Graph Projection Techniques for Self-Organizing Maps

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

Abstract

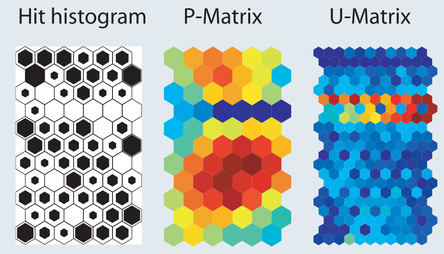
The Self-Organizing Map is a popular neural network model for data analysis, for which a wide variety of visualization techniques exists. We present two novel techniques that take the density of the data into account. Our methods define graphs resulting from nearest neighbor- and radius-based distance 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.

Related SOM Visualization Techniques

SOMs can be visualized in various ways:

- U-Matrix shows the distances of neighboring map units and hints at the clustering structure
- Hit histograms show the distribution of the data on the map
- P-Matrix shows the relative density

Examples: SOM trained on Iris data

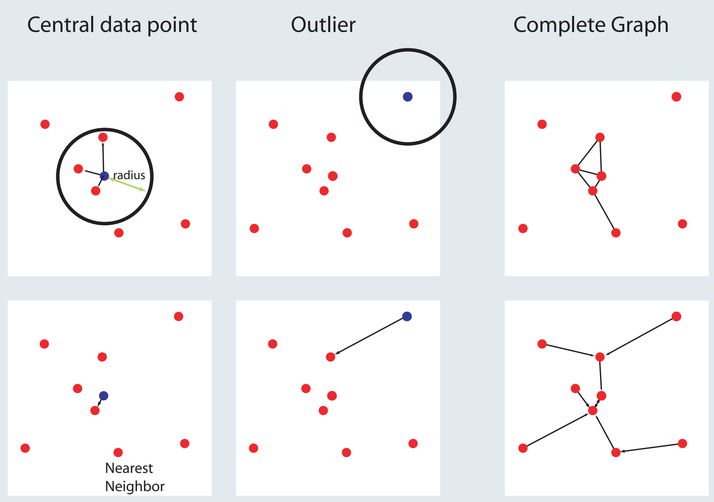


Methods for Graph Calculation (in Input Space)

Our method aims to visualize the density of data on top of the SOM lattice. A graph is computed in input space such that the most similar points are connected. This can be done in two different ways:

- connecting points that lie inside a hypersphere of a certain radius
- connecting each data point to its k Nearest Neighbors

Below, the two methods are compared and shown on a data point in the center of the data set, an outlying point, and the complete graph structure.

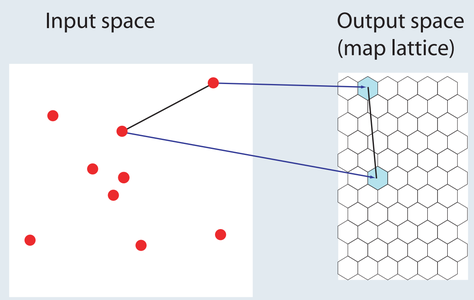


Graph Projection

The graphs are then projected onto the map lattice, connecting pairs of BMUs, giving insight into how the map is foldet onto the data set.

Mapping to output space:

- Like hit histogram,, data points are mapped to BMUs
- Graphs structure is also mapped, connecting BMUs

