CLUSTERING WIH SOM: U*C

Alfred Ultsch
Databionics Research Group
University of Marburg
Marburg, Germany
Ultsch@Informaik.uni-marburg.de

Abstract –*A new clustering algorithm based on emergent SOM is proposed. This algorithm, called U*C, uses distance information together with density structures. No particular geometrical cluster model is imposed on the data by U*C. In contrast to other clustering algorithms, U*C identifies cluster structures that are not separable by contiguous surfaces. The number of clusters is determined automatically. The validity of the clusters found is assessed using the U*-Matrix. The U*-Matrix gives a combined visualization of distance and density structures of a high dimensional data set. U*C clustering is superior to standard clustering algorithms such as K-means and hierarchical clustering. This is demonstrated on a set of critical clustering problems called FCPS, which is published on our web site.*

Key words – SOM, visualization, clustering, density based clustering algorithms.

1 Introduction

Contrary to common belief, Self Organizing Maps (SOM) are not clustering algorithms. The identification of the neurons of a SOM with clusters results in SOMs with very few neurons, since the number of clusters is usually small. It can be shown, that this usage of SOMs is just a variant of the K-means clustering algorithm [1]. SOMs with a large number of neurons can be regarded as a nonlinear projection from high dimensional data space to a map in the geographical sense. Such SOM disentangle cluster structures that are linear not separable. The ChainLink example (see Fig. 1) was among the first to illustrate this [2]. Distance relationships in a high dimensional data space can be visualized on a SOM in form of a U-Matrix [3]. The recently introduced P-Matrix allows a visualization of density structures of the high dimensional data space [4]. In this paper we present a combined visualization of distance and density in form of the U*-Matrix (Chapter 2). Then we define a novel clustering algorithm, called U*C, which uses distance and density information (Chapter 3). U*C is tested on some crucial clustering problems (Chapter 4). Its performance is compared to K-means and popular hierarchical clustering algorithms (Chapter 5). The validity and precision of the resulting clusters can be judged using the U*-Matrix. The performance of U*C is discussed in chapter 6.

2 SOM with emerging distance and density structures

SOM with a large number of neurons are mappings from a high dimensional data space $D \subset R^n$ onto neurons on a map, in a geographical sense. In the following a trained SOM with a sufficiently large number of neurons k is presumed, typically k >4000. The SOM training algorithm constructs a nonlinear and topology preserving mapping of the input data set E = $\{x_1,...,x_d\}$ with $x_i \in D$ onto the set of neurons M=$\{n_1,...,n_k\}$ with associated weight vectors W = $\{w_1,\ldots,w_k\}$. Each data point x_i is mapped to its bestmatch neuron $bm(x_i)= n_b \in M$ such that $d(x,w_b) \leq d(x, w_j)\ \forall\ w_j \in W$, where d is the distance on the data set. The neurons are arranged on a two dimensional map: each neuron i possesses a set of two coordinates embedded in a two dimensional surface. The set of immediate neighbors of a neuron n_i on the map is denoted by N(i). It is assumed, that the SOM gives a mapping with minimal, or at least tolerable, topological errors. Compare [8] for map dimensions and surface structures to minimize projection errors. For the U-, P- and U*-Matrix defined below, the map is the floor space layout for a landscape like visualization of distance- and density structures of the high dimensional data space. Structures emerge on top of the map by the cooperation of many neurons. Single neurons are only tiny parts of these structures. SOM showing such emerging structures have been called emergent SOM (ESOM) [5].

