CHAPTER

10 Clustering Analysis

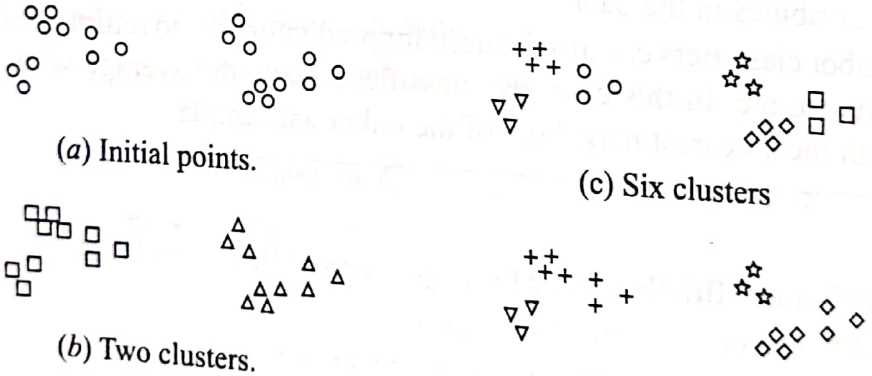
INSIDE THIS CHAPTER

10.1 Introduction; **10.2** Goals of Clustering; **10.3** What Cluster Analysis is Not; **10.4** Applications; **10.5** Requirements of Clustering in Data Mining; **10.6** Data Types in Cluster Analysis; **10.7** Categorization of Clustering Methods; **10.8** Partitioning Methods; **10.9** Hierarchical Clustering; **10.10** Density-Based Methods; **10.11** Grid-Based Methods; **10.12** Model-Based Clustering Methods; **10.13** Outlier Analysis.

10.1 INTRODUCTION

In this chapter we continue with the theme of extracting information from unlabelled data and turn to the important topic of *clustering*. When human beings try to make sense of complex questions, our natural tendency is to break the subject into smaller pieces, each of which can be explained more simply. Clustering is a technique used for combining observed objects into groups or clusters such that:

- Each group or cluster is homogeneous or compact with respect to certain characteristics. That is, objects in each group are similar to each other.



(*a*) Initial points.

(c) Six clusters

(*b*) Two clusters.

(*d*) Four clusters.

- Each group should be different from other groups with respect to the same characteristics; that is, objects of one group should be different from the objects of other groups.

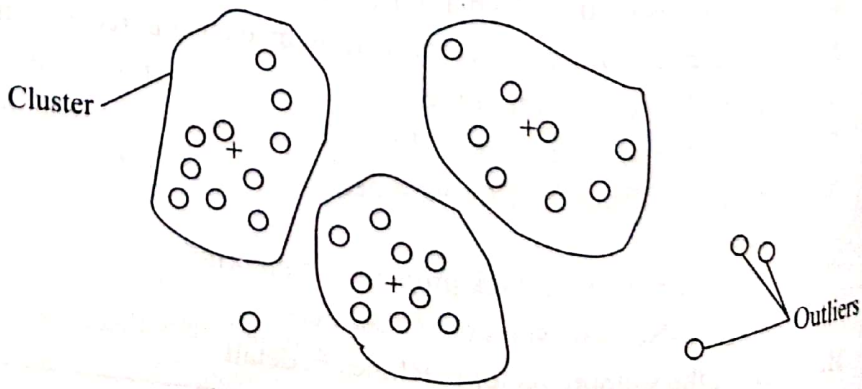


Fig. 10.1

Outliers are objects that do not belong to any cluster or form clusters of very small cardinality

Clustering can be defined as the process of grouping a collection of N patterns into distinct segments or clusters based on a suitable notion of closeness or similarity among these patterns. Good clusters show high similarity within a group and low similarity between patterns belonging to two different groups. For applications such as customer or product segmentation, clustering is the primary goal.

In many fields there are obvious benefits to be had from grouping together similar objects. For example:

- In an economics application we might be interested in finding countries whose economies are similar.
- In a financial application we might wish to find clusters of companies that have similar financial performance.
- In a marketing application we might wish to find clusters of customers with similar buying behaviour.
- In a medical application we might wish to find clusters of patients with similar symptoms.
- In a document retrieval application we might wish to find clusters of documents with related content.
- In a crime analysis application we might look for clusters of high volume crimes such as burglaries or try to cluster together much rarer (but possibly related) crimes such as murders.

