A SOM based cluster visualization and its application for false coloring

Johan Himberg
Helsinki University of Technology
Laboratory of Computer and Information Science
P.O. Box 5400, FIN-02015 HUT, Finland
email: Johan.Himberg@hut.fi

December 15, 1999

Abstract

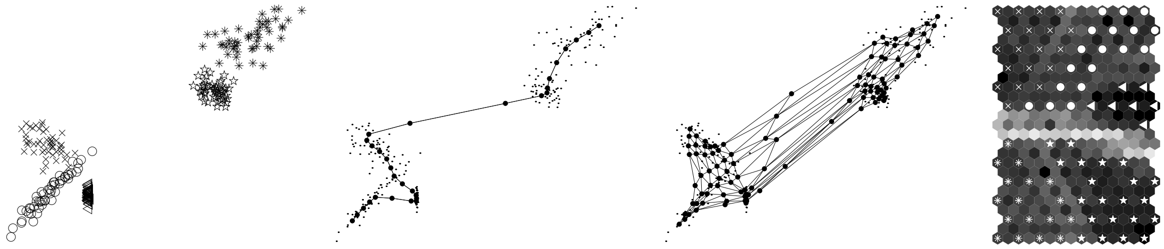
The self-organizing map (SOM) is widely used as a data visualization method in various engineering applications. It performs a non-linear mapping from a high-dimensional data space to a lower dimensional visualization space. In this paper, a simple method for visualizing the cluster structure of SOM model vectors is presented. The method may be used to produce tree-like visualizations, but the main application here is to get different color codings that express the approximate cluster structure of the SOM model vectors. This coloring may be exploited in making false color (pseudo color) presentations of the original data. The method is especially meant for making an easily implementable, explorative cluster visualization tool.

1 Introduction

The Self-Organizing Map (SOM) [4] consist of map units (neurons) that are ordered in a regular, often one or two-dimensional grid. The grid fixes the topological relations between the units. In the input space (data space), a model vector $\mathbf{m}_i \in \mathcal{R}^k$, where k is the dimension of the vectors, is attached to each unit. During the training algorithm the SOM behaves like a flexible net that folds onto the "cloud" formed by the input data (see Fig. 1). The SOM can be used to perform a topology preserving mapping from the high-dimensional data space to map units. Since the neighboring units in the grid tends to model similar regions of the data space, the SOM can be used for presenting overviews of high-dimensional data. The regular grid structure of the SOM is often used to visualize some characteristics of the model vectors or the cluster structure of them.

Since the map is an organized presentation of the data, one can think of a simple false coloring (pseudo coloring) method using the map grid: A feature vector ($\mathbf{x}_k$) is extracted from each region R_k of a picture and a (two-dimensional) SOM is trained using these data. Next, each unit of the SOM is given a color according to its location on the map. A straight-forward choice is to set a rectangular slice of the Red-Green-Blue cube (RGB) color model on the grid for this purpose. The regions of the picture are recolored using the color assigned to the best-matching unit (BMU) of its feature data vector $\mathbf{x}_k$. (The BMU for a data vector $\mathbf{x}_k$ is the SOM unit c whose model vector $\mathbf{m}_c$ is closest to that data vector.) Examples for coloring images or maps in different applications using SOM include [1, 3, 6, 7].

It would be appealing to color the SOM units so that the result would somehow reflect the cluster structure of the model vectors. In [3], a rigorous method for this purpose is presented. It uses a stochastic optimization process to project the SOM model vectors onto a two-dimensional plane in such a way that the distances between the model vectors of neighboring map units are approximately preserved. The optimization may be further constrained to fit the perceptually based CIELab color model. As a result, the perceived differences between colors assigned to nearby units on the grid reflect the local distances between the model vectors. On the same time, the overall topological ordering of the SOM reflects the larger structures (dissimialrities) on the data.



