Simplexes, Multi-Dimensional Scaling and Self-Organized Mapping

Włodzisław Duch and Antoine Naud
Department of Computer Methods, Nicholas Copernicus University
ul. Grudziądzka 5, 87–100 Toruń, Poland.
e-mail: duch @ phys.uni.torun.pl, naud @ phys.uni.torun.pl
WWW: http://www.phys.uni.torun.pl/kmk

Abstract— **The self-organizing map (SOM) of Kohonen is one of the most successful models of unsupervised learning. Its popularity is partially due to the visualization (topography preservation) of relations among clusters in high-dimensional input space. SOM learns slowly, especially in the initial phase, and the preservation of topography by SOM maps is not based on any quantitative criteria. We have obtained the best possible two-dimensional representation of simplexes in spaces of up to 20 dimensions, minimizing the error function measuring the unavoidable distortion of the original input space topography. This two-dimensional representation is used to select neurons during initialization of the SOM network. After such initialization in the learning phase a small radius of the neighborhood function is sufficient to obtain quick convergence with minimal topological distortions.**

1 Introduction

The self-organizing mapping (SOM) algorithm [1] is usually presented as a particular type of artificial neural network. The network is first trained on the high-dimensional input vectors until the weight vectors of the array of neurons in the output layer tend to approximate the probability density function of the high-dimensional data. This iterative learning process is unsupervised, or self-organizing, defining the mapping from the high-dimensional input data space to the low-dimensional target space. The mapping is well ordered if the topographical relations of the input vectors are preserved in the topographical relations in the target space.

Kohonen worked on pattern recognition (especially speech recognition) when he developed the SOM algorithm, and his intentions were the following [2]: "The SOM ... is a clustering, visualization, and abstraction method. Anybody wishing to implement decision and classification processes should use LVQ (Learning Vector Quantization) instead of SOM." Despite this warning SOM is used quite frequently as a network for classification (cf. the book [3] or full bibliography on SOM stored in the ftp archive `cochlea.hut.fi` in the `/pub/ref/` catalog, file `references.bib.Z`). Undoubtedly the most interesting aspect of SOM is the topography preservation or the ability to visualize the high-dimensional data. Unfortunately the method does not provide any measures of the quality of this visualization.

The multidimensional scaling (MDS) technique is a statistical clusterization method almost unknown outside of the mathematical psychology field. It has foundations in the work of Torgerson [4] and in the Coombs theory of data [5]. Computer programs and applications of MDS have been developed, among others, by Kruskal [6] at the Bell Laboratories, by Lingoes, Roskam and Borg [7] in Ann Arbor, and by Shepard [8] in Palo Alto [9]. MDS was used by experts in mathematical psychology who wanted to obtain a lower-dimensional representation of psychological data. These data are related to perception (perceived nearness of objects, preferences, feature intensity or affinity) and are obtained by subjective evaluation of similarities or dissimilarities between different items, characterizing a small part of human psychological spaces. MDS techniques were developed to provide a two or three-dimensional image of the observed data, reducing their complexity and allowing their analysis by a human expert. MDS was rediscovered under the name of nonlinear mapping by Sammon [10] and as quantitative measure for topographical distortions in SOM by Duch [11].

Since MDS guarantees the best preservation of topography it is interesting to compare the results of this procedure with maps obtained by SOM in a few simple cases. We have performed such a comparison [12] using as an example the points on a surface of a sphere and the corners of hypercubes. SOM maps are far from optimal, leading to significantly larger values of the measures of topographical distortion. Unfortunately MDS is computationally even more intensive than SOM, requiring for n input data vectors global minimization of a function of $n(n-1)/2$ variables. It is definitely not suited for large amount of data or for on-line learning.

There are several options leading to optimal topographical mappings. First we will give describe the multidimensional scaling and the self-organizing map algorithms, then we will present simplexes in 3-20 dimensional spaces mapped by the MDS procedure onto a two-dimensional target space. In the fifth section one of the simplest approaches to create topographical mappings using simplex results for initialization of SOM networks is introduced and in the last section other ways of creating good topographical maps are discussed.