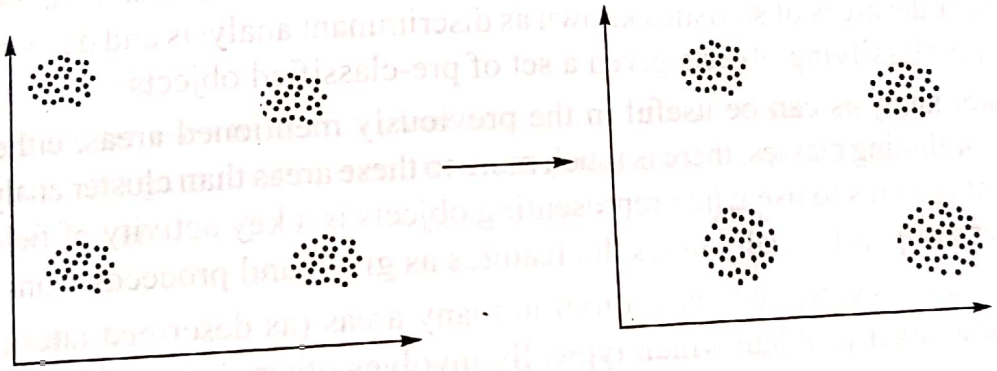


Fig. 10.2

In the example, we easily identify the 4 clusters into which the data can be divided; the similarity criterion is distance: 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*.

Clustering is the process of grouping the data into classes or clusters so that objects within a cluster have high similarity in comparison to one another, but are very dissimilar to objects in other clusters.

10.1.1 Euclidean Distance

This is the most usual, "natural" and intuitive way of computing a distance between two samples. It takes into account the difference between two samples directly, based on the magnitude of changes in the sample levels. This distance type is usually used for data sets that are suitably normalized or without any special distribution problem.

10.1.2 Manhattan Distance

Also known as city-block distance, this distance measurement is especially relevant for discrete data sets. While the **Euclidean distance** corresponds to the length of the shortest path between two samples (*i.e.*, "as the crow flies"), the Manhattan distance refers to the **sum of distances** along each dimension (*i.e.*, "walking round the block").

10.2 GOALS OF CLUSTERING

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*).

10.3 WHAT CLUSTER ANALYSIS IS NOT

Cluster analysis is a classification of objects from the data, where by *classification* we mean a labeling of objects with class (group) labels. As such, clustering does not use previously assigned class labels, except perhaps for verification of how well the clustering worked. Thus, cluster analysis is distinct from pattern recognition or the areas of statistics known as discriminant analysis and decision analysis, which seek to find rules for classifying objects given a set of pre-classified objects.

While cluster analysis can be useful in the previously mentioned areas, either directly or as a preliminary means of finding classes, there is much more to these areas than cluster analysis. For example, the decision of what features to use when representing objects is a key activity of fields such as pattern recognition. Cluster analysis typically takes the features as given and proceeds from there.

Thus, cluster analysis, while a useful tool in many areas (as described later), is normally only part of a solution to a larger problem which typically involves other steps and techniques.

10.4 APPLICATIONS

Cluster analysis is an important human activity. Early in childhood, one learns how to distinguish between cats and dogs, or between animals and plants, by continuously improving subconscious classification schemes. Cluster analysis has been widely used in numerous applications, including pattern recognition, data analysis, image processing, and market research. By clustering, one can identify crowded and sparse regions, and therefore, discover overall distribution patterns and interesting correlations among data attributes.

In business, clustering may help marketers discover distinct groups in their customer bases and characterize customer groups based on purchasing patterns. In biology, it can be used to derive plant and animal taxonomies, categorize genes with similar functionality, and gain insight into structures inherent in populations. Clustering may also help in the identification of areas of similar land use in an earth observation database, and in the identification of groups of motor insurance policy holders with a high average claim cost, as well as the identification of groups of houses in a city according to house type, value, and geographical location. It may also help classify documents on the WWW for information discovery.

As a data mining function, cluster analysis can be used as a stand-alone tool to gain insight into the distribution of data, to observe the characteristics of each cluster, and to focus on a particular set of clusters for further analysis. Alternatively, it may serve as a preprocessing step for other algorithms, such as classification and characterization, operating on the detected clusters.

In machine learning, clustering is an example of ***unsupervised learning***. Unlike classification, clustering and unsupervised learning do not rely on predefined classes and class-labeled training examples. For this reason, *clustering is a form of* ***learning by observation***, *rather than* ***learning by examples***. In conceptual clustering, a group of objects forms a class only if it is describable by a concept. This differs from conventional clustering which measures similarity based on geometric distance. Conceptual clustering consists of two components:

1. It discovers the appropriate classes, and
2. It forms descriptions for each class, as in classification. The guideline of striving for high interclass similarity and low interclass similarity still applies.

In data mining, efforts have focused on finding methods for efficient and effective cluster analysis in large databases.

10.4.1 Applications of Clustering

- **Market Research:** It can help marketers discover distinct groups in their customer bases and characterize customer groups based on purchasing patterns.
- **Pattern Recognition**
- **Data analysis:** It can also be used to help classify documents on the Web for information discovery.
- **Image Processing**
- **Biology:** It can be used to derive plant and animal taxonomies, categorize genes with similar functionality, and gain insight into structures inherent in populations.
- **Geography:** It helps in the identification of areas of similar land use in an earth observation database and in the identification of groups of houses in a city according to house type, value, and geographic location.
- **Automobile Insurance:** The identification of groups of automobile insurance policy holders with a high average claim cost.
- **Outlier detection:** The detection of credit card fraud and the monitoring of criminal activities in electronic commerce.

