Survey and Comparison of Quality Measures for Self-Organizing Maps

Georg Pölzlbauer

Department of Software Technology
Vienna University of Technology
Favoritenstr. 9-11/188, A-1040 Vienna, Austria
+43 1 58801 - 18813

poelzlbauer@ifs.tuwien.ac.at

Abstract. Self-Organizing Maps have a wide range of beneficial properties for data mining, like vector quantization and projection. Several measures exist that quantify the quality of either of these properties. The scope of this work is to describe and compare some of the most well-known measures. This is done by conducting a series of experiments for different map topologies with several well-known data sets. The measures are examined whether they are suited to determine hyperparameters like the optimal map size, how well the measure itself is suited to compare different maps, and if they allow comparison to other algorithms similar to the SOM (e.g. Sammons Mapping).

1 Introduction

The Self-Organizing Map [7] is a very popular artificial neural network (ANN) algorithm based on unsupervised learning. The SOM is used in various data mining tasks. It provides several very beneficial properties, like vector quantization and projection. To make the SOM comparable to other algorithms of these categories (e.g. projection methods like Sammon's mapping), and to measure the quality of the training algorithms, several measures have been proposed. This paper provides an overview of some of the most important ones:
The Quantization Error is described in Section 4.1. It is a well-known measure from the domain of clustering and vector quantization. In Section 4.2, the Topographic Product is described, which is one of the oldest quality measures of the SOM. It measures the input space representation abilities of a given map. In Section 4.3, the Topographic Error is outlined, which is the simplest of the functions that assess the topology preservation. Section 4.4 is dedicated to the Trustworthiness and Neighborhood Preservation measures which are based on ranking pair wise distances for both the input and output dimension. Finally, in Section 4.5 the SOM Distortion Measure is described. It

is particularly important since it has been shown to be the cost function the SOM algorithm optimizes.
The experiments described in the later chapters are conducted with the following datasets:
The Iris dataset is a popular multivariate data set which was introduced by R. A. Fisher as an example for discriminant analysis [15]. This dataset is included in many applications that deal with statistics and data analysis [1]. The Iris data reports on four characteristics of the iris flower, "sepal length", "sepal width", "petal length", and "petal width". The data set contains 50 samples for each of the three species ("Setosa", "Virginica" and "Versicolor"), with a total of 150 samples. Setosa iris flowers are clearly different from the other two species, while Virginica and Versicolor are harder to distinguish. Furthermore, the values for the petal variables (width and length) are highly correlated.
Another dataset that will be used in experiments in Chapter 5 is the epileptics dataset which was collected on two-week seizure counts for 59 epileptics. The dataset contains 236 samples and 8 variable dimensions[2].
The experiments described in this work have been performed with the help of the Mathworks' Matlab 6.5[3], a software tool and computer language for scientific computing. For the most part, the SOM toolbox for Matlab[4] [17] has been used.
The remainder of this paper is organized as follows:

Chapter 2 surveys related work. Chapter 3 provides a short description of the SOM algorithm and some of its most relevant features and advantages. In Chapter 4, the actual methods for quality estimation are described and explained. Also, the datasets used for benchmarking and testing of the measures are presented. Chapter 5 provides benchmarks and experiments based on these methods. In Chapter 6, the findings described in this document are summarized and an outlook and possible directions for future work are provided.

2 Related Work

Surveys for SOM quality measures are generally rare in literature. Earlier approaches have been mainly concerned with finding energy functions for the SOM [10, 19].
Usually, whenever a new method for quantifying the quality of the SOM is proposed, it is compared to existing ones, and a short overview is presented in the introductions [2, 8].

[1] Included, for example, in the SOM Toolbox [17]
[2] Included, for example, in the statistical program R: http://www.r-project.org
[3] http://www.mathworks.com
[4] http://www.cis.hut.fi/projects/somtoolbox

For general topographic mapping methods, a wide range of energy functions and quality measures exist, see for example [9]. These are, however, not specifically designed for assessment of SOM quality.
Several overviews exist for comparable clustering algorithms [20]. These usually also include sections on cluster quality measures. Another paper [21] deals exclusively with cluster validity.

