3 FUZZY CLUSTERING ALGORITHMS

An effective approach to the identification of complex nonlinear systems is to partition the available data into subsets and approximate each subset by a simple model. Fuzzy clustering can be used as a tool to obtain a partitioning of data where the transitions between the subsets are gradual rather than abrupt. This chapter gives an introduction to the basic concepts of fuzzy clustering, and simultaneously serves as a reference to clustering algorithms that can be used to construct fuzzy models from data. The basic notions of clustering and the different types of partitions are defined in Sections 3.1 and 3.2. Section 3.3 presents the basic idea of fuzzy clustering with objective function and the fuzzy c-means algorithm. Sections 3.4 and 3.5 address algorithms which can detect clusters contained in linear subspaces of the data space. These methods include clustering with an adaptive distance measure, clustering with linear prototypes, and fuzzy regression clustering. Section 3.6 presents the approach known as possibilistic clustering. Section 3.7 addresses the determination of an appropriate number of clusters and Section 3.8 deals with pre-processing of the data. The aim of this chapter is to explain clustering at a level necessary to understand the subsequent chapters. For a more detailed treatment of the subject, the reader may refer to the classical monographs by Duda and Hart (1973), Bezdek (1981) and Jain and Dubes (1988). A more recent overview can be found in a collection of Bezdek and Pal (1992), and the monograph by Backer (1995). The notation and terminology in this chapter closely follows Bezdek (1981).

3.1 Cluster Analysis

The objective of cluster analysis is the classification of objects according to similarities among them, and the organizing of data into groups. Clustering techniques are among the *unsupervised* (learning) methods, since they do not use prior class identifiers. Most clustering algorithms also 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, not only for classification and pattern recognition, but also for the reduction of complexity in modeling and optimization.

3.1.1 The Data

Clustering techniques can be applied to data that is quantitative (numerical), qualitative (categoric), or a mixture of both. In this book, the clustering of quantitative 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 $\mathbf{Z} = \{\mathbf{z}_k | k = 1, 2, \ldots, N\}$, and is represented as an $n \times N$ matrix:

$$\mathbf{Z} = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1N} \\ z_{21} & z_{22} & \cdots & z_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ z_{n1} & z_{n2} & \cdots & z_{nN} \end{bmatrix}. \tag{3.1}$$

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 $\mathbf{Z}$ contain samples of time signals, and the rows are, for instance, physical variables observed in the system (position, velocity, temperature, etc.). In order to represent the system's dynamics, past values of the variables are typically included in $\mathbf{Z}$ as well. More details on the choice of an appropriate representation for dynamic systems are given in Section 4.2.

3.1.2 What Are Clusters?

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 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, see Section 3.5.

Data can reveal clusters of different geometrical shapes, sizes and densities as demonstrated in Figure 3.1. While clusters (a) are spherical, clusters (b) to (d) can be characterized as linear and nonlinear subspaces of the data space. Algorithms that can detect subspaces of the data space are of particular interest for identification and will be discussed in detail later on. 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. The separation of clusters is influenced by the scaling and normalization of the data, as discussed in Section 3.8.

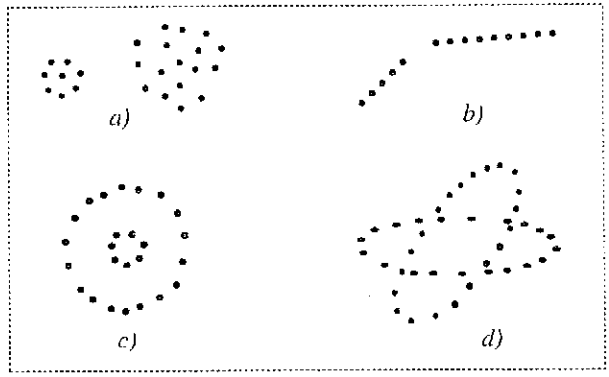


Figure 3.1. Clusters of different shapes and dimensions in $\mathbb{R}^2$. After Jain and Dubes (1988).

3.1.3 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). 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, as 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 memberships. The discrete nature of the hard partitioning

also causes analytical and algorithmic intractability of 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. The third class of clustering algorithms uses an *objective function* to measure the desirability of partitions. Nonlinear optimization algorithms are used to search for local extrema of the objective function.

The remainder of this chapter focuses on fuzzy clustering with objective function. These methods lead to least-squares optimization, and hence there are close relationships between clustering with fuzzy objective function and statistical regression and systems identification methods. In fuzzy clustering, the objective function is differentiable, which is a useful property for optimization. The objective function methods are also relatively well understood, and mathematical results are available concerning the convergence properties and cluster validity assessment.

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

3.2.1 Hard Partition

The objective of clustering is to partition the data set $\mathbf{Z}$ into c clusters. 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 $\{A_i | 1 \leq i \leq c\} \subset P(\mathbf{Z})$ with the following properties (Bezdek, 1981):

$$\bigcup_{i=1}^{c} A_i = \mathbf{Z}, \tag{3.2a}$$

$$A_i \cap A_j = \emptyset, \qquad 1 \leq i \neq j \leq c, \tag{3.2b}$$

$$\emptyset \subset A_i \subset \mathbf{Z}, \qquad 1 \leq i \leq c. \tag{3.2c}$$

Equation (3.2a) means that the subsets A_i collectively contain all the data in $\mathbf{Z}$. The subsets must be disjoint, as stated by (3.2b), and none of them is empty nor contains all the data in $\mathbf{Z}$ (3.2c). In terms of *membership functions*, equations (3.2) can be expressed as:

$$\bigvee_{i=1}^{c} \mu_{A_i} = \mathbf{1}, \tag{3.3a}$$

$$\mu_{A_i} \wedge \mu_{A_j} = \mathbf{0}, \qquad 1 \leq i \neq j \leq c, \tag{3.3b}$$

$$\mathbf{0} < \mu_{A_i} < \mathbf{1}, \qquad 1 \leq i \leq c. \tag{3.3c}$$

Here $\mathbf{0}$ and $\mathbf{1}$ denote zero and one function, respectively, and μ_{A_i} is the membership function of A_i. To simplify the notation, in this chapter we use μ_i instead of the usual μ_{A_i}. Further, by denoting $\mu_i(\mathbf{z}_k)$ by μ_{ik}, partitions can be conveniently represented in a matrix notation. A $c \times N$ matrix $\mathbf{U} = [\mu_{ik}]$ represents a hard partition if and only if its elements satisfy the conditions:

$$\mu_{ik} \in \{0, 1\}, \qquad 1 \leq i \leq c, \quad 1 \leq k \leq N, \tag{3.4a}$$

$$\sum_{i=1}^{c} \mu_{ik} = 1, \qquad 1 \leq k \leq N, \tag{3.4b}$$

$$0 < \sum_{k=1}^{N} \mu_{ik} < N, \qquad 1 \leq i \leq c, \tag{3.4c}$$

which directly follow from equations (3.3). The ith row of the hard partition matrix $\mathbf{U}$ contains values of the characteristic function of the ith subset A_i of $\mathbf{Z}$. The above discussion can be summarized in the following definition of hard partitioning space (Bezdek, 1981).