10.5 REQUIREMENTS OF CLUSTERING IN DATA MINING

1. **Scalability:** Many clustering algorithms work well in small data sets containing less than 200 data objects; however, a large database may contain millions of objects. Clustering on a sample of a given large data set may lead to biased results. Therefore, we need highly scalable clustering algorithms for large databases.
2. **Ability to Deal with Different Types of Attributes:** Many algorithms are designed to cluster interval-based (numerical) data. However, applications may require clustering other types of data, such as binary, categorical (nominal), and ordinal data, or mixtures of these data types.
3. **Insensitivity to the Order of Input Record:** Some clustering algorithms are sensitive to the order of input data, e.g., the same set of data, when presented with different orderings to such an algorithm, may generate dramatically different clusters. It is important to develop algorithms which are insensitive to the order of input.

4. **Discovery of Clusters with Arbitrary Shape:** Many clustering algorithms determine clusters based on Euclidean *or Manhattan distance measures*. Algorithms based on such distance measures tend to find spherical clusters with similar size and density. However, a cluster could be of any shape. It is important to develop algorithms which can detect clusters of arbitrary shape.
5. **Minimal Requirements for Domain knowledge to Determine Input Parameters:** Most of the clustering algorithms require users to input certain parameters in cluster analysis (such as the number of desired clusters). Slight variations in these parameters may lead to very different results. These parameters are very difficult to determine, especially for data sets containing high-dimensional objects. This is actually a burden on users, but also it makes the quality of clustering difficult to control.
6. **Ability to Deal with Noisy Data:** Most real-world databases contain outliers or missing, unknown, or erroneous data. Some clustering algorithms are sensitive to such data and may lead to clusters of poor quality.
7. **High Dimensionality:** A database or a data warehouse may contain several dimensions or attributes. Many clustering algorithms are good at handling low-dimensional data, involving only two to three dimensions. Human eyes are good at judging the quality of clustering for up to three dimensions. It is challenging to cluster data objects in high-dimensional space, especially considering that data in high-dimensional space can be very sparse and highly skewed.
8. **Constraint-based Clustering:** Real-world applications may need to perform clustering under various kinds of constraints. Suppose we want to choose the locations for a given number of new ATMs in a city. To decide upon this, you may cluster households while considering constraints such as the city's rivers and highway networks, and customer requirements per region. A challenging task is to find groups of data with good clustering behaviour that satisfy specified constraints.
9. **Interpretability and Usability:** Users expect clustering results to be interpretable, comprehensible, and usable. That is, clustering may need to be tied up with specific semantic interpretations and applications. It is important to study how an application goal may influence the selection of clustering methods.

Example: What is Cluster Analysis?

Ans.

1. The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering.
2. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. A cluster of data objects can be treated collectively as one group and so may be considered as a form of data compression.
3. First the set is partitioned into groups based on data similarity (e.g., using clustering), and then labels are assigned to the relatively small number of groups.
4. It is also called **unsupervised learning**. Unlike classification, clustering and unsupervised learning do not rely on predefined classes and class-labeled training examples. For this reason, clustering is a form of learning by observation, rather than learning by examples.
5. **Advantages** of such a clustering-based process:
   - adaptable to changes
   - helps single out useful features that distinguish different groups.

$$S(x, y) = \frac{x' \cdot y}{\| x \| \| y \|}$$

where x' is a transpositions of vector x.

$\| x \|$ is the Euclidean norm of vector x. The Euclidean norm of vector $x = (x_1, x_2, \ldots, x_p)$ defined as $\sqrt{x_1^2 + x_2^2 + \ldots + x_p^2}$

$\| y \|$ is the Euclidean norm of vector y.

and S is essentially the consine of the angle between vectors x and y.

When variables are binary-valued (0 or 1), the above similarity function can be interpreted in terms of shared features and attributes. Suppose an object x possesses the i^{th} attribute of $x_i = 1$.

Then $x' \cdot y$ is the number of attributes possessed by both x and y and $|x||y|$ is the geometric mean of the number of attributes possessed by x and the number possessed by y. Thus $S(x, y)$ is a measure of relative possession of common attributes.

Example: Suppose, we are given two vectors $x = (1, 0, 1, 0)$ and $y = (0, 0, 1, 1)$. By equation, the similarity between x and y is