(a) The data set (b) 1D SOM and data (c) 2D SOM and data (d) U-matrix

Figure 1: Subfigure (a) presents the two-dimensional data set used in this paper. Data were generated by randomly picking samples from five Gaussian distributions, 50 points from each. Subfigures (b) and (c) contain the one and two-dimensional SOMs that are used in this paper: both are trained with the data in (a) using the Batch SOM-algorithm [4] implemented in the SOM Toolbox software package [2]. The larger dots represent units, the smaller ones the data, and the lines show the SOM topology. Subfigure (d) presents the unified distance matrix (U-matrix) [5] for the SOM in (c). In U-matrix, the distances between the adjacent SOM units are presented using gray levels. Light gray stands for long distances and dark for short ones. On the U-matrix, the symbols ($\times$, $\circ$, $\triangleleft$, $*$, $\star$) express the cluster in (a) from which the majority of data maps to that unit.

In this paper, the idea of relying on the topological ordering of the SOM is used, but instead of rigorously optimizing the local distances, a simple contraction model is constructed. It may be used to visualize fast the approximate cluster structure of the map model vectors, and especially, it is a basis for cluster based coloring of the SOM and data.

2 Simple contraction model for cluster visualization

The contraction model for enhancing the visualization of the SOM is intuitively described in the following way: The distances between the model vectors $\mathbf{m}_i$ of a SOM are calculated and the resulting distances are transformed to similarities using some suitable transformation. The units of the SOM have also the prespecified (topological) coordinates on a low-dimensional output space (the grid). These unit coordinates are used as starting point in the contraction process. New coordinates for a unit are calculated as the weighted average of the locations of the units. The weights are the unit's similarities to itself and others (normalized so that they add to one). If the averaging operation is repeated, all units eventually fall to one point. The traces of the points may be visualized as a tree-like figure (see Fig. 2).

More precisely explained, there exist a set of objects $\Omega_i, i = 1, \ldots, N$. Between these objects a dissimilarity matrix D is defined, so that the element d_{ij} is $\delta(\Omega_i, \Omega_j)$ where δ is a dissimilarity measure. Using D, we define a similarity matrix $\tilde{S}$, setting its element $\tilde{s}_{ij} = f(d_{ij})$. The function f is chosen so that it reflects the similarity: a reasonable f is a monotonic function having value 1 for exact similarity (zero distance) and value 0 for infinite dissimilarity (distance). In this paper the function

$$\tilde{s}_{ij} = f(d_{ij}) = e^{-\frac{d_{ij}^2}{T}} \tag{1}$$

is heuristically used. T is a free parameter that is used to control the peakness of f. Next, $\tilde{S}$ is normalized so that in the resulting matrix S every row add to one: $s_{ij} = \tilde{s}_{ij} / \sum_{j=1}^{n} \tilde{s}_{ij}, i = 1, \ldots, n$. We define some initial coordinates $\mathbf{x}_i = (x_1 \; x_2 \; \ldots \; x_k)^T$, $\mathbf{x}_i \in \mathcal{R}^k$ for each object Ω_i, and a matrix $X_0 = (\mathbf{x}_1 \; \mathbf{x}_2 \; \ldots \; \mathbf{x}_n)^T$ of these coordinates. Now, the contraction process becomes

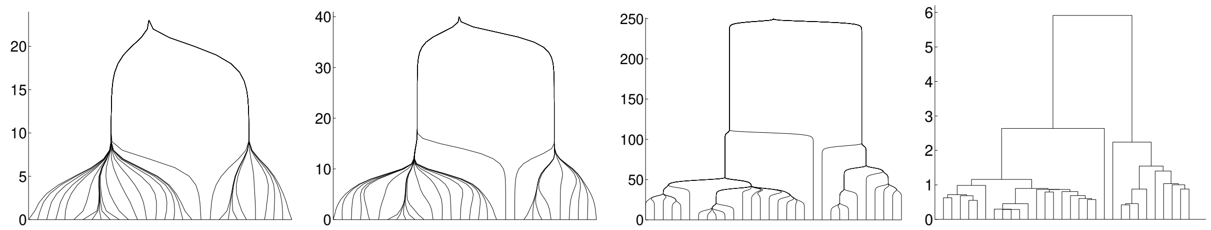
$$X_{i+1} = SX_i, \;\; i = 0, 1, \ldots, r \tag{2}$$

