1

Fundamentals of Fuzzy Clustering

Rudolf Kruse, Christian Döring, and Marie-Jeanne Lesot

Department of Knowledge Processing and Language Engineering,
University of Magdeburg, Germany

1.1 INTRODUCTION

Clustering is an unsupervised learning task that aims at decomposing a given set of *objects* into subgroups or *clusters* based on similarity. The goal is to divide the data-set in such a way that objects (or example cases) belonging to the same cluster are as similar as possible, whereas objects belonging to different clusters are as dissimilar as possible. 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 a set of unordered objects. In this 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 arranging similar objects into clusters one tries to reconstruct the unknown structure in the hope that every cluster found represents an actual type or category of objects. Clustering methods can also be used for data reduction purposes. 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 example cases, which is equivalent to measuring their similarity. As a result one yields a partition of the data-set into clusters regarding the chosen dissimilarity relation.

All clustering methods that we consider in this chapter are partitioning algorithms. Given a positive integer c, they aim at finding the best partition of the data into c groups based on the given dissimilarity measure and they regard the space of possible partitions into c subsets only. Therein partitioning clustering methods are different from hierarchical techniques. The latter organize data in a nested sequence of groups, which can be visualized in the form of a dendrogram or tree. Based on a dendrogram one can decide on the number of clusters at which the data are best represented for a given purpose. Usually the number of (true) clusters in the given data is unknown in advance. However, using the partitioning methods one is usually required to specify the number of clusters c as an input parameter. Estimating the actual number of clusters is thus an important issue that we do not leave untouched in this chapter.

Advances in Fuzzy Clustering and its Applications Edited by J. Valente de Oliveira and W. Pedrycz

A common concept of all described clustering approaches is that they are prototype-based, i.e., the clusters are represented by *cluster prototypes* $C_i, i = 1, \ldots, c$. Prototypes are used to capture the structure (distribution) of the data in each cluster. With this representation of the clusters we formally denote the set of prototypes $C = \{C_1, \ldots, C_c\}$. Each prototype C_i is an n-tuple of parameters that consists of a *cluster center* $\mathbf{c}_i$ (location parameter) and maybe some additional parameters about the size and the shape of the cluster. The cluster center $\mathbf{c}_i$ is an instantiation of the attributes used to describe the domain, just as the data points in the data-set to divide. The size and shape parameters of a prototype determine the extension of the cluster in different directions of the underlying domain. The prototypes are constructed by the clustering algorithms and serve as prototypical representations of the data points in each cluster.

The chapter is organized as follows: Section 1.2 introduces the basic approaches to hard, fuzzy, and possibilistic clustering. The objective function they minimize is presented as well as the minimization method, the alternating optimization (AO) scheme. The respective partition types are discussed and special emphasis is put on a thorough comparison between them. Further, an intuitive understanding of the general properties that distinguish their results is presented. Then a systematic overview of more sophisticated fuzzy clustering methods is presented. In Section 1.3, the variants that modify the used distance functions for detecting specific cluster shapes or geometrical contours are discussed. In Section 1.4 variants that modify the optimized objective functions for improving the results regarding specific requirements, e.g., dealing with noise, are reviewed. Lastly, in Section 1.5, the alternating cluster estimation framework is considered. It is a generalization of the AO scheme for cluster model optimization, which offers more modeling flexibility without deriving parameter update equations from optimization constraints. Section 1.6 concludes the chapter pointing at related issues and selected developements in the field.

1.2 BASIC CLUSTERING ALGORITHMS

In this section, we present the fuzzy C-means and possibilistic C-means, deriving them from the hard c-means clustering algorithm. The latter one is better known as k-means, but here we call it (hard) C-means to unify the notation and to emphasize that it served as a starting point for the fuzzy extensions. We further restrict ourselves to the simplest form of cluster prototypes at first. That is, each prototype only consists of the center vectors, $C_i = (\mathbf{c}_i)$, such that the data points assigned to a cluster are represented by a prototypical point in the data space. We consider as a distance measure d an inner product norm induced distance as for instance the Euclidean distance. The description of the more complex prototypes and other dissimilarity measures is postponed to Section 1.3, since they are extensions of the basic algorithms discussed here.

All algorithms described in this section are based on *objective functions* J, which are mathematical criteria that quantify the goodness of cluster models that comprise prototypes and data partition. Objective functions serve as cost functions that have to be minimized to obtain optimal cluster solutions. Thus, for each of the following cluster models the respective objective function expresses desired properties of what should be regarded as ''best'' results of the cluster algorithm. Having defined such a criterion of optimality, the clustering task can be formulated as a function optimization problem. That is, the algorithms determine the best decomposition of a data-set into a predefined number of clusters by minimizing their objective function. The steps of the algorithms follow from the optimization scheme that they apply to approach the optimum of J. Thus, in our presentation of the hard, fuzzy, and possibilistic c-means we discuss their respective objective functions first. Then we shed light on their specific minimization scheme.

The idea of defining an objective function and have its minimization drive the clustering process is quite universal. Aside from the basic algorithms many extensions and modifications have been proposed that aim at improvements of the clustering results with respect to particular problems (e.g., noise, outliers). Consequently, other objective functions have been tailored for these specific applications. We address the most important of the proposed objective function variants in Section 1.4. However, regardless of the specific objective function that an algorithm is based on, the objective function is a goodness measure.

Thus it can be used to compare several clustering models of a data-set that have been obtained by the same algorithm (holding the number of clusters, i.e., the value of c, fixed).

In their basic forms the hard, fuzzy, and possibilistic C-means algorithms look for a predefined number of c clusters in a given data-set, where each of the clusters is represented by its center vector. However, hard, fuzzy, and possibilistic C-means differ in the way they assign data to clusters, i.e., what type of data partitions they form. In classical (hard) cluster analysis each datum is assigned to exactly one cluster. Consequently, the hard C-means yield exhaustive partitions of the example set into non-empty and pairwise disjoint subsets. Such hard (crisp) assignment of data to clusters can be inadequate in the presence of data points that are almost equally distant from two or more clusters. Such special data points can represent hybrid-type or mixture objects, which are (more or less) equally similar to two or more types. A crisp partition arbitrarily forces the full assignment of such data points to one of the clusters, although they should (almost) equally belong to all of them. For this purpose the fuzzy clustering approaches presented in Sections 1.2.2 and 1.2.3 relax the requirement that data points have to be assigned to one (and only one) cluster. Data points can belong to more than one cluster and even with different degrees of membership to the different clusters. These gradual cluster assignments can reflect present cluster structure in a more natural way, especially when clusters overlap. Then the memberships of data points at the overlapping boundaries can express the ambiguity of the cluster assignment.

The shift from hard to gradual assignment of data to clusters for the purpose of more expressive data partitions founded the field of fuzzy cluster analysis. We start our presentation with the hard C-means and later on we point out the relatedness to the fuzzy approaches that is evident in many respects.

1.2.1 Hard c-means

In the classical C-means model each data point $\mathbf{x}_j$ in the given data-set $\mathrm{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, $\mathrm{X} \subseteq \mathbb{R}^p$ is assigned to exactly one cluster. Each cluster Γ_i is thus a subset of the given data-set, $\Gamma_i \subset \mathrm{X}$. The set of clusters $\Gamma = \{\Gamma_1, \ldots, \Gamma_c\}$ is required to be an exhaustive partition of the data-set X into c non-empty and pairwise disjoint subsets Γ_i, $1 < c < n$. In the C-means such a data partition is said to be optimal when the sum of the squared distances between the cluster centers and the data points assigned to them is minimal (Krishnapuram and Keller, 1996). This definition follows directly from the requirement that clusters should be as homogeneous as possible. Hence the objective function of the hard C-means can be written as follows:

$$J_h(\mathrm{X}, \mathrm{U}_h, \mathrm{C}) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij} d_{ij}^2, \tag{1.1}$$

where $\mathrm{C} = \{C_1, \ldots, C_c\}$ is the set of cluster prototypes, d_{ij} is the distance between $\mathbf{x}_j$ and cluster center $\mathbf{c}_i$, U is a $c \times n$ binary matrix called partition matrix. The individual elements

$$u_{ij} \in \{0, 1\} \tag{1.2}$$

indicate the assignment of data to clusters: $u_{ij} = 1$ if the data point $\mathbf{x}_j$ is assigned to prototype C_i, i.e., $\mathbf{x}_j \in \Gamma_i$; and $u_{ij} = 0$ otherwise. To ensure that each data point is assigned exactly to one cluster, it is required that:

$$\sum_{i=1}^{c} u_{ij} = 1, \quad \forall j \in \{1, \ldots, n\}. \tag{1.3}$$

This constraint enforces exhaustive partitions and also serves the purpose to avoid the trivial solution when minimizing J_h, which is that no data is assigned to any cluster: $u_{ij} = 0, \forall i, j$. Together with $u_{ij} \in \{0, 1\}$ it is possible that data are assigned to one or more clusters while there are some remaining clusters left empty. Since such a situation is undesirable, one usually requires that:

$$\sum_{j=1}^{n} u_{ij} > 0, \quad \forall i \in \{1, \ldots, c\}. \tag{1.4}$$

J_h depends on the two (disjoint) parameter sets, which are the cluster centers c and the assignment of data points to clusters U. The problem of finding parameters that minimize the C-means objective function is NP-hard (Drineas *et al.*, 2004). Therefore, the hard C-means clustering algorithm, also known as *ISODATA algorithm* (Ball and Hall, 1966; Krishnapuram and Keller, 1996), minimizes J_h using an alternating optimization (AO) scheme.

Generally speaking, AO can be applied when a criterion function cannot be optimized directly, or when it is impractical. The parameters to optimize are split into two (or even more) groups. Then one group of parameters (e.g., the partition matrix) is optimized holding the other group(s) (e.g., the current cluster centers) fixed (and vice versa). This iterative updating scheme is then repeated. The main advantage of this method is that in each of the steps the optimum can be computed directly. By iterating the two (or more) 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 applied objective function J. However, alternating optimization is the commonly used parameter optimization method in clustering algorithms. Thus for each of the algorithms in this chapter we present the corresponding parameter update equations of their alternating optimization scheme.

