UNIT 2 FUZZY CLUSTERING

Structure **Page No**

2.1 INTRODUCTION

Clustering is an important data analysis tool that provides improved data understanding. It plays a key role in searching for structures in data. Cluster analysis is aimed at dividing data whose identity is not known in advance, into homogeneous groups or clusters such that similar data objects belong to the same cluster and dissimilar data objects to different clusters. A *cluster* is, therefore, a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters. In other words, given a finite set of data, X, the problem of clustering in X is to find several cluster centers that are required to form a *partition* of X such that the degree of association is strong for data within blocks of the partition and weak for data in different blocks. However, this requirement is too strong in many practical applications, and it is thus desirable to replace it with a weaker requirement. When the requirement of a crisp partition of X is replaced with a weaker requirement of a *fuzzy partition* or a *fuzzy pseudo partition* on X, we refer to the emerging problem area as *fuzzy clustering* as in Section 2.4. Fuzzy pseudo partitions are often called *fuzzy c-partitions*, where *c* designates the number of fuzzy classes in the partition.

Clustering is an active research area and there are wide arrays of clustering algorithms that have been proposed in literature are discussed in Section 2.6. These algorithms differ from each other in their approach, in the cluster quality they achieve and also in terms of computational efficiency. In Section 2.7, we will look at these issues in the light of fuzzy clustering techniques. There are two basic methods of fuzzy clustering. One of them, which are based on fuzzy c-partitions, is called a *fuzzy equivalence relation-based hierarchical clustering method*. We will see these methods in details in this unit. Various modifications of these methods can be found in the literature. We have also introduced some major application areas of fuzzy clustering in Section 2.8.

Objectives

After studying this unit, you should be able to

- know the essence the concept of clustering;
- define the fuzzy clustering;
- differentiate the classical cluster analysis and the fuzzy cluster analysis;
- know various distance function useful for clustering;
- apply fuzzy clustering to solve real-life problems.

2.2 WHAT IS CLUSTERING?

Data clustering is a common technique for statistical data analysis, which is used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics. Clustering is the classification of similar objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait – often proximity according to some defined distance measure. Machine learning typically regards data clustering as a form of unsupervised learning. Besides the term *data clustering* (or just *clustering*), there are a number of term with similar meanings, including *cluster analysis*, *automatic classification*, *numerical taxonomy*, *botryology* and *typological analysis*.

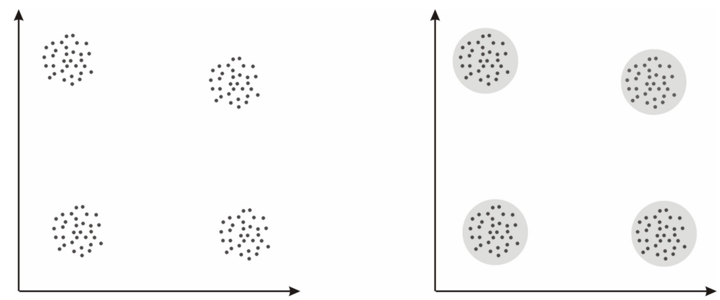

Fig. 1: An example of clusters.

Clustering deals with finding a *structure* in a collection of unlabeled data. A loose definition of clustering could be "the process of organizing objects into groups whose members are similar in some way". A *cluster* is, therefore, a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters. We can show this with a simple graphical example shown in Fig.1.

In this case we easily identify the 4 clusters into which the data can be divided where the similarity criterion is *distance*, i.e., two or more objects belong to the same cluster if they are "close' according to a given distance (in this case geometrical distance). This is called *distance-based clustering*. Another kind of clustering is *conceptual clustering* in which two or more objects belong to the same cluster if it defines a concept *common* to all that objects. In other words, objects are grouped according to their fit to descriptive concepts, not according to simple similarity measures.

The goal of clustering is to determine the intrinsic grouping in a set of unlabeled data. But how to decide what constitutes a good clustering? It can be shown that there is no absolute "best" criterion which would be independent of the final aim of the clustering. Consequently, it is the user which must supply this criterion, in such a way that the result of the clustering will suit their needs.

For instance, we could be interested in finding representatives for homogeneous groups (*data reduction*), in finding "natural clusters" and describe their unknown properties (*"natural" data types*), in finding useful and suitable groupings (*"useful data classes*) or in finding unusual data objects (*outlier detection*).

Clustering techniques have been extensively studied in statistics, machine learning and data mining. Clustering has many interesting commercial applications, and is extensively applied for studying the data pertaining to an organisational application. It helps in discovering hitherto unknown patterns in data, which can be fruitfully engaged in taking future marketing decisions. Some common uses of clustering are stated here:

- **Computational Biology and Bioinformatics –** In transcriptomics, clustering is used to build groups of genes with related expression patterns. Often such groups contain functionally related proteins, such as enzymes for a specific pathway, or genes that are co-regulated. High throughput experiments using expressed sequence tags (ESTs) or DNA microarrays can be a powerful tool for genome annotation, a general aspect of genomics.

