4 FUZZY CLUSTERING

Clustering techniques are mostly unsupervised methods that can be used to organize data into groups based on similarities among the individual data items. Most clustering algorithms do not rely on assumptions common to conventional statistical methods, such as the underlying statistical distribution of data, and therefore they are useful in situations where little prior knowledge exists. The potential of clustering algorithms to reveal the underlying structures in data can be exploited in a wide variety of applications, including classification, image processing, pattern recognition, modeling and identification.

This chapter presents an overview of fuzzy clustering algorithms based on the c-means functional. Readers interested in a deeper and more detailed treatment of fuzzy clustering may refer to the classical monographs by Duda and Hart (1973), Bezdek (1981) and Jain and Dubes (1988). A more recent overview of different clustering algorithms can be found in (Bezdek and Pal, 1992).

4.1 Basic Notions

The basic notions of data, clusters and cluster prototypes are established and a broad overview of different clustering approaches is given.

4.1.1 The Data Set

Clustering techniques can be applied to data that are quantitative (numerical), qualitative (categorical), or a mixture of both. In this chapter, the clustering of quantita-

tive data is considered. The data are typically observations of some physical process. Each observation consists of n measured variables, grouped into an n-dimensional column vector $\mathbf{z}_k = [z_{1k}, \ldots, z_{nk}]^T$, $\mathbf{z}_k \in \mathbb{R}^n$. A set of N observations is denoted by $\mathbf{Z} = \{\mathbf{z}_k \mid k = 1, 2, \ldots, N\}$, and is represented as an $n \times N$ matrix:

$$\mathbf{Z} = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1N} \\ z_{21} & z_{22} & \cdots & z_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ z_{n1} & z_{n2} & \cdots & z_{nN} \end{bmatrix}. \tag{4.1}$$

In the pattern-recognition terminology, the columns of this matrix are called *patterns* or objects, the rows are called the *features* or attributes, and $\mathbf{Z}$ is called the *pattern* or *data matrix*. The meaning of the columns and rows of $\mathbf{Z}$ depends on the context. In medical diagnosis, for instance, the columns of $\mathbf{Z}$ may represent patients, and the rows are then symptoms, or laboratory measurements for these patients. When clustering is applied to the modeling and identification of dynamic systems, the columns of $\mathbf{Z}$ may contain samples of time signals, and the rows are, for instance, physical variables observed in the system (position, pressure, temperature, etc.). In order to represent the system's dynamics, past values of these variables are typically included in $\mathbf{Z}$ as well.

4.1.2 Clusters and Prototypes

Various definitions of a cluster can be formulated, depending on the objective of clustering. Generally, one may accept the view that a cluster is a group of objects that are more similar to one another than to members of other clusters (Bezdek, 1981; Jain and Dubes, 1988). The term "similarity" should be understood as mathematical similarity, measured in some well-defined sense. In metric spaces, similarity is often defined by means of a *distance norm*. Distance can be measured among the data vectors themselves, or as a distance from a data vector to some *prototypical object* (prototype) of the cluster. The prototypes are usually not known beforehand, and are sought by the clustering algorithms simultaneously with the partitioning of the data. The prototypes may be vectors of the same dimension as the data objects, but they can also be defined as "higher-level" geometrical objects, such as linear or nonlinear subspaces or functions.

Data can reveal clusters of different geometrical shapes, sizes and densities as demonstrated in Figure 4.1. While clusters (a) are spherical, clusters (b) to (d) can be characterized as linear and nonlinear subspaces of the data space. The performance of most clustering algorithms is influenced not only by the geometrical shapes and densities of the individual clusters, but also by the spatial relations and distances among the clusters. Clusters can be well-separated, continuously connected to each other, or overlapping each other.

4.1.3 Overview of Clustering Methods

Many clustering algorithms have been introduced in the literature. Since clusters can formally be seen as subsets of the data set, one possible classification of clustering methods can be according to whether the subsets are *fuzzy* or *crisp* (hard).

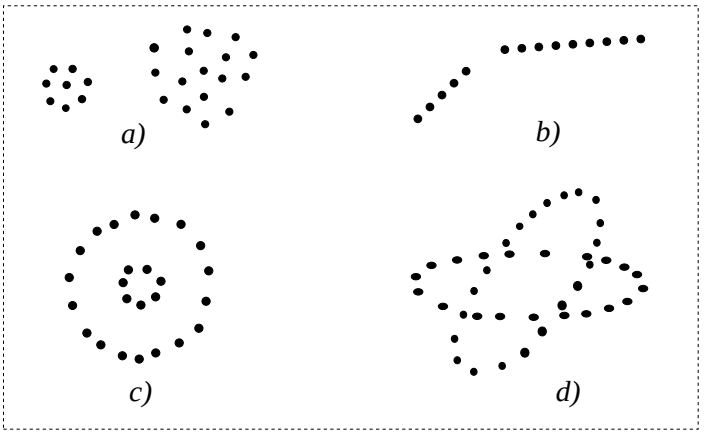

Figure 4.1. Clusters of different shapes and dimensions in $\mathbb{R}^2$. After (Jain and Dubes, 1988).

Hard clustering methods are based on classical set theory, and require that an object either does or does not belong to a cluster. Hard clustering means partitioning the data into a specified number of mutually exclusive subsets.

Fuzzy clustering methods, however, allow the objects to belong to several clusters simultaneously, with different degrees of membership. In many situations, fuzzy clustering is more natural than hard clustering. Objects on the boundaries between several classes are not forced to fully belong to one of the classes, but rather are assigned membership degrees between 0 and 1 indicating their partial membership. The discrete nature of the hard partitioning also causes difficulties with algorithms based on analytic functionals, since these functionals are not differentiable.

Another classification can be related to the algorithmic approach of the different techniques (Bezdek, 1981).