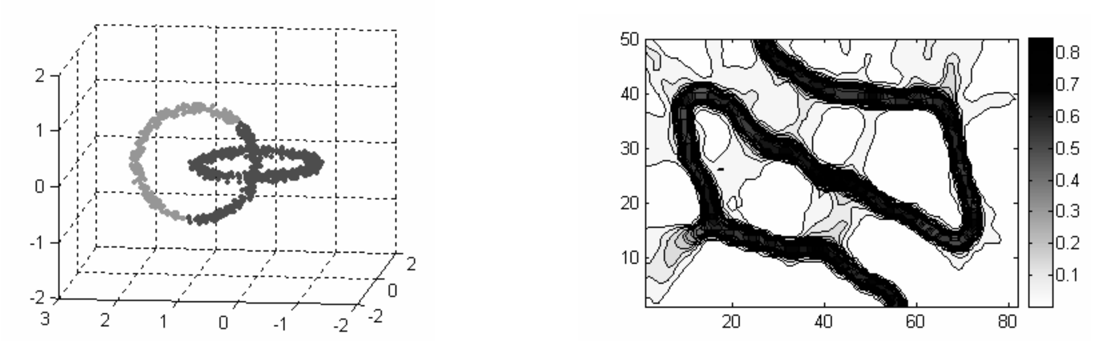

Figure 1: a) ChainLink with Ward clustering b) U-Matrix showing two disentangled clusters

2.1 U-Matrix

The U-height for each neuron n_i is the average distance of n_i's weight vectors to the weight vectors of its immediate neighbors N(i). The U-height uh(i) is calculated as follows:

$$uh(i) = \frac{1}{n}\sum_{j} d(w_i, w_j)\ ,\ j \in N(i),\ n = |N(i)|.$$

A display of all U-heights on top of the map is called a U-Matrix [3]. A single U-height shows the local distance structure. The local average distance at w_i is shown at neuron n_i. The overall structure of densities emerges, if a global view of a U-Matrix is regarded. Figure 1b shows an example of a U-Matrix on ESOM with 50x82 neurons.

2.2P-Matrix

The P-height ph(i) for a neuron n_i is a measure of the density of data points in the vicinity of w_i: $ph(i) = |\{x \in E \mid d(x, w_j) < r > 0, r \in R\}|$. A display of all P-heights on top of the grid G is called a P-Matrix [4]. Figure 3b, 4b and 5b show examples of P-Matrices. The P- height is the number of data points within a hypersphere of radius r. The radius r should be chosen such that ph(i) approximates the probability density function of the data points. The usage of the ParetoRadius, as described in [4], is a choice for r. Median filtering within the N(i) window can be applied on the P-Matrix. This reduces local fluctuations (noise) in the density estimation without disturbing the overall picture. Such a filtering preserves, however, the density gradients important for clustering.

2.3 U*-Matrix

For the identification of clusters in data sets it is sometimes not enough to consider distances between the data points. Consider, for example, the TwoDiamonds data set depicted in Figure 2. The data consists of two clusters of points on a plane. Inside each “diamond” the values for each data point were drawn independently from uniform distributions. At the central region, marked with an arrow in Figure 2, the distances between the data points are very small. For distance based cluster algorithms it is hard to detect correct boundaries for the clusters. Distance oriented clustering methods such as single linkage, complete linkage, Ward etc. produce classification errors. The picture changes, however, when the data’s density is regarded (see Figures 2, 3 and 5). The density at the touching point of the two diamonds is only half as big as the densities in the center regions of the clusters.