  In sequence analysis, clustering is used to group homologous sequences into gene families. This is a very important concept in bioinformatics, and evolutionary biology in general. See evolution by gene duplication.

- **Plant and Animal Ecology** – In the fields of plant and animal ecology, clustering is used to describe and to make spatial and temporal comparisons of communities (assemblages) of organisms in heterogeneous environments; it is also used in plant systematic to generate artificial phylogenies or clusters of organisms (individuals) at the species, genus or higher level that share a number of attributes.

- **Marketing Research** – Cluster analysis is widely used in market research when working with multivariate data from surveys and test panels. Market researchers use cluster analysis to partition the general population of consumers into market segments and to better understand the relationship between different groups of consumers/potential customers for the following purposes.

  - Segmenting the Market and determining target markets
  - Product positioning
  - New Product development
  - Selecting test markets

- **Insurance** – Insurance companies use clustering to identify policy holders with high average policy claims; identifying frauds.

- **City-Planning –** Identifying groups of houses according to their house type, value and geographical location.

- **Earthquake Studies** – Past data on earthquake is clustered to study the nature of continental faults.

- **Document Clustering –** Clustering proves to be one of the most promising techniques to reign in the huge unwieldy collection of documents on electronic media. Clustering similar documents together and representing them with a cluster center helps users get a glimpse of contents of a whole collection together without the tedious task of reading all of them.

- **Data Mining** – Many data mining applications involve partitioning data items into related subsets; the marketing applications discussed above represent some examples. Another common application is the division of documents, such as World Wide Web pages, into genres.

- **Social Network Analysis** – In the study of social network, clustering may be used to recognize communities within large groups of people.

- **Image Segmentation** – Clustering can be used to divide a digital image into distinct regions for border detection or object recognition.

- **WWW –** Document classification; clustering web-log data to discover groups of similar access patterns.

The main requirements that a clustering algorithm should satisfy are:

- Scalability
- Dealing with different types of attributes

- Discovering clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Ability to deal with noise and outliers
- Insensitivity to order of input records
- High dimensionality
- Interpretability and usability

There are a number of limitations with clustering. Among them:

- current clustering techniques do not address all the requirements adequately (and concurrently);
- dealing with large number of dimensions and large number of data items can be problematic because of time complexity;
- the effectiveness of the method depends on the definition of "distance" (for distance-based clustering);
- if and *obvious* distance measure doesn't exist we must "define" it, which is not always easy, especially in multi-dimensional spaces;
- the result of the clustering algorithm (that in many cases can be arbitrary itself) can be interpreted in different ways.

Now we shall discuss fuzzy cluster analysis in the following sections.

2.3 FUZZY CLUSTER ANALYSIS

Cluster analysis divides data into groups (clusters) such that similar data objects belong to the same cluster and dissimilar data objects to different clusters. The resulting data partition improves data understanding and reveals its internal structure. Partitional clustering algorithms divide up a data set into clusters or classes, where similar data objects are assigned to the same cluster whereas dissimilar data objects should belong to different clusters. In real applications there is very often no sharp boundary between clusters so that fuzzy clustering is often better suited for the data. Membership degree between zero and one are used in fuzzy clustering instead of crisp assignments of the data to clusters. The most prominent fuzzy clustering algorithm is the fuzzy c-means, a fuzzification of k-means.

Areas of application of fuzzy cluster analysis include for example data analysis, pattern recognition, and image segmentation. The detection of special geometrical shapes like circles and ellipses can be achieved by so-called shell clustering algorithms. Fuzzy clustering belongs to the group of soft computing techniques (which include neural nets, fuzzy systems, and genetic algorithms).

2.4 CLASSICAL VS. FUZZY CLUSTERING

Clustering is motivated by the need to find interesting patterns or groupings in a given set of data. For example, a market analysis company may want to collect data about a group of customers (e.g., through a survey questionnaire or interviews) and analyze these data by finding interesting groupings of customers. The result of such an analysis can be used by a company to develop its marketing strategy or to develop a family of product lines that are targeted toward various customer groupings.

In the area of pattern recognition and image processing, clustering is often used to perform the task of "segmenting" the image (i.e., partitioning pixels on an image into regions that correspond to different objects or different faces of objects in the images).

This is because image segmentation can be viewed as a kind of data clustering problem where each datum is described by a set of image features (e.g., intensity, color, texture, etc.) of a pixel.

Classical clustering approaches generate partitions in a partition; each pattern belongs to only one cluster. So, the clusters in a partition are disjoint. In other words, classical clustering algorithms find a "hard partition" of a given dataset based on certain criteria that evaluate the goodness of a partition. By "hard partition" we mean that each datum belongs to exactly one cluster of the partition. More formally, we can define the concept "hard partition" as follows:

Definition 1 (Hard Partition): Let X be a set of data, and x_l be an element of X. A partition $P = \{C_1, C_2, \ldots, C_k\}$ of X is "hard" if and only if the following two condition hold.

