Chapter – 6 :: Cluster Analysis: Basic Concepts and Algorithms

Introduction: Cluster analysis divides data into groups (clusters) that are meaningful, useful, of both. If meaningful groups are the goal, then the clusters should capture the natural structure of the data. That is, conceptually meaningful groups of objects that share common characteristics, play an important role in how people analyze and desire the world. For example, young children can quickly label the objects in a photograph as buildings, vehicles, people, animals, plants, etc. In this case, clusters are potential classes and cluster analysis is the study of techniques for automatically finding classes.

6.1 What is Clustering in Data Mining?

- **Clustering** is a group of objects and that objects in a group (cluster) are similar to one another and different from (or unrelated to) the objects in other groups.
- **Cluster:** is a collection of data objects that are "similar" to one another and thus can be treated collectively as one group.
- **Cluster analysis** is a function of data mining that may be used to form the groups of data objects (clusters) based only on information which found in the data that describes the objects and their relationships.
- Cluster analysis can also used as a preprocessing step for characterization attribute subset selection and classification algorithms.
- Clustering can also be used to help in classifying documents on web for detecting information.
- In many applications, a cluster is not well defined. To better understand the difficulty of deciding what constitutes a cluster.
- Consider the following figure that shows Four set of points. The shapes of the markers indicate cluster membership.

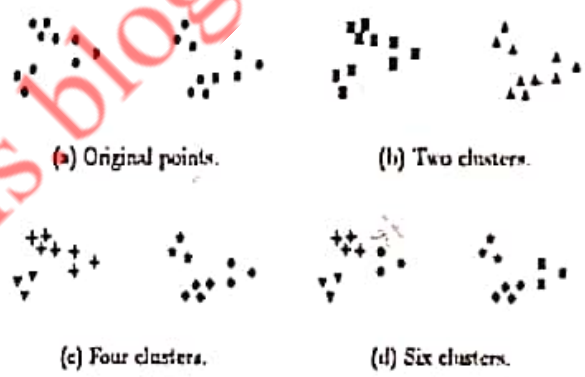


Figure 8.1. Different ways of clustering the same set of points.

6.2 Types of Clustering: (Explain different types of clustering methods) – HPMGD

The types of Clustering are grouped into the following categories.

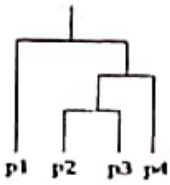
1. Hierarchical method
2. Partitioning method
3. Model based method
4. Grid-based method
5. Density-based method.

1. Hierarchical Clustering: A hierarchical clustering method works by grouping data objects into a tree form of clusters. Hierarchical methods can be categorized as agglomerative clustering and divisive hierarchical clustering.

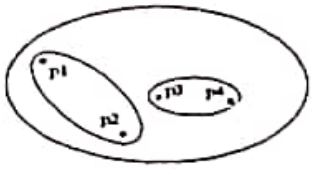
- These two methods construct the clusters by recursively partitioning the instances in either a top-down or bottom-up fashion.

1. Agglomerative hierarchical clustering: — each object initially represents a cluster of its own. Then clusters are successively merged until the desired cluster structure is obtained.
2. Divisive hierarchical clustering: All objects initially belong to one cluster. Then the cluster is divided into sub-clusters, which are successively divided into their own sub-clusters. This process continues until the desired cluster structure is obtained.

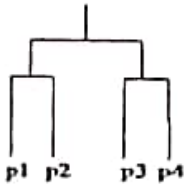
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

Partitioning Methods: Partitioning methods relocate instances by moving them from one cluster to another, starting from an initial partitioning. A relocation method iteratively relocates points between the k clusters.

Model-based Methods: Model-based methods build a cluster on the basis of a model. A density function is built by model-based algorithms to locate clusters. The density function defines the spatial distribution of the data points.

A model-based algorithm is based on standard statistic and taking into account noise or outliers can automatically find the number of clusters.

The most frequently used induction methods are decision trees and neural networks.

1 Decision Trees: In decision trees, the data is represented by a hierarchical tree, where each leaf refers to a concept and contains a probabilistic description of that concept. Several algorithms produce classification trees for representing the unlabelled data. The most well-known algorithms are: COBWEB and CLASSIT.