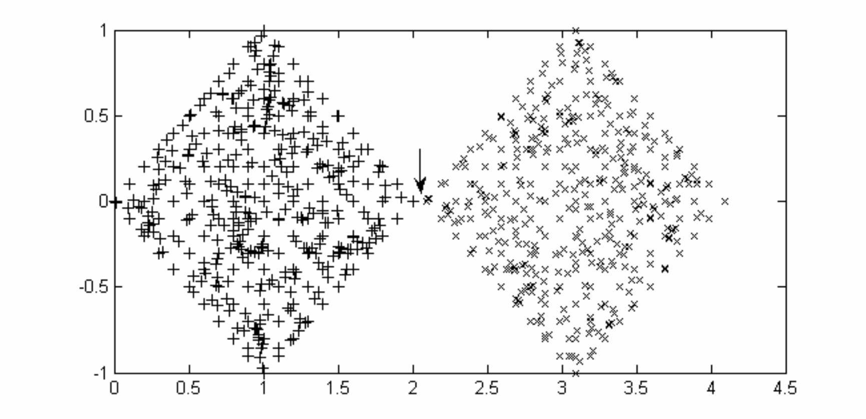

Figure 2: TwoDiamonds data set

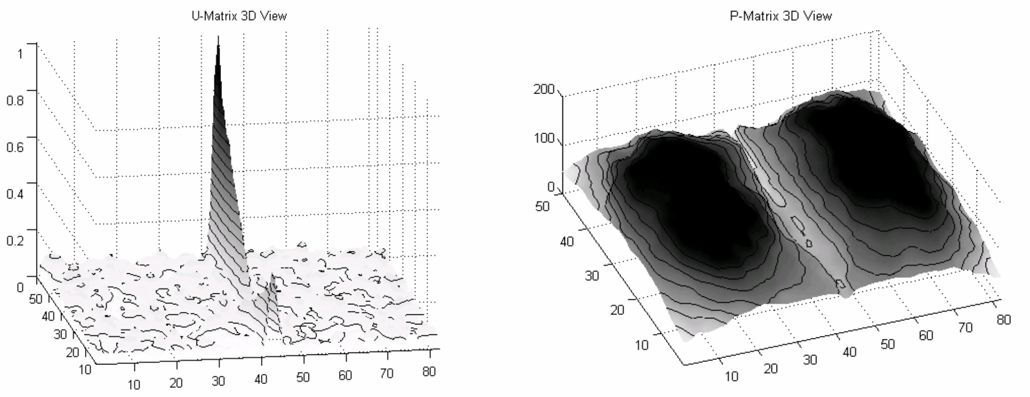

Figure 3: a) U-Matrix of TwoDiamonds b) P-Matrix of TwoDiamonds

As the TwoDiamonds data set shows, a combination of distance relationships and density relationships is necessary to give an appropriate clustering. The combination of a U-Matrix and a P-Matrix is called U*-Matrix. The U*-height u*h(i) for a neuron n_i is the U-height multiplied with the probability that the local density, as measured by ph(i), is low. As an estimate of this probability the empirical density function can be used:

$$\mathrm{plow(i)} = Pr(\text{data density is low for neuron } \mathrm{n_i}) \cong \frac{\left|\left\{\mathrm{p} \in \mathrm{P\text{-}matrix} \mid \mathrm{p} > \mathrm{ph(i)}\right\}\right|}{\left|\left\{\mathrm{p} \in \mathrm{P\text{-}matrix}\right\}\right|} \quad \text{(i).}$$

The U*height is then calculated as u*h(i) = uh(i) · plow(i). If the local data density is low: u*h(i) = uh(i). This happens at the presumed border of clusters. If the data density is high, then u*h(i) = 0. This is in the central regions of clusters. For neurons with median density holds: u*h(i) = uh(i)*0.5. The U*-Matrix exhibits therefore the local data distances as heights, when the data density is low (cluster border).

If the data density is high, the distances are scaled down to zero (cluster center). An alternative to formula (i) is, to adjust the multiplication factor such that u*h(i)=uh(i) for median P-heights and u*h(i)=0 for the top 20 percent of P-heights [7].

The cluster structure of the data can be seen more clearly on the U*-Matrix than on the U-Matrix. Figure 4 compares, for example, the U-Matrix taken from [6] to a U*-Matrix of the same data set. Since density and distance play different roles in the definition of clusters, we think that the three different matrices, U-, P- and U*-Matrix together give an appropriate impression of the cluster structure of any high dimensional data (see Figure 4).

