e-PGPathshala

Subject : Computer Science

Paper: Machine Learning

Module: Introduction to Clustering

Module No: CS/ML/23

Quadrant I – e-text

Welcome to the e-PG Pathshala Lecture Series on Machine Learning.In this module we will discuss clustering, a very important function of machine learning called clustering. We will discuss the broad categories of clustering with illustrative examples.

Learning Objectives:

The learning objectives of this module are as follows:

- To understand Clustering and its applications
- To understand Hierarchical Clustering
- To understand Agglomerative Clustering with an Illustrative Example

23.1Introduction

Clustering is the most important **unsupervised learning**approach associated with machine learning. It can be viewed as a method for **data exploration**which essentially means looking for patterns or structures in the data space that may be of interest in a collection of unlabeled data. Essentially no classes are associated with data instances a priori as in the case of supervised learning.

Now let us look at simplistic definition of clustering. Clustering can be defined as the method of organizing data instances into groups based on their similarity. In other words a cluster is a collection or group of data instances that are similar to each other and dissimilar to data instances belonging to other clusters.

23.1.1 Natural Grouping -Clustering is subjective

A set of data instances or samples can be grouped differently based on different criteria or features, in other words clustering is subjective. Figure 23.1 shows a set of seven people. They have been grouped into three clusters based on whether they are school employees, they belong to a family or based on the gender. Therefore choosing the attributes or features based on which

clustering is to be carried out is an important aspect of clustering just as it was for classification.

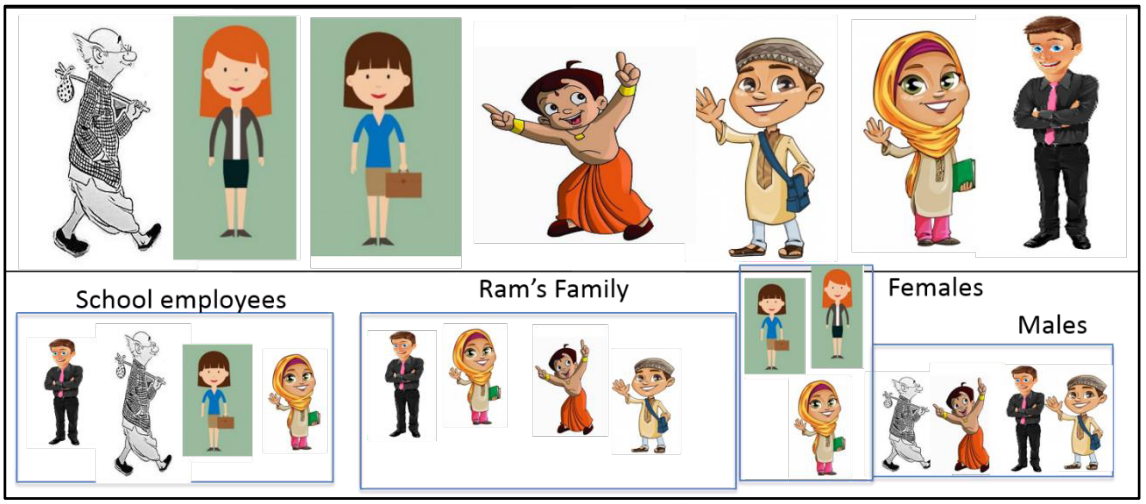


Figure 23.1 Grouping of People based on Different Criteria

23.1.2 Clusters – Distance viewpoint

When we are given a set of instances or examples represented as a set of points, we need to define the notion of distance between these points. We then group the points into some number of clusters, such that members of a cluster are close or similar to each other while members of different clusters are dissimilar or farther apart than members belonging to the same cluster. Figure 23.2 shows a data set that has three natural clusters where the data points group together based on the distance. An outlier is a data point that is isolated from all other data points.

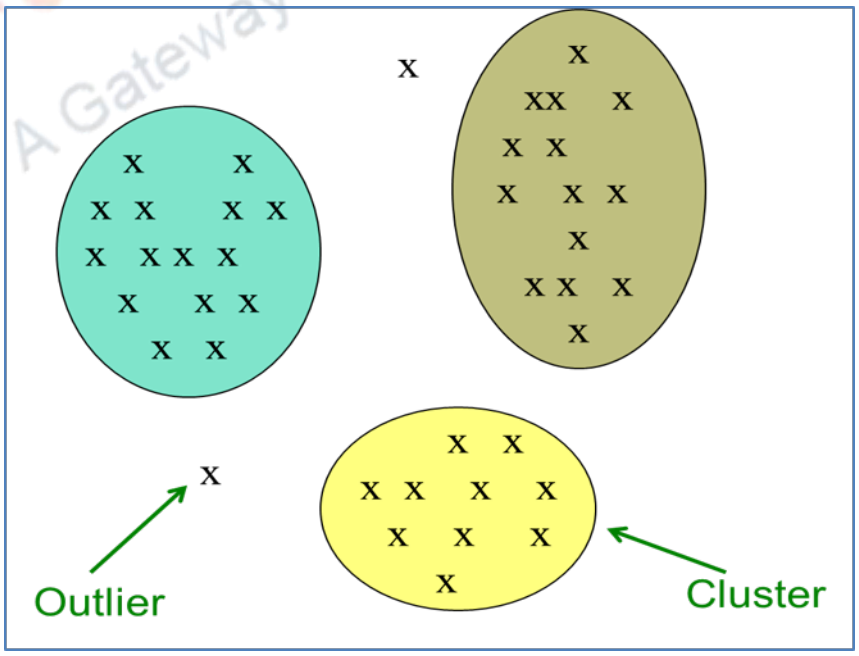


Figure 23.2 Clusters and Outlier

23.2 Applications of Clustering

In the context of machine learning, clustering is one of the functions that has many interesting applications. One of the applications of clustering is for understanding. Understanding is achieved through appropriate grouping. Grouping related documents for browsing, grouping genes and proteins that have similar functionality, or grouping stocks with similar price fluctuations are some examples that help in understanding the commonalities and differences between groups. Another use of clustering is in summarization, in other words we reduce the size of large data sets. Some examples of clustering are shown in Figure 23.3 (a) and (b). The example in Figure 23.3 (a) shows how Google news uses clustering of news articles to help in better presentation of news. In fact by using appropriate features for clustering we can also bring out a personalized presentation. Figure 23.3 (b) shows the use of clustering to show areas clustered based on the amount of precipitation.