- *Agglomerative hierarchical methods* and *splitting hierarchical methods* form new clusters by reallocating memberships of one point at a time, based on some suitable measure of similarity.
- With *graph-theoretic methods*, $\mathbf{Z}$ is regarded as a set of nodes. Edge weights between pairs of nodes are based on a measure of similarity between these nodes.
- Clustering algorithms may use an *objective function* to measure the desirability of partitions. Nonlinear optimization algorithms are used to search for local optima of the objective function.

The remainder of this chapter focuses on fuzzy clustering with objective function. These methods are relatively well understood, and mathematical results are available concerning the convergence properties and cluster validity assessment.

4.2 Hard and Fuzzy Partitions

The concept of *fuzzy partition* is essential for cluster analysis, and consequently also for the identification techniques that are based on fuzzy clustering. Fuzzy and possi-

bilistic partitions can be seen as a generalization of *hard partition* which is formulated in terms of classical subsets.

4.2.1 Hard Partition

The objective of clustering is to partition the data set $\mathbf{Z}$ into c clusters (groups, classes). For the time being, assume that c is known, based on prior knowledge, for instance. Using classical sets, a *hard partition* of $\mathbf{Z}$ can be defined as a family of subsets $\{A_i \mid 1 \leq i \leq c\} \subset \mathcal{P}(\mathbf{Z})$[1] with the following properties (Bezdek, 1981):

$$\bigcup_{i=1}^{c} A_i = \mathbf{Z}, \tag{4.2a}$$

$$A_i \cap A_j = \emptyset, \quad 1 \leq i \neq j \leq c, \tag{4.2b}$$

$$\emptyset \subset A_i \subset \mathbf{Z}, \quad 1 \leq i \leq c\,. \tag{4.2c}$$

Equation (4.2a) means that the union subsets A_i contains all the data. The subsets must be disjoint, as stated by (4.2b), and none of them is empty nor contains all the data in $\mathbf{Z}$ (4.2c). In terms of *membership (characteristic) functions*, a partition can be conveniently represented by the *partition matrix* $\mathbf{U} = [\mu_{ik}]_{c\times N}$. The ith row of this matrix contains values of the membership function μ_i of the ith subset A_i of $\mathbf{Z}$. It follows from (4.2) that the elements of $\mathbf{U}$ must satisfy the following conditions:

$$\mu_{ik} \in \{0, 1\}, \quad 1 \leq i \leq c, \quad 1 \leq k \leq N, \tag{4.3a}$$

$$\sum_{i=1}^{c} \mu_{ik} = 1, \quad 1 \leq k \leq N, \tag{4.3b}$$

$$0 < \sum_{k=1}^{N} \mu_{ik} < N, \quad 1 \leq i \leq c\,. \tag{4.3c}$$

The space of all possible hard partition matrices for $\mathbf{Z}$, called the hard partitioning space (Bezdek, 1981), is thus defined by

$$M_{hc} = \left\{ \mathbf{U} \in \mathbb{R}^{c\times N} \middle| \mu_{ik} \in \{0, 1\}, \forall i, k;\ \sum_{i=1}^{c} \mu_{ik} = 1, \forall k;\ 0 < \sum_{k=1}^{N} \mu_{ik} < N, \forall i \right\}.$$

Example 4.1 Hard partition. Let us illustrate the concept of hard partition by a simple example. Consider a data set $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{10}\}$, shown in Figure 4.2.

A visual inspection of this data may suggest two well-separated clusters (data points $\mathbf{z}_1$ to $\mathbf{z}_4$ and $\mathbf{z}_7$ to $\mathbf{z}_{10}$ respectively), one point in between the two clusters ($\mathbf{z}_5$), and an "outlier" $\mathbf{z}_6$. One particular partition $\mathbf{U} \in M_{hc}$ of the data into two subsets (out of the

[1] $\mathcal{P}(Z)$ is the power set of Z.

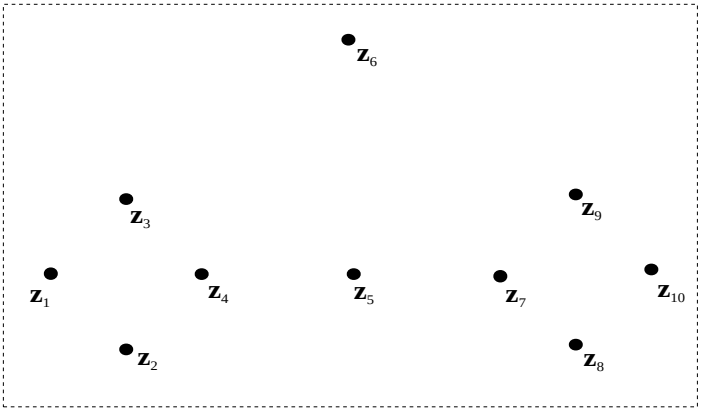

Figure 4.2. A data set in $\mathbb{R}^2$.

2^{10} possible hard partitions) is

$$\mathbf{U} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

The first row of $\mathbf{U}$ defines point-wise the characteristic function for the first subset of $\mathbf{Z}$, A_1, and the second row defines the characteristic function of the second subset of $\mathbf{Z}$, A_2. Each sample must be assigned exclusively to one subset (cluster) of the partition. In this case, both the boundary point $\mathbf{z}_5$ and the outlier $\mathbf{z}_6$ have been assigned to A_1. It is clear that a hard partitioning may not give a realistic picture of the underlying data. Boundary data points may represent patterns with a mixture of properties of data in A_1 and A_2, and therefore cannot be fully assigned to either of these classes, or do they constitute a separate class. This shortcoming can be alleviated by using fuzzy and possibilistic partitions as shown in the following sections.