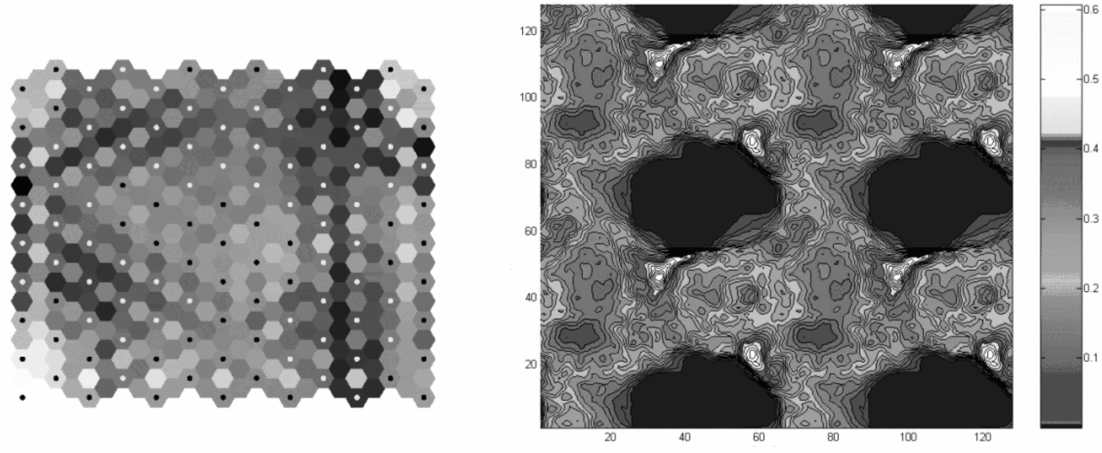

Figure 4: a) U-Matrix of DNA microarray data [6], b) U*-Matrix of the same data.

3 U*C Clustering Algorithm

A topological correct ESOM projects a cluster onto a coherent area on the map (cluster area). Points within the cluster are mapped to the inside of the cluster area. Data points at the border (surface) of the cluster are projected to the border of the cluster area. Consider a data point x at the surface of a cluster C, with ni = bm(x). The weight vectors of its neighbors N(i) are either within the cluster, in a different cluster or interpolate between clusters. If we assume that the inter cluster distances are locally larger then the local inner cluster distances, then the U-heights in N(i) will be large in such directions which point away from the cluster C. This means, a gradient descent on the U-Matrix will lead away from cluster borders. A movement from one neuron ni to another neuron nj with the result that wj is more within a cluster C than wi is called immersive. For data points well within C, a gradient descent on a U-Matrix will, however, not necessarily be immersive.

The P-heights follow the density structure of a cluster. Under the assumption that the core parts of a cluster are those regions with largest density, a gradient ascent on the P-Matrix is immersive. Clusters may also be defined by density alone instead of distance. See, for example, the

EngyTime data set shown in Figure 5a and its density structure as shown by the P-Matrix in Figure 5b. This data set represents situations where the data generation can be described appropriately by Hidden Markov Models.