2 Multidimensional scaling

MDS techniques emerged from the need to visualize in a two- or three-dimensional space high dimensional objects described by some measure of their similarities or dissimilarities. Although the problem may be defined in a very general way in non-metric spaces here we are interested only in the simpler case when similarity measures are distances defined by some metric in the input space. The problem is to find the coordinates of points representing the multidimensional data vectors in the two or three-dimensional space in such a manner that the distances among points in the low-dimensional target space are as close as possible to the original distances. MDS takes as input a symmetric matrix of the distances (similarities or dissimilarities) between objects. If there are n input vectors there are $n(n-1)/2$ distances between these vectors. MDS does not deal directly with values of vector components in the input space, as SOM does. SOM algorithm needs $n \times N$ input values, where N is the dimension of the input vectors. If the number of input vectors $n > 2N+1$ Kohonen map uses less information than MDS, therefore SOM is better suited for large number of input data.

Let n be the number of input vectors in the high-dimensional space, $X_1, X_2, ..., X_n$, and let R_{ij} be the observed similarities between vectors X_i, equivalent to distances $R_{ij} = \|X_i - X_j\|$ in metric spaces. Let Y_i be the low dimensional target space point representing the input vector X_i and let r_{ij} be the distance between Y_i and Y_j. We have to place the points $\{Y_i, i = 1, ..., n\}$ in the target space in such a way that the distances r_{ij} are as close as possible to the original distances R_{ij}. A sum-of-squared error function can be used as a criterion to decide whether a given configuration of image points is better than another. There are two commonly used criteria:

Kruskal's `stress`: $S = \sqrt{\frac{\sum_{i>j}(R_{ij}-r_{ij})^2}{\sum_{i>j} R_{ij}^2}}$

Lingoes' `alienation coefficient`: $K = \sqrt{\frac{1-\sum_{i>j}(R_{ij}\cdot r_{ij})^2}{\sum r_{ij}^2}}$

In this paper we use the following measure [11] of preservation of topography:

$$D = \sum_{i>j} \left(R_{ij} - \alpha r_{ij}\right)^2 / \sum_{i>j} r_{ij}^2 \text{with} \alpha = \sum_{i>j} R_{ij} \cdot r_{ij} / \sum_{i>j} r_{ij}^2 \tag{1}$$

these measures may also be used to compare the quality of different SOM maps. The best configuration of points in the target space is found iteratively:

0. Define a starting configuration for the points Y_i randomly or by a principal components analysis,
While the criterion function significantly decreases, do: 1. Compute the distances r_{ij}.
2. Compute the value of the criterion functions S, K or D.
3. Find a new configuration of the points Y_i by a gradient-descent procedure such as Kruskal's linear regression or Guttman's rank-image permutation. To be sure that global minimum is found we use here the simulated annealing procedure.

Other quantitative measures of the preservation of topography between the high-dimensional input and low-dimensional target spaces may be introduced, for example:

$$D_2(Y;R) = \sum_{i>j}^{n} \left(R_{ij}^2 - r_{ij}\right)^2 = \sum_{i>j}^{n} \left(R_{ij}^2 - \sum_{l=1}^{k} \left(y_i^{(l)} - y_j^{(l)}\right)^2 \right)^2 \tag{2}$$

where $y_i^{(l)}$ are components of Y_i objects in the k-dimensional target space and the reduction in the number of the degrees of freedom going from N dimensions to k dimensions is taken into account by setting all

components of $Y_0 = 0$ and $k-1$ components of Y_1 to zero, $y_1^{(l)} = 0, l = 1..k-1$. For this measure we may obtain the best representation by solving a set of non-linear equations [11] instead of minimization:

$$\sum_{j \neq i}^{n} \left(y_i^{(m)} - y_j^{(m)} \right)^3 + \sum_{j \neq i}^{n} \left(y_i^{(m)} - y_j^{(m)} \right) \sum_{l \neq m}^{k} \left(y_i^{(l)} - y_j^{(l)} \right)^2 - \sum_{j \neq i}^{n} R_{ij}^2 \left(y_i^{(m)} - y_j^{(m)} \right) = 0 \qquad (3)$$

Unfortunately it is as hard to solve this system of nonlinear equations as it is to minimize the stress function.