(i) for all $x_i \in X$ there exists a partition $C_i \in P$ such that $x_i \in C_j$.

(ii) for all $x_i \in X, x_i \in C_j \Rightarrow x_i \notin C_k$ where $k \neq j, C_k C_j \in P$.

The first condition in the definition assures that the partition covers all data points in X, whereas the second condition assures that all clusters in the partition are mutually exclusive.

In many real-world clustering problems, however, some data points partially belong to multiple clusters, rather than to a single cluster exclusively. For example, a pixel in a Magnetic Resonance Image (MRI) may correspond to a mixture of two different types of tissues. A particular customer may be a "borderline case" between two groups of customers (e.g., between moderate conservatives and moderate liberals). These observations motivated the development of the "fuzzy clustering" algorithm. A fuzzy algorithm finds "soft partition" of a given data set based on certain criteria. In a soft partition, a datum can partially belong to multiple clusters.

Formally this can be defined as follows:

Definition 2 (Soft Partition): Let X be a set of data, and x_i be an element of X. A partition $P = \{C_1, C_2, \ldots, C_k\}$ of X is "soft" if and only if the following two conditions hold.

(i) for all $x_i \in X$ and for all $C_j \in P, 0 \leq \mu_{cj}(x_i) \leq 1$ such that $x_i \in C_j$.

(ii) for all $x_i \in X$ there exists $C_j \in P$ such that $\mu_{cj}(x_i) > 0$.

where $\mu_{cj}(x_i)$ denote clustering of special interest to which x_i belongs to cluster C_j.

A type of fuzzy clustering of special interest is one that ensures the membership degree of a point x in all clusters adding up to one, i.e.,

$$\sum_j \mu_{cj}(x_i) = 1 \qquad \forall x_i \in X$$

A soft partition that satisfies this additional condition is called a *constrained soft partition*. The fuzzy c-means algorithm, which is best known fuzzy clustering algorithm, produces a constrained partition.

2.5 DISTANCE MEASURE IN CLUSTERING

The central idea of clustering is to group data points in such a way so that similar elements belong to the same cluster and objects belonging to different clusters are more different from each other than those belonging to the same cluster. To formalize the idea of *closeness* or *nearness*, central to clustering is the concept of a *distance measure* between data points. If the components of the data instance vectors are all in the same physical units then it is possible that the simple Euclidean distance metric is sufficient to successfully group similar data instances. However, even in this case the Euclidean distance can sometimes be misleading. Fig. 2 illustrates this with an example of the width and height measurements of an object. Despite both measurements being taken in the same physical units, an informed decision has to be made as to the relative scaling. As Fig. 2 shows, different scaling can lead to different clustering.

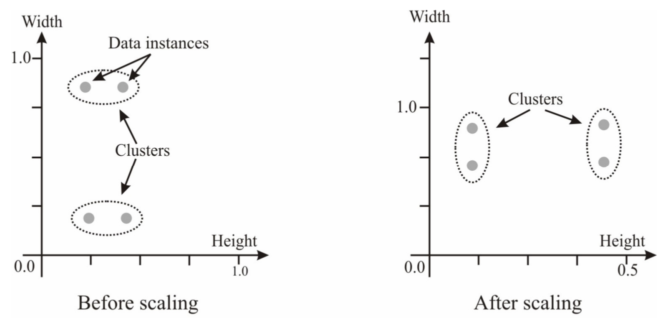

Fig. 2: Clusters before scaling and after scaling.

Notice however that this is not only a graphic issue: the problem arises from the mathematical formula used to combine the distances between the single components of the data feature vectors into a unique distance measure that can be used for clustering purposes: different formulas leads to different clustering. Again, domain knowledge must be used to guide the formulation of a suitable distance measure for each particular application.

For higher dimensional dart, a popular measure is the Minkowski metric,

$$d_p(x, y) = \left(\sum_{i=1}^{d} | x_i - y_i |^p \right)^{1/p}$$

where d is the dimensionality of the data. The *Euclidean* distance is a special case where $p = 2$, while *Manhattan* metric has $p = 2$. However, there are no general theoretical guidelines for selecting a measure for any given application. It is often the case that the components of the data feature vectors are not immediately comparable. It can be that the components are not continuous variables, like length, but nominal categories, such as the days of the week. In these cases again, domain knowledge must be used to formulate an appropriate measure.

Suppose x and y are two data points in a dataset and $d(x, y)$ denotes the distance between x and y and satisfies the following properties:

(i) $d(x, x) = 0$ i.e., a data item is at distance zero from itself.

(ii) $d(x, y) = d(y, x)$ i.e., distance is a symmetric notion.

(iii) $d(x, y) \le d(x, y) + d(z, y)$ i.e., distance measure satisfies the triangle inequality.

