

Advanced visualization techniques for Self-Organizing Maps with graph-based methods

Georg Pözlzbauer¹, Andreas Rauber¹, and Michael Dittenbach²

¹ Department of Software Technology
Vienna University of Technology
Favoritenstr. 11-13, Vienna, Austria
{poelzlbauer,rauber}@ifs.tuwien.ac.at
² eCommerce Competence Center – ec3
Donau-City-Str. 1, Vienna, Austria
michael.dittenbach@ec3.at

Abstract. The Self-Organizing Map is a popular neural network model for data analysis, for which a wide variety of visualization techniques exists. We present a novel technique that takes the density of the data into account. Our method defines graphs resulting from nearest neighbor- and radius-based distance calculations in data space and shows projections of these graph structures on the map. It can then be observed how relations between the data are preserved by the projection, yielding interesting insights into the topology of the mapping, and helping to identify outliers as well as dense regions.

1 Introduction

The Self-Organizing Map [2] is a very popular artificial neural network algorithm based on unsupervised learning. It has been extended in several ways [1, 3]. It provides several beneficial properties, like vector quantization and topology preserving mapping from a high-dimensional input space to a usually two-dimensional map space. This projection can be visualized in numerous ways in order to reveal the characteristics of the input data or to analyze the quality of the obtained mapping. In this paper, we present a novel graph-based visualization technique, which provides an overview of the cluster structure and uncovers topology violations of the mapping. We propose two methods for defining the graph in input space. The first one computes a graph structure based on nearest neighbor calculations, and is especially useful for large SOMs, where map units outnumber data samples. The second method creates a graph structure based on pairwise distances between data points in input space, and its advantages are the easy identification of outliers and insight into the density of a region on the map. We provide experimental results to illustrate our methods on SOMs trained on the Ionosphere data set [7].

The remainder of this paper is organized as follows. In Section 2 a brief introduction to related visualization techniques is given. Section 3 details our

proposed method, followed by detailed description of its properties and experimental results with this method provided in Section 4. Finally, some conclusions are drawn in Section 5.

2 Related Work

In this section, we briefly describe visualization concepts related to our method. The most common ones are component planes and the U-Matrix. For an in-depth discussion, see [9]. The emphasis of our paper lies on visualization techniques that take the distribution of the data set in input space and its density into account. Most commonly, this is visualized as hit histograms, which display the number of data points projected to each map node. A more advanced method is the P-Matrix [8] that visualizes how densely populated each unit is by counting the number of data points within the sphere of a certain radius around the model vector in question. Another recently proposed technique that aims at depicting both density and cluster structures is the Smoothed Data Histogram [4], which relies on a parameter that determines how blurred the visualization will be. There are also techniques that depict the contribution of the individual variable dimensions to the clustering structure, like LabelSOM [5]. Other techniques providing insight into the distribution of the data manifold are projection methods like PCA and Sammon’s Mapping.

3 A Graph Projection Method

Our method investigates the proximity of the data vectors in input space and the preservation of pairwise distances after projection. First, we introduce a notation for both the SOM and the required concepts from graph theory. The input data set X contains N sample vectors x_i of dimension D^{input} . The SOM consists of M model vectors m_i of the same dimension as the input data, which are arranged on a two-dimensional map lattice, usually either in a rectangular or hexagonal fashion. Since the SOM is a vector projection technique, all data samples can be assigned a position on the map lattice. This is performed by finding the best-matching unit (BMU), formally

$$\phi(x_i) = \arg \min_j d(x_i, m_j) \quad (1)$$

where d is a suitable distance metric, like Euclidean Distance. The BMU is the prototype vector which is closest to the data sample x_i . The position of the model vector m_j on the map in the form of its two-dimensional position is also the projection of data vector x_i .

Next, we require some definitions from graph theory. A graph is a set of vertices and edges, formally $G = \{V, E\}$. The edges are usually represented by a square adjacency matrix (e_{ij}) . In case the graph is undirected the adjacency matrix is symmetric. We require that there are no connections from vertices to themselves, so the diagonal elements are all zero.

Then, we compute a graph G^{input} that captures the characteristics of the data set. The vertices of this graph are the data vectors. Our goal is to obtain a

set of edges that connect those data samples which satisfy a certain condition of proximity. We then aim at depicting the projection of G^{input} on the map lattice that visually link the corresponding map nodes with lines, indicating whether the original distances are preserved. In the following, we describe two methods of how to define the edges of G^{input} . The first one requires a parameter r and defines the data sample x_i to be adjacent to x_j if it lies within a sphere of radius r and center x_i . The entries of the $N \times N$ adjacency matrix are computed as

$$e_{ij}^{rad} = \begin{cases} 1 & \text{if } i \neq j \wedge d(x_i, x_j) \leq r \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The resulting graph is necessarily undirected due to the symmetry condition of distance metric d . The radius r serves as a threshold value. The number of edges increases monotonically with increasing r .

