On the Use of Self-organizing Maps for Clustering and Visualization

Arthur Flexer
The Austrian Research Institute for Artificial Intelligence
Schottengasse 3, A-1010 Vienna, Austria
arthur@ai.univie.ac.at

Abstract

We show that the number of output units used in a self-organizing map (SOM) influences its applicability for either clustering or visualization. By reviewing the appropriate literature and theory and own empirical results, we demonstrate that SOMs can be used for clustering or visualization separately, for simultaneous clustering and visualization, and even for clustering via visualization. For all these different kinds of application, SOM is compared to other statistical approaches. This will show SOM to be a flexible tool which can be used for various forms of explorative data analysis but it will also be made obvious that this flexibility comes with a price in terms of impaired performance.

Keywords: Self-organizing map, Clustering, Visualization, Multi-dimensional scaling

1 Introduction

Self-organizing maps (SOM) [14] are a very popular tool used for a range of different purposes including clustering and visualization of high dimensional data spaces. Although there is vast literature available concerning SOMs, an extensive survey [15] contains about 2000 entries, it is still far from clear when and how to apply SOMs for either clustering or visualization or even how these two purposes and goals relate to each other. In a comprehensive monograph [15] SOM is said to "project and visualize high-dimensional data spaces". The fact that there is a relation to clustering and visualization techniques is also well known, see e.g. [1], [10], [15], [4] and [24]. Theoretical analysis of SOM concentrates on issues *within* the method (e.g. convergence) rather than commenting on how and for what SOM should actually be used (see [7] for a survey of results).

However, there is also a considerable amount of criticism formulated both in terms of empirical and theoretical comparison. In [1] as well as [30] SOM

is compared to various clustering algorithms on artificial data. In [2] SOM is compared to principal component analysis and Sammon mapping on a series of artificial and real world data sets. In [10] SOM is compared to a combined method of vector quantization plus Sammon mapping of the codebook using multivariate normal data. Most of these empirical studies show SOM to perform equal or worse than the statistical approaches. There also exist two alternative re-formulations of the original idea of SOMs in more principled probabilistic frameworks ([4] and [24]). In [4] SOM is criticized for not defining a density model, for not optimizing an objective error function and for the lack of a guaranteed convergence property.

Albeit the wealth of work which has been done using and analysing SOMs and even although considerable amounts of criticism have already been formulated, what is still missing are some constructive guidelines as to clarify when and how to use SOMs for either clustering and visualization and how these notions relate to each other in the context of SOMs. This is exactly what this paper tries to achieve by showing that the number of output units used in a SOM influences its applicability for either clustering or visualization. Appropriate literature and theory will be reviewed and own empirical results will be presented which compare SOM to other statistical approaches. The usage of SOM in the two data analysis software tools CLEMENTINE and WEBSOM will also be discussed.

2 SOM for Clustering

According to a standard text book on pattern recognition [22] "Clustering algorithms are methods to divide a set of n observations into g groups so that members of the same group are more alike than members of different groups...the groups are called clusters". Since the main point about a cluster solution is that members of the same group are more alike than members of different groups, one has to have a means of measuring the likeness of such members. The most obvious measure of similarity (or dissimilarity) between two members is the distance between them. To state things more formally, let us say a clustering algorithm or vector quantizer (VQ) [1] is a mapping, q, that assigns to each of n input vectors x a reproduction (codebook) vector $\hat{x} = q(x)$ drawn from a finite reproduction alphabet $\hat{A} = \{\hat{x}_i, i = 1, \ldots, N\}$. Demanded is a VQ that produces a mapping q for which the expected distortion caused by reproducing the input vectors x by codebook vectors $q(x)$ is at least locally minimal. The expected distortion is usually estimated by using the average distortion D, where the most common distortion measure is the squared-error distortion (i.e. Euclidean

[1] What is referred to as "Clustering" in, amongst others, psychometric literature is known as "Vector Quantization" within the signal processing community. In this article both terms are used as synonyms, without further distinction.

distance) d:

$$D = \frac{1}{n} \sum_{j=0}^{n-1} d(x_j, q(x_j)) \tag{1}$$

$$d(x, \hat{x}) = \sum_{o=0}^{k-1} \mid x_o - \hat{x}_o \mid^2 \tag{2}$$

The average distortion D measures the total squared error of representing n samples $x_0, \ldots, x_{n-1}$ by the N codebook vectors (cluster centers) $\hat{x}_1, \ldots, \hat{x}_N$. Both x and $\hat{x}$ are of dimension k. D is small if all the distances between cluster members and cluster centers within each cluster are small.

A classical technique to achieve such a clustering is the K-means approach developed in the cluster analysis literature (starting from [19]). Closely related to SOM is online K-means clustering (oKMC) consisting of the following steps:

1. Initialization: Given N = number of codebook vectors, k = dimensionality of the vectors, n = number of input vectors, a training sequence $\{x_j; j = 0, \ldots, n-1\}$, an initial set $\hat{A}_0$ of N codebook vectors $\hat{x}$ and a discrete-time coordinate $t = 0 \ldots, n-1$.

