CLUSTERING

Clustering is similar to classification, in that data are grouped. However, unlike the classification, the groups are not predefined. Instead the grouping is accomplished by finding the similarities between data according to characteristics found in the actual data. The groups are called clusters.

So, **cluster** is a collection of data objects, in which the objects similar to one another within the same cluster and dissimilar to the objects in other clusters

Cluster analysis is the process of finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters.

Clustering is **unsupervised classification**: no predefined classes

Typical applications

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Land use: Identification of areas of similar land use in an earth observation database
- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults

Definition

- Clustering: Given a database D = {t1, t2, .., tn}, a distance measure dis(ti, tj) defined between any two objects ti and tj, and an integer value k, the clustering problem is to define a mapping f: D $\rightarrow$ {1, …, k} where each ti is assigned to one cluster K_j, $1 \leq j \leq k$.
  Here 'k' is the number of clusters.

Classification of clustering techniques:
There are two main approaches to clustering:

a. Hierarchical Clustering.
b. Partitioning Clustering

Besides, clustering algorithm differ among the different types of attributes, numerical and categorical, in accuracy of handle disk-resident data.

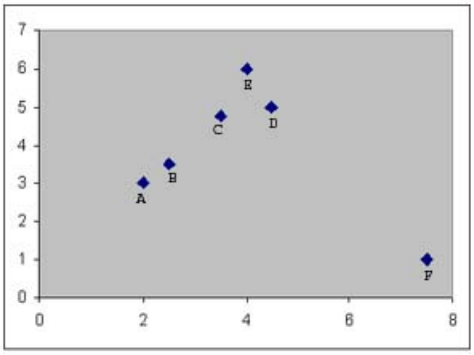
Hierarchical Vs Partitioning:
The partition clustering techniques partition the database into a predefined number of clusters. That is, only one set of cluster is created. They attempt to determine the k partitions that optimize the certain criterion function. The partition clustering algorithms are of two types: k-means algorithm and k-medoid algorithm. Another type of algorithm is the k-mode algorithm.

With hierarchal clustering, a nested set of cluster is created. Each level in the hierarchy has the separate set of clusters. At the lowest level, each item is in its own unique cluster. At the highest level, all items belong to the same cluster. With hierarchical clustering, the desired number of clusters is not input.
The hierarchical clustering are of two types

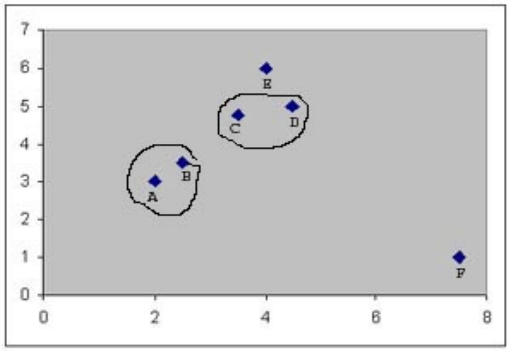
- Agglomerative:
  Agglomerative starts with as many clusters as there are records, with each cluster having only one record. Then pairs of clusters are successively merged until the number of clusters reduces to k. at each stage, the pair of clusters are merged which are nearest to each other. If the merging is continued, it terminates in the hierarchy of clusters which is built with just a single cluster containing all the records.

- Divisive:
  Divisive algorithm takes the opposite approach from the agglomerative techniques. These starts with all the records in one cluster, and then try to split tat clusters into smaller pieces.

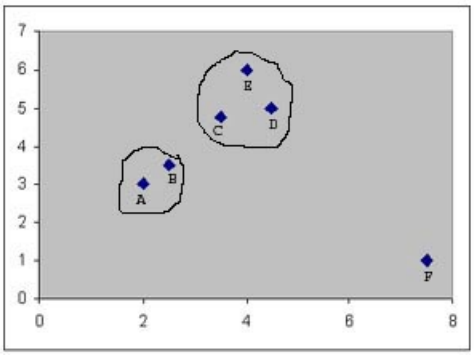
Example of Hierarchical Clustering
Consider we need to cluster six elements {A,B,C,D,E,F}.