Given a k-dimensional data set, the data item x and y can be represented as $x = (x_1, x_2, \ldots, x_k)$ and $y = (y_1, y_2, \ldots, y_k)$. Some of the commonly used distance measures for clustering can be summarized as follows:

(i) Euclidean distance or "L_2 norm" which is given by

$$d(x, y) = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}$$

(ii) Manhattan distance or "L_1 norm" which is given by

$$d(x, y) = \sum_{i=1}^{k} | x_i - y_i |$$

(iii) Maximum of dimensions or L_∞ norm which is given by

$$\max_{i=1}^{k} | x_i - y_i |$$

(iv) Cosine measure – This is a non-Euclidean distance measure and represents a similarity measure between two patterns. If the two patterns are represented as two vectors, the cosine of angle between the two vectors is identical to their correlation coefficient. The pattern similarity measure is thus given by

$$\text{similarity } (x, y) = \frac{\sum (xy)}{\sum x^2 \sum y^2}$$

(v) Minkowski metric – This is given by

$$d_p(x, y) = \left(\sum_{i=1}^{d} | x_i - y_i |^p \right)^{1/p}$$

where, d is the dimensionality of the data.

In the following section, we shall discuss clustering algorithm.

2.6 CLUSTERING ALGORITHMS

Clustering algorithms may be classified as listed below:

- **Exclusive Clustering** – In this approach data are grouped in an exclusive way, so that if a certain datum belongs to a definite cluster then it could not be included in another cluster. A simple example of it is shown in the Fig. 3, where the separation of points is achieved by a straight line on a bi-dimensional plane. K-means algorithm is an example of exclusive clustering.

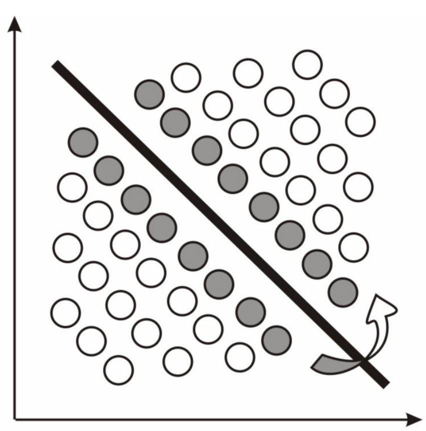

Fig. 3: Clusters before scaling and after scaling.

- **Overlapping Clustering –** The overlapping clustering, uses fuzzy sets to cluster data, so that each point may belongs to two or more clusters with different degrees of membership. In this case, data will be associated to an appropriate membership value. Fuzzy c-means is an overlapping clustering.

- **Hierarchical Clustering –** Instead, a hierarchical clustering algorithm is based on the union between the two nearest clusters. The beginning condition is realized by setting every datum as a cluster. After a few iterations it reaches the final clusters wanted.
- **Probabilistic Clustering** – Such type of clustering use a completely probabilistic approach. Mixture of Gaussian is an example of probabilistic clustering.

2.7 FUZZY CLUSTERING ALGORITHMS

As discussed earlier, traditional clustering approaches generate partitions; in a partition, each pattern belongs to only one cluster. So, the clusters in a partition are disjoint. Fuzzy clustering extends this notion to associate each partition with every cluster using a membership function. The output of such algorithms is a clustering, but not a partition. Given a set data $X = \{x_1, x_2, \ldots, x_n\}$, where x_k , in general, is a vector

$$x_k[x_{k1}, x_{k2}, \ldots, x_{kp}] \in \Re^p$$

For all $k \in N_n$ the problem of fuzzy clustering is to find a fuzzy pseudo partition and the associated cluster centers by which the structure of the data is represented as best as possible.

Definition 3 (Fuzzy Pseudo Partition or Fuzzy c-Partition): Let $x = \{x_1, x_2, \ldots, x_n\}$ be a set of given data. A fuzzy pseudo partition or fuzzy c-partition of X is a family of fuzzy subsets of X, denoted by $P\{A_1, A_2, \ldots, A_n\}$ which satisfies

$$\sum_{i=1}^{c} A_1(x_k) = 1$$

for all $k \in N_n$ and

$$0 < \sum_{k=1}^{n} A_i(x_k) < n$$

for all $i \in N_c$, where c is a positive integer.

Example 1: Let $X = \{x_1, x_2, x_3\}$ and

$$A_1 = 0.6/x_1 + 1/x_2 + 0.1/x_3$$

$$A_2 = 0.4/x_1 + 0/x_2 + 0.9/x_3$$

Then $\{A_1, A_2\}$ is a pseudo partition or fuzzy 2-partition of X. Fuzzy quantizations (or granulations) of variables in fuzzy systems are also examples of fuzzy pseudo partition.

Some of the objective function-based clustering algorithms includes, amongst others, the

- *Fuzzy c-means algorithm (FCM)*: It generates spherical clusters of approximately the same size.
- *Gustafson-Kessel algorithm (GK)*: It provides ellipsoidal clusters with approximately the same size. There are also axis-parallel variants of this algorithm. The algorithm can also be used to detect lines (to some extent).
- *Gath-Geva algorithm (GG)/Gaussian mixture decom*position *(GMD)*: This is used to generate ellipsoidal clusters with varying size. This algorithm has also

axis-parallel variants. The algorithm can also be used to detect lines (to some extent).