In the case of the hard C-means the iterative optimization scheme works as follows: at first initial cluster centers are chosen. This can be done randomly, i.e., by picking c random vectors that lie within the smallest (hyper-)box that encloses all data; or by initializing cluster centers with randomly chosen data points of the given data-set. Alternatively, more sophisticated initialization methods can be used as well, e.g., Latin hypercube sampling (McKay, Beckman and Conover, 1979). Then the parameters C are held fixed and cluster assignments U are determined that minimize the quantity of J_h. In this step each data point is assigned to its closest cluster center:

$$u_{ij} = \begin{cases} 1, & \text{if } i = \operatorname{argmin}_{l=1}^{c} d_{lj} \\ 0, & \text{otherwise} \end{cases}. \tag{1.5}$$

Any other assignment of a data point than to its closest cluster would not minimize J_h for fixed clusters. Then the data partition U is held fixed and new cluster centers are computed as the mean of all data vectors assigned to them, since the mean minimizes the sum of the square distances in J_h. The calculation of the mean for each cluster (for which the algorithm got its name) is stated more formally:

$$\mathbf{c}_i = \frac{\sum_{j=1}^{n} u_{ij}\mathbf{x}_j}{\sum_{j=1}^{n} u_{ij}}. \tag{1.6}$$

The two steps (1.5) and (1.6) are iterated until no change in C or U can be observed. Then the hard C-means terminates, yielding final cluster centers and data partition that are possibly locally optimal only.

Concluding the presentation of the hard C-means we want to mention its expressed tendency to become stuck in local minima, which makes it necessary to conduct several runs of the algorithm with different initializations (Duda and Hart, 1973). Then the best result out of many clusterings can be chosen based on the values of J_h.

We now turn to the fuzzy approaches, that relax the requirement $u_{ij} \in \{0, 1\}$ that is placed on the cluster assignments in classical clustering approaches. The extensions are based on the concepts of fuzzy sets such that we arrive at gradual memberships. We will discuss two major types of gradual cluster assignments and fuzzy data partitions altogether with their differentiated interpretations and standard algorithms, which are the (probabilistic) fuzzy C-means (FCM) in the next section and the possibilistic fuzzy C-means (PCM) in Section 1.2.3.

1.2.2 Fuzzy c-means

Fuzzy cluster analysis allows gradual memberships of data points to clusters measured as degrees in [0,1]. This gives the flexibility to express that data points can belong to more than one cluster. Furthermore, these membership degrees offer a much finer degree of detail of the data model. Aside from assigning a data point to clusters in 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 (Zadeh, 1965). Thus, fuzzy clustering allows fine grained solution spaces in the form of fuzzy partitions of the set of given examples $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. Whereas the clusters Γ_i of data partitions have been classical subsets so far, they are represented by the fuzzy sets μ_{Γ_i} of the data-set X in the following. Complying with fuzzy set theory, the cluster assignment u_{ij} is now the membership degree of a datum $\mathbf{x}_j$ to cluster Γ_i, such that: $u_{ij} = \mu_{\Gamma_i}(\mathbf{x}_j) \in [0, 1]$. Since memberships to clusters are fuzzy, there is not a single label that is indicating to which cluster a data point belongs. Instead, fuzzy clustering methods associate a fuzzy label vector to each data point $\mathbf{x}_j$ that states its memberships to the c clusters:

$$\mathbf{u}_j = (u_{1j}, \ldots, u_{cj})^T. \tag{1.7}$$

The $c \times n$ matrix $U = (u_{ij}) = (\mathbf{u}_1, \ldots, \mathbf{u}_n)$ is then called a fuzzy partition matrix. Based on the fuzzy set notion we are now better suited to handle ambiguity of cluster assignments when clusters are badly delineated or overlapping.

So far, the general definition of fuzzy partition matrices leaves open how assignments of data to more than one cluster should be expressed in form of membership values. Furthermore, it is still unclear what degrees of belonging to clusters are allowed, i.e., the solution space (set of allowed fuzzy partitions) for fuzzy clustering algorithms is not yet specified. 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. In our discussion we begin with the most widely used type, the probabilistic 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 approaches and, in the next section, p for the possibilistic models. The latter constitute the second type of fuzzy partitions.

Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ 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 $U_f = (u_{ij}) = (\mu_{\Gamma_i}(\mathbf{x}_j))$ a *probabilistic cluster partition* of X if

$$\sum_{j=1}^{n} u_{ij} > 0, \quad \forall i \in \{1, \ldots, c\}, \qquad \text{and} \tag{1.8}$$

$$\sum_{i=1}^{c} u_{ij} = 1, \quad \forall j \in \{1, \ldots, n\} \tag{1.9}$$

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

Constraint (1.8) guarantees that no cluster is empty. This corresponds to the requirement in classical cluster analysis that no cluster, represented as (classical) subset of X, is empty (see Equation (1.4)). Condition (1.9) 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 (see Equation (1.3)). As a consequence of both constraints no cluster can contain the full membership of all data points. Furthermore, condition (1.9) 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.

Example: Figure 1.1 shows a (probabilistic) fuzzy classification of a two-dimensional symmetric data-set with two clusters. The grey scale indicates the strength of belonging to the clusters. The darker shading in the image indicates a high degree of membership for data points close to the cluster centers, while membership decreases for data points that lie further away from the clusters. The membership values of the data points are shown in Table 1.1. They form a probabilistic cluster partition according to the definition above. The following advantages over a conventional clustering representation can be noted: points in the center of a cluster can have a degree equal to 1, while points close to boundaries can be

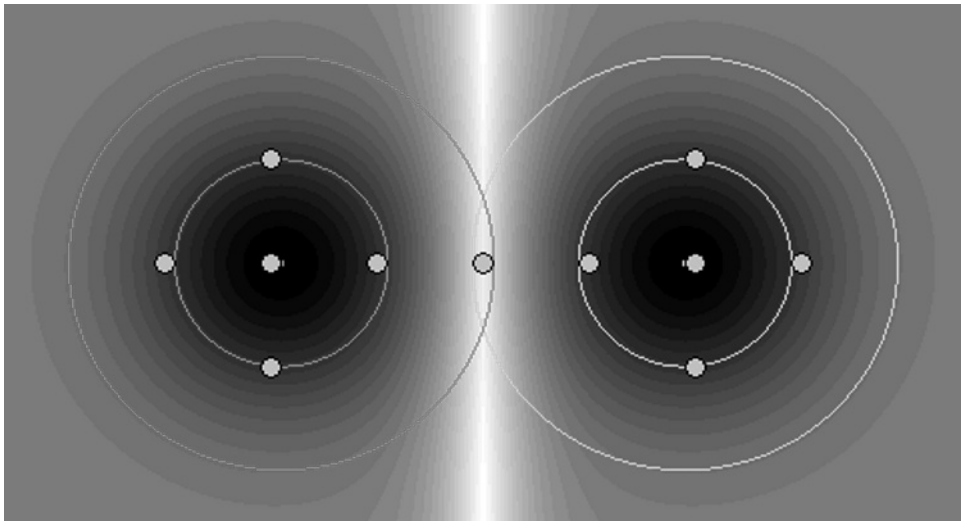

Figure 1.1 A symmetric data-set with two clusters.

identified as such, since their membership degree to the cluster they are closer to is considerably smaller than 1. Points on class boundaries may be classified as undetermined with a degree of indeterminacy proportional to their similarity to core points. The equidistant data point $\mathbf{x}_5$ in the middle of the figure would have to be arbitrarily assigned with full weight to one of the clusters if only classical ('crisp') partitions were allowed. In this fuzzy partition, however, it can be associated with the equimembership vector $(0.5, 0.5)^T$ to express the ambiguity of the assignment. Furthermore, crisp data partitions cannot express the difference between data points in the center and those that are rather at the boundary of a cluster. Both kinds of points would be fully assigned to the cluster they are most similar to. In a fuzzy cluster partition they are assigned degrees of belonging depending on their closeness to the centers.

After defining probabilistic partitions we can turn to developing an objective function for the fuzzy clustering task. Certainly, the closer a data point lies to the center of a cluster, the higher its degree of membership should be to this cluster. Following this rationale, one can say that the distances between the cluster centers and the data points (strongly) assigned to it should be minimal. Hence the problem to divide a given data-set into c clusters can (again) be stated as the task to minimize the squared distances of the data points to their cluster centers, since, of course, we want to maximize the degrees of membership. The probabilistic fuzzy objective function J_f is thus based on the least sum of squared distances just as J_h

Table 1.1 A fuzzy partition of the symmetric data-set.

j	x	y	u_{0j}	u_{1j}
0	−3	0	0.93	0.07
1	−2	0	0.99	0.01
2	−1	0	0.94	0.06
3	−2	1	0.69	0.31
4	−2	−1	0.69	0.31
5	0	0	**0.50**	**0.50**
6	1	0	0.06	0.94
7	2	0	0.01	0.99
8	3	0	0.07	0.93
9	2	1	0.31	0.69
10	2	−1	0.31	0.69

of the hard C-means. More formally, a fuzzy cluster model of a given data-set X into c clusters is defined to be optimal when it minimizes the objective function:

$$J_f(\mathrm{X}, \mathrm{U}_f, \mathrm{C}) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m d_{ij}^2, \tag{1.10}$$

under the constraints (1.8) and (1.9) that have to be satisfied for probabilistic membership degrees in U_f. The condition (1.8) avoids the trivial solution of minimization problem, i.e., $u_{ij} = 0, \forall i, j$. The normalization constraint (1.9) leads to a 'distribution' of the weight of each data point over the different clusters. Since all data points have the same fixed amount of membership to share between clusters, the normalization condition implements the known partitioning property of any probabilistic fuzzy clustering algorithm. The parameter $m, m > 1$, is called the *fuzzifier* or *weighting exponent*. The exponentiation of the memberships with m in J_f can be seen as a function g of the membership degrees, $g(u_{ij}) = u_{ij}^m$, that leads to a generalization of the well-known least squared error functional as it was applied in the hard c-means (see Equation (1.1)). The actual value of m then determines the 'fuzziness' of the classification. It has been shown for the case $m = 1$ (when J_h and J_f become identical), that cluster assignments remain hard when minimizing the target function, even though they are allowed to be fuzzy, i.e., they are not constrained in $\{0, 1\}$ (Dunn, 1974b). For achieving the desired fuzzification of the resulting probabilistic data partition the function $g(u_{ij}) = u_{ij}^2$ has been proposed first (Dunn, 1974b). The generalization for exponents $m > 1$ that lead to fuzzy memberships has been proposed in (Bezdek, 1973). With higher values for m the boundaries between clusters become softer, with lower values they get harder. Usually $m = 2$ is chosen. Aside from the standard weighting of the memberships with u_{ij}^m other functions g that can serve as fuzzifiers have been explored. Their influence on the memberships will be discussed in Section 1.4.2.

The objective function J_f is alternately optimized, i.e., first the membership degrees are optimized for fixed cluster parameters, then the cluster prototypes are optimized for fixed membership degrees:

$$\mathrm{U}_\tau = j_U(\mathrm{C}_{\tau-1}), \quad \tau > 0 \qquad \text{and} \tag{1.11}$$

$$\mathrm{C}_\tau = j_C(\mathrm{U}_\tau). \tag{1.12}$$

In each of the two steps the optimum can be computed directly using the parameter update equations j_U and j_C for the membership degrees and the cluster centers, respectively. The update formulae 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 (1.9)). The resulting equations for the two iterative steps form the fuzzy C-means algorithm.