2. Given $\hat{A}_t = \{\hat{x}_i; i = 1, \ldots, N\}$, find the minimum distortion partition $P(\hat{A}_t) = \{S_i; i = 1, \ldots, N\}$. Compute $d(x_t, \hat{x}_i)$ for $i = 1, \ldots, N$. If $d(x_t, \hat{x}_i) \leq (x_t, \hat{x}_l)$ for all l, then $x_t \in S_i$.

3. Update the codebook vector with the minimum distortion

$$\hat{x}_{(t+1)}(S_i) = \hat{x}_{(t)}(S_i) + \alpha[x_{(t)} - \hat{x}_{(t)}(S_i)] \tag{3}$$

   where α is a learning parameter to be defined by the user. Define $\hat{A}_{t+1} = \hat{x}(P(\hat{A}_t))$, replace t by $t+1$, if $t = n-1$, halt. Else go to step 2.

The only difference between the SOM-algorithm and oKMC is the fact that the codebook vectors are the weight vectors of SOM's output units which are ordered either on a line or on a planar grid (i.e. in a one or two dimensional output space). The iterative procedure is the same as with oKMC where Equ. 3 is replaced by

$$\hat{x}_{(t+1)}(S_i) = \hat{x}_{(t)}(S_i) + h[x_{(t)} - \hat{x}_{(t)}(S_i)] \tag{4}$$

and this update is not only computed for the $\hat{x}_i$ that gives minimum distortion, but also for all the codebook vectors which are in the neighbourhood of this $\hat{x}_i$ on the line or planar grid. The degree of neighbourhood and amount of codebook vectors which are updated together with the $\hat{x}_i$

that gives minimum distortion is expressed by h, a function that decreases both with distance on the line or planar grid and with time. The neighbourhood term h also includes an additional learning parameter α. Since codebook vectors in close neighbourhood in the one or two dimensional output space are always updated together, the result of the SOM-algorithm is a set of topologically ordered codebook vectors, i.e. codebook vectors which distance is small in the low dimensional output space are also close to each other in the input space. If the degree of neighbourhood is decreased to zero, the SOM-algorithm becomes essentially equal to the oKMC-algorithm. Since clustering problems in general are plagued by a large number of local minimas, the solutions obtained via SOM and oKMC might nevertheless be different due to different trajectories through the search space. Whereas local convergence is guaranteed for oKMC (at least for decreasing α, [5]), no general proof for the convergence of SOM with nonzero neighbourhood is known. In [15] it is noted that the last steps of the SOM algorithm should be computed with zero neighbourhood in order to guarantee "the most accurate density approximation of the input samples".

One of the main problems in clustering data is to decide for the correct number of clusters (i.e. codebook vectors). Clearly N, the number of cluster centers or output units, should be equal g, the number of clusters present in the data. In [8] it is argued that one should compute successive partitions of the data with an ever growing number of clusters N. If samples are really grouped into g compact, well separated clusters, one would expect to see any error function based on within or between cluster variance (the same obviously holds for average distortion) decrease rapidly until $N = g$. Such error functions should decrease much more slowly thereafter until they reach zero at $N = n$.

The two most comprehensive studies on SOM's clustering ability ([1] and [30]) use SOMs and cluster algorithms with N always set equal to g, the number of clusters known to be in the data. In [30] SOM is compared to five different cluster algorithms on 2580 artificial data sets. One-dimensional SOMs are being used with zero neighbourhood at the end of learning and consequently SOMs and K-means clustering perform equally well in terms of data points misclassified[2], both being better than the other hierarchical cluster methods.

In [1] SOM is compared to K-means clustering on 108 multivariate normal clustering problems but the SOM neighbourhood is not decreased to zero at the end of learning. SOM performs significantly worse in terms of

[2]Although SOM is an unsupervised technique not built for classification, the number of points misclassified to a wrong cluster center *is* an appropriate and commonly used performance measure for cluster procedures if the true cluster structure is known. Given $N = g$, all members of one true cluster in the data space should be members of just *one* cluster in the obtained partition. All exchanges between clusters constitute data points misclassified.

data points misclassified since the additional neighbourhood term tends to pull the obtained cluster centers away from the true ones (the SOM cluster centers are pulled towards each other). In [15] this effect is described as two "opposing forces" where the weight vectors of the output units tend to describe the density function of the inputs and the local interactions between output units tend to preserve topology.

3 SOM for Simultaneous Clustering and Visualization

SOM is however more than just a technique to cluster data. It has the appealing property to do clustering *and* visualization at the same time by preserving the topological ordering of the input data reflected by an ordering of the codebook vectors in a one or two dimensional output space. Note that in order to use SOM for visualization *and* clustering at the same time it is again necessary that N, the number of output units, is equal g, the number of clusters in the data set.