Definition 3.1 (Hard partitioning space) *Let $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N\}$ be a finite set and $2 \leq c < N$ be an integer. The hard partitioning space for $\mathbf{Z}$ is the set*

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

Example 3.1 Consider a data set $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{10}\}$, shown in Figure 3.2.

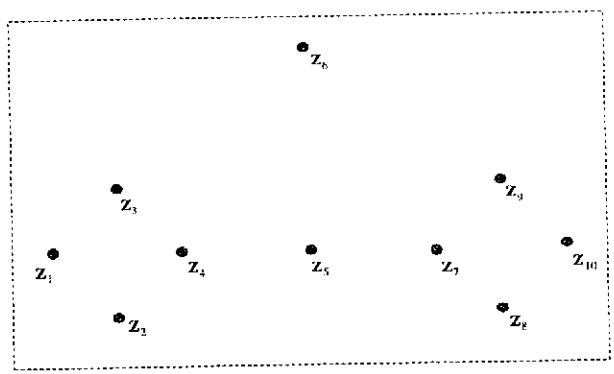


Figure 3.2. A data set in $\mathbb{R}^2$.

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" $\mathbf{z}_6$. A possible hard partition $\mathbf{U} \in M_{hc}$ of the data into two subsets is given by:

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

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. □

3.2.2 *Fuzzy Partition*

Generalization of the hard partition to the fuzzy case follows directly by allowing μ_{ik} to attain real values in $[0, 1]$ (Ruspini, 1970). Conditions for a fuzzy partition matrix, analogical to (3.4) then are given by:

$$\mu_{ik} \in [0, 1], \qquad 1 \leq i \leq c, \quad 1 \leq k \leq N, \tag{3.5a}$$

$$\sum_{i=1}^{c} \mu_{ik} = 1, \qquad 1 \leq k \leq N, \tag{3.5b}$$

$$0 < \sum_{k=1}^{N} \mu_{ik} < N, \qquad 1 \leq i \leq c. \tag{3.5c}$$

Definition 3.2 (Fuzzy partitioning space) *Let $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N\}$ be a finite set and let $2 \leq c < N$ be an integer. The fuzzy partitioning space for $\mathbf{Z}$ is the set*

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

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 (3.5b) constrains the sum of each column to 1, and thus the total membership of each $\mathbf{z}_k$ in $\mathbf{Z}$ equals one.

Example 3.2 Consider the data set from Example 3.1. 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 (3.5b) 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. □

3.2.3 *Possibilistic Partition*

A more general form of fuzzy partition, the *possibilistic partition*,[1] can be obtained by relaxing the constraint (3.5b). 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 (3.5b) can be replaced by a less restrictive constraint $\forall k,\ \exists i,\ \mu_{ik} > 0$. The conditions for a possibilistic fuzzy partition matrix, analogical to (3.5) are:

$$\mu_{ik} \in [0,1], \qquad 1 \le i \le c, \quad 1 \le k \le N, \tag{3.6a}$$

$$\exists i,\ \mu_{ik} > 0, \qquad \forall k, \tag{3.6b}$$

$$0 < \sum_{k=1}^{N} \mu_{ik} < N, \qquad 1 \le i \le c. \tag{3.6c}$$

Definition 3.3 (Possibilistic partitioning space) *Let* $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N\}$ *be a finite set and* $2 \le c < N$ *be an integer. The possibilistic partition space for* $\mathbf{Z}$ *is the set*

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

Example 3.3 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 a lower degree of typicality of this point for both sets. □

3.3 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 (Dunn, 1974a).

[1] The term "possibilistic" (partition, clustering, etc.) has been introduced by Krishnapuram and Keller (1993). In the literature, the terms "constrained fuzzy partition" and "unconstrained fuzzy partition" are also used to denote partitions (3.5) and (3.6), respectively.

3.3.1 The Fuzzy c-Means Functional

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

$$J(\mathbf{Z};\mathbf{U},\mathbf{V}) = \sum_{i=1}^{c}\sum_{k=1}^{N}(\mu_{ik})^m \|\mathbf{z}_k - \mathbf{v}_i\|_{\mathbf{A}}^2 \tag{3.7a}$$

where

$$\mathbf{U} = [\mu_{ik}] \in M_{fc} \tag{3.7b}$$

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

$$\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c], \quad \mathbf{v}_i \in \mathbb{R}^n \tag{3.7c}$$

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

$$D_{ik\mathbf{A}}^2 = \|\mathbf{z}_k - \mathbf{v}_i\|_{\mathbf{A}}^2 = (\mathbf{z}_k - \mathbf{v}_i)^T\mathbf{A}(\mathbf{z}_k - \mathbf{v}_i) \tag{3.7d}$$

is a squared inner-product distance norm, and

$$m \in [1, \infty) \tag{3.7e}$$

is a weighting exponent which determines the fuzziness of the resulting clusters. The measure of dissimilarity in (3.7a) is the squared distance between each data point $\mathbf{z}_k$ and the cluster prototype $\mathbf{v}_i$. This distance is weighted by the power of the membership degree of that point $(\mu_{ik})^m$. The value of the cost function (3.7a) can be seen as a measure of the total variance of $\mathbf{z}_k$ from $\mathbf{v}_i$.

3.3.2 The Fuzzy c-Means Algorithm

The minimization of the c-means functional (3.7a) represents a nonlinear optimization problem that can be solved by using a variety of available methods, ranging from grouped coordinate minimization (Bezdek, et al., 1987; Hathaway and Bezdek, 1991a), over simulated annealing (DeSarbo, 1982), to genetic algorithms (Babu and Murty, 1994). The most popular method, however, is a simple Picard iteration through the first-order conditions for stationary points of (3.7a), known as the fuzzy c-means (FCM) algorithm, which is given in Algorithm 3.1.

The stationary points of the objective function (3.7a) can be found by adjoining the constraint (3.5b) to J by means of Lagrange multipliers:

$$\bar{J}(\mathbf{Z};\mathbf{U},\mathbf{V},\lambda) = \sum_{i=1}^{c}\sum_{k=1}^{N}(\mu_{ik})^m D_{ik\mathbf{A}}^2 + \sum_{k=1}^{N}\lambda_k\left[\sum_{i=1}^{c}\mu_{ik} - 1\right], \tag{3.8}$$

and by setting the gradients of $\bar{J}$ with respect to $\mathbf{U}$, $\mathbf{V}$ and λ to zero. 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 (3.7a) only if

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{c}(D_{ik\mathbf{A}}/D_{jk\mathbf{A}})^{2/(m-1)}}, \quad 1 \le i \le c \quad 1 \le k \le N, \tag{3.9a}$$

and

$$\mathbf{v}_i = \frac{\sum_{k=1}^{N} (\mu_{ik})^m \mathbf{z}_k}{\sum_{k=1}^{N} (\mu_{ik})^m}; \quad 1 \le i \le c. \tag{3.9b}$$

This solution also satisfies the remaining constraints (3.5a) and (3.5c). Note that eq. (3.9b) 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". The FCM algorithm iterates through (3.9a) and (3.9b).