The membership degrees have to be chosen according to the following update formula that is independent of the chosen distance measure (Bezdek, 1981; Pedrycz, 2005):

$$u_{ij} = \frac{1}{\sum_{l=1}^{c} \left(\frac{d_{ij}^2}{d_{lj}^2} \right)^{\frac{1}{m-1}}} = \frac{d_{ij}^{-\frac{2}{m-1}}}{\sum_{l=1}^{c} d_{lj}^{-\frac{2}{m-1}}}. \tag{1.13}$$

In this case there exists a cluster i with zero distance to a datum $\mathbf{x}_j$, $u_{ij} = 1$ and $u_{lj} = 0$ for all other clusters $l \neq i$. The above equation clearly shows the relative character of the probabilistic membership degree. It depends not only on the distance of the datum $\mathbf{x}_j$ 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. In the case of the basic fuzzy C-means model the cluster center vectors serve as prototypes, while an inner product norm induced metric is applied as distance measure. Consequently the derivations of J_f w.r.t. the centers yield (Bezdek, 1981):

$$\mathbf{c}_i = \frac{\sum_{j=1}^{n} u_{ij}^m \mathbf{x}_j}{\sum_{j=1}^{n} u_{ij}^m}. \tag{1.14}$$

The choice of the optimal cluster center points for fixed memberships of the data to the clusters has the form of a generalized mean value computation for which the fuzzy C-means algorithm has its name.

The general form of the AO scheme of coupled equations (1.11) and (1.12) starts with an update of the membership matrix in the first iteration of the algorithm ($\tau = 1$). The first calculation of memberships is based on an initial set of prototypes C_0. Even though the optimization of an objective function could mathematically also start with an initial but valid membership matrix (i.e., fulfilling constraints (1.8) and (1.9)), a C_0 initialization is easier and therefore common practice in all fuzzy clustering methods. Basically the fuzzy C-means can be initialized with cluster centers that have been randomly placed in the input space. The repetitive updating in the AO scheme can be stopped if the number of iterations τ exceeds some predefined number of maximal iterations τ_{max}, or when the changes in the prototypes are smaller than some termination accuracy. The (probabilistic) fuzzy C-means algorithm is known as a stable and robust classification method. Compared with the hard C-means it is quite insensitive to its initialization and it is not likely to get stuck in an undesired local minimum of its objective function in practice (Klawonn, 2006). Due to its simplicity and low computational demands, the probabilistic fuzzy C-means is a widely used initializer for other more sophisticated clustering methods. On the theoretical side it has been proven 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).

1.2.3 Possibilistic c-means

Although often desirable, the 'relative' character of the probabilistic membership degrees can be misleading (Timm, Borgett, Döring and Kruse, 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 Figure 1.2. Datum $\mathbf{x}_1$ 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 $\mathbf{x}_2$ 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 $\mathbf{x}_2$ 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, Klawonn, Kruse and Runkler 1999): 'If the datum $\mathbf{x}_i$ has to be assigned to a cluster, then with the probability u_{ij} to the cluster i'.

The normalization of memberships can further lead to undesired effects in the presence of noise and outliers. The fixed data point weight may result in high membership of these points to clusters, even though they are a large distance from the bulk of data. Their membership values consequently affect the clustering results, since data point weight attracts cluster prototypes. By dropping the normalization constraint (1.9) in the following definition one tries to achieve a more intuitive assignment of degrees of membership and to avoid undesirable normalization effects.

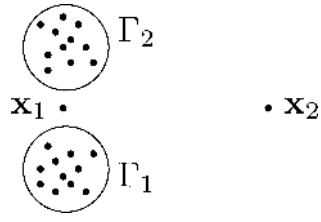

Figure 1.2 A situation in which the probabilistic assignment of membership degrees is counterintuitive for datum $\mathbf{x}_2$.

Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ 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 $U_p = (u_{ij}) = (\mu_{\Gamma_i}(\mathbf{x}_j))$ a possibilistic cluster partition of X if

$$\sum_{j=1}^{n} u_{ij} > 0, \quad \forall i \in \{1, \ldots, c\} \tag{1.15}$$

holds. The $u_{ij} \in [0, 1]$ are interpreted as the degree of representativity or typicality of the datum $\mathbf{x}_j$ to cluster Γ_i.

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 (Davé and Krishnapuram, 1997; Krishnapuram and Keller, 1993).

The objective function J_f that just minimizes the squared distances between clusters and assigned data points would not be appropriate for possibilistic fuzzy clustering. Dropping the normalization constraint leads to the mathematical problem that the objective function would reach its minimum for $u_{ij} = 0$ for all $i \in \{1, \ldots, c\}$ and $j \in \{1, \ldots, n\}$, i.e., data points are not assigned to any cluster and all clusters are empty. In order to avoid this trivial solution (that is also forbidden by constraint (1.15)), a penalty term is introduced, which forces the membership degrees away from zero. That is, the objective function J_f is modified to

$$J_p(X, U_p, C) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m d_{ij}^2 + \sum_{i=1}^{c} \eta_i \sum_{j=1}^{n} (1 - u_{ij})^m, \tag{1.16}$$

where $\eta_i > 0\,(i = 1, \ldots, c)$ (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 $(1 - u_{ij})^m$ become 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 a 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 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 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 J_p by setting its derivative to zero is (Krishnapuram and Keller, 1993):

$$u_{ij} = \frac{1}{1 + \left(\frac{d_{ij}^2}{\eta_i}\right)^{\frac{1}{m-1}}}. \tag{1.17}$$

First of all, this update equation clearly shows that the membership of a datum $\mathbf{x}_j$ 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.

Equation (1.17) 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 cluster's shape the η_i have different geometrical interpretation. If hyperspherical clusters as in the possibilistic C-means 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 (Höoppner, Klawonn, Kruse and Runkler 1999) (see Section 1.3.2). 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 clustering model of the given data-set. The η_i are then estimated by the fuzzy intra-cluster distance using the fuzzy memberships matrix 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 ($i = 1, \ldots, n$):

$$\eta_i = \frac{\sum_{j=1}^{n} u_{ij}^m d_{ij}^2}{\sum_{j=1}^{n} u_{ij}^m}. \tag{1.18}$$

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 U_p fixed). Looking at both objective functions J_f and J_p 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}. Thus the cluster centers in the possibilistic C-means algorithm are re-estimated as in Equation (1.14).

1.2.4 Comparison and Respective Properties of Probabilistic and Possibilistic Fuzzy c-means

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

Example: Figures 1.3 and 1.4 illustrate a probabilistic and a possibilistic fuzzy C-means classification of the Iris data-set into three clusters (Blake and Merz, 1998; Fisher, 1936). The displayed partitions of the data-set are the result of alternatingly optimizing J_f and J_p, respectively (Timm, Borgelt, Döring and Kruse, 2004). The grey 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. On the left, the data-set is divided into three clusters. On the right, the possibilistic fuzzy C-means algorithm detects only two clusters, since two of the three clusters in the upper right of Figure 1.4 are identical. Note that this behaviour 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.

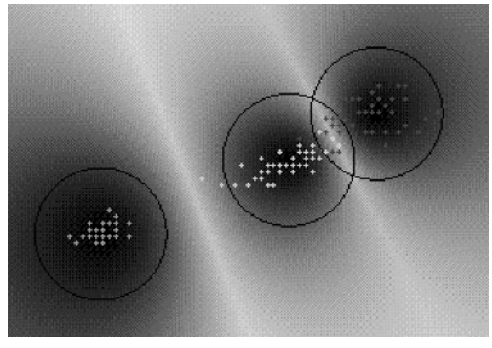

Figure 1.3 Iris data-set classified with probalistic fuzzy C-means algorithm. Attributes petal length and petal width.

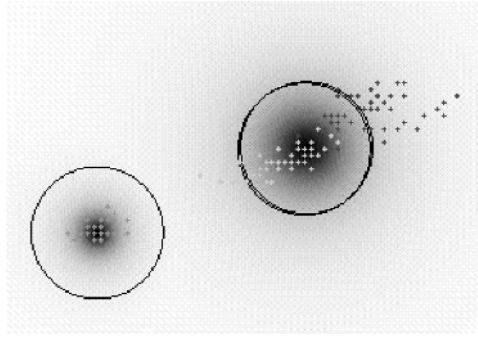

Figure 1.4 Iris data-set classifed with possibilistic fuzzy C-means algorithm. Attribtes petal length and petal width.

1.2.4.1 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 exhaustively 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, Cappellini and Mecocci, 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, Klawonn, Jruse and Runkler, 1999).

This described behaviour is exhibited, since J_p treats each cluster independently. Every cluster contributes to some extent to the value of the objective function J_p regardless of other clusters. The resulting behaviour 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, Klawonn, Kruse and Runkler, 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, Klawonn, Jruse and Runkler, 1999).

1.2.4.2 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, Borgelt, Döring and Kruse, 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 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 PCM model in Figure 1.4 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 (see Section 1.4.4). These modifications of J_p 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 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 using a combination of both fuzzy and possibilistic memberships (see Section 1.4.4).

1.2.4.3 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. The latter ones are known to be unable 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 Figure 1.3). Higher memberships to data points will be assigned in directions pointing away from the overlap. Thus the centers are repelling each other. If complex prototypes are used, detected 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. In probabilistic fuzzy clustering, noise clustering techniques are widely appreciated (see Section 1.4.1). In one of the noise handling approaches, the objective function J_f is modified such that a virtual noise cluster "seizes" parts of the data point weight of noise points and outliers. This leads to better detection of actual cluster structure in probabilistic models.

1.3 DISTANCE FUNCTION VARIANTS

In the previous section, we considered the case where the distance between cluster centers and data points is computed using the Euclidean distance, leading to the standard versions of fuzzy C-means and possibilistic C-means. This distance only makes it possible to identify spherical clusters. Several variants have been proposed to relax this constraint, considering other distances between cluster centers and data points. In this section, we review some of them, mentioning the fuzzy Gustafson–Kessel algorithm, fuzzy shell clustering algorithms and kernel-based variants. All of them can be applied both in the fuzzy probabilistic and possibilistic framework.

Please note that a more general algorithm is provided by the fuzzy relational clustering algorithm (Hathaway and Bezdek, 1994) that takes as input a distance matrix. In this chapter, we consider the variants that handle object data and do not present the relational approach.

