Fuzzy Clustering Based on Modified Distance Measures*

Frank Klawonn[1] and Annette Keller[2]

[1] Department of Electrical Engineering and Computer Science
Ostfriesland University of Applied Sciences
Constantiaplatz 4
D-26723 Emden, Germany
[2] Institute for Flight Guidance
German Aerospace Center
Lilienthalplatz 7
D-38108 Braunschweig, Germany

Abstract. The well-known fuzzy c-means algorithm is an objective function based fuzzy clustering technique that extends the classical k-means method to fuzzy partitions. By replacing the Euclidean distance in the objective function other cluster shapes than the simple (hyper-)spheres of the fuzzy c-means algorithm can be detected, for instance ellipsoids, lines or shells of circles and ellipses. We propose a modified distance function that is based on the dot product and allows to detect a new kind of cluster shape and also lines and (hyper-)planes.

1 Introduction

Fuzzy clustering techniques aim at finding a suitable fuzzy partition for a given data set. For a fuzzy partition a datum is not necessarily assigned to a unique class or cluster, but has membership degrees between zero and one to each cluster. Fuzzy clustering algorithms are applied for various reasons:

- The membership degrees give information about the ambiguity of the classification.
- Fuzzy clustering can adapt to noisy data and classes that are not well separated.
- Since most fuzzy clustering approaches are based on optimizing an objective function, membership degrees represent continuous parameters so that a continuous optimization problem has to be solved.
- Fuzzy clustering can be applied to learning fuzzy rules from data.

In this paper we briefly review the principal objective function-based fuzzy clustering approach in section 2. Various modifications of the distance function in the objective function have been proposed in order to model different cluster forms. In section 3 we introduce a new angle-based distance measure that

* This work was supported by the European Union under grant EFRE 98.053

is suitable for data sets with a smaller number of extreme values and a large number of 'normal' values. Section 4 modifies this approach and we obtain a clustering algorithm to detect lines and (hyper-)planes that can be applied to line recognition as well as to constructing Takagi-Sugeno fuzzy rule systems (see for instance [13]) that describe a function in terms of local linear models.

2 Objective Function-Based Fuzzy Clustering

We cannot give a complete overview on fuzzy clustering here and mention only the basic ideas in order to provide the background for our new algorithms. For a thorough overview on fuzzy clustering we refer to [2, 9]. Most fuzzy clustering algorithms aim at minimizing the objective function of weighted distances of the data to the clusters

$$J(X, U, v) \;=\; \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^{m} d^2(v_i, x_k) \tag{1}$$

under the constraints

$$\sum_{k=1}^{n} u_{ik} \;>\; 0 \qquad \text{for all } i \in \{1, \ldots c\} \tag{2}$$

and

$$\sum_{i=1}^{c} u_{ik} \;=\; 1 \qquad \text{for all } k \in \{1, \ldots n\}. \tag{3}$$

$X = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^p$ is the data set, c is the number of fuzzy clusters, $u_{ik} \in [0, 1]$ is the membership degree of datum x_k to cluster i, v_i is the prototype or the vector of parameters for cluster i, and $d(v_i, x_k)$ is the distance between prototype v_i and datum x_k. The parameter $m > 1$ is called fuzziness index. For $m \to 1$ the clusters tend to be crisp, i.e. either $u_{ik} \to 1$ or $u_{ik} \to 0$ resulting in the hard c-means algorithm, for $m \to \infty$ we have $u_{ik} \to 1/c$. Usually $m = 2$ is chosen. (2) ensures that no cluster is empty, (3) enforces that for each datum its classification can be distributed over different clusters, but the sum of the membership degrees to all clusters has to be one for each datum. Therefore, for this approach the membership degrees can be interpreted as probabilities and the corresponding clustering approach is called probabilistic. The strict probabilistic constraint was relaxed by Davé who introduced the concept of noise clustering [6, 7]. An additional noise cluster is added and all data have a (large) constant distant to this noise cluster. Therefore, noise data that are far away from all other clusters are assigned to the noise cluster with a high membership degree. Krishnapuram and Keller [12] developed possibilistic clustering by completely neglecting the probabilistic constraint (3) and adding a term to the objective function that avoids the trivial solution assigning no data to any cluster. We cannot discuss the details of these approaches here and restrict our considerations to the probabilistic fuzzy clustering approach. However, our algorithms can be

applied in the context of noise and possibilistic clustering in a straight forward way.