3 Self-organizing maps

The SOM algorithm allows to perform in an unsupervised manner a visualization of high-dimensional input data, usually in a target space of one, two or three-dimensions. We will assume here a two-dimensional array of nodes. SOM seems to preserve the topography of the input data. This means that if some of the high-dimensional data points are grouped in clusters their representations in the map are also grouped in clusters and the relative distances between clusters are to some degree preserved. The SOM network takes as input a set of labeled sample vectors and gives as output an array of network nodes with the input vector labels attached to these nodes. Let N be the dimension and n the number of sample vectors $X(t) \in \Re^N, t = 1, 2, ..., n$, where each sample vector $X(t)$ is identified by a label. The two-dimensional output layer contains $i = 1, ..., x_{dim} \times y_{dim}$ nodes W_i, each serving as a codebook vector of dimension N. The training of the weight (codebook) vectors of the map's nodes is realized by the following algorithm: For a given number of iterations do:

1. Pick up randomly one sample vector $X(t)$
2. Find the nearest weight vector W_c: $\|X - W_c\| = \min_j \{\|X - W_j\|\}$
3. Update the weights W_i according to the rule:

$$W_i(t+1) = W_i(t) + h_{ci}(t) \cdot [X(t) - W_i(t)] \qquad (4)$$

where $h_{ci}(t)$ is the neighborhood function that can be of type:

- "bubble": $h_{ci}(t) = \alpha(t)$ if $\|W_c - W_i\| \leq r(t)$ and $h_{ci}(t) = 0$ if $\|W_c - W_i\| > r(t)$;
- "gaussian": $h_{ci}(t) = \alpha(t) \cdot \exp\left(\frac{-\|W_c - W_i\|}{2\sigma^2(t)}\right)$

Only the neurons within the neighborhood $h_{ci}(t)$ are moved near to $X(t)$. The learning rate $\alpha(t) \in [0, 1]$ decreases monotonically with time, $\sigma(t)$ and $r(t)$ are neighborhood radiuses decreasing also monotonically. Although one-dimensional Kohonen maps have been analyzed in some details little is known about the self-organization process in two or three dimensions [2]. The main problem is the lack of quantitative measure to determine what exactly "the good map" is. Such a measure is provided by MDS, therefore we will use MDS maps of simplexes to improve the quality of SOM mappings.

4 Mapping of simplexes

Minimization in MDS is usually done via gradient procedures. Since we are looking for a global minimum we have also used simulated annealing method for minimization, although it requires much more extensive computations. N-dimensional simplex has $N+1$ corners and the distances between any two corners are all equal to one. Configuration of points obtained by mapping corners of simplexes in 3-20 dimensions by MDS is shown in Figures 1-4. For comparison a typical SOM map is for 10 dimensions. For all MDS maps the final values of D measure of topography preservation is given.

The MDS configurations in the target space are highly symmetric and we are rather confident that they really represent absolute minima. Up to 6 dimensions the best configurations consists of equally spaced points on a circle. In higher dimensional spaces the points on the outer circle are also equally spaced but additional points appear. In 7-9 dimensions one point in the middle of the circle appears, in 10 dimensions this point is replaced by the inner circle with two points, in 11-13 dimensions with three points, 14-17 with four points, and in 18 and 19 dimensions there are 5 points on the middle circle. In 20 dimensions

