Graph projection techniques for Self-Organizing Maps

Georg Pölzlbauer[1], Andreas Rauber[1], Michael Dittenbach[2] *

1- Vienna University of Technology - Department of Software Technology
Favoritenstr. 9 – 11 / 188, A–1040 Wien, Austria
{poelzlbauer, rauber}@ifs.tuwien.ac.at

2- eCommerce Competence Center – ec3
Donau-City-Str. 1, A–1220 Wien, 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 two novel techniques that take the density of the data into account. Our methods define graphs resulting from nearest neighbor- and radius-based distance calculations in data space and show 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.

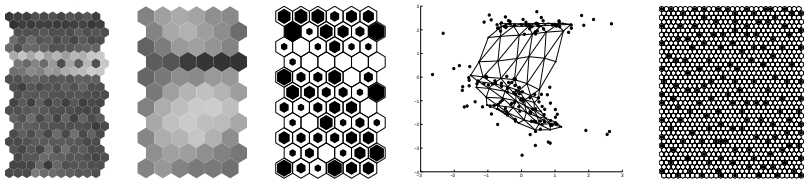
1 Introduction

The Self-Organizing Map [1] is a very popular artificial neural network algorithm based on unsupervised learning. It provides several beneficial properties, like vector quantization and topology preserving mapping from a high-dimensional input space to a usually 2-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 two novel graph-based visualization techniques, which provide an overview of the cluster structure and uncover topology violations of the mapping. The first of the methods visualizes a graph structure based on nearest neighbor calculations, and is especially useful for so-called "emergent maps" [2], 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 Fisher's well-known Iris data set. We show the differences between SOMs with different numbers of map units.

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

*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.



(a) (b) (c) (d) (e)

Fig. 1: Iris 6×11 SOM: (a) U-Matrix, (b) P-Matrix, (c) hit histogram, (d) PCA projection of data and codebook, (e) Iris 30×40 SOM: Hit histogram

proposed methods, followed by experimental results 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 novel methods. The most common ones are component planes and the U-Matrix [2]. For an in-depth discussion, see [3]. The emphasis of our paper lies on visualization techniques that take the distribution of the data set in input space into account. Most commonly, this is visualized as hit histograms, which display the number of data points projected to each map node. More advanced methods are the P-Matrix [4] or Smoothed Data Histograms [5] that visualize the density of the data in input space. Other techniques providing insight into the distribution of the data manifold are projection methods like PCA, but for higher-dimensional input spaces, the quality of the projection rapidly decreases. Some of the visualization methods are based on graph theoretic concepts like Voronoi Tesselation or Delaunay Graphs [6]. The methods we propose in this paper are also related to graph structures and will be discussed in Section 3.

In Figure 1, these visualizations are depicted for SOMs trained on the Iris data set with 6×11 and 30×40 map units, respectively. The feature dimensions have been normalized to unit variance. The U-Matrix, P-Matrix and hit histogram visualizations for the small map (Figures 1(a)–(c)) show a cluster boundary between the upper third and the lower two thirds of the map. In Figure 1(d), a PCA projection of both data samples and the map codebook is depicted. It reveals some very important characteristics of this SOM: The first three rows of the map are very dense in the vertical direction, which renders the lines of the map grid barely undistinguishable; slightly below, the interpolating region clearly divides the two parts of the data samples; and the lower two thirds of the map grid are highly skewed. However, the dimensionality of the data manifold is relatively low, and the first two axes of the plot explain more than 95% of the variance, a quality of projection which is very unlikely to be observed for larger and higher dimensional data sets. For the large SOM trained on the Iris data, Figure 1(e) shows the hit histogram. The number of data sam-

ples mapped onto the units is either zero or one. The U-Matrix visualization for this map is shown in Figures 2(d)–(f) as background images. It reveals vague cluster borders between the individual data samples' locations.

Apart from visualization, topology preservation and ordering of the SOM is a domain with connections to our methods. For a comprehensive overview of these methods, see [7]. Of particular importance is the SOM Distortion Measure, which has been shown to be the energy function which the SOM optimizes in cases of a fixed kernel radius. The connection to our methods, especially the one that is based on radius calculations, will be discussed in Section 4.

3 Two Graph Projection Methods

In this section, we present two novel visualization methods based on k-nearest neighbor- and radius-induced graph structures. These can be applied to any SOM with 2-dimensional map lattice. Our methods define graph structures that are derived from the data manifold in input space. The projection of these graphs to the map is then visualized. The presented methods have been implemented based on the infrastructure provided by the SOM Toolbox[1].

