Available online at www.sciencedirect.com

ScienceDirect

Computational Statistics & Data Analysis 51 (2006) 192 – 214

COMPUTATIONAL STATISTICS & DATA ANALYSIS

www.elsevier.com/locate/csda

Data analysis with fuzzy clustering methods

Christian Döring*, Marie-Jeanne Lesot, Rudolf Kruse

Department of Knowledge Processing and Language Engineering, Otto-von-Guericke-University of Magdeburg, Universitätsplatz 2, D-39106 Magdeburg, Germany

Available online 15 May 2006

Abstract

An encompassing, self-contained introduction to the foundations of the broad field of fuzzy clustering is presented. The fuzzy cluster partitions are introduced with special emphasis on the interpretation of the two most encountered types of gradual cluster assignments: the fuzzy and the possibilistic membership degrees. A systematic overview of present fuzzy clustering methods is provided, highlighting the underlying ideas of the different approaches. The class of objective function-based methods, the family of alternating cluster estimation algorithms, and the fuzzy maximum likelihood estimation scheme are discussed. The latter is a fuzzy relative of the well-known expectation maximization algorithm and it is compared to its counterpart in statistical clustering. Related issues are considered, concluding with references to selected developments in the area.

Keywords: Probabilistic and possibilistic cluster partitions; Objective function-based methods; Alternating cluster estimation; Fuzzy maximum likelihood estimation; Comparison with expectation maximization; Noise and outlier handling; Current research

1. Introduction

Cluster analysis is a technique for classifying data, i.e., to divide a given set of objects into a set of classes or *clusters* based on similarity. The goal is to divide the data set in such a way that cases assigned to the same cluster should be as similar as possible whereas two objects from different clusters should as dissimilar as possible. Thus one tries to model the human ability to group similar objects or cases into homogeneous classes or categories. The motivation for finding and building classes in this way can be manifold (Bock, 1974). Cluster analysis is primarily a tool for discovering previously hidden structure in the set of unordered objects. In that case one assumes that a “true” or natural grouping exists in the data. However, the assignment of objects to the classes and the description of these classes are unknown. By collecting similar objects into clusters one tries to reconstruct the unknown grouping in the hope that every found cluster represents an actual type or category of objects. Cluster analysis can as well be used as a method for data reduction. Then it is merely aiming at a simplified representation of the set of objects which allows for dealing with a manageable number of homogeneous groups instead of with a vast number of single objects. Only some mathematical criteria can decide on the composition of clusters when classifying data sets automatically. Therefore, clustering methods are endowed with distance functions that measure the dissimilarity of presented examples. As a result, one yields a partition of the data set into clusters regarding the chosen dissimilarity relation.

* Corresponding author. Tel.: +49 391 67 11358; fax: +49 391 67 12018.

E-mail addresses: doering@iws.cs.uni-magdeburg.de (C. Döring), lesot@iws.cs.uni-magdeburg.de (M.-J. Lesot), kruse@iws.cs.uni-magdeburg.de (R. Kruse).

0167-9473/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2006.04.030

This introductory article is organized as follows: Section 2 motivates the use of gradual assignments of data to clusters in cluster analysis. The introduced notion of membership degrees and fuzzy clusters is substantiated by the theory of fuzzy sets. Section 3 introduces the two major types of gradual assignments: the fuzzy memberships and the possibilistic membership degrees. Definitions and examples of the two kinds of fuzzy data partitions are given. Section 4 is dedicated to fuzzy clustering algorithms. Approaches that are based on a global criteria for optimality of clustering results are presented in Section 4.1. These algorithms minimize the objective functions they are based on using an alternating optimization (AO) scheme in order to yield either fuzzy (Section 4.1.1) or possibilistic clustering models (Section 4.1.2). The fuzzy and the possibilistic variants of the objective function-based algorithms are compared in Section 4.2. The alternating cluster estimation (ACE) framework is presented in Section 4.3. The ACE methodology allows the formulate a separate class of algorithms which is patterned on AO but abandons the formulation and minimization of criterion functions for clustering. Instead, the user is given the flexibility to choose desired fuzzy set shapes for the clusters that meet application-specific requirements. Fuzzy maximum likelihood estimation (FMLE) is the matter of Section 4.4 as well as a comparison to its close relative—the expectation maximization (EM) algorithm. Section 5 concludes the paper pointing to related issues and selected developments in the field.

2. What is fuzzy with fuzzy clustering?

Clustering methods can be distinguished regarding how they assign data to clusters, i.e., what type of partitions they form. In classical cluster analysis each datum must be assigned to exactly one cluster. These classical methods yield exhaustive partitions of the example set into non-empty and pairwise disjoint subsets. Such hard assignment of data to cluster can be inadequate in presence of data points that are equally distant to two or more clusters. Such special data points can represent hybrid-type or mixture objects which are equally similar to two or more types. A hard partition forces the full assignment of such data points to one of the clusters, although they should equally belong to all of them. For improved expressiveness of cluster solutions, some of these “crisp” cluster analysis techniques allow overlapping clusters. Then each datum must be assigned to at least one cluster, but with the possibility of simultaneous assignment to further clusters. Even though membership in multiple clusters is allowed, the hard cluster assignments in traditional clustering methods do not reflect the (un-)certainty with which data are assigned to the different classes.

Fuzzy cluster analysis therefore allows gradual memberships of data points to clusters in [0, 1]. This gives the flexibility to express that data points belong to more than one cluster at the same time. Furthermore, these membership degrees offer a much finer degree of detail of the data model. Aside from assigning a data point to clusters in (equal) shares, membership degrees can also express how ambiguously or definitely a data point should belong to a cluster. The concept of these membership degrees is substantiated by the definition and interpretation of fuzzy sets. A *fuzzy set* μ of a set X is a mapping (Zadeh, 1965):

$$\mu : X \to [0, 1]. \tag{1}$$

Let μ_A be a fuzzy set of the set $\mathscr{X}$ and let B be a classical subset of $\mathscr{X}$, $B \subset \mathscr{X}$. Then the value $\mu_A(x)$ is called grade of membership or *membership degree* of x to μ_A. It specifies to what extent the object x satisfies some imprecise property A modeled by μ_A (or to what degree it fits to a vague concept A defined by its corresponding membership function μ_A). While for the conventional subset B some object x either belongs to B ($x \in B$) or not ($x \notin B$), the fuzzy set of μ_A allows smooth transitions for the membership of the elements of its universe $\mathscr{X}$ between $\mu_A(x) = 1$ (full membership) and $\mu_A(x) = 0$ (not a member). A value of $\mu_A(x)$ close to 0 means a low degree of membership, a value close to 1 means high degree of belonging of x to the fuzzy set. From the definition it is obvious that fuzzy sets are actually functions. They can be seen as generalization of characteristic functions of classical sets. Characteristic functions indicate membership to the set with 1, and otherwise with 0, if the given element is not a member of the set. Fuzzy sets, however, allow all degrees of belonging in [0, 1]. In applications of fuzzy sets their values lying in the unit interval are either defined by people who use fuzzy sets for modeling vague concepts (like linguistic terms) or they come from data. Then the membership functions are fitted to the data for obtaining hopefully accurate and useful models of the process from whence the data are derived.

In fuzzy clustering the memberships of data points to clusters have been fuzzified to allow fine grained solution spaces in form of fuzzy partitions of the set of given examples $\mathbf{X} = \{\vec{x}_1, \dots, \vec{x}_n\}$. Whereas the clusters Γ_i of data partitions have been classical subsets in traditional methods, they are now represented by the fuzzy sets μ_{Γ_i} of the data set $\mathbf{X}$. Complying to fuzzy set theory, the cluster assignment u_{ij} is now the membership degree of a datum $\vec{x}_j$ to

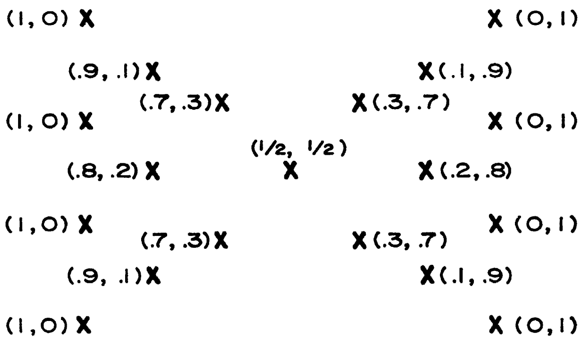

Fig. 1. The first example for a fuzzy partitioning by Ruspini (1969).

cluster Γ_i, such that: $u_{ij} = \mu_{\Gamma_i}(\vec{x}_j) \in [0, 1]$. Since memberships to clusters are fuzzy, there is not a single label that is indicating to which cluster a data point belongs to. Instead, fuzzy clustering methods associate a fuzzy label vector to each data point $\vec{x}_j$ that states its memberships to the c clusters:

$$\vec{u}_j = \left(u_{1j}, \ldots, u_{cj}\right)^{\mathrm{T}}. \tag{2}$$

The $c \times n$ matrix $\mathbf{U} = \left(u_{ij}\right) = (\vec{u}_1, \ldots, \vec{u}_n)$ is then called a fuzzy partition matrix. There are two commonly used types of fuzzy data partitions that can be distinguished by the set of constraints that the gradual memberships have to satisfy. The definitions and differentiated interpretations of the partition types are given in the next section.

Example. Ruspini (1969) was the first researcher who suggested the use of fuzzy sets in clustering in 1969. His idealized and first example of a fuzzy classification of the simple "butterfly" data set with two clusters is shown in Fig. 1 (it is kind of the butterfly that started everything). The data points in the figure are associated with their label vectors, which state the membership degrees to the left and to the right cluster. The "cores" of the left and the right cluster are the outer left and the outer right column of examples, respectively. Ruspini (1969) noted the following advantages over a conventional clustering representation: points in the center of a cluster can have a degree equal to 1, while boundary points between the core and some other class can be identified as such (i.e., their membership degree to the class they are closer to is not equal to 1). "Bridges" or stray points may be classified as undetermined with a degree of indeterminacy proportional to their similarity to core points. The characteristics of the data set in Fig. 1 help to depict the higher expressiveness of fuzzy partitions. The equidistant data point in the middle of the figure would have to be arbitrarily assigned with full weight to one of the clusters if classical ("crisp") partitions were allowed only. In this fuzzy partition, however, it can be associated with the equimembership vector $(0.5, 0.5)^{\mathrm{T}}$. Crisp data partitions can further not express the difference between data points in the center and those that are rather at the boundary of a cluster. Both kind of points would be fully assigned to the cluster they are most similar to.