1.3.1 Gustafson–Kessel Algorithm

The Gustafson–Kessel algorithm (Gustafson and Kessel, 1979) replaces the Euclidean distance by a cluster-specific Mahalanobis distance, so as to adapt to various sizes and forms of the clusters. For a cluster i, its associated Mahalanobis distance is defined as

$$d^2(\mathbf{x}_j, C_i) = (\mathbf{x}_j - \mathbf{c_i})^T \Sigma_i^{-1} (\mathbf{x}_j - \mathbf{c_i}), \tag{1.19}$$

where Σ_i is the covariance matrix of the cluster. Using the Euclidean distance as in the algorithms presented in the previous section is equivalent to assuming that $\forall i, \Sigma_i = I$, i.e., all clusters have the same covariance that equals the identity matrix. Thus it only makes it possible to detect spherical clusters, but it cannot identify clusters having different forms or sizes.

The Gustafson–Kessel algorithm models each cluster Γ_i by both its center $\mathbf{c}_i$ and its covariance matrix $\Sigma_i, i = 1, \ldots, c$. Thus cluster prototypes are tuples $C_i = (\mathbf{c}_i, \Sigma_i)$ and both $\mathbf{c}_i$ and Σ_i are to be learned. The eigenstructure of the positive definite $p \times p$ matrix Σ_i represents the shape of cluster i. Specific constraints can be taken into account, for instance restricting to axis-parallel cluster shapes, by considering only diagonal matrices. This case is usually preferred when clustering is applied for the generation of fuzzy rule systems (Höppner, Klawonn, Kruse, and Runkler, 1999). 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 $\det(\Sigma_i) = 1$.

The objective function is then identical to the fuzzy C-means (see Equation (1.10)) or the possibilistic one (see Equation (1.16)), using as distance the one represented above in Equation (1.19). The update equations for the cluster centers $\mathbf{c}_i$ are not modified and are identical to those indicated in Equation (1.14). The update equations for the membership degrees are identical to those indicated in Equation (1.13) and Equation (1.17) for the FCM and PCM variants respectively, replacing the Euclidean distance by the cluster specific distance given above in Equation (1.19). The update equations for the covariance matrices are

$$\Sigma_i = \frac{\Sigma_i^*}{\sqrt[p]{\det(\Sigma_i^*)}}, \qquad \text{where} \qquad \Sigma_i^* = \frac{\sum_{j=1}^n u_{ij} (\mathbf{x}_j - \mathbf{c_i})(\mathbf{x}_j - \mathbf{c_i})^T}{\sum_{j=1}^n u_{ij}}. \tag{1.20}$$

They are defined as the covariance of the data assigned to cluster i, modified to incorporate the fuzzy assignment information.

The Gustafson–Kessel algorithm tries to extract much more information from the data than the algorithms based on the Euclidean distance. It is more sensitive to initialization, therefore it is recommended to initialize it using a few iterations of FCM or PCM depending on the considered partition type. Compared with FCM or PCM, the Gustafson–Kessel algorithm exhibits higher computational demands due to the matrix inversions. A restriction to axis-parallel cluster shapes reduces computational costs.

1.3.2 Fuzzy Shell Clustering

The clustering approaches mentioned up to now search for convex ''cloud-like'' clusters. The corresponding algorithms are called *solid* clustering algorithms. They are ''specially useful'' in data analysis applications. Another area of application of fuzzy clustering algorithms is image recognition and analysis. Variants of FCM and PCM have been proposed to detect lines, circles or ellipses on the data-set, corresponding to more complex data substructures; the so-called *shell* clustering algorithms (Klawonn, Kruse, and Timm, 1997) extract prototypes that have a different nature than the data points. They need to modify the definition of the distance between a data point and the prototype and replace the Euclidean by other distances. For instance the fuzzy c-varieties (FCV) algorithm was developed for the recognition of lines, planes, or hyperplanes; each cluster is an affine subspace characterized by a point and a set of

orthogonal unit vectors, $C_i = (\mathbf{c}_i, \mathbf{e}_{i1}, \ldots, \mathbf{e}_{iq})$ where q is the dimension of the affine subspace. The distance between a data point $\mathbf{x}_j$ and cluster i is then defined as

$$d^2(\mathbf{x}_j, C_i) = ||\mathbf{x}_j - \mathbf{c}_i||^2 - \sum_{l=1}^{q} (\mathbf{x}_j - \mathbf{c}_i)^T \mathbf{e}_{il}.$$

The fuzzy c-varieties (FCV) algorithm is able to recognize lines, planes or hyperplanes (see Figure 1.5). These algorithms can also be used for the construction of locally linear models of data with underlying functional interrelations.

Other similar FCM and PCM variants include the adaptive fuzzy c-elliptotypes algorithm (AFCE) that assigns disjoint line segments to different clusters (see Figure 1.6). Circle contours can be detected by the fuzzy c-shells and the fuzzy c-spherical shells algorithm. Since objects with circle-shaped boundaries in 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 Figures 1.7 and 1.8. The shell clustering techniques have also been extended to non-smooth structures such as rectangles and other polygons. Figures 1.9 and 1.10 illustrate results obtained with the fuzzy c-rectangular (FCRS) and fuzzy c-2-rectangular shells (FC2RS) algorithm. The interested reader may be referred to Höppner, Klawonn, Kruse, and Runkler (1999) and Bezdek, Keller, Krishnapuram, and Pal (1999) for a complete discussion of this branch of methods.

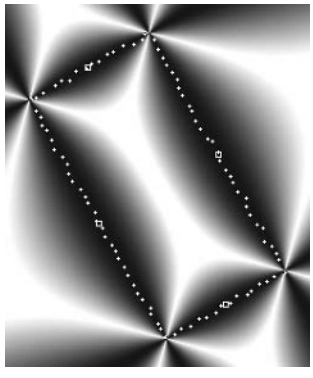

Figure 1.5 FCV analysis.

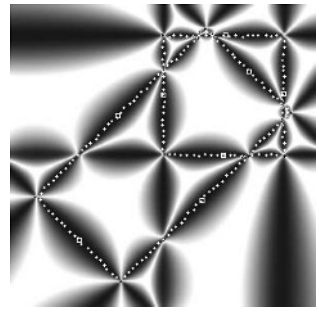

Figure 1.6 AFCE analysis.

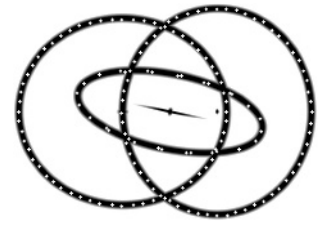

Figure 1.7 FCQS analysis.

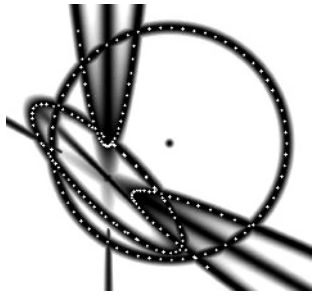

Figure 1.8 FCQS analysis.

Figure 1.9 FCRS analysis.

Figure 1.10 FC2RS analysis.

1.3.3 Kernel-based Fuzzy Clustering

The kernel variants of fuzzy clustering algorithms further modify the distance function to handle non-vectorial data, such as sequences, trees, or graphs, without needing to modify completely the algorithms themselves. Generally speaking, kernel learning methods (see e.g., Schölkopf and Smola (2002); Vapnik (1995)) constitute a set of machine learning algorithms that make it possible to extend, in a formal framework, classic linear algorithms. This extension addresses a double aim: on the one hand, it makes it possible to address tasks that require a richer framework than the linear one, while still preserving this generally simple formalism. On the other hand, it makes it possible to apply algorithms to data that are not described in a vectorial form, but as more complex objects, such as sequences, trees or graphs. More generally, kernel methods can be applied independently of the data nature, without needing to adapt the algorithm. In this section, data points can be vectorial or not, therefore we denote them x_j instead of $\mathbf{x}_j$.

1.3.3.1 Principle

Kernel methods are based on an *implicit* data representation transformation $\phi : \mathcal{X} \rightarrow \mathcal{F}$ where $\mathcal{X}$ denotes the input space and $\mathcal{F}$ is called the *feature space*. $\mathcal{F}$ is usually of high or even infinite dimension and is only constrained to be a Hilbert space, i.e., to dispose of a scalar product. The second principle of kernel methods is that data are not handled directly in the feature space, which could lead to expensive costs given its dimension; they are only handled through their scalar products that are computed using the initial representation. To that aim, the so-called *kernel function* is used: it is a function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, such that $\forall x, y \in \mathcal{X}, \langle \phi(x), \phi(y) \rangle = k(x, y)$ Thus the function ϕ is not needed to be known explicitly, scalar products in the feature space only depend on the initial representation.

In order to apply this *kernel trick*, kernel methods are algorithms written only in terms of scalar products between the data. The data representation enrichment then comes from using a scalar product based on an implicit transformation of the data, instead of being only the Euclidean one. The possibility to apply the algorithm to non-vectorial data only depends on the availability of a function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ having the properties of a scalar product (Schölkopf and Smola, 2002).

1.3.3.2 Kernel Fuzzy Clustering

The kernel framework has been applied to fuzzy clustering and makes it possible to consider other distances than the Euclidean one. It is to be underlined that fuzzy shell clustering, discussed in Section 1.3.2, also takes into account other metrics, but it has an intrinsic difference: it aims at extracting prototypes that have a different nature than the data points, and thus it modifies the distance between points and cluster prototypes. In the kernel approach, the similarity is computed between pairs of data points and does not involve cluster centers; the kernel function influences more directly that points are to be grouped in the same clusters, and does not express a comparison with a cluster representative. Usually, cluster representatives have no explicit representation as they belong to the feature space. Thus the kernel approach can be applied independently of the data nature whereas fuzzy shell algorithms must be specified for each desired prototype nature. On the other hand, kernel methods do not have an explicit representative of the cluster and cannot be seen as prototype-based clustering methods.