- *Fuzzy c-varieties algorithm (FCV)*: It is generally used for the detection of linear manifolds (infinite lines in 2D).
- *Adaptive fuzzy c-varieties algorithm (AFC)*: It can be used to detect line segments in 2D data.
- *Fuzzy c-shells algorithm (FCS)*: It is used for detection of circles (no closed form solution for prototypes).
- *Fuzzy c-spherical shells algorithm (FCSS)*: It detects circular patterns in data.
- *Fuzzy c-rings algorithm (FCR)*: It is also used for detection of circles.
- *Fuzzy c-quadric shells algorithm (FCQS)*: It is used for detection of rectangles (and variants thereof).

Now, let us focus on fuzzy C-mean algorithm.

2.8 FUZZY C-MEANS (FCM) ALGORITHM

The most frequently used fuzzy clustering algorithm is the Fuzzy c-Means (FCM) which is a fuzzification of the k-means algorithm. FCM is a method of clustering which allows one piece of data to belong to two or more clusters. This method was originally proposed by Dunn in 1973 and then improved upon by Bezdek in 1981.

As discussed in Section 2.6, the problem of fuzzy clustering is to find a fuzzy pseudo partition and the associated cluster centers by which the structure of the data is represented as best as possible. This requires some criterion expressing the general idea that associations (in the sense described by the criterion) be strong within clusters and weak between clusters. To solve the problem of fuzzy clustering, the criterion can be formulated in term of a *performance index* which is usually based upon cluster centers. Given a pseudo partition $P = \{A_1, A_2, \ldots, A_c\}$, the cluster centers, $v_1, v_2, \ldots, v_c$ associated with the partition are calculated by the formula given below:

$$v_i = \frac{\sum_{k=1}^{n} [A_i(x_k)]^m x_k}{\sum_{k=1}^{n} [A_i(x_k)]^m}$$

for all $i \in N_c$, where $m > 1$ is a real number that governs the influence of membership grades. It can be observed in the above equation that the cluster center v_i of a fuzzy class A_i is actually the weighted average of data in A_i. The weight of a datum x_k is the m^{th} power of the membership grade of x_k in the fuzzy set A_i.

The performance index of a fuzzy pseudo partition P, $J_m(P)$, is then defined in terms of the cluster centers by the following formula.

$$J_m(P) = \sum_{k=1}^{n} \sum_{j=1}^{c} \left[A_i(x_k)^m\right] \|x_k - v_i\|^2$$

where $\|*\|$ is some inner product-induced norm in space R^p and $\|x_k - v_i\|$ represents the distance between x_k and v_i. This performance index measures the weighted sum of distances between cluster centers and elements in the corresponding fuzzy clusters. Clearly, the smaller the values of $J_m(P)$, the better the fuzzy pseudo partition P. Therefore, the goal of the fuzzy c-means clustering method is to find a fuzzy pseudo partition P that minimizes the performance index $J_m(P)$. That is, the clustering problem is an optimization problem.

Fuzzy c-means Algorithm

Input: Desired number of clusters c, a real number $m \in (1, \infty)$ and a small positive number ε, serving as a stopping criterion.

Output: A fuzzy pseudo partition and the associated cluster centers.

Steps:

1. Let $t = 0$. Select an initial fuzzy pseudo partition $P^{(0)}$.

2. Calculate the c cluster centers $v_1^{(t)}, v_2^{(t)}, \ldots, v_c^{(t)}$, by using the equation

$$v_i = \frac{\sum_{k=1}^{n} [A_i(x_k)]^m x_k}{\sum_{k=1}^{n} [A_i(x_k)]^m}$$

   For $p^{(t)}$ and the chosen value of m.

3. Update $p^{(*)}$ by the following procedure: For each $x_k \in X$, if $\left\| x_k - v_i^{(t)} \right\|^2 > 0$ for all $i \in N_c$, then define

$$A_i^{(t+1)}(x_k) = \left[\sum_{j=1}^{c} \left(\frac{\left\| x_k - v_i^{(t)} \right\|^2}{\left\| x_k - v_j^{(t)} \right\|^2} \right)^{\frac{1}{m-1}} \right]^{-1}$$

   If $\left\| x_k - v_i^{(t)} \right\|^2 = 0$ for some $i \in I \subseteq N_c$, then define $A_i^{(t+1)}(x_k)$ for $i \in I$ by any nonnegative real number satisfying

$$\sum_{i \in I} A_i^{(t+1)}(x_k) = 1$$

   And define $A_i^{(t+1)}(x_k) = 0$ for $i \in N_c - I$

4. Compare $P^{(t)}$ and $P^{(t+1)}$. That is, compute the distance $| P^{(t+1)} - P^{(t)} |$ between $P^{(t)}$ and $P^{(t+1)}$ in the space $R^{n \times c}$. If $| P^{(t+1)} - P^{(t)} | \leq \varepsilon$, then stop; otherwise, increase t by one and return to step 2.