3. Fuzzy partitions

Fuzzy data partitions are meant to provide a much richer means for representing cluster structure. The fuzzy clustering methods described in the next section are thus enabled to find more realistic models, since boundaries between many classes are in fact very badly delineated (i.e., really fuzzy). In the field of fuzzy clustering two types of fuzzy cluster partitions have evolved. They differ in the constraints they place on the membership degrees and how the membership values should be interpreted. We begin our discussion with the most widely used type, the probabilistic fuzzy partitions, since they have been proposed first. Notice that in literature they are sometimes just called fuzzy partitions (dropping the word "probabilistic"). We use the subscript f for the probabilistic fuzzy approaches and, in the sequel, p for the possibilistic models. Latter constitute the second type of fuzzy partitions. Both can be seen as generalizations of classical (hard) data partitions.

Definition 3.1 (*Probabilistic fuzzy cluster partition*). Let $\mathbf{X} = \{\vec{x}_1, \ldots, \vec{x}_n\}$ be the set of given examples and let c be the number of clusters $(1 < c < n)$ represented by the fuzzy sets μ_{Γ_i}, $(i = 1, \ldots, c)$. Then we call $\mathbf{U}_f = \left(u_{ij}\right) = \left(\mu_{\Gamma_i}\left(\vec{x}_j\right)\right)$

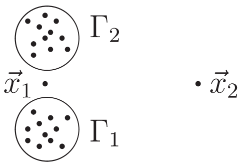

Fig. 2. A situation in which the probabilistic assignment of membership degrees is counterintuitive for datum $\vec{x}_2$.

a (probabilistic) fuzzy cluster partition of **X** if

$$\sum_{j=1}^{n} u_{ij} > 0 \quad \forall i \in \{1, \ldots, c\} \tag{3}$$

and

$$\sum_{i=1}^{c} u_{ij} = 1 \quad \forall j \in \{1, \ldots, n\} \tag{4}$$

hold. We interpret $u_{ij} \in [0, 1]$ as the membership degree of datum $\vec{x}_j$ to cluster Γ_i relative to all other clusters.

Constraint (3) guarantees that no cluster is empty. This corresponds to the requirement in classical cluster analysis that no cluster, a subset of **X**, is empty. Condition (4) ensures that the sum of the membership degrees for each datum equals 1. This means that each datum receives the same weight in comparison to all other data and, therefore, that all data are (equally) included into the cluster partition. This is related to the requirement in classical clustering that partitions are formed exhaustively. As a consequence of both constraints no cluster can contain the full membership of all data points.

Example. Ruspini's fuzzy classification of the "butterfly" data set in the previous section (in tandem with its intended meaning) is a (probabilistic) fuzzy data partition.

Condition (4) corresponds to a normalization of the memberships per datum. Thus the membership degrees for a given datum *formally resemble* the probabilities of its being a member of the corresponding cluster. Although often desirable, the "relative" character of the probabilistic membership degrees can be misleading (Timm et al., 2004). Fairly high values for the membership of datum in more than one cluster can lead to the impression that the data point is typical for the clusters. But this is not always the case. Consider, for example, the simple case of two clusters shown in Fig. 2. Datum $\vec{x}_1$ has the same distance to both clusters and thus it is assigned a membership degree of about 0.5. This is plausible. However, the same degrees of membership are assigned to datum $\vec{x}_2$ even though this datum is further away from both clusters and should be considered less typical. Because of the normalization, however, the sum of the memberships has to be 1. Consequently, $\vec{x}_2$ receives fairly high membership degrees to both clusters. For a correct interpretation of these memberships one has to keep in mind that they are rather degrees of sharing than of typicality, since the constant weight of 1 given to a datum must be distributed over the clusters. A better reading of the memberships, avoiding misinterpretations, would be (Höppner et al., 1999): "If the datum $\vec{x}_i$ has to be assigned to a cluster, then with the probability u_{ij} to the cluster i."

Probabilistic memberships are somewhat counterintuitive and do not reflect how typical the data point is, especially if the data point is further away from just one cluster or even the bulk of all clusters. Dropping the normalization constraint (4) in Definition 3.1 would allow more intuitive membership degrees for these data points, which could just depend on their similarity (or typicality) to a cluster. In that case datum $\vec{x}_2$ in Fig. 2 could receive lower degrees of membership to both clusters than the equimembership of 0.5 of $\vec{x}_1$, since this datum is further away from both clusters and therefore less representative for both of them.

Definition 3.2 (*Possibilistic cluster partition*). Let $\mathbf{X} = \{\vec{x}_1, \ldots, \vec{x}_n\}$ be the set of given examples and let c be the number of clusters $(1 < c < n)$ represented by the fuzzy sets μ_{Γ_i}, $(i = 1, \ldots, c)$. Then we call $\mathbf{U}_p = \left(u_{ij}\right) = \left(\mu_{\Gamma_i}\left(\vec{x}_j\right)\right)$

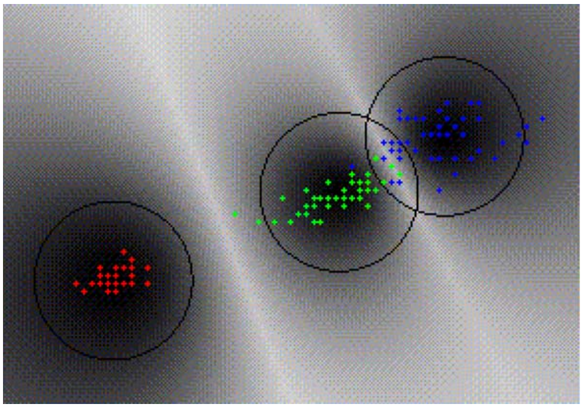

Fig. 3. Iris data set classified with probabilistic fuzzy c-means algorithm. Attributes petal length and petal width.

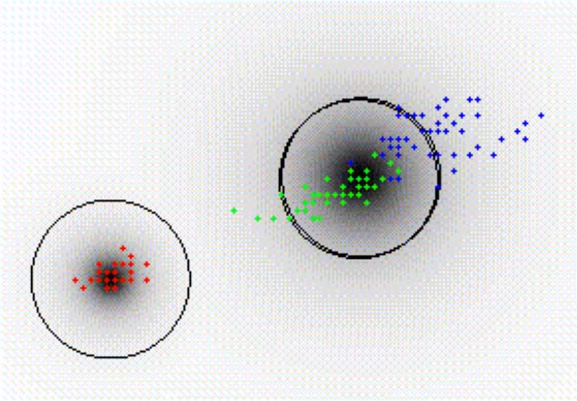

Fig. 4. Iris data set classified with possibilistic fuzzy c-means algorithm. Attributes petal length and petal width.

a possibilistic cluster partition of **X** if

$$\sum_{j=1}^{n} u_{ij} > 0 \quad \forall i \in \{1, \ldots, c\} \tag{5}$$

holds. We interpret $u_{ij} \in [0, 1]$ as the degree of representativity or typicality of the datum $\vec{x}_j$ to cluster Γ_i.

With this definition one tries to achieve a more intuitive assignment of degrees of membership avoiding the undesirable normalization effect discussed above. The membership degrees for one datum now *resemble* the possibility (in the sense of possibility theory, Dubois and Prade, 1988) of its being a member of the corresponding cluster (Krishnapuram and Keller, 1993; Davé and Krishnapuram, 1997).

Example. Fig. 3 illustrates a probabilistic and Fig. 4 shows a possibilistic classification of the Iris data set (Fisher, 1936; Blake and Merz, 1998). The gray scale indicates the membership to the closest cluster. While probabilistic memberships rather divide the data space, possibilistic membership degrees only depend on the typicality to the respective closest clusters. The displayed partitions of the data set are optimal in the sense of being quite good solutions of the clustering problem. Finding good cluster partitions is the matter of the next section.

4. Fuzzy clustering algorithms

There is a large variety of methods that have been proposed in an even larger amount of papers and books all aiming at finding the fuzzy clusters that might exist in some given example set. For the presentation of such a wide field of methods, it is fruitful to focus on their underlying ideas, since the methods can be put into different classes regarding the principles they are based on. It should then be sufficient to present most prominent representatives of each class in detail, and to refer to the literature for the specialties.

At first we take a closer look on methods that try to find a good fuzzy partition and cluster prototypes (that represent the clusters) using global criteria for optimality in form of an *objective function*. The clustering task can then be formulated as a function optimization problem. The objective function depends on both the cluster prototypes and the memberships of data points to the clusters. It cannot be optimized directly and therefore an AO scheme is usually applied that optimizes one group of parameters (e.g., the membership degrees) holding the other group (e.g., the prototypes) fixed and vice versa. This iterative updating scheme (grouped coordinate descent) is repeated in the hope to approach the global optimum of the criterion function. The user then yields a fuzzy data partition and descriptions of clusters that are considered optimal regarding the chosen objective function.

The AO scheme for optimizing an objective function gave rise to the second class of methods called ACE. This family of algorithms is presented secondly. They generalize the AO updating scheme and the user can choose among multiple update equations for the prototypes and membership degrees. Consequently, the iterated steps for estimating the cluster prototypes and the memberships do not necessarily reflect the optimization of a particular criterion function. In this way, the user is given more freedom to use prototypes that satisfy some desirable properties as well as to choose membership functions for the clusters that have better suited shapes for a particular application. Solving the clustering problem within this framework trades in mathematical interpretability for having the flexibility of tailoring new clustering algorithms for specific needs. ACE is justified by the observation that convergence is seldom a problem in practical examples (local minima or saddle points can be avoided). It should be explicitly stated that the ACE framework encompasses all those algorithms that follow from the minimization of objective functions as long as their respective update equations are chosen. These follow from the necessary conditions for an optimum of the objective function.

The large variety of fuzzy clustering approaches exists due to modifications of the objective functions. These modifications are aiming at improvements of the results with respect to particular problems (e.g., noise, outliers; see Section 5.2). Other objective functions are tailored for specific applications. In this section we present the unmodified forms of the objective functions. We thereby focus on the differences between probabilistic and possibilistic variants of the algorithms. They are likewise regarded in the following subsections. Since we want to highlight the cross-references to statistical clustering approaches, we discuss the FMLE algorithm in the third subsection. It is a relative of the EM algorithm. FMLE is widely known in the fuzzy community and lies on the boundary to "strongly" probabilistic clustering as it is pursued in the statistical community. We discuss whether this cross-fertilization led to a well performing clustering approach by citing some empirical evidence that we gathered in Döring et al. (2004).

4.1. Objective function-based algorithms

In objective function-based clustering each cluster is represented by a *cluster prototype*. This prototype consists of a *cluster center* (whose name already indicates its meaning) and maybe some additional information about the size and the shape of the cluster. The cluster center is an instantiation of the attributes used to describe the domain, just as the data points in the data set to divide. However, the cluster center is computed by the clustering algorithm and may or may not appear in the data set. The size and shape parameters determine the extension of the cluster in different directions of the underlying domain.

The degrees of membership to which a given data point belongs to the different clusters are computed from the distances (i.e., dissimilarity) of the data point to the cluster centers w.r.t. the size and the shape of the cluster (as stated by the additional prototype information). Since clusters are required to be as homogeneous as possible, the problem to divide a data set $\mathbf{X} = \{\vec{x}_1, \ldots, \vec{x}_n\} \subseteq \mathbb{R}^p$ into c clusters can be stated as to assign data to clusters such a way that the sum of the (squared) distances between the clusters (i.e., their prototypes) and the data points assigned to them is minimal. This is the basic idea of the applied objective functions J in fuzzy clustering.

