Visualising Clusters in Self-Organising Maps with Minimum Spanning Trees

Rudolf Mayer and Andreas Rauber

Institute of Software Technology & Interactive Systems
Vienna University of Technology, Austria
http://www.ifs.tuwien.ac.at/~mayer

Abstract. The Self-Organising Map (SOM) is a well-known neural-network model that has successfully been used as a data analysis tool in many different domains. The SOM provides a topology-preserving mapping from a high-dimensional input space to a lower-dimensional output space, a convenient interface to the data. However, the real power of this model can only be utilised with sophisticated visualisations that provide a powerful tool-set for exploring and understanding the characteristics of the underlying data. We thus present a novel visualisation technique that is able to illustrate the structure inherent in the data. The method builds on minimum spanning trees as a graph of similar data items, which is subsequently visualised on top of the SOM grid.

1 Introduction

The Self-Organising Map [2] (SOM) is a prominent tool for data analysis and mining tasks. It's main characteristic is a topology-preserving mapping (vector projection) from a high-dimensional input to a lower-dimensional output space. The output space is often a two-dimensional, rectangular lattice of nodes, which offers a convenient platform for plotting the topology of the high dimensional data for subsequent analysis tasks.

However, to fully exploit the potential of SOMs for data analysis and mining, it has to be combined with visualisations that additionally uncover the properties of the map and underlying data, e.g. cluster boundaries and densities. In this paper, we thus propose a novel technique that is able to visualise the similarity relationships. It is based on constructing a minimum spanning tree, which is then visualised on the SOM grid. This visualisation indicates, by connections across the map, which parts of the SOM are similar, and thus can uncover groups or clusters of related areas on the map. We compare this novel method with earlier visualisation techniques, and evaluate the benefits of the new method.

The remainder of this paper is organised as follows. Section 2 gives an overview on the SOM and its visualisations. Section 3 then introduces the concept of minimum spanning trees, and details how they can be applied to Self-Organising Maps. In Section 4, we present an experimental evaluation of the method on two benchmark datasets. Finally, Section 5 provides a conclusion.

K. Diamantaras, W. Duch, L.S. Iliadis (Eds.): ICANN 2010, Part II, LNCS 6353, pp. 426–431, 2010.

2 Self-Organising Map and Visualisations

The SOM performs both a vector quantisation, i.e. finding prototypical representatives in the data, similar to k-Means clustering, as well as a vector projection, i.e. a reduction of dimensionality. The SOM projection is, as faithfully as possible, preserving the topology of the input data, i.e. items located close to each other in input space will also be mapped close to each other on the map, while items distant in the input space will be mapped to different regions of the SOM.

The SOM consists of a grid of nodes (or units), each being associated with a model vector in input space. The grid (or lattice) is usually two-dimensional, due to the convenience of visualising two dimensions and the analogy to conventional maps. The nodes are commonly arranged in rectangular or hexagonal structures. The model vector of node i is denoted as $m_i = [m_{i1}, m_{i2}, ...m_{in}]^T \in \Re^n$, and is of the same dimensionality as the input vectors $x_i = [x_{i1}, x_{i2}, ...x_{in}]^T \in \Re^n$.

After initialisation of the model vectors, the map is trained to optimally describe the domain of observations. This process consists of a number of iterations of two steps. First, a vector x of the input patterns is randomly selected. The node with the model vector most similar to x is computed, and referred to as winner or best matching unit (BMU) c.

In the second step, the SOM is learning from the input sample to improve the mapping, i.e. some model vectors m_i of the SOM are adapted, by moving them towards x. The degree of this adaptation is influenced by two factors. The *learning rate* α determines how much a vector is adapted, and should be a time-decreasing function. The *neighbourhood function* h_{ci} is typically designed to be symmetric around the BMU, with a radius σ; its task is to impose a spatial structure on the amount of model vector adaptation.

As noted earlier, the SOM grid itself does not reveal much information about the relationships inherent to the data, besides their location on the map. A set of visualisation techniques uncovering more of the data and map structure has thus been developed. They are generally superimposed on the SOM, focusing on different aspects of the data.