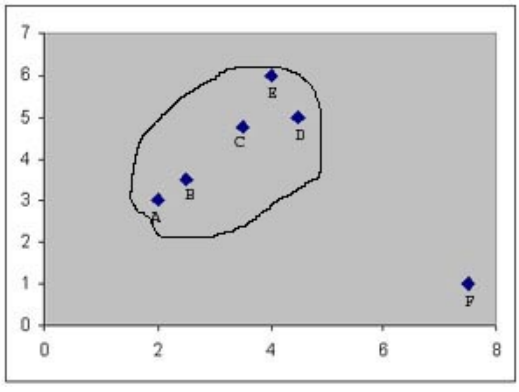
a. six clusters



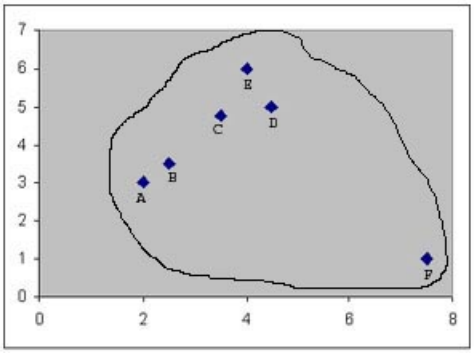
b. four clusters



c. Three Clusters

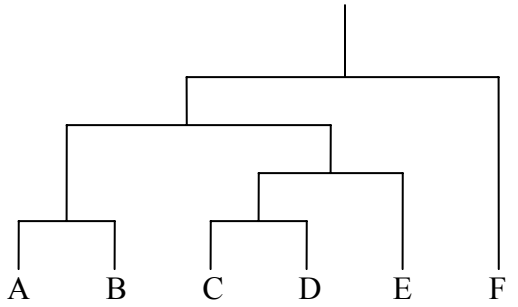


d. Two Clusters



e. One Clusters

In the above fig, in part (a) each cluster is viewed to consists of a single element. Part (b) illustrates four cluster. Here there are two sets of two-element clusters. These clusters are formed at this level because these two elemenrs are close to each other than any of other elements. Part (c) shows the new cluster formed by adding a close element to one of the two-element clusters. In part (d), the two-element and three-element clusters are merged to give a five-element clusters. This is done because these two clusters are closer to each other than to the remote element cluster. At the last stage, part (e) all the six elements are merged. The corresponding dendrogram is shown below



Dendrogram for the above example

Partitioning Algorithms: Basic Concept

- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters, such that we have the minimum sum of squared distance

- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means*: Each cluster is represented by the center of the cluster
  - *k-medoids* or PAM (Partition around medoids): Each cluster is represented by one of the objects in the cluster