4.1.1. *Probabilistic fuzzy clustering*

Most probabilistic fuzzy clustering algorithms that determine an optimal (probabilistic) fuzzy partition of a given data set $\mathbf{X}$ into c clusters minimize the objective function (Bezdek, 1973):

$$J_f\left(\mathbf{X}, \mathbf{U}_f, \mathbf{C}\right) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m d_{ij}^2, \tag{6}$$

subject to the constraints (3) and (4) of the probabilistic membership degrees in $\mathbf{U}_f$. d_{ij} is the distance between datum x_j and cluster i. The set of clusters $\mathbf{C}$ comprises all properties of the clusters by stating location parameters (i.e., the cluster center $\vec{c}_i$) and maybe size and shape parameters for each cluster. The condition (3) avoids the trivial solution of minimization problem ($u_{ij} = 0$ for all $i \in \{1, \ldots, c\}$ and $j \in \{1, \ldots, n\}$). The partitioning property of any probabilistic clustering algorithm, which "distributes" the weight of a datum to the different clusters, is due to the normalization constraint (4). The parameter m, $m > 1$, is called the *fuzzifier* or *weighting exponent*.

Strictly speaking, the objective function-based methods use a generalization of the least-squared error functional that has already been known from the hard c-means algorithm. This algorithm and its target function, however, lead to hard ("crisp") assignments of data points to the clusters (Duda and Hart, 1973). Dunn introduced a fuzzy version of the c-means algorithm in order to treat data belonging to several clusters accordingly (Dunn, 1974). He showed that for $m = 1$ cluster assignments are hard when minimizing J_f even though they are not constrained in $\{0, 1\}$. Thus he proposed to weight the memberships with u_{ij}^2, since this function of the u_{ij} leads to the desired fuzzification of the resulting data partition. The generalization for exponents $m > 1$ has been proposed in Bezdek (1973), but usually $m = 2$ is chosen. m determines the "fuzziness" of the classification: with higher values for m the boundaries between clusters become softer, with lower values they get harder.

Unfortunately, the objective function J_f cannot be minimized directly. Therefore, an iterative algorithm is used, which alternately optimizes the membership degrees and the cluster parameters. That is, first the membership degrees are optimized for fixed cluster parameters, then the cluster parameters are optimized for fixed membership degrees:

$$\mathbf{U}_t = j_U\left(\mathbf{C}_{t-1}\right), \quad t > 0 \tag{7}$$

and

$$\mathbf{C}_t = j_C\left(\mathbf{U}_t\right). \tag{8}$$

The main advantage of this scheme is that in each of the two steps the optimum can be computed directly. By iterating the two steps the joint optimum is approached (although it cannot be guaranteed that the global optimum will be reached—the algorithm may get stuck in a local minimum of the objective function J_f).

The update formulae j_U and j_C are derived by simply setting the derivative of the objective function J_f w.r.t. the parameters to optimize equal to zero (taking into account the constraint (4)). Independent of the chosen distance measure the following update formula for the membership degrees is obtained from J_f (Höppner et al., 1999):

$$u_{ij} = \frac{d_{ij}^{-2/(m-1)}}{\sum_{t=1}^{c} d_{tj}^{-2/(m-1)}}. \tag{9}$$

This update equation clearly shows the relative character of the (probabilistic) fuzzy membership degree. It depends not only on the distance of the datum $\vec{x}_j$ to cluster i, but also on the distances between this data point and other clusters.

The update formulae j_C for the cluster parameters depend, of course, on the parameters used to describe a cluster (location, shape, size) and on the chosen distance measure. Therefore, a general update formula cannot be given. For the "classical" objective function-based algorithms (and their specific cluster parameters as well as distance measures) the respective update equations are developed later. Notice that the clustering models can, of course, not only be optimized using AO algorithms. Reformulation of the criterion function (Hathaway and Bezdek, 1995; Bezdek et al., 1999a) and the optimization with genetic algorithms (Babu and Murty, 1994; Klawonn and Keller, 1998) are possible as well.

4.1.2. *Possibilistic fuzzy clustering*

In possibilistic fuzzy clustering one tries to achieve a more intuitive assignment of degrees of membership by dropping the normalization constraint (4), which allows the memberships to express typicality. However, this leads to the mathematical problem that the above objective function is now minimized by assigning $u_{ij}=0$ for all $i \in \{1, \dots, c\}$ and $j \in \{1, \dots, n\}$, i.e., data points are not assigned to any cluster and all clusters are empty. In order to avoid this trivial solution, a penalty term is introduced, which forces the membership degrees away from zero. That is, the objective function J_f is modified to

$$J_p\left(\mathbf{X}, \mathbf{U}_p, \mathbf{C}\right) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m d_{ij}^2 + \sum_{i=1}^{c} \eta_i \sum_{j=1}^{n} \left(1 - u_{ij}\right)^m, \tag{10}$$

where $\eta_i > 0$ $(i = 1, \dots, c)$ (Krishnapuram and Keller, 1993). The first term leads to a minimization of the weighted distances. The second term suppresses the trivial solution since this sum rewards high memberships (close to 1) that make the expression $\left(1 - u_{ij}\right)^m$ becomes approximately 0. Thus, the desire for (strong) assignments of data to clusters is expressed in the objective function J_p. In tandem with the first term the high membership can be expected especially for data that are close to their clusters, since with high degree of belonging the weighted distance to a closer cluster is smaller than to clusters further away. The cluster specific constants η_i are used to balance the contrary objectives expressed in the two terms of J_p. It is a reference value stating at what distance to a cluster a data point should receive higher membership to it. These considerations mark the difference to probabilistic clustering approaches. While in probabilistic fuzzy clustering each data point has a constant weight of 1, possibilistic clustering methods have to learn the weights of data points.

The formula for updating the membership degrees that is derived from the possibilistic objective function J_p by setting its derivative to zero is (Krishnapuram and Keller, 1993)

$$u_{ij} = \frac{1}{1 + \left(d_{ij}^2 \big/ \eta_i\right)^{1/(m-1)}}. \tag{11}$$

First of all, this update equation clearly shows that the membership of a datum $\vec{x}_j$ to cluster i depends only on its distance d_{ij} to this cluster. Small distance corresponds to high degree of membership, whereas larger distances (i.e., strong dissimilarity) results in low membership degrees. Thus, the u_{ij} have typicality interpretation.

Eq. (11) further helps to explain the parameters η_i of the clusters. Considering the case $m = 2$ and substituting η_i for d_{ij}^2 yields $u_{ij} = 0.5$. It becomes obvious that η_i is a parameter that determines the distance to the cluster i at which the membership degree should be 0.5. Since that value of membership can be seen as definite assignment to a cluster, the permitted extension of the cluster can be controlled with this parameter. Depending on the clusters' shape the η_i have different geometrical interpretation. If hyper-spherical clusters are considered, $\sqrt{\eta_i}$ is their mean diameter. In shell clustering $\sqrt{\eta_i}$ corresponds to the mean thickness of the contours described by the cluster prototype information (Höppner et al., 1999). If such properties of the clusters to search for are known prior to the analysis of the given data, η_i can be set to the desired value. If all clusters have the same properties, the same value can be chosen for all clusters. However, the information on the actual shape property described by η_i is often not known in advance. In that case these parameters must be estimated. Good estimates can be found using a probabilistic fuzzy clustering model of the given data set. The η_i are then estimated by the fuzzy intra-cluster distance using the memberships matrix $\mathbf{U}_f$ as it has been determined by the probabilistic counterpart of the chosen possibilistic algorithm (Krishnapuram and Keller, 1993). That is, for all clusters $(i = 1, \dots, n)$:

$$\eta_i = \frac{\sum_{j=1}^{n} u_{ij}^m d_{ij}^2}{\sum_{j=1}^{n} u_{ij}^m}. \tag{12}$$

Update equations j_C for the prototypes are as well derived by simply setting the derivative of the objective function J_p w.r.t. the prototype parameters to optimize equal to zero (holding the membership degrees $\mathbf{U}_p$ fixed). Looking at both objective functions it can be inferred that the update equations for the cluster prototypes in the possibilistic algorithms must be identical to their probabilistic counterparts. This is due to the fact that the second additional term in J_p vanishes in the derivative for fixed (constant) memberships u_{ij}. As already mentioned, several fuzzy clustering algorithms can be distinguished depending on the additional size and shape information contained in the cluster prototypes, the way in which the distances are determined. The most prominent and widely used algorithms are described in the next paragraphs altogether with their prototypes, capabilities, and typical practical application.

4.1.3. Classical fuzzy AO algorithms

Fuzzy c-means (*FCM*) *algorithm*: The FCM algorithm recognizes a known or given number of c hyper-spherical clouds of points in a given p-dimensional data set $\mathbf{X}=\{\vec{x}_1,\ldots,\vec{x}_n\}\subseteq\mathbb{R}^p$. The clusters are assumed to be of approximately the same size. The prototypes of the FCM model are rather simple. Each cluster is only represented by its center $\vec{c}_i$. The Euclidean distance between a datum and a prototype is used as a measure for dissimilarity of a data point to a cluster. The minimization of either objective function from above (Eq. (6) or (10)) with respect to the cluster prototypes leads to (Höppner et al., 1999):

$$\vec{c}_i=\frac{\sum_{j=1}^{n}u_{ij}^{m}\vec{x}_j}{\sum_{j=1}^{n}u_{ij}^{m}}. \tag{13}$$

The choice of the optimal cluster center points for given memberships of the data to the clusters has the form of a generalized mean value computation from which the algorithm has its name. Given this update equations for the cluster centers and using either Eq. (9) or (11) for the updates of the memberships leads to the specific AO scheme of the probabilistic or the possibilistic FCM, respectively.

The general form of the AO scheme of coupled equations (7) and (8) starts with an update of the membership matrix in the first iteration of the algorithm ($t=1$). The first calculation of memberships is based on an initial set of prototypes $\mathbf{C}_0$. Even though the optimization of an objective function could mathematically also start with an initial but valid membership matrix (in the sense of Definition 3.1 or 3.2), a $\mathbf{C}_0$ initialization is easier and therefore common practice in all fuzzy clustering methods. Basically, FCM can be initialized by prototypes that have been randomly placed in the input space. For a better coverage of the data space with initial clusters, however, Latin hyper-cube sampling is recommendable especially for more complex algorithms (McKay et al., 1979). It is further common practice to start more complicated clustering methods with initial prototypes that result from simpler algorithms. This approach acknowledges the fact that the optimization of complex models is rather sensitive to their initialization. The higher number of prototype parameters to optimize in complex models leads to a higher vulnerability to get stuck in a local minimum of an objective function. Thus, the probability to yield only sub-optimal clustering results can be reduced. The repetitive updating in the AO scheme can be stopped if the number of iterations t exceeds some predefined number of maximal iterations $t_{\max}$. However, AO is usually stopped when the changes in the prototypes are smaller than some termination accuracy. In analogy to the initialization, termination of AO is better implemented using the prototypes. Detecting the stabilization of prototypes $\mathbf{C}$ requires less comparisons than measuring the change in the partition matrix $\mathbf{U}$.