(a) (b)

Figure 23.3 Applications of Clustering

Let us see some real-life examples of clustering. When designing T-shirts making them to fit each person is too expensive while one-size-fits-all is not a satisfactory policy. We could group people with similar sizes to design “small”, “medium” and “large” T-shirts. Example 1: groups people of similar sizes together to make “small”, “medium” and “large” T-Shirts.

In today’s world of online marketing, segmenting customers according to their similarities would help in targeted marketing. Features such as previous products bought, effect of discounts etc.. could be used for such marketing.

Another example of clustering is in document clustering where given a collection of text documents, we cancluster them according to their content similarities in order to create a topic hierarchy.

As we can see from the varied applications clustering is one of the most utilized data mining techniques. In image processing it is used to cluster images based on their visual content. In the web scenario it is used to cluster groups of users based on their access patterns on webpages or cluster searchers based on their search behavior or to cluster webpages based on their content and links. In bioinformatics clustering can be used to group similar proteins based on similarity of their chemical structure and/or functionality. It has been used in almost every field, e.g., medicine, psychology, botany, sociology, biology, archeology, marketing, insurance, libraries, etc. Due to the large increase of online documents text clustering is now becoming very important.

23.3 Aspects of clustering

In general clustering deals with high dimensional data. Example of such data include text documents and images. Dealing with such high dimensional data is an important aspect of clustering. Another important aspect that influences the effectiveness of clustering is the choice of distance function or the similarity or dissimilarity measure used. The basic clustering algorithms can be divided into two types namely hierarchicalclustering and partitional clustering. The choice of which type and which algorithm is another important aspect of clustering. We also need to decide on the parameters to evaluate clustering quality. In general the algorithms strive to maximize inter-cluster distance and minimize intra-clusterdistance (Figure 23.4). In other words the quality of clustering depends on the algorithm used, the distance function selected , and the application for which it is to be used.

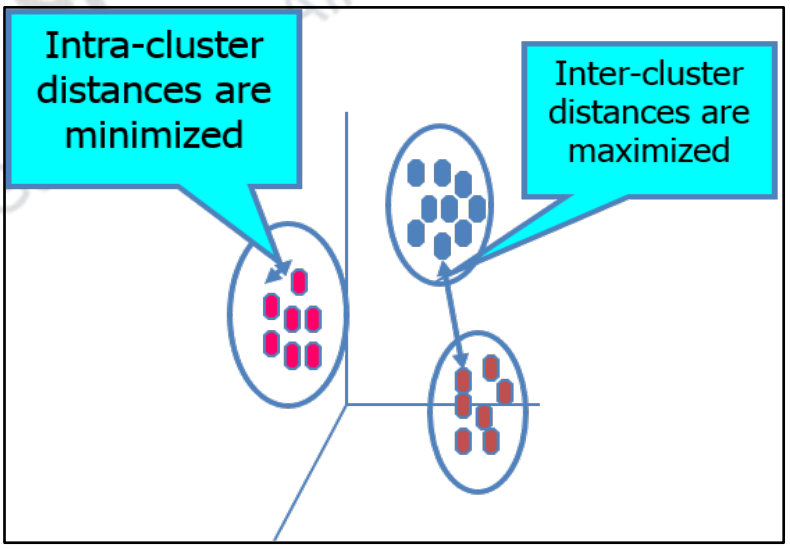


Figure 23.4 Intra and Inter Cluster Distances

23.3.1 High Dimensional Data

Often as we discussed clustering needs to deal with high dimensional data where given a cloud of data points we want to understand its structure.

Clustering needs to capture the structure using some dimensionality techniques and then performing clustering (Figure 23.5).

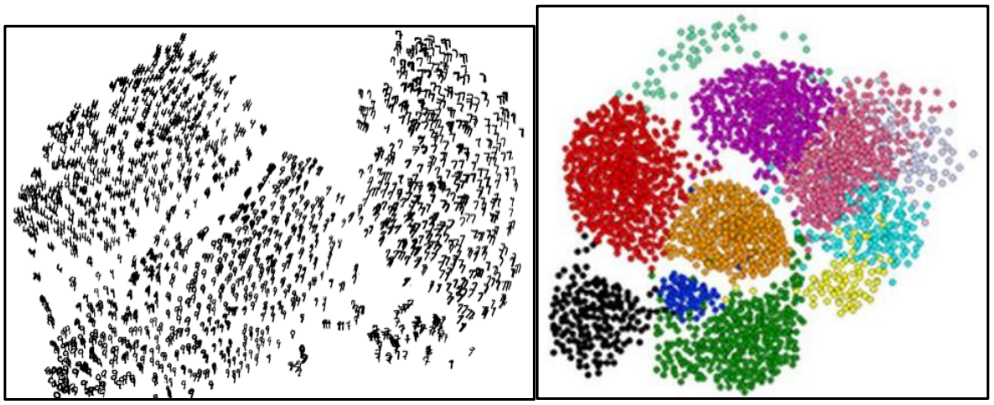


Figure 23.5 Patterns in High Dimensional Data

23.4 Similarity Measures

As we have already discussed clustering is the grouping together of "similar" data. Choosing an appropriate (dis)similarity measure is a critical step in clustering. Similarity measure is often described as the inverse of the distance function that is less the distance more is the similarity. There are many distance functions for the different types of data such as numeric data, nominal data etc. Distance measures can also be defined specifically for different applications.