The second way to define the graph structure involves nearest neighbor calculations. It requires the integer parameter k which indicates how many neighbors to include. A sample x_j is connected to x_i if it is among its set of k nearest neighbors $N_k(x_i)$, formally

$$x_j \in N_k(x_i) \iff \text{Card}\{x_l \in X : l \neq i, j \wedge d(x_l, x_i) < d(x_j, x_i)\} < k \quad (3)$$

where Card denotes the number of elements of a set. In case of a tie in the ranking of the distances, this formula does not lead to a set of exactly k members, a policy to handle this exception has to be applied. Other than the radius method, the nearest neighbor relationship is not necessarily symmetric, so the definition of the elements of the adjacency matrix are defined as

$$e_{ij}^{kNN} = \begin{cases} 1 & \text{if } i \neq j \wedge (x_i \in N_k(x_j) \vee x_j \in N_k(x_i)) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Here, edges are defined if x_i is k nearest neighbor to x_j or vice versa. As with the radius method, increasing the value of k leads to more edges in the graph.

Once the graph is computed, it is ready for projection. A second graph G^{SOM} is defined for the output space. The vertices v_i^{SOM} are the prototype vectors m_i . The edges are preserved from the original structure such that a pair of model vectors are connected if two connected data samples are mapped onto them. The elements of the $M \times M$ adjacency matrix (e_{ij}^{SOM}) are defined as

$$e_{ij}^{SOM} = \begin{cases} 1 & \text{if } i \neq j \wedge \exists x_k, x_l : e_{kl}^{input} = 1 \wedge m_i = \phi(x_k) \wedge m_j = \phi(x_l) \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where e^{input} is the graph for the connectivity of the input space, either e^{rad} or e^{kNN} . While the number of vertices of the projection can be either greater or lesser than the number of vertices in input space, depending on whether the prototype vectors outnumber the sample vectors or vice versa, the number of projected edges are at most equal to the number of edges before the mapping is applied. Connected data points mapped to the same map unit are not counted

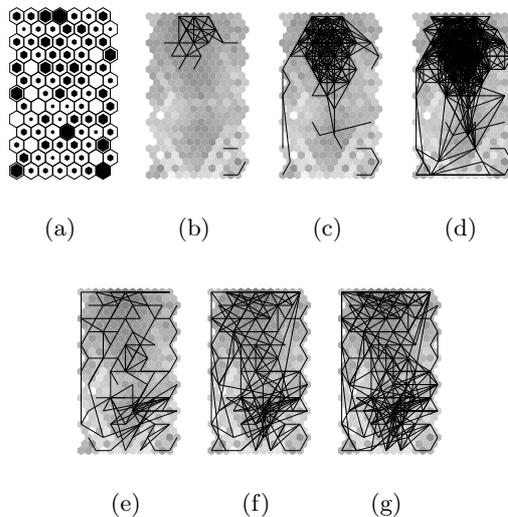


Fig. 1. Ionosphere 7×13 SOM: (a) hit histogram, Radius method: (b) $r = 1.0$, (c) $r = 2.0$, (d) $r = 3.0$, Nearest neighbors method: (e) $k = 1$, (f) $k = 2$, (g) $k = 3$

as edges, and this is generally an indication of good projection quality, since samples close in input space are close on the SOM as well in this case.

Finally, we can visualize the results on the two-dimensional map. This is performed by drawing lines connecting those map units for which edges in graph G^{SOM} exist. The resulting image reveals which areas of the SOM are densely populated, where interpolating units and regions lie, and where outliers are located. The interpretation of this visualization depends on the size of the map and the choice of parameter r or k , respectively.

4 Experimental results and properties of our method

In this section, we will see some experimental results with maps trained on the Ionosphere data set, which consists of 351 sample vectors of 34 nominal and numerical dimensions. There is a 35^{th} variable that serves as a class label and is omitted in SOM training, because of the SOM's unsupervised nature. The input space is densely populated in one region, while very sparsely in others. We will see that this property can be illustrated by our technique. Before training, the data is normalized to unit variance. The SOMs we investigate are trained with a two-dimensional lattice with hexagonal map units, one with a grid consisting of 7×13 nodes, and a larger one with 40×60 nodes, where data vectors are outnumbered by map units.

In Figure 1, the smaller version of the SOMs is visualized. Figure 1(a) shows the hit histogram that depicts the data vectors projected onto the map lattice. The U-Matrix is shown in the background of the other plots, indicating cluster borders with bright colors. Figures 1(b)–(d) depict the radius induced method