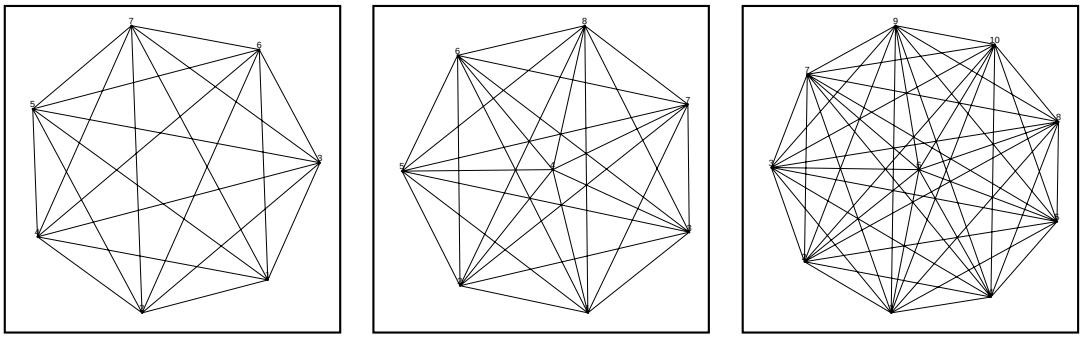

Figure 1: MDS mapping of simplexes in 6, 7 and 9 dimensions, D = 0.086, 0.095, 0.110

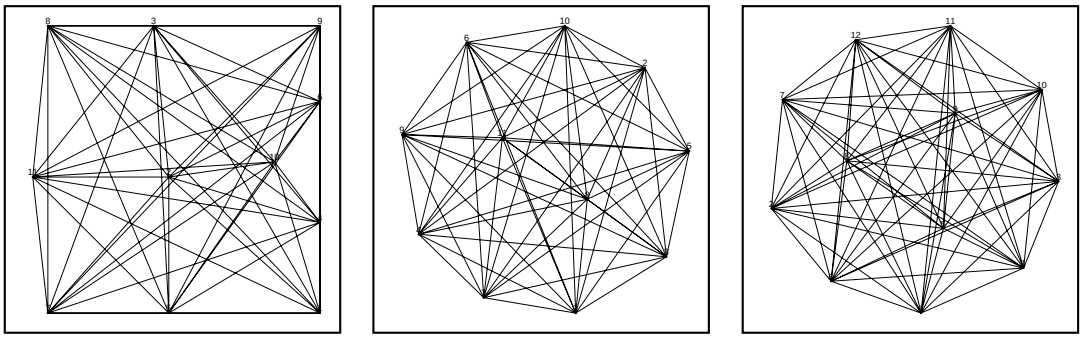

Figure 2: SOM map of 10 dimensional simplex and MDS maps of 10 and 11 dimensional simplexes, D = 0.116, 0.121

the best MDS configuration has one point in the middle of the circle, an inner circle with 6 point and an outer circle with 14 points.

High symmetry of these structures suggests that one should compare the error measures of perfectly symmetric configurations, which one can do minimizing in one dimension (radius of the mapped circle) or three dimensions (scales of two circles and an angle of mutual rotation).

In figures 2-4 all corners of the simplexes are connected by lines. All these lines are of unit length in the input space but obviously it is impossible to preserve such topography in two dimensions. SOM tries to use all neurons and places many codebook vectors at the boarders, missing the best configuration even in the three dimensional case. Perhaps with infinitely slow learning (i.e. decreasing the neighborhood function $h_{ci}(t)$ to zero infinitely slowly) it would be possible to avoid "freezing" wrong configurations, but even repeating 10000 times the presentation of 4 data points in three dimensions (or 11 data points in 10 dimensions) were not sufficient to bring SOM maps closer to MDS results. SOM results could be much better if instead of a rectangular network a circular one was used but it is impossible to say a priori what shape should the network layer have.