23.4.1 Distance functions for Numeric Attributes

In general in the case of numeric attributes distance is denoted as*dist*(x_i, x_j), where x_i and x_j are data points. Please note that these data points can be vectors. The most commonly used distance functions in this context are Euclidean distance and Manhattan (city block) distance. These two distance functions are special cases of Minkowski distance.

23.4.1.1 Minkowski Distance

Given below is the Minkowski distance, where h is any positive integer. In other words the distance between two data points x_i and x_j is defined as the h^{th} root of the sum of the difference between them in each dimension taken to the power of h.

$$dist(\mathbf{x}_i, \mathbf{x}_j) = (|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + ..+ |x_{ir} - x_{jr}|^h)^{\frac{1}{h}}$$

23.4.1.2 Euclidean Distance

In the case of Euclidean distance h=2.

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{ir} - x_{jr})^2}$$

In other words Euclidean distance between two data points x_i and x_j is the square root of the sum of the squares of the difference between them in each dimension. This is one of the most commonly used distance function for clustering data points with numerical attributes

23.4.1.3 Manhattan Distance

In the case of Manhattan distance h=1.

$$dist(\mathbf{x}_i, \mathbf{x}_j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \ldots + |x_{ir} - x_{jr}|$$

This distance is sometimes used because of the reduced computational cost as in this case there is no need for calculation of power and power root functions. We need to note this is not a trivial issue since we need to calculate the distance between each and every data point and the number of data points is usually large. Moreover each data point in turn may be represented by a high dimensional vector which further affects the computational cost.

23.4.1.4 Weighted Euclidean distance

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{w_1(x_{i1} - x_{j1})^2 + w_2(x_{i2} - x_{j2})^2 + \ldots + w_r(x_{ir} - x_{jr})^2}$$

Here the sum of the weights $w_1, w_2, \ldots w_r = 1$ In the case of weighted Euclidean distance the difference between the data points in each dimension is weighted. Each dimension in the vector representing the points corresponds to different attributes or features so this essentially means that we can weight each feature according to its importance in defining the cluster.

23.4.1.5 Chebychev distance

This distance is equal to the maximum difference between the values of any one of the attributes and the distance measure is given as:

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \max(|x_{i1} - x_{j1}|, |x_{i2} - x_{j2}|, \ldots, |x_{ir} - x_{jr}|)$$

23.4.2 Distance functions for BinaryAttributes

Binary attributes have two values or states but no ordering relationships, e.g., attribute gender has two values male and female.

In the case of binary attributes normally a confusion matrix is used where the *i*th and *j*th data points are represented asvectors x_i and x_j (Figure 23.6).We use a confusion matrix to introduce the distance functions/measures.

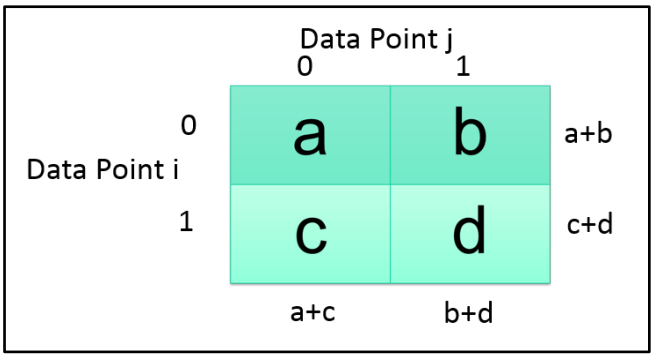


Figure 23.6 Confusion Matrix

In Figure 23.6 “a” corresponds to the number of attributes with the value of 0 for both data pointsx_i and x_j, “b” corresponds to 0 for x_i*and 1 for* x_j, “c” corresponds to 1 for x_i*and 0 for* x_j, while “d” corresponds to the number of attributes with the value of 1 for both data pointsx_i and x_j,

The confusion matrix can be used when the binary attribute is symmetric that is if both states (0 and 1) have equal importance, and carry the same weights. Then the distance function is the proportion of mismatches of their values:

dist(x_i,x_j) = (b+c)/(a+b+c+d)

However sometimes the binary attributes are asymmetric that is one of the states is more important than the other. We assume that state 1 represents the important state in which case the Jaccard measure using the confusion matrix can be defined as:

JDist(x_i,x_j) = (b+c)/(a+b+c)

For text documents normally we usecosine similarity which is a similarity measure not a distance measure.Cosine similarity is a measure of similarity between two vectors obtained by measuring the cosine of the angle between them (Figure 23.7). The similarity between any two given documents d_j and d_k, represented as vectors is given as:

$$sim(d_j, d_k) = \frac{\vec{d}_j \cdot \vec{d}_k}{\left|\vec{d}_j\right|\left|\vec{d}_k\right|} = \frac{\sum_{i=1}^{n} w_{i,j} w_{i,k}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2}\sqrt{\sum_{i=1}^{n} w_{i,k}^2}}$$

In this case w_i is a weight probably based on the frequency of words in the documents.