□

4.2.2 Fuzzy Partition

Generalization of the hard partition to the fuzzy case follows directly by allowing μ_{ik} to attain real values in $[0, 1]$. Conditions for a fuzzy partition matrix, analogous to (4.3) are given by (Ruspini, 1970):

$$\mu_{ik} \in [0, 1], \quad 1 \leq i \leq c, \quad 1 \leq k \leq N, \tag{4.4a}$$

$$\sum_{i=1}^{c} \mu_{ik} = 1, \quad 1 \leq k \leq N, \tag{4.4b}$$

$$0 < \sum_{k=1}^{N} \mu_{ik} < N, \quad 1 \leq i \leq c\,. \tag{4.4c}$$

The ith row of the fuzzy partition matrix $\mathbf{U}$ contains values of the ith *membership function* of the fuzzy subset A_i of $\mathbf{Z}$. Equation (4.4b) constrains the sum of each column to 1, and thus the total membership of each $\mathbf{z}_k$ in $\mathbf{Z}$ equals one. The fuzzy

partitioning space for $\mathbf{Z}$ is the set

$$M_{fc} = \left\{ \mathbf{U} \in \mathbb{R}^{c \times N} \middle| \mu_{ik} \in [0,1], \forall i,k;\ \sum_{i=1}^{c} \mu_{ik} = 1, \forall k;\ 0 < \sum_{k=1}^{N} \mu_{ik} < N, \forall i \right\}.$$

Example 4.2 Fuzzy partition. Consider the data set from Example 4.1. One of the infinitely many fuzzy partitions in $\mathbf{Z}$ is:

$$\mathbf{U} = \begin{bmatrix} 1.0 & 1.0 & 1.0 & 0.8 & 0.5 & 0.5 & 0.2 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.2 & 0.5 & 0.5 & 0.8 & 1.0 & 1.0 & 1.0 \end{bmatrix}.$$

The boundary point $\mathbf{z}_5$ has now a membership degree of 0.5 in both classes, which correctly reflects its position in the middle between the two clusters. Note, however, that the outlier $\mathbf{z}_6$ has the same pair of membership degrees, even though it is further from the two clusters, and thus can be considered less typical of both A_1 and A_2 than $\mathbf{z}_5$. This is because condition (4.4b) requires that the sum of memberships of each point equals one. It can be, of course, argued that three clusters are more appropriate in this example than two. In general, however, it is difficult to detect outliers and assign them to extra clusters. The use of possibilistic partition, presented in the next section, overcomes this drawback of fuzzy partitions.

□

4.2.3 Possibilistic Partition

A more general form of fuzzy partition, the *possibilistic partition,*[2] can be obtained by relaxing the constraint (4.4b). This constraint, however, cannot be completely removed, in order to ensure that each point is assigned to at least one of the fuzzy subsets with a membership greater than zero. Equation (4.4b) can be replaced by a less restrictive constraint $\forall k,\ \exists i,\ \mu_{ik} > 0$. The conditions for a possibilistic fuzzy partition matrix are:

$$\mu_{ik} \in [0,1], \quad 1 \le i \le c, \quad 1 \le k \le N, \tag{4.5a}$$

$$\exists i,\ \mu_{ik} > 0, \quad \forall k, \tag{4.5b}$$

$$0 < \sum_{k=1}^{N} \mu_{ik} < N, \quad 1 \le i \le c\,. \tag{4.5c}$$

Analogously to the previous cases, the possibilistic partitioning space for $\mathbf{Z}$ is the set

$$M_{pc} = \left\{ \mathbf{U} \in \mathbb{R}^{c \times N} \middle| \mu_{ik} \in [0,1], \forall i,k;\ \forall k,\ \exists i,\ \mu_{ik} > 0;\ 0 < \sum_{k=1}^{N} \mu_{ik} < N, \forall i \right\}.$$

[2]The term "possibilistic" (partition, clustering, etc.) has been introduced in (Krishnapuram and Keller, 1993). In the literature, the terms "constrained fuzzy partition" and "unconstrained fuzzy partition" are also used to denote partitions (4.4) and (4.5), respectively.

Example 4.3 Possibilistic partition. An example of a possibilistic partition matrix for our data set is:

$$\mathbf{U} = \begin{bmatrix} 1.0 & 1.0 & 1.0 & 1.0 & 0.5 & 0.2 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.5 & 0.2 & 1.0 & 1.0 & 1.0 & 1.0 \end{bmatrix}.$$

As the sum of elements in each column of $\mathbf{U} \in M_{fc}$ is no longer constrained, the outlier has a membership of 0.2 in both clusters, which is lower than the membership of the boundary point $\mathbf{z}_5$, reflecting the fact that this point is less typical for the two clusters than $\mathbf{z}_5$.

□

4.3 Fuzzy c-Means Clustering

Most analytical fuzzy clustering algorithms (and also all the algorithms presented in this chapter) are based on optimization of the basic c-means objective function, or some modification of it. Hence we start our discussion with presenting the fuzzy c-means functional.

4.3.1 The Fuzzy c-Means Functional

A large family of fuzzy clustering algorithms is based on minimization of the *fuzzy c-means* functional formulated as (Dunn, 1974; Bezdek, 1981):

$$J(\mathbf{Z}; \mathbf{U}, \mathbf{V}) = \sum_{i=1}^{c} \sum_{k=1}^{N} (\mu_{ik})^m \|\mathbf{z}_k - \mathbf{v}_i\|_{\mathbf{A}}^2 \tag{4.6a}$$

where

$$\mathbf{U} = [\mu_{ik}] \in M_{fc} \tag{4.6b}$$

is a fuzzy partition matrix of $\mathbf{Z}$,

$$\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c], \quad \mathbf{v}_i \in \mathbb{R}^n \tag{4.6c}$$

is a vector of *cluster prototypes* (centers), which have to be determined,

$$D_{ik\mathbf{A}}^2 = \|\mathbf{z}_k - \mathbf{v}_i\|_{\mathbf{A}}^2 = (\mathbf{z}_k - \mathbf{v}_i)^T \mathbf{A} (\mathbf{z}_k - \mathbf{v}_i) \tag{4.6d}$$

is a squared inner-product distance norm, and

$$m \in [1, \infty) \tag{4.6e}$$

is a parameter which determines the fuzziness of the resulting clusters. The value of the cost function (4.6a) can be seen as a measure of the total variance of $\mathbf{z}_k$ from $\mathbf{v}_i$.