We also do not consider the problem of determining the number of clusters in this paper and refer to the overview given in [9].

The basic fuzzy clustering algorithm is derived by differentiating the Lagrange function of (1) taking the constraint (3) into account. This leads to the necessary condition

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{d^2(v_i, x_k)}{d^2(v_j, x_k)} \right)^{\frac{1}{m-1}}} \tag{4}$$

for the membership degrees for a (local) minimum of the objective function, given the prototypes are fixed. In the same way, we can derive equations for the prototypes, fixing the membership degrees, when we have chosen the parameter form of the prototypes and a suitable distance function.

The corresponding clustering algorithm is usually based on the so-called alternating optimization scheme that starts with a random initialization and alternatingly applies the equations for the prototypes and the membership degrees until the changes become very small. Convergence to a local minimum or (in practical applications very seldom) a saddle point can be guaranteed [1, 3].

The most simple fuzzy clustering algorithm is the fuzzy c-means (FCM) (see f.e. [2]) where the distance d is simply the Euclidean distance and the prototypes are vectors $v_i \in \mathbb{R}^p$. It searches for spherical clusters of approximately the same size and by differentiating (1) we obtain the necessary conditions

$$v_i = \frac{\sum_{k=1}^{n} u_{ik}^m x_k}{\sum_{k=1}^{n} u_{ik}^m} \tag{5}$$

for the prototypes that are used alternatingly with (4) in the iteration procedure.

Gustafson and Kessel [8] designed a fuzzy clustering method that can adapt to hyper-ellipsoidal forms. The prototypes consist of the cluster centres v_i as in FCM and (positive definite) covariance matrices C_i. The Gustafson and Kessel algorithm replaces the Euclidean distance by the transformed Euclidean distance

$$d^2(v_i, x_k) = (\det C_i)^{1/p} \cdot (x_k - v_i)^\top C^{-1} (x_k - v_i).$$

Besides spherical or ellipsoidal cluster shapes also other forms can be detected by choosing a suitable distance function. For instance, the prototypes of the fuzzy c-varieties algorithm (FCV) describe linear subspaces, i.e. lines, planes and hyperplanes [2, 4]. The equations for the prototypes of this algorithm require the computation of eigenvalues and eigenvectors of (weighted) covariance matrices. FCV can be applied to image recognition (line detection) and to construct local linear (fuzzy) models. Shell clustering algorithms are another class of fuzzy clustering techniques that are mostly applied to image recognition and detect clusters in the form of boundaries of circles, ellipses, parabolas etc. (For an overview on shell clustering see [9, 11].)

In principal any kind of prototype parameter set and distance function can be chosen in order to have flexible cluster shapes. However, the alternating optimization scheme, that can at least guarantee for some weak kind of convergence, can only be applied, when the corresponding distance function is differentiable. But even for differentiable distance functions we usually obtain equations for the prototypes that have no analytical solution (for instance [5]). This means that we have to cope with numerical problems and need in each iteration step a numerical solution of a coupled systems of non-linear equations. Other approaches try to optimize the objective function directly by evolutionary algorithms (for an overview see [10]). Nevertheless, fuzzy clustering approaches with distance functions that do not allow an analytical solution for the prototypes are usually very inefficient. In the following we introduce a new fuzzy clustering approach that admits also an analytical solution for the prototypes.

3 Clustering with Angle-Based Distances for Normalized Data

The idea of our approach is very similar to the original neural network competitive learning approach as it is for instance described in [14]. Instead of the Euclidean distance between a class representative and a given datum that Kohonen's self organizing feature maps use, the simple competitive learning approach computes the dot product of these vectors.

For normalized vectors the dot product is simply the cosine of the angle between the two vectors, i.e. the dot product is one if and only if the (normalized) vectors are identical, otherwise we obtain values between -1 and 1. Therefore, we define as the (modified) distance between a normalized prototype vector v and a normalized data vector x

$$d^2(v,x) \;=\; 1 - v^\top x. \tag{6}$$

Thus we have $0 \leq d^2(v,x) \leq 2$ and, in case of normalized vectors, $d^2(v,x) = 0 \Leftrightarrow x = v$.