$$S(x, y) = \frac{0 + 0 + 1 + 0}{\sqrt{2}\sqrt{2}} = \frac{1}{2} = 0.5$$

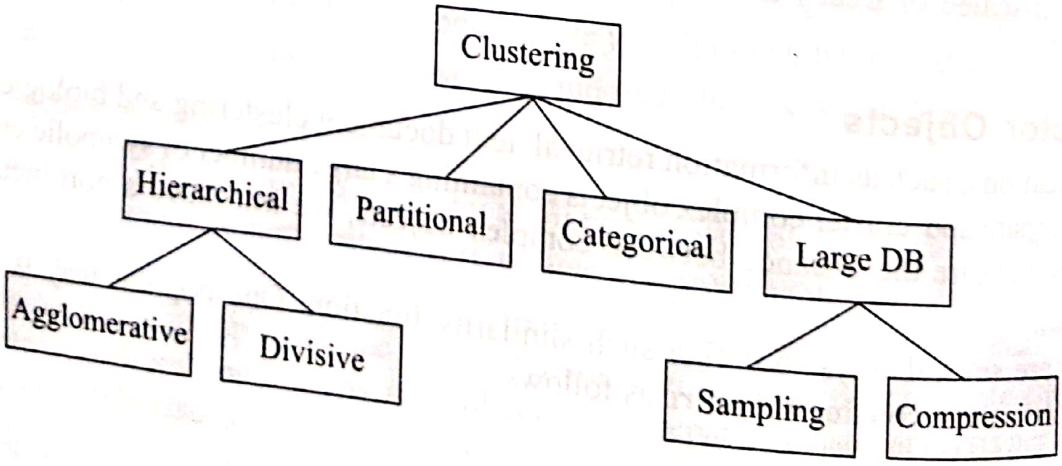
10.7 CATEGORIZATION OF CLUSTERING METHODS

There are many clustering algorithms. But there is no crisp categorization of clustering methods because these categories may be overlapping. So that a method may have features from several categories.

Typically, the major clustering methods can be classified into the following categories:

1. Partitioning methods
2. Hierarchical methods
3. Density-based methods
4. Grid-based methods
5. Model-based methods.

Aside from the above categories of clustering methods, there are two classes of clustering tasks the require special attention. One is **clustering high-dimensional data** and the other is **constraint based clustering.**



1. Partitioning Methods: Given a database of n objects or data tuples, a partitioning method constructs k partitions of the data, where each partition represents a cluster and $k \leq n$. That is, it classifies the data into k groups, which together satisfy the following requirements:

1. each group must contain at least one object, and
2. each object must belong to exactly one group.

Given k, the number of partitions to construct, a partitioning method creates an initial partitioning. It then uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another. Examples are the *k-means algorithm* and the *k-medoids algorithm*.

2. Hierarchical Methods: A hierarchical method creates a hierarchical decomposition of the given set of data objects. A hierarchical method can be classified as being either agglomerative or divisive, based on how the hierarchical decomposition is formed. The *agglomerative approach*, also called the bottom-up approach, starts with each object forming a separate group. It successively merges the objects or groups that are close to one another, until all of the groups are merged into one (the topmost level of the hierarchy), or until a termination condition holds. The *divisive approach*, also called the top-down approach, starts with all of the objects in the same cluster. In each successive iteration, a cluster is split up into smaller clusters, until eventually each object is in one cluster, or until a termination condition holds.

3. Density-based Methods: The general idea is to continue growing the given cluster as long as the density in the "neighbourhood" exceeds some threshold. Such a method can be used to filter out noise (outliers) and discover clusters of arbitrary shape. **DBSCAN** and its extension, **OPTICS**, are typical density-based methods.

4. Grid-based Methods: Grid-based methods quantize the object space into a finite number of cells that form a grid structure. All of the clustering operations are performed on the grid structure. The main advantage of this approach is its fast processing time, which is typically independent of the number of data objects and dependent only on the number of cells in each dimension in the quantized space. STING is a typical example of a grid-based method.

5. Model-based Methods: Model-based methods hypothesize a model for each of the clusters and find the best fit of the data to the given model. A model-based algorithm may locate clusters by constructing a density function that reflects the spatial distribution of the data points.

Clustering High-dimensional Data is a particularly important task in cluster analysis because many applications require the analysis of objects containing a large number of features or dimensions. As the number of dimensions increases, the data become increasingly sparse so that the distance measurement between pairs of points become meaningless and the average density of points anywhere in the data is likely to be low. Therefore, a different clustering methodology needs to be developed for high-dimensional data. **CLIQUE** and **PROCLUS** are two influential subspace clustering methods.

Constraint-based Clustering is a clustering approach that performs clustering by incorporation of user-specified or application-oriented constraints. A constraint expresses a user's expectation or describes "properties" of the desired clustering results, and provides an effective means for communicating with the clustering process.

10.8 PARTITIONING METHODS

Given a database of n objects, and k, the number of clusters to form, a partitioning algorithm organizes the objects into k partitions ($k \leq n$), where each partition represents a cluster. The clusters are formed to optimize an objective partitioning criterion, often called a ***similarity function***, such as distance, so

that the objects within a cluster are "similar", whereas the objects of different clusters are "dissimilar" in terms of the database attributes.

The most well-known and commonly used partitioning methods are ***k-means***, ***k-medoids***, and their variations.

Both these techniques are based on the idea that a center point can represent a cluster. For *k*-means, we use the notion of a centroid, which is the mean or median point of a group of points. Note that a centroid almost never corresponds to an actual data point. For *K*-medoid, we use the notion of a medoid, which is the most representative (central) point of a group of points. By its definition a medoid is required to be an actual data point.

10.8.1 Centroid-based Technique: The K-means Method

Simply speaking it is an algorithm to classify or to group our objects based on attributes/features into *k* number of group, *k* is positive integer number. The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid. Thus, the purpose of *k*-mean clustering is to classify the data.