3 The Self-Organizing Map

In this chapter, the SOM Training algorithm will be outlined in Section 3.1, along with its most important properties in Section 3.2. These are the properties that will later be quantified by the quality measures, each concerning a specific aspect of SOM features.

3.1 SOM Training Algorithm

The Self-Organizing Map has been originally developed by Teuvo Kohonen [7] and is mostly used for the visualization of nonlinear relations of multidimensional data. It has been subject to extensive research and has applications ranging from full text and financial data analysis, pattern recognition, image analysis, process monitoring and control to fault diagnosis; for a comprehensive list of references, see for example [11].
The SOM consists of a two-dimensional lattice that contains a number of neurons. These neurons are usually arranged in a rectangular or hexagonal way. The position of the units in the grid, especially the distances between them and the neighborhood relations, are very important for the learning algorithm. A prototype vector (also "model" or "codebook" vector) is associated with each neuron, which is a vector of the same dimension as the input data set. This prototype vector approximates a subset of the sample vectors. The dimension of the sample is called input dimension, and is usually larger than 2, the dimension of the lattice, which is called output dimension.
For training and visualization purposes, the sample vectors are assigned to the most similar prototype vector, or best-matching unit (BMU), formally

$$c(x) = \arg\min_i \{\| x - m_i(t) \|\} \quad (1)$$

where m_i are the prototype vectors, and x is the sample vector for which the BMU is determined. The learning process itself gradually adapts the model vectors to match the samples and to reflect their internal properties as faithfully as possible, which means that input vectors which are relatively close in input space should be mapped to units that are relatively close on the lattice. To achieve this, the training algorithm updates the model vectors iteratively during a number of training steps t, where a sample x(t) is selected randomly, and then the BMU and its neighbors are updated as follows:

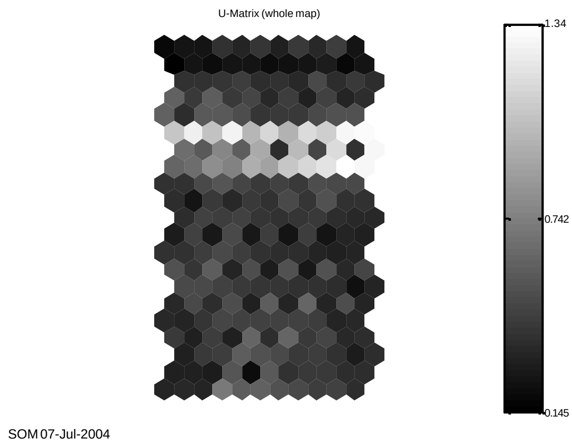

Fig. 1. U-Matrix [16] of the Iris SOM with 11x6 nodes. High values indicate gaps between the map nodes. Here, it can be seen that the upper 1st third of the map forms a coherent region, and, below the border region, another homogeneous area is located.

$$m_k(t+1) = m_k(t) + a(t)h_{c(x)k}(t)[x(t) - m_k(t)] \qquad (2)$$

where a(t) is the learning rate (which is decreasing monotonically over time) and $h_{ck}(t)$ is the neighborhood kernel. The neighborhood kernel determines the influence to the neighboring model vectors and its radius s(t) is also decreasing with time. Thus, the learning process is gradually shifting from an initial rough learning phase with a big influence area and fast-changing prototype vectors to a fine-tuning phase with small neighborhood radius and prototype vectors that adapt slowly to the samples.
The algorithm described above is referred to as "sequential training" or "basic SOM". Another important learning rule is called "Batch map", which is based on fixed point iteration, and is significantly faster in terms of computation time. At each step, the BMUs for all input samples are calculated at once, and the model vectors are updated as follows:

$$m_k(t+1) = \frac{\sum_{i=1}^{N} h_{c(x_i)k}(t)x_i}{\sum_{i=1}^{N} h_{c(x_i)k}(t)} \qquad (3)$$

with N the number of sample vectors.
Once the map has been trained, it is ready for post-processing and evaluation. Some of the figures in the later chapters refer to the SOM trained with the Iris dataset, for which the U-Matrix [16] visualization is shown in Figure 1.