4.3.2 The Fuzzy c-Means Algorithm

The minimization of the c-means functional (4.6a) represents a nonlinear optimization problem that can be solved by using a variety of methods, including iterative minimization, simulated annealing or genetic algorithms. The most popular method is a simple Picard iteration through the first-order conditions for stationary points of (4.6a), known as the fuzzy c-means (FCM) algorithm.

The stationary points of the objective function (4.6a) can be found by adjoining the constraint (4.4b) to J by means of Lagrange multipliers:

$$\bar{J}(\mathbf{Z};\mathbf{U},\mathbf{V},\boldsymbol{\lambda}) = \sum_{i=1}^{c}\sum_{k=1}^{N}(\mu_{ik})^m D_{ik\mathbf{A}}^2 + \sum_{k=1}^{N}\lambda_k\left[\sum_{i=1}^{c}\mu_{ik} - 1\right], \tag{4.7}$$

and by setting the gradients of $\bar{J}$ with respect to $\mathbf{U}$, $\mathbf{V}$ and $\boldsymbol{\lambda}$ to zero. It can be shown that if $D_{ik\mathbf{A}}^2 > 0, \forall i,k$ and $m > 1$, then $(\mathbf{U},\mathbf{V}) \in M_{fc} \times \mathbb{R}^{n\times c}$ may minimize (4.6a) only if

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{c}(D_{ik\mathbf{A}}/D_{jk\mathbf{A}})^{2/(m-1)}}, \quad 1 \le i \le c, \quad 1 \le k \le N, \tag{4.8a}$$

and

$$\mathbf{v}_i = \frac{\sum_{k=1}^{N}(\mu_{ik})^m \mathbf{z}_k}{\sum_{k=1}^{N}(\mu_{ik})^m}; \quad 1 \le i \le c\,. \tag{4.8b}$$

This solution also satisfies the remaining constraints (4.4a) and (4.4c). Equations (4.8) are first-order necessary conditions for stationary points of the functional (4.6a). The FCM (Algorithm 4.1) iterates through (4.8a) and (4.8b). Sufficiency of (4.8) and the convergence of the FCM algorithm is proven in (Bezdek, 1980). Note that (4.8b) gives $\mathbf{v}_i$ as the weighted mean of the data items that belong to a cluster, where the weights are the membership degrees. That is why the algorithm is called "c-means".

Some remarks should be made:

1. The purpose of the "if ... otherwise" branch at Step 3 is to take care of a singularity that occurs in FCM when $D_{is\mathbf{A}} = 0$ for some $\mathbf{z}_k$ and one or more cluster prototypes $\mathbf{v}_s$, $s \in S \subset \{1,2,\ldots,c\}$. In this case, the membership degree in (4.8a) cannot be computed. When this happens, 0 is assigned to each μ_{ik}, $i \in \bar{S}$ and the membership is distributed arbitrarily among μ_{sj} subject to the constraint $\sum_{s\in S}\mu_{sj} = 1, \forall k$.

2. The FCM algorithm converges to a *local* minimum of the c-means functional (4.6a). Hence, different initializations may lead to different results.

3. While steps 1 and 2 are straightforward, step 3 is a bit more complicated, as a singularity in FCM occurs when $D_{ik\mathbf{A}} = 0$ for some $\mathbf{z}_k$ and one or more $\mathbf{v}_i$. When this happens (rare in practice), zero membership is assigned to the clusters

Algorithm 4.1 Fuzzy c-means (FCM).

Given the data set $\mathbf{Z}$, choose the number of clusters $1 < c < N$, the weighting exponent $m > 1$, the termination tolerance $\epsilon > 0$ and the norm-inducing matrix $\mathbf{A}$. Initialize the partition matrix randomly, such that $\mathbf{U}^{(0)} \in M_{fc}$.

Repeat for $l = 1, 2, \ldots$

Step 1: Compute the cluster prototypes (means):

$$\mathbf{v}_i^{(l)} = \frac{\sum_{k=1}^{N} \left(\mu_{ik}^{(l-1)}\right)^m \mathbf{z}_k}{\sum_{k=1}^{N} \left(\mu_{ik}^{(l-1)}\right)^m}, \quad 1 \le i \le c\,.$$

Step 2: Compute the distances:

$$D_{ik\mathbf{A}}^2 = (\mathbf{z}_k - \mathbf{v}_i^{(l)})^T \mathbf{A} (\mathbf{z}_k - \mathbf{v}_i^{(l)}), \quad 1 \le i \le c, \quad 1 \le k \le N\,.$$

Step 3: Update the partition matrix:
for $1 \le k \le N$
if $D_{ik\mathbf{A}} > 0$ for all $i = 1, 2, \ldots, c$

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^{c} (D_{ik\mathbf{A}}/D_{jk\mathbf{A}})^{2/(m-1)}},$$

otherwise

$$\mu_{ik}^{(l)} = 0 \text{ if } D_{ik\mathbf{A}} > 0, \text{ and } \mu_{ik}^{(l)} \in [0,1] \text{ with } \sum_{i=1}^{c} \mu_{ik}^{(l)} = 1\,.$$

until $\|\mathbf{U}^{(l)} - \mathbf{U}^{(l-1)}\| < \epsilon$.