Let us for the moment assume that the data vectors are already normalized. How we actually carry out the normalization will be discussed later on. With the distance function (6) the objective function (1) becomes

$$\begin{aligned} J(X,U,v) &= \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^m (1 - v_i^\top x_k) \\ &= \sum_{i=1}^{c}\sum_{k=1}^{n} \left(u_{ik}^m - u_{ik}^m \sum_{\ell=1}^{p} v_{i\ell} x_{k\ell} \right) \end{aligned}$$

where $v_{i\ell}$ and $x_{k\ell}$ is the ℓth coordinate/component of vector v_i and x_k, respectively. By taking into account the constraint that the prototype vectors v_i have to be normalized, i.e.

$$\| v_i \|^2 = \sum_{t=1}^{p} v_{it}^2 \;=\; 1 \tag{7}$$

we obtain the Lagrange function

$$L \;=\; \sum_{i=1}^{c}\sum_{k=1}^{n}\left(u_{ik}^{m} - u_{ik}^{m}\sum_{\ell=1}^{p} v_{i\ell}x_{k\ell}\right) + \sum_{s=1}^{c}\lambda_s\left(\sum_{t=1}^{p} v_{st}^{2} \;- 1\right). \tag{8}$$

The partial derivative of L w.r.t. $v_{i\ell}$ yields

$$\frac{\partial L}{\partial v_{il}} \;=\; -\sum_{k=1}^{n} u_{ik}^{m}x_{k\ell} \;+\; 2\lambda_i v_{i\ell}.$$

Since the first derivative has to be zero in a minimum, we obtain

$$v_{i\ell} \;=\; \frac{1}{2\lambda_i}\sum_{k=1}^{n} u_{ik}^{m}x_{k\ell}. \tag{9}$$

Making use of the constraint (7), we have

$$1 \;=\; \frac{1}{4\lambda_i^2}\sum_{\ell=1}^{p}\left(\sum_{k=1}^{n} u_{ik}^{m}x_{k\ell}\right)^{2},$$

which gives us

$$2\lambda_i \;=\; \sqrt{\sum_{\ell=1}^{p}\left(\sum_{k=1}^{n} u_{ik}^{m}x_{k\ell}\right)^{2}}$$

so that we finally obtain

$$v_{i\ell} \;=\; \frac{\sum_{k=1}^{n} u_{ik}^{m}x_{k\ell}}{\sqrt{\sum_{t=1}^{p}\left(\sum_{k=1}^{n} u_{ik}^{m}x_{kt}\right)^{2}}} \tag{10}$$

as the updating rule for the prototypes.

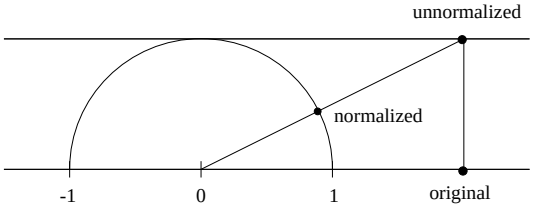

Fig. 1. Normalization of a datum

For this formula we have assumed that the data vectors are normalized. When we simply normalize the data vectors, we loose information, since collinear vectors are mapped to the same normalized vector. In order to avoid this effect

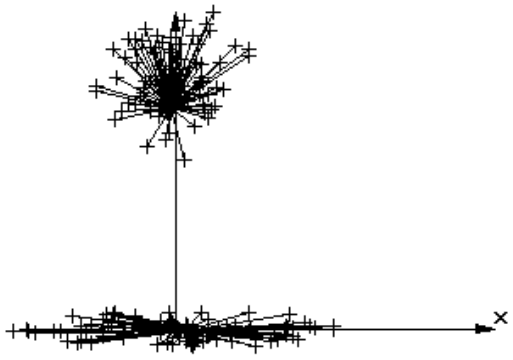

Fig. 2. Two clusters

we extend our data vectors by one component which we set one for all data vectors and normalize these $(p+1)$-dimensional data vectors. In this way, the data vectors in $\mathbb{R}^p$ are mapped to the upper half of the unit sphere in $\mathbb{R}^{p+1}$. Figure 1 illustrates the normalization for one-dimensional data.

Figure 2 shows a clustering result for a two-dimensional data set (i.e. the clustering is actually carried out on the normalized three-dimensional data). The membership degrees are not illustrated in the figure. We have connected each datum with the cluster centre (that we obtain by reversing the normalization procedure) to which it has the highest membership degree.