The *probabilistic* FCM algorithm is known as a stable, reliable, and fast classification method. It is quite insensitive to its initialization and it is not likely to get stuck in a local minimum of its objective function in practice. Due to its simplicity and low computational demands, the probabilistic FCM is a widely used initializer for other clustering methods. On the theoretical side, it has been proved that either the iteration sequence itself or any convergent subsequence of the probabilistic FCM converges in a saddle point or a minimum—but not in a maximum—of the objective function (Bezdek, 1981). For a number of other clustering techniques convergence proofs were also provided, but unfortunately no general result on the convergence for all probabilistic techniques that are based on AO of J_f is known at the moment (Höppner et al., 1999).

Gustafson–Kessel (*GK*) *algorithm*: In this locally adaptive fuzzy clustering algorithm the clusters prototypes are endowed with a fuzzy covariance matrix in addition to the center vectors for detecting ellipsoidal clusters, $\mathbf{C}=\{C_i\mid C_i=\{\vec{c}_i,\Sigma_i\},i=1,\ldots,c\}$ (Gustafson and Kessel, 1979). The eigenstructure of the positive-definite $p\times p$ matrices Σ_i represents the shape of the cluster i. The sizes of the clusters, if known in advance, can be controlled using the constants $\varrho_i>0$ demanding that $\det(\Sigma_i)=\varrho_i$. Usually, the clusters are assumed to be of equal size setting $\det(\Sigma_i)=1$. Since each cluster can have its special size and shape, the distance of a datum to a particular cluster has to take into account its specific expansion. Consequently, the dissimilarity is calculated respecting the parameters in C_i:

$$d^2\left(\vec{x}_j,C_i\right)=\det\left(\Sigma_i\right)^{1/p}\left(\vec{x}_j-\vec{c}_i\right)^{\mathrm{T}}\Sigma_i^{-1}\left(\vec{x}_j-\vec{c}_i\right). \tag{14}$$

Using this distance measure in J_f or J_p, the objective function also depends on the parameters C_i. The respective update equations derived from differentiation of the objective function lead to Eq. (13) for the center vectors

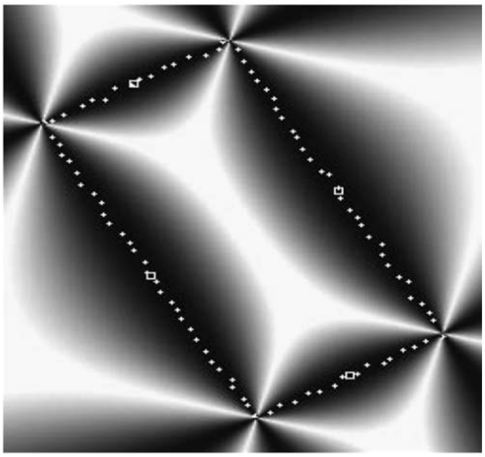

Fig. 5. FCV analysis.

and to:

$$\Sigma_i = \frac{\sum_{j=1}^{n} u_{ij} \left(\vec{x}_j - \vec{c}_i\right) \left(\vec{x}_j - \vec{c}_i\right)^{\mathrm{T}}}{\sum_{j=1}^{n} u_{ij}} \tag{15}$$

for the fuzzy covariance matrix of cluster i. Of course, initialization of the GK algorithm with a few iterations of the probabilistic FCM is strongly recommended. Compared to FCM the algorithm of Gustafson and Kessel exhibits higher computational demands due to the matrix inversions. A restriction to axis-parallel cluster shapes reduces computational costs. Axis-parallel ellipsoids are usually preferred when clustering is applied for the generation of fuzzy rule systems (Höppner et al., 1999).

Other non-point-prototypes clustering models: There is a large variety of models with more complex prototypes. *Solid* clustering algorithms search for “cloud-like” clusters such as the GK algorithm. They are mainly of interest in data analysis applications. Another area of application of fuzzy clustering algorithms is image recognition and analysis. So-called *shell* clustering algorithms are used for segmentation and the detection of special geometrical contours in images such as borders of circles and ellipses. All of the following examples are objective function-based algorithms and have probabilistic as well as possibilistic variants (Figs. 5–10) (Höppner et al., 1999). However, the used distance measure has been modified for detecting the special cluster forms. The fuzzy c-varieties (FCV) and the fuzzy c-elliptotypes algorithm are able to recognize lines, planes or hyper-planes (Fig. 5). This algorithm can also be used for the construction of locally linear models of data with underlying functional interrelations. The adaptive fuzzy c-elliptotypes (AFCE) algorithm assigns disjoint line segments to different clusters (Fig. 6). Circle contours can be detected by the fuzzy c-shells and the fuzzy c-spherical shells algorithm. Since objects with circle shaped boundaries in 3D are projected into the picture plane the recognition of ellipses can be necessary. The fuzzy c-ellipsoidal shells algorithm is able to solve this problem. The fuzzy c-quadric shells algorithm (FCQS) is furthermore able to recognize hyperbolas, parabolas or linear clusters. Its flexibility can be observed in Figs. 7 and 8. The shell clustering techniques have also been extended to non-smooth structures such as rectangles and other polygons. See Figs. 9 and 10 for examples of the fuzzy-c-rectangular (FCRS) and fuzzy-c-2-rectangular shells (FC2RS) algorithm. The interested reader may be referred to Höppner et al. (1999) and Bezdek et al. (1999a) for a complete discussion of this branch of methods.

4.2. Possibilistic vs. probabilistic models and algorithms

Aside from the different interpretation of memberships, there are some general properties that distinguish the results of the possibilistic and probabilistic fuzzy clustering approaches.

Example. Figs. 3 and 4 illustrate a probabilistic and a possibilistic FCM classification of the Iris data set into three clusters (Fisher, 1936; Blake and Merz, 1998). The displayed partitions of the data set are the result of alternatingly

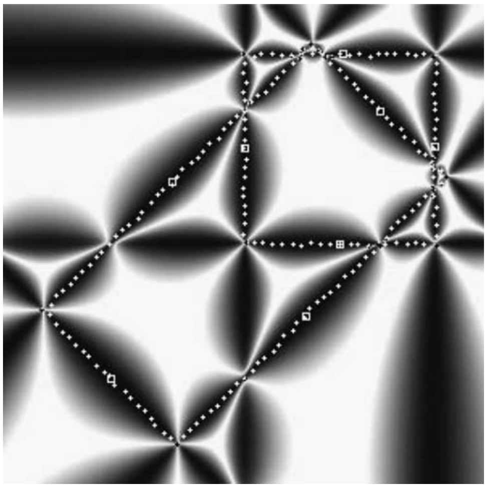

Fig. 6. AFCE analysis.

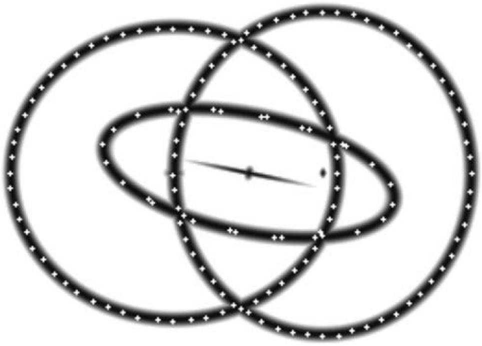

Fig. 7. FCQS analysis.

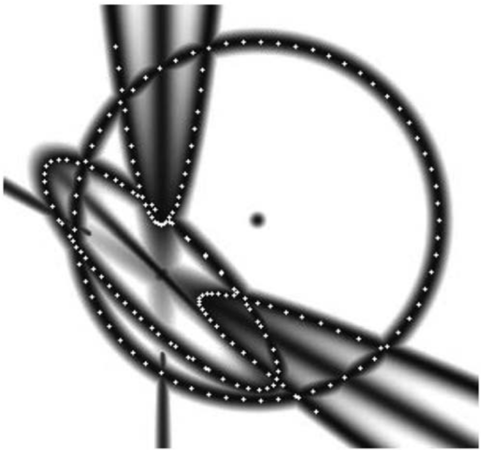

Fig. 8. FCQS analysis.

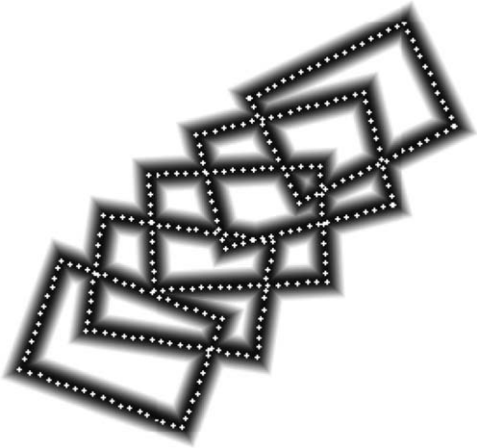

Fig. 9. FCRS analysis.

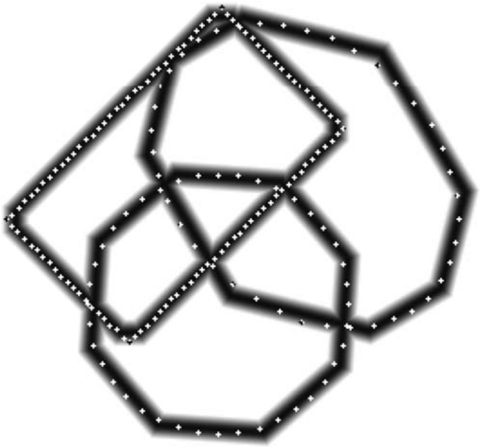

Fig. 10. FC2RS analysis.

optimizing J_f and J_p (Timm et al., 2004). On the left, the data set is divided into three clusters. On the right, the possibilistic FCM algorithm detects only two clusters, since the two of the three clusters in the upper right of Fig. 4 are identical. Note that this behavior is specific to the possibilistic approach. In the probabilistic counterpart the cluster centers are driven apart, because a cluster, in a way, "seizes" part of the weight of a datum and thus leaves less that may attract other cluster centers. Hence sharing a datum between clusters is disadvantageous. In the possibilistic approach there is nothing that corresponds to this effect.