3.2 Properties of the SOM

Data sets cannot be visualized on a sheet of paper or on a monitor if the dimensionality is higher than 2. Vector projection (VP) aims at reducing the input space dimensionality to a lower number of dimensions in the output space, and mapping vectors in input space to this lower dimensional space; the "lower dimensional space" is usually 2-dimensional for visualization on a monitor or for printing to paper. The VP mapping should occur in a way that the distances in input space are preserved as faithfully as possible, such that similar vectors in input space are mapped to positions close to each other in output space, and vectors that are distant in input space are mapped to different coordinates in output space. Most VP algorithms emphasize on the preservation of distances of vectors that are close to each other, while not necessarily preserving relatively large distances (as long as two samples that are far apart in input space are not placed next to each other in output space, it does not really matter just how far apart they are in output space). Other VP methods include Metric Multidimensional Scaling (MDS) [12], Principal Component Analysis (PCA), and Sammon's Nonlinear Mapping [13].

The task of finding a suitable subset that describes and represents a larger set of data vectors is called vector quantization (VQ) [14]. In other words, VQ aims at reducing the number of sample vectors or at substituting them with representative centroids. The resulting centroids do not necessarily have to be from the set of samples but can also be an approximation of the vectors assigned to them, for example their average. VQ is closely related to clustering. Obviously, the SOM performs VQ since the sample vectors are mapped to a (smaller) number of prototype vectors.

4 SOM QUALITY MEASURES

This chapter presents the quality measurement methods of the Self-Organizing Map. Measuring SOM Quality can be done in many ways. Two of the properties most of the measures try to evaluate are vector projection, which is sometimes refered to as "topology preservation", and vector quantization. Technically, there is a tradeoff between these two, increasing projection quality usually decreases the projection properties. Topology preservation is a property that is not easy to define and even harder to measure. Since usually a major reduction of dimensionality is performed, information is necessarily lost in the projection process. Generally, the proximity in input and output spaces is compared, often relying on ranking orders and nearest neighbor determination.

The methods can be further classified by what they can be used for. Some of the measures can be applied to select a hyperparameter. For example, the Topographic Product (Section 4.2) can be used to optimize the map size for any given dataset. Most of the quality measures, however, do not have this property, e.g. the Quantization Error decreases monotonically for increasing map sizes. Other measures can be used to benchmark a series of maps that have been trained from the same dataset. The Quantization Error and Distortion Measure are of this type. Another type of categorization can be made between methods applicable to the SOM only and methods that can compare the results to results from other algorithms. For example, the Topographic Error and the Trustworthiness both measure the projection quality, but only the latter

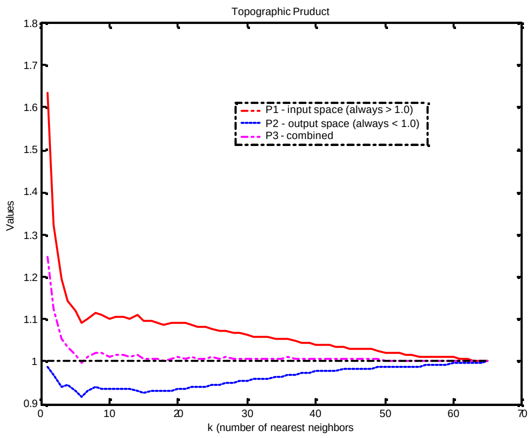

Fig. 2. Topographic Product;
SOM with 11x6 map units trained with Iris dataset;
Total Topographic Product P = 0.0113, which indicates that the map is slightly too small to represent the dataset

can be used to determine the quality of other topographic mappings, like e.g. Sammon's Nonlinear Mapping.

Note that the descriptions of the methods proposed in the remainder of this chapter do not contain exact definitions and formulas; for more in-depth information on the concepts' internals, please refer to the referenced publications.