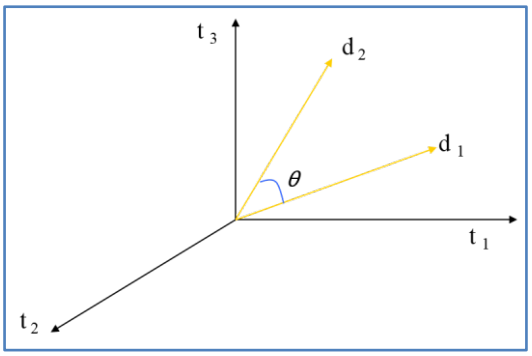


Figure 23.7 The Cosine Angle between vectors

The result of the Cosine function is equal to 1 when the angle is 0, and it is less than 1 when the angle is of any other value. As the angle between the vectors decreases, the cosine value approaches 1, that is the two vectors are closer, and the similarity between the documents increases.

23.5 Methods of Clustering

The basic method of clustering is the hierarchical method which is of two types agglomerative and divisive. Agglomerative clustering is a bottom up method where initially we assume that each data point is by itself a cluster. Then we repeatedly combine the two "nearest" clusters into one. On the other hand divisive clustering is a top down procedure where we start with one cluster and recursively split the clusters until no more division is possible. We normally carry out point assignment where we maintain a set of clusters and allocate points to nearest cluster.

23.6 Hierarchical Clustering

In the hierarchical clustering approach we carry out partitioning of the data set in a sequential manner. The approach constructs nested partitions layer by layer by grouping objects into a tree of clusters (Figure 23.8). In this context there is no need to know the number of clusters in advance. In general the distance matrix is used as the clustering criteria.

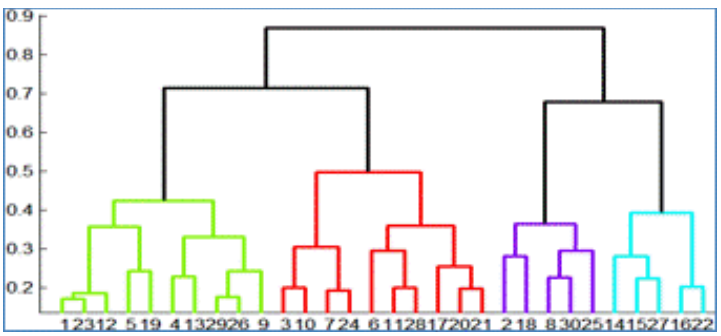


Figure 23.8 Hierarchical Clustering

23.6.1 Types of Hierarchical Clustering

Hierarchical clustering methods can be further classified as either agglomerative or divisive, depending on whether the hierarchical decomposition is formed in a bottom-up (merging) or top-down (splitting) fashion. As we have discussed the hierarchical approach sequentially partitions the data points and constructs a tree of clusters. The following are two sequential clustering strategies for constructing the tree of clusters. The important issues in both cases are cluster distance to be considered and the termination condition to be used.

23.6.1.1 Agglomerative Clustering

Agglomerative clustering is a bottom-up strategy where initially each data pointforms its own (atomic) cluster. We then merge these atomic clusters into larger and larger clusters based on some distance metric. The algorithm terminates when all the data points are in a single cluster or merging is continued until certain termination conditions are satisfied. Most hierarchical clustering methods belong to this category.Agglomerative and divisive clustering on the data set {a, b, c, d ,e } is shown in Figure 23.9.

23.6.1.2 Divisive Clustering

Divisive clustering is a top-down strategywhich does the reverse of agglomerative hierarchical clustering where initially all data points together form a single cluster. Then the cluster is subdivided into smaller and smaller clusters based on some distance metric.It subdivides the clusters into smaller and smaller pieces, until it satisfies certain termination conditions, such as a desired number of cluster or the diameter of each cluster is within a certain threshold

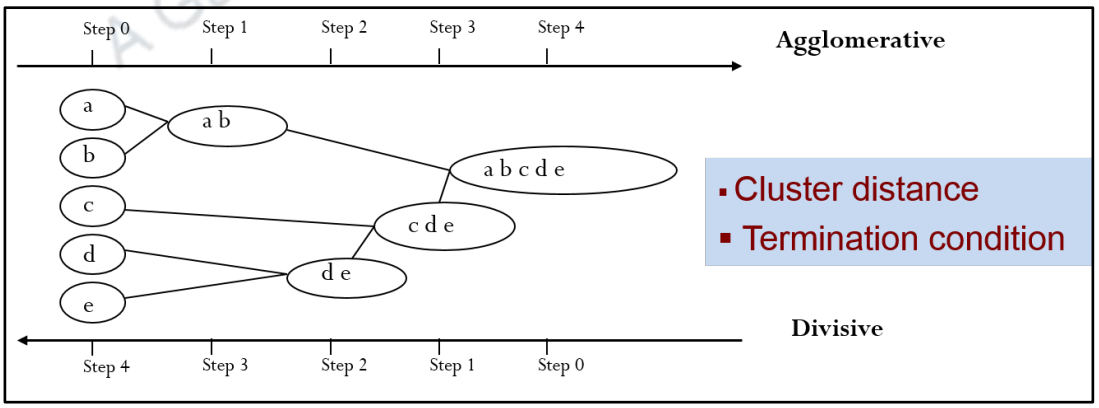


Figure 23.9 Agglomerative and Divisive Hierarchical Clustering

23.7 Hierarchical Clustering: The Algorithm

Hierarchical clustering takes as input a set of points. It then creates a tree in which the points are leaves and the internal nodes reveal the similarity structure of the points.The tree is often called a “dendogram.”The method is summarized below:

- Place all points into their own clusters
- While there is more than one cluster, do
- Merge the closest pair of clusters

The behavior of the algorithm depends on how “closest pair of clusters” is defined

23.8Hierarchical Clustering: Merging Clusters

Now an importantcriteria for merging clusters is the cluster distance. There are three different ways in which this cluster distance can be defined.