Cluster coincidence: One of the major characteristics in which the approaches differ lies in the fact that probabilistic algorithms are forced to partition the data while the corresponding possibilistic approaches are not compelled to do so. The former distribute the total membership of the data points (sums up to one), whereas the latter are rather required to determine the data point weights by themselves. Probabilistic algorithms attempt to cover all data points with clusters, since sharing data point weight is disadvantageous. In the possibilistic case, there is no interaction between clusters. Thus, the found clusters in possibilistic models can be located much closer to each other than in a probabilistic clustering. Clusters can even coincide, which has been widely observed (Barni et al., 1996; Krishnapuram and Keller, 1996). This leads to solutions in which one cluster being actually present in a data set can be represented by two clusters in the possibilistic model. In worse cases there is data left in other regions of the input space that has cluster structure, but which is not covered by clusters in the model. Then possibilistic algorithms show the tendency to interpret data points in such left over regions as outliers by assigning low memberships for these data to all clusters (close to 0) instead of further adjusting the possibly non-optimal cluster set (Höppner et al., 1999).

This described behavior is exhibited, since J_p treats each cluster independently. Every cluster contributes to some amount to the value of the objective function J_p regardless of other clusters. The resulting behavior has been regarded by stating that possibilistic clustering is a rather mode-seeking technique, aimed at finding meaningful clusters (Krishnapuram and Keller, 1996). The number c of known or desired clusters has been interpreted as an upper bound, since cluster coincidence in effect leads to a smaller number of clusters in the model (Höppner et al., 1999). For reducing the tendency of coinciding clusters and for a better coverage of the entire data space usually a probabilistic analysis is carried out before (exploiting its partitional property). The result is used for the prototype initialization of the first run of the possibilistic algorithm as well as for getting the initial guesses of the η_i (and c). After the first possibilistic analysis has been carried out, the values of the η_i are re-estimated once more using the first possibilistic fuzzy partition. The improved estimates are used for running the possibilistic algorithm a second time yielding the final cluster solution (Höppner et al., 1999).

Cluster repulsion: Dealing with the characteristics of the possibilistic clustering techniques as above is a quite good measure. However, there are theoretical results, which put forth other developments. We discovered that the objective function J_p is, in general, truly minimized only if all cluster centers are identical (Timm et al., 2004). The possibilistic objective function can be decomposed into c independent terms, one for each cluster. This is the amount by which each cluster contributes to the value of J_p. If there is a single optimal point for a cluster center (as it will usually be the case, since multiple optimal points would require a high symmetry in the data), all cluster centers moved to that point results in the lowest value of J_p for a given data set. Consequently, other results than all cluster centers being identical are achieved only because the algorithm gets stuck in a local minimum of the objective function. In the example of the possibilistic FCM model in Fig. 4 the cluster on the lower left in the figure has been found, because it is well separated and thus forms a local minimum of the objective function. This, of course, is not a desirable situation. Good solutions w.r.t the minimization of J_p unexpectedly do not correspond to what we regard as a good solution of the clustering problem. Hence the possibilistic algorithms can be improved by modifying the objective function in such a way that the problematic property examined above is removed. In Timm et al. (2004), this has been tried with an additional term for J_p that implements cluster repulsion forces. The strength of the repulsion of clusters can be controlled with a parameter that is balancing the two objectives in clustering: homogeneity within the clusters vs. heterogeneity between the clusters. The results show that the proposed modifications lead to better detection of the shape of very close or overlapping clusters. Such closely located point accumulations have been problematic, since possibilistic clusters "wander" in the direction where the most of the data can be found in their η_i environment, which easily leads to cluster coincidence. Nevertheless, the modified possibilistic techniques should also be initialized with the corresponding probabilistic algorithms as described in the last paragraph. It is a good measure for improving the chances that all data clouds will be regarded in the resulting possibilistic model leaving no present cluster structure unclassified. Recent developments that try to alleviate the problematic properties of the possibilistic clustering algorithms propose to use a combination of both fuzzy and possibilistic memberships (see Section 5.4).

Recognition of positions and shapes: The possibilistic models do not only carry problematic properties. Memberships that depend only on the distance to a cluster while being totally independent from other clusters lead to prototypes that better reflect human intuition. Calculated based on weights that reflect typicality, the centers of possibilistic clusters as well as their shape and size better fit the data clouds compared to their probabilistic relatives. Latter ones are known to be not able to recognize cluster shapes as perfectly as their possibilistic counterparts. This is due to the following reasons: if clusters are located very close or are even overlapping, then they are separated well because sharing membership is disadvantageous (see upper right in Fig. 3). Higher memberships to data points will be assigned in directions pointing away from the overlap. Thus, the centers are repelling each other. Cluster shapes are likely to be slightly distorted compared to human intuition. Noise and outliers are another reason for little prototype distortions. They have weight in probabilistic partitions and therefore attract clusters which can result in small prototype deformations and less intuitive centers. Possibilistic techniques are less sensitive to outliers and noise. Low memberships will be assigned due to greater distance. Due to this property and the more intuitive determination of positions and shapes, possibilistic techniques are attractive tools in image processing applications. However, especially the probabilistic approaches profit from the widely appreciated noise clustering techniques (see Section 5.2). By modifying the objective function a virtual noise cluster "seizes" large parts of the data weight of noise points and outliers. This leads to better detection of actual cluster structure in probabilistic models.

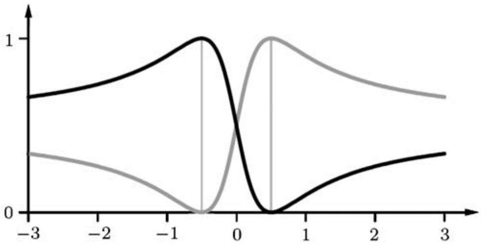

Fig. 11. The membership functions obtained by probabilistic AO for two clusters at −0.5 and 0.5.

4.3. Alternating cluster estimation

ACE techniques adopt the AO algorithm of objective function-based methods but use heuristic equations to build partitions and estimate cluster parameters. Since the cluster generation rules are chosen heuristically, this approach can be useful when cluster models become too complex to minimize them analytically or the objective function lacks differentiability (Höppner et al., 1999). The general architecture of the AO algorithm (Eqs. (7), (8)) is generalized, but the purpose of minimizing objective functions with j_U and j_C is abandoned. Thus, the classification task is directly described by the update equations as chosen by the data analysts.

The higher flexibility of choosing among different update equations is of particular interest when clustering is applied for the construction of fuzzy rule-based systems. In such applications the fuzzy sets carry semantic meaning, e.g., they are assigned linguistic labels like "low", "approximately zero", or "high". Consequently, the fuzzy sets, in fuzzy controllers for instance, are required to be convex, or even monotonous (Zadeh, 1965). Furthermore, they have to have limited support, i.e., membership degrees different from zero are allowed only within a small interval of their universe. ACE provides the flexibility to define fuzzy clustering algorithms that produce clusters Γ_i whose corresponding fuzzy sets μ_{Γ_i} fulfil these requirements. The clusters and membership degrees $\mu_{\Gamma_i}\left(\vec{x}_j\right)=u_{ij}$ obtained with the objective function-based clustering techniques contrarily do not carry the desired properties. The u_{ij} obtained by AO as in the previous section can be interpreted as discrete samples of continuous membership functions $\mu_i : \mathbb{R}^p \rightarrow [0,1]$ for each cluster. The actual shape that is taken on by these membership functions results from the respective update equations for the membership degrees. For the probabilistic fuzzy AO algorithms the continuous membership function follows from Eq. (9), with d_{ij} being the Euclidian distance $\|\cdot\|$:

$$\mu_i(\vec{x})=\frac{\|\vec{x}-\vec{c}_i\|^{-2/(m-1)}}{\sum_{t=1}^{c}\|\vec{x}-\vec{c}_t\|^{-2/(m-1)}}. \tag{16}$$

Fig. 11 shows the membership functions that would result from the probabilistic FCM algorithm for two clusters. Obviously, the membership functions μ_i are not convex ($i=\{1,2\}$). The membership for data points at first decreases the closer they are located the other cluster center, but beyond the other center membership to the first cluster increases again due to normalization constraint. Possibilistic membership functions, that result from a continuous extension according to Eq. (11) are convex, but they are not restricted to local environments around their centers (i.e., the memberships will never reach zero for larger distances). Thus, if fuzzy sets with limited support as in fuzzy controllers are desired, possibilistic memberships functions are inadequate as well. The transformation of the membership functions of the objective function-based techniques into the desired forms for the particular application is possible, but often leads to approximation errors and less accurate models.

Therefore, ACE allows to choose other membership functions aside from those that stem from an objective function-based AO scheme. Desired membership function properties can easily be incorporated in ACE. The user can chose from parameterized Gaussian, trapezoidal, Cauchy, and triangular functions (Höppner et al., 1999). We present the triangular shaped fuzzy set exemplarily in Fig. 12, since it has all desired properties considered above:

$$\mu_i(\vec{x})=\begin{cases}1-\left(\dfrac{\|\vec{x}-\vec{c}_i\|}{r_i}\right)^{\alpha} & \text{if } \|\vec{x}-\vec{c}_i\|\leqslant r_i,\\ 0 & \text{otherwise,}\end{cases} \tag{17}$$

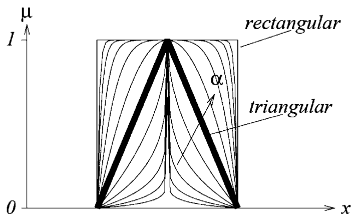

Fig. 12. The parameterized triangular fuzzy set.

where r_i are the radii of the clusters, $\alpha \in \mathbb{R}_{>0}$. In an ACE algorithm using hyper-cone shaped clusters ($\alpha = 1$) the memberships of data to fixed clusters are estimated using the above equation, such that $u_{ij} = \mu_i(\vec{x}_i)$.

Deviating from AO of objective functions the user can also choose between alternative update equations for the cluster prototypes. In ACE, a large variety of parameterized equations stemming from defuzzification methods are offered for the re-estimation of cluster centers for fixed memberships. The reference to defuzzification techniques arises, since a "crisp" center is computed from fuzzily weighted data points. Also higher order prototypes like lines, line segments, and elliptotypes have been proposed for the ACE scheme (Höppner et al., 1999). In the simplest case, however, when clusters are represented by their centers only, new centers vectors could be calculated as the weighted mean of data points assigned to them (like in the FCM; see Eq. (13)).

After the user has chosen the update equations for **U** and **C**, memberships and cluster parameters are alternatingly estimated (or updated, but not necessarily optimized w.r.t. some criterion function) as defined. This leads to a sequence $\{(\mathbf{U}_1, \mathbf{C}_1), (\mathbf{U}_2, \mathbf{C}_2), \ldots\}$ that is terminated after a predefined number of iterations $t_{\max}$ or when the $\mathbf{C}_t$ have stabilized. Some instances of the ACE might be sensitive to the initialization of the cluster centers. Thus determining $\mathbf{C}_0$ with some iterations of the probabilistic FCM might be recommended. Notice that all conventional objective function-based algorithms can be represented as instances of the more general ACE framework by selecting their membership functions as well as their prototype update equations. An experimental comparison between "real" ACE algorithms that do not reflect the minimization of an objective function and classical AO algorithms as presented above can be found in Höppner et al. (1999).