4.1 Quantization Error

The Quantization Error (QE) is traditionally related to all forms of vector quantization and clustering algorithms. Thus, this measure completely disregards map topology and alignment. The quantization error is computed by determining the average distance of the sample vectors to the cluster centroids by which they are represented. In case of the SOM, the cluster centroids are the prototype vectors. Measuring the Quantization Errors can be extended such that it works with datasets containing missing values.

For any given dataset, the Quantization Error can be reduced by simply increasing the number of map nodes, because then the data samples are distributed more sparsely on the map.

Because of the tradeoff between vector quantization and projection properties of the SOM, changing the training process such that the QE is lowered usually leads to distortion of the map's topology.

4.2 Topographic Product

The Topographic Product [4] is one of the oldest measures that quantify the topology preservation properties of the Self-Organizing Map. The result of the computation of the Topographic Product indicates whether the size of the map is appropriate to fit onto the dataset.

For computation of the Topographic Product, only the map's codebook is regarded. This algorithm relies heavily on the comparison of ranking orders in input and output

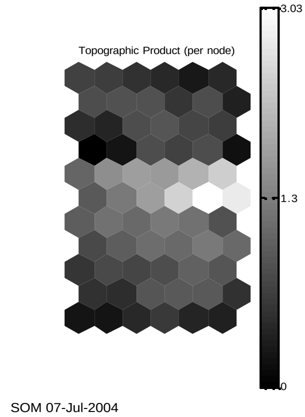

Fig. 3. Topographic Product per node;
Note that the distortion here lies two lines below the border identified by the U-Matrix visualization

space. First, a parameter k is selected that governs up to which rank the calculation is performed (e.g. for k=3, the 3 nearest neighbors are investigated). This is computed as two ratios, P1 and P2, where P1 measures the distortion in input space, and P2 in output space; if the ranking is perfectly equal in both spaces, P1 and P2 are equal to 1. Otherwise, P1 is greater than 1, and P2 is between 0 and 1. The value P3 combines these two measures into a single measure by calculating the geometric mean of P1 and P2. However, the measures P1, P2 and P3 are functions of k and can be evaluated for single map units only; therefore, a combined single value for all map units is obtained by summing up all values of P3, resulting in the Topographic Product P. This value can then be interpreted easily: If $P < 0$, the map is too small (i.e. has too few map nodes); for $P > 0$, the map is too big for the data space it represents.
An advantage of the Topographic Product is that the value P can be used to summarize the quality of the topology preservation by a single number, while the actual causes can be examined by plotting curves for P1, P2 and P3, or by visualizing the errors per map unit on a SOM lattice. An example for the values for a map trained with the Iris dataset is given in Figures 2 and 3. The Topographic Product can be computed for maps trained with datasets that contain missing values, since the codebook itself never contains any missing values.
It was shown by Villmann et al. that the Topographic Error presents reliable results only for nearly linear datasets. For further discussion see [18].

4.3 Topographic Error

The Topographic Error is the most simple of the topology preservation measures. Other than the Topographic Product, the dataset is required to determine the Topographic Error. This computation is performed as follows: For all data samples, the respective best and second-best matching units are determined. If these are not adjacent on the map lattice, this is considered an error. The total error is then normalized to a range form 0 to 1, where 0 means perfect topology preservation.

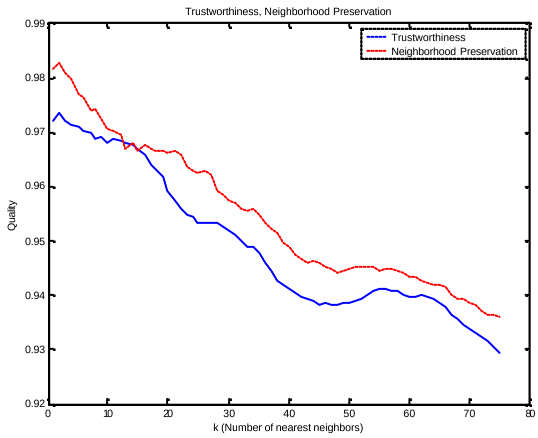

Fig. 4. Trustworthiness and Neighborhood Preservation SOM with 11x6 map units trained with Iris dataset

Usually, a single value is returned that quantifies this property. It is however possible to decompose the Topographic Error such that they can be visualized on a map lattice. This can be done, for example, by raising the error for a unit every time it is selected as BMU by a data sample, and the 2nd BMU is not adjacent in output space.
The Topographic Error can be computed for datasets containing missing values.