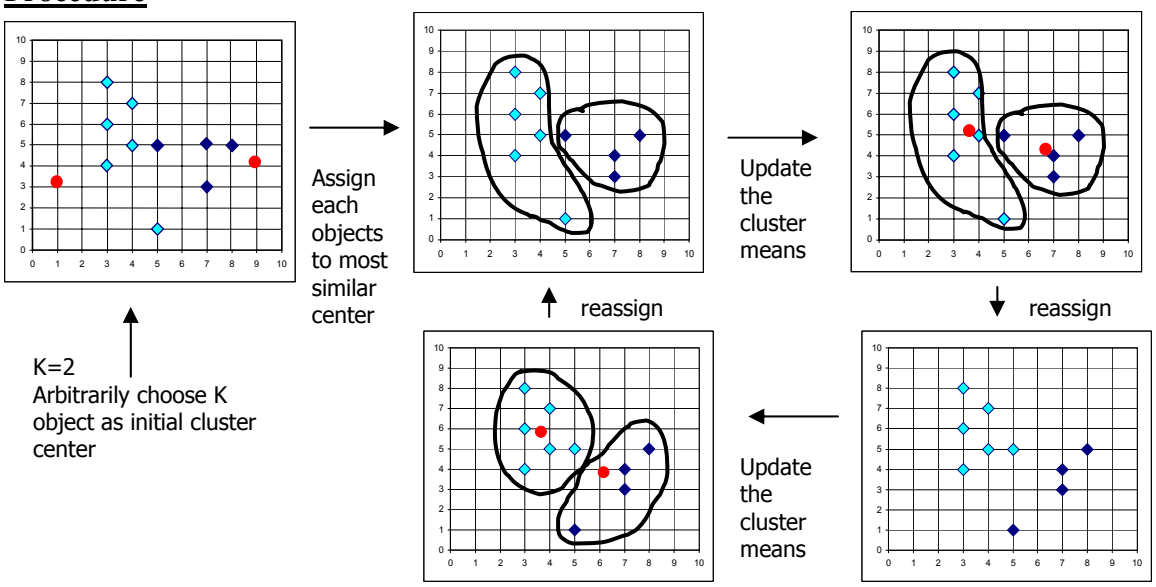
The *K-Means* Clustering Algorithm

1) choose *k*, number of clusters to be determined
2) Choose k objects randomly as the initial cluster centers
3) repeat

a) Assign each object to their closest cluster center
   i) Using Euclidean distance
b) Compute new cluster centers
   i) Calculate mean points

4) until
   a) No change in cluster centers or
   b) No object change its cluster

Procedure



Example

Consider the following instances [in the two-dimensional form]

Instance	X	Y
1	1.0	1.5
2	1.0	4.5
3	2.0	1.5
4	2.0	3.5
5	3.0	2.5
6	5.0	6.1

1. If the objects are to be partitioned into 2 clusters then take K=2.
2. Next , chose two points at random representing initial cluster centers:
   Object 1 and 3 are chosen as cluster centers; i.e.
   C1:= (1.0, 1.5) and C2:= (2.0, 1.5) are chosen as the initial centroid
3. Euclidean distance between point i and j
   $D(i - j)=((Xi – Xj)^2 + (Yi – Yj)^2)^{1/2}$
   - Initial cluster centers **C1:(1.0,1.5) C2:(2.0,1.5)**
     - For object '1'
       D(C1 – 1) = 0.00 D(C2 –1) = 1.00
       Since D(C1-1)<D(C2-1) the object '1' falls in cluster **C1**

- For object '2'
  D(C1 – 2) = 3.00 D(C2 –2) = 3.16
  Since D(C1-2)<D(C2-2) the object '2' falls in cluster **C1**

- For object '3'
  D(C1 – 3) = 1.00 D(C2 –3) = 0.00
  Since D(C2-3)<D(C1-3) the object '3' falls in cluster **C2**

- For object '4'
  D(C1 – 4) = 2.24 D(C2 –4) = 2.00
  Since D(C2-4)<D(C1-4) the object '4' falls in cluster **C2**

- For object '5'
  D(C1 – 5) = 2.24 D(C2 –5) = 1.41
  Since D(C2-5)<D(C1-5) the object '5' falls in cluster **C2**

- For object '6'
  D(C1 – 6) = 6.02 D(C2 –6) = 5.41
  Since D(C2-6)<D(C1-6) the object '6' falls in cluster **C2**

- Then the cluster C1 and C2 contain the following objects respectively
  C1 : {1,2}
  C2 : {3.4.5.6}

4. Recomputing cluster centers [taking the mean]
   a. for C1:
      $X_{C1} = (1.0+1.0)/2 = 1.0$
      $Y_{C1} = (1.5+4.5)/2 = 3.0$
   b. For C2:
      $X_{C2} = (2.0+2.0+3.0+5.0)/4 = 3.0$
      $Y_{C2} = (1.5+3.5+2.5+6.0)/4 = 3.375$

   Thus the new cluster centers are C1(1.0,3.0) and C2(3.0,3.375)