---

for which $D_{ik\mathbf{A}} > 0$ and the memberships are distributed arbitrarily among the clusters for which $D_{ik\mathbf{A}} = 0$, such that the constraint in (4.4b) is satisfied.

4. The alternating optimization scheme used by FCM loops through the estimates $\mathbf{U}^{(l-1)} \to \mathbf{V}^{(l)} \to \mathbf{U}^{(l)}$ and terminates as soon as $\|\mathbf{U}^{(l)} - \mathbf{U}^{(l-1)}\| < \epsilon$. Alternatively, the algorithm can be initialized with $\mathbf{V}^{(0)}$, loop through $\mathbf{V}^{(l-1)} \to \mathbf{U}^{(l)} \to \mathbf{V}^{(l)}$, and terminate on $\|\mathbf{V}^{(l)} - \mathbf{V}^{(l-1)}\| < \epsilon$. The error norm in the termination criterion is usually chosen as $\max_{ik}(|\mu_{ik}^{(l)} - \mu_{ik}^{(l-1)}|)$. Different results

may be obtained with the same values of ϵ, since the termination criterion used in Algorithm 4.1 requires that more parameters become close to one another.

4.3.3 Parameters of the FCM Algorithm

Before using the FCM algorithm, the following parameters must be specified: the number of clusters, c, the 'fuzziness' exponent, m, the termination tolerance, ϵ, and the norm-inducing matrix, $\mathbf{A}$. Moreover, the fuzzy partition matrix, $\mathbf{U}$, must be initialized. The choices for these parameters are now described one by one.

Number of Clusters. The number of clusters c is the most important parameter, in the sense that the remaining parameters have less influence on the resulting partition. When clustering real data without any a priori information about the structures in the data, one usually has to make assumptions about the number of underlying clusters. The chosen clustering algorithm then searches for c clusters, regardless of whether they are really present in the data or not. Two main approaches to determining the appropriate number of clusters in data can be distinguished:

1. *Validity measures.* Validity measures are scalar indices that assess the goodness of the obtained partition. Clustering algorithms generally aim at locating well-separated and compact clusters. When the number of clusters is chosen equal to the number of groups that actually exist in the data, it can be expected that the clustering algorithm will identify them correctly. When this is not the case, misclassifications appear, and the clusters are not likely to be well separated and compact. Hence, most cluster validity measures are designed to quantify the separation and the compactness of the clusters. However, as Bezdek (1981) points out, the concept of cluster validity is open to interpretation and can be formulated in different ways. Consequently, many validity measures have been introduced in the literature, see (Bezdek, 1981; Gath and Geva, 1989; Pal and Bezdek, 1995) among others. For the FCM algorithm, the Xie-Beni index (Xie and Beni, 1991)

$$\chi(\mathbf{Z};\mathbf{U},\mathbf{V}) = \frac{\sum_{i=1}^{c}\sum_{k=1}^{N}\mu_{ik}^{m} \parallel \mathbf{z}_k - \mathbf{v}_i \parallel^2}{c \cdot \min_{i\neq j}\Big(\parallel \mathbf{v}_i - \mathbf{v}_j \parallel^2\Big)} \tag{4.9}$$

   has been found to perform well in practice. This index can be interpreted as the ratio of the total within-group variance and the separation of the cluster centers. The best partition minimizes the value of $\chi(\mathbf{Z};\mathbf{U},\mathbf{V})$.

2. *Iterative merging or insertion of clusters.* The basic idea of cluster merging is to start with a sufficiently large number of clusters, and successively reduce this number by merging clusters that are similar (compatible) with respect to some well-defined criteria (Krishnapuram and Freg, 1992; Kaymak and Babuška, 1995). One can also adopt an opposite approach, i.e., start with a small number of clusters and iteratively insert clusters in the regions where the data points have low degree of membership in the existing clusters (Gath and Geva, 1989).

Fuzziness Parameter. The weighting exponent m is a rather important parameter as well, because it significantly influences the fuzziness of the resulting partition. As m approaches one from above, the partition becomes hard ($\mu_{ik} \in \{0, 1\}$) and $\mathbf{v}_i$ are ordinary means of the clusters. As $m \to \infty$, the partition becomes completely fuzzy ($\mu_{ik} = 1/c$) and the cluster means are all equal to the mean of $\mathbf{Z}$. These limit properties of (4.6) are independent of the optimization method used (Pal and Bezdek, 1995). Usually, $m = 2$ is initially chosen.

Termination Criterion. The FCM algorithm stops iterating when the norm of the difference between $\mathbf{U}$ in two successive iterations is smaller than the termination parameter ϵ. For the maximum norm $\max_{ik}(|\mu_{ik}^{(l)} - \mu_{ik}^{(l-1)}|)$, the usual choice is $\epsilon = 0.001$, even though $\epsilon = 0.01$ works well in most cases, while drastically reducing the computing times.

Norm-Inducing Matrix. The shape of the clusters is determined by the choice of the matrix $\mathbf{A}$ in the distance measure (4.6d). A common choice is $\mathbf{A} = \mathbf{I}$, which gives the standard Euclidean norm:

$$D_{ik}^2 = (\mathbf{z}_k - \mathbf{v}_i)^T(\mathbf{z}_k - \mathbf{v}_i). \tag{4.10}$$

Another choice for $\mathbf{A}$ is a diagonal matrix that accounts for different variances in the directions of the coordinate axes of $\mathbf{Z}$:

$$\mathbf{A} = \begin{bmatrix} (1/\sigma_1)^2 & 0 & \cdots & 0 \\ 0 & (1/\sigma_2)^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & (1/\sigma_n)^2 \end{bmatrix}. \tag{4.11}$$

This matrix induces a diagonal norm on $\mathbb{R}^n$. Finally, $\mathbf{A}$ can be defined as the inverse of the covariance matrix of $\mathbf{Z}$: $\mathbf{A} = \mathbf{R}^{-1}$, with

$$\mathbf{R} = \frac{1}{N}\sum_{k=1}^{N}(\mathbf{z}_k - \bar{\mathbf{z}})(\mathbf{z}_k - \bar{\mathbf{z}})^T. \tag{4.12}$$

Here $\bar{\mathbf{z}}$ denotes the mean of the data. In this case, $\mathbf{A}$ induces the Mahalanobis norm on $\mathbb{R}^n$.