Topology preserving algorithms aim at representing high dimensional data spaces in a low dimensional space while preserving as far as possible the structure of the data in the high dimensional data space. This is achieved by mapping "points in one space to points in another space such that nearby points map to nearby points (and sometimes in addition far-away points map to far-away-points)" [11].

Formally, a topology preserving algorithm is a transformation $\Phi : R^k \mapsto R^p$, that either preserves *similarities* or just *similarity orderings* of the points in the input space R^k when they are mapped into the output-space R^p. For most algorithms it is the case that both the number of input vectors $\mid x \in R^k \mid$ and the number of output vectors $\mid \dot{x} \in R^p \mid$ are equal to n. A transformation $\Phi : \dot{x} = \Phi(x)$, that preserves *similarities* poses the strongest possible constraint since $d(x_i, x_j) = \dot{d}(\dot{x}_i, \dot{x}_j)$ for all $x_i, x_j \in R^k$, all $\dot{x}_i, \dot{x}_j \in R^p$, $i, j = 0, \ldots, n-1$ and d ($\dot{d}$) being a measure of distance in R^k (R^p). Such a transformation is said to produce an *isometric* image. For a transformation $\Phi : \ddot{x} = \Phi(x)$ that preserves only *similarity orderings*, $d(x_1, x_2) \leq d(x_3, x_4) \Rightarrow \ddot{d}(\ddot{x}_1, \ddot{x}_2) \leq \ddot{d}(\ddot{x}_3, \ddot{x}_4)$ must hold for all $x_1, x_2, x_3, x_4 \in R^k$.

Techniques for finding such transformations Φ are, among others, various forms of *multidimensional scaling*[3] (MDS) like Sammon mapping [23], but also principal component analysis (PCA) (see e.g. [13]) or SOM. Sammon mapping is doing MDS by minimizing the following via steepest descent:

[3]Note that for MDS not the actual coordinates of the points in the input space but only their distances or the ordering of the latter are needed.

$$\frac{1}{\sum_{i=0}^{n-1}\sum_{j<i} d(x_i,x_j)} \sum_{i=0}^{n-1}\sum_{j<i} \frac{(d(x_i,x_j) - \dot{d}(\dot{x}_i,\dot{x}_j))^2}{d(x_i,x_j)} \tag{5}$$

where $\dot{d}(\dot{x}_i,\dot{x}_j)$ is the distance in the output space that corresponds to the distance $d(x_i,x_j)$ in the input space and n is the number of points to be mapped. Since SOM has been designed heuristically and not to find an extremum for a certain energy function[4], the theoretical connection to other MDS algorithms remains unclear. It should be noted that for SOM the number of output vectors $\mid \dot{x} \in R^p \mid$ is limited to N, the number of cluster centroids $\hat{x}$ and that the $\dot{x}$ are further restricted to lie on a planar grid. This restriction entails a discretization of the output-space R^p which allows only $\sum_{i=2}^{s} i_{,(s\geq 2)}$ different distances in an $s \times s$ planar grid instead of $\frac{N(N-1)}{2}$ different distances for $N = s \times s$ cluster centroids mapped via e.g. Sammon mapping.

In what we believe to be the only existing empirical study on SOM's ability of doing both clustering and visualization at the same time, we have compared SOM to a combined technique of online K-means clustering plus Sammon mapping of the cluster centroids [10]. Our new combined approach (abbreviated oKMC+) consists of simply finding the set of $\hat{A} = \{\hat{x}_i, i = 1,\ldots,N\}$ codebook vectors that give the minimum distortion partition $P(\hat{A}) = \{S_i; i = 1,\ldots,N\}$ via oKMC and then using the $\hat{x}_i$ as input vectors to Sammon mapping and thereby obtaining a two dimensional representation of the $\hat{x}_i$ via minimizing the term in Equ. 5. Contrary to SOM, this two dimensional representation is not restricted to any fixed form and the distances between the N mapped $\hat{x}_i$ directly correspond to those in the original higher dimension. In [24] a similar combined technique is proposed with the difference that clustering and visualization is achieved simultaneously and not one after the other.

The empirical comparison was done using multivariate normal distributions generated by a procedure which is standard for comparisons of cluster algorithms (see [21] and [1]). The marginal normal distributions gave internal cohesion of the clusters by warranting that more than 99% of the data lie within 3 standard deviations (σ). External isolation was defined as having the first dimension non-overlapping by truncating the normal distributions in the first dimension to $\pm 2\sigma$ and defining the cluster centroids to be 4.5σ apart. In all other dimensions the clusters were allowed to overlap by setting the distance per dimension between two centroids randomly to lie between $\pm 6\sigma$. The data was normalized to zero mean and unit variance in all dimensions. We produced 18 data sets with number of clusters being 4 or 9, and the number of dimensions being 4, 6 or 8. This yielded 3 data sets for every combination of "number of clusters" and "number of dimensions". For

[4] In [9] it is even shown that such an objective function cannot exist for SOM.