Some methods rely solely on the **model-vectors**. Among them, *Component Planes* are projections of single dimensions of the model vectors $\mathbf{m}_i$. With increasing dimensionality, however, it becomes more difficult to perceive important information from the many illustrations. The *U-Matrix* [7] shows local cluster boundaries by depicting pair-wise distances of neighbouring model vectors. The Gradient Field [4] has some similarity with the U-Matrix, but applies smoothing over a broader neighbourhood. It uses a vector field style of representation, where each arrow points to its closest cluster centre.

A second category of techniques take into account the **data distribution** on the map. Labelling techniques plot the names and categories of data samples. Hit histograms show how many data samples are mapped to a unit (c.f. the textual markers in Figure 1, where units with no marker contain no data inputs, and are so-called interpolation units). More sophisticated methods include Smoothed Data Histograms [3], which show data densities by mapping each data sample to a number of map units. The P-Matrix [6] depicts the number of samples that lie

within a sphere of a certain radius around the model vectors. The density-graph method [5] shows the density of the dataset on the map. It also indicates clusters that are close to each other in the input space, but further apart on the map, i.e. topology violations. The graph is computed in input space, and consists of a set of edges connecting data samples which are 'close' to each other. Closeness can be fined based on a k-nearest neighbours scheme, where edges are created to the k closest peers of each data sample. A second approach connects samples with a pairwise distance below a threshold value r, i.e. the samples within the hyper-sphere of radius r. The parameters k and r thus determine the density of the resulting graph. To visualise the graph, all samples are projected onto the SOM grid, and connecting lines between two nodes are drawn if there is an edge between any of the vertices mapped on those two nodes.

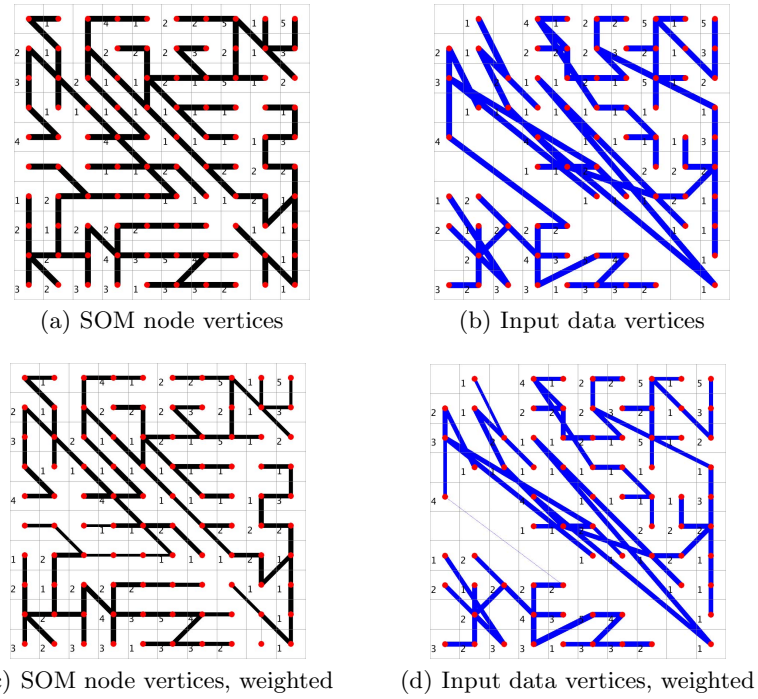
3 Minimum Spanning Tree Visualisation

A *spanning tree* is a sub-graph of a connected, undirected graph. More precisely, it is a tree, i.e. a graph without cycles, that connects all the vertices together. A graph can have several different spanning trees. By assigning a weight to each edge, one can compute an overall weight of a spanning tree. A minimum spanning tree is then a spanning tree with the minimum weight of all spanning trees.

The weights assigned to the edges often denote how *unfavourable* a connection is. A Minimum Spanning Trees then represents a sub-graph which indicates a favoured set of edges on the graph. Applied to SOMs, a Minimum Spanning Tree can be used to connect similar nodes with each other, and can thus visualise related nodes on the map. A graph on the SOM can be defined by using either the input data samples or the SOM nodes as vertices. The weights of the edges are computed by a distance metric between the vectors of the vertices, i.e. the input vectors or the model vectors, respectively.