or

$$X_r = S^r X_0 \tag{3}$$

where r denotes the number of successive averagings. The contraction process may be visualized by drawing traces of points — presenting the objects — when they travel from some prespecified initial positions X_0 to the average point $S^r X_0$, $r \to \infty$. Visualization, in general, is possible only up to three geometric dimensions. If the objects are high-dimensional vectors, the problem is how to choose suitable starting points X_0 for the visualization. The SOM offers a solution for this as it defines an order for the high-dimensional data on a low-dimensional space. When S is calculated using the distances between the model vectors $\mathbf{m}_i$ in the input (data) space, the initial coordinates X_0 are set to be the coordinates of the SOM units in the grid (output space). Similarities s_{ij} are calculated, as explained previously from the distance matrix between map model vectors. Here $d_{ij} = \|\mathbf{m}_i - \mathbf{m}_j\|$, where $\|\cdot\|$ denotes the Euclidean norm.

Examples for the one-dimensional SOM in Fig. 1(b) using different values for T are shown in Fig. 2. The initial locations $X_0 = (x_1 \; x_2 \; x_3 \; \ldots \; x_N)^T$ of the N units are simply set to 0, $1/N$, $2/N, \ldots,$ N/N. It is



(a) T=1 (b) T=0.5 (c) T=0.07 (d) Single-linkage

Figure 2: Subfigures (a)–(c) show the contraction process visualization for the SOM presented in Fig. 1(b). Values for T (in Eq. 1) are expressed below subfigures (a)–(c). In order to speed up the contraction process, r (in Eq. 3) are chosen to be $r = 2^i, i = 0, 1, \ldots, n$ as this may easily be implemented by successive squarings of S: the y-axis of subfigures (a)–(c) show values of i. In subfigure (d), there is a dendrogram that is formed using a single-linkage agglomerative clustering for the same model vectors. The y-axis shows here, by convention, the cluster distances when linking. It seems to be that the contraction model for SOM produces visually similar results when $T \to 0$, if the y-axis has logarithmic scale with repect to r, as here. At the same time the number of required steps grows.

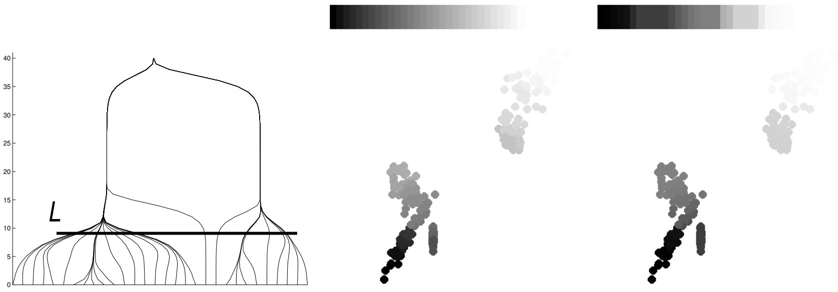
possible to draw similar tree-like visualizations using a two-dimensional map, as in Fig. 4(c), though this is evidently more difficult to inspect.

3 Coloring the SOM and data

The main purpose of the presented contraction model is to make a simple false coloring scheme for a two-dimensional SOM. The idea is, however, first presented using the one-dimensional SOM in Fig. 1(b). This is done because a gray level coding is sufficient for the one-dimensional model. Thereafter, the two-dimensional case is briefly sketched using a gray level–size pseudo color coding.

The coloring of the one-dimensional SOM is done according to Fig. 3. We have an intensity (gray level) coding $I(X)$, where X contains locations of the SOM units at some step of the contraction iteration. For the iteration step r the gray level for units are $I(S^r X_0)$. In order to gain more intensity resolution the coordinates may be linearly normalized so that the minumum and maximum values are always 0 and 1, respectively.

The units of a two-dimensional SOM may be given initial colors when a slice of the RGB cube is set on the map grid, see Figs. 4(a) and 4(b). The obtained coloring may be transferred to some other presentation of the data as in Fig. 3.



(a) Contraction visualization

(b) Initial SOM and data coloring

(c) SOM and data coloring using the level L in (a)

Figure 3: Subfigure (a) shows the same contraction process as Fig. 2(b). In (b) the gray level stripe presents the original, linear intensity coding of the SOM units. The data points are colored according to their BMUs on the SOM. In (c) the gray level coding is taken from level L in (a): the linear gray level stripe is set on level L and the units pick the gray level which their trace happen to intersect. The obtained gray levels are scaled between maximum and minimum intensities in (c). In original screen shot of (c), five "clusters" can be approximately seen as separate bands on the gray level stripe. In paper copy the bands may not be clearly visible.