2 Neural Networks: This type of algorithm represents each cluster by a neuron or "prototype". The input data is also represented by neurons, which are connected to the prototype neurons. Each such connection has a weight, which is learned adaptively during learning. A very popular neural algorithm for clustering is the self-organizing map (SOM). This algorithm constructs a single-layered network.

The SOM algorithm is successfully used for vector quantization and speech recognition. It is useful for visualizing high-dimensional data in 2D or 3D space. However, it is sensitive to the initial selection of weight vector, as well as to its different parameters, such as the learning rate and neighborhood radius.

4. Grid-based Methods: Grid-based method consists of a grid structure formed by qualifying the objects space into a finite number of cells. It is an approach of representing data objects using a multi-resolution gird data structure on which all of the clustering operations are performed. The examples of grid-based approach are,

1. Wave cluster: A wavelet transform approach is used to cluster objects.
2. String: It is grid-based multi-resolution technique that stores statistical information in rectangular cells.
3. CLIQUE: It is used for clustering high-dimensional data.

5. Density-based Methods: The aim of this method is to identify the clusters and their distribution parameters. These methods are designed for discovering clusters of arbitrary shape.

The DBSCAN algorithm (density-based spatial clustering of applications with noise) discovers clusters of arbitrary shapes and is efficient for large spatial databases. The algorithm searches for clusters by searching the neighborhood of each object in the database and checks if it contains more than the minimum number of objects.

AUTOCLASS is a widely-used algorithm that covers a broad variety of distributions, including Gaussian, Bernoulli, Poisson, and log-normal distributions. Other well-known density-based methods include: SNOB and MCLUST.

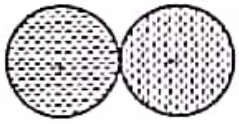
6.3 Types of Clusters: The different types of clusters are:
1. Well separated, 2. Prototype based 3. Graph based, 4.Density based, 5. Shared property.

(a) Well-separated clusters. Each point is closer to all of the points in its cluster than to any point in another cluster.

1. Well separated: A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.

Figure(a) gives an example of well separated cluster that consists of two groups of points in two-dimensional space. The distance b/w any two clusters is larger than the distance b/w the objects within the clusters.

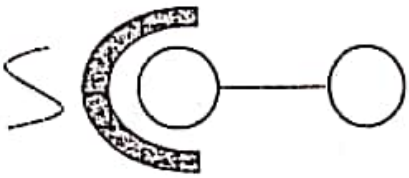
2.Prototype-Based: A cluster is a set of objects such that an object in a cluster is closer (more similar) to the "center" of a cluster, than to the center of any other cluster.



(b) Center-based clusters. Each point is closer to the center of its cluster than to the center of any other cluster.

- The center of a cluster is often a centroid, the average of all the points in the cluster, or a medoid, the most "representative" point of a cluster.
- For data with continuous attributes, the prototypes of a cluster are often a centroid, i.e. the average (mean) of all the points in the cluster. The figure (b) shows an example of center-based clusters.

3.Graph-Based: The graph based cluster represents the data in the form of a graph where in the nodes indicate objects and links indicate objects. Such clusters are claimed as connected component.

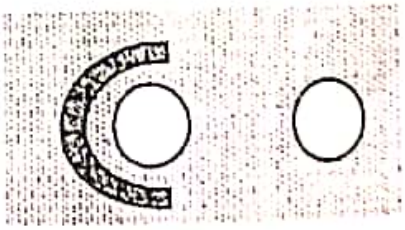


(c) Contiguity-based clusters. Each point is closer to at least one point in its cluster than to any point in another cluster.

- A contiguity based clusters serves as an example of graph based clusters where the two objects connectivity entirely depends on the distance b/w them. i.e. if two objects are close to each other they are said to be connected. And each individual objects in this cluster is closer to another objects within the cluster and a distance from different cluster.