In data mining, ***k*-means clustering** is a method of cluster analysis which aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean.

The basic step of *k*-means clustering is simple. In the beginning we determine number of cluster K and we assume the centroid or center of these clusters. We can take any random objects as the initial centroids or the first *k* objects in sequence can also serve as the initial centroids.

Here's how the algorithm works:

1. The algorithm arbitrarily selects *k* points as the initial cluster centers ("means").
2. Each point in the dataset is assigned to the closed cluster, based upon the Euclidean distance between each point and each cluster center.

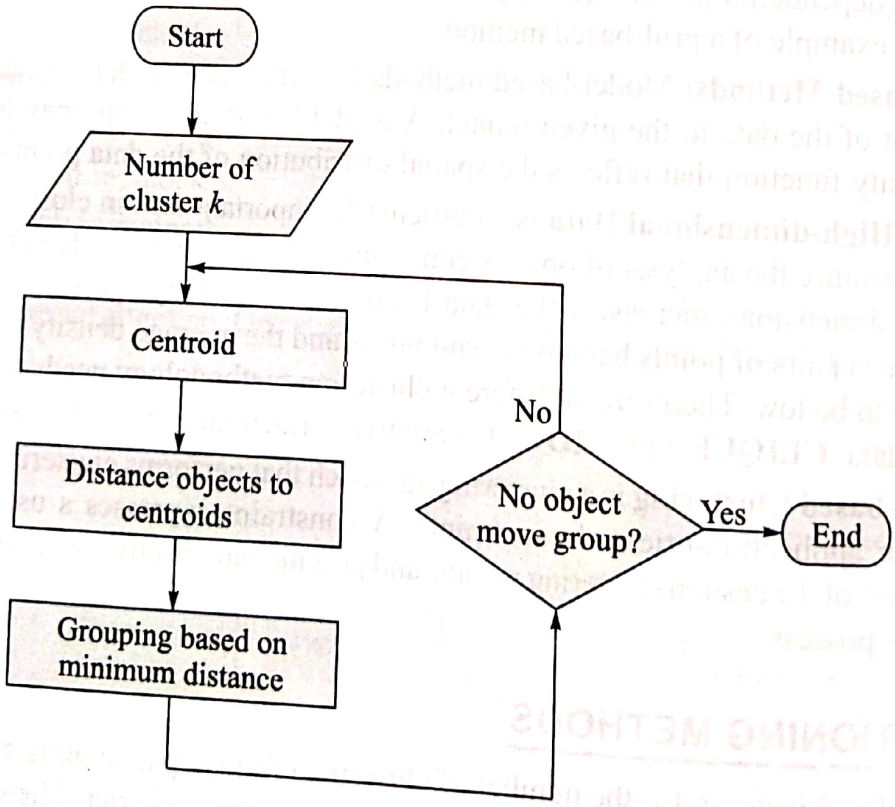


Fig. 10.5

3. Each cluster center is recomputed as the average of the points in that cluster.
4. Steps 2 and 3 repeat until the clusters converge. Convergence may be defined differently depending upon the implementation, but it normally means that either no observations change clusters when steps 2 and 3 are repeated or that the changes do not make a material difference in the definition of the clusters.

The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid.

When no point is pending, the first step is completed and an early groupage is done. At this point we need to recalculate k new centroids as barycenters of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more.

Finally, this algorithm aims at minimizing an *objective function*, in this case a squared error function. The objective function is

$$j = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the n data points from their respective cluster centers.

The algorithm is composed of the following steps:

1. *Place k points into the space represented by the objects that are being clustered. These points represent initial group centroids.*
2. *Assign each object to the group that has the closest centroid.*
3. *When all objects have been assigned, recalculate the positions of the k centroids.*
4. *Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.*

Example: Cluster the following eight points (with (x, y) representing locations) into three clusters A1(2, 10) A2(2, 5) A3(8, 4) A4(5, 8) A5(7, 5) A6(6, 4) A7(1, 2) A8(4, 9). Initial cluster centers are: A1(2, 10), A4(5, 8) and A7(1, 2). The distance function between two points a = (x_1, y_1) and b = (x_2, y_2) is defined as: $\rho(a, b) = |x_2 - x_1| + |y_2 - y_1|$.

Use k-means algorithm to find the three cluster centers after the second iteration.

Solution:

Iteration 1

	(2, 10)	(5, 8)	(1, 2)		
	Point	**Dist Mean 1**	**Dist Mean 2**	**Dist Mean 3**	**Cluster**
A1	(2, 10)				
A2	(2, 5)				
A3	(8, 4)				
A4	(5, 8)				
A5	(7, 5)				
A6	(6, 4)				
A7	(1, 2)				
A8	(4, 9)				

First we list all points in the first column of the table above. The initial cluster centers – means, are (2, 10), (5, 8) and (1, 2) - chosen randomly.