4.4. Fuzzy maximum likelihood estimation

FMLE is based on a mixture model for the process that generated the data. Each cluster is characterized by a p-variate probability distribution, which is described by a prior probability of the cluster and a conditional probability density function (cpdf). From mixture model assumption, Gath and Geva (1989) derived a distance measure such that the dissimilarity of a data point to a cluster is inversely proportional to the probability that a datum was generated by that cluster. The idea behind the definition of FMLE algorithm is twofold. Firstly, the intuitively defined distance measure is used in the update equation for the probabilistic membership degrees (Eq. (9)). Secondly, the update equations for the prototype parameters are fuzzifications of maximum likelihood estimators for the prior probabilities and the parameters of the cpdfs. For a comparison with the EM algorithm (Dempster et al., 1977) the two defining parts of the FMLE are developed next. After the description of similarities and differences, details of an experimental comparison are given.

In a mixture model it is assumed that a given data set **X** has been drawn from a population of c clusters. The data generation process may then be imagined as follows: in the first step a cluster i, $i \in \{1, \ldots, c\}$, is chosen for an example, indicating the cpdf to be used, and then the example is sampled from this cpdf. Consequently, the probability of occurrence of a data point $\vec{x}$ can be computed as

$$p_{\vec{X}}(\vec{x}; \mathbf{C}) = \sum_{i=1}^{c} p_C(i; C_i)\, p_{\vec{X}|C}(\vec{x}|i; C_i),$$

where C is a random variable describing the cluster i chosen in the first step, $\vec{X}$ is a random vector describing the attribute values of the data point. The set $\mathbf{C} = \{C_1, \ldots, C_c\}$ encompasses all model parameters with each C_i, $i = 1, \ldots, c$, containing the parameters for one cluster (that is, its prior probability and the parameters of the cpdf) (Everitt and Hand, 1981).

Furthermore, it may be assumed that the joint cpdf of the numeric attributes is a multivariate normal distribution, i.e.,

$$p_{\vec{X}|C}\left(\vec{x}|i; C_i\right) = N\left(\vec{x}; \vec{c}_i, \mathbf{\Sigma}_i\right) = \frac{1}{\sqrt{(2\pi)^p \left|\mathbf{\Sigma}_i\right|}} \exp\left(-\frac{1}{2}(\vec{x}-\vec{\mu})^\top \mathbf{\Sigma}_i^{-1}\left(\vec{x}-\vec{\mu}\right)\right),$$

where $\vec{c}_i$ is the mean vector and $\mathbf{\Sigma}_i$ the covariance matrix of the normal distribution, $i = 1, \ldots, c$. Consequently, the parameters C_i of the ith cluster are $C_i = \{\theta_i, \vec{c}_i, \mathbf{\Sigma}_i\}$, where θ_i is the prior probability of the ith cluster.

Assuming that the examples in a data set are independent and are drawn from the same distribution (i.e., that the distributions of their underlying random vectors $\vec{X}_j$ are identical), the probability of occurrence of the data set $\mathbf{X}$ is

$$P(\mathbf{X}; \mathbf{C}) = \prod_{j=1}^{n} \sum_{i=1}^{c} \theta_i N\left(\vec{x}_j; \vec{c}_i, \mathbf{\Sigma}_i\right).$$

Note that the value of the random variable C_j, which indicates the cluster for each example case $\vec{x}_j$ is unknown. However, given the data point, the posterior probability that a data point has been sampled from the cpdf of the ith cluster can be computed using Bayes' rule:

$$p_{C|\vec{X}}\left(i|\vec{x}; \mathbf{C}\right) = \frac{p_C\left(i; \theta_i\right) p_{\vec{X}|C}\left(\vec{x}|i; \theta_i\right)}{p_{\vec{X}}\left(\vec{x}; \mathbf{C}\right)} = \frac{p_C\left(i; \theta_i\right) p_{\vec{X}|C}\left(\vec{x}|i; \theta_i\right)}{\sum_{t=1}^{c} p_C\left(t; \theta_t\right) p_{\vec{X}|C}\left(\vec{x}|t; \theta_t\right)}. \tag{18}$$

This posterior probability may be used to complete the data set w.r.t. the cluster, namely by splitting each example $\vec{x}_j$ into c examples, one for each cluster, which are weighted with the posterior probability $p_{C|\vec{X}_j}\left(i|\vec{x}_j; \mathbf{C}\right)$. This idea is used in the well-known EM algorithm (Dempster et al., 1977).

The mixture model provides the means to define the similarity measure, which constitutes the first part of the idea behind the FMLE algorithm. Gath and Geva (1989) define the distance d_{ij} between the datum $\vec{x}_j$ and cluster i as the reciprocal of the probability that the datum $\vec{x}_j$ occurred *and* that it was generated by the component distribution underlying the cluster i. Then a high-probability results in a small distance value, whereas a low probability that the datum was created by the distribution of cluster i indicates a large distance. Constructed in this intuitive way the distance measure is the reciprocal of the numerator in Eq. (18) of the posterior probabilities. We obtain

$$\begin{aligned} d_{ij} &= \frac{1}{p_{C_j}\left(i; C_i\right) p_{\vec{X}_j|C_j}\left(\vec{x}_j|i; C_i\right)} \\ &= \frac{\sqrt{(2\pi)^p \left|\mathbf{\Sigma}_i\right|}}{\theta_i \exp\left(-\frac{1}{2}\left(\vec{x}_j - \vec{c}_i\right)^\top \mathbf{\Sigma}_i^{-1}\left(\vec{x}_j - \vec{c}_i\right)\right)}. \end{aligned}$$

This definition has another interesting property: inserted into the update equation for the membership degrees (9) it yields membership degrees that are equal to the posterior probabilities of the data points *provided* the fuzzifier $m = 2$ (see Eq. (18)). Only for this value of the weighting exponent the partial assignments u_{ij} of the data points to the clusters are their posterior probabilities (Bezdek et al., 1999a).

In the FMLE the equations for the re-estimation of the prototype parameters are similar to the maximum likelihood estimators for the prior probabilities and the parameters of the cpdfs. During the re-estimation each generated instance has a certain and fixed case weight in each cluster. Deviating from the EM algorithm, the cases are not weighted with their posterior probability $p_{C|\vec{X}_j}\left(i|\vec{x}_j; \mathbf{C}\right)$. Referring to the equations of the GK algorithm for detecting hyper-ellipsoidal clusters with a fuzzy covariance matrix (see Eq. (15)), Gath and Geva suggested to choose the weights u_{ij}^m for arriving at analogous equations. Thus, the update equations of the FMLE are the maximum likelihood estimators in which the u_{ij}^m have been substituted for the posterior probabilities. They look as follows:

$$\theta_i = \frac{1}{n} \sum_{j=1}^{n} u_{ij}^m, \tag{19}$$

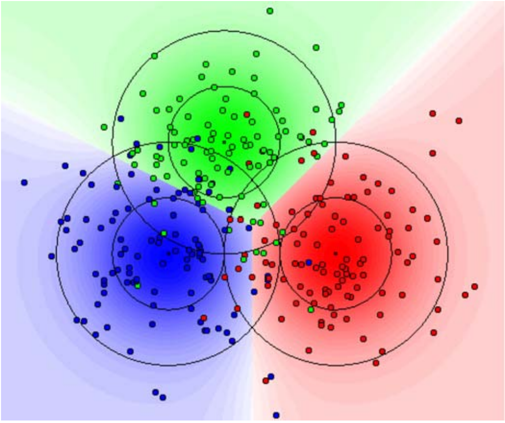

Fig. 13. The generating model.

$$\vec{c}_i = \frac{\sum_{j=1}^{n} u_{ij}^m \vec{x}_j}{\sum_{j=1}^{n} u_{ij}^m}, \tag{20}$$

$$\boldsymbol{\Sigma}_i = \frac{\sum_{j=1}^{n} u_{ij}^m \left(\vec{x}_j - \vec{c}_i\right)\left(\vec{x}_j - \vec{c}_i\right)^\top}{\sum_{j=1}^{n} u_{ij}^m}. \tag{21}$$

Discussion: FMLE is very similar to the well-known EM algorithm applied to mixture decomposition under the assumption of a hidden variable indicating the class membership (Dempster et al., 1977). The relatedness of the methods is intimated by the almost identical derivation of the estimators. The differences, however, are made explicit in the different choice of the instance weights in both algorithms.

The EM algorithm maximizes the probability of the data set to occur by first calculating the posterior probabilities that a data point was created by the cpdfs of the clusters. This is done for fixed model parameters and yields the partial weights with which a data point belongs to the clusters (see Eq. (18)). Then these weighted assignments to clusters are used as instance weights in the estimators in order to optimize the likelihood. Thus, in the EM algorithm the partial assignments of a datum to the clusters are identical to the instance weight that is used when estimating the parameters.

In FMLE, on the other hand, the partial assignments, namely the membership degrees u_{ij}, are different from the instance weights that are used when re-estimating the parameters, which are u_{ij}^m. Even in the case $m = 2$, where the calculated membership degrees u_{ij} correspond to the posterior probabilities (due to the special definition of the distance measure that is used in this approach), the case weights $u_{ij}^m \leqslant u_{ij}$.

There is no choice of m such that the FMLE algorithm becomes identical to the EM algorithm, because m also appears in formula (9) for computing the membership degrees, which rules out the choice $m = 1$. It is the coupling of the exponents m and $2/(m-1)$ that distinguishes FMLE from EM.

In experiments EM proved to be the much more stable algorithm (Döring et al., 2004). This is exemplarily illustrated using a simple data set of three closely located clusters generated as shown in Figs. 13 and 14. FMLE tends to reduce the prior probability of one or more clusters to (almost) zero—as can already be guessed from the fact that in Fig. 16 the blue (left) cluster is driven far to left, thus covering fewer data points than the other clusters. Clusters may be driven far away from the major point clouds, thus covering fewer data points than the other clusters. Sometimes this becomes extreme especially if FMLE is not initialized with the result of the FCM algorithm, while EM can be run directly without problems (initialized with Latin hyper-cube sampling, see Fig. 15). FMLE becomes more stable, but only if the attributes provide clear information about the cluster structure. Since is less stable and more sensitive to the initialization of the cluster prototypes, FMLE seems to be inferior to the classical EM algorithm. We conjecture that the described behavior of the FMLE is due to the deviating weighting of the data points which has been inferred just by analogy (Fig. 16).

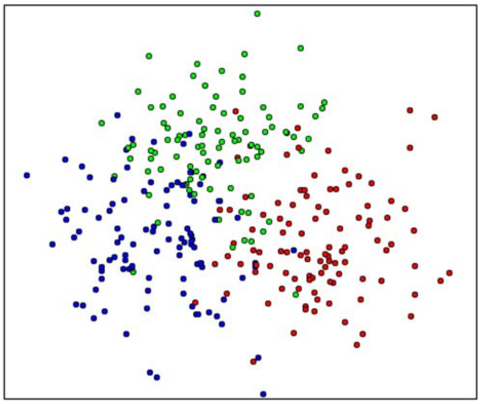

Fig. 14. The data points with classes.