When constructing the MST with the SOM nodes, it can be visualised by connecting lines between the two nodes that represent the vertices in each edge of the MST. When using the input data samples, first the best-matching-unit of each of the vertices is computed, and then again these two nodes are connected by a line. An illustration of these two visualisation technique is given in Figure 1(a) and 1(b). It can be observed that in both versions, sub-groups emerge.

The tree, by definition, fully connects all vertices, which, at first glance, makes it more difficult to spot the clusters. In Figure 1, this is especially the case when using the SOM nodes as vertices. Thus, a slight modification of the visualisation indicates the weights of the edges via the line thickness. We define the thickness as inverse proportional to the distance of the two nodes, i.e. to the weight of the edge, normalised by the maximum distance in the tree. Therefore, edges in the tree between very similar vertices are indicated by thick lines, while thin lines indicate a large distance. This approach is illustrated in Figure 1(c) and 1(d). It can be observed that the clusters are now visually much more separated than before.



(a) SOM node vertices (b) Input data vertices

(c) SOM node vertices, weighted (d) Input data vertices, weighted

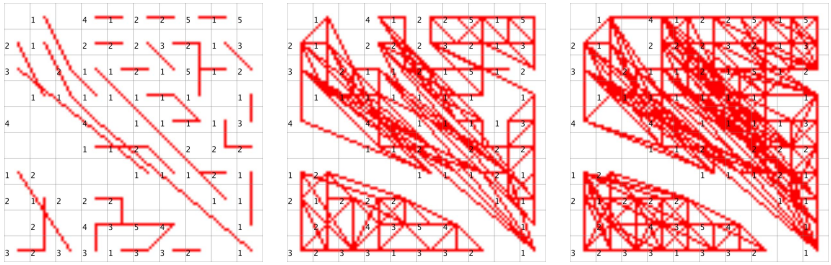
Fig. 1. MST visualisation, two sources of vertices (Iris dataset, 10x10 nodes)

4 Experimental Evaluation

We evaluate our visualisation method by applying it to two benchmark datasets. The first is the Iris dataset [1], a well-known reference dataset, describing three kinds of Iris flowers by four features: sepal length, sepal width, petal length, and petal width. The classes contain 50 samples each. One class is linearly separable from the remaining two, which in turn are not linearly separable from each other. This separation can be easily seen in the MST visualisation in Figure 1. Connections concentrate within the one separable class in the lower-left corner, and the two other classes in the rest of the map. Only one connection cuts across the boundary, as an implication of the full connectivity of the MST. Applying the line-thickness weighting clearly reveals the separation.

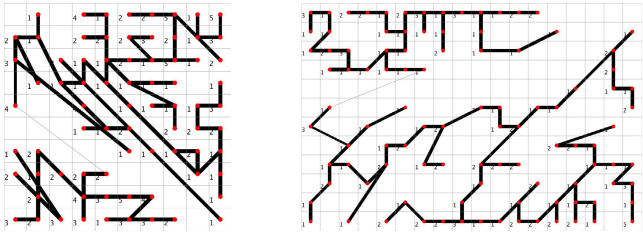
A comparison to the density graph is given in Figure 2. With a k value of 1, both visualisation reveal similar information. The MST visualisation seems clearer when it comes to within cluster relations, e.g. in the upper-right area. With a k of 1, the density graph doesn't indicate the relations between the many nodes that form small sub-graphs with one other node. With higher k, the display of local relations is traded for a better display of the cluster density.