Single Link: In this case the distance between two clusters is the distance between the closest points in the clusters (Figure 23.10 (a)). This method is also called neighbor joining. The cluster distance $d(C_i, C_j)$ between the clusters $C_i$and C_j in the case of single link is given as minimum distance between the data points x_{ip} and x_{jq}in the two clusters.

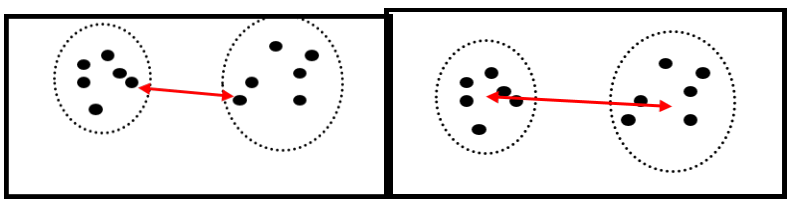
$$d(C_i, C_j) = \min\{d(x_{ip}, x_{jq})\}$$

Average Link: In this case the distance between two clusters is the distance between the cluster centroids (Figure 23.10 (b)).The cluster distance $d(C_i, C_j)$ between the clusters $C_i$and C_j in the case of single link is given as averagedistance between the data points x_{ip} and x_{jq}in the two clusters.

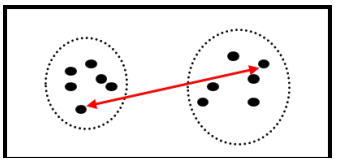
$$d(C_i, C_j) = \text{avg}\{d(x_{ip}, x_{jq})\}$$

Complete Link: In this case the distance between two clusters is the distance between the farthest pair of points (Figure 23.10 (c)). The cluster distance $d(C_i, C_j)$ between the clusters $C_i$and C_j in the case of single link is given as minimum distance between the data points x_{ip} and x_{jq}in the two clusters.

$$d(C_i, C_j) = \max\{d(x_{ip}, x_{jq})\}$$



(a) (b)



(c)

Figure 23.10 Cluster Distance Measures

The distance calculation is also illustrated in Figure 23.11.

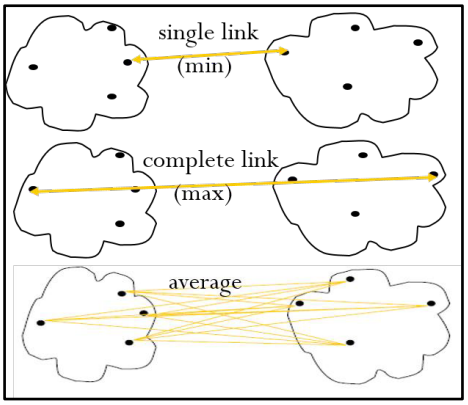


Figure 23.11 Evaluation of Cluster Measures

Example: Given a data set of five objects characterised by a single feature, assume that there are two clusters: C1: {a, b} and C2: {c, d, e}. Assume that we are given the distance matrix. Calculate three cluster distances between C1 and C2.

	a	b	c	d	e
a	0	1	3	4	5
b	1	0	2	3	4
c	3	2	0	1	2
d	4	3	1	0	1
e	5	4	2	1	0

Figure 23.12 Distance Matrix

Single Link

$$\begin{aligned} dist(C_1, C_2) &= \min\{d(a,c), d(a,d), d(a,e), d(b,c), d(b,d), d(b,e)\} \\ &= \min\{3,4,5,2,3,4\} = 2 \end{aligned}$$

Complete Link

$$\begin{aligned} dist(C_1, C_2) &= \max\{d(a,c), d(a,d), d(a,e), d(b,c), d(b,d), d(b,e)\} \\ &= \max\{3,4,5,2,3,4\} = 5 \end{aligned}$$

Average Link

$$\text{dist}(C_1, C_2) = \frac{d(a, c) + d(a, d) + d(a, e) + d(b, c) + d(b, d) + d(b, e)}{6}$$

$$= \frac{3 + 4 + 5 + 2 + 3 + 4}{6} = \frac{21}{6} = 3.5$$

23.9Agglomerative Algorithm

The *Agglomerative* algorithm is a type of hierarchical clustering algorithm that can be carried out in three steps (Figure 23.13):

- Convert object attributes to distance matrix
- Set each object as a cluster (thus if we have *N* objects, we will have *N* clusters at the beginning)
- Repeat until number of cluster is one (or known # of clusters)
    - Merge two closest clusters
    - Update distance matrix

