On global self-organizing maps.

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

Abstract. Self-Organizing Feature-Mapping (SOFM) algorithm is frequently used for visualization of high-dimensional (input) data in a lower-dimensional (target) space. This algorithm is based on adaptation of parameters in local neighborhoods and therefore does not lead to the best global visualization of the input space data clusters. SOFM is compared here with alternative methods of global visualization of multidimensional data, such as the multidimensional scaling (MDS) and Sammon non-linear mapping, methods based on minimization of error function measuring topographical distortions. SOFM is inferior as a visualization method but facilitates faster classification. A combination of global methods with SOFM should be useful for visualization and classification.

1. Introduction

The Self-Organizing Feature-Mapping (SOFM) algorithm introduced in 1981 by Kohonen [1] is usually presented as a particular type of artificial neural network. The network is first trained on the high-dimensional samples in such a way that 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, since there is no intervention of a teacher correcting the errors or providing information about clusters in the data space. Kohonen's intentions in developing SOFM were the following “The SOM has not been meant for statistical pattern recognition; it is a clustering, visualization, and abstraction method. Anybody wishing to implement decision and classification processes should use LVQ (Learning Vector Quantization) instead of SOM.” Nevertheless the method is used very often as a classification method (cf. [2]) because of its unique ability to display multidimensional data in two or three-dimensional maps. Human experts can analyze data in at most three dimensions, therefore evaluation of high-dimensional data is possible only if its dimensionality is reduced. SOFM facilitates classification by visual inspection, enabling estimation of the relation of new inputs to the data clusters already learned.

Statistical method known as the multidimensional scaling (MDS) technique has foundations in the work of Torgerson [3] and in the Coombs theory of data [4]. Computer programs and applications of MDS have been developed, among others, by Kruskal at the Bell Laboratories, by Lingoes, Roskam and Borg, and by Shepard [5]. MDS was used primarily to obtain a lower-dimensional representation of psychological data. These data are obtained by subjective evaluation of similarities or dissimilarities between different items (perceived nearness of objects, preferences, feature intensity or affinity), 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. Unfortunately MDS methods are almost unknown outside the field of mathematical psychology. A very similar nonlinear mapping method for visualization and data analysis has been developed by Sammon [6] and by one of us [7].

After a brief presentation of the SOFM algorithm the global visualization algorithms, MDS, Sammon mapping and quadratic measures are presented. These algorithms are compared with Kohonen's approach for mapping of points on the sphere and corners of hypercubes to a two-dimensional target space. The deficiencies of SOFM in visualization tasks and global methods in classification tasks are apparent. Finally a reliable visualization and classification method using a combination of MDS and SOFM ideas is recommended.

2. Self-organizing feature maps

We will assume here a two-dimensional array of nodes, equivalent to the two-dimensional target space for visualization. Unfortunately analytical results in two or more dimensions are hard to obtain. SOFM seems to preserve the topography of the input data, i.e. if some of the high-dimensional data vectors are grouped in clusters then their representations in the map are also grouped in clusters and the relative distances between nearby clusters are to some degree preserved. The SOFM 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 of the n sample vectors $\mathbf{X}(t) \in \Re^N, t = 1, 2..n$, where each sample vector $\mathbf{X}(t)$ is identified by a label. The two-dimensional output layer contains a rectangular mesh of k = 1, ..., $x_{dim} \times y_{dim}$ nodes, each serving as a codebook vector $\mathbf{W}_k$ of dimension N. The training of the weight (codebook) vectors of the map's nodes is realized by the following algorithm [1]:

For a given number of iterations do:

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

$$\mathbf{W}_i(t+1) = \mathbf{W}_i(t) + h_{ci}(t)\,[\mathbf{X}(t) - \mathbf{W}_i(t)]$$

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

$h_{ci}(t) = \alpha(t) \exp(-||\mathbf{W}_c-\mathbf{W}_i|| / 2\sigma^2(t))$ or of a local “bubble” type [1].