In the fuzzy c-means algorithm, the parameter m is selected according to the problem under consideration. When $m \to 1$, the fuzzy c-means converges to a "generalized" classical c means. When $m \to \infty$, all clusters tend towards the centroid of the data set X i.e., is the partition becomes fuzzier with increasing m. Currently, there is no theoretical basis for an optimal choice for the value of m. However, it is established that the algorithm converges for any $m \in (1, \infty)$.

We will now illustrate the functioning of the algorithm using a simple one-dimensional data set.

Example 2: Let us suppose the data points are distributed on an axis, say x-axis as shown in Fig. 4.

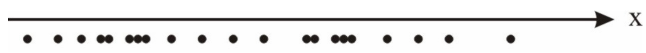

Fig. 4: Data points and their positions.

Visual inspection of Fig. 4 reveals the existence of two prominent clusters in the data which may be centered around the two locations where the points are more concentrated than other places. Suppose these two clusters are named A and B, and their divisions are shown with the dotted line in Fig. 5. Using the partitional hard clustering algorithm, i.e. the k-means algorithm, each data point will be associated to either cluster A or B. Hence if a membership function is drawn to represent the membership of a datum to cluster A, the function would look as shown in Fig. 5. It shows that points belonging to cluster A have membership value 0 to cluster B.

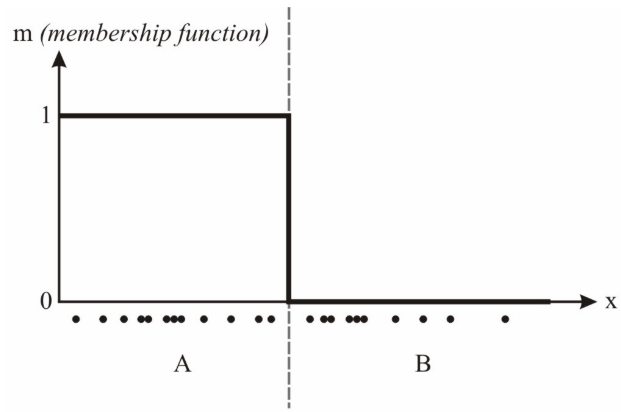

Fig. 5: Membership function for partitional hard clustering of data.

With the Fuzzy c means clustering algorithm, however each datum may not belong exclusively to one cluster. Rather a datum may be placed in both the clusters. A possible membership function for cluster A that can model this situation is shown below. The datum shown as a red spot belongs more to cluster B than to cluster A. Its membership value to cluster A is 0.2.

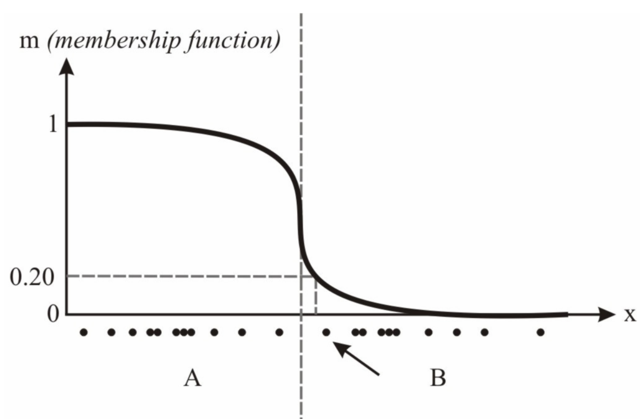

Fig. 6: Fuzzy membership function for cluster 'A'.

Now, instead of using a graphical representation, a matrix $U = [A_i\ (x_k)]$ can be used to represent the membership values to the different clusters. Matrix U will have as many rows as the number of data points and as many columns as the number of clusters chosen. For a hard clustering approach, the values in this matrix are either 0 or 1, as shown by the left hand side matrix for $c = 2$. For a fuzzy clustering matrix however, these values lie between 0 and 1. For example, the right side of Fig. 7 shows that the first datum has membership values 0.8 and 0.2 to clusters A and B, respectively, while the second datum had membership values 0.3 and 0.7 to clusters A and B, respectively.

$$U_{n\times c}=\begin{bmatrix}1 & 0\\ 0 & 1\\ 1 & 0\\ \dots & \dots\\ 0 & 1\end{bmatrix}\quad U_{n\times c}=\begin{bmatrix}0.8 & 0.2\\ 0.3 & 0.7\\ 0.6 & 0.4\\ \dots & \dots\\ 0.1 & 0.1\end{bmatrix}$$

Fig. 7: Data points and their positions.

Following is another example, where a set of one-dimensional data points are divided into three clusters. Starting with an arbitrary membership value as shown in Fig. 8(a), the Fig. 8(b) and 8(c) show how the fuzzy membership values evolve, when the algorithm is implemented with a value of 2 for m. The color of the data is that of the nearest cluster according to be membership function. The final convergence shown in Fig. 8(c) had occurred for $\max_{ik}\left\{\left|A_i(x_k)^{(t+1)}-A_i(x_k)^{(t)}\right|\right\}<0.3$ which was achieved after 37 iterations. It should be kept in mind that different initializations may lead to different final membership values. It may also be the case that starting with different initializations, the same membership values may be obtained, though with a different number of iterations.