The first method we propose is based on nearest neighbor calculation. For each data sample x, the set $N_k(x)$ of its k nearest neighbors of data points in input space is determined. Then, the best-matching units (BMUs) of sample x and its nearest neighbors $n_j \in N(x)$ are connected visually on the map lattice by drawing a line between these units. If the number of map units is lower than the number of data samples, it will often happen that x and some of its k nearest neighbors are projected to the same map unit, in this case no line is drawn, which hints at a good projection quality. After all lines have been plotted, a graph-like structure can be observed that provides deeper insight into the proximity of the map units' weigth vectors. The structure revealed by this method does not necessarily coincide with the neighborhood of the map lattice. Distant map nodes being connected are an indication of topology violation. The higher the value of k is chosen, the more lines are plotted. High values of k can be useful to identify clusters in multimodal distributions where homogeneous areas are fully connected, while cluster borders are not bridged by lines connecting them. This visualization technique is most useful for large maps where the number of map units is much higher than the number of data samples, because more map space leads to more granular connections.

The second method we propose is visualized in a similar way, but differs in the decision which data points are to be connected. For each data sample x the set S_{rad} of samples which lie within a sphere of radius rad with center x is computed, formally

$$S_{rad}(x) = \{v_j | d(x, v_j) < rad\} \tag{1}$$

where d is a suitable distance metric, usually Euclidean distance. The BMU of sample x is then connected with the projections of the data points in $S_{rad}(x)$. This set may be empty if the parameter rad is either too low, or if x is very

[1] SOM Toolbox for Matlab: http://www.cis.hut.fi/projects/somtoolbox

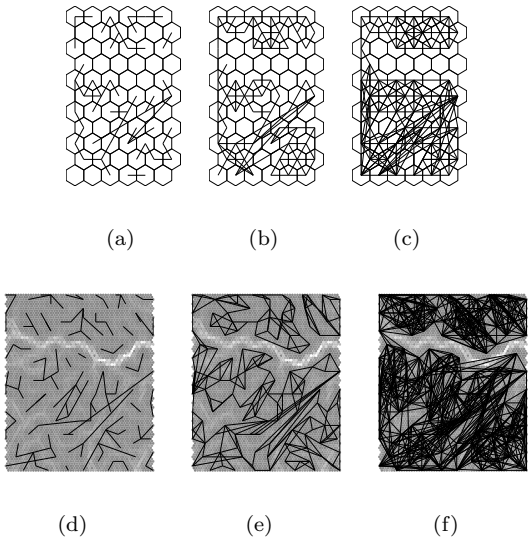


Fig. 2: Iris SOMs, map lattice connected with k-nearest neighbors method; (a)–(c) 6×11 map units, (d)–(f) 30×40 map units, visualized on top of U-Matrix; (a) $k = 1$, (b) $k = 3$, (c) $k = 10$, (d) $k = 1$, (e) $k = 3$, (f) $k = 10$

distant from the rest of the data points. The latter effect is actually desired, since this allows easy identification of outliers. Dense areas can be identified as regions where many lines point to.

The graph that is displayed on the map lattice is also related to the single linkage clustering method [8]. When single linkage is performed, nodes are joined within a certain distance. Our radius method works similarly, hence, the graph structure with radius *rad* reflects the clustering at level *rad* in single linkage. The radius method is also related to the P-Matrix visualization technique described in Section 2, but displays the density of the data manifold by connecting units with lines instead of color coding of the map lattice.

4 Examples

In this section, we will demonstrate the k-nearest neighbors and radius techniques and compare them to existing visualizations. In Figure 2, the k-nearest neighbors method is shown for different parameters k and SOMs of different sizes. For the small map, it can be seen that, especially for $k = 1$ and $k = 3$ as depicted in Figures 2(a)–(b), the upper right and lower left corners of the lower cluster of the map are connected diagonally. This is due to the fact that these regions are actually very close in input space, which has been shown earlier by the PCA projection in Figure 1(d). In Figure 2(c), no details can be distinguished anymore, but it shows the two clusters being clearly separated. The same method is shown for the larger SOM in Figures 2(d)–(f). The nearest