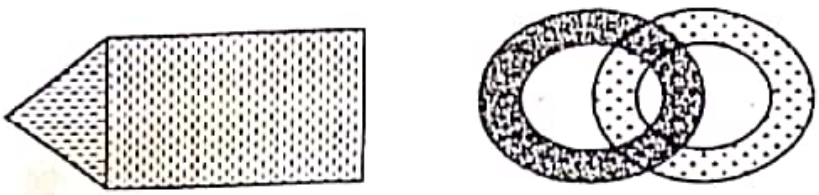
The figure shows cluster with tow-dimensional points and it defines a cluster that useful when clusters are irregular or intertwined, but can have trouble when noise is present.

4. Density Based: A cluster is a dense region of objects that is surrounded by a region of low density. Figure (d) shows that density-based clusters for data created by adding noise to the data of figure (c).
Here two clusters are not combined because the link prevailing b/w them is transformed into noise.



(d) Density-based clusters. Clusters are regions of high density separated by regions of low density.

5.Shared property(Conceptual clusters): Clusters that share some common property or represent a particular concept. For example, objects in a center-based share the property that they are all closest to the same centriod or medoid. Figure shows (e) that a triangular area is adjacent to a rectangular one, and there are two intertwined circles.



(e) Conceptual clusters. Points in a cluster share some general property that derives from the entire set of points. (Points in the intersection of the circles belong to both.)

6.4.K-means: There are many clustering techniques that were used to find objects which are closer to prototype. The most important techniques are K-means and K-medoid.

- The K-means defines a prototype in terms of a centroid which can be find the mean value of a group of points and it applied to objects in a continuous n-dimensional space.
- The K-medoid defines a prototype to terms of a medoid which is the most representative points for a group of points and it can be applied to a wide range of data but it requires pair of objects.

Definition: *k*-means clustering is a method of vector quantization originally from signal processing that is popular for cluster analysis in data mining. K-means clustering aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells.

6.4.1.The Basic K-means Algorithm: The K-means technique is an iterative clustering algorithm in which objects are moved among sets of clusters until the desired set is achieved. It is the most popular and commonly used method.

- K-means algorithm is used to find centriod in a cluster. The centroid is represented by symbol '+'. →During the process, each point is assigned to the closest centroid and each collection of points assigned to a centroid is a cluster.
- The centroid of each cluster is then updated based on the points assigned to the cluster.
- The assignment and update are repeated until no point changes in cluster.

K-means is described by algorithm and its operation is illustrated in fig. The operation of K-means start from three centrioids, the final clusters are found in four assignment update steps.

Algorithm 8.1 Basic K-means algorithm.

1: Select K points as initial centroids.
2: repeat
3: Form K clusters by assigning each point to its closest centroid.
4: Recompute the centroid of each cluster.
5: until Centroids do not change.

- The first step assigns the points to the initial centroid. After points are assigned to a centroid, the centroid is then updated.
- In the second step, points are assigned to the updated centroid and the cetroids are updated again.
- The second step is repeated until centroids do not change.
- Finally, the symbol '+' is moved to the center point of each cluster and formed as centriod with symbol '+'

- This is shown in following figure that shows the operation of each iteration.

(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4.

- When the K-means algorithm terminates in figure (d), the no more changes occur and the centriods identifying the natural grouping of points.

6.4.2: Time Space Complexity of K-means:

- The space complexity of K-means are calculated based on the data points and centroids. The required storage of centroid is O((m + K)n).
  - ✓ Where 'm' is number of objects, and 'n' is the number of attributes.
- The time Complexity is O(n * K * I * d)

n = number of points, K = number of clusters,
I = number of iterations, d = number of attributes

6.4.3. Limitations of K-means:

- K-means has problems when clusters are of differing from
  - Sizes
  - Densities
  - Non-globular shapes
- K-means has problems when the data contains outliers.

6.5. K-means: Additional Issues: The issues encountered in *k*-means are 1. Handling Empty Clusters, 2. Outliers.

1. Handling Empty Clusters: In the cluster if no points are allocated to find centroid during the assignment step is called the *squared error (SE)*. If this error occur, then need to replacement the centroid, otherwise the squared error will be eliminated by split the cluster and reduce the overall SSE of the clustering. If there are several empty clusters, then this process can be repeated several times.

2. Outliers: The outliers may be of particular interest, such as in the case of fraud detection, where outliers may indicate fraudulent activity. Thus, outlier detection and analysis is an interesting data mining task, referred to as outlier mining or outlier analysis.