4.4 Trustworthiness and Neighborhood Preservation

In [5], two very similar topology preservation measures are proposed, the Trustworthiness and the Neighborhood Preservation. Of these, the Trustworthiness is slightly more important since it quantifies a property that is usually disregarded by topology quality measures, namely whether the projected data points which are actually visualized are close to each other in input space. Most other measures regard the other direction, whether the neighborhoods in input space are properly preserved in output space, which is not as important according to Venna and Kaski.
As with the Topographic Error, the dataset is required to determine the Trustworthiness. It is computed separately for every rank order k, similar to the computation of P1 of the Topographic Product. With a given k, the set of the k nearest neighbors are determined in output space (where the projected data points are considered, not the map units). Then, the k nearest neighbors in input space are determined. If the sets differ, an error counter is increased by the number of ranks in output space that exceeds the number k. In other words, whenever one of the neighbors on the map lattice is not one of the closest neighbors in the original data space, the error counter is increased. The Trustworthiness value is then computed as one minus the average of these errors, such that values range from 0 to 1, where 1 is perfect preservation.
There is, however, a number of problems with implementing the algorithm. Since the output space is discrete in case of the SOM, distances between data vectors is often exactly equal (e.g. all data samples that share the same BMU have distances of 0), resulting in many ex-aequo situations. This is very unlikely for rankings in input space, since it rarely occurs in (continuous) real world data that pairwise distances occur. For

this algorithm, this is a problem that has to be dealt with properly. The behavior of the experiments conducted in the original publication [5] could not be reproduced by the author. The strategies taken have been the following:

- Optimist approach: Consider only the nearest (in input space) of the set of equally distant neighbors on the lattice. This leads to a considerably higher Trustworthiness value.
- Pessimist approach: Use all the data samples from the nearest neighbor set, even if it exceeds the maximum number of elements k. This leads to problems with normalizing the final value to a range between 0 and 1 since the number of errors can possibly exceed the usual limit and is hard thus to determine. Also, a strong penalty for highly populated map units is obtained, which is not a property in question for a topology quality measure.
- Random approach: From the set of k-nearest neighbors on the map lattice, members are eliminated randomly until the set contains exactly k samples. This seems plausible, but introduces an element of randomness that is not desired into deterministic quality measures.
- Averaging approach: The errors resulting from the samples in the ex-aequo set are averaged and weighted accordingly. This approach seems most plausible and will be used for the rest of this paper.

When comparing the above strategies, the last one (averaging approach) is believed by the author to be the one described in the original publication. However, the original results could not be reproduced (the author's implementation differs significantly by 10 – 20 % from the original results).

Neighborhood Preservation is very similar to Trustworthiness, only the input and output space rankings are swapped. It measures rank orders in input space and compares it to the neighbors in output space, thus penalizing data points close in input space and far apart on the map lattice. Again, the Neighborhood Preservation is scaled to a value between 0 and 1, with 1 being the best possible value.

Both Trustworthiness and Neighborhood Preservation can be visualized on the map lattice, but experiments have shown that these plots are not very meaningful.

Figure 4 shows an example for Trustworthiness and Neighborhood Preservation curves accumulated over all units for different values of k.

Trustworthiness and Neighborhood Preservation can not be evaluated for datasets that contain missing values, since pair wise distances have to be calculated in input space. However, if further research is conducted in this direction, a solution can surely be found.

