L, Merz, C.J. (1998). UCI Repository of machine learning databases http://www.ics.uci.edu/ mlearn/MLRepository.html. Irvine, CA: University of California, Department of Information and Computer Science.
13. Naud A, Duch W. (2000) Interactive data exploration using MDS mapping. In: 5th Conf. on Neural Networks and Soft Computing, Zakopane, Poland, pp. 255-260



(a) 100 code vectors, $S = 0.0384$

(b) 500 code vectors, $S = 0.0358$

(c) 1000 code vectors, $S = 0.0342$

(d) 1300 code vectors, $S = 0.0339$

Fig. 3. Visualization of `Satimage` dataset using different numbers of code vectors for LVQ clustering, to which the whole dataset has been added by relative mapping.