The kernel variant of fuzzy clustering (Wu, Xie, and Yu, 2003) consists of transposing the objective function to the feature space, i.e., applying it to the transformed data $\phi(x)$. The cluster centers then belong to the feature space, we therefore denote them $c_i^\phi, i = 1, \ldots, c$ ($c_i^\phi \in \mathcal{F}$). They are looked for in the form of linear combinations of the transformed data, as

$$c_i^\phi = \sum_{r=1}^{n} a_{ir} \phi(x_r). \tag{1.21}$$

This formulation is coherent with the solution obtained with standard FCM. Optimization must then provide the a_{ir} values, together with the membership degrees. Due to the previous form of the centers, the Euclidean distance between points and centers in the feature space can be computed as

$$d\phi_{ir}^2 = ||\phi(x_r) - c_i^{\phi}||^2 = k_{rr} - 2\sum_{s=1}^{n} a_{is}k_{rs} + \sum_{s,t=1}^{n} a_{is}a_{it}k_{st}, \tag{1.22}$$

where we denote $k_{rs} = k(x_r, x_s) = \langle \phi(x_r), \phi(x_s) \rangle$. Thus, the objective function becomes

$$J^{\phi} = \sum_{i=1}^{c}\sum_{r=1}^{n} u_{ir}^m \left(k_{rr} - 2\sum_{s=1}^{n} a_{is}k_{rs} + \sum_{s,t=1}^{n} a_{is}a_{it}k_{st} \right). \tag{1.23}$$

The minimization conditions then lead to the following update equations

$$u_{ir} = \frac{1}{\sum_{l=1}^{c} \left(\frac{d\phi_{ir}^2}{d\phi_{lr}^2} \right)^{\frac{1}{m-1}}}, \qquad a_{ir} = \frac{u_{ir}^m}{\sum_{s=1}^{n} u_{is}^m}, \qquad \text{i.e.,} \quad c_i^{\phi} = \frac{\sum_{r=1}^{n} u_{ir}^m \phi(x_r)}{\sum_{s=1}^{n} u_{is}^m}. \tag{1.24}$$

Thus the update equations, as well as the objective function, can be expressed solely in terms of the kernel function, i.e., in terms of scalar products. Equation (1.24) shows that membership degrees have the same form as in the standard FCM (see Equation (1.13)), replacing the Euclidean distance by the distance in the feature space, as defined in Equation (1.22). The expression of the cluster centers is comparable to the standard case (see Equation (1.14)), as the weighted mean of the data. The difference is that cluster centers belong to the feature space and have no explicit representation, only the weighting coefficients are known.

There exist other variants for the kernelization of the fuzzy C-means, as for instance the one proposed by Zhang and Chen (2003a,b). The latter is specific insofar as it only considers the Gaussian kernel $k(x, y) = \exp(-d(x, y)^2/\sigma^2)$ and exploits its properties to simplify the algorithm. More precisely it makes the hypothesis that cluster centers can be looked for explicitly in the input space ($c_i \in \mathcal{X}$), and considers its transformation to the feature space $\phi(c_i)$. This differs from the general case, as presented above, where cluster centers are only defined in the feature space. The objective function then becomes

$$J = \sum_{i=1}^{c}\sum_{r=1}^{n} u_{ir}^m ||\phi(c_i) - \phi(x_j)||^2 = 2\sum_{i=1}^{c}\sum_{r=1}^{n} u_i^m (1 - \mathrm{e}^{-d(c_i, x_j)^2/\sigma^2}), \tag{1.25}$$

exploiting the fact that the Gaussian kernel leads to $d\phi^2(x, y) = k(x, x) + k(y, y) - 2k(x, y) = 2(1 - k(x, y))$. Thus this method constitutes a special case of the FCM kernelization and cannot be applied to any type of data independently of their nature. It is to be noted that this objective function (Equation (1.25)) is identical to the one proposed by Wu and Yang (2002) in the framework of robust variants of FCM, as described in the next section.

It should be noticed that the application of a kernel method needs to select the kernel and its parameters, which may be difficult. This task can be seen as similar to the problem of feature selection and data representation choice in the case of non-kernel methods.

1.4 OBJECTIVE FUNCTION VARIANTS

The previous variants of fuzzy C-means are obtained when considering different distance functions that lead to a rewrite of the objective functions and in some cases modify the update equations. In this section, we consider other variants that are based on deeper modifications of the objective functions. The modifications aim at improving the clustering results in specific cases, for instance when dealing with noisy data. It is to be noticed that there exists a very high number of variants for fuzzy clustering algorithms, we only mention some of them.

We organized them in the following categories: some variants are explicitly aimed at handling noisy data. Others study at a theoretical level the role of the fuzzifier m in the objective function (see notations in Equation (1.10)) and propose some modifications. Other variants introduce new terms in the objective function so as to optimize the cluster number instead of having it fixed at the beginning of the process. Lastly, we mention some variants that are aimed at improving the possibilistic C-means, in particular with respect to the coinciding cluster problem (see Section 1.2.4).

It is to be noted that the limits between these categories are not clear-cut and that for instance the modification of the fuzzifier can influence the noise handling properties. We categorize the methods according to their major characteristics and underline their other properties.

When giving update equations for cluster prototypes, we consider only the case where the Euclidean distance is used and when prototypes are reduced to cluster centers. Most methods can be generalized to other representations, in particular those including size and form parameters. The interested reader is referred to the original papers.

1.4.1 Noise Handling Variants

The first variants of fuzzy C-means we consider aim at handling noisy data. It is to be noticed that PCM is a solution to this problem, but it has difficulty of its own as mentioned in Section 1.2.4 (cluster coincidence problem, sensitivity to initialization). Therefore other approaches take FCM as the starting point and modify it so as to enable it to handle noisy data. When giving the considered objective functions, we do not recall the constraints indicated in Equations (1.8) and (1.9) that apply in all cases.

The aim of these variants is then to define robust fuzzy clustering algorithms, i.e., algorithms whose results do not depend on the presence or absence of noisy data points or outliers[1] in the data-set. Three approaches are mentioned here: the first one is based on the introduction of a specific cluster, the so-called noise cluster that is used to represent noisy data points. The second method is based on the use of robust estimators, and the third one reduces the influence of noisy data points by defining weights denoting the point representativeness.

1.4.1.1 Noise Clustering

The noise clustering (NC) algorithm was 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, from all data points. This means that all points have a priori the same 'probability' of belonging 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

$$J = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m} d_{ij}^{2} + \sum_{k=1}^{n} \delta^{2} \left(1 - \sum_{i=1}^{c} u_{ik}\right)^{m}. \tag{1.26}$$

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

[1]Outliers correspond to atypical data points, that are very different from all other data, for instance located at a high distance from the major part of the data. More formally, according to Hawkins (1980), an outlier is 'an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism'.

on the standard cluster: as PCM, the noise clustering approach relaxes the FCM normalization constraint expressed in Equation (1.9) according to which membership degrees to good clusters must sum to 1.

Further comparison between NC and PCM (see Equations (1.26) and (1.16) shows that the algorithms are identical in the case of a single cluster, with δ^2 corresponding to η (Davé and Sen, 1997, 1998). In the case $c > 1$, the difference is that PCM considers one η_i per cluster, whereas a single parameter is defined in the NC case. This means that PCM has the advantage of having one noise class per good cluster, whereas NC has only one (the NC generalization described hereafter overcomes this drawback). As a consequence, the membership degrees to the noise cluster differ for the two methods: in the PCM case, they are, for each noise cluster, the complement to 1 to the membership to the associated good cluster. In noise clustering, as there is a single noise cluster, the membership degree to it is the complement to the sum of all other memberships.

Another difference between PCM and NC comes from the fact that the PCM cost function can be decomposed into c independent terms (one per cluster), whereas in the noise clustering approach such a decomposition is not possible. This decomposition is one of the reasons why PCM leads to coinciding clusters. Thus Davé and Krishnapuram (1997) interpret NC as a robustified FCM, whereas PCM behaves like c independent NC algorithms.

The objective function (1.26) requires the setting of parameter δ. In the initial NC algorithm, it was set to

$$\delta^2 = \lambda \frac{1}{c \cdot n} \left(\sum_{i=1}^{c} \sum_{j=1}^{n} d_{ij}^2 \right), \tag{1.27}$$

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.

Noise clustering has been generalized to allow the definition of several δ, and to define a noise scale per cluster. To that aim, each point is associated to a noise distance $\delta_j, j = 1, \ldots, n$, the latter being defined as the size of the cluster the point maximally belongs to, as in PCM: $\delta_j = \eta_{i*}$ for $i* = \arg\max_l u_{lj}$ (Davé and Sen, 1997, 1998). In this case, the difference between PCM and NC about distance scale vanishes, the only remaining difference is the independence of clusters in the PCM objective function that does not appear in the noise clustering case.

1.4.1.2 Robust Estimators

Another approach to handle noisy data-sets is based on the exploitation of robust estimators: as indicated in Section 1.2.2, the fuzzy C-means 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 FCM gives unsatisfactory results when applied to data-sets contaminated with noise and outliers. Therefore, it has been proposed to introduce a robust estimator in the FCM classic objective function (see Equation (1.10)), leading to consider

$$J = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m \rho_i(d_{ij}), \tag{1.28}$$

where ρ_i are robust symmetric positive definite functions having their minimum in 0 (Frigui and Krishnapuram, 1996). According to the robust M-estimator framework, ρ should be chosen such that $\rho(z) = \log(J(z)^{-1})$ 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 another 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 behaviour, defining the robust c-prototypes (RCP) algorithm.

In the case where clusters are represented only by centers and a probabilistic partition is looked for (i.e., with constraint (1.9)), the update equations for the membership degrees and cluster prototypes derived from Equation (1.28) then become (Frigui and Krishnapuram, 1996)

$$\mathbf{c}_i = \frac{\sum_{j=1}^{n} u_{ij}^m f_{ij} \mathbf{x}_j}{\sum_{j=1}^{n} u_{ij}^m f_{ij}}, \qquad u_{ij} = \frac{1}{\sum_{k=1}^{c} \left[\frac{\rho(d_{ij}^2)}{\rho(d_{kj}^2)} \right]^{\frac{1}{m-1}}}, \tag{1.29}$$

where $f_{ij} = f(d_{ij})$ and $f = \frac{\mathrm{d}\rho(z)}{\mathrm{d}z}$. It is to be noted that outliers still have membership degrees $u_{ij} = 1/c$ for all clusters. The difference and advantage as compared with 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).

Other robust clustering algorithms include the method proposed by Wu and Yang (2002) that consider the modified objective function

$$J = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m (1 - \mathrm{e}^{-\beta d_{ij}^2}), \tag{1.30}$$

where β is a user-defined parameter that the authors propose to set to the inverse of the sample covariance matrix. This function is first proposed as a replacement of the Euclidean distance by the more robust exponential metric; yet, as pointed out by Zhang and Chen (2004), the mapping $(x, y) \mapsto \exp(-\beta d(x, y))$ is not a metric. Still, the analysis of the above objective function in the robust estimator framework holds and shows that this function leads to a robust fuzzy clustering algorithm that can handle noisy data-sets Wu and Yang (2002).

Davé and Krishnapuram (1996, 1997) show that PCM can be interpreted in this robust clustering framework based on the M-estimator. They consider a slightly different formalization, where the objective function for each cluster is written

$$J = \sum_{j=1}^{n} \rho(\mathbf{x}_j - \mathbf{c}), \quad \text{leading to} \quad \mathbf{c} = \frac{\sum_{j=1}^{n} w(d_{ij}) \mathbf{x}_j}{\sum_{j=1}^{n} w(d_{ij})}, \quad \text{where} \quad w(z) = \frac{1}{z} \frac{\mathrm{d}\rho}{\mathrm{d}z}. \tag{1.31}$$

Comparing with the update equations of PCM, this makes it possible to identify a weight function w and by integration to deduce the associated estimator ρ. Davé and Krishnapuram (1996, 1997) show the obtained ρ is indeed a robust function. This justifies at a formal level the qualities of PCM as regards noise handling.