Weights of neurons laying in the neighborhood $h_{ci}(t)$ of the winning neuron are moved closer to $\mathbf{X}(t)$. The learning rate $\alpha(t) \in [0,1]$ decreases monotonically with time, $\sigma(t)$ determining the radius of the neighborhood also decreases monotonically. After many iterations and slow reduction of $\alpha(t)$ and $\sigma(t)$ until the neighborhood covers only a single node the map is formed: neurons with weights that are close in the parameter space $\mathbf{W}$ are also close on the mesh and can be labeled with names (classes) of input clusters. A lot of work has been done in recent years on improving the convergence properties of the SOFM algorithm using statistical methods and information-theoretic models, but the main problem with the lack of quantitative measures to determine when „a good map" is formed seems to be unresolved. This problem is addressed directly by the methods presented in the section below.

3. Multidimensional scaling

MDS techniques emerged from the need to visualize in a one, two- or three-dimensional spaces high dimensional objects described by some measure of their similarities or dissimilarities. The problem is to find coordinates of points representing these multdimensional objects in a low-dimensional target space in such a way that the low-dimensional interpoint distances would correspond to the similarities of the original objects. MDS takes as input a symmetric matrix of the similarities or dissimilarities between objects, whereas SOFM needs absolute coordinates of these objects in the high-dimensional space. Note that for the MDS input space does not even need to be a metric space. If a given observation concerns n objects there are $n(n\text{-}1)/2$ distances between these objects. SOFM algorithm in the same case needs $n\times N$ input values, where N is the dimension of the input vectors. Coordinates are also sufficient to compute distances for MDS but the reverse is not true. If the number of objects $n<2N+1$ SOFM algorithm requires more information than MDS.

Let δ_{ij} be the observed similarities between objects $\mathbf{X}_i$, i = 1,2 .. n, equivalent to distances $\delta_{ij} = \|\mathbf{X}_i\text{-}\mathbf{X}_j\|$ in metric spaces. Let $\mathbf{Y}_i$ be the low dimensional target space point representing the input object $\mathbf{X}_i$ and let d_{ij} be the distance between $\mathbf{Y}_i$ and $\mathbf{Y}_j$. We have to place the points $\{\mathbf{Y}_i,\ i = 1, .., n\}$ in the target space in such a way that the distances d_{ij} are as close as possible to the original distances δ_{ij}. Kruskal proposed [5] the **stress coefficient** $S(\delta_{ij},d_{ij})$ while other authors [5] advocated the **alienation coefficient** $A(\delta_{ij},d_{ij})$ as the measure of topographical agreement between the input and the target space. Sammon [6] has proposed an error function $E(\delta_{ij},d_{ij})$

$$S=\sqrt{\frac{\Sigma_{i>j}(\delta_{ij}-d_{ij})^2}{\Sigma_{i>j}\,\delta_{ij}^2}}\ ;A=\sqrt{\frac{1-\Sigma_{i>j}(\delta_{ij}\cdot d_{ij})^2}{\Sigma\, d_{ij}^2}}\ ;E=\sum_{i<j}\frac{(\delta_{ij}-d_{ij})^2}{\delta_{ij}}\Bigg/\sum_{i<j}\delta_{ij}$$

In all cases the goal is achieved by iterative minimization of one of these coefficients as follows: compute δ_{ij} and define a starting configuration for the points $\mathbf{Y}_\mathrm{i}$ randomly or by a principal components analysis.

While the criterion function significantly decreases, do:

1. Compute the distances d_{ij}.
2. Compute the value of the criterion functions S, A or E.
3. Find a new configuration of points $\mathbf{Y}_\mathrm{i}$ by a minimization procedure, such as rank-image permutation procedure or monotone regression transformation procedure.

Looking for quantitative measures of the preservation of topography between the high-dimensional input and low-dimensional target spaces in the SOFM algorithm Duch [7] has introduced the stress-like measure $D_1(\delta_{ij},d_{ij}) = S(\delta_{ij},d_{ij})$ and its quadratic version written here in an unnormalized form, explicitly dependent on the absolute coordinates in the target space rather than on the distances d_{ij}:

$$D_2(\mathrm{Y};\delta)=\sum_{i>j}^{n}\left(\delta_{ij}^2-\sum_{l=1}^{k}\left(y_i^{(l)}-y_j^{(l)}\right)^2\right)^2$$