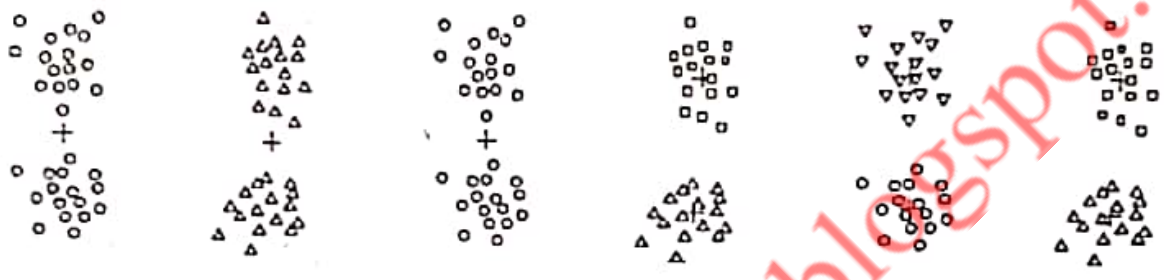
When the squared error is occur in the cluster with outlier (i.e. noise), the resulting cluster centriods may not be as representative as the SSE.

6.6: Bisecting K-means: The Bisecting K-means algorithm is extension of the basic K-means algorithm. The idea is to obtain K clusters set of points are split into two clusters and among which one cluster is chosen to split, and so on. The details of bisecting K-means are given by Algorithm.

Algorithm 3 Bisecting K-means Algorithm.

1: Initialize the list of clusters to contain the cluster containing all points.
2: repeat
3: Select a cluster from the list of clusters
4: for i = 1 to *number_of_iterations* do
5: Bisect the selected cluster using basic K-means
6: end for
7: Add the two clusters from the bisection with the lowest SSE to the list of clusters.
8: until Until the list of clusters contains K clusters

Example: Bisecting K-means and Initialization: To illustrate that bisecting K-means is less susceptible to initialization problem. This is shown in fig.



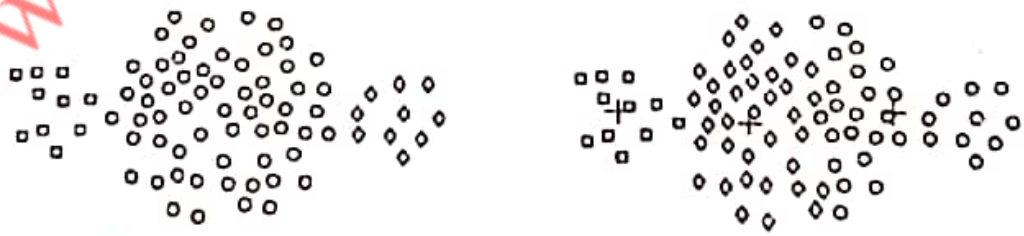
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3.

Figure 8.8. Bisecting K-means on the four clusters example.

- In the figure.
  - The iteration1 found two pairs of clusters,
  - The teration2, the rightmost pair of clusters is split and
  - The iteration 3, the leftmost pair of clusters is split.
- Bisecting K-means has less trouble with initialization because it performs several trial bisections and takes the one with the lowest SSE and there are only two centriods at each step.

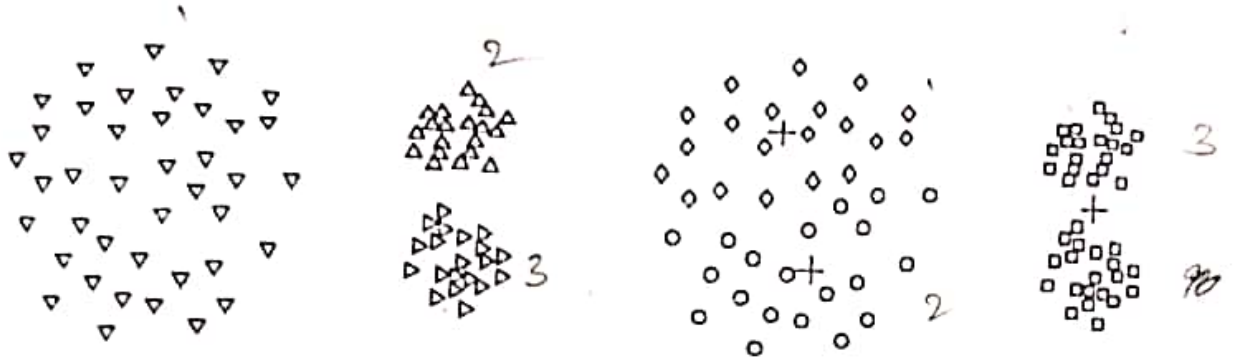
6.7: K-means and Different Types of Clusters:

K-means have a number of limitations with respect to finding different types of clusters. In particular, K-means has difficulty to detect the "natural" clusters, when clusters have non-spherical shapes or widely different sizes or densities. This is illustrated by following figures: (8.9, 8.10, 8.11, 8.12)



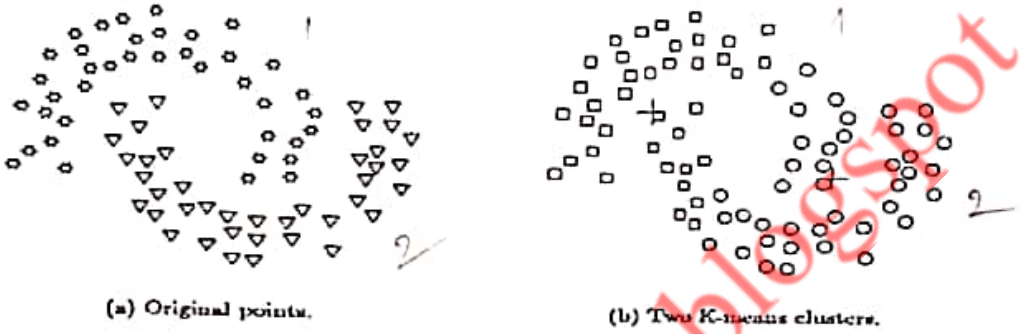
(a) Original points. (b) Three K-means clusters.

Figure 8.9. K-means with clusters of different size.



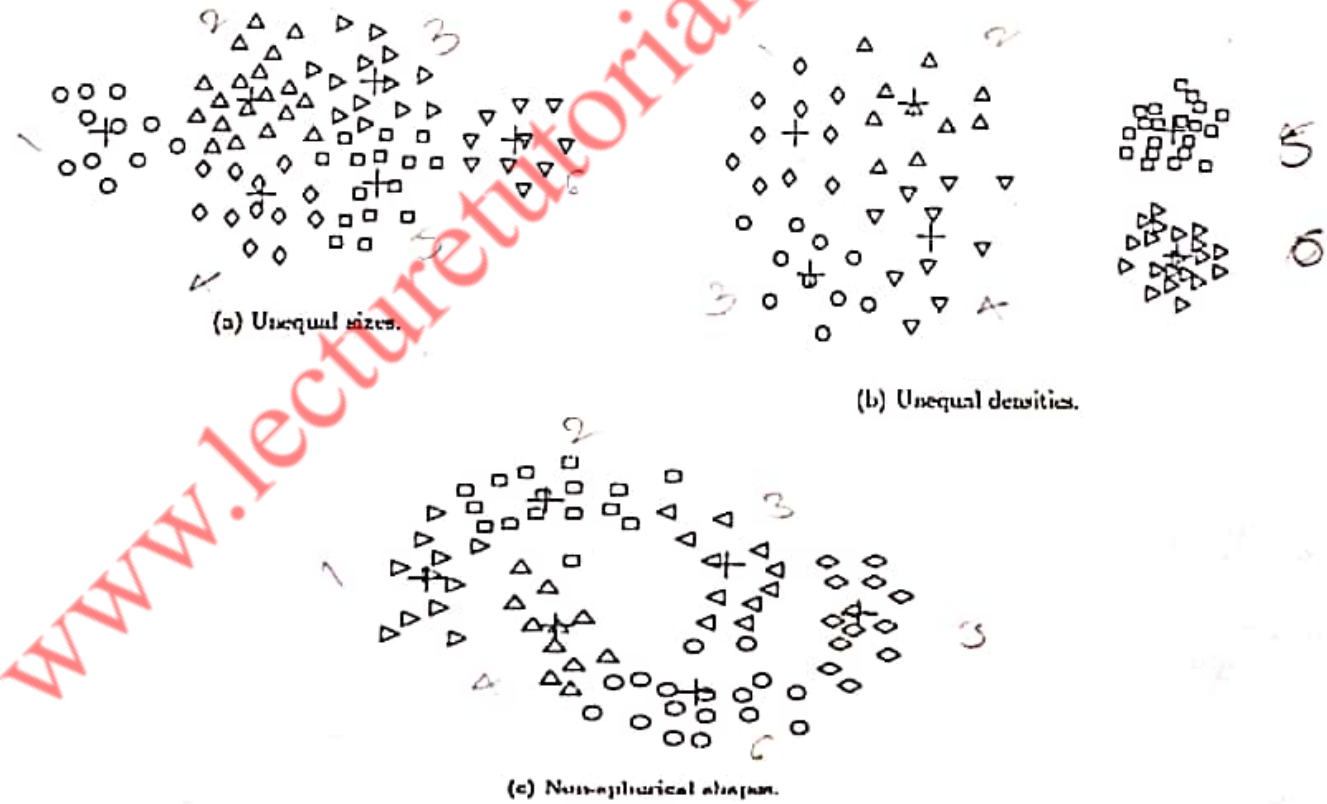
(a) Original points. (b) Three K-means clusters.