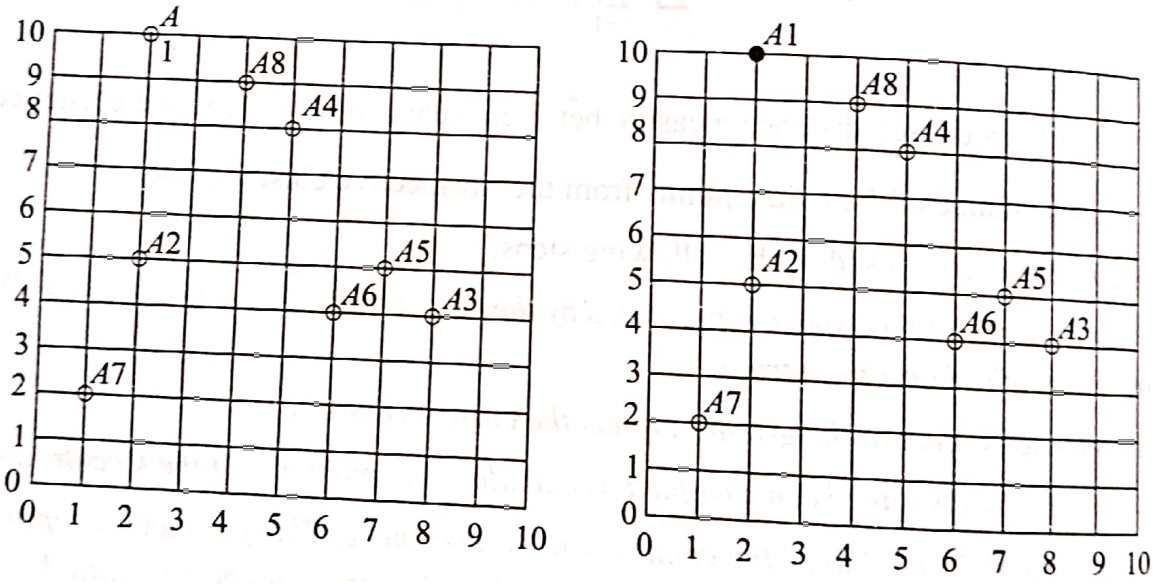


Fig. 10.6.

Next, we will calculate the distance from the first point (2, 10) to each of the three means, by using the distance function:

point	**mean 1**
x_1, y_1	x_2, y_2
(2, 10)	(2, 10)

$$\rho(a, b) = |x_2 - x_1| + |y_2 - y_1|$$

$$\rho(\text{point, mean 1}) = |x_2 - x_1| + |y_2 - y_1|$$

$$= |2 - 2| + |10 - 10|$$

$= 0 + 0$

$= 0$

mean 2

point	x_2, y_2
x_1, y_1	(5, 8)
(2, 10)	

$\rho(a, b) = |x_2 - x_1| + |y_2 - y_1|$

$\rho(\text{point, mean 2}) = |x_2 - x_1| + |y_2 - y_1|$

$= |5 - 2| + |8 - 10|$

$= 3 + 2$

$= 5$

mean 3

point	x_2, y_2
x_1, y_1	(1, 2)
(2, 10)	

$\rho(a, b) = |x_2 - x_1| + |y_2 - y_1|$

$\rho(\text{point, mean 2}) = |x_2 - x_1| + |y_2 - y_1|$

$= |1 - 2| + |2 - 10|$

$= 1 + 8$

$= 9$

So, we fill in these values in the table:

		(2, 10)	(5, 8)	(1, 2)	
	Point	**Dist Mean 1**	**Dist Mean 2**	**Dist Mean 3**	**Cluster**
A1	(2, 10)	0	5	9	1
A2	(2, 5)				
A3	(8, 4)				
A4	(5, 8)				
A5	(7, 5)				
A6	(6, 4)				
A7	(1, 2)				
A8	(4, 9)				

So, *which cluster should the point (2, 10) be placed in?*

The one, where the point has the shortest distance to the mean – that is mean 1 (cluster 1), since the distance is 0.

Cluster 1	Cluster 2	Cluster 3
(2, 10)		

So, we go to the second point (2, 5) and we will calculate the distance to each of the three means, by using the distance function:

point	mean 1
x_1, y_1	x_2, y_2
(2, 5)	(2, 10)

$$\rho(a, b) = |x_2 - x_1| + |y_2 - y_1|$$
$$\rho(\text{point, mean 1}) = |x_2 - x_1| + |y_2 - y_1|$$
$$= |2 - 2| + |10 - 5|$$
$$= 0 + 5$$
$$= 5$$

point	mean 2
x_1, y_1	x_2, y_2
(2, 5)	(5, 8)

$$\rho(a, b) = |x_2 - x_1| + |y_2 - y_1|$$
$$\rho(\text{point, mean 2}) = |x_2 - x_1| + |y_2 - y_1|$$
$$= |5 - 2| + |8 - 5|$$
$$= 3 + 3$$
$$= 6$$

point	mean 3
x_1, y_1	x_2, y_2
(2, 5)	(1, 2)

$$\rho(a, b) = |x_2 - x_1| + |y_2 - y_1|$$
$$\rho(\text{point, mean 2}) = |x_2 - x_1| + |y_2 - y_1|$$
$$= |1 - 2| + |2 - 5|$$
$$= 1 + 3$$
$$= 4$$

So, we fill in these values in the table:

Iteration 1

	(2,10)	(5,8)	(1,2)	
	Point	**Dist Mean 1**	**Dist Mean 2**	**Dist Mean 3**	**Cluster**
A1	(2,10)	0	5	9	1
A2	(2,5)	5	6	4	3
A3	(8,4)				
A4	(5,8)				
A5	(7,5)				
A6	(6,4)				
A7	(1,2)				
A8	(4,9)				

So, which cluster should the point (2, 5) be placed in?