4.4 SOM Distortion

It has been shown that the SOM can not be considered the gradient of any energy function in the general case [10]. If the neighborhood kernel radius is made constant, then there exists an cost function the SOM optimizes, the SOM Distortion Measure [3]. This function can be used to compute an error value for the whole map. One of the major advantages is that the error can be decomposed in many ways:

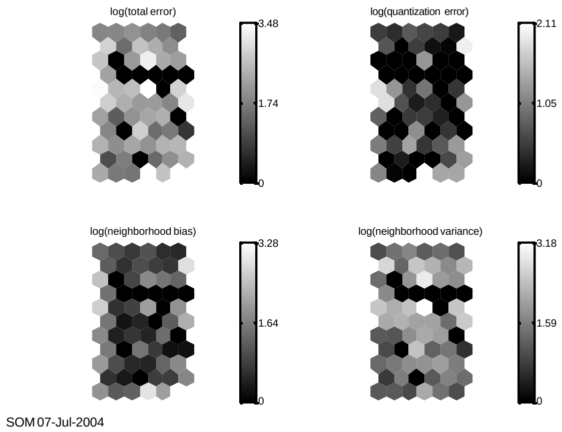

Fig. 5. SOM Distortion Measure (Log-Scaled)
The total error per unit is shown on the upper left, the upper right figure shows the Quantization Error, on the lower left the Neighborhood Bias is depicted, and the lower right figure shows the Neighborhood Variance

- per unit, such that the error can be localized on the map lattice
- per component (variable dimension)
- according to [1], the Distortion Measure can be decomposed into these parts:
    - Quantization Error
    - Neighborhood Bias
    - Neighborhood Variance

The decomposition per unit is visualized in Figure 5. Note that the nodes with values of 0 are not selected as BMU by any data samples and therefore do not cause any distortion (like the area on the right side of the upper 3rd part of the map).
For computation of the SOM Distortion Measure, the dataset is required in addition to the trained SOM. Missing values in the dataset are allowed. Since the Distortion Measure is the cost function of the SOM algorithm, it can be used to select the best fitting map from several maps trained with the same dataset.

5 EXPERIMENTS

The experiments conducted with the quality measures do not solely compare the findings for any given map but test how sensitive the measures are to changes in map size and data set size. The test settings and results are described in the remainder of this chapter.

5.1 Comparing maps of different sizes

In this section, the behavior of the different quality measures with maps of the same data set but a different number of map units is investigated. For this purpose, 3 SOMs have been trained for the Iris dataset, with the following sizes:

- small Iris map: 5x3 (15 nodes)
- medium Iris map: 11x6 (66 nodes)
- large Iris map: 23x11 (253 nodes)

and for the Epil dataset:

- small Epil map: 5x4 (20 nodes)
- medium Epil map: 10x8 (80 nodes)
- large Epil map: 22x15 (330 nodes)

The Quantization Error QE, as shown in Table 1, declines as the map becomes larger. This is not a surprising result. Thus, the Quantization Error can not be used to compare maps of different sizes.

Table 1. Quantization Errors

	Iris	**Epil**
small	0.6833	1.7325
medium	0.3925	1.1371
large	0.2092	0.6774

The Topographic Product declines as the map becomes larger. This can be seen in Table 2. The Topographic Product for the Medium Epil is exactly 0, so the map size fits perfectly according to this measure.

Table 2. Topographic Product

	Iris	**Epil**
small	0.0388	0.0097
medium	0.0113	0.0000
large	0.0045	-0.0128

The values for the Topographic Error can be seen in Table 3. For maps that are too large for the dataset, unnecessary folds occur and are penalized with a higher error value. The high values for the small maps are partly due to the fact that this measure is almost overly simplistic and suffers from the discrete nature of the output space. According to the author's test results, the Topographic Error is not reliable for small maps.

Table 3. Topographic Error

	Iris	Epil
small	0.0267	0.0551
medium	0.0067	0.0127
large	0.0467	0.0169

Results for Trustworthiness are summarized in Table 4. It has to be noted, that this function is defined for k ranging from 1 to half the number of data samples, so the smallest and largest values for k are depicted here. The Trustworthiness rises with increasing map size. For greater values of k, no such rule can be established.