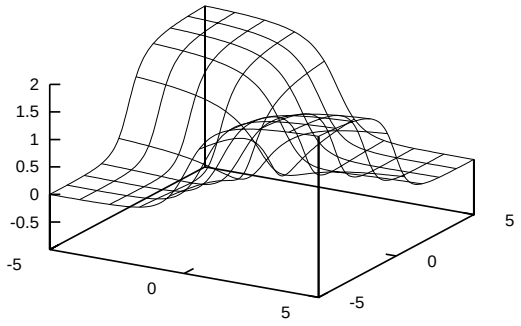

Fig. 3. The one-dimensional distance function

It can be seen in figure 2 that the prototype of the upper cluster is slightly lower than one might expect. The reason is that the distance function is not affine

invariant. We can already see in figure 2 that vectors near zero keep almost their Euclidean distance when we normalize them, whereas very long vectors are all mapped to the very lower part of the semi-circle.

Figure 3 shows distance values of two one-dimensional vectors. (The distance is computed for the normalized two-dimensional vectors.) Of course, the distance is zero at the diagonal and increases when we go away from the diagonal. But the distance is increasing very quickly with the distance to the diagonal near zero, whereas it increases slowly, when we are far away from the diagonal.

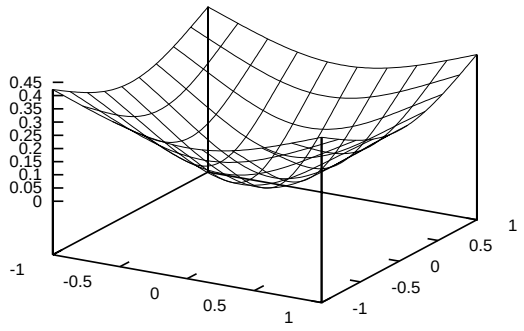

Fig. 4. Distance to the point $(0, 0)$

Figures 4 and 5 also illustrate this effect. In Figure 4 the distance to the (non-normalized) two-dimensional vector (cluster centre) $(0, 0)^\top$ is shown. It is a symmetrical distance function. However, when we replace the cluster centre $(0, 0)^\top$ by the vector $(1, 0)^\top$, we obtain the function in figure 5.

Here we can see that the distance is asymmetrical in the sense that it increases faster when we look in the direction of $(0, 0)^\top$. This can be an undesired effect for certain data sets. But there are also data sets for which this effect has a positive influence on the clustering result. Consider for instance data vectors with the annual salary of a person as one component. When we simply normalize each component, the effect is that a few outliers (persons with a very high income) force that almost all data are normalized to values very near to zero. This means that the great majority simply collapses to one cluster (near zero) and few outliers build single clusters. Instead of a standard normalization, we can also choose a logarithmic scale in order to avoid this effect. But the above mentioned clustering approach offers an interesting alternative.

Figure 6 shows a clustering result of data of bank customers with the attributes age, income, amount in depot, credit, and guarantees for credits. The number of clusters was automatically determined by a validity criterion, resulting in three clusters. The axes shown in the figure are credit, income, and amount

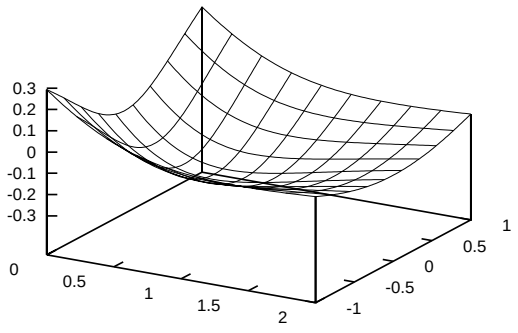

Fig. 5. Distance to the point $(1, 0)$

in depot. It is worth noticing that there is a compact cluster in the centre, representing the majority of average customers, whereas there are two other clusters covering custumers with high credit or a large amount of money, respectively.