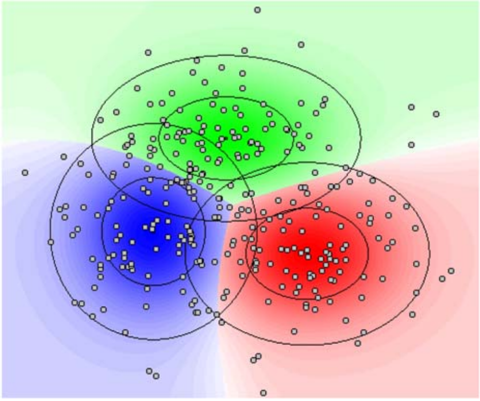

Fig. 15. Result of EM.

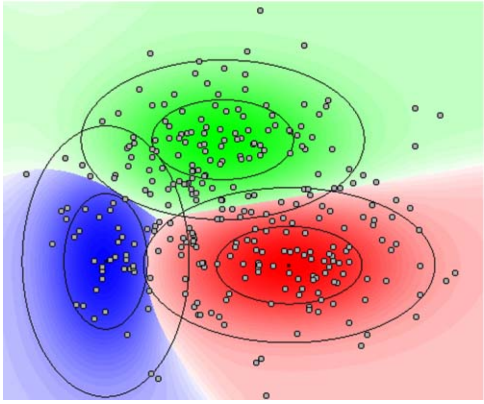

Fig. 16. Result of FMLE.

5. Related issues and current research

Fuzzy clustering is an unsupervised learning strategy in order to group data. It is very useful for constructing fuzzy if–then rules from data and it is also applied in image processing. In this introductory article we focused on three different branches of methods: objective function-based approaches, ACE, and the FMLE scheme. We tried to give an all-embracing view of the field by pointing out the major directions. Some related issues and (selected) current developments in the field shall be mentioned to conclude this article.

5.1. *Clustering of non-vectorial data*

All of the described methods in this article presuppose that the objects to cluster have a feature vector representation. In some applications such a feature vector representation is not given, but a dissimilarity relation between the objects can be defined. There exists a large variety of fuzzy clustering techniques for such settings (Bezdek et al., 1999b). These relational clustering algorithms can be divided into hierarchical and partitional approaches. The latter ones are mostly based upon the ideas that have been presented here, i.e., they are objective function-driven. Also, the ACE scheme has been adopted for grouping relational data (Runkler and Bezdek , 1998). Relational alternating cluster estimation (RACE) has been applied successfully in the field of Web mining (Runkler and Bezdek, 2003). Aside from relational data, fuzzy clustering methods have also been proposed for analysis of data that is fuzzy itself, see e.g., D'Urso and Giordani (2006). The fuzzy approach to clustering is a suitable measure to handle vague or imprecise data as observed in some applications. Fuzzy methods for clustering time-varying data can be found in Coppi and D'Urso (2003, 2006).

5.2. *Handling noise and outliers*

Noise and outliers are a common problem that has to be dealt within clustering. However, the described methods are not equally troubled by these phenomena. Possibilistic clustering algorithms based on AO can handle noise points and outliers quite naturally by assigning low data point weights to them. Thus, detection of the true clusters is not disturbed. The same holds for ACE algorithms with appropriately chosen membership functions. Outliers are located at a high distance from the major part of the data. They cannot affect the detection of clusters as long as the chosen membership function assigns low weight or even no membership to far away points. Latter is the case for fuzzy sets with limited support. Then outliers have small or even no influence on the estimated cluster centers. Contrary to the behavior of the possibilistic and selected ACE approaches, aberrant data points strongly affect probabilistic fuzzy clustering methods. Aside from the highly sensitive least-square approach to noise, the normalization of the membership degrees (Eq. (4)) in combination with the calculation of the inverse squared distances in Eq. (9) leads to high memberships for data points further away from the bulk of the data (Döring et al., 2005). The latter tendency is already visible in the two cluster example (see Fig. 11), where almost equal memberships to both clusters would be assigned for larger distances from the two clusters (in left and right direction). Since strongly deviating points is given considerable membership, their weight influences prototype calculations. This results in cluster shape deformations and slightly shifted cluster centers (see Section 4.2).

The approaches we consider here consequently aim at handling noisy data in (probabilistic) fuzzy clustering models. Their goal is to define robust variants of fuzzy clustering algorithms, i.e., algorithms whose results do not depend on the presence or absence of noisy data points or outliers in the data set. Two approaches are mentioned here: the first one is based on the introduction of a specific cluster, the so-called noise cluster, to represent noisy data points. The second method is based on the use of robust estimators.

Noise clustering (*NC*): The NC algorithm has been initially proposed by Davé (1991) and was later extended (Davé and Sen, 1997, 1998). It consists in adding, beside the c clusters to be found in a data set, the so-called noise cluster: the latter aims at grouping points that are badly represented by normal clusters, such as noisy data points or outliers. It is not explicitly associated to a prototype, but directly to the distance between an implicit prototype and the data points: the center of the noise cluster is considered to be at a constant distance, δ, of all data points. This means that all points have a priori the same "probability" to belong to the noise cluster. During the optimization process, this "probability" is then adapted as a function of the probability according to which points belong to normal clusters. The noise cluster is then introduced in the objective function, as any other cluster, leading to

$$J=\sum_{i=1}^{c}\sum_{j=1}^{n}u_{ij}^{m}d_{ij}^{2}+\sum_{k=1}^{n}\delta^{2}\left(1-\sum_{i=1}^{c}u_{ik}\right)^{m}. \tag{22}$$

The added term is similar to the terms in the first sum: the distance to the cluster prototype is replaced by δ and the membership degree to this cluster is defined as the complement to 1 of the sum of all membership degrees to the standard clusters. This in particular implies that outliers can have low membership degrees to the standard clusters, and high degree to the noise cluster, which makes it possible to reduce their influence on the standard cluster: as in possibilistic clustering, the noise clustering approach relaxes the normalization constraint expressed in Eq. (4) according to which membership degrees to good clusters must sum to 1.

The objective function (22) requires the setting of parameter δ. In the initial NC algorithm, it was set to

$$\delta^{2}=\lambda\frac{1}{c\cdot n}\left(\sum_{i=1}^{c}\sum_{j=1}^{n}d_{ij}^{2}\right), \tag{23}$$

i.e., its squared value is a proportion of the mean of the squared distances between points and other cluster prototypes, with λ a user-defined parameter determining the proportion: the smaller the λ, the higher the proportion of points that are considered as outliers. In the noise clustering model aberrant data points are identified as such by high membership to the noise cluster. The remaining data point weight that can be distributed to other clusters is accordingly reduced which leads to a better detection of position and shape of the good clusters.

Robust estimators: Another approach to handle noisy data sets is based on the exploitation of robust estimators: as indicated in Section 4.1.1, the FCM approach is based on a least-square objective function. It is well known that the least-square approach is highly sensitive to aberrant points, which is why objective function-based probabilistic clustering gives unsatisfactory results when applied to data sets contaminated with noise and outliers. Therefore, it has been proposed (Frigui and Krishnapuram, 1996) to introduce a robust estimator in the traditional objective function (see Eq. (6)), leading to consider

$$J=\sum_{i=1}^{c}\sum_{j=1}^{n}u_{ij}^{m}\rho_{i}\left(d_{ij}\right), \tag{24}$$

where ρ_i are robust symmetric positive definite functions having their minimum in 0. According to the robust M-estimator framework, ρ should be chosen such that $\rho(z)=\log\left(J(z)^{-1}\right)$ represents the contribution of error z to the objective function and J the distribution of these errors. Choosing $\rho(z)=z^2$ as it is usually the case is equivalent to assuming a normal distribution of the errors z and leads to constant weighting functions. That is, big errors have the same weight as small errors, and play too important a role on the correction applied to the parameters, making the objective function sensitive to outliers. Therefore, it is proposed to use other ρ, whose weighting functions tend to 0 for big values of z. Frigui and Krishnapuram (1996) design their own robust estimator to adapt to the desired behavior, defining the Robust c-prototypes (RCP) algorithm.

In the case where clusters are represented only by centers and a probabilistic partition is looked for (see constraint (4)), the update equations for the membership degrees and cluster prototypes derived from Eq. (24) then becomes (Frigui and Krishnapuram, 1996)

$$\vec{c}_i=\frac{\sum_{j=1}^{n}u_{ij}^{m}f_{ij}\vec{x}_j}{\sum_{j=1}^{n}u_{ij}^{m}f_{ij}}\quad\text{and}\quad u_{ij}=\frac{1}{\sum_{k=1}^{c}\left(\rho\left(d_{ij}^{2}\right)\big/\rho\left(d_{kj}^{2}\right)\right)^{1/(m-1)}}, \tag{25}$$

where $f_{ij}=f\left(d_{ij}\right)$ and $f=\mathrm{d}\rho(z)/\mathrm{d}z$. It is to be noted that outliers still have membership degrees $u_{ij}=1/c$ for all clusters. The difference and advantage as compared to FCM comes from their influence on the center, which is reduced through the f_{ij} coefficient (see Frigui and Krishnapuram, 1996 for the f_{ij} expression). Another robust clustering algorithm is the method proposed in Wu and Yang (2002) that considers a different modification of the objective function in order to handle noisy data sets.

5.3. Cluster validity and unknown number of clusters problem

All of the described clustering techniques require to specify the number of clusters c to search for. Usually, this number is not known in advance. Starting from initial guesses, several cluster partitions can be determined for different numbers of clusters. But then question arises, which of the partitions is better or correct. Further, different clustering algorithms can be used that assume certain shapes and sizes of the clusters. But are these assumptions satisfied for the given data set? These problems are addressed using validity measures. A large variety of validity indices have been proposed in the literature (Bezdek et al., 1999a; Höppner et al., 1999). *Global* validity functions evaluate complete cluster partitions and help to determine the right number of clusters. Then, clustering is carried out with different values for c. Comparing the values of the validity function for the yielded partitions the most suitable number of clusters can be chosen. Another strategy is to start with an upper bound for c and to use *local* validity functions that assess the goodness of individual clusters. The clusters are compared to each other such that similar clusters can be joined to one new cluster, whereas bad clusters can be eliminated. Then cluster analysis is carried out again with a reduced number of clusters. The procedure is repeated until no clusters have to be merged or removed. For a detailed discussion of validity functions and the specific properties of clustering results they are tailored for, the reader may be referred to Bezdek et al. (1999a) and Höppner et al. (1999).

