

# 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. The usage of SOM in the data mining community is covered by discussing its application in the data mining tools CLEMENTINE and WEBSOM.

## 1 Introduction

Self-organizing maps (SOM) [12] are a very popular tool used for a range of different purposes including clustering and visualization of high dimensional data spaces. SOM is also used in two prominent data mining 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 [15] and [14]).

Although there is vast literature available concerning SOMs, a recent survey [13] 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 recent comprehensive monograph [13] 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], [13], [4] and [21]. 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 [24] 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 [21]). 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 mining tools CLEMENTINE and WEBSOM will be discussed.

## 2 SOM for Clustering

According to a standard text book on pattern recognition [19] “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”. A classical technique to achieve such a grouping is the  $K$ -means approach developed in the cluster analysis literature (starting from [16]). 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  $\{x_j; j = 0, \dots, n - 1\}$ , an initial set  $\hat{A}_0$  of  $N$  codebook vectors  $\hat{x}$  and a discrete-time coordinate  $t = 0 \dots, n - 1$ .
2. Given  $\hat{A}_t = \{\hat{x}_i; i = 1, \dots, N\}$ , find the minimum distortion partition  $P(\hat{A}_t) = \{S_i; i = 1, \dots, N\}$ . Compute  $d(x_t, \hat{x}_i)$  for  $i = 1, \dots, N$ . If  $d(x_t, \hat{x}_i) \leq d(x_t, \hat{x}_l)$  for all  $l$ , then  $x_t \in S_i$  ( $d$  is usually Euclidean distance).
3. Update the codebook vector with the minimum distortion

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

where  $\alpha$  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 main difference between the SOM-algorithm and oKMC is the fact that the codebook vectors are the weight vectors of the 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. 1 is replaced by

$$\hat{x}_{(t)}(S_i) = \hat{x}_{(t-1)}(S_i) + h[x_{(t)} - \hat{x}_{(t-1)}(S_i)] \quad (2)$$

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 and that also includes an additional learning parameter  $\alpha$ . If the degree of neighbourhood is decreased to zero, the SOM-algorithm becomes equal to the oKMC-algorithm. Whereas local convergence is guaranteed for oKMC (at least for decreasing  $\alpha$ , [5]), no general proof for the convergence of SOM with nonzero neighbourhood is known. In [13] 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 [24]) use SOMs and cluster algorithms with  $N$  always set equal to  $g$ , the number of clusters known to be in the data. In [24] 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<sup>1</sup>, 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 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 [13] 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.

---

<sup>1</sup> 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.

### 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.

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  $|x \in R^k|$  and the number of output vectors  $|\dot{x} \in R^p|$  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  $\dot{x}_i, \dot{x}_j \in R^p$ ,  $i, j = 1, \dots, n-1$  and  $d(\dot{\cdot})$  being a measure of distance in  $R^k$  ( $R^p$ ). Such a transformation is said to produce an *isometric* image.

Techniques for finding such transformations  $\Phi$  are, among others, various forms of *multidimensional scaling*<sup>2</sup> (MDS) like Sammon mapping [20], but also principal component analysis (PCA) (see e.g. [11]) or SOM. Sammon mapping is doing MDS by minimizing the following via steepest descent:

$$\frac{1}{\sum_{i=0}^{n-1} \sum_{j<i} d(x_i, x_j)} \sum_{i=0}^{n-1} \sum_{j<i} \frac{(d(x_i, x_j) - \dot{d}(\dot{x}_i, \dot{x}_j))^2}{d(x_i, x_j)} \quad (3)$$

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. Since SOM has been designed heuristically and not to find an extremum for a certain energy function<sup>3</sup>, the theoretical connection to other MDS algorithms remains unclear. It should be noted that for SOM the number of output vectors  $|\dot{x} \in R^p|$  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. Our new combined approach (abbreviated oKMC+) consists of simply finding the set of  $\hat{A} = \{\hat{x}_i, i = 1, \dots, N\}$  codebook vectors that give the minimum distortion partition  $P(\hat{A}) = \{S_i; i = 1, \dots, N\}$  via oKMC and then using the  $\hat{x}_i$  as input vectors to Sammon mapping and thereby obtaining a

<sup>2</sup> 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.

<sup>3</sup> In [9] it is even shown that such an objective function cannot exist for SOM.

two dimensional representation of the  $\hat{x}_i$  via minimizing the term in Equ. 3. 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 [21] a similar combined technique is proposed with the difference that clustering and visualization is achieved simultaneously and not one after the other.