Algorithm 3.1 (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):**

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

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:**

if $D_{ik\mathbf{A}} > 0$ for $1 \le i \le c, \quad 1 \le k \le N$,

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

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

Remark 1. A singularity in FCM occurs at Step 3 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 (3.9a) 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 $\sum_{s \in S} \mu_{sj} = 1, \forall k$.

Remark 2. Equations (3.9) are only first-order necessary conditions for stationary points of the functional (3.7a). Sufficiency of these conditions and convergence of the algorithm has been proven by Bezdek (1980).

Remark 3. The alternating optimization scheme used by FCM loops through the estimates $\mathbf{U}^{(l-1)} \rightarrow V^{(l)} \rightarrow \mathbf{U}^{(l)}$ and terminates as soon as $\|\mathbf{U}^{(l)} - \mathbf{U}^{(l-1)}\| < \epsilon$. Alternatively, the algorithm can be initialized with $V^{(0)}$, loop through $V^{(l-1)} \rightarrow \mathbf{U}^{(l)} \rightarrow V^{(l)}$, and terminate on $\|V^{(l)} - V^{(l-1)}\| < \epsilon$. 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 3.1 requires that more parameters become close to one another. The usual setting of the termination criterion is $\epsilon = 0.001$, even though $\epsilon = 0.01$ works well in most cases.

Remark 4. The weighting exponent m is a rather important parameter, as it significantly influences 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 $m \rightarrow \infty$, the partition becomes maximally fuzzy ($\mu_{ik} = 1/c$) and the cluster means are all equal to the grand mean of $\mathbf{Z}$. These limit properties of (3.7) are independent of the optimization method used (Pal and Bezdek, 1995). Usually, $m = 2$ is chosen.

Remark 5. The number of clusters c is the most important parameter, in the sense that the remaining parameters have secondary effects on $\mathbf{U}$, compared to the effects of the number of clusters. The choice of the number of clusters is discussed in Sections 3.7 and 4.5.

3.3.3 *Inner-product Norms*

The shape of the clusters is determined by the choice of the matrix $\mathbf{A}$ in the distance measure (3.7d). A common choice is $\mathbf{A} = \mathbf{I}$, which induces the standard Euclidean norm:

$$D_{ik}^2 = (\mathbf{z}_k - \mathbf{v}_i)^T (\mathbf{z}_k - \mathbf{v}_i). \tag{3.10}$$

$\mathbf{A}$ can be chosen as an $n \times n$ diagonal matrix that accounts for different variances in the directions of the coordinate axes of $\mathbf{Z}$:

$$\mathbf{A}_D = \begin{bmatrix} (1/\sigma_1)^2 & 0 & \cdots & 0 \\ 0 & (1/\sigma_2)^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & (1/\sigma_n)^2 \end{bmatrix}. \tag{3.11}$$

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

$$\mathbf{R} = \frac{1}{N} \sum_{k=1}^{N} (\mathbf{z}_k - \bar{\mathbf{z}})(\mathbf{z}_k - \bar{\mathbf{z}})^T. \tag{3.12}$$

Here $\bar{\mathbf{z}}$ denotes the sample mean of the data. In this case, $\mathbf{A}$ induces the Mahalanobis norm on $\mathbb{R}^n$ (Bezdek, 1981).

The norm metric influences the clustering criterion by changing the measure of dissimilarity. The Euclidean norm induces hyperspherical clusters, i.e., clusters whose surfaces of constant membership are hyperspheres. Both the diagonal and the Mahalanobis norm generate hyperellipsoidal clusters, the difference is that 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 3.3. A common limitation of clustering algorithms based on a fixed distance norm is that such a norm induces a fixed topological structure on $\mathbb{R}^n$ and forces the objective function to prefer clusters of that shape even if they are not present (see Example 3.4). In Section 3.4, we will see that the norm-inducing matrix $\mathbf{A}$ can be adapted by using estimates of the data covariance, and can be used to estimate the dependence of the data in each cluster.

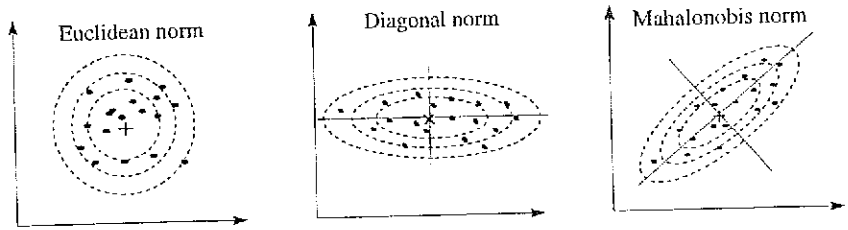


Figure 3.3. Different distance norms used in fuzzy clustering.

Example 3.4 *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 3.4. 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. From the membership level curves, one can see that the FCM algorithm strictly imposes a circular shape on both clusters, even though the lower cluster is rather elongated. The norm-inducing matrix was set to $\mathbf{A} = \mathbf{I}$ for both clusters, the weighting exponent was $m = 2$, and the termination criterion $\epsilon = 0.01$. The algorithm was initialized with a random partition matrix and converged after 4 iterations.

Note that it is of no help to use another $\mathbf{A}$, since the clusters may differ both in shape and orientation. Generally, different matrices $\mathbf{A}_i$ are required for the different clusters, but there is no guideline as to how to choose them a priori. Section 3.4.1 presents the partition obtained with the Gustafson–Kessel algorithm based on an adaptive distance norm. □

The following sections present several extensions of the basic c-means algorithm. A common feature of these algorithms is that they can detect clusters which lie in subspaces of the data space. The methods can be broadly classified into two groups:

- Algorithms using an adaptive distance measure, such as the Gustafson–Kessel algorithm (Gustafson and Kessel, 1979), or the fuzzy maximum likelihood estimation algorithm (Gath and Geva, 1989). These algorithms are presented in Section 3.4.