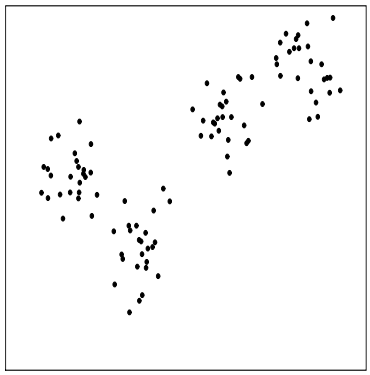

Figure 1: Four Gaussian clusters in two dimensions where the marginal distributions are allowed to overlap only in one dimension (y-axis). See Sec. 3 for details.

every data set we created 25 points for each cluster which gave sizes of 100 and 225 points for data sets comprised of 4 and 9 clusters. An illustrative example in only two dimensions is given Fig. 1.

Table 1: Results for comparing SOM to oKMC+. Mean performances for SOM and oKMC+ are printed in bold letters, see Sec. 3 for details.

algorithm	no. clusters	dimension	msqe	Rand	corr.
SOM	4	4	0.53	1.00	0.64
		6	1.53	0.91	0.72
		8	1.15	0.99	0.74
	9	4	0.33	0.97	0.48
		6	0.54	0.97	0.66
		8	0.81	0.96	0.74
	mean SOM		**0.81**	**0.97**	**0.67**
oKMC+	4	4	0.53	0.99	0.87
		6	1.06	0.99	0.87
		8	1.17	1.00	0.91
	9	4	0.29	0.98	0.89
		6	0.47	0.99	0.87
		8	0.56	0.98	0.86
	mean oKMC+		**0.68**	**0.99**	**0.88**

The empirical comparison was done using a 3 factorial experimental

design with 3 dependent variables.

Factor 1, Type of algorithm: The number of codebook vectors of both the SOM and the oKMC+ were set equal to the number of clusters known to be in the data. The SOMs were planar grids consisting of 2×2 (3×3) codebook vectors. During the first phase (1000 codebook updates) α was set to 0.05 and the radius of the neighbourhood to 2 (5). During the second phase (10000 codebook updates) α was set to 0.02 and the radius of the neighbourhood to 0 to guarantee the most accurate clustering [15]. The oKMC+ algorithm had the parameter α fixed to 0.02 and was trained using each data set 20 times during its clustering phase (2000 or 4500 codebook updates for data sets with 100 or 225 points). The minimization in Equ. 5 was stopped after 100 iterations. Both SOM and oKMC+ were run 10 times on each data set and only the best solutions, in terms of mean squared error, were used for further analysis.

Factor 2, Number of clusters was set to 4 and 9.

Factor 3, Number of dimensions was set to $4, 6, or 8$.

Dependent variable 1: mean squared error was computed using Equ. 1.

Dependent variable 2, Rand index (see [12]) is a measure of agreement between the true, known partition structure and the obtained clusters. Both the numerator and the denominator of the index reflect frequency counts. The numerator is the number of times a pair of data is either in the same or in different clusters in both known and obtained clusterings for all possible comparisons of data points. Since the denominator is the total number of all possible pairwise comparisons, an index value of 1.0 indicates an exact match of the clusterings.

Dependent variable 3, correlation is a measure of the topology preserving abilities of the algorithms. The Pearson correlation of the distances $d(x_1, x_2)$ in the input space and the distances $\dot{d}(\hat{x}_i, \hat{x}_j)$ in the output space for all possible pairwise comparisons of data points is computed. Note that for SOM the coordinates of the codebook vectors on the planar grid were used to compute the $\dot{d}$. An algorithm that preserves all distances in every neighbourhood would produce an *isometric* image and yield a value of 1.0 (see [2] for a discussion of measures of topology preservation).

A multiple analysis of variance (MANOVA) applied to the full-factorial $2 \times 2 \times 3$ design yielded the following significant effects at the .05 error level (see Tab. 1 for results):

The mean squared error is lower for oKMC+ than for SOM, it is lower for the 9-cluster problem than for the 4-cluster problem and is higher for higher dimensional data. There is also a combined effect of the number of clusters and dimensions on the mean squared error. The Rand index is higher for oKMC+ than for SOM, there is also a combined effect of the number of clusters and dimensions. The correlation index is higher for oKMC+ than for SOM. Since the main interest of this study is the effect of the type of algorithm on the dependent variables, the mean performances for SOM

and oKMC+ are printed in bold letters in the table. Note that the overall differences in the performances of the two algorithms are blurred by the significant effects of the other factors and that therefore the differences of the grand means across the type of algorithms appear rather small. Only by applying a MANOVA, effects of the factor 'type of algorithms' that are masked by additional effects of the other two factors 'number of clusters' and 'number of dimensions' could still be detected.