5.4. Some current research issues

Shape and size regularization: More sophisticated fuzzy clustering algorithms, like the GK algorithm, the FMLE algorithm, but also the EM algorithm are capable of inducing clusters of ellipsoidal shape and different sizes. However, the additional cluster parameters can reduce the robustness of the algorithms, even rendering their application problematic sometimes. Thus, the more complex algorithms were usually initialized with simpler models. Lately, methods have been suggested that introduce shape and size constraints to handle the higher degrees of freedom in this algorithms effectively. Experiments show that these regularization methods improve the robustness of the more sophisticated fuzzy clustering algorithms, which without them suffer from instabilities even on fairly simple data sets. Regularized and constrained clustering is so robust that it can even be used without an initialization by the FCM algorithm. With a time-dependent shape regularization parameter one may even obtain a soft transition from the FCM algorithm (spherical clusters) to the GK algorithm (general ellipsoidal clusters) (Borgelt and Kruse, 2005).

Understanding the principles behind the fuzzifier: The fuzzifier m is a parameter that features all fuzzy clustering methods. This fuzzifier controls how much clusters may overlap. However, there other functions of the membership degrees in objective functions than u_{ij}^m that result in fuzzy data partitions. Recent investigations provide a more general framework for implementing fuzzifiers. It helps to overcome some negative effects and problems due to clusters with varying data density, noisy data, and large data sets with a higher number of clusters (Klawonn and Höppner, 2003a, b).

Overcome the problems of the possibilistic c-means: While possibilistic degrees make it possible to reduce the influence of outliers, the (probabilistic) fuzzy memberships ensure the necessary assignment of all data points in a data set, leaving no data unclassified. Since a good clustering result requires both the partitioning property of (probabilistic) FCM and the mode-seeking robust property of the possibilistic c-means, the use of both possibilistic degrees as well as fuzzy memberships in a clustering algorithm has been proposed in Pal et al. (1997, 2004). In the more recent work (Pal et al., 2004), clustering with both typicality assignments (here: t_{ij}) and normalized degrees $\left(u_{ij}\right)$ is performed through the optimization of the following objective function:

$$J = \sum_{i=1}^{c}\sum_{j=1}^{n}\left(au_{ij}^{m} + bt_{ij}^{\eta}\right)d_{ij}^{2} + \sum_{i=1}^{c}\eta_i\sum_{j=1}^{n}\left(1 - t_{ij}\right)^{\eta}, \tag{26}$$

while the known constraints apply to both respective types of the gradual assignments. a and b are user-defined parameters that rule the importance the two terms must play. In the case where the Euclidean distance is used, the update equations are then

$$u_{ij} = \frac{1}{\sum_{k=1}^{c}\left(d_{ij}^2/d_{kj}^2\right)^{1/(m-1)}}, \quad t_{ij} = \frac{1}{1+\left(\left(b/\eta_j\right)\, d_{ij}^2\right)^{1/(\eta-1)}}, \quad \vec{c}_i = \frac{\sum_{j=1}^{n}\left(a u_{ij}^{m} + b t_{ij}^{\eta}\right)\vec{x}_j}{\sum_{j=1}^{n}\left(a u_{ij}^{m} + b t_{ij}^{\eta}\right)}.$$

Thus, u_{ij} are similar to the membership degrees of FCM (see Eq. (9)), and t_{ij} to the possibilistic coefficients of PCM when replacing η_i with η_i/b (see Eq. (11)). Cluster centers then depend on both coefficients, parameters a, b, m, and η rule their relative influence. This shows that if b is higher than a the centers will be more influenced by the possibilistic coefficients than the membership degrees. Thus, to reduce the influence of outliers, a bigger value for b than a should be used. Still, it is to be noticed that four parameters are to be defined by the user, and that their influence is correlated, making it somewhat difficult to determine their optimal value. Furthermore, the problem of this method is that due to their interaction the interpretation of the obtained coefficients, in particular, of the t_{ij} deviates from purely possibilistic c-means models.

References

Babu, G.P., Murty, M.N., 1994. Clustering with evolutionary strategies. Pattern Recognition 27, 321–329.

Barni, M., Cappellini, V., Mecocci, A., 1996. Comments on a possibilistic approach to clustering. IEEE Trans. Fuzzy Systems 4, 393–396.

Bezdek, J.C., 1973. Fuzzy mathematics in pattern classification. Ph.D. Thesis, Applied Mathematics Center, Cornell University, Ithaca.

Bezdek, J.C., 1981. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York.

Bezdek, J.C., Keller, J., Krishnapuram, R., Pal, N.R., 1999a. Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. Kluwer, Boston, London.

Bezdek, J.C., Keller, J., Krishnapuram, R., Pal, N.R., 1999b. Cluster analysis for relational data. In: Fuzzy Models and Algorithms for Pattern Recognition and Image Processing.Kluwer, Boston, London, pp. 137–182 (Chapter 3).

Blake, C.L., Merz, C.J., 1998. UCI repository of machine learning databases. URL ⟨http://www.ics.uci.edu/∼mlearn/MLRepository.html⟩.

Bock, H.H., 1974. Automatische Klassifikation. Vadenhoeck & Ruprecht, Göttingen, Zürich.

Borgelt, C., Kruse, R., 2005. Fuzzy and probabilistic clustering with shape and size constraints. In: Proceedings of the 11th International Fuzzy Systems Association World Congress, IFSA'05, Beijing, China, pp. 945–950.

Coppi, R., D'Urso, P., 2003. Three-way fuzzy clustering models for lr fuzzy time trajectories. Comput. Statist. Data Anal. 43 (2), 149–177.

Coppi, R., D'Urso, P., 2006. Fuzzy unsupervised classification of multivariate time trajectories with the Shannon entropy regularization. Comput. Statist. Data Anal. 50 (6), 1452–1477.

Davé, R., 1991. Characterization and detection of noise in clustering. Pattern Recognition Lett. 12, 657–664.

Davé, R.N., Krishnapuram, R., 1997. Robust clustering methods: a unified view. IEEE Trans. Fuzzy Systems 5, 270–293.

Davé, R., Sen, S., 1997. On generalising the noise clustering algorithms. In: Proceedings of the Seventh IFSA World Congress, IFSA'97. pp. 205–210.

Davé, R., Sen, S., 1998. Generalized noise clustering as a robust fuzzy c-m-estimators model. In: Proceedings of the 17th Annual Conference of the North American Fuzzy Information Processing Society: NAFIPS'98. pp. 256–260.

Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood estimation from incomplete data via the EM algorithm (with discussion). J. Roy. Statist. Soc. Ser. B 39, 1–38.

Döring, C., Borgelt, C., Kruse, R., 2004. Fuzzy clustering of quantitative and qualitative data. In: Dick, S., Kurgan, L., Musilek, P., Pedrycz, W., Reformat M. (Eds.), Proceedings of the 2004 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS), IEEE, Banff, Alberta, Canada, pp. 84–89.

Döring, C., Borgelt, C., Kruse, R., 2005. Effects of irrelevant attributes in fuzzy clustering. In: Proceedings of the 14th IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2005, Reno, Nevada, USA.IEEE Press, Piscataway, NJ, USA, pp. 862–866.

Dubois, D., Prade, H., 1988. Possibility Theory. Plenum Press, New York, NY, USA.

Duda, R., Hart, P., 1973. Pattern Classification and Scene Analysis. Wiley, New York, USA.

Dunn, J.C., 1974. A fuzzy relative of the isodata process and its use in detecting compact, well separated clusters. J. Cybernet. 3, 95–104.

D'Urso, P., Giordani, P., 2006. A weighted fuzzy c-means clustering model for fuzzy data. Comput. Statist. Data Anal. 50 (6), 1496–1523.

Everitt, B.S., Hand, D.J., 1981. Finite Mixture Distributions. Chapman & Hall, London.

Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems. Ann. Eugenics 7 (2), 179–188.

Frigui, H., Krishnapuram, R., 1996. A robust algorithm for automatic extraction of an unknown number of clusters from noisy data. Pattern Recognition Lett. 17, 1223–1232.

Gath, I., Geva, A.B., 1989. Unsupervised optimal fuzzy clustering. IEEE Trans. Pattern Anal. Mach. Intelligence 11, 773–781.

Gustafson, E.E., Kessel, W.C., 1979. Fuzzy clustering with a fuzzy covariance matrix. In: Proceedings of the IEEE Conference on Decision and Control, San Diego, CA.IEEE Press, Piscataway, NJ, pp. 761–766.

Hathaway, R.J., Bezdek, J.C., 1995. Optimization of clustering criteria by reformulation. IEEE Trans. Fuzzy Systems 3, 241–245.

Höppner, F., Klawonn, F., Kruse, R., Runkler, T., 1999. Fuzzy Cluster Analysis. Wiley, Chichester, UK.

Klawonn, F., Höppner, F., 2003a. An alternative approach to the fuzzifier in fuzzy clustering to obtain better clustering results. In: Proceedings of the Third Conference for Fuzzy Logic and Technology (Eusflat), Zittau/Goerlitz. pp. 730–734.

Klawonn, F., Höppner, F., 2003b. What is fuzzy about fuzzy clustering? Understanding and improving the concept of the fuzzifier. In: Advances in Intelligent Data Analysis V, Lecture Notes in Computer Science, vol. 2810. Springer GmbH, Berlin, pp. 254–264, 3-540-40383-3.

Klawonn, F., Keller, A., 1998. Fuzzy clustering with evolutionary algorithms. Internat. J. Intelligent Systems 13, 975–991.

Krishnapuram, R., Keller, J., 1993. A possibilistic approach to clustering. IEEE Trans. Fuzzy Systems 1, 98–110.

Krishnapuram, R., Keller, J., 1996. The possibilistic c-means algorithm: insights and recommendations. IEEE Trans. Fuzzy Systems 4, 385–393.

McKay, M.D., Beckman, R.J., Conover, W.J., 1979. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 21 (2), 239–245.

Pal, N., Pal, K., Bezdek, J., 1997. A mixed c-means clustering model. In: Proceedings of the FUZZ-IEEE'97. pp. 11–21.

Pal, N., Pal, K., Keller, J., Bezdek, J., 2004. A new hybrid c-means clustering model. In: Proceedings of the FUZZ-IEEE'04. pp. 179–184.

Runkler, T.A., Bezdek, J.C., 1998. Race: relational alternating cluster estimation and the wedding table problem. In: Brauer, W. (Ed.), Fuzzy-Neuro-Systems '98, München, Proceedings in Artificial Intelligence, vol. 7, pp. 330–337

Runkler, T.A., Bezdek, J.C., 2003. Web mining with relational clustering. Internat. J. Approx. Reasoning 32 (2–3), 217–236.

Ruspini, E.H., 1969. A new approach to clustering. Inform. Control 15 (1), 22–32.

Timm, H., Borgelt, C., Döring, C., Kruse, R., 2004. An extension to possibilistic fuzzy cluster analysis. Fuzzy Sets and Systems 147, 3–16.

Wu, K., Yang, M., 2002. Alternating c-means clustering algorithms. Pattern Recognition 35, 2267–2278.

Zadeh, L.A., 1965. Fuzzy sets. Inform. Control 8, 338–353.