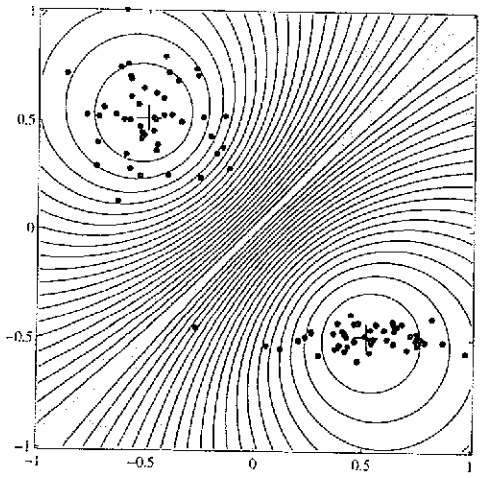


Figure 3.4. The fuzzy c-means algorithm imposes a spherical shape on the clusters, regardless of the actual data distribution. The dots represent the data points, '+' are the cluster means. Also shown are level curves of the clusters. Dark shading corresponds to membership degrees around 0.5.

- 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., 1981a), and fuzzy regression models (Hathaway and Bezdek, 1993b). These methods are presented in Section 3.5.

In addition, Section 3.6 presents a class of possibilistic clustering algorithms, which search for possibilistic partitions in the data, i.e., partitions where the constraint (3.5b) is relaxed.

3.4 Clustering with Fuzzy Covariance Matrix

A family of algorithms can be derived from the basic FCM scheme by adapting the inner-product norm (3.7d). Two of them are presented in this section: the Gustafson–Kessel algorithm and the algorithm based on fuzzy maximum likelihood estimates.

3.4.1 Gustafson–Kessel Algorithm

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:

$$D^2_{ik\mathbf{A}_i} = (\mathbf{z}_k - \mathbf{v}_i)^T \mathbf{A}_i (\mathbf{z}_k - \mathbf{v}_i)\,. \tag{3.13}$$

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. Let $\mathbf{A}$ denote a c-tuple of the norm-inducing matrices: $\mathbf{A} = (\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_c)$. The objective functional of the GK algorithm is defined by:

$$J(\mathbf{Z}; \mathbf{U}, \mathbf{V}, \mathbf{A}) = \sum_{i=1}^{c} \sum_{k=1}^{N} (\mu_{ik})^m D^2_{ik\mathbf{A}_i} \tag{3.14}$$

where $\mathbf{U} \in M_{fc}$, $\mathbf{V} \in \mathbb{R}^{n \times c}$ and $m > 1$. The solutions,

$$(\mathbf{U}, \mathbf{V}, \mathbf{A}) = \underset{M_{fc} \times \mathbb{R}^{n \times c} \times \mathrm{PD}^n}{\arg\min} J(\mathbf{Z}; \mathbf{U}, \mathbf{V}, \mathbf{A}), \tag{3.15}$$

are stationary points of J, where PD^n denotes a space of $n \times n$ positive definite matrices. For a fixed $\mathbf{A}$, conditions (3.9) can be directly applied. However, the objective function (3.14) cannot be directly minimized with respect to $\mathbf{A}_i$, since it is linear in $\mathbf{A}_i$. J could be made as small as desired by making $\mathbf{A}_i$ less positive definite. 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 $\mathbf{A}_i$. Allowing the matrix $\mathbf{A}_i$ to vary with its determinant fixed corresponds to optimizing the cluster's shape while its volume remains constant:

$$|\mathbf{A}_i| = \rho_i, \quad \rho_i > 0, \quad \forall i\,. \tag{3.16}$$

Using the Lagrange multiplier method, the following expression for $\mathbf{A}_i$ is obtained:

$$\mathbf{A}_i = [\rho_i \det(\mathbf{F}_i)]^{1/n} \mathbf{F}_i^{-1}, \tag{3.17}$$

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

$$\mathbf{F}_i = \frac{\sum_{k=1}^{N} (\mu_{ik})^m (\mathbf{z}_k - \mathbf{v}_i)(\mathbf{z}_k - \mathbf{v}_i)^T}{\sum_{k=1}^{N} (\mu_{ik})^m}. \tag{3.18}$$

Note that the substitution of equations (3.17) and (3.18) into (3.13) gives a generalized squared Mahalanobis distance norm between $\mathbf{z}_k$ and the cluster mean $\mathbf{v}_i$, where the covariance is weighted by the membership degrees in $\mathbf{U}$. The GK algorithm is given in Algorithm 3.2.

Remark 1. The same applies to the choice of m as in the case of the FCM algorithm, see Section 3.3.2.

Remark 2. Without any prior knowledge, the cluster volumes ρ_i are simply fixed at 1 for each cluster. A drawback of the GK algorithm is that due to the constraint (3.16), it only can find clusters of approximately equal volumes. Repetitive application of GK clustering with varying volumes is mentioned in Section 4.5.2 in combination with a compatible cluster merging procedure.

Remark 3. The eigenstructure of the cluster covariance matrix 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 3.5. Linear subspaces of the data space are represented by flat hyperellipsoids, which can

Algorithm 3.2 (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 $\epsilon > 0$. 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):**

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

Step 2: **Compute the cluster covariance matrices:**

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

Step 3: **Compute the distances:**

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

Step 4: **Update the partition matrix:**

if $D_{ik\mathbf{A}_i} > 0$ for $1 \leq i \leq c, \quad 1 \leq k \leq N$,

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

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

be seen 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, as shown in Lemma 5.1.

Remark 4. An advantage of the GK algorithm over FCM is that GK can detect clusters of different shape and orientation in one data set, as demonstrated in Ex-

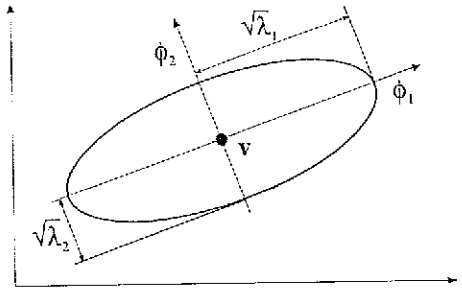


Figure 3.5. Equation $(\mathbf{z}-\mathbf{v})^T\mathbf{F}^{-1}(\mathbf{x}-\mathbf{v}) = 1$ defines a hyperellipsoid. The length of the jth axis of this hyperellipsoid is given by $\sqrt{\lambda_j}$ and its direction is spanned by ϕ_j, where λ_j and ϕ_j are the jth eigenvalue and the corresponding eigenvector of $\mathbf{F}$, respectively.

ample 3.5. It is, however, computationally more involved than FCM, since the inverse and determinant of the cluster covariance matrix must be calculated in each iteration.