Figure 8.10. K-means with clusters of different density.



(a) Original points. (b) Two K-means clusters.

Figure 8.11. K-means with non-globular clusters.



(a) Unequal sizes.

(b) Unequal densities.

(c) Non-spherical shapes.

Figure 8.12. Using K-means to find clusters that are subclusters of the natural clusters.

- The figure 8.9, K-means cannot find the three natural clusters because one of the clusters is much larger than the other two and the larger cluster is broken.
- The figure 8.10, K-means fails to find the three natural clusters because the two smaller clusters are much denser than the larger cluster.
- The figure 8.11, K-means finds two clusters that mix portions of the two natural clusters because the shape of the natural clusters is not globular.
- The figure 8.12 shows what happens to the three previous data sets if we find six clusters instead of two or three.

6.8: K-means as an Optimization Problem:

One way to solve this problem "to find a global optimum" is to enumerate all possible ways of dividing the points into clusters and then choose the set of clusters that best satisfies the objective function. Eg: Minimize the total SSE K-means algorithm and Derivation of K-means for SAE.

Derivation of K-means as an Algorithm to Minimize the SSE: The K-means algorithm which is used for centriod can be mathematically derived when the proximity function is Euclidean distance and the objective is to minimize the SSE. In order to minimize the cluster SSE, the cluster centroid is updated efficiently. The minimized equation repeated for one dimensional data as.

$$SSE = \sum_{i=1}^{K} \sum_{x \epsilon C_i} (c_i - x)^2$$

Here
→C_i is the i^{th} cluster,
→x is a point in C_i
→c_i is the mean of the i^{th} cluster.

This can solve for the K^{th} centroid c_k, is minimize equation by differentiating the SSE.

$$d/d_{ck}\, SSE = d/d_{ck} \sum_{i=1}^{K} \sum_{x \epsilon C_i} (c_i - x)^2$$

$$= \sum_{i=1}^{K} d/d_{ck} \sum_{x \epsilon C_i} (c_i - x)^2$$

$$= \sum_{x \epsilon C_i} 2 * (c_k - x_k) = 0$$

Derivation of K-means for SAE : To demonstrate that the K-means algorithm can be applied to a variety of different objective functions, and partition the data into K clusters such that the sum of the Manhattan (L_1) distances of points from the center of their clusters is minimized. The Sum of the L_1 absolute errors (SAE) is minimized by the following equation.

Considering one dimensional data dist $L_1 = |c_i - x|$.

$$SAE = \sum_{i=1}^{K} \sum_{x \epsilon C_i} dist_{L1}(c_i, x)$$

Here
→C_i is the i^{th} cluster,
→x is a point in C_i
→c_i is the mean of the i^{th} cluster.