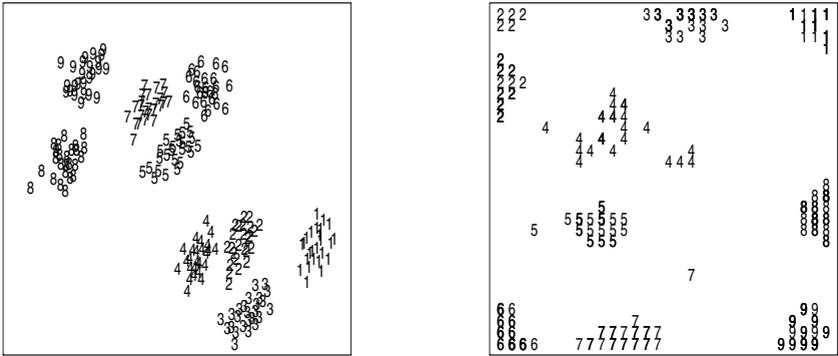
The empirical comparison was done using multivariate normal distributions generated by a procedure which is standard for comparisons of cluster algorithms (see [18] and [1]). We produced 36 data sets with number of clusters being 4 or 9, and the number of dimensions being 4, 6 or 8. All clusters showed internal cohesion as well as external isolation. The latter was defined as having all clusters non-overlapping in the first dimension.

We compared two-dimensional SOMs with numbers of output units set equal to the numbers of clusters known to be in the data (4 or 9) to oKMC+ models with corresponding sizes of codebooks. SOM performed almost equally well as oKMC+ in recovering the structure of the clusters (measured via the so-called Rand index which is closely related to data points misclassified) which is as expected since we set the neighbourhood to zero at the end of training. We used Pearson correlation to measure how well the topology is preserved by both SOM and oKMC+. We computed the Pearson correlation of the distances  $d(x_1, x_2)$  in the input space and the distances  $\hat{d}(\hat{x}_i, \hat{x}_j)$  in the output space for all possible pairwise comparisons of data points. Note that for SOM the coordinates of the codebook vectors on the planar grid were used to compute the  $\hat{d}$ . 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). SOM performed significantly worse in preserving the topology, we obtained a correlation 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 non-zero neighbourhood at the end of SOM training did not warrant any significant improvements. Full details of this study are given in [10].

## 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 [13]). 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.



**Fig. 1.** Output representations after mapping nine eight-dimensional clusters via Sammon mapping (left) and SOM (right). Numbers indicate true cluster membership.

We did an own study on visualization with SOM using the same 36 data sets described in Sec. 3. We computed SOMs consisting of  $20 \times 20$  (for data sets consisting of 4 clusters and 100 points) or  $30 \times 30$  (for 9 clusters and 225 points) codebook vectors for all 36 data sets which gave an average correlation of 0.77 between the distances  $d_i$  and  $\hat{d}_i$ . 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. This result together with the previously described study [2] indicates that even using more output units than input vectors available does not help against the drawbacks of SOM's discretization of the output-space. This rigidity of the output map is clearly visible if one compares examples of output maps given in Fig. 1.

## 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. [23] and [3]).

It should be clear that for this type of application SOMs with large amounts of output units will be best suited. However, it has long been known within the clustering community that doing clustering via visualization bears some pitfalls. In [22] 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 [17] 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. 1, 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.

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 36 data sets described in Sec. 3. Although the data sets with either 100 or 225 data points contained either 4 or 9 easily separable clusters, CLEMENTINE always automatically chose a  $7 \times 5$  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 Secs. 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 [14] 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 [23] is used "to indicate the clustering tendency" as shades of gray on the output grid.

## 6 Conclusions

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.

Concerning the use of SOM in the data mining 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:* 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 Science and Transport. 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., Nikhil R.P.: 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. Jolliffe I.T.: *Principal Component Analysis*, Springer, 1986.
12. Kohonen T.: *Self-Organization and Associative Memory*, Springer, 1984.
13. Kohonen T.: *Self-organizing maps*, Springer, Second Extended Edition, Springer Series in Information Sciences, Vol. 30, 1997.
14. 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.
15. 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.
16. 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.
17. Mezzich J.: Evaluating clustering methods for psychiatric diagnosis, *Biological Psychiatry*, 13, 265-346, 1978.
18. 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.
19. Ripley B.D.: *Pattern Recognition and Neural Networks*, Cambridge University Press, 1996.
20. Sammon J.W.: A Nonlinear Mapping for Data Structure Analysis, *IEEE Transactions on Comp.*, Vol. C-18, No. 5, p.401-409, 1969.
21. 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.
22. Sneath P.H.A.: The risk of not recognizing from ordinations that clusters are distinct, *Classification Society Bulletin*, 4, 22-43, 1980.
23. Ultsch A.: Self-organizing Neural Networks for Visualization and Classification, in Opitz O., et al.(eds.), *Information and Classification*, Springer, Berlin, 307-313, 1993.
24. 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.