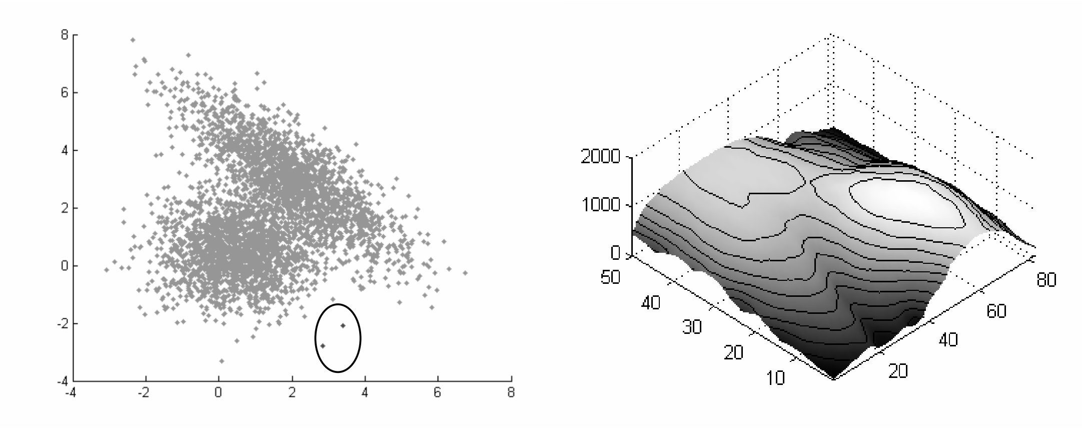

Figure 5: a) EngyTime/SingleLinkage clustering b) P-Matrix of EngyTime

At the borders of a cluster the measurement of density is, however, critical. At cluster borders the local density of the points should decrease substantially. In most cases the cluster borders are defined either by low point densities (see Figure 5) or by “empty space” between clusters (= large inter cluster distances). For empirical estimates of the point density a gradient ascent on a P-Matrix may therefore not be immersive for points at cluster borders. A movement on a SOM map which follows first a gradient descent on a U-Matrix and then a gradient ascent on a P-Matrix is called immersion. Let I denote the end points of immersion starting from every neuron on a map. If the density within a cluster is constant, immersion will not converge to a single point for a cluster for all starting pointswithin a cluster. The U*-Matrix is then used to determine which points in I belong to the same cluster. The watersheds of the U*-Matrix are calculated using the algorithm described in [5]. Points that are separated by a watershed are assigned to different clusters, points within the same basin to a single cluster. The following pseudocode summarizes the U*C clustering algorithm described above.

```
U*C clustering Algorithm:  given ESOM with U-Matrix, P-Matrix, U*-Matrix, I = {};
Immersion:
  For all neurons n of an ESOM:
    1) from neuron n follow a gradient descent on the U-Matrix until a minimum is reached
       in neuron u
    2) from neuron u follow a gradient ascent on the P-Matrix until a maximum is reached
       in neuron p.
    3) $I = I \cup \{p\}$;  Immersion(n) = p.
Cluster assignment:
    1) calculate the watersheds for the U*-Matrix ( e.g. using [5]).
    2) partition I using these watersheds into clusters $C_1,\dots C_c$
    3) assign a data point x  to a cluster $C_j$ if  Immersion(bm(x) ) $\in C_{j.}$
```

4 Fundamental Clustering Problems Suite

The efficiency of the U*C clustering algorithm is tested using a set of ten clustering problems called Hepta, Lsun, Tetra, Chainlink, Atom, EngyTime, Target, TwoDiamonds, WingNut and GolfBall. Any reasonable clustering algorithm should be able to solve these problems correctly [1]. As can be seen below, however, standard algorithms like K-means, and hierarchical clustering algorithms, like single linkage and Ward have difficulties on several data sets. The suite of data sets is called Fundamental Clustering Problem Suite (FCPS). The suite can be downloaded from the website of the author (http://www.informatik.uni-marburg.de/~ databionics). FCPS poses some hard clustering problems. Chainlink and Atom are not separable by hyperplanes. The GolfBall data set consists of points that are equidistant on the surface of a sphere. This data set is used to address the problem to impose cluster structures when no such structure is present. The problem of outliers is addressed by the Target data set shown in Figure 6.

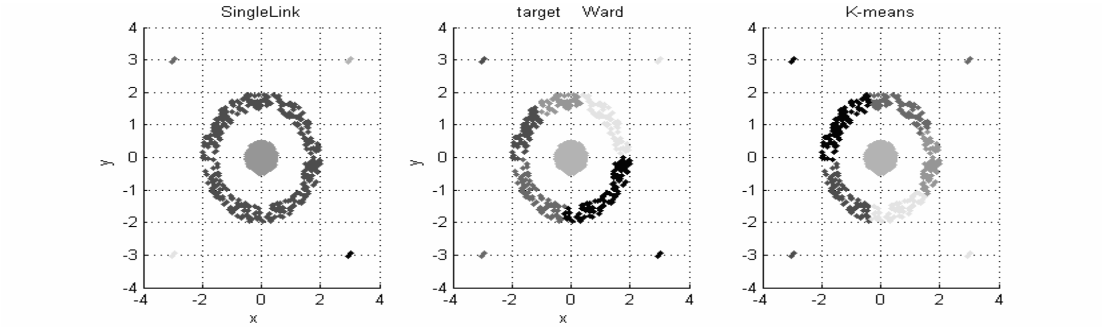

Figure 6: SL, Ward and K-means clustering algorithms on the Target data set

The following is a short description of the data set and the problem it poses to cluster algorithms. Pictures of the data sets are shown in the Figures of this paper.