The norm influences the clustering criterion by changing the measure of dissimilarity. The Euclidean norm induces hyperspherical clusters (surfaces of constant membership are hyperspheres). Both the diagonal and the Mahalanobis norm generate hyperellipsoidal clusters. With the diagonal norm, the axes of the hyperellipsoids are parallel to the coordinate axes, while with the Mahalanobis norm the orientation of the hyperellipsoid is arbitrary, as shown in Figure 4.3.

A common limitation of clustering algorithms based on a fixed distance norm is that such a norm forces the objective function to prefer clusters of a certain shape even if they are not present in the data. This is demonstrated by the following example.

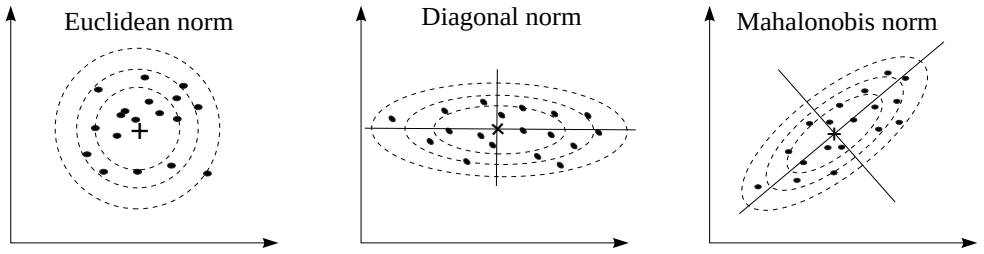

Figure 4.3. Different distance norms used in fuzzy clustering.

Example 4.4 Fuzzy c-means clustering. Consider a synthetic data set in $\mathbb{R}^2$, which contains two well-separated clusters of different shapes, as depicted in Figure 4.4. The samples in both clusters are drawn from the normal distribution. The standard deviation for the upper cluster is 0.2 for both axes, whereas in the lower cluster it is 0.2 for the horizontal axis and 0.05 for the vertical axis. The FCM algorithm was applied to this data set. The norm-inducing matrix was set to $\mathbf{A} = \mathbf{I}$ for both clusters, the weighting exponent to $m = 2$, and the termination criterion to $\epsilon = 0.01$. The algorithm was initialized with a random partition matrix and converged after 4 iterations. From the membership level curves in Figure 4.4, one can see that the FCM algorithm imposes a circular shape on both clusters, even though the lower cluster is rather elongated.

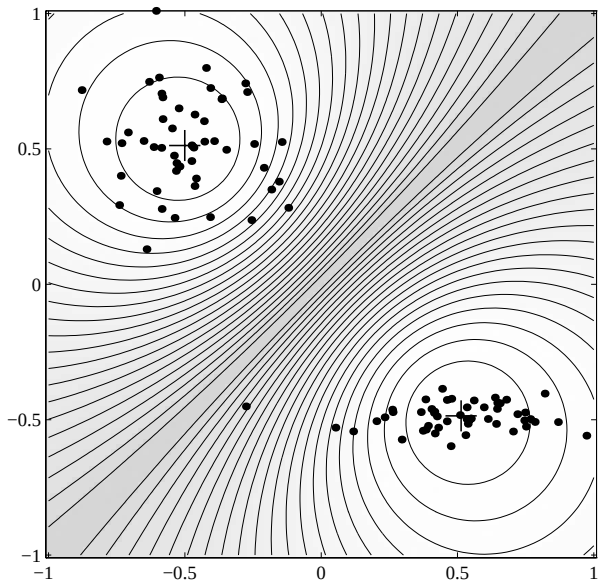

Figure 4.4. The fuzzy c-means algorithm imposes a spherical shape on the clusters, regardless of the actual data distribution. The dots represent the data points, '$+$' are the cluster means. Also shown are level curves of the clusters. Dark shading corresponds to membership degrees around 0.5.

Note that it is of no help to use another $\mathbf{A}$, since the two clusters have different shapes. Generally, different matrices $\mathbf{A}_i$ are required, but there is no guideline as

to how to choose them a priori. In Section 4.4, we will see that these matrices can be adapted by using estimates of the data covariance. A partition obtained with the Gustafson–Kessel algorithm, which uses such an adaptive distance norm, is presented in Example 4.5.

□

Initial Partition Matrix. The partition matrix is usually initialized at random, such that $\mathbf{U} \in M_{fc}$. A simple approach to obtain such $\mathbf{U}$ is to initialize the cluster centers $\mathbf{v}_i$ at random and compute the corresponding $\mathbf{U}$ by (4.8a) (i.e., by using the third step of the FCM algorithm).

4.3.4 Extensions of the Fuzzy $\mathbf{c}$*-Means Algorithm*

There are several well-known extensions of the basic c-means algorithm:

- Algorithms using an adaptive distance measure, such as the Gustafson–Kessel algorithm (Gustafson and Kessel, 1979) and the fuzzy maximum likelihood estimation algorithm (Gath and Geva, 1989).
- Algorithms based on hyperplanar or functional prototypes, or prototypes defined by functions. They include the fuzzy c-varieties (Bezdek, 1981), fuzzy c-elliptotypes (Bezdek, et al., 1981), and fuzzy regression models (Hathaway and Bezdek, 1993).
- Algorithms that search for possibilistic partitions in the data, i.e., partitions where the constraint (4.4b) is relaxed.