Example 3.5 The GK algorithm was applied to the data set from Example 3.4, using the same initial settings as the FCM algorithm. Figure 3.4 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:

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

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, $\phi_2 = [0.0134, 0.9999]^T$, 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. □

3.4.2 *Fuzzy Maximum Likelihood Estimates Clustering*

The fuzzy maximum likelihood estimates (FMLE) clustering algorithm employs a distance norm based on the fuzzy maximum likelihood estimates, proposed by Bezdek and Dunn (1975):

$$D_{ik\Sigma_i} = \frac{[\det \Sigma_i]^{1/2}}{P_i} \exp\left[\frac{1}{2}(\mathbf{z}_k - \mathbf{v}_i)^T \Sigma_i^{-1}(\mathbf{z}_k - \mathbf{v}_i)\right]. \tag{3.19}$$

Note that, contrary to the GK algorithm, this distance norm involves an exponential term and thus decreases faster than the inner-product norm. Σ_i denotes the fuzzy

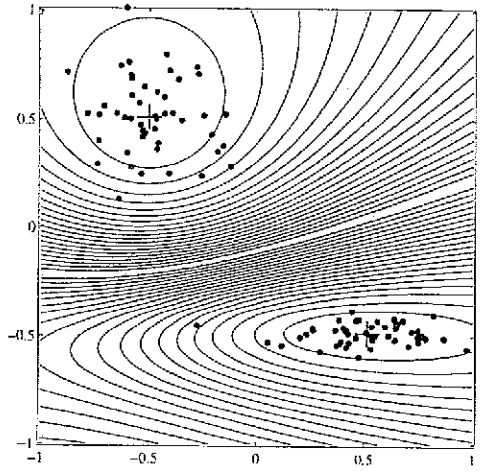


Figure 3.6. The Gustafson–Kessel algorithm can detect clusters of different shape and orientation. The points represent the data, '+' are the cluster means. Also shown are level curves of the clusters. Dark shading corresponds to membership degrees around 0.5.

covariance matrix of the ith cluster, given by:

$$\Sigma_i = \frac{\sum_{k=1}^{N} \mu_{ik}(\mathbf{z}_k - \mathbf{v}_i)(\mathbf{z}_k - \mathbf{v}_i)^T}{\sum_{k=1}^{N} \mu_{ik}} . \tag{3.20}$$

The difference between the matrix $\mathbf{F}_i$ in (3.18) and the Σ_i defined above is that the latter does not involve the weighting exponent m. This is because the two weighted covariance matrices arise as generalizations of the classical covariance from two different concepts. P_i is the prior probability of selecting cluster i, given by:

$$P_i = \frac{1}{N} \sum_{k=1}^{N} \mu_{ik} . \tag{3.21}$$

The membership degrees μ_{ik} are interpreted as the posterior probabilities, $\mu_{ik} \approx h(i|\mathbf{z}_k)$, of selecting the ith cluster given the data point $\mathbf{z}_k$. The iterative scheme of the FMLE algorithm (3.3) is very similar to that of the GK algorithm. Gath and Geva (1989) reported that the FMLE algorithm is able to detect clusters of varying shapes, sizes and densities. This is because the cluster covariance matrix is used in conjunction with an "exponential" distance, and the clusters are not constrained in volume. However, FMLE needs a good initialization, as due to the exponential distance norm, it tends to converge to a near local optimum.

3.5 Clustering with Linear Prototypes

In the algorithms described so far, the clusters are represented by their *prototypical points* (centers), $\mathbf{v}_i \in \mathbb{R}^n$, i.e., geometrical structures of the same "type" as the data.

Algorithm 3.3 (Fuzzy maximum likelihood estimate clustering)

Given the data set $\mathbf{Z}$ and a good initial partition matrix $\mathbf{U}^{(0)} \in M_{fc}$, choose the termination tolerance $\epsilon > 0$.

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

Step 1: **Compute cluster prototypes (means):**

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

Step 2: **Compute cluster covariance matrices and prior probabilities:**

$$\Sigma_i = \frac{\sum_{k=1}^{N} \mu_{ik}^{(l-1)} (\mathbf{z}_k - \mathbf{v}_i^{(l)})(\mathbf{z}_k - \mathbf{v}_i^{(l)})^T}{\sum_{k=1}^{N} \mu_{ik}^{(l-1)}}, \quad 1 \leq i \leq c,$$

$$P_i = \frac{1}{N} \sum_{k=1}^{N} \mu_{ik}^{(l-1)}, \quad 1 \leq i \leq c.$$

Step 3: **Compute the distances:**

$$D_{ik\Sigma_i} = \frac{[\det(\Sigma_i)]^{1/2}}{P_i} \exp[\frac{1}{2}(\mathbf{z}_k - \mathbf{v}_i^{(l)})^T \Sigma_i^{-1} (\mathbf{z}_k - \mathbf{v}_i^{(l)})],$$
$$1 \leq i \leq c, \; 1 \leq k \leq N.$$

Step 4: **Update the partition matrix:**

if $D_{ik\Sigma_i} > 0$ for $1 \leq i \leq c, \quad 1 \leq k \leq N$,

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

otherwise

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

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

The FCM algorithm uses a fixed distance norm, and thus strongly prefers clusters of a geometrical shape induced by that norm. The GK and FMLE algorithms remedy this drawback by locally adapting the distance norm. A conceptually different approach is to define the prototypes as r-dimensional linear or nonlinear subspaces of the data space, where $0 \leq r \leq n-1$. Algorithms based on this approach are reviewed in the following sections.

3.5.1 Fuzzy c-Varieties

The main idea of the fuzzy c-varieties (FCV) algorithm (Bezdek, et al., 1981a) is to measure the distances of data from r-dimensional linear varieties, i.e., lines ($r = 1$), planes ($r = 2$) or hyperplanes ($2 < r < n$). This family of algorithms can detect clusters lying in r-dimensional linear subspaces of $\mathbb{R}^n$. The corresponding objective functional is given by:

$$J_V(\mathbf{Z}; \mathbf{U}, V_r) = \sum_{i=1}^{c} \sum_{k=1}^{N} (\mu_{ik})^m D_{rik}^2, \tag{3.22}$$

where V_r is a set of c linear r-dimensional varieties $V_r = (v_{r1}, v_{r2}, \ldots, v_{rc})$, and D_{rik}^2 is a squared orthogonal distance from $\mathbf{z}_k$ to the linear variety v_{ri}:

$$D_{rik}^2 = \|\mathbf{z}_k - \mathbf{v}_i\|^2 - \sum_{j=1}^{r} \langle \mathbf{z}_k - \mathbf{v}_i, \mathbf{s}_{ij} \rangle^2 . \tag{3.23}$$

$(\mathbf{s}_{i1}, \mathbf{s}_{i2}, \ldots, \mathbf{s}_{ir})$ is an r-tuple of linearly independent vectors spanning the variety v_{ir}, $\mathbf{v}_i$ is a point through which the variety passes, and $\langle \cdot, \cdot \rangle$ denotes the scalar product. The functional $J_V(\mathbf{Z}; \mathbf{U}, V_r)$ attains its local minimum with respect to $\mathbf{U}$ if and only if the conditions (3.9) hold and $\mathbf{s}_{ij}$ is the unit eigenvector of the cluster covariance matrix (3.18) corresponding to the jth largest eigenvalue (Bezdek, et al., 1981a). By substituting the distance measure (3.23) into eq. (3.9a), the FCV algorithm (3.4) follows as a straightforward generalization of FCM. A major drawback of the FCV algorithm is that the linear variety is not limited in size, and thus the algorithm tends to connect collinear clusters that may be well separated. Moreover, FCV also does not partition correctly when the sizes of the varieties vary from cluster to cluster. In such case, the algorithm usually gets stuck in poor local minima (Dave, 1992).