4 Clustering with Angle-Based Distances for Non-normalized Data

In the previous section we have assumed that the data vectors are normalized or that we normalize them for the clustering. In this section we discuss what happens, when we refrain from normalizing the data vectors and the cluster centres. In order to avoid negative distances, we have to modify the distance function to

$$d^2(v, x) \;=\; (1 - v^\top x)^2. \tag{11}$$

The geometrical meaning of this distance function is the following. A datum x has distance zero to the cluster v, if and only if $v^\top x = 1$ holds. This equation describes a hyperplane, i.e. the hyperplane of all $x \in \mathbb{R}^p$ of the form

$$\frac{v}{\parallel v \parallel} + \sum_{s=1}^{p-1} \lambda_s w_s \tag{12}$$

where the vectors $w_1, \ldots, w_{p-1} \in \mathbb{R}^p$ span the hyperplane perpendicular to v and $\lambda_1, \ldots, \lambda_{p-1} \in \mathbb{R}$.

This means that we can find clusters in the form of linear varieties like the FCV algorithm. We will return to a comparison of FCV and this approach later on. Figure 7 shows the distance to the prototype $v^\top = (0.5, 0)$. This prototype describes the line

$$\begin{pmatrix} 2 \\ 0 \end{pmatrix} + \lambda \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

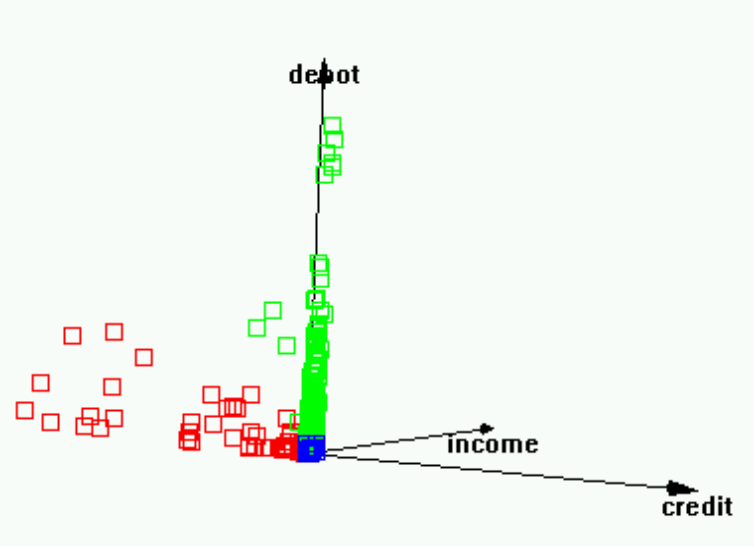

Fig. 6. Clustering result for bank customers

In order to derive equations for the prototypes we insert the distance function (11) into the objective function (1) and take the first derivative w.r.t. $v_{i\ell}$:

$$\frac{\partial J}{\partial v_{i\ell}} \;=\; -2\sum_{k=1}^{n} u_{ik}^{m}(1 - v_i^{\top} x_k)x_{k\ell}$$

These derivatives have to be zero at a minimum and we obtain the system of linear equations

$$\sum_{k=1}^{n} u_{ik}^{m}(1 - v_i^{\top} x_k)x_k \;=\; 0.$$

Making use of the fact that $(v_i^{\top} x_k)x_k \;=\; (x_k x_k^{\top})v_i$ holds, we obtain for the prototypes

$$v_i \;=\; \left(\sum_{k=1}^{n} u_{ik}^{m} x_k x_k^{\top}\right)^{-1} \sum_{k=1}^{n} u_{ik}^{m} x_k\,. \tag{13}$$

Note that the matrix $\sum_{k=1}^{n} u_{ik}^{m} x_k x_k^{\top}$ is the (weighted) covariance matrix (assuming mean value zero) and can therefore be inverted unless the data are degenerated.

An example of the detection of two linear clusters is shown in figure 8.

The difference of this approach to the FCV algorithm is in the computing scheme that requires inverting a matrix whereas for the FCV algorithm all eigenvalues and eigenvectors have to be computed. Another difference is caused by the non-Euclidean distance function that is again not affine invariant. Problems

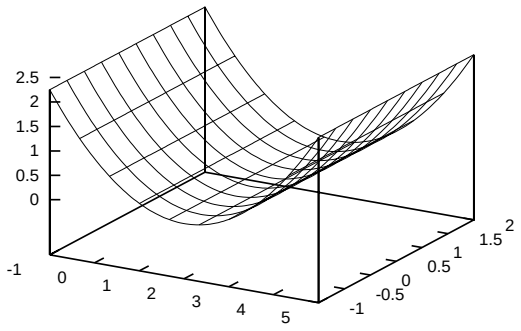

Fig. 7. Distance to $(0.5, 0)$

can arise when lines are near to $(0, 0)^\top$, since then the corresponding prototype vector v is very large, and even small deviations from the linear cluster lead to large distances. These problems are well known for other fuzzy clustering algorithms with non-Euclidean distance functions [11] and have to be treated in a similar way.