Name	Cases	# Vars	#Clusters	Main Clustering Problem
Hepta	212	3	7	different densities in clusters
Lsun	400	2	3	different variances in clusters
Tetra	400	3	4	large inner distances vs. small inter distances
Chainlink	1000	3	2	not separable by linear decision boundaries
Atom	800	3	2	linear not sep., different densities and variances
EngyTime	4096	2	2	density defined clusters
Target	770	2	6	outliers
TwoDiamonds	800	2	2	touching clusters
WingNut	1070	2	2	largest densities at cluster borders
GolfBall	4002	3	1	equidistant points, no cluster at all

5 Results

The results of U*C Clustering are compared to K-means as most popular partitioning cluster algorithm. The hierarchical cluster algorithms SingleLinkage and Ward were applied to the FCPS data sets. All algorithms were used as implemented in MATLAB™. Since K-means converges to a local minimum of a cost function, the best of 100 repetitions was kept. The correct number of clusters was given as parameter to all standard algorithms. Shown are the overall accuracies. Performances lower than 80% are emphasized. There is no data set on which U*C performs worse than any of the other clustering algorithms.

Data Set	**SingleLinkage**	**Ward**	**K-means**	**U*C Clustering**
Hepta	100 %	100 %	100 %	100 %
Lsun	100 %	50 %	50 %	100 %
Tetra	0.01 %	90 %	100 %	100 %
Chainlink	100 %	50 %	50 %	100 %
Atom	100 %	50 %	50 %	100 %
EngyTime	0 %	90 %	90 %	90 %
Target	100 %	25 %	25 %	100 %
TwoDiamonds	0 %	100 %	100 %	100 %
WingNut	0 %	80 %	80 %	100 %
GolfBall	100 %	50 %(best)	*)	100 %

Table 1: Accuracy (= nr of correct classifications) of the clustering algorithms on FCPS

*) see last paragraph of dicsussion.

6 Discussion

SingleLinkage (SL) clustering imposes a chain of data points as cluster model on the data. This algorithm is usually misled, if the local inner cluster distances are in the same range as the inter cluster distances. This can be seen in the performance on the Tetra, EngyTime, Two Diamonds and WingNut data set. SL is, however, able to separate clusters that are not separable by hyper planes, e.g. ChainLink and Atom. The fundamental cluster model of Ward is an hyper ellipsoid. For data sets which do not fit this model (5 of the 10 sets in FCPS), Ward produces an erroneous clustering. Most clustering algorithms require the knowledge of the number of clusters. U*C determines the number of clusters automatically. It is, however, crucial to assess the validity of a clustering. Dendrograms as used in hierarchical clustering algorithms seem to address this, are, however, sometimes misleading as the following example shows. Figure 7a shows a dendrogram of the GolfBall data using Ward hierarchical clustering. Such a dendrogram would suggest 3 or 6 cluster.