Table 4. Trustworthiness

	Iris		Epil	
	k=1	k=74	k=1	k=117
small	0.898	0.931	0.886	0.793
medium	0.972	0.934	0.969	0.827
large	0.986	0.932	0.989	0.804

The Neighborhood Preservation (as summarized in Table 5) does not vary too significantly for increasing map sizes for small values of k, but diverges for larger k. These results, and the results of the Trustworthiness, have to be investigated more and extended by additional tests and experiments to come to a valid interpretation.

Table 5. Neighborhood Preservation

	Iris		Epil	
	k=1	k=74	k=1	k=117
small	0.976	0.973	0.952	0.869
medium	0.982	0.939	0.969	0.827
large	0.986	0.925	0.974	0.818

The results for the SOM Distortion Measure can be seen in Table 7. The error increases for larger maps. The SOM Distortion Measure is not suited to determine the right size of the map, but for comparing maps of the same size.

Table 6. SOM Distortion Measure

	Iris	Epil
small	7.7044	24.0505
medium	3.8800	17.4798
large	1.6411	9.6531

5.2 Comparing the distance measures for smaller data sets

In this section, the Iris data set is reduced to one half of its original size, removing every other sample in the dataset, so 75 data samples remain (25 of each species). The data is tested with the various measures against the maps trained in the previous section and the results compared. This test aims at identifying whether the quality measures are invariant on the factor of dataset size. Also, it can be of advantage to use a smaller, still representative subset of the data samples to reduce the computation time of the measuring algorithm (some of the methods are computationally very heavy).
For the Topographic Product, there is nothing to test at all since the computation does not rely on any data vectors. For Quantization Error and SOM Distortion Measure, the values are approximately the same for both settings. The values for the Topographic Error are so low in all instances, so that a comparison does not make sense.
The Trustworthiness and Neighborhood Preservation require additional attention. The values for k=1 are almost equal in all cases; for larger k, the resulting quality values change twice as much in the smaller dataset setting. The maximum value for the parameter k is also half the original k value (the max value is computed by dividing the size of the dataset by 2). In the author's opinion, this is very misleading. Since the two measures can only be computed for any given k, the resulting values should be comparable even for different dataset sizes. This could be fixed by normalizing the value k to a range between 0 and 1, for example.

6 CONCLUSION

In this paper, some of the major SOM quality measuring methods have been presented and compared. In Chapter 3, an introduction to the necessary prerequisites has been given. The experiments have tested empirically how well the measures are suited for different map sizes and how changes of the size of the dataset affect the outcome of the algorithms. The advantages and disadvantages of the methods are summarized in the following lines:

- Quantization Error: Measures vector quantization properties. It can not be used to compare maps of different sizes. It can be visualized on the map lattice, and can be applied to any vector quantization and clustering algorithm.

- Topographic Product: Only the map codebook is required for computation, not the dataset. It can be used to determine the best size of the map for the underlying dataset. Can be visualized on the map lattice. Can be applied to any projection method.
- Topographic Error: Unreliable for small maps, results in very low values for maps that do not contain overly high numbers of nodes. Can be visualized on the map lattice, but requires additional testing for plausibility and usefulness. Can not be applied to other methods than the SOM.
- Trustworthiness, Neighborhood Preservation: Provide a series of values that show the quality dependent on parameter k. To speed up the computation, the dataset can be reduced. There is no overall single result value. Trustworthiness can be visualized to show the per-unit error on the map lattice. Both methods can be applied to all vector projection algorithms.
- SOM Distortion Measure: Under special circumstances, this is the cost function of the SOM, and thus this measure has several advantageous properties. It can be decomposed in various ways and can be visualized on the map lattice. Can not be used to compare maps of different sizes, but can be used to benchmark maps of the same size. Can not be applied to other algorithms than the SOM.

There exist further methods such as the measure proposed in [8] that combines Topographic Error and Quantization Error, which can also be used to determine the best suited map size; and the Topographic Function [6].