In the following sections we will focus on the Gustafson–Kessel algorithm.

4.4 Gustafson–Kessel Algorithm

Gustafson and Kessel (Gustafson and Kessel, 1979) extended the standard fuzzy c-means algorithm by employing an adaptive distance norm, in order to detect clusters of different geometrical shapes in one data set. Each cluster has its own norm-inducing matrix $\mathbf{A}_i$, which yields the following inner-product norm:

$$D^2_{ik\mathbf{A}_i} = (\mathbf{z}_k - \mathbf{v}_i)^T \mathbf{A}_i (\mathbf{z}_k - \mathbf{v}_i)\,. \tag{4.13}$$

The matrices $\mathbf{A}_i$ are used as optimization variables in the c-means functional, thus allowing each cluster to adapt the distance norm to the local topological structure of the data. The objective functional of the GK algorithm is defined by:

$$J(\mathbf{Z}; \mathbf{U}, \mathbf{V}, \{\mathbf{A}_i\}) = \sum_{i=1}^{c} \sum_{k=1}^{N} (\mu_{ik})^m D^2_{ik\mathbf{A}_i} \tag{4.14}$$

This objective function cannot be directly minimized with respect to $\mathbf{A}_i$, since it is linear in $\mathbf{A}_i$. To obtain a feasible solution, $\mathbf{A}_i$ must be constrained in some way. The usual way of accomplishing this is to constrain the determinant of $\mathbf{A}_i$:

$$|\mathbf{A}_i| = \rho_i, \quad \rho_i > 0, \quad \forall i\,. \tag{4.15}$$

Allowing the matrix $\mathbf{A}_i$ to vary with its determinant fixed corresponds to optimizing the cluster's shape while its volume remains constant. By using the Lagrange-multiplier method, the following expression for $\mathbf{A}_i$ is obtained (Gustafson and Kessel, 1979):

$$\mathbf{A}_i = [\rho_i \det(\mathbf{F}_i)]^{1/n} \mathbf{F}_i^{-1}, \tag{4.16}$$

where $\mathbf{F}_i$ is the *fuzzy covariance matrix* of the ith cluster given by

$$\mathbf{F}_i = \frac{\sum_{k=1}^{N} (\mu_{ik})^m (\mathbf{z}_k - \mathbf{v}_i)(\mathbf{z}_k - \mathbf{v}_i)^T}{\sum_{k=1}^{N} (\mu_{ik})^m}. \tag{4.17}$$

Note that the substitution of equations (4.16) and (4.17) into (4.13) gives a generalized squared Mahalanobis distance norm, where the covariance is weighted by the membership degrees in $\mathbf{U}$. The GK algorithm is given in Algorithm 4.2 and its MATLAB implementation can be found in the Appendix. The GK algorithm is computationally more involved than FCM, since the inverse and the determinant of the cluster covariance matrix must be calculated in each iteration.

Algorithm 4.2 Gustafson–Kessel (GK) algorithm.

Given the data set $\mathbf{Z}$, choose the number of clusters $1 < c < N$, the weighting exponent $m > 1$ and the termination tolerance $\epsilon > 0$ and the cluster volumes ρ_i. Initialize the partition matrix randomly, such that $\mathbf{U}^{(0)} \in M_{fc}$.

Repeat for $l = 1, 2, \ldots$

Step 1: Compute cluster prototypes (means):

$$\mathbf{v}_i^{(l)} = \frac{\sum_{k=1}^{N} \left(\mu_{ik}^{(l-1)}\right)^m \mathbf{z}_k}{\sum_{k=1}^{N} \left(\mu_{ik}^{(l-1)}\right)^m}, \quad 1 \le i \le c\,.$$

Step 2: Compute the cluster covariance matrices:

$$\mathbf{F}_i = \frac{\sum_{k=1}^{N} \left(\mu_{ik}^{(l-1)}\right)^m (\mathbf{z}_k - \mathbf{v}_i^{(l)})(\mathbf{z}_k - \mathbf{v}_i^{(l)})^T}{\sum_{k=1}^{N} \left(\mu_{ik}^{(l-1)}\right)^m}, \quad 1 \le i \le c\,.$$

Step 3: Compute the distances:

$$D^2_{ik\mathbf{A}_i} = (\mathbf{z}_k - \mathbf{v}_i^{(l)})^T \left[\rho_i \det(\mathbf{F}_i)^{1/n} \mathbf{F}_i^{-1}\right] (\mathbf{z}_k - \mathbf{v}_i^{(l)}),$$
$$1 \le i \le c, \quad 1 \le k \le N\,.$$

Step 4: Update the partition matrix:
for $1 \le k \le N$
if $D_{ik\mathbf{A}_i} > 0$ for all $i = 1, 2, \ldots, c$

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^{c} (D_{ik\mathbf{A}_i} / D_{jk\mathbf{A}_i})^{2/(m-1)}},$$

otherwise

$$\mu_{ik}^{(l)} = 0 \text{ if } D_{ik\mathbf{A}_i} > 0, \text{ and } \mu_{ik}^{(l)} \in [0,1] \text{ with } \sum_{i=1}^{c} \mu_{ik}^{(l)} = 1\,.$$

until $\|\mathbf{U}^{(l)} - \mathbf{U}^{(l-1)}\| < \epsilon$.