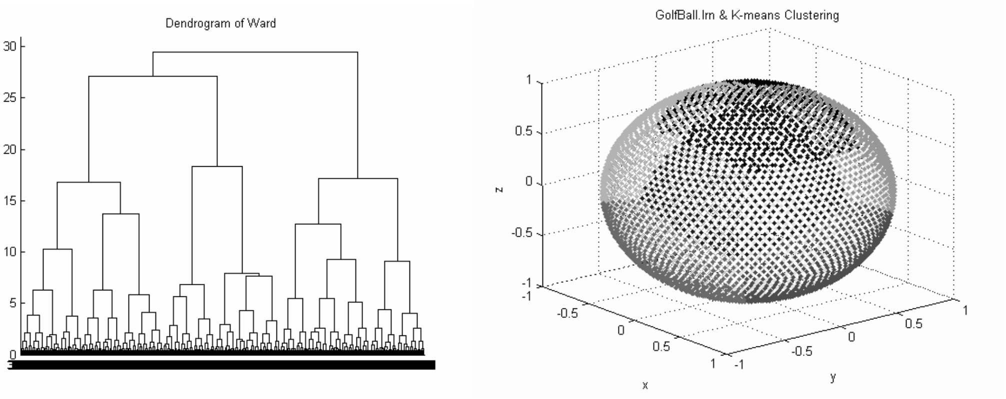

Figure 7: a)GolfBall Dendrogram of Ward b) Golf Ball data & K-means clustering

K-means requires that the clusters are compact and have about the same variance. The boundary separating two clusters in K-means is always a hyperplane. This results in the bad performance of K-means on 5 of the 10 data sets in the FCPS. K-means enforces any number of clusters on the data. In Figure 7b a K-means clustering of GolfBall with k = 6 is shown. For high dimensional data this result may, however, be taken serios, in particular in the light of the dendrogram of Figure 7a. The U*-Matrix serves as a feedback of the cluster structure in U*C. Compare Figure 4b, for example, to see, whether there are cluster structures in a very high dimensional data set.

7 Conclusion

A new clustering algorithm based on emergent SOM (ESOM) is proposed. This algorithm uses distance structures (U-Matrix) as well as density structures (P-Matrix) of the data set. It inherits the nonlinear disentangeling of cluster structures from the underlying SOM. No particular geometrical cluster model is imposed on the data by U*C. Other clustering algorithms impose such a model and are performing poor, if the data set is of a different structure. The number of clusters is determined automatically in U*C. The correctness and validity of the clusters found can be assessed using the U*-Matrix. The U*-Matrix shows a combined picture of distance and density structures of a high dimensional data set. U*C performs superior to standard clustering algorithms such as K-means and the most popular hierarchical algorithms [1]. This is demonstrated on a group of data sets which represent fundamental clustering problems, like different variances, outliers and other structural difficulties. U*C and other tools for ESOM, see [9], together with an extended version of this paper, can be downloaded from our web site.

References

[1] A.K. Jain, R.C.Dubes(1998): Algorithms for Clustering Data, New York, Wiley.

[2] A.Ultsch, C. Vetter(1994), Selforganizing Feature Maps vs. statistical clustering, *Dept. of Computer Science University of Marburg,* Research Report 9.

[3] A.Ultsch, H.P.Siemon(1990), Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis, *Proc. Intern. Neural Networks,* Kluwer Academic Press, Paris, 305-308.

[4] A.Ultsch(2003), Maps for the Visualization of high-dimensional Data Spaces, *Proc. WSOM*, Kyushu, Japan, 225-230.

[5] V., Luc, P. Soille(1991), Watersheds in Digital Spaces: An Efficient Algorithm Based on Im-mersion Simulations, *IEEE Transactions of Pattern Analysis and Machine Intelligence*, Vol. 13(6), 583-598.

[5] A. Ultsch(1999), Data Mining and Knowledge Discovery with Emergent Self-Organizing Feature Maps for Multivariate Time Series, *E. Oja ,S. Kaski (eds), Kohonen Maps,* 33-46.

[6] S.Kaski et al (1999), Analysis and Visualisation of Gene Expression Data using Self Organizing Maps, *Proc NSIP*, 99-100.

[7] A.Ultsch(2003), U*-Matrix: A Tool to visualize Clusters in high dimensional Data, Dept. of Computer Science University of Marburg, Research Report 36.

[8] A. Ultsch, L. Herrmann (2005), The architecture of Emergent Self-Organizing Maps to reduce projection errors, ESANN, Brugges 2005, pp 1-6.

[9] A.Ultsch, F. Mörchen(2005), ESOM-Maps: tools for clustering, visualization, and classification with ESOM, Dept. of Computer Science Univ.of Marburg, Research Report. 46.