3.5.2 Fuzzy c-Elliptotypes

The fuzzy c-elliptotypes (FCE) algorithm (Bezdek, et al., 1981b) attempts to alleviate some of the drawbacks of the fuzzy c-varieties algorithm by forcing each cluster to have a center of gravity $\mathbf{v}_i$, and by measuring the distance as a convex combination of the FCM and FCV distances:

$$D_{eik} = \alpha D_{ik} + (1 - \alpha) D_{rik}, \tag{3.24}$$

Algorithm 3.4 (Fuzzy c-varieties)

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

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

Step 1: **Compute cluster centers (means):**

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

Step 2: **Compute cluster covariance matrices:**

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

Step 3: **Extract principal eigenvectors.** Extract from each $\mathbf{F}_i^{(l)}$ its r principal eigenvectors $\mathbf{s}_{ij}^{(l)}$, $j = 1, 2, \ldots, r$ (eigenvectors corresponding to the r largest eigenvalues).

Step 4: **Compute the distances:**

$$D_{rik}^2 = \|\mathbf{z}_k - \mathbf{v}_i^{(l)}\|^2 - \sum_{j=1}^{r} (\langle \mathbf{z}_k - \mathbf{v}_i^{(l)}, \mathbf{s}_j^{(l)} \rangle)^2,$$
$$1 \leq i \leq c, \quad 1 \leq k \leq N.$$

Step 5: **Update the partition matrix:**

if $D_{rik} > 0$ for $1 \leq i \leq c, \quad 1 \leq k \leq N$,

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

otherwise

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

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

where $\alpha \in [0,1]$, D_{ik} is the Euclidean distance of $\mathbf{z}_k$ from $\mathbf{v}_i$ (3.10), and D_{rik} is given by eq. (3.23). Substituting equations (3.10) and (3.23) into (3.24) yields:

$$D_{eik} = (\mathbf{z}_k - \mathbf{v}_i)^T(\mathbf{z}_k - \mathbf{v}_i) - \alpha \sum_{j=1}^{r} (\mathbf{z}_k - \mathbf{v}_i, \mathbf{s}_{ij})^2 . \tag{3.25}$$

The first term in this expression measures the distance from the cluster center, and thus restricts the cluster size. The geometric interpretation of eq. (3.25) is that the level surfaces of the convex combinations D_{eik} are hyperellipsoids obtained by stretching a hypersphere defined by the Euclidean norm in the directions of vectors $\mathbf{s}_{ij}$. The FCE algorithm is identical to FCV, with the exception that (3.25) is used to compute the distances and that the mixing coefficient α must be defined.

As shown in Section 4.4.2, the FCE algorithm does not completely correct the problems of FCV. Moreover the value for α must be chosen carefully. If α is common to all clusters, the algorithm will seek clusters of the same elliptical shape. Techniques have also been proposed to adaptively select the mixing coefficient α for each cluster (Gunderson, 1983).

3.5.3 Fuzzy c-Regression Models

The last fuzzy clustering algorithm presented in this chapter is the fuzzy c-regression models (FCRM) algorithm proposed by Hathaway and Bezdek (1993b). This algorithm estimates parameters of c regression models together with a fuzzy c-partitioning of the data. The regression models take the general form

$$y_k = f_i(\mathbf{x}_k; \boldsymbol{\theta}_i), \tag{3.26}$$

where the functions f_i are parameterized by $\boldsymbol{\theta}_i \in \mathbb{R}^{p_i}$. The membership degree $\mu_{ik} \in \mathbf{U}$ is interpreted as a weight representing the extent to which the value predicted by the model $f_i(\mathbf{x}_k; \boldsymbol{\theta}_i)$ matches y_k. The prediction error is defined by:

$$E_{ik}(\boldsymbol{\theta}_i) = [y_k - f_i(\mathbf{x}_k; \boldsymbol{\theta}_i)]^2 \tag{3.27}$$

but other measures can be applied as well, provided they fulfill the minimizer property stated by Hathaway and Bezdek (1993b). The family of objective functions for fuzzy c-regression models is defined for $\mathbf{U} \in M_{fc}$ and $(\boldsymbol{\theta}_1, \ldots \boldsymbol{\theta}_c) \in \mathbb{R}^{p_1} \times \mathbb{R}^{p_2} \times \ldots \times \mathbb{R}^{p_c}$ by:

$$E_m(\mathbf{U}, \{\boldsymbol{\theta}_i\}) = \sum_{i=1}^{c} \sum_{k=1}^{N} (\mu_{ik})^m E_{ik}(\boldsymbol{\theta}_i) . \tag{3.28}$$

One possible approach to minimize the objective function (3.28) is the grouped coordinate minimization method (Hathaway and Bezdek, 1991a), given in Algorithm 3.5.

A specific situation for Step 1 of the algorithm arises when the regression functions f_i in (3.26) are linear in the parameters $\boldsymbol{\theta}_i$. In such a case, the parameters can be obtained as a solution of a weighted least-squares problem where the membership

Algorithm 3.5 (Fuzzy c-regression models)

Given a set of data $\mathbf{Z} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, specify c, the structure of the regression models (3.26) and the error measure (3.27). Choose the weighting exponent $m > 1$ and the termination tolerance $\epsilon > 0$. Initialize the partition matrix randomly, such that $\mathbf{U}^{(0)} \in M_{fc}$.

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

Step 1: **Calculate values for the model parameters** $\boldsymbol{\theta}_i^*$ that globally minimize the function $E_m(\mathbf{U}^{(l)}, \{\boldsymbol{\theta}_i\})$.

Step 2: **Update the partition matrix:**

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^{c} (E_{ik}/E_{jk})^{2/(m-1)}}, \quad 1 \leq i \leq c, \quad 1 \leq k \leq N.$$

if $E_{ik} = 0$ for some $i = s$, set $\mu_{sk} = 1$ and $\mu_{ik} = 0, \forall i \neq s$.