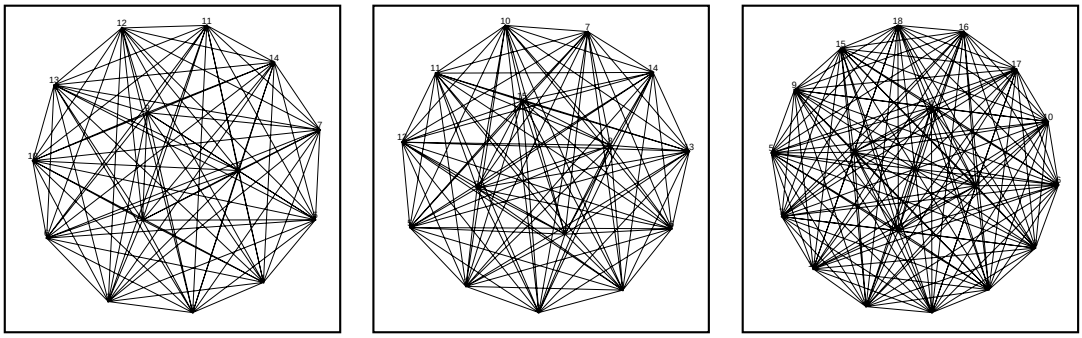

Figure 3: MDS mapping of simplexes in 13, 14 and 17 dimensions, D = 0.128, 0.131, 0.138

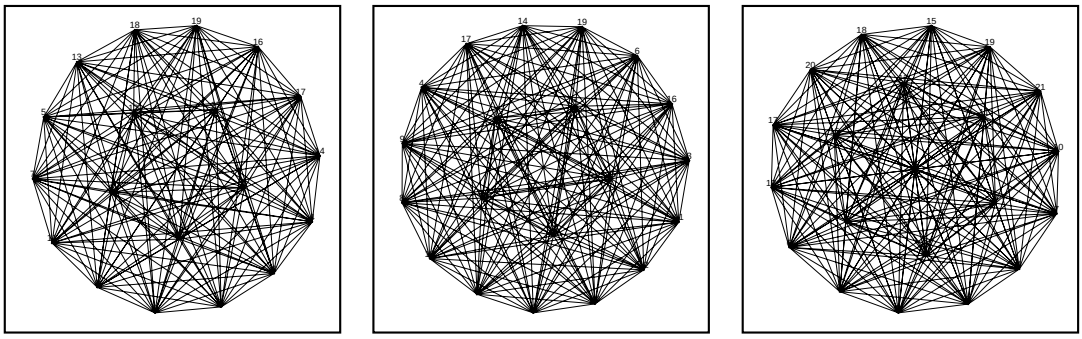

Figure 4: MDS mapping of simplexes in 18, 19 and 20 dimensions, D = 0.140, 0.141, 0.143

5 Simplexes for initialization of SOM

The results obtained in the previous section may be used to improve the convergence and to reduce the topographical distortions of SOM method. In the preparation phase MDS is performed for simplexes in spaces of growing dimensionality. The sets of two-dimensional images of the corners of simplexes are stored. This step is done once and the data is stored in tables or computed from analytical functions. We will provide such data in computer-readable form soon.

SOM network should be placed in the target space in such a way that each neuron has coordinates that are close to images of the simplex corners. All weights of SOM neurons are initialized to zero. The algorithm may be summarized as follows:

1. First scan through the data: The coordinate system is placed in the middle of $N+1$ dimensional simplex in such a way that each corner is at a unit distance and lies on a sphere. All N-dimensional input vectors $X = (X_1, X_2, ..., X_N)$ are extended by adding an additional dimension $V = (X, \sqrt{1 - ||X||^2})$. These vectors are normalized to one and define a point on the $N+1$ dimensional sphere. Information about the magnitude of the input vectors is explicitly added as the last component of the extended vectors.

The input data vectors are read and the three nearest simplex corners C_1, C_2, C_3 for each V vector are identified. The image v of the input vector is found by triangulation based on the images c_1, c_2, c_3 of the three simplex corners in the target space. First the scale factor is defined, $s = \frac{d(c_1,c_2)}{D(C_1,C_2)}$ where $d(\cdot,\cdot)$ is the target space metric and $D(\cdot,\cdot)$ is the input space metric. The distance $R_1 = D(V, C_1)$ between the input vector V and the nearest simplex corner is mapped to $r_1 = sR_1$, defining a radius of a circle around c_1. The second circle defined around c_2 has radius $r_2 = sD(V, C_2)$. The two intersections of these circles define two possible images of the V vector. The distance from the C_3 simplex corner is used to select the final position of v.