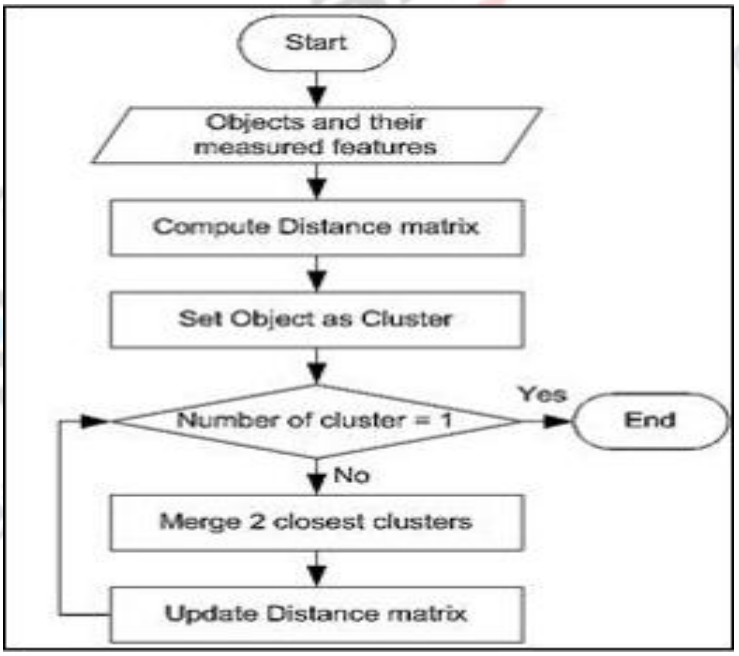


Figure 23.13 Outline of Agglomerative Algorithm

23.10 Clustering Example using Agglomerative Algorithm

Example illustrates single-link clustering in Euclidean space on 6 data points.Let us explain the example. Initially we start with A, B, C, D, E and F. We merge clusters D and F into cluster (D, F) at distance 0.50 Then we merge cluster A and cluster B into (A, B) at distance 0.71 We merge clusters E and (D, F) into ((D, F), E) at distance 1.00. Then we merge clusters ((D, F), E) and C into (((D, F), E), C) at distance 1.41. We merge clusters (((D, F), E), C) and

(A, B) into ((((D, F), E), C), (A, B)) at distance 2.50. The last cluster contains all the data points and hence we conclude the computation.

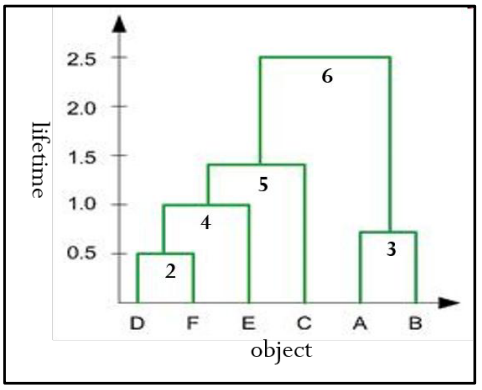


Figure 23.14 Example of Dendrogram TreeRepresentation

23.10 Issues associated with Hierarchical Clustering

As we have already seen the key operation associated with hierarchical clustering is - Repeatedly combining the two nearest clusters

23.10.1 Representation of cluster of many data points

As you merge clusters, how do you represent the “location” of each cluster, to tell which pair of clusters is closest? One solution is the **Euclidean case where** each cluster is associated with a ***centroid***which is the average of its datapoints.

However in the non-Euclidean case, we cannot talk about locations and we cannot talk about average of two points and therefore we need other approaches. In this case we define clusteroid to represent a cluster of data points. Clusteroid is defined as the data point closest to other points. The concept of closeness can be defined in various ways such as smallest maximum distance to other points, smallest average distance to other points or as smallest sum of squares of distances to other points.

Centroid is the avg. of all (data)points in the cluster. This means centroid is an “artificial” point. However clusteroid is an existing (data) point that is “closest” to all other points in the cluster (Figure 23.15).

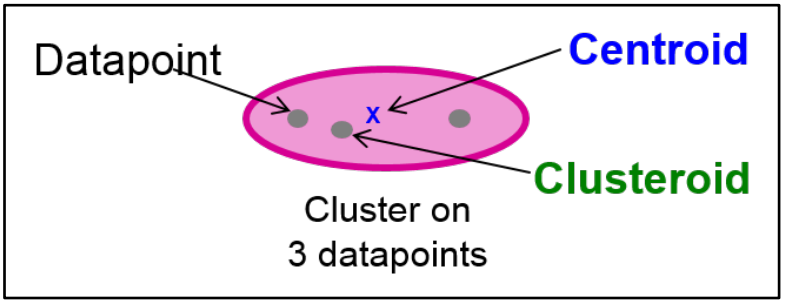


Figure 23.15 Centroid and Clusteroid

23.10.2 Definition of “nearness” of clusters?

When we are dealing with Euclidean space we generally measure cluster distances or nearness of clusters by determining the distances between centroids of the clusters. In the case of non-Euclideancase thedefined clusteroids are treated as centroids to find inter cluster distances.

Nearness can also be defined using other approaches. One such approach could be by using i**ntercluster distance** which is defined as the minimum of the distances between any two points, one from each cluster

$$\min_{c} \sum_{x \in C} d(x, c)^2$$

The third approach is using the concept of “cohesion” of clusters, for example the maximum distance from the clusteroid and we merge clusters whose union is most cohesive.

We will now define the concept of cohesiveness. Cohesion itself can be defined using the diameter of the merged cluster which is the maximum distance between points in the cluster. Another notion of cohesion is the use of the average distancebetween points in the cluster. A density based cohesion takes the diameter or average distanceand divide it by the number of points in the cluster.

23.10.3 Stopping Criteria for combining clusters

Normally we can stop whenwehavekclusters. Another approach is to stopwhenthecohesionofthecluster resulting fromthebestmergerfallsbelowathreshold. Finally we could stopwhenthereisasuddenjumpinthecohesionvalue.