1.4.1.3 Weight Modeling

A third approach to handle outliers is exemplified by Keller (2000). It consists of associating each data point a weight to control the influence it can have on the cluster parameters. The considered objective function is

$$J = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m \frac{1}{\omega_j^q} d_{ij}^2, \quad \text{under constraint} \quad \sum_{j=1}^{n} \omega_j = \omega, \tag{1.32}$$

where the factor ω_j represents the weight for data point j, q a parameter to control the influence of the weighting factor and ω a normalizing coefficient. The minimization conditions of this objective function lead to the following update equations:

$$u_{ij} = \frac{1}{\sum_{l=1}^{c} \left(\frac{d_{ij}^2}{d_{lj}^2} \right)^{\frac{1}{m-1}}}, \qquad \mathbf{c}_i = \frac{\sum_{j=1}^{n} \frac{u_{ij}^m}{\omega_j^q} \mathbf{x}_j}{\sum_{j=1}^{n} \frac{u_{ij}^m}{\omega_j^q}}, \qquad \omega_j = \frac{\left(\sum_{i=1}^{c} u_{ij}^m d_{ij}^2 \right)^{\frac{1}{q+1}}}{\sum_{l=1}^{n} \left(\sum_{i=1}^{c} u_{il}^m d_{il}^2 \right)^{\frac{1}{q+1}}} \omega.$$

Thus, the membership degrees are left unchanged, whereas the cluster centers take into account the weights; points with high representativeness play a more important role than outliers. Representativeness depends on the weighted average distance to cluster centers.

1.4.2 Fuzzifier Variants

Another class of FCM variants is based on the study of the fuzzifier, i.e., the exponent m in Equation (1.10): as indicated in Section 1.2.2, FCM can be derived from the hard C-means algorithm by relaxing the partition constraints, so that membership degrees belong to [0,1] and not {0,1}. To prevent membership degrees from being restricted to the two values 0 and 1, the objective function must be modified and the m fuzzifier is introduced.

Now as can be observed and proved (Klawonn and Höppner, 2003b; Rousseeuw, Trauwaert, and Kautman, 1995), actually membership degrees do not exactly cover the range [0,1]: they never equal 0 or 1 (except in the special case where a data point coincides with a cluster center), i.e., in fact they belong to]0,1[. In other words, membership functions have a core reduced to a single point (the cluster center) and unbounded support. This is a drawback in the case of noisy data-sets, as in the case of clusters with different densities (Klawonn and Höppner, 2003b; Rousseeuw, Trauwaert and Kautman, 1995): high density clusters tend to influence or completely attract other prototypes (note that this problem can be handled by using other distances than the Euclidean one).

To overcome this problem, Rousseeuw, Trauwaert and Kaufman, (1995) proposed replacing the objective function by

$$J = \sum_{i=1}^{c} \sum_{j=1}^{n} [\alpha u_{ij} + (1 - \alpha) u_{ij}^2] d_{ij}^2, \tag{1.33}$$

where α is a user-defined weight determining the influence of each component. When $\alpha = 1$, the objective function reduces to the hard C-means function (see Equation (1.1)), leading to a maximal contrast partition (membership degrees take only values 0 or 1). On the contrary, $\alpha = 0$ leads to the fuzzy C-means with $m = 2$ and a low contrast partition (outliers for instance have the same membership degree as all clusters). α makes it possible to obtain a compromise situation, where membership degrees in]0,1[are reserved for points whose assignment is indeed unclear, whereas the others, and in particular outliers, have degrees 0 or 1.

Klawonn and Höppner, (2003a,b) also take as their starting point the observation that membership degrees actually never take the values 0 and 1. They perform the analysis in a more formal framework that allows more general solutions: they proposed considering as objective function

$$J = \sum_{i=1}^{c} \sum_{j=1}^{n} g(u_{ij}) d_{ij}^2. \tag{1.34}$$

Note that robust approaches proposed applying a transformation to the distances, whereas here a transformation is applied to the membership degrees. Taking into account the constraints on u_{ij} normalization (see Equation (1.9)), and setting the derivative to 0, the partial derivative of the associated Lagrangian leads to

$$g'(u_{ij}) d_{ij}^2 - \lambda_j = 0, \tag{1.35}$$

where λ_j is the Lagrange multiplier associated with the normalization constraint concerning x_j. As it is independent of i, this equation implies $g'(u_{ij}) d_{ij}^2 = g'(u_{kj}) d_{kj}^2$ for all i, k. This explains why zero membership degrees can never occur: the standard function $g(u) = u^m$ yields $g'(0) = 0$. Thus, in order to balance the two terms, no matter how large d_{ij}^2 and how small d_{kj}^2 are, u_{ij} cannot be 0.

Therefore, they proposed replaceing the standard g function with other ones. The conditions g must satisfy are $g(0) = 0$ and $g(1) = 1$, increasing and differentiable. Further, the derivative g' must be

increasing and must satisfy $g'(0) \neq 0$. Klawonn and Höppner, (2003b) consider the same function as Rousseeuw, Trauwaert, and Kautman (1995), i.e., $g(u) = \alpha u^2 + (1-\alpha)u$. Gaussian functions $g(u) = (e^{\alpha u} - 1)/(e^{\alpha} - 1)$ were also suggested, since the parameter α has a similar effect to the fuzzifier m in the standard fuzzy clustering: the smaller the α, the crisper the partition tends to be (Klawonn and Höppner, 2003a). Klawonn (2004) proposesd dropping the differentiability condition and considering a piecewise linear transformation to obtain more flexibility than with a single parameter α. For instance, non-increasing functions that are flatter around 0.5 make it possible to avoid ambiguous membership degrees forcing them to tend to 0 or 1.

1.4.3 Cluster Number Determination Variants

Partitioning clustering algorithms consist of searching for the optimal fuzzy partition of the data-set into c clusters, where c is given as input to the algorithm. In most real data mining cases, this parameter is not known in advance and must be determined. Due to the cluster merging phenomenon, the definition of an appropriate c value for PCM is not so important as for FCM. Yet, as mentioned earlier, at a theoretical level, PCM relies on an ill-posed optimization problem and other approaches should be considered. They usually consist of testing several c values and comparing the quality of the obtained partition using so-called validity criteria (see for instance Halkidi, Batistakis, and Vazirgiannis (2002); this solution is computationally expensive. Other approaches, presented in this section, consist of considering the c value as a parameter to be optimized.

Now with this respect the FCM objective function is minimal when $c = n$, i.e., each cluster contains a single point as in this case $d_{ij} = 0$. Thus a regularization term is added, that is minimal when all points belong to the same cluster, so as to penalize high c values. Then the combination of terms in the objective function makes it possible to find the optimal partition in the smallest possible number of clusters.

Following this principle, Frigui and Krishnapuram (1997) proposed the competitive agglomeration (CA) algorithm based on the objective function

$$J = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m d_{ij}^2 - \alpha \sum_{i=1}^{c}\left(\sum_{j=1}^{n} u_{ij}\right)^2. \tag{1.36}$$

The additional term is the sum of squares of cardinalities of the clusters, which is indeed minimal when all points are assigned to a single cluster and all others are empty. The optimization process for this function does not exactly follow the AO scheme and involves competition between clusters, based on their sizes and distances to the points. Small clusters are progressively eliminated. A robust extension to CA has been proposed in Frigui and Krishnapuram (1999): the first term in Equation (1.36) is then replaced by the term provided in Equation (1.28) to exploit the robust estimator properties.

Sahbi amd Boujemaa (2005) proposed using as regularizer an entropy term, leading to

$$J = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m d_{ij}^2 - \alpha \frac{1}{n}\sum_{j=1}^{n} - \sum_{i=1}^{c} u_{ij}\log_2(u_{ij}).$$

To verify the constraints on the memberships $u_{ij} \in [0,1]$, they proposed considering Gaussian membership functions in the form $u_{ij} = \exp(-\mu_{ij})$ and estimating the μ_{ij} parameters. α then intervenes in the parameter of the exponential and is to be interpreted as a scaling factor: when it is underestimated, each point is a cluster; when it is overestimated, the membership functions are approximately constant, and one gets a single big cluster. The number of clusters is then indirectly determined.

1.4.4 Possibilistic c-means Variants

As indicated in Section 1.24, the possibilistic C-means may lead to unsatisfactory results, insofar as the obtained clusters may be coincident. This is due to the optimized objective function, whose global

minimum is obtained when all clusters are identical (see Section 1.2.4). Hence the possibilistic C-means can be improved by modifying its objective function. We mention here two PCM variants, based on the adjunction of a penalization term in the objective function and the combination of PCM with FCM.

1.4.4.1 Cluster Repulsion

In order to hinder cluster merging, Timm and Kruse (2002) and Timm, Borgelt, Döreing, and Kruse (2004) proposed including in the objective function a term expressing repulsion between clusters, so as to force them to be distinct: the considered objective function is written

$$J = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m} d_{ij}^{2} + \sum_{i=1}^{c} \eta_i \sum_{i=1}^{n} (1-u_{ij})^{m} + \sum_{i=1}^{c} \gamma_i \sum_{k=1,k\neq i}^{c} \frac{1}{\xi d(\mathbf{c}_i,\mathbf{c}_j)^2}. \tag{1.37}$$

The first two terms constitute the PCM objective function (see Equation (1.16)), the last one expresses the repulsion between clusters: it is all the bigger as the distance between clusters is small. γ_i is a parameter that controls the strength of the cluster repulsion: it balances the two clustering objectives, namely the fact that clusters should be both compact and distinct. This coefficient depends on clusters so that repulsion can get stronger when the number of points associated with cluster i increases (Timm, Borgelt, Döring, and Kruse, 2004). Parameter ξ makes repulsion independent of the normalization of data attributes. The minimization conditions lead to the update equation

$$\mathbf{c}_i = \frac{\sum_{j=1}^{n} u_{ij}^{m}\mathbf{x}_j - \gamma_i \sum_{k=1,k\neq i}^{c} \frac{1}{d(\mathbf{c}_i,\mathbf{c}_k)^4}\mathbf{c}_k}{\sum_{j=1}^{n} u_{ij}^{m} - \gamma_i \sum_{k=1,k\neq i}^{c} \frac{1}{d(\mathbf{c}_i,\mathbf{c}_k)^4}} \tag{1.38}$$

(the update equation for the membership degrees is not modified and is identical to Equation (1.17)). Equation (1.38) shows the effect of repulsion between clusters: a cluster is attracted by the data assigned to it and it is simultaneously repelled by the other clusters.