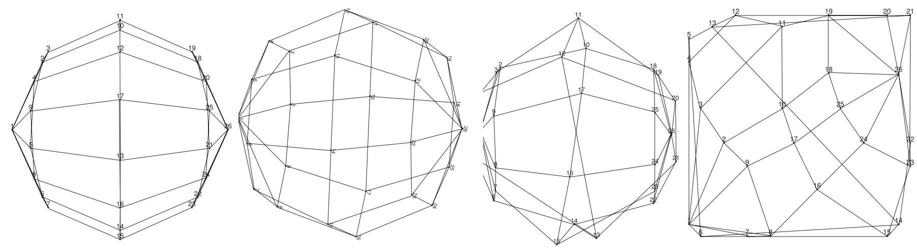
where $y_i^{(l)}$ are components of $\mathbf{Y}_i$ vectors in the k-dimensional target space. The reduction in the number of the degrees of freedom going from N dimensions to k dimensions is taken into account by setting all components of $\mathbf{Y}_0$=0 and k-1 components of $\mathbf{Y}_1$ to zero (fixing origin and rotation of the coordinate system in the target space). For this measure we may obtain the best representation by solving the following set of non-linear equations [7] instead of minimization:

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

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

4. Comparison of SOFM and MDS maps

Minimization in MDS is usually done via gradient procedure. Since we are looking for a global minimum we have used simulated annealing method for minimization. We have applied SOFM and MDS algorithms to a number of cases in which the quality of maps could be assessed easily. Due to the lack of space only two cases will be presented: mapping 3D points from the sphere with radius 1, the 26 points taken are the 2 poles and points at the intersection of 3 parallels with 8 meridians regularly spaced by an angle of 45 degrees, and mapping the corners of hypercubes from 3, 4 and 5 dimensions to the two-dimensional target space. Configurations of points obtained from SOFM and MDS are compared in figures below.

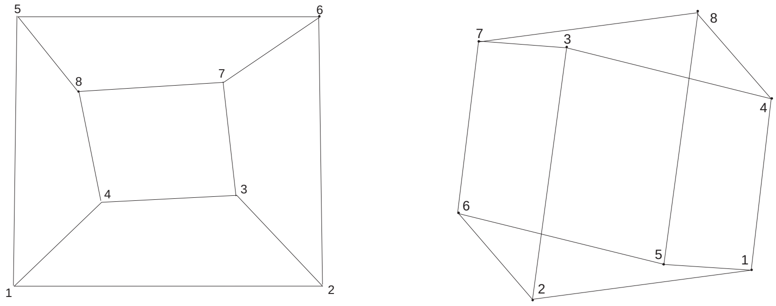


The two-dimensional representations of the 26 points on the sphere obtained by minimization of *S*, *E*, *A*, and by SOFM (left to right) with a 20 x 20 neurons map.

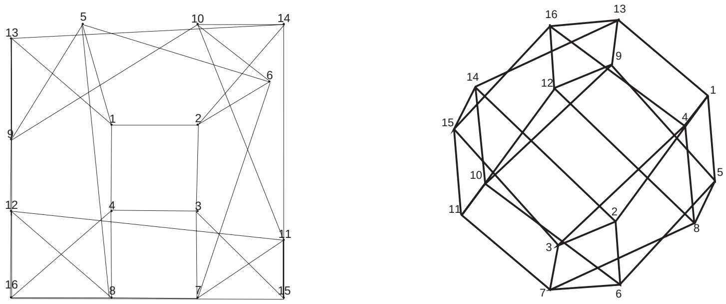
In figures 1-4 all points that are adjacent to each other in the input space are connected by lines. In a good map all these lines should be short but SOFM tries to use all neurons and places many codebook vectors at the borders, missing the best configuration even in the 3D case. Perhaps with infinitely slow learning it would be possible to avoid “freezing” wrong configurations but in our simulations even 10000 iterations were not sufficient to obtain a global minimum of the stress function.