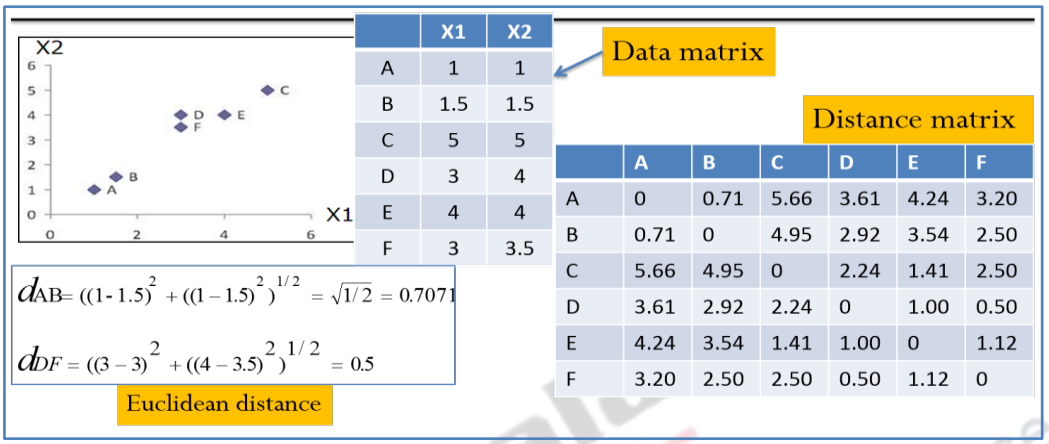
23.11 Example-Agglomerative Algorithm

We will now illustrate the agglomerative algorithm using an example. . Let us assume that we have 6 data points (A, B, C, D, E and F). The data points, the initial data matrix and distance matrix is shown in Figure 23.16 (a). In the first iteration, from the distance matrix we find that the data points D and F are the closest with minimum Euclidean distance (0.50).Now we merge the two data points D and F to form the cluster (D,F) (Figure 23.16 (b)) and update the distance matrix accordingly (Figure 23.16 (c)).

In the second iteration, in the updated matrix, we find that the data points A and B are the closest with minimum Euclidean distance (0.71). Now we merge the two data points A and B to form the cluster (A,B) (Figure 23.16 (d)). In iteration

3, we merge cluster (D,F) and data point E to form the new data point ((D,F),E) and update the distance matrix accordingly (Figure 23.16 (e)).

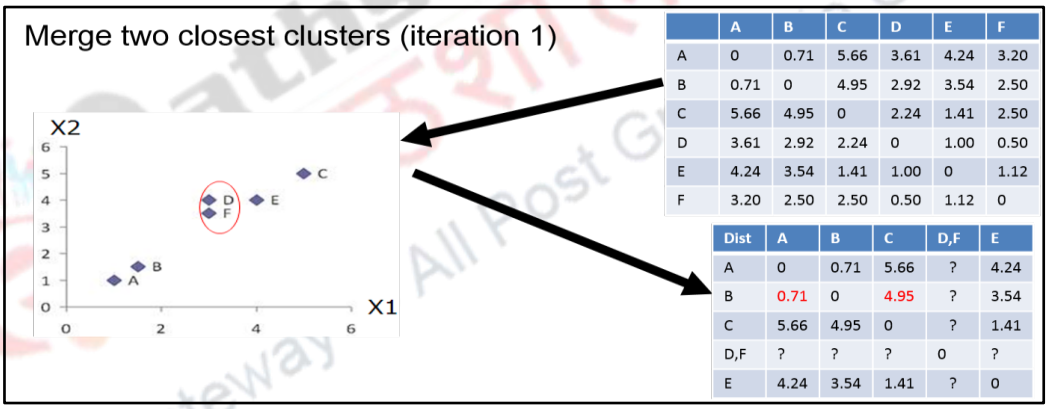
In the 4th iteration, we merge two clusters ((D,F),E) and C to form the new cluster (((D,F),E),C) and update the distance matrix accordingly (Figure 23.16 (f)). The final result is shown in (Figure 23.16 (g)).



	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5

	A	B	C	D	E	F
A	0	0.71	5.66	3.61	4.24	3.20
B	0.71	0	4.95	2.92	3.54	2.50
C	5.66	4.95	0	2.24	1.41	2.50
D	3.61	2.92	2.24	0	1.00	0.50
E	4.24	3.54	1.41	1.00	0	1.12
F	3.20	2.50	2.50	0.50	1.12	0

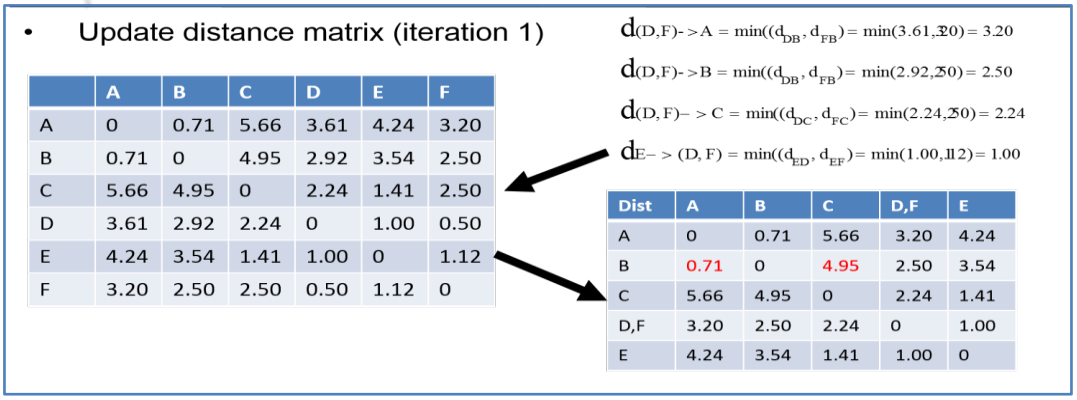
(a)



	A	B	C	D	E	F
A	0	0.71	5.66	3.61	4.24	3.20
B	0.71	0	4.95	2.92	3.54	2.50
C	5.66	4.95	0	2.24	1.41	2.50
D	3.61	2.92	2.24	0	1.00	0.50
E	4.24	3.54	1.41	1.00	0	1.12
F	3.20	2.50	2.50	0.50	1.12	0