4 Discussion

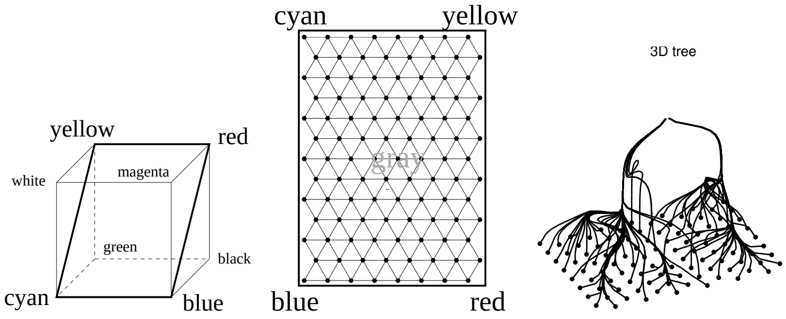
Figure 2(d) shows a dendrogram resulting of the traditional single-linkage agglomerative clustering procedure. It seems to be that when T is low, the contraction method produces a visualization that is very close to the dendrogram. However, when T is small the convergence is so slow that S has to be successively squared in order to get very high power for the similarity matrix S, as was done in Fig. 2. It is obvious that this is comptutationally of order $O(MN^3)$, which results of multiplying two $N \times N$ similarity matrices. (N is the number of map units and M is a constant referring to number of iterations which may be high). The standard single-linkage clustering is more close to $O(N^2)$ where the heaviest operation is the distance calculation.

There is, of course, no sense in running the contraction process to end using very low T – just in order to get computationally expensive version of the single-linkage clustering process. Instead, it may be used in order to get "soft" clustering visualizations for explorative purposes, as in Fig. 4. Then the interesting part of the process are probably some first steps, and Eq. 2 may be applied, and the computational load remains closer to $O(MN^2)$.

The single-linkage dendrogram (or any other dendrogram based on crisp agglomerative clustering), might be used to similar kind of coloring scheme as in this paper, but then the changes in coloring from one cluster level to another would be discrete. The contraction method can be tuned to produce tree-like structures with different speed of convergence using the parameter T. The parameter T and iteration step r can be used as tunable parameters in order to implement an explorative tool for inspecting the cluster structure of the SOM: T sets the "resolution" between fast convergence (an overall picture) and a single-linkage styled visualization, while r sets the level of clusterization.

To simulate a clustering procedure using the principle of contraction or gravitational model (e.g., [8]), is of course, a frequently used idea. However, the objective in this paper is not to propose a new clustering method, but an implementationally simple scheme for adding cluster information to the visualization of a SOM grid. It was originally designed for the same purpose as [3] but, instead of optimized distance visualization, it produces a "soft" cluster visualization using a reasonable heuristic.

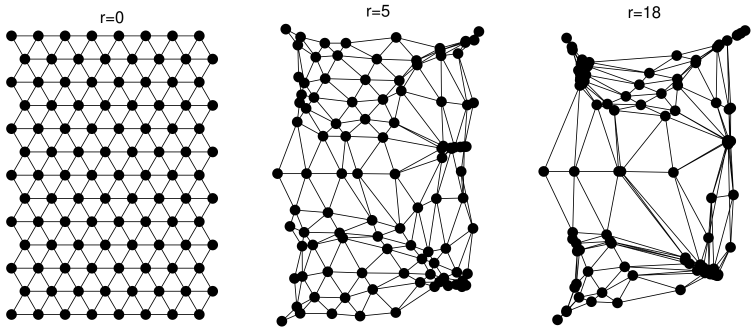
In this paper, a common procedure to set a slice of the RGB cube onto the map grid was used. The RGB model is actually not a very good choice as the perceived (dis)similarities between the colors and their distances in the color model space do not match. There exist standard models, such as CIELab (a textbook reference of CIE colorimetric systems, is e.g., [9]), where the distances in the color model space are better