In contrast to a study [30] reported in Sec. 2, SOM performed worse in terms of mean squared error despite the use of zero neighbourhood at the end of training. SOM performed almost equally well as oKMC+ in recovering the structure of the clusters (measured via the Rand index). It seems to be clear that a solution can have a quite high Rand index and still show increased distortion measured via mean squared error. SOM performed significantly worse in preserving the topology, we obtained a correlation of 0.67 for SOM and of 0.88 for oKMC+. This is a direct implication of SOM's restriction to planar grids described above. Using a nonzero neighbourhood at the end of SOM training did not warrant any significant improvements for the topology preservation.

4 SOM for Visualization

Another possibility to apply SOM is to use it for visualization only thereby neglecting its clustering ability. It is then not necessary to try to set the number of output units equal to a presumed number of clusters in the data. It is possible and even common practice to apply SOM with numbers of output units N that are a multiple of the number of input vectors n available for training (see e.g. the "poverty map" example given in [15]). This means of course that SOMs employing numbers of codebook vectors which are comparable to or are even a multiple of the number of input vectors available can be used for visualization purposes only. If one uses more or even only the same amount of codebook vectors than input vectors during clustering, each codebook vector will become identical to one of the input vectors in the limit of learning. So every x_i is replaced with an identical $\hat{x_i}$, which does not make any sense in terms of clustering.

In [2] SOM is compared to principal component analysis and Sammon mapping on six artificial data sets with different numbers of points and dimensionality and different shapes of input distributions and on the Anderson IRIS data. The degree of preservation of the spatial ordering of the input data is measured via a Spearman rank correlation instead of Pearson correlation similar to our approach described above. The traditional techniques preserve the distances much more effectively than SOM, the performance of which decreases rapidly with increasing dimensionality of the input data.

We did an own study on visualization with SOM using the same 18 data

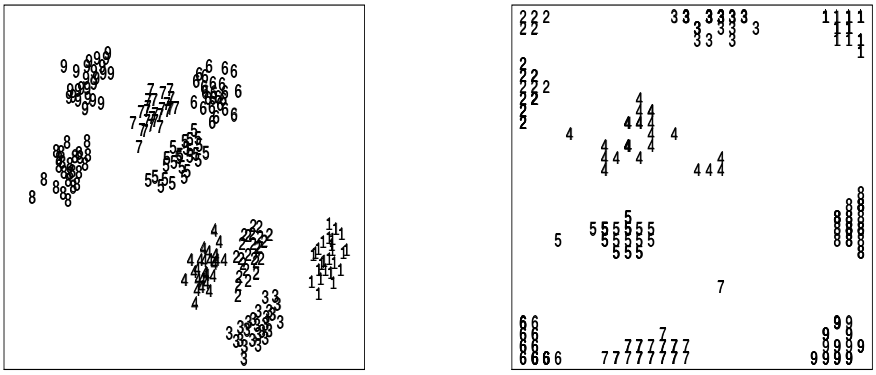

Figure 2: Output representations after mapping nine eight-dimensional clusters via Sammon mapping (left) and SOM (right). Numbers indicate true cluster membership.

sets described in Sec. 3. We computed SOMs consisting of 20×20 (for data sets consisting of 4 clusters and 100 points) or 30×30 (for 9 clusters and 225 points) codebook vectors for all 18 data sets which gave an average correlation of 0.77 between the distances d_i and $\dot{d}_i$. This is significantly worse at the .05 error level compared to the average correlation of 0.95 achieved by Sammon mapping applied to the input data directly. The discretization and the rigidity of SOM's output space are clearly visible if one compares output maps given in Fig. 2.

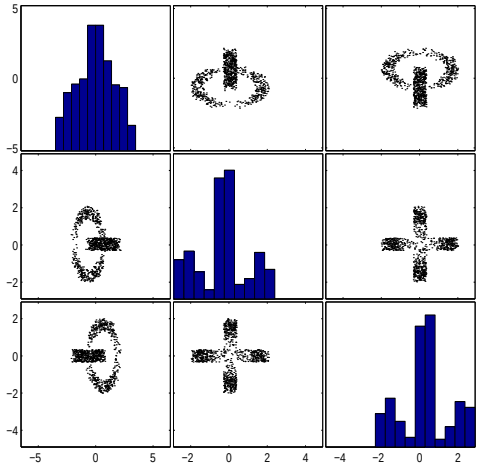

Figure 3: Scatter plot of two intertwined 3-dimensional rings.