The neuron in the SOM network that is closest to the image of the target vector is selected as a winner. If the neuron has not been used before its weights become equal to the input vector. If it has been used a normal iterative updating of the weights of neighboring neurons is performed.

Subsequent data scans: usual SOM procedure is performed. To avoid global reorganization of the map the neighborhood radius is always kept small relatively to the distances between images of simplex corners. To increase the accuracy of the data representation (decrease the quantization error) the density of neurons should be high. Significant computational costs savings are obtained if the identification of a winning neuron, requiring comparison of new data vector with weights of all N_{net} neurons in the network is replaced by identification base on comparison with N corners of the simplex, as described above.

As a result after a single scanning of the input data the map has the best topographical structure and only the tuning of the weights is needed in the final learning steps.

6 Summary and conclusions

We have presented here a simple and computationally inexpensive method for creation of SOM networks that are optimal from the point of view of topographical distortions of the relations among the original data vectors. In contrast to the Kohonen's SOM procedure some neurons after learning may have

zero weights and could be removed from the map. Large and dense initial networks may be used for higher accuracy since the cost of the method is proportional to the number of active neurons only. The procedure requires availability of the MDS mappings of simplex corners. Although we have discussed two-dimensional target spaces only there is no reason why three or higher-dimensional target spaces could not be used in the same way.

In the input spaces of perhaps up to 10 dimensions one could also use corners of hypercubes, but already for 10 dimensions there are 1024 hypercube corners and only 10 simplex corners. Since in the first pass through data distances to all corners are computed simplexes have definite advantage.

Several other ways of creating near-optimal topographical representations, based on MDS and SOM, are possible:

1. Use MDS map of a simplex for reference. Store only the images (coordinates in the target space) of the input data that differ on more than a specified threshold. This method should be useful for visualization, dimensionality reduction and classification.

2. Introduce an adaptive coordinate mesh in the MDS target space. Each node has a local coordinate system associate with it, parametrized by scaling factors and angles between the axes. Adaptive parameters are trained in supervised way to reduce classification or approximation error.

3. Combine the term coming from gradient minimization of stress measure with local adaptation of weights by the winning neuron. The learning in this case would be even slower than in SOM but maps should preserve global character.

Acknowledgments Support by the Polish Committee for Scientific Research, grant 8T11F 00308, is gratefully acknowledged.

References

[1] T. Kohonen, The self-organizing map. Proceedings of the IEEE 78 (1990) 1464-1480.

[2] T. Kohonen, Self-organizing maps. (Heidelberg Berlin, Springer-Verlag 1995).

[3] Zupan Jure, Gasteiger Johann, Neural networks for chemists: an introduction (VCH, Weinheim, Germany 1993)

[4] W.S. Torgerson, Multidimensional scaling. I. Theory and method. Psychometrika, 17 (1952) 401-419

[5] C.H. Coombs, A theory of data. (New York: J. Wiley 1964)

[6] J.B. Kruskal, Nonmetric multidimensional scaling: a numerical method. Psychometrika 29 (1964) 115-129.

[7] J.C. Lingoes, E.E. Roskam, I. Borg (Eds.), Geometric representation of relational data. (Ann Arbor: Mathesis Press 1979).

[8] R.N. Shepard, The analysis of proximities: multidimensional scaling with an unknown distance function. Psychometrika 27 (1962) 125-140.

[9] MDS programs are available now in a few statistical packages such as STATISTICA or SYSTAT for Windows. Fortran routines may be obtained from the Netlib repository of software, http://www.netlib.org/mds/index.html

[10] J.W. Sammon Jr., A nonlinear mapping for data structure analysis. IEEE Transactions on Computers C-18 (1969) 401-409.

[11] W. Duch, Quantitative measures for the self-organized topographical mapping. Open Systems and Information Dynamics 2 (1995) 295-302

[12] Duch W and Naud A (1996) On global self-organization. Proc. of the European Symposium on Artificial Neural Networks (ESANN'96), Bruge 22-26.4.1996, pp. 91-96