The one, where the point has the shortest distance to the mean – that is mean 3 (cluster 3), since the distance is 0.

Cluster 1	Cluster 2	Cluster 3
(2, 10)		(2, 5)

Similarly, we fill in the rest of the table, and place each point in one of the clusters:

Iteration 1

	(2, 10)		(5, 8)	(1, 2)	
	Point	**Dist Mean 1**	**Dist Mean 2**	**Dist Mean 3**	**Cluster**
A1	(2, 10)	0	5	9	1
A2	(2, 5)	5	6	4	3
A3	(8, 4)	12	7	9	2
A4	(5, 8)	5	0	10	2
A5	(7, 5)	10	5	9	2
A6	(6, 4)	10	5	7	2
A7	(1, 2)	9	10	0	3
A8	(4, 9)	3	2	10	2

Cluster 1	Cluster 2	Cluster 3
(2, 10)	(8, 4)	(2, 5)
	(5, 8)	(1, 2)
	(7, 5)	
	(6, 4)	
	(4, 9)	

Next, we need to recompute the new cluster centers (means).

We do so, by taking the mean of all points in each cluster.

For Cluster 1, we only have one point A1 (2, 10), which was the old mean, so the cluster center remains the same.

For Cluster 2, we have $((8 + 5 + 7 + 6 + 4)/5, (4 + 8 + 5 + 4 + 9)/5) = (6, 6)$

For Cluster 3, we have $((2 + 1)/2, (5 + 2)/2) = (1.5, 3.5)$

New clusters are

Cluster 1: {A1}

Cluster 2: {A3, A4, A5, A6, A8}

Cluster 3: {A2, A7}

Centers of the new clusters are:

C1 = (2, 10)

C2 = (6, 6)

C3 = (1.5, 3.5)

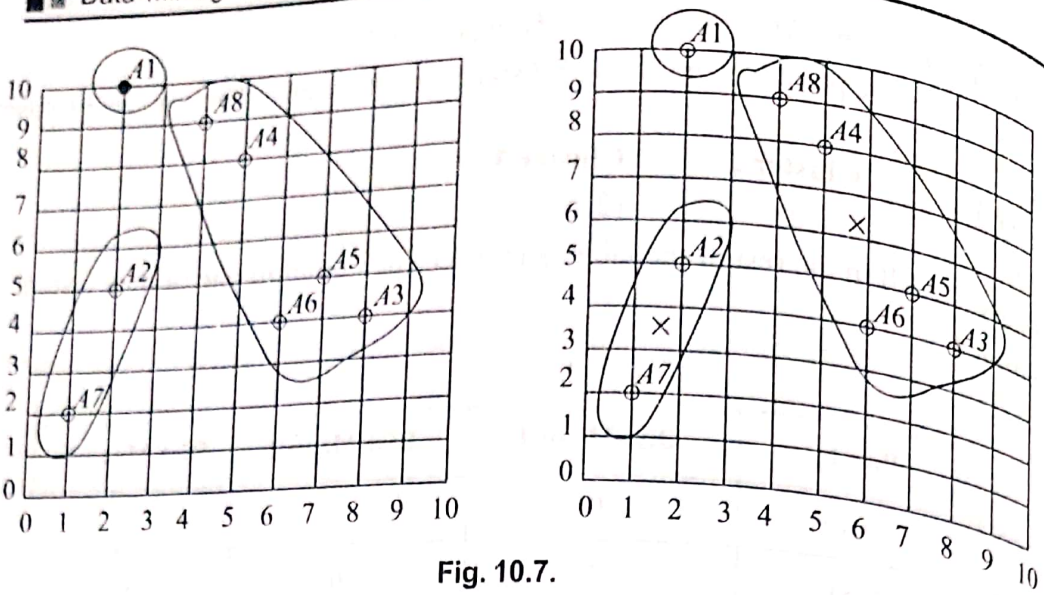


Fig. 10.7.

Next, we go to Iteration 2, Iteration 3, and so on until the means do not change anymore.

In Iteration 2, we basically repeat the process from Iteration 1 this time using the new means we computed.