Additionally, we also did an experiment using non-normal data. We

used the so-called chain-link problem [27] which consists of two intertwined 3-dimensional rings. One of the rings is extended into x-y direction and the other one into x-z direction. Each of the rings consists of 500 data points. A scatter plot of the data plus histograms of the marginal densities is given in Fig. 3. We computed a SOM consisting of 40×40 codebook vectors which gave a correlation of 0.64 between the distances d_i and $\dot{d}_i$. A Sammon mapping applied to the input data directly achieved a correlation of 0.82. Output maps for both methods are given in Fig. 4. Whereas the two rings are still visible in the case of Sammon mapping, no such structure can be seen in the SOM output map.

These results together with the previously described study [2] indicate that even using more output units than input vectors available does not help against the drawbacks of SOM's discretization of the output-space.

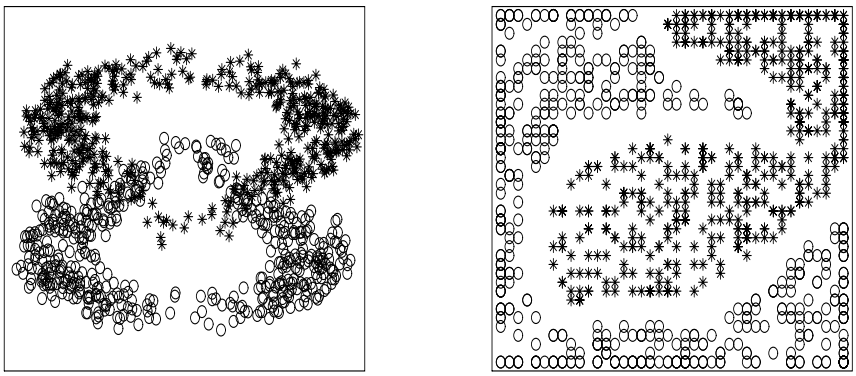

Figure 4: Output representations after mapping two intertwined 3-dimensional rings via Sammon mapping (left) and SOM (right). Data points for the two different rings are drawn as rings and stars.

5 SOM for Clustering via Visualization

Yet another possible application of SOM is to use it to cluster data via visualization. This is done by first visualizing the data via a SOM output map and then using ones own subjective judgement by just looking at the resulting output map and counting how many clusters one is able to see. Reviewing clustering studies employing SOM quickly shows that indeed SOMs are often used for this kind of clustering via visualization. There is even work on trying to augment cluster visibility in SOM output maps (see e.g. [26], [3], [28] and [29]).

It should be clear that for this type of application SOMs with large amounts of output units will be best suited. A higher number of output units should help against the negative effect of the discretization of SOM's output space. However, it has long been known within the clustering community that doing clustering via visualization bears some pitfalls. In [25] it is shown that there is a high probability that a researcher will conclude that a subset of points comprise one cluster, when in fact the points comprise two or more clusters. This is due to the reduction in dimensionality produced by the mapping to the output space which impairs the user's ability to detect clusters that existed in the space defined by the original variables. In [20] it is shown that even if researchers are asked to determine cluster membership from identical two-dimensional representations, their inter-rater reliability is on average as low as 0.77.

If one compares output maps obtained by SOM and Sammon mapping given in Fig. 2, it seems that whereas the 9 clusters are still clearly visible in the Sammon mapping picture this is not so clear in SOM's output map. Clusters 2 and 4 are no longer coherent and members of cluster 5 and 7 appear as outliers. The chain-link problem described in Sec. 3 is a good example for the possible loss of information entailed by mapping to a lower dimensionality. The fact that the two rings are intertwined but still separate cannot be depicted in only two dimensions. Whereas Sammon mapping maintains the ring-like structures, SOM is only able to depict two separate groups [5] of points but looses all structural information (see Fig. 4).

6 SOM in data analysis software tools

SOM is also used in two prominent data analysis software tools: it is one of the algorithms implemented in CLEMENTINE [6] and it is at the heart of WEBSOM, a system for automatic organization of large text document collections (see [17] and [16]).

Both data mining applications CLEMENTINE and WEBSOM use SOM for clustering via visualization. The CLEMENTINE user guide [6, p.8] states that SOMs "are a type of neural network that perform clustering" but does not advise the reader how such a clustering can be achieved. However, besides an "Expert Training Method" which requires the user herself to choose the number of output units, there is a "Simple Training Method" available which automatically chooses this parameter. We trained SOMs using the CLEMENTINE function "Train Kohonen" with the "Simple Training Method" on all of the 18 data sets described in Sec. 3. Although the data

[5] Please note that the "border" between the two groups in the right hand graph of Fig. 4 is caused by unallocated output units, i.e. codebook vectors $\hat{x}$ to which no data points with minimum distance $d(x, \hat{x})$ exist. Such unallocated output units are possible since we employed a SOM with more output units than data points available.

sets with either 100 or 225 data points contained either 4 or 9 easily separable clusters, CLEMENTINE always automatically chose a 7×5 grid of output units. This means that if CLEMENTINE's SOMs are used with the "Simple Training Method", the aim is not to do clustering or simultaneous clustering and visualization as described in Sec. 2 and 3 since the number of output units is far from estimating the correct number of clusters present in the data. CLEMENTINE rather uses SOM for visualization only or, if we follow the user guide's advice that SOMs "perform clustering", for clustering via visualization.