For the Trustworthiness and Neighborhood Preservation methods, additional testing and evaluations should be conducted, since the results obtained in this paper do not come to a definite conclusion on the properties of these algorithms.

REFERENCES

[1] Vesanto, J., Sulkava, M., and Hollmen, J. On the Decomposition of the Self-Organizing Map Distortion Measure. *Proceedings of the Workshop on Self-Organizing Maps (WSOM'03)*, pp 11-16, Hibikino, Kitakyushu, Japan, September 2003

[2] Kiviluoto, J. Topology preservation in selforganizing maps. *In Proceedings of the International Conference on Neural Networks (ICNN'96)*, 1:294–299, Piscataway, New Jersey, USA, IEEE Neural Networks Council, June 1996.

[3] Lampinen, O. Clustering Properties of Hierarchical Self-Organizing Maps. *Journal of Mathematical Imaging and Vision*, 2(2-3):261–272, November 1992

[4] Bauer, H. U., Pawelzik, K. R. Quantifying the neighborhood preservation of selforganizing feature maps. *IEEE Transactions on Neural Networks*, 3(4):570–579, July 1992

[5] Venna, J., Kaski, S. Neighborhood preservation in nonlinear projection methods: An experimental study. *Lecture Notes in Computer Science*, 2130:485–491, 2001

[6] Villmann, T., Der, R., Herrmann, M., Martinetz, T. M. Topology preservation in self-organizing feature maps: exact denition and measurement. *IEEE Transactions on Neural Networks*, 8(2):256–266, 1997

[7] Kohonen, T. Self-Organizing Maps. *Volume 30 of Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, 3rd edition 2001

[8] Kaski, S., Lagus, K. Comparing Self-Organizing Maps. *Proceedings of International Conference on Artificial Neural Networks (ICANN) 96*, 809-814, 1996

[9] Goodhill, J., Sejnowski, T: Quantifying neighbourhood preservation in topographic mappings. *Technical Report INC-9505, Institute for Neural Computation*, November 1995

[10] Erwin, E., Obermayer, K., Schulten, K. Self-Organizing Maps: Ordering, convergence properties and energy functions. *Biological Cybernetics*, 67(1):47-55, 1992

[11] Oja, M., Kaski, S., Kohonen, T., Bibliography of Self-Organizing Map (SOM) Papers: 1998-2001 Addendum. *Neural Computing Surveys*, 3: 1-156, 2003

[12] Richardson, M. *Psychological Bulletin*. 35: 659-660, 1938

[13] Sammon, J. A Non-Linear Mapping for Data Structure Analysis. *IEEE Transactions on Computers*, C-18(5): 401-409, May 1969

[14] Gray, R. Vector Quantization. *IEEE ASSP Magazine*, 4-29, April 1984

[15] Fisher, R. The use of multiple measurements in taxonomic problems. *The Annals of Eugenics*, 7: 179-188, 1936

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

[17] Vesanto, J., Himberg, J., Alhoniemi, E., Parhankangas, J. Self-Organizing Map in Matlab: the SOM Toolbox. *Proceedings of the Matlab DSP Conference 1999*, 35-40, Espoo, Finland, November 1999

[18] Villmann, T., Der, R., Martinez, T. A new quantitative measure of topology preservation in Kohonen's feature maps, *Proceedings of the IEEE International Conference on Neural Networks 94, Orlando, Florida, USA*, 645-648, June 1994

[19] Heskes, T. Energy functions for Self-Organizing Maps. *Proceedings of Workshop on Self-Organizing Maps (WSOM 99)*, 303-315, 1999

[20] Jain, A., Murty, M., Flynn, P. Data Clustering: A Review. *ACM Computing Surveys* 31(3): 264-323, 1999

[21] Halkidi, M., Batistakis, Y., Vazirgiannis, M. On Clustering Validation Techniques. *Intelligent Information Systems Journal*, Kluwer Pulishers, 17(2-3): 107-145, 2001