1.4.4.2 PCM Variants Based on Combination with FCM

Pal, Pal, and Bezdek (1997) and Pal, Pal, Keller, and Bezdek (2004) proposed another approach to overcome the problems encountered with the possibilistic C-means: they argued that both possibilistic degrees and membership degrees are necessary to perform clustering. Indeed, possibilistic degrees make it possible to reduce the influence of outliers whereas membership degrees are necessary to assign points. Likewise, Davé and Sen (1998) underlined that a good clustering result requires both the partitioning property of FCM and the modeseeking robust property of PCM.

In Pal, Pal, and Bezdek (1997) the combination of FCM and PCM is performed through the optimization of the following objective function:

$$J = \sum_{i=1}^{c}\sum_{j=1}^{n} (u_{ij}^{m} + t_{ij}^{\eta}) d_{ij}^{2}, \quad \text{under the constraints} \quad \begin{cases} \forall j & \sum_{i=1}^{c} u_{ij} = 1 \\ \forall i & \sum_{j=1}^{n} t_{ij} = 1 \end{cases}. \tag{1.39}$$

This means that u_{ij} is a membership degree, whereas t_{ij} corresponds to a possibilistic coefficient. Indeed, it is not submitted to the normalization constraint on the sum across the clusters. The normalization constraint it must hold aims at preventing the trivial result where $t_{ij} = 0$ for all i,j. As pointed out in several papers (Davé and Sen, 1998; Pal, Pal, Keller, and Bezdek, 2004) the problem is that the relative scales of probabilistic and possibilistic coefficients are then different and the membership degrees dominate the equations. Moreover, the possibilistic coefficients take very small values in the case of big data-sets.

Therefore Pal, Pal, Keller and Bezdek (2004) proposed another combination method, in the form

$$J = \sum_{i=1}^{c}\sum_{j=1}^{n}(au_{ij}^{m} + bt_{ij}^{\eta})d_{ij}^{2} + \sum_{i=1}^{c}\eta_i\sum_{j=1}^{n}(1-t_{ij})^{\eta}, \tag{1.40}$$

which uses the same constraint for t_{ij} as in the standard PCM (second term in J), and combines possibilistic and membership degrees. 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

$$u_{ij} = \frac{1}{\sum_{l=1}^{c}\left(\frac{d_{ij}^2}{d_{lj}^2}\right)^{\frac{1}{m-1}}}, \qquad t_{ij} = \frac{1}{1+\left(\frac{b}{\eta_i}d_{ij}^2\right)^{\frac{1}{\eta-1}}}, \qquad \mathbf{c}_i = \frac{\sum_{j=1}^{n}(au_{ij}^{m} + bt_{ij}^{\eta})\mathbf{x}_j}{\sum_{j=1}^{n}(au_{ij}^{m} + bt_{ij}^{\eta})}.$$

Thus u_{ij} are similar to the membership degrees of FCM (see Equation (1.13)), and t_{ij} to the possibilistic coefficients of PCM when replacing η_i with η_i/b (see Equation (1.17)). 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 these 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 it loses the interpretation of the obtained coefficients; in particular, due to their interaction, t_{ij} cannot be interpreted as typicality anymore.

1.5 UPDATE EQUATION VARIANTS: ALTERNATING CLUSTER ESTIMATION

In this section, we study the fuzzy clustering variants that generalize the alternating optimization scheme used by the methods presented up to now. The notion alternating cluster estimation (ACE) stands for a distinguished methodology to approach clustering tasks with the aim of having the flexibility to tailor new clustering algorithms that better satisfy application-specific needs. Instead of reformulating the clustering task as a minimization problem by defining objective functions, the data analyst chooses cluster prototypes that satisfy some desirable properties as well as cluster membership functions that have better suited shapes for particular applications. This is possible, since the ACE framework generalizes the iterative updating scheme for cluster models that stems from the alternating optimization approaches (Equation (1.11 and 1.12)). However, the purpose of minimizing objective functions with expressions for j_U and j_C is abandoned. Instead, the user chooses heuristic equations to re-estimate partitions and cluster parameters by which the resulting algorithm iteratively refines the cluster model. Thus the classification task is directly described by the chosen update equations, which do not necessarily reflect the minimization of some criterion anymore.

Alternating cluster estimation is justified by the observation that convergence is seldom a problem in practical examples (local minima or saddle points can be avoided). The ACE framework is particularly useful when cluster models become too complex to minimize them analytically or when the objective function lacks differentiability (Höppner, Klawonn, Kruse, and Runkler 1999). However, it is to be noted that the ACE framework also encompasses all those algorithms that follow from the minimization of objective functions as long as their respective update equations are chosen (which follow from the necessary conditions for a minimum).

When clustering is applied to the construction of fuzzy rule-based systems, the flexibility of ACE framework in choosing among different update equations is of particular interest. 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 $\mu_{\Gamma_i}(\mathbf{x}_j) = u_{ij}$ 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 Equation (1.13), with d_{ij} being the Euclidian distance $||\cdot||$:

$$\mu_i(\mathbf{x}) = \frac{||\mathbf{x} - \mathbf{c}_i||^{-\frac{2}{m-1}}}{\sum_{l=1}^{c} ||\mathbf{x} - \mathbf{c}_l||^{-\frac{2}{m-1}}}. \tag{1.41}$$

Figure 1.11 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 to 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 Equation (1.17) 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 membership 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 you 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 choose from parameterized Gaussian, trapezoidal, Cauchy, and triangular functions (Höppner, Klawonn, Kruse, and Runkler, 1999). We present the triangular shaped fuzzy set as an example in Figure 1.12, since it has all the desired properties considered above:

$$\mu_i(\mathbf{x}) = \begin{cases} 1 - \left(\frac{||\mathbf{x}-\mathbf{c}_i||}{r_i}\right)^{\alpha} & \text{if} \quad ||\mathbf{x} - \mathbf{c}_i|| \leq r_i \\ 0 & \text{otherwise,} \end{cases} \tag{1.42}$$

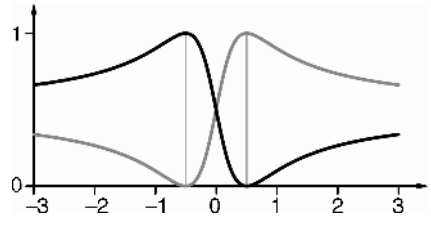

Figure 1.11 The membership functions obtained by probabilistic AO for two clusters at −0.5 and 0.5.

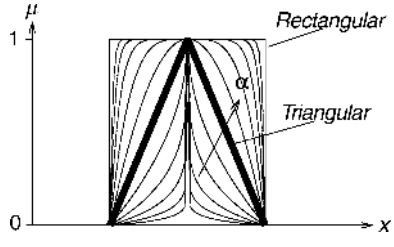

Figure 1.12 The parameterized triangular fuzzy set.

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

Deviating from alternating optimization 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, Klawonn, Kruse, and Runkler, 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 Equation (1.14)).

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 $\{(\mathrm{U}_1, C_1), (\mathrm{U}_2, C_2), \ldots\}$ that is terminated after a predefined number of iterations $t_{\max}$ or when the C_t have stabilized. Some instances of the ACE might be sensitive to the initialization of the cluster centers. Thus determining 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, Klawonn, Kruse, and Runkler, 1999).

1.6 CONCLUDING REMARKS

In this chapter we attempted to give a systematic overview of the fundamentals of fuzzy clustering, starting from the basic algorithms and underlining the difference between the probabilistic and possibilistic paradigms. We then described variants of the basic algorithms, adapted to specific constraints or expectations. We further pointed out major research directions associated with fuzzy clustering. The field is so broad that it is not possible to mention all of them. In this conclusion we briefly point out further research directions that we could not address in the main part of the chapter due to length constraints.

1.6.1 Clustering Evaluation

An important topic related to clustering is that of cluster evaluation, i.e., the assessment of the obtained clusters quality: clustering is an unsupervised learning task, which means data points are not associated with labels or targets that indicate the desired output. Thus no reference is provided to which the obtained results can be compared. Major cluster validity approaches include the evaluation of the trade off between cluster compactness and cluster separability (Dunn 1974a; Rezaee, Lehieveldt and Reiber, 1998; Xie and Beni, 1991) and stability based approaches (see e.g., Ben-Hur, Elisseeff, and Guyon (2002)).

Some criteria are specifically dedicated to fuzzy clustering: the partition entropy criterion for instance computes the entropy of the obtained membership degrees,

$$PE = -\sum_{i,j} u_{ij} \log u_{ij},$$

and must be minimized (Bezdek, 1975). Indeed, it takes into account that the fuzzy membership degrees are degrees of freedom that simplify the optimization of the objective function, but that the desired clustering output is still a crisp partition. A data partition that is too fuzzy rather indicates a bad adequacy between the cluster number and the considered data-set and it should be penalized. Other fuzzy clustering dedicated criteria can be found in Bezder (1974) or Windham (1981).

Such criteria can be used to evaluate quantitatively the clustering quality and to compare algorithms one with another. They can also be applied to compare the results obtained with a single algorithm, when

the parameter values are changed. In particular they can be used in order to select the optimal number of clusters: applying the algorithm for several c values, the value c^* leading to the optimal decomposition according to the considered criterion is selected.

1.6.2 Shape and Size Regularization

As presented in Section 1.3.1, some fuzzy clustering algorithms make it possible to identify clusters of ellipsoidal shapes and with various sizes. This flexibility implies that numerous cluster parameters are to be adjusted by the algorithms. The more parameters are involved the more sensitive the methods get to their initialization. Furthermore, the additional degrees of freedom lead to a lack of robustness.

Lately, a new approach has been proposed (Borgelt and Kruse, 2005) that relies on regularization to introduce shape and size constraints to handle the higher degrees of freedom effectively. With a time-dependent shape regularization parameter, this method makes it possible to perform a soft transition from the fuzzy C-means (spherical clusters) to the Gustafson–Kessel algorithm (general ellipsoidal clusters).

1.6.3 Co-clustering

Co-clustering, also called bi-clustering, two mode clustering, two way clustering or subspace clustering, has the specific aim of simultaneously identifying relevant subgroups in the data and relevant attributes for each subgroup: it aims at performing both clustering and local attribute selection. It is in particular applied in the bio-informatics domain, so as to detect groups of similar genes and simultaneously groups of experimental conditions that justify the gene grouping. Other applications include text mining, e.g., for the identification of both document clusters and their characteristic keywords (Kummamuru, Dhawale, and Krishnapuram, 2003). Many dedicated clustering algorithms have been proposed, including fuzzy clustering methods as for instance Frigui and Nasraoui (2000).