WEBSOM organizes large collections of text documents by mapping vectorial representations (which are related to word frequencies) onto a two-dimensional display using a SOM. In an example given in [16] $1,124,134$ documents from "80 very different Usenet newsgroups" are being mapped onto a SOM with $104,040$ output units. Again it should be clear that SOM is used for clustering via visualization since the huge number of output units stands in no relation to the assumed number of clusters present in the data (80 clusters corresponding to 80 different newsgroups). The method described in [26] is used "to indicate the clustering tendency" as shades of gray on the output grid.

7 Discussion

In this work we tried to make the notion of using SOM as a "data visualization tool" more concrete by showing that the number of output units used in a SOM influences its applicability for either clustering or visualization. We showed that if the number of output units N is set equal to g, the number of clusters present in the data set, SOM can be used both for clustering alone and for clustering plus simultaneous visualization. Theoretical as well as empirical results make clear that for these purposes the degree of neighbourhood should be set to zero at the end of learning which makes SOM equivalent to online K-means Clustering. Our own empirical results show that the simultaneous visualization of cluster centers (output units) is impaired due to SOM's discretization of the output space. SOM can also be used for visualization only or for clustering via visualization and then the number of output units N can be in the order of the number of input vectors n or even a multiple of it. SOM's visualization ability does again suffer from the discretization of the output space which is exemplified via empirical results. As about clustering via visualization, it is known from the literature that this bears the high risk of missing the true cluster structure. We conclude that SOM is a flexible tool which can be used for various forms of clustering and visualization but that this flexibility comes with a price in terms of impaired performance.

It should be noted that this account of SOMs is about the "classical"

formulation of the algorithm as given in [15] and described in Sec. 2. We do not comment on the numerous existing re-formulations and changes to the original algorithm. Nevertheless, since the role of the number of output units is left unchanged in most of these alternative formulations, our line of argumentation should still hold for these models. We did also not comment on practical issues of implementation or speed of algorithm or anything the-like. It is e.g. true that Sammon mapping is a rather involved and slow technique compared to SOM. Sammon mapping has the additional disadvantage that it computes a fixed mapping from a set of input points to a set of output points. Whenever a new previously unseen input point is encountered, the whole mapping has to be recomputed. One can get around this problem by resorting to the following heuristic: find the input point x_i^{old} that is closest to the new input point x^{new}; use the output point $\dot{x}_i^{old}$ that corresponds to x_i^{old} as the mapped representation of x^{new}. In fact this is exactly what SOM does whenever new data has to be mapped after the training phase has been finished. A more principled answer to the problem of a "fixed mapping" is to use a neural network to compute the mapping from a high dimensional input space to a lower dimensional output space [18]. Such a neural network minimizes an energy function similar to the one given for Sammon mapping in Equ. 5 and has the advantages of increased learning speed as well as true generalization ability (i.e. it computes a true mapping from input to output space and not just from one set of points to another).

Concerning the use of SOM in the data analysis community as discussed in the context of CLEMENTINE and WEBSOM, it has to be said that these tools rely on SOM's ability to do clustering via visualization. Users of CLEMENTINE and WEBSOM should be aware of the possible pitfall of missing the true cluster structure as well as of the impaired visualization due to the discretization of the output display.

Acknowledgements: Thanks are due to James Pardey, University of Oxford, for the Sammon code. The SOM_PAK, Helsinki University of Technology, was used for all computations of self-organizing maps. Parts of this work were done within the BIOMED-2 BMH4-CT97-2040 project SIESTA, funded by the EC DG XII. The Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministry of Education, Science and Culture. The author was supported by a doctoral grant of the Austrian Academy of Sciences.

References

[1] Balakrishnan P.V., Cooper M.C., Jacob V.S., Lewis P.A.: A study of the classification capabilities of neural networks using unsupervised learning: a comparison with k-means clustering, Psychometrika, Vol. 59, No. 4, 509-525, 1994.