The value of m controls the smoothness of the membership function curves. For a large value of m, the transition values of membership from one cluster to another cluster are smoother than those obtained with lower values of m. When $m=1$, the transition is abrupt and represents the hard clustering technique. Fig.9(a) and (b) show two membership curve for $m=1.5$ and $m=2$, respectively.

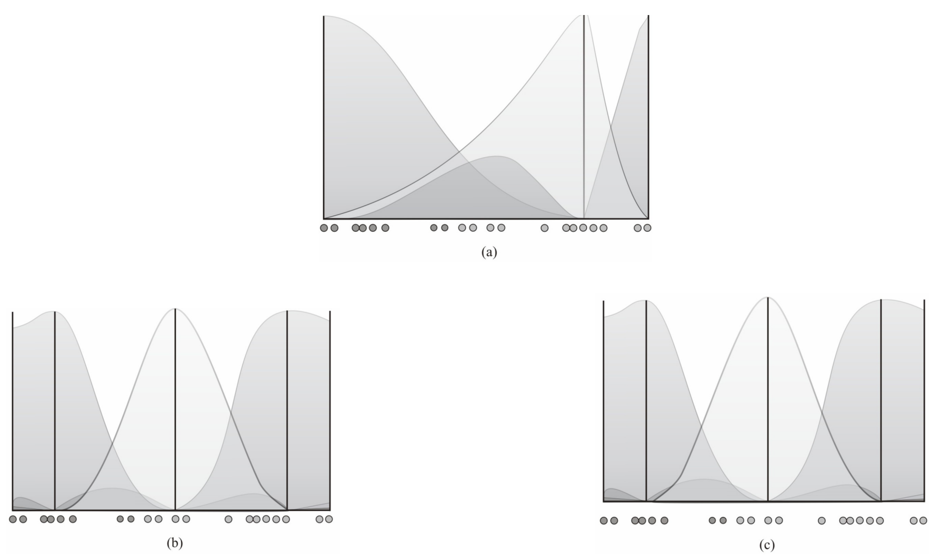

Fig. 8: (a) Initial cluster membership functions for 3 clusters; (b) Intermediate cluster membership functions for 3 clusters; and (c) Final cluster membership functions for 3 clusters.

Fig. 9: Fuzzy clustering membership functions for (a) m = 1.5; and (b) m = 2.

Try the following exercises.

E1) Consider a dataset of six points given in the following table, each of which has two features f_1 and f_2. Assuming the values of the parameters c and m as 2 and the initial cluster centers $v_1 = (5, 5)$ and $v_2(10, 10)$, apply FCM algorithm to find the new cluster center after one iteration.

	f_1	f_2
x_1	2	12
x_2	4	9
x_3	7	13
x_4	11	5
x_5	12	7
x_6	14	4

E2) Consider the following two-dimensional dataset that consists of 15 points in R^2.

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_{k1}	0	0	0	1	1	1	2	3	4	5	5	5	6	6	6
x_{k2}	0	2	4	1	2	3	2	2	2	1	2	3	0	2	4

Assume that we want to determine a fuzzy pseudo partition with two clusters (i.e., $c = 2$). Assume further that $m = 1.25$; $\|*\|$ is the Euclidean distance, and the initial fuzzy pseudo partition is $P^{(0)} = \{A_1, A_2\}$ with

$$A_1 = .854/x_1 + .854/x_2 + \cdots + .854/x_{15}$$
$$A_2 = .146/x_1 + .146/x_2 + \cdots + .146/x_{15}$$

Starting with the initial membership values given above, obtain the final fuzzy pseudo partition and the cluster centers assuming that convergence is achieved when the difference between the two values is ≤ 0.01.

Let us now summarize the unit.

2.9 SUMMARY

In this unit, we have discussed the fundamental concepts of the fuzzy clustering and their applications. We have also discussed how fuzzy clustering differ with classical clustering process. Specifically we have covered the following:

1. Elaborated the essence of clustering and that of fuzzy clustering to solve real life complex problems.
2. Provided a brief introduction of the distance measures used to measure the closeness of the objects.
3. Provided a comparative introduction of the classical clustering and fuzzy clustering.
4. Introduce different classes of clustering algorithms.
5. Provided details about fuzzy clustering algorithms with focus on fuzzy c-means algorithms.