1.6.4 Relational Clustering

The methods described in this chapter apply to object data, i.e., consider the case where a description is provided for each data point individually. In other cases, this information is not available, the algorithm input takes the form of a pairwise dissimilarity matrix. The latter has size $n \times n$, each of its elements indicates the dissimilarity between point couples. Relational clustering aims at identifying clusters exploiting this input. There exists a large variety of fuzzy clustering techniques for such settings (Bezdek, Keller, Krishnapuram, and Pal, 1999; Hathaway and Bezdek, 1994) that are also based on objective function optimization or the ACE scheme (runkler and Bezdek, 2003). The interested reader is also referred to the respective chapter in Bezdek, Keller, Krishnapuram, and Pal (1999).

1.6.5 Semisupervised Clustering

Clustering is an unsupervised learning task. Yet it may be the case that the user has some a priori knowledge about couples of points that should belong to the same cluster. Semisupervised clustering is concerned with this learning framework, where some partial information is available : the clustering results must then verify additional constraints, implied by these pieces of information. Specific clustering algorithms have been proposed to handle these cases; the interested reader is referred to chapter 7 in this book.

ACKNOWLEDGEMENTS

Marie-Jeanne Lesot was supported by a Lavoisier grant from the French Ministère des Affaires Etrangères.

REFERENCES

Ball, G. and Hall, D. (1966) 'Isodata an iterative method of multivariate data analysis and pattern classification'. *IEEE Int. Comm. Conf. (Philadelphia, PA)*, vol. 2715 (2003) of *Lecture Notes in Artifical Intelligence*. IEEE Press, Piscataway, NJ, USA,

Barni, M. Cappellini, V. and Mecocci, A. (1996) 'Comments on a possibilistic approach to clustering'. *IEEE Transactions on Fuzzy Systems* **4**, 393–396.

Ben-Hur, A., Elisseeff, A. and Guyon, I. (2002) 'A stability based method for discovering structure in clustered data' In *Pacific Symposium on Biocomputing* (ed. Scientific W), vol. 7, pp. 6–17.

Bezdek, J. (1973) *Fuzzy Mathematics in Pattern Classification* PhD thesis Applied Math. Center, Cornell University, Ithaca, USA.

Bezdek, J. (1974) 'Cluster validity with fuzzy sets'. *Journal of Cybernetics* **3**(3), 58–73.

Bezdek, J. (1975) 'Mathematical models for systematics and taxonomy' *Proc. of the 8th Int. Conf. on Numerical Taxonomy*, pp. 143–166. Freeman.

Bezdek, J. (1981) *Pattern Recognition With Fuzzy Objective Function Algorithms*. Plenum Press, New York.

Bezdek, J. C., Keller, J., Krishnapuram, R. and Pal, NR. (1999) *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing* Kluwer Boston, London chapter 3. Cluster Analysis for Relational Data, pp. 137–182.

Blake, CL. and Merz, C. J. (1998) UCI repository of machine learning databases.

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

Borgelt, C. and Kruse, R. (2005) 'Fuzzy and probabilistic clustering with shape and size constraints' *Proc. 11th Int. Fuzzy Systems Association World Congress (IFSA'05, Beijing, China)*, pp. 945–950. Tsinghua University Press and Springer-Verlag, Beijing, China, and Heidelberg, Germany.

Davé, R. (1991) 'Characterization and detection of noise in clustering'. *Pattern Recognition Letters* **12**, 657–664.

Davé, R. and Krishnapuram, R. (1996) 'M-estimators and robust fuzzy clustering' In *Proc. of the Int. Conf. of the North Americam Fuzzy Information Processing Society, NAFIPS'96* (ed. Smith M, Lee M, Keller J and Yen J), pp. 400–404. IEEE.

Davé, R. and Krishnapuram, R. (1997) 'Robust clustering methods: a unified view'. *IEEE Transactions on Fuzzy Systems* **5**, 270–293.

Davé, R. and Sen, S. (1997) 'On generalising the noise clustering algorithms' *Proc. of the 7th IFSA World Congress, IFSA'97*, pp. 205–210.

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

Drineas, P., *et al*. (2004) 'Clustering large graphs via the singular value decomposition'. *Machine Learning* **56**, 933.

Dubois, D. and Prade, H. (1988) *Possibility Theory.* Plenum Press, New York, NY, USA.

Duda, R. and Hart, P. (1973) *Pattern Classification and Scene Analysis*. J. Wiley & Sons, Inc., New York, NY, USA.

Dunn, J. (1974a) 'Well separated clusters and optimal fuzzy partitions'. *Journal of Cybernetics* **4**, 95–104.

Dunn, J. C. (1974b) 'A fuzzy relative of the isodata process and its use in detecting compact, well separated clusters'. *Journal of Cybernetics* **3**, 95–104.

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

Frigui, H. and Krishnapuram, R. (1996) 'A robust algorithm for automatic extraction of an unknown number of clusters from noisy data'. *Pattern Recognition Letters* **17**, 1223–1232.

Frigui, H. and Krishnapuram, R. (1997) 'Clustering by competitive agglomeration'. *Pattern Recognition* **30**(7), 1109–1119.

Frigui, H. and Krishnapuram, R. (1999) 'A robust competitive clustering algorithm with applications in computer vision'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(5), 450–465.

Frigui, H. and Nasraoui, O. (2000) 'Simultaneous clustering and attribute discrimination' *Proc. of the 9th IEEE Int. Conf. on Fuzzy Systems, Fuzz-IEEE'00*, pp. 158–163.

Gustafson, E. E. and Kessel , W. C. (1979) 'Fuzzy clustering with a fuzzy covariance matrix' *Proc. of the IEEE Conference on Decision and Control, San Diego, Californien*, pp. 761–766. IEEE Press, Piscataway, NJ.

Halkidi, M., Batistakis, Y. and Vazirgiannis, M. (2002) 'Cluster validity methods: Part I and part II'. *SIGMOD Record* **31**(2), 19–27 and 40–45.

Hathaway, R. and Bezdek, J. (1994) 'Nerf c-means: Non-euclidean relational fuzzy clustering'. *Pattern Recognition* **27**(3), 429–437.

Hawkins, D. (1980) *Identification of Outliers*. Chapman and Hall, London.

Höppner, F., Klawonn, F., Kruse, R, and Runkler, T. (1999) *Fuzzy Cluster Analysis*. J. Wiley & Sons, Ltd, Chichester, United Kingdom.

Keller, A. (2000) 'Fuzzy clustering with outliers' In *Proc. of the 19th Int. Conf. of the North American Fuzzy Information Processing Society, NAFIPS'00* (ed. Whalen T), pp. 143–147.

Klawonn, F. (2004) 'Fuzzy clustering: Insights and a new approach'. *Mathware and soft computing* **11**, 125–142.

Klawonn, F. (2006) 'Understanding the membership degrees in fuzzy clustering'. In *Proc. of the 29th Annual Conference of the German Classification Society, GfKl 2005* (ed. Spiliopoulou M, Kruse R, Borgelt C, Nrnberger A and Gaul W), pp. 446–454 Studies in Classification, Data Analysis, and Knowledge Organization. Springer.

Klawonn, F. and Höppner, F. (2003a) 'An alternative approach to the fuzzifier in fuzzy clustering to obtain better clustering results' *Proceedings 3rd Eusflat*, pp. 730–734.

Klawonn, F. and Höppner, F. (2003b) 'What is fuzzy about fuzzy clustering? – understanding and improving the concept of the fuzzifier' In *Advances in Intelligent Data Analysis V* (ed. Berthold M, Lenz HJ, Bradley E, Kruse R and Borgelt C), pp. 254–264. Springer.

Klawonn, F., Kruse, R. and Timm, H. (1997) 'Fuzzy shell cluster analysis' In *Learning, networks and statistics* (ed. della Riccia, G., Lenz, H. and Kruse, R.) Springer pp. 105–120.

Krishnapuram, R. and Keller, J. (1993) 'A possibilistic approach to clustering'. *IEEE Transactions on Fuzzy Systems* **1**, 98–110.

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

Kummamuru, K., Dhawale, A. K. and Krishnapuram, R. (2003) 'Fuzzy co-clustering of documents and keywords' *Proc. of the IEEE Int. Conf. on Fuzzy Systems, Fuzz-IEEE'03*.

McKay, M. D , Beckman, R. J. and 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. and Bezdek, J. (1997) 'A mixed c-means clustering model' *Proc. of FUZZ'IEEE'97*, pp. 11–21.

Pal, N., Pal, K., Keller, J. and Bezdek, J. (2004) 'A new hybrid c-means clustering model' *Proc. of FUZZ IEEE'04*, pp. 179–184.

Pedrycz, W. (2005) *Knowledge-Based Clustering: From Data to Information Granules*. J. Wiley & Son Inc., Holboken, USA.

Rezaee, M., Lelieveldt, B. and Reiber, J. (1998) 'A new cluster validity index for the fuzzy C-means'. *Pattern Recognition Letters* **19**, 237–246.

Rousseeuw, P., Trauwaert, E. and Kaufman, L. (1995) 'Fuzzy clustering with high contrast'. *Journal of Computational and Applied Mathematics* **64**, 81–90.

Runkler, T. A. and Bezdek, J. C. (2003) 'Web mining with relational clustering'. *Int. Jo. Approx. Reasoning* **32**(2–3), 217–236.

Sahbi, H. and Boujemaa, N. (2005) 'Validity of fuzzy clustering using entropy regularization' *Proc. of the IEEE Int. Conf. on Fuzzy Systems*.

Schölkopf, B. and Smola, A. (2002) *Learning with Kernels*. MIT Press.

Timm, H. and Kruse, R. (2002) 'A modification to improve possibilistic fuzzy cluster analysis' *Proc. of FUZZ-IEEE'02*.

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

Vapnik, V. (1995) *The Nature of Statistical Learning Theory*. Springer, New York, USA.

Windham, M. P. (1981) 'Cluster validity for fuzzy clustering algorithm'. *Fuzzy Sets and Systems* **5**, 177–185.

Wu, K. and Yang, M. (2002) 'Alternating c-means clustering algorithms'. *Pattern Recognition* **35**, 2267–2278.

Wu, Z., Xie, W. and Yu, J. (2003) 'Fuzzy c-means clustering algorithm based on kernel method' *Proc. of ICCIMA'03*, pp. 1–6.

Xie, X. and Beni, G. (1991) 'A validity measure for fuzzy clustering'. *IEEE Transactions on pattern analysis and machine intelligence* **13**(4), 841–846.

Zadeh, L. A. (1965) 'Fuzzy sets'. *Information Control* **8**, 338–353.

Zhang, D. and Chen, S. (2003a) 'Clustering incomplete data using kernel-based fuzzy c-means algorithm'. *Neural Processing Letters* **18**(3), 155–162.

Zhang, D. and Chen, S. (2003b) 'Kernel-based fuzzy and possibilistic c-means' *Proc. of ICANN'03*, pp. 122–125.

Zhang, D. and Chen, S. (2004) 'A comment on 'alternative c-means clustering algorithms''. *Pattern Recognition* **37**(2), 179–174.