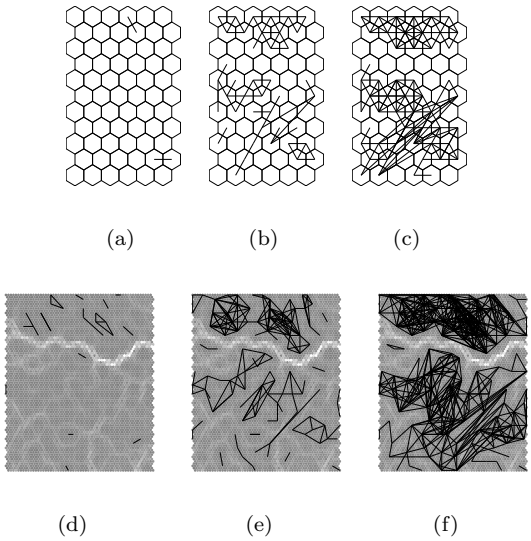


Fig. 3: Iris SOMs, map lattice connected with radius method; (a)–(c) 6×11 map units, (d)–(f) 30×40 map units, visualized on top of U-Matrix; (a) $rad = 0.2$, (b) $rad = 0.4$, (c) $rad = 0.6$, (d) $rad = 0.2$, (e) $rad = 0.4$, (f) $rad = 0.6$

neighbors of the data samples are of course the same as before, since the data manifold in input space does not change. However, the shape of the projected graph is different for individually trained maps of different sizes. For $k = 1$, small subgraphs can be observed indicating large proximity in input space. Again, for $k = 10$, the upper and lower regions are clearly separated.

The radius method is shown in Figure 3 for both map sizes with radii ranging from 0.2 to 0.6. Note that the feature dimensions are normalized to unit variance. There is a number of differences to the former technique: The actual density of the data manifold is taken into account with the top part of the map being more tightly connected than the lower two thirds. This is evident for both SOMs in Figures 3(b) and (e). Also, outliers can be identified as they are not connected to other map units. This is not so obvious in the nearest neighbors method, because even outliers have nearest neighbors in input space. This is also the reason for the connection between the two clusters in Figure 2(c), which is not present in Figure 3(c). Another difference is that the radius method graph tends to be composed of more closed geometric figures like triangles than the nearest neighbor graph, where star-like shapes are more common. This happens, because the nearest neighbor relation is not symmetric in a mathematical sense, i.e. if a's nearest neighbor is b, b's nearest neighbor is not necessarily a. Contrarily, the relation induced by the radius method is symmetric, since, if node b is within a sphere of radius rad around a, then a is also inside the sphere around b of the same radius. Thus, the nearest neighbor graph is directed, while the radius

graph is undirected.

Experiments comparing our method to topology and ordering measures for the SOM find that the SOM Distortion Measure, which is minimized during the training algorithm, is related to our radius method. The Distortion, which can be computed for each map unit, shows an inverse correlation to the density of lines pointing to the unit. However, this has been determined empirically, and further research is required to investigate the relation to this and other quality measures, such as the Topographic Function and the Topographic Product.

5 Conclusion

In this paper, we have presented two novel methods for visualization of Self-Organizing Maps that take data samples into account. These techniques can easily be implemented for 2-dimensional map lattices. The first method 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, P-Matrix, hit histograms, and projection methods like PCA. 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. Further research is being conducted in application to larger data sets and will be published in [9].

References

[1] T. Kohonen. *Self-Organizing Maps, 3rd edition.* Springer, 2001.

[2] A. Ultsch. Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series. In *Kohonen Maps*, pages 33–46. Elsevier Science, 1999.

[3] J. Vesanto. *Data Exploration Process Based on the Self-Organizing Map.* PhD thesis, Helsinki University of Technology, 2002.

[4] A. Ultsch. Maps for the visualization of high-dimensional data spaces. In *Proc. Workshop on Self organizing Maps*, Kyushu, Japan, 2003.

[5] E. Pampalk, A. Rauber, and D. Merkl. Using smoothed data histograms for cluster visualization in self-organizing maps. In *Proc. Intl. Conf. on Artifical Neural Networks (ICANN'02)*, Madrid, Spain, 2002. Springer.

[6] M. Aupetit. High-dimensional labeled data analysis with gabriel graphs. In *Proc. Intl. European Symp. on Artificial Neural Networks (ESANN'03)*, Bruges, Belgium, 2003. D-side publications.

[7] D. Polani. Measures for the organization of self-organizing maps. In Udo Seiffert and Lakhmi C. Jain, editors, *Self-Organizing Neural Networks: Recent Advances and Applications*, pages 13–44. Physica-Verlag, 2002.

[8] A. Rauber, E. Pampalk, and J. Paralic. Empirical evaluation of clustering algorithms. *Journal of Information and Organizational Sciences (JIOS)*, 24(2):195–209, 2000.

[9] G. Pölzlbauer, A. Rauber, and M. Dittenbach. Advanced visualization techniques for self-organizing maps with graph-based methods. In *Proc. International Symposium on Neural Networks (ISSN'05)*, Chongqing, China, 2005.