A common measure of topographical distortions D defined [7] by:

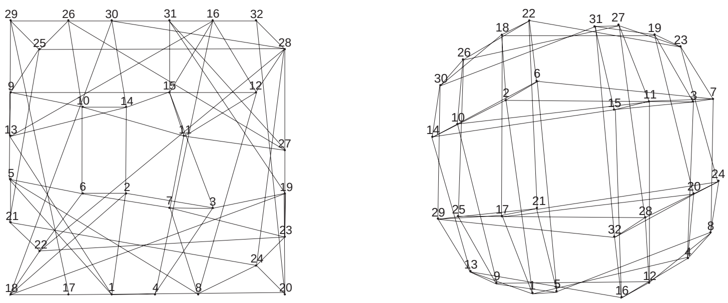
$$D=\Sigma_{i<j}\,(\delta_{ij}-\alpha.d_{ij})^2\Big/\Sigma_{i<j}\,d_{ij}^2\ \text{, with }\alpha=\Sigma_{i<j}\,\delta_{ij}.d_{ij}\Big/\Sigma_{i<j}\,d_{ij}^2$$



The two-dimensional representations of the 8 points of the 3D cube obtained by SOFM (left) with a 20 x 20 neuron map, a training of 10000 cycles, a final quantization error of 0.001 and stress 0.321, MDS (right) has the final stress value 0.246 after 22 iterations.



The two-dimensional representations of the 16 points of the 4D hypercube obtained by SOFM (left) with a 20 x 20 neurons map, a training of 10000 cycles, quantization error of 0.001, stress value 0.327, MDS (right) has the final



The two-dimensional representations of the 32 points of the 5D hypercube obtained by SOFM (left) with a 20 x 20 neurons map, a training of 10000 cycles, the stress value of 0.353 and by MDS (right) with a final stress of 0.333 after 18 iterations.

applied to these 4 cases gives 0.0636 for stress minimization, 0.0652 for Sammon's mapping, 0.0674 for alienation coefficient minimization and 0.0891 for SOFM.

5. Conclusions

Theoretical considerations as well as computational experience shows that MDS and other approaches based on minimization of global coefficients lead to a better visualization of multidimensional data than SOFM. Global visualization methods provide well-defined measures of the quality of Kohonen and other such maps. Computational demands of the two approaches in the learning or mapping phase are similar. Since SOFM doesn't give good maps and is not recommended as a classification method the question arises: what is it good for? Modifications of SOFM are useful in biologicaly-oriented models of cortical maps [8].

It is also possible to modify the original Kohonen SOFM algorithm to take into account the minimization of stress measure together with minimization of quantization error (Duch and Naud, in preparation). Such maps may be useful for classification. A new data added to MDS input requires costly minimization, therefore our recommendation is to use a combination of MDS and SOFM. At first minimization of stress-like coefficient is performed for the training data. Then the coordinate mesh in the target space is introduced. The codebook vectors in each node of this mesh are obtained by topological interpolation [9] from MDS results. This method should be useful not only in visualization and classification but also in approximation problems (Duch and Naud, work in progress).

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

References

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

[2] Zupan Jure, Gasteiger Johann, Neural networks for chemists: an introduction (VCH, Weinheim, Germany 1993); Bibliography of more than 1500 papers on SOFM is stored in: colchea.hut.fi, in the catalog /pub/ref file references.bib.Z

[3] W.S. Torgerson, Multidimensional scaling. Psychometrika, 17, 401-419 (1952)

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

[5] J.B. Kruskal, Nonmetric multidimensional scaling: a numerical method. Psychometrika 29, 115-129 (1964); J.C. Lingoes, E.E. Roskam, I. Borg (Eds.), Geometric representation of relational data. (Ann Arbor: Mathesis Press 1979); R.N. Shepard, The analysis of proximities. Psychometrika 27, 125-140 (1962)

[6] J.W. Sammon Jr., A nonlinear mapping for data structure analysis. IEEE Transactions on Computers C18, 401-409 (1969)

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

[8] E. Erwin, K. Obermayer, K. Schulten, Models of orientation and ocular dominance columns in the visual cortex: a critical comparison. Neural Comput. 7, 425-468 (1995)

[9] J. Göppert, W. Rosentiel, Topological interpolation in SOM by affine transformations. ESANN'95, pp. 15-20 (D facto publications, Brussels 1995)