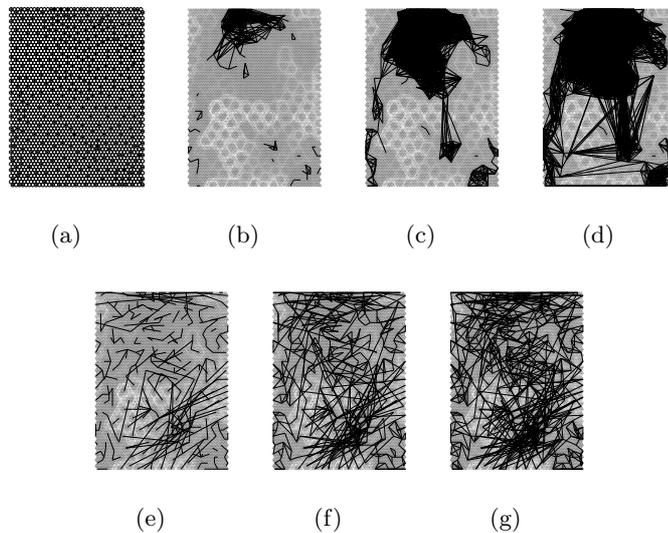


Fig. 2. Ionosphere 40×60 SOM: (a) hit histogram, Radius method: (b) $r = 1.0$, (c) $r = 2.0$, (d) $r = 3.0$, Nearest neighbors method: (e) $k = 1$, (f) $k = 2$, (g) $k = 3$

at different levels of $r = 1, 2, 3$. It can be clearly seen that the density of the data points is higher in the upper half of the map. This is not so obvious in either the U-Matrix and hit histogram visualizations. The nearest neighbors method is shown for $k = 1, 2, 3$ in Figures 1(e)–(g). Obviously, more lines are plotted. The emphasis here lies not on the identification of dense areas, but rather to single out regions where the mapping is distorted, as in the center of the bottom part of the map. Here, many lines point to distant areas of the map, which is an indication that the input space cannot be as easily clustered and projected as the model vectors in the upper half.

The larger version of the map is depicted in Figure 2 with the same parameter values as before. The graphs in input space is of course the same, only the mapping is different. It can be seen that the dense regions identified by the radius method is very similar to the smaller version. Outliers can be detected as those areas that do not show connections for high values of r , like the upper left corner. Due to the higher resolution, the lines can be distinguished more easily. The nearest neighbors method, depicted in Figures 2(e)–(g), again shows an evenly distributed picture of the connections. The region in the center of the bottom part seems distorted with lines running diagonally through it, although the radius method shows that it is not sparsely populated. Thus, topology violations due to the loss of dimensionality during the mapping are likely to have occurred.

Another interesting property is that the radius method tends to form more closed geometric figures like triangles, while these forms are star-shaped in the nearest neighbors method. This is due to the different relation, which is symmet-

ric in the radius case, and while not transitive in a mathematical sense, tends to group points together. The radius method is related to single linkage clustering [6]. When single linkage is performed, nodes are joined within a certain distance. Our radius method works similarly, hence, the graph structure with radius r reflects the clustering at level r in single linkage.

5 Conclusion

In this paper, we have seen a novel method for visualization of Self-Organizing Maps that is based on the set of data samples. This technique can easily be implemented for 2-dimensional map lattices. Two different definitions of proximity have been introduced, one that defines connectivity as a nearest neighbor relationship, while the second employs a density-based approach. Our experiments have shown that they are best applied in combination with other SOM visualization methods, like U-Matrix and hit histograms. We have found the nearest neighbor approach to be especially useful for maps with a large number of units compared to the number of data points. The radius method is more reliable with respect to outliers.

Acknowledgements

Part of this work was supported by the European Union in the IST 6. Framework Program, MUSCLE NoE on Multimedia Understanding through Semantics, Computation and Learning, contract 507752.

References

1. E. Oja J. Pakkanen, J. Iivarinen. The evolving tree—a novel self-organizing network for data analysis. *Neural Processing Letters*, 20(3):199–211, 2004.
2. T. Kohonen. *Self-Organizing Maps, 3rd edition*. Springer, 2001.
3. D. Merkl M. Dittenbach, A. Rauber. Uncovering hierarchical structure in data using the growing hierarchical self-organizing map. *Neurocomputing*, 48(1–4):199–216, 2002.
4. E. Pampalk, A. Rauber, and D. Merkl. Using smoothed data histograms for cluster visualization in self-organizing maps. In *Proc. Intl. Conf. on Artificial Neural Networks (ICANN'02)*, Madrid, Spain, 2002. Springer.
5. A. Rauber and D. Merkl. Automatic labeling of self-organizing maps: Making a treasure-map reveal its secrets. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'99)*, Beijing, China, 1999. Springer.
6. A. Rauber, E. Pampalk, and J. Paralic. Empirical evaluation of clustering algorithms. *Journal of Information and Organizational Sciences (JIOS)*, 24(2):195–209, 2000.
7. V. Sigillito, S. Wing, L. Hutton, and K. Baker. Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest*, 10:262–266, 1989.
8. A. Ultsch. Maps for the visualization of high-dimensional data spaces. In *Proc. Workshop on Self organizing Maps*, Kyushu, Japan, 2003.
9. J. Vesanto. *Data Exploration Process Based on the Self-Organizing Map*. PhD thesis, Helsinki University of Technology, 2002.