After the second iteration the results would be:

1: { A1, A8}

2: { A3, A4, A5, A6}

3: {A2, A7}

Centers of the new clusters are:

C1 = (3, 9.5)

C2 = (6.5, 5.25)

C3 = (1.5, 3.5)

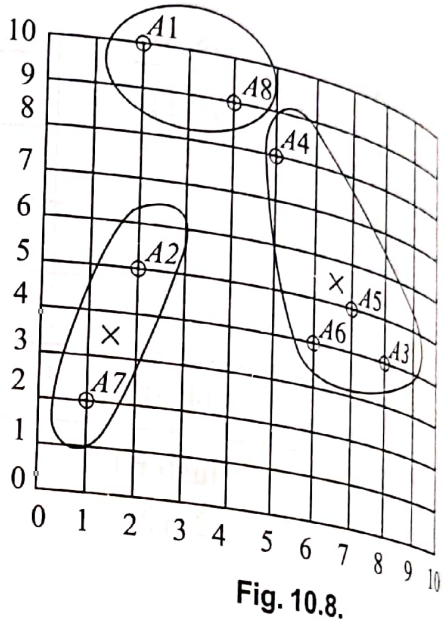


Fig. 10.8.

After the third iteration the results would be:

1: { A1, A4, A8}

2: { A3, A5, A6}

3: {A2, A7}

Centers of the new clusters are:

C1 = (3.66, 9)

C2 = (7, 4.33)

C3 = (1.5, 3.5)

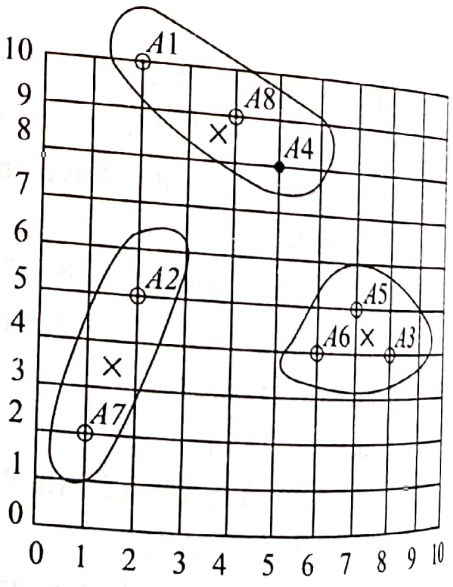
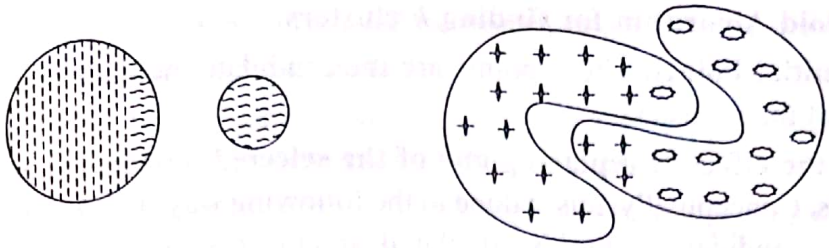


Fig. 10.9.

Limitations and problems: *k*-means attempts to minimize the squared or absolute error of points with respect to their cluster centroids. While this is sometimes a reasonable criterion and leads to a simple algorithm, *k*-means has a number of limitations and problems. In particular, Figures *a* and *b* show the problems that result when clusters have widely different sizes or have convex shapes.



(*a*) (*b*)

Fig. 10.10

The difficulty in these two situations is that the *k*-means objective function is a mismatch for the kind of clusters we are trying to find. The *k*-means objective function is minimized by globular clusters of equal size or by clusters that are well separated. The *k*-means algorithm is also normally restricted to data in Euclidean spaces because in many cases the required means and medians do not make sense. (This is not true in all cases, e.g., documents.) A related technique, ***k-medoid clustering***, does not have this restriction and is discussed in the next section.

Advantages

- *k*-means is relatively scalable and efficient in processing large data sets
- The computational complexity of the algorithm is **O(nkt)**
  - **n:** the total number of objects
  - **k:** the number of clusters
  - **t:** the number of iterations
  - **Normally:** $k<<n$ and $t<<n$

Disadvantage

- Can be applied only when the mean of a cluster is defined
- Users need to specify *k*
- *k*-means is not suitable for discovering clusters with non convex shapes or clusters of very different size
- It is sensitive to noise and outlier data points (can influence the mean value)

There are quite a few variants of the *k*-means method. These can differ in the selection of the initial *k*-means, the calculation of dissimilarity, and the strategies for calculating cluster means. An interesting strategy which often yields good results is to first apply a hierarchical agglomeration algorithm to determine the number of clusters and to find an initial classification, and then use iterative relocation to improve the classification.

Another variant to *k*-means is the ***k*-modes method** which extends the *k*-means paradigm to cluster categorical data by replacing the means of clusters with modes, using new dissimilarity measures to deal with categorical objects, and using a frequency-based method to update modes of clusters. The *k*-means and the *k*-modes methods can be integrated to cluster data with mixed numeric and categorical values, resulting in the ***k*-prototypes** method.

10.8.2 *k*-medoid Clustering

The objective of *k*-medoid clustering is to find a non-overlapping set of clusters such that each cluster has a most representative point, *i.e.*, a point that is most centrally located with respect to some measure, e.g., distance. These representative points are called **medoids.**