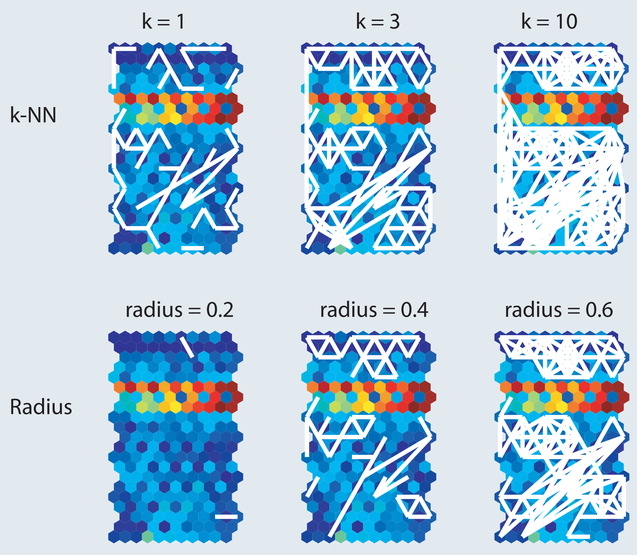


Properties of the visualization

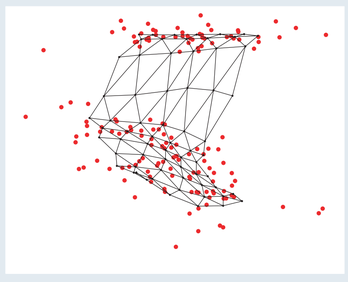
- The Radius method can be used to identify outliers
- Long lines hint at topology violations
- Short lines indicate that the topology of the data manifold is preserved
- The k-NN method is useful for large maps where the number of map units is high compared to the number of data points
- Shows which regions of the map are actually close in input space
- High parameter values for both methods show clusters

Example 1

6 x 11 SOM trained on Iris data, visualized on top of U-Matrix

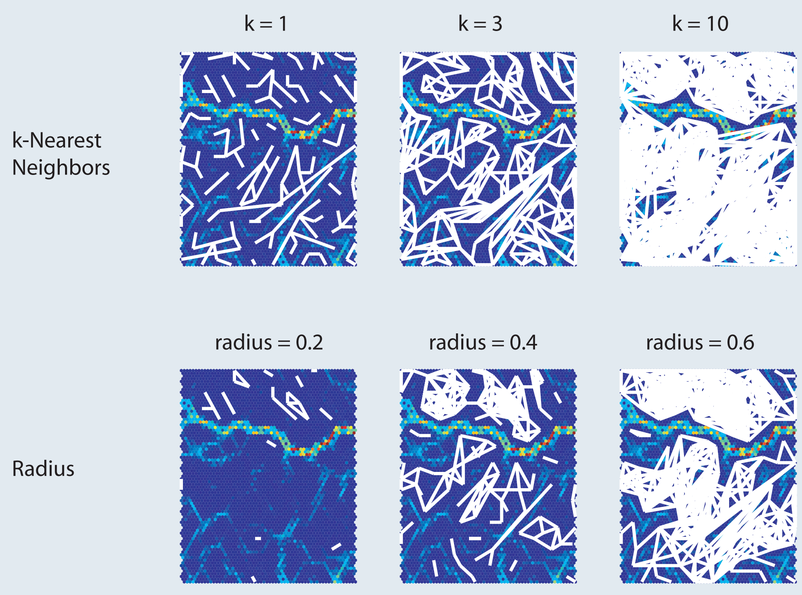


Here, a PCA projection of the data manifold and the codebook onto the two most important axes is depicted, explaining 95 % of the variance. The lower part shows how the map grid is skewed; this can be seen in the U-Matrix visualizations to the left, where the long diagonal lines indicate a strong proximity between the connected regions.



Example 2

Sparse 30 x 40 SOM trained on Iris data, visualized on top of U-Matrix



Practical Implications & Conclusion

- Best used together with other visualization techniques (U-Matrix, P-Matrix, hit histograms, Gradient Visualization)
- Applying the Radius method shows outliers as non-connected areas
- Use high parameter values to visualize clusters (e.g. 10-NN)
- k-NN method suitable for sparse maps
- Complexity is quadratic in number of data points, large datasets should be reduced for visualization purposes

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.

Author Affiliations

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

2. eCommerce Competence Center - ec3
   Donau-City-Str. 1
   1220 Vienna, Austria
   michael.dittenbach@ec3.at