Dist	A	B	C	D,F	E
A	0	0.71	5.66	?	4.24
B	0.71	0	4.95	?	3.54
C	5.66	4.95	0	?	1.41
D,F	?	?	?	0	?
E	4.24	3.54	1.41	?	0

(b)

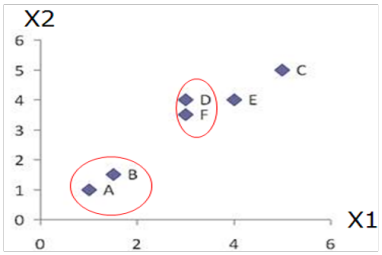


	A	B	C	D	E	F
A	0	0.71	5.66	3.61	4.24	3.20
B	0.71	0	4.95	2.92	3.54	2.50
C	5.66	4.95	0	2.24	1.41	2.50
D	3.61	2.92	2.24	0	1.00	0.50
E	4.24	3.54	1.41	1.00	0	1.12
F	3.20	2.50	2.50	0.50	1.12	0

Dist	A	B	C	D,F	E
A	0	0.71	5.66	3.20	4.24
B	0.71	0	4.95	2.50	3.54
C	5.66	4.95	0	2.24	1.41
D,F	3.20	2.50	2.24	0	1.00
E	4.24	3.54	1.41	1.00	0

(c)

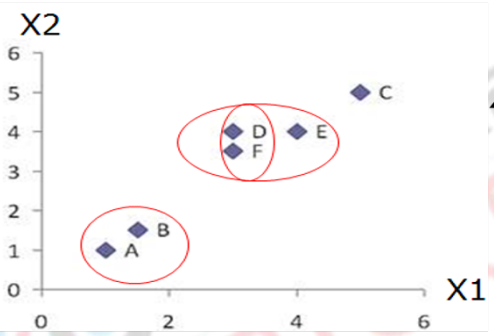
- Merge two closest clusters (iteration 2)



Dist	A	B	C	D,F	E
A	0	0.71	5.66	3.20	4.24
B	0.71	0	4.95	2.50	3.54
C	5.66	4.95	0	2.24	1.41
D,F	3.20	2.50	2.24	0	1.00
E	4.24	3.54	1.41	1.00	0

Dist	A,B	C	D,F	E
A,B	0	?	?	?
C	?	0	2.24	1.41
D,F	?	2.24	0	1.00
E	?	1.41	1.00	0

(d)

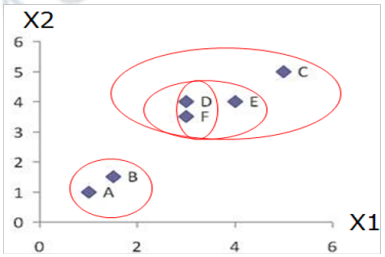


Dist	A,B	C	D,F	E
A,B	0	4.95	2.50	3.54
C	4.95	0	2.24	1.41
D,F	2.50	2.24	0	1.00
E	3.54	1.41	1.00	0

Dist	A,B	C	(D,F),E
A,B	0	4.95	2.50
C	4.95	0	1.41
(D,F),E	2.50	1.41	0

(e)

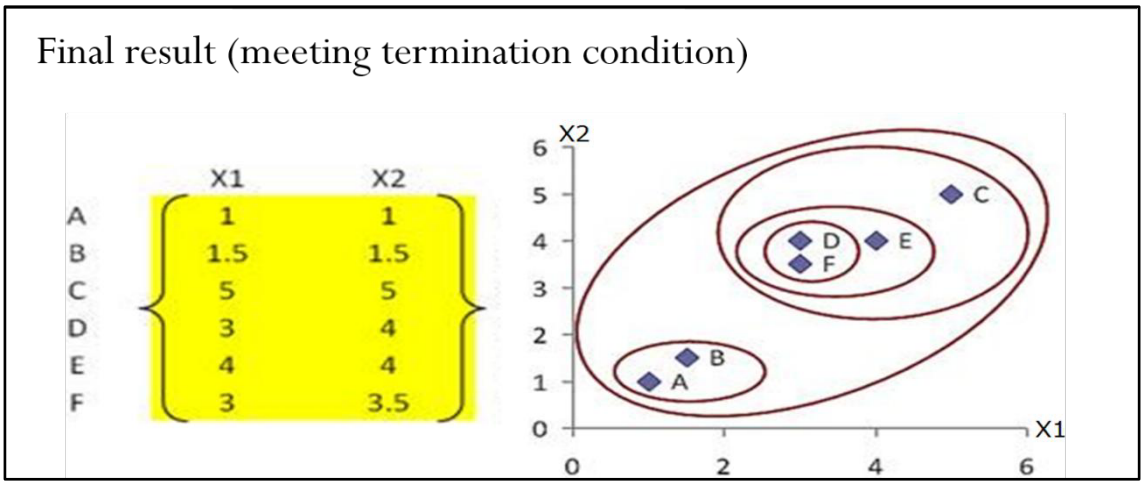
- Merge two closest clusters/update distance matrix (iteration 4)



Dist	A,B	C	(D,F),E
A,B	0	4.95	2.50
C	4.95	0	1.41
(D,F),E	2.50	1.41	0

Dist	A,B	(D,F),E
A,B	0	2.50
((D,F),E) ,C	2.50	0

(f)



(g)

Figure 23.16 Example of Agglomerative Clustering

Summary

- Explained Clustering and its applications
- Discussed Hierarchical Clustering along with different distance measures
- Explained Agglomerative Clustering with an Illustrative Example