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

degrees of the fuzzy partition matrix $\mathbf{U}$ serve as the weights. Define the matrix $\mathbf{X} \in \mathbb{R}^{N \times p}$, the vector $\mathbf{y} \in \mathbb{R}^N$, and the matrix $\mathbf{W}_i \in \mathbb{R}^{N \times N}$, as follows:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \quad \mathbf{W}_i = \begin{bmatrix} \mu_{i1} & 0 & \cdots & 0 \\ 0 & \mu_{i2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mu_{iN} \end{bmatrix}. \tag{3.29}$$

The optimal parameters $\boldsymbol{\theta}_i$ are then computed by:

$$\boldsymbol{\theta}_i = \left[\mathbf{X}_e^T \mathbf{W}_i \mathbf{X}_e\right]^{-1} \mathbf{X}_e^T \mathbf{W}_i \mathbf{y}. \tag{3.30}$$

The FCRM algorithm suffers from the same drawback as the FCV algorithms, as the clusters are not limited in size. The advantage of the algorithm is that is can also fit locally nonlinear models to data, such as polynomials, which are still linear in their parameters and hence lead to a linear estimation problem in Step 1 of Algorithm 3.5.

3.6 Possibilistic Clustering

The clustering approaches derived from the FCM functionals use the "probabilistic" constraint (3.5b), which states that the sum of membership degrees of each data point equals one. It has been recognized that the membership degrees generated by FCM-based algorithms do not always correspond to the degree of typicality. These problems

arise in situations, where the total membership of a data point to all the clusters does not equal one, as in the presence of outliers, see Example 3.3. Several approaches have been suggested to replace (3.5b) by a less restrictive constraint. The method proposed by Krishnapuram and Keller (1993) uses the following objective function:

$$J(\mathbf{Z}, \eta; \mathbf{U}, \mathbf{V}) = \sum_{i=1}^{c}\sum_{k=1}^{N}(\mu_{ik})^m \|\mathbf{z}_k - \mathbf{v}_i\|_A^2 + \sum_{i=1}^{c}\eta_i \sum_{k=1}^{N}(1-\mu_{ik})^m, \qquad (3.31)$$

where η_i are positive constants. The first term is identical to the FCM objective function (3.7a). The second term forces the memberships μ_{ik} to be as large as possible, thus avoiding the trivial solution of the optimization problem, $\mathbf{U} = 0$, which would occur after simply removing constraint (3.5b). Note that the columns in $\mathbf{U}$ are now independent, which makes it possible to decompose the global objective function (3.31) into c individual objective functions for the c clusters. Differentiating with respect to $\mathbf{U}$ and setting to zero leads to the following necessary condition for μ_{ik}:

$$\mu_{ik} = \frac{1}{1 + \left(\frac{D_{ik\mathbf{A}_i}}{\eta_i}\right)^{2/(m-1)}}, \qquad (3.32)$$

where $D_{ik\mathbf{A}}^2 = \|\mathbf{z}_k - \mathbf{v}_i\|_{\mathbf{A}}^2$ is the squared inner-product norm. The necessary conditions for the prototypes $\mathbf{v}_i$ are identical to the corresponding conditions for FCM and its derivatives. The value of η_i determines the width of the resulting possibility distribution and simultaneously specifies the relative weighting of the second term in (3.31). The same value may be chosen for all clusters, if they all are expected to be similar, or it can be chosen based on the initial partition, in proportion to the average intra-cluster distance:

$$\eta_i = \frac{\sum_{k=1}^{N}(\mu_{ik})^m D_{ik\mathbf{A}}^2}{\sum_{k=1}^{N}(\mu_{ik})^m}. \qquad (3.33)$$

The basic "possibilistic" c-means (PCM) algorithm, which follows as a straightforward generalization of the FCM iteration, is given in Algorithm 3.6.

As mentioned above, by removing the constraint (3.5b), the membership functions of the c clusters become independent of each other. This makes PCM more sensitive to initialization, since nothing prevents the algorithm from converging to degenerate possibilistic partitions where all clusters are identical, or very similar to each other. Typically, FCM may be used to find an initial partition for PCM. The concept of possibilistic clustering has also been applied to the GK algorithm and other FCM derivatives (Krishnapuram and Keller, 1993).

Example 3.6 To illustrate the difference between FCM and PCM, these two algorithms are applied to an artificial data set, similar to the set in Example 3.1. In both cases, the settings of the parameters are: $\mathbf{A}_i = \mathbf{I}$ for all clusters, $m = 2$ and $\epsilon = 0.01$. The FCM algorithm is initialized with a random partition matrix, and the PCM with the partition generated by FCM. In the following, we refer to the left cluster as cluster 1 and to the right cluster as cluster 2.

By comparing the level curves in Figure 3.7a and Figure 3.7b, one can see that the membership degrees generated by the possibilistic algorithm correspond to the

Algorithm 3.6 (Possibilistic c-means (PCM))

Given the data set $\mathbf{Z}$ and a good initial partition $\mathbf{U}^{(0)} \in M_{pc}$, choose the number of clusters $1 < c < N$, the weighting exponent $m > 1$, the termination tolerance $\epsilon > 0$ and the norm-inducing matrix $\mathbf{A}$. Estimate η_i using (3.33).

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

Step 1: **Compute cluster prototypes (means):**

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

Step 2: **Compute the distances:**

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

Step 3: **Update the partition matrix:**

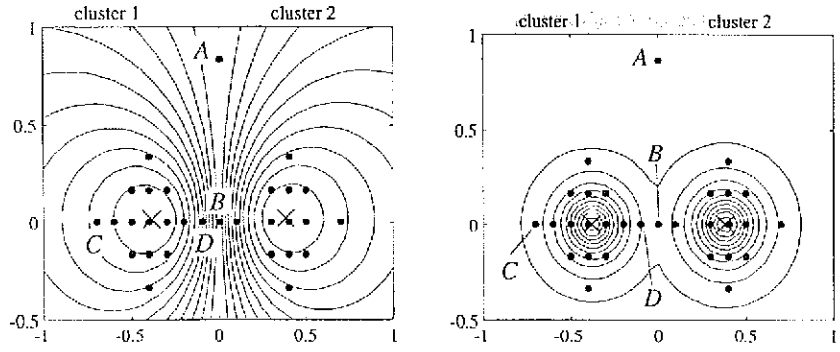
$$\mu_{ik}^{(l)} = \frac{1}{1 + \left(\frac{D_{ik\mathbf{A}}}{\eta_i} \right)^{2/(m-1)}} \quad 1 \leq i \leq c, \quad 1 \leq k \leq N.$$

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

distances from the cluster prototype, and that they are not influenced by the neighboring cluster (the level curves generated by PCM are almost circular while in the case of FCM they are distorted near the cluster boundary).