[2] Bezdek J.C., Pal N.R.: An index of topological preservation for feature extraction, Pattern Recognition, Vol. 28, No. 3, pp.381-391, 1995.
[3] Bishop C.M., Svensen M., Williams C.K.I.: Magnification factors for the SOM and GTM algorithms, Proc. of WSOM'97: Workshop on Self-Organizing Maps, Helsinki, pp. 333-338, 1997.
[4] Bishop C.M., Svensen M., Williams C.K.I.: GTM: The Generative Topographic Mapping, Neural Computation, Vol. 10, Issue 1, p.215-234, 1998.
[5] Bottou L., Bengio Y.: Convergence Properties of the K-Means Algorithms, in Tesauro G., et al.(eds.), Advances in Neural Information Processing System 7, MIT Press, Cambridge, MA, pp.585-592, 1995.
[6] Clementine User Guide, Integral Solutions Limited, 1998.
[7] Cottrell M., Fort J.C., Pages G.: Theoretical aspects of the SOM algorithm, Neurocomputing, (21)1-3, pp.119-138, 1998.
[8] Duda R.O., Hart P.E.: Pattern Classification and Scene Analysis, John Wiley & Sons, N.Y., 1973.
[9] Erwin E., Obermayer K., Schulten K.: Self-organizing maps: ordering, convergence properties and energy functions, Biological Cybernetics, 67, 47- 55, 1992.
[10] Flexer A.: Limitations of Self-Organizing Maps for Vector Quantization and Multidimensional Scaling, in Mozer M.C., et al.(eds.), Advances in Neural Information Processing Systems 9, MIT Press/Bradford Books, pp.445-451, 1997.
[11] Goodhill G.J., Sejnowski T.J.: Quantifying neighbourhood preservation in topographic mappings, Proceedings of the 3rd Joint Symposium on Neural Computation, San Diego and California Institute of Technology, 6, Pasadena, CA: California Institute of Technology, 61-82, 1996.
[12] Hubert L.J., Arabie P.: Comparing partitions, J. of Classification, 2, 63-76, 1985.
[13] Jolliffe I.T.: Principal Component Analysis, Springer, 1986.
[14] Kohonen T.: Self-Organization and Associative Memory, Springer, 1984.
[15] Kohonen T.: Self-organizing maps, Springer, Second Extended Edition, Springer Series in Information Sciences, Vol. 30, 1997.
[16] Kohonen T.: Self-Organization of Very Large Document Collections: State of the Art, in Niklasson L., et al.(eds.), Proceedings of the 8th International Conference on Artificial Neural Networks, Springer, 2 vols., pp.65-74, 1998.
[17] Lagus K., Honkela T., Kaski S., Kohonen T.: Self-Organizing Maps of Document Collections: A New Approach to Interactive Exploration, in Simoudis E. & Han J.(eds.), KDD-96: Proceedings Second International Conference on Knowledge Discovery & Data Mining, AAAI Press/MIT Press, pp.238-243, 1996.
[18] Lowe D., Tipping M.E.: NeuroScale: Novel Topographic Feature Extraction Using RBF Networks, in Mozer M.C. et al. (eds.), Advances in Neural Information Processing Systems 9 (NIPS'97), MIT Press/Bradford Books, Cambridge/London, pp.543-549, 1997.
[19] MacQueen J.: Some Methods for Classification and Analysis of Multivariate Observations, Proc. of the Fifth Berkeley Symposium on Math., Stat. and Prob., Vol. 1, pp. 281-296, 1967.
[20] Mezzich J.: Evaluating clustering methods for psychiatric diagnosis, Biological Psychiatry, 13, 265-346, 1978.

[21] Milligan G.W., Cooper M.C.: An examination of procedures for determining the number of clusters in a data set, Psychometrika 50(2), 159-179, 1985.
[22] Ripley B.D.: Pattern Recognition and Neural Networks, Cambridge University Press, 1996.
[23] Sammon J.W.: A Nonlinear Mapping for Data Structure Analysis, IEEE Transactions on Comp., Vol. C-18, No. 5, p.401-409, 1969.
[24] Schwenker F., Kestler H., Palm G.: Adaptive Clustering and Multidimensional Scaling of Large and High-Dimensional Data Sets, in Niklasson L., et al.(eds.), Proceedings of the 8th International Conference on Artificial Neural Networks, ICANN'98, Springer, pp.911-916, 1998.
[25] Sneath P.H.A.: The risk of not recognizing from ordinations that clusters are distinct, Classification Society Bulletin, 4, 22-43, 1980.
[26] Ultsch A.: Self-organizing Neural Networks for Visualization and Classification, in Opitz O., et al.(eds.), Information and Classification, Springer, Berlin, 307-313, 1993.
[27] Ultsch A., Vetter C.: Self-Organizing-Feature-Maps versus Statistical Clustering Methods: A Benchmark, University of Marburg, FG Neuroinformatik & Kuenstliche Intelligenz, Research Report 0994, 1994.
[28] Vesanto J.: SOM-based data visualization methods, Intelligent Data Analysis, Vol. 3, Nr. 2, 111-126, 1999.
[29] Vesanto J., Alhoniemi E.: Clustering of the Self-Organizing Map, IEEE Transactions on Neural Networks, Vol. 11, No. 3, p.586-600, 2000.
[30] Waller N.G., Kaiser H.A., Illian J.B., Manry M.: A comparison of the classification capabilities of the 1-dimensional Kohonen neural network with two partitioning and three hierarchical cluster analysis algorithms, Psychometrika, Vol. 63, No.1, 5-22, 1998.