5 Conclusions

We have introduced fuzzy clustering algorithms using dot product-based distance functions that lead to new cluster shapes in the normalized case and to linear clusters in the non-normalized case. They represent a further extension of the already known objective function-based fuzzy clustering approaches.

References

1. Bezdek, J.C.: A Convergence Theorem for the Fuzzy ISODATA Clustering Algorithms. IEEE Trans. Pattern Analysis and Machine Intelligence **2** (1980) 1-8.
2. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981).
3. Bezdek, J.C., Hathaway, R.H., Sabin, M.J., Tucker, W.T.: Convergence Theory for Fuzzy c-Means: Counterexamples and Repairs. IEEE Trans. Systems, Man, and Cybernetics **17** (1987) 873-877.
4. Bock, H.H.: Clusteranalyse mit unscharfen Partitionen. In: Bock, H.H. (ed.). Klassifikation und Erkenntnis: Vol. III: Numerische Klassifikation. INDEKS, Frankfurt (1979), 137-163.
5. Davé, R.N.: Fuzzy Shell Clustering and Application to Circle Detection in Digital Images. Intern. Journ. General Systems **16** (1990) 343-355.
6. Davé, R.N.: Characterization and Detection of Noise in Clustering. Pattern Recognition Letters **12** (1991) 657-664.

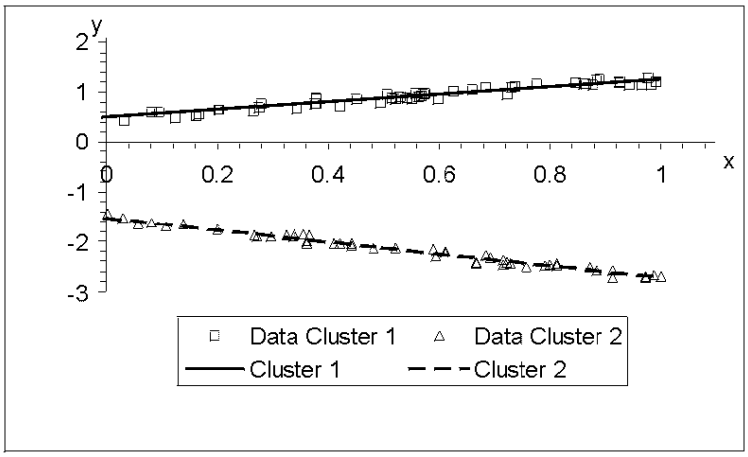

Fig. 8. Two linear clusters

7. Davé, R.N.: On Generalizing Noise Clustering Algorithms. In: Proc. 7th Intern. Fuzzy Systems Association World Congress (IFSA'97) Vol. III, Academia, Prague (1997), 205-210.
8. Gustafson, D.E., Kessel, W.C.: Fuzzy Clustering with a Fuzzy Covariance Matrix. Proc. IEEE CDC, San Diego (1979), 761-766.
9. Höppner, F. Klawonn, F., Kruse, R., Runkler, T.: Fuzzy Cluster Analysis. Wiley, Chichester (1999).
10. Klawonn, F., Keller, A.: Fuzzy Clustering with Evolutionary Algorithms. Intern. Journ. of Intelligent Systems **13** (1998) 975-991.
11. Krishnapuram, R., Frigui, H., Nasraoui, O.: Fuzzy and Possibilistic Shell Clustering Algorithms and Their Application to Boundary Detection and Surface Approximation – Part 1 & 2. IEEE Trans. on Fuzzy Systems **3** (1995) 29-60.
12. Krishnapuram, R., Keller, J.: A Possibilistic Approach to Clustering. IEEE Trans. Fuzzy Systems **1** (1993) 98-110.
13. Kruse, R., Gebhardt, J., Klawonn, F.: Foundations of fuzzy systems. Wiley, Chichester (1994).
14. Nauck, D., Klawonn, F., Kruse, R.: Neuro-Fuzzy Systems. Wiley, Chichester (1997).