Note also that the performance of PCM is not influenced by the presence of the outlier A. As expected, the membership degrees assigned to this point by FCM are $\mu(A) = [0.4973, 0.5027]^T$, thus approximately equal to the membership degrees of B, $\mu(B) = [0.4931, 0.5069]^T$. It is obvious that point B is much closer to the prototypes of both clusters than A, and thus should have a greater degree of membership. The PCM algorithm accounts for this difference by assigning much lower membership degrees to A than to B ($\mu(A) = [0.0215, 0.0244]^T$ and $\mu(B) = [0.1147, 0.1263]^T$, respectively).

Further, it is interesting to note that the possibilistic partition correctly reflects the symmetrical form of the clusters (the level curves in Figure 3.7b are almost circular). This observation can be confirmed by examining the memberships of points C and D that are both approximately at the same distance from the center of cluster 1. Since, in the fuzzy partition, the membership degrees are relative to the distance from a point to *all* the clusters, point C receives greater membership in the cluster 1,



(a) Fuzzy clustering. *(b) Possibilistic clustering.*

Figure 3.7. Comparison of the partitions generated by fuzzy and possibilistic c-means algorithms.

$\mu(C) = [0.9234, 0.0766]^T$, than point D with $\mu(D) = [0.7367, 0.2633]^T$ which is closer to cluster 2. The PCM algorithm considers points C and D equally typical for cluster 1 and assigns them similar membership degrees $\mu(C) = [0.1500, 0.0177]^T$ and $\mu(D) = [0.1935, 0.0832]^T$ respectively. □

3.7 Determining the Number of Clusters

When clustering real data without any a priori information about the data structure, one usually has to make assumptions about the number of underlying subgroups (clusters) c in the data. 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:

- Clustering data for different values of c, and using *validity measures* to assess the goodness of the obtained partitions. Different scalar validity measures have been proposed in the literature. Section 4.5.1 gives an overview of validity measures used with the adaptive distance clustering algorithms, and demonstrates their performance on several examples.
- Starting with a sufficiently large number of clusters, and successively reducing this number by merging clusters that are similar (compatible) with respect to some predefined criteria. This approach, called *compatible cluster merging*, is presented in Section 4.5.2.

3.8 Data Normalization

Distance norms are sensitive to variations in the numerical ranges of the different features. The Euclidean distance, for example, assigns more weighting to features with wide ranges than to those with narrow ranges. The result of clustering can thus be neg-

atively influenced by, for instance, choosing different measurement units. In pattern recognition literature, it is often suggested that the data should be appropriately normalized before clustering (Jain and Dubes, 1988). The simplest type of normalization is the subtraction of the feature means $\bar{z}_j$:

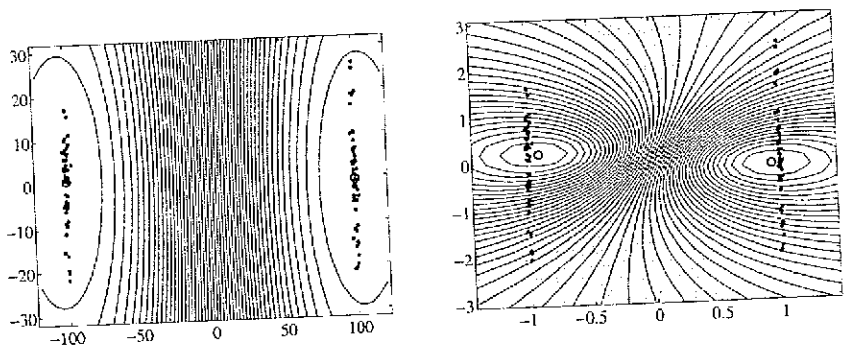
$$z_{jk} = z^*_{jk} - \bar{z}_j, \tag{3.34}$$

which makes the feature values invariant to rigid displacements of the coordinates. The asterisk denotes the raw (unscaled) data. Another type of normalization translates and scales the axes so that all the features have zero mean and unit variance:

$$z_{jk} = \frac{z^*_{jk} - \bar{z}_j}{\sigma_j}. \tag{3.35}$$

However, normalization is not always desirable, as it may alter the separation between clusters and negatively influence the results of clustering. It turns out that clustering algorithms based on adaptive distance measure, see Section 3.4, are less sensitive to data scaling, since the adaptation of the distance measure automatically compensates for the differences in scale. The following example illustrates this property.

Example 3.7 An artificial data set contains two well-separated clusters which are relatively far apart along the horizontal axis. The clusters are practically linear and parallel to the vertical axis. Figure 3.8a shows level curves of a fuzzy partition obtained with the FCM algorithm on the original (unscaled data).

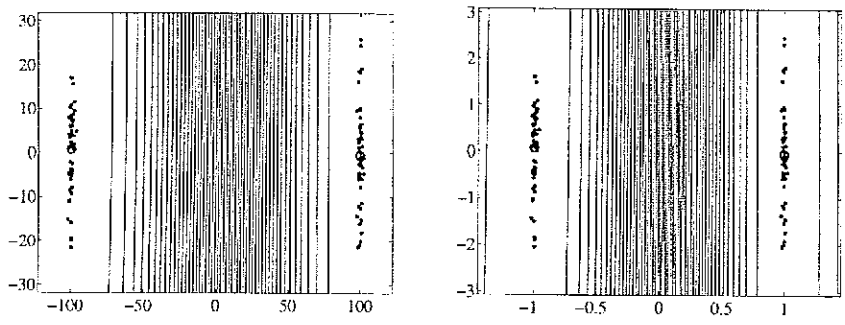


(a) Original data. *(b) Normalized data.*

Figure 3.8. The fuzzy c-means algorithm is sensitive to the scaling (normalization) of data. The dots represent the data points, the circles are the cluster means.

Because of the large distance between the two clusters, FCM is able to pick the cluster shapes quite correctly. The same algorithm was applied to data normalized by eq. (3.35). Note that the scales in Figure 3.8a and Figure 3.8b are different. Since the distance between the clusters becomes considerably smaller due to scaling, the partition reflects the influence of the two clusters on each other, and the geometrical shape of

the clusters no longer corresponds to the underlying data structure. Figure 3.9 gives the results for the GK algorithm. Note that almost identical partitions are obtained for both the raw and the normalized data sets. □



(a) Original data. *(b) Normalized data.*

Figure 3.9. The Gustafson-Kessel algorithm is less sensitive to the data scale. The dots represent the data points, the circles are the cluster means.

3.9 Summary and Concluding Remarks

Fuzzy clustering is a powerful unsupervised method for data analysis. A large number of clustering algorithms have been proposed in the literature, and applied to a variety of real-world problems. In this chapter, methods that can be used to detect clusters contained in subspaces of the data space have been presented. These methods can be applied to the approximation of nonlinear systems, and can facilitate the task of building and analyzing models of complex systems based on numerical data, as shown in Chapter 5. This particular aim imposes some requirements on the performance and validation of the clustering algorithms that may be quite different from those usually considered in the pattern recognition literature. A discussion of this issue and analysis of the selected algorithms is presented in the following chapter.