5) As the cluster centers have changed the algorithm performs another iteration

- New cluster centers C1(1.0,3.0) and C2(3.0,3.375)
  - D(C1 – 1) = 1.50 D(C2 –1) = 2.74
    Object '1' falls in C1

  - D(C1 – 2) = 1.50 D(C2 –2) = 2.29
    Object '2' falls in C1

  - D(C1 – 3) = 1.80 D(C2 –3) = 2.13
    Object '3' falls in C1

- D(C1 – 4) = 1.12 D(C2 –4) = 1.01
  Object '4' falls in C2

- D(C1 – 5) = 2.06 D(C2 –5) = 0.88
  Object '5' will be in C2

- D(C1 – 6) = 5.00 D(C2 –6) = 3.30
  Object '6' will be in C2

- Then the cluster C1 and C2 contain the following objects respectively
  C1 : {1,2.3}
  C2 : {4.5.6}

6. computing new cluster centers
   - For C1:
     X_{C1} = (1.0+1.0+2.0)/3 = 1.33
     Y_{C1} = (1.5+4.5+1.5)/3 = 2.50
   - For C2:
     X_{C2} = (2.0+3.0+5.0)/3 = 3.33
     Y_{C2} = (3.5+2.5+6.0)/3 = 4.00
- Thus the new cluster centers are C1(1.33,2.50) and C2(3.33,4.3.00)
- As the cluster centers have changed the algorithm performs another iteration

[repeat the process until there is no change in cluster centers or no object change its cluster]

Comments:

- each initial cluster centers may end up with different final cluster configuration
- Finds local optimum but not necessarily the global optimum
  - Based on sum of squared error differences
  - Between objects and their cluster centers
- Choose a terminating criterion such as
  - Execute K-Means algorithm until satisfying the condition

<u>Weaknesses of K-Means Algorithm</u>

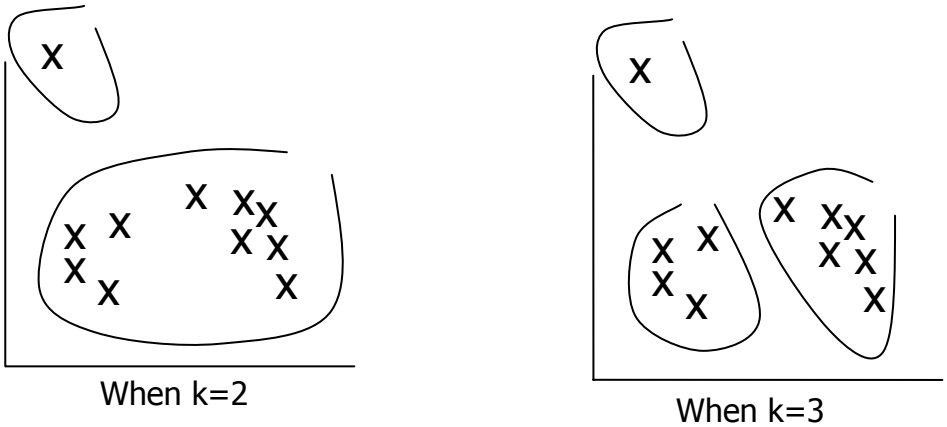
- Applicable only when *mean* is defined, then what about categorical data?
- Need to specify *K,* the *number* of clusters, in advance
  - run the algorithm with different K values
- Unable to handle noisy data and *outliers*
- Not suitable to discover clusters with *non-convex shapes*
  - Works best when clusters are of approximately of equal size

Outliers

Outliers are the points with values much different from those of the remaining set of data. Outliers may represent the error in the data or could be correct data values that are simply much different from the remaining data.

The outliers can be viewed as the solitary clusters. However, if a clustering algorithm attempts to find the larger clusters, these outliers will be force to be placed in some cluster. This process may result the creation of poor clusters by combining two existing cluster and leaving the outlier in its own cluster.

The following example depicts the problem:



When k=2 the two natural clusters are not captured

Fig. The presence of cluster