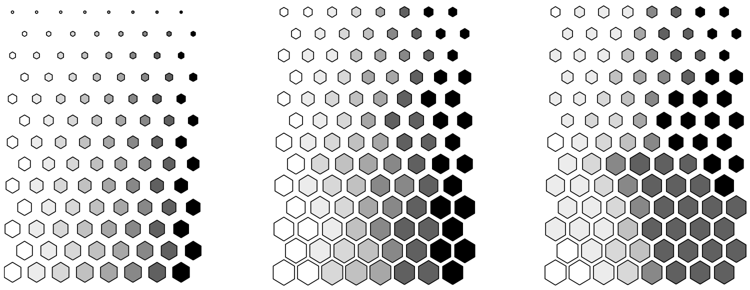
(a) Linear section of the RGB cube

(b) Initial color coding of the grid

(c) Contraction process



(d) Locations of the units on different stages of process



(e) Gray level–size pseudo codes for the coloring of the units

Figure 4: Subfigures (a) and (b) show the initial color coding for units: (a) shows the linear section of the RGB cube which is set, properly scaled, on the SOM grid in (b). The units get the color from the continuous palette according to their location In (c), a dendrogram- like visualization of the whole contraction process is shown. In (d), the locations of the units at some steps (r in Eq. 2) of the contraction process are shown with their topological connections. In (d) a gray level–size pseudocolor coding is used to roughly present the resulting cluster coloring, when the projections in (c) are used to obtain colors for SOM units by setting the color code on the projection as in (b). The impression of clusterization given by (c) can be compared to 1(d), and they seem to be in accordance. The value of T was 0.5 in this experiment.

in accordance with the perceived color dissimilarities. However, the method in this paper starts from the rectangular form of the SOM, so the sail-shaped chromaticity diagram of CIELab would have to be cut drastically in order to get the rectangular section. A practical problem is also that the calibration of displays or printers are often elaborate to do properly. Due to these complications the simple device dependent RGB model was still used as a working solution despite of its known deficiencies.

5 Conclusion

In this paper, a simple contraction process for the Self-Organizing Map (SOM) model vectors in order to make a hierarchical soft clustering visualization was presented. The motivation for this kind of method was to obtain an implementationally simple tool for making a color coding that brings up the cluster structure of a SOM. The contraction method presented here is especially suitable for a computing environment that is optimized for matrix computations, as the method is essentially based in successive averaging the original SOM output space unit coordinates using a normalized similarity matrix. Method includes parameters for getting different level and resolution of cluster visualization (coloring), which may be used to exploratively to produce different visualizations of the SOM cluster structure.

6 Acknowledgments

This work has been carried out in the technology program "Adaptive and Intelligent Systems Applications" financed by the Technology Development Center of Finland.

References

[1] Ewa J. Ainsworth. Classification of Ocean Colour Using Self-Organizing Feature Maps. In *Proceedings of IIZUKA'98*, volume 2, pages 996–999, Japan, October 1998.

[2] J. Vesanto et al. Self-Organizing Map for Data Mining in Matlab: the SOM Toolbox. *Simulation News Europe*, (25):54, March 1999.

[3] S. Kaski, J. Venna, and T. Kohonen. Coloring that Reveals High-Dimensional Structures in Data. In *Proceedings of ICONIP'99*, volume II, pages 729–734, 1999.

[4] Teuvo Kohonen. *Self-Organizing Maps*. Springer, 1995.

[5] A. Ultsch and H.P. Siemon. Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis. In *Proceedings of the International Neural Network Confererence (INNC'90)*, pages 305–308, Dordrecht, Netherlands, 1990. Kluwer.

[6] A. Varfis. On the use of two traditional statistical techniques to improve the readibility of Kohonen Maps. In *Proc. of NATO ASI workshop on Statistics and Neural Networks*, 1993.

[7] T. Villman. Topology preservation in self-organizing maps. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 288–290. Elsevier, 1999.

[8] W. Wright. Gravitational Clustering. *Pattern Recognition*, 9:151–166, 1977.

[9] G. Wyszeecki and W.S. Stiles. *Color Science*. Wiley, 2nd edition, 1982.