4.4.1 Parameters of the Gustafson–Kessel Algorithm

The same parameters must be specified as for the FCM algorithm (except for the norm-inducing matrix $\mathbf{A}$, which is adapted automatically): the number of clusters c, the 'fuzziness' exponent m, the termination tolerance ϵ. Additional parameters are the

cluster volumes ρ_i. Without any prior knowledge, ρ_i is simply fixed at 1 for each cluster. A drawback of the this setting is that due to the constraint (4.15), the GK algorithm only can find clusters of approximately equal volumes.

4.4.2 Interpretation of the Cluster Covariance Matrices

The eigenstructure of the cluster covariance matrix $\mathbf{F}_i$ provides information about the shape and orientation of the cluster. The ratio of the lengths of the cluster's hyperellipsoid axes is given by the ratio of the square roots of the eigenvalues of $\mathbf{F}_i$. The directions of the axes are given by the eigenvectors of $\mathbf{F}_i$, as shown in Figure 4.5. The GK algorithm can be used to detect clusters along linear subspaces of the data space. These clusters are represented by flat hyperellipsoids, which can be regarded as hyperplanes. The eigenvector corresponding to the smallest eigenvalue determines the normal to the hyperplane, and can be used to compute optimal local linear models from the covariance matrix.

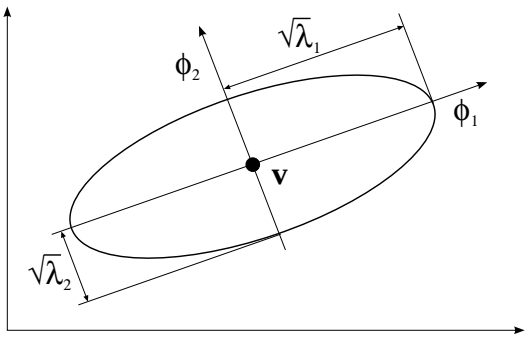

Figure 4.5. Equation $(\mathbf{z}-\mathbf{v})^T\mathbf{F}^{-1}(\mathbf{x}-\mathbf{v}) = 1$ defines a hyperellipsoid. The length of the jth axis of this hyperellipsoid is given by $\sqrt{\lambda}_j$ and its direction is spanned by $\boldsymbol{\phi}_j$, where λ_j and $\boldsymbol{\phi}_j$ are the jth eigenvalue and the corresponding eigenvector of $\mathbf{F}$, respectively.

Example 4.5 Gustafson–Kessel algorithm. The GK algorithm was applied to the data set from Example 4.4, using the same initial settings as the FCM algorithm. Figure 4.4 shows that the GK algorithm can adapt the distance norm to the underlying distribution of the data. One nearly circular cluster and one elongated ellipsoidal cluster are obtained. The shape of the clusters can be determined from the eigenstructure of the resulting covariance matrices $\mathbf{F}_i$. The eigenvalues of the clusters are given in Table 4.1.

One can see that the ratios given in the last column reflect quite accurately the ratio of the standard deviations in each data group (1 and 4 respectively). For the lower cluster, the unitary eigenvector corresponding to λ_2, $\boldsymbol{\phi}_2 = [0.0134, 0.9999]^T$, can be seen as a normal to a line representing the second cluster's direction, and it is, indeed, nearly parallel to the vertical axis.

□

Table 4.1. Eigenvalues of the cluster covariance matrices for clusters in Figure 4.6.

cluster	λ_1	λ_2	$\sqrt{\lambda}_1/\sqrt{\lambda}_2$
upper	0.0352	0.0310	1.0666
lower	0.0482	0.0028	4.1490

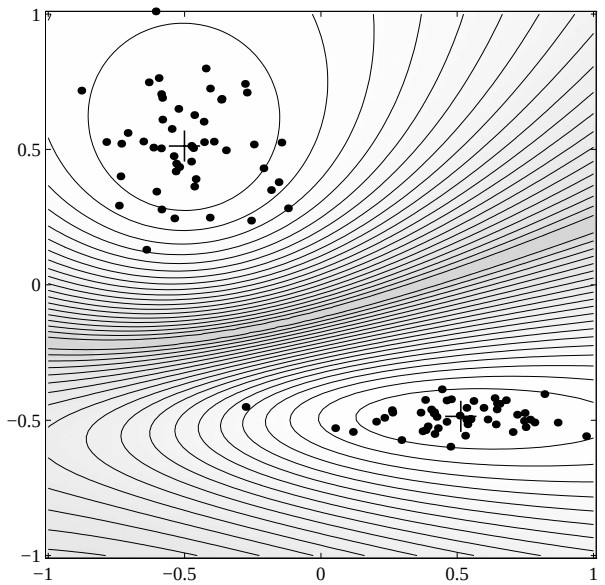

Figure 4.6. The Gustafson–Kessel algorithm can detect clusters of different shape and orientation. The points represent the data, '+' are the cluster means. Also shown are level curves of the clusters. Dark shading corresponds to membership degrees around 0.5.

4.5 Summary and Concluding Remarks

Fuzzy clustering is a powerful unsupervised method for the analysis of data and construction of models. In this chapter, an overview of the most frequently used fuzzy clustering algorithms has been given. It has been shown that the basic c-means iterative scheme can be used in combination with adaptive distance measures to reveal clusters of various shapes. The choice of the important user-defined parameters, such as the number of clusters and the fuzziness parameter, has been discussed.

4.6 Problems

1. State the definitions and discuss the differences of fuzzy and non-fuzzy (hard) partitions. Give an example of a fuzzy and non-fuzzy partition matrix. What are the advantages of fuzzy clustering over hard clustering?

2. State mathematically at least two different distance norms used in fuzzy clustering. Explain the differences between them.

3. Name two fuzzy clustering algorithms and explain how they differ from each other.

4. State the fuzzy c-mean functional and explain all symbols.

5. Outline the steps required in the initialization and execution of the fuzzy c-means algorithm. What is the role and the effect of the user-defined parameters in this algorithm?