2.10 SOLUTIONS/ANSWERS

E1) The initial membership values of the data point, x_1, in the two clusters can be calculated by using the equation:

$$\mu_{c1}(x_1) = \frac{1}{\sum_{j=1}^{2}\left(\frac{\|x_1 - v_1\|^2}{\|x_1 - v_1\|}\right)^2}$$

$$\|x_1 - v_1\|^2 = 3^2 + 7^2 = 9 + 49 = 58$$

$$\|x_1 - v_2\|^2 = 8^2 + 2^2 = 64 + 4 = 68$$

$$\mu_{c1}(x_1) = \frac{1}{\frac{58}{58} + \frac{58}{68}} = \frac{1}{1 + 0.853} = 0.5397$$

Similarly, we obtain the following:

$$\mu_{c2}(x_1) = \frac{1}{\frac{68}{58} + \frac{68}{68}} = 0.4603$$

$$\mu_{c1}(x_2) = \frac{1}{\frac{17}{17} + \frac{17}{37}} = 0.6852$$

$$\mu_{c2}(x_2) = \frac{1}{\frac{37}{17} + \frac{37}{37}} = 0.3148$$

$$\mu_{c1}(x_3) = \frac{1}{\frac{68}{68} + \frac{68}{18}} = 0.2093$$

$$\mu_{c2}(x_3) = \frac{1}{\frac{18}{68} + \frac{18}{18}} = 0.7907$$

$$\mu_{c1}(x_4) = \frac{1}{\frac{36}{36} + \frac{36}{26}} = 0.4194$$

$$\mu_{c2}(x_4) = \frac{1}{\frac{26}{36} + \frac{26}{26}} = 0.5806$$

$$\mu_{c1}(x_5) = \frac{1}{\frac{53}{53} + \frac{53}{13}} = 0.197$$

$$\mu_{c2}(x_5) = \frac{1}{\frac{13}{53} + \frac{13}{13}} = 0.803$$

$$\mu_{c1}(x_6) = \frac{1}{\frac{82}{82} + \frac{82}{52}} = 0.3881$$

$$\mu_{c2}(x_6) = \frac{1}{\frac{52}{82} + \frac{52}{52}} = 0.6119$$

Now by using equation $v_1 = \frac{\sum_{k=1}^{6} (\mu_{c1}(x_k))^2 \times x_k}{\sum_{k=1}^{6} (\mu_{c1}(x_k))^2}$ the new co-ordinate values for the centre v_1 can be calculated as:

$$\frac{0.5397^2 \times (2,12) + 0.6852^2 \times (4,9) + 0.2093^2 \times (7,13) + 0.4194^2 \times (11,5) + 0.197^2 \times (12,7) + 0.3881^2 \times (14,4)}{0.5397^2 + 0.6852^2 + 0.2093^2 + 0.4194^2 + 0.197^2 + 0.3881^2}$$

$$= \frac{732761}{1.0979}, \frac{10.044}{1.0979} = (6.6273,\ 9.1484)$$

Similarly, by using the equation $v_2 = \frac{\sum_{k=1}^{6} (\mu_{c2}(x_k))^2 \times x_k}{\sum_{k=1}^{6} (\mu_{c2}(x_k))^2}$ the new co-ordinate values for the center v_2 can be calculated as:

$$\frac{0.4603^2 \times (2,12) + 0.3148^2 \times (4,9) + 0.7909^2 \times (7,13) + 0.5806^2 \times (11,5) + 0.803^2 \times (12,7) + 0.6119^2 \times (14,4)}{0.4603^2 + 0.3148^2 + 0.7909^2 + 0.5806^2 + 0.803^2 + 0.6119^2}$$

$$= \frac{22.326}{2.2928}, \frac{19.4629}{2.2928} = (9.7374,\ 8.4887)$$

E2) The algorithm stops for $t = 6$, and we obtain the fuzzy pseudo partition defined in the following table:

Fuzzy pseudo partition $\mathbf{P^{(6)} = \{A_1, A_2\}}$

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$A_1(x_k)$	0.99	1	0.99	1	1	1	0.99	0.47	0.01	0	0	0	0.01	0	0.01
$A_2(x_k)$	0.01	0	0.01	0	0	0	0.01	0.53	0.99	1	1	1	0.99	1	0.99

The two cluster centers are: $v_1 = (0.88,\ 2)$ and $v_2 = (5.14,\ 2)$.

2.11 PRACTICAL ASSIGNMENT

Session 1 & 2

Implement the FCM algorithm, using C/C++ language and test it to find the final fuzzy pseudo partition and cluster centers for the two-dimensional data sets given in Table 1 and 2, assuming that convergence is achieved when the difference between the two values is less than or equal to 0.05. Plot the data to determine how many clusters you should assume. Starting with different initial membership matrices check whether the final membership functions are always same? After how many iterations did you achieve the final value in each case? What can you say about the clusters?

Table 1 (Dataset-1)

k	1	2	3	4	5	6	7	8	9	10
x_{k1}	0.958	1.043	1.907	0.780	0.579	0.003	0.001	0.014	0.007	0.004
x_{k2}	0.003	0.001	0.003	0.002	0.001	0.105	1.748	1.839	1.021	0.214

Table 2 (Dataset-2)

k	k	1	2	3	4	5	6	7	8
x_{k1}	x_{k1}	0.958	1.043	1.907	0.780	0.579	0.003	0.001	0.014
x_{k2}	x_{k2}	0.003	0.001	0.003	0.352	0.556	0.105	1.748	1.839