In the display of the MST visualisation based on the model vectors of the SOM, the separation is not so apparent. While the MST considers all vertices, in a SOM, as mentioned earlier, there is often a number of nodes that do not



(a) k=1 (b) k=5 (c) k=10

Fig. 2. Density Neighbourhood Graph on Iris Dataset (10x10 nodes)



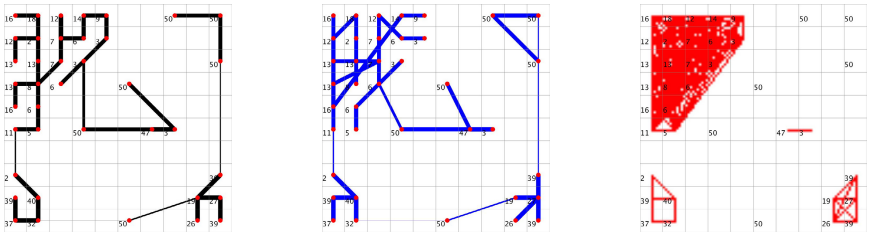
(a) Small map (10x10 nodes) (b) Large map (18x12 nodes)

Fig. 3. MST visualisation on Iris dataset, SOM node vertices, weighted lines, no interpolation nodes

hold any data samples, but serve as interpolation units along cluster boundaries. The model vectors of these nodes are located in an area in the input space that is either very sparsely populated, or not populated at all. To alleviate this problem, the user can select a mode that skips these interpolation nodes. Illustrations of the previously mentioned map, and a larger map, are given in Figure 3. Applying this filtering technique, the cluster structure is now more clearly visualised.

The second dataset is artificially created[1], to demonstrate how a data analysis method deals with clusters of different densities and shapes when these different characteristics are present in the same dataset. It consists of ten sub-datasets that are placed in a 10-dimensional space; some of the subsets live in spaces of lower dimensions. Figure 4 shows the visualisations of this dataset: (a) depicts the SOM nodes based MST visualisation, with weighted line thickness. As in this map the number of interpolation nodes is very high, only the variant skipping interpolation nodes yields a clear illustration. The MST on the input data is given in (b), compared to the neighbourhood density graph in (c). The two variants of the MST visualisation show very similar structures, with just minor differences. Compared to the density graph, the MST visualisation better depicts the relation between the subsets in the centre and upper-right corner.

[1] The dataset can be obtained at `http://www.ifs.tuwien.ac.at/dm/dataSets.html`



(a) SOM node vertices (b) Input data vertices (c) Density Graph, k=10

Fig. 4. Visualisations of the artificial dataset: MST (a), (b) and density graph (c)

5 Conclusions

We presented a visualisation technique for Self-Organising Maps based on Minimum Spanning Trees. The method is able to reveal groups of similar items, based on graphs built either of the input data, or the SOM nodes.

We evaluated the visualisation, and compared it to the density graph method, and found it to reveal similar information. The visualisation is not dependent on a specific user parameter, which is beneficial for novice users. The method operating on the SOM node vertices generally has lower computation time than the density graph method, as the number of nodes in a SOM is generally a magnitude smaller than the number of data samples. This variant can also be computed when the training data is not available. The visualisation can be superimposed on other techniques, such as the U-Matrix or Smoothed Data Histograms, which enables the display of various types of different information at once, without having to compare different figures.

References

1. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annual Eugenics 7(Part II), 179–188 (1936)
2. Kohonen, T.: Self-Organizing Maps. Springer, Heidelberg (2001)
3. Pampalk, E., Rauber, A., Merkl, D.: Using Smoothed Data Histograms for Cluster Visualization in Self-Organizing Maps. In: Proceedings of the International Conference on Artifical Neural Networks, Madrid, Spain. Springer, Heidelberg (2002)
4. Pölzlbauer, G., Dittenbach, M., Rauber, A.: Advanced visualization of self-organizing maps with vector fields. Neural Networks 19(6-7), 911–922 (2006)
5. Pölzlbauer, G., Rauber, A., Dittenbach, M.: Graph projection techniques for self-organizing maps. In: Proceedings of the European Symposium on Artificial Neural Networks, Bruges, Belgium. d-side publications (2005)
6. Ultsch, A.: Maps for the visualization of high-dimensional data spaces. In: Proceedings of the Workshop on Self-organizing Maps (2003)
7. Ultsch, A., Siemon, H.P.: Kohonen's Self-Organizing Feature Maps for Exploratory Data Analysis. In: Proceedings of the International Neural Network Conference. Kluwer Academic Press, Dordrecht (1990)