Minimizing the equation (1) by differentiating the SSE for i^{th} centriod C_i. It sets the SSE to zero.

$$d/d_{ck} SAE = d/d_{ck} \sum_{i=1}^{K} \sum_{x \in C_i} dist_{L1}(c_i, x)$$

$$= \sum_{i=1}^{K} \sum_{x \in C_i} d/d_{ck}\, dist_{L1}(c_i - x)$$

$$= \sum_{x \in C_i} sign(c_i - x)$$

c_i = Median $x \in C_i$ which is the median of the points in the clusters.

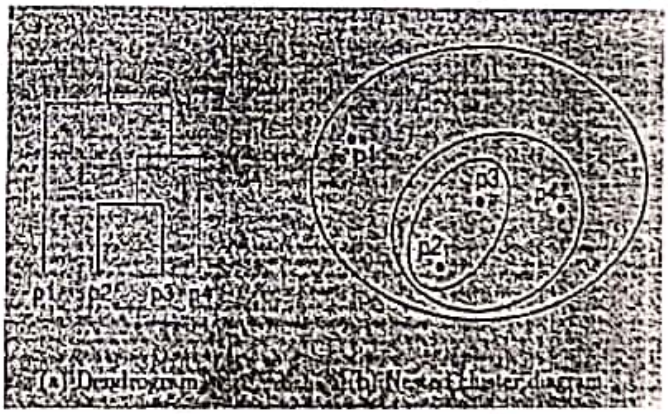
6.9. Agglomerative Hierarchical Clustering: Hierarchical clustering techniques are a second important category of clustering methods. But these techniques are most commonly used in many clustering algorithms. There are two basic approaches for generating a hierarchical clustering:

1. **Agglomerative:** This technique is used to start with the points as individual and at each step, it merge the closest pair of clusters. This defines a notation of cluster proximity.
2. **Divisive:** This technique is used start with one, all-inclusive cluster and at each step, split a cluster until only singleton clusters of individual points remain. In this case, we need to decide which cluster to split at each step and how to do the splitting.

A hierarchical clustering is often displayed graphically using a tree-like diagram called a **dendrogram.** This displays both the cluster and sub-cluster relationships and the order in which the clusters were merged(agglomerative) or split(divisive). This approach is expressed in Algorithm.

Basic agglomerative hierarchical clustering algorithm:

1. Compute the proximity matrix, if necessary.
2. Repeat the steps 3 & 4 until only one cluster remains
3. Merge the closest two clusters
4. Update the proximity matrix to reflect the proximity b/w the new cluster and the original clusters.
5. Until Only one cluster remains.

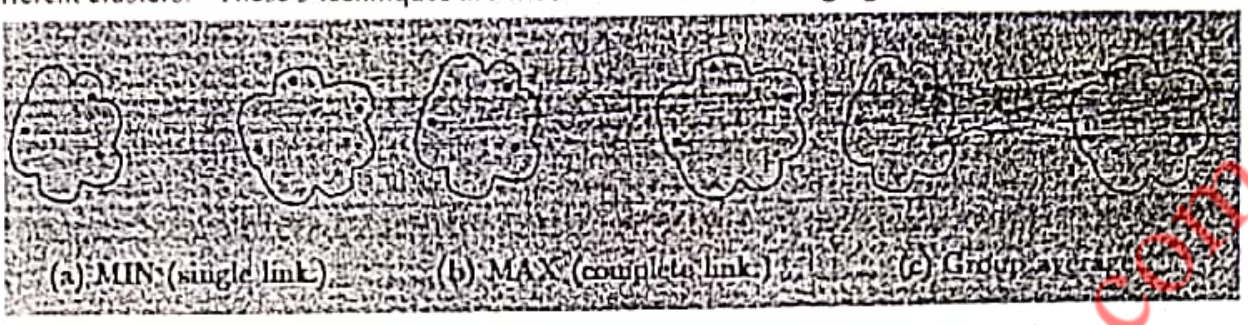


Many agglomerative hierarchical clustering techniques are MIN, MAX, and Group Average.

MIN: It defines cluster proximity as the proximity b/w the closest two points that are in different clusters, or using graph terms, the shortest edge b/w two nodes in different subsets of nodes.

MAX: This technique defines the cluster proximity as proximity b/w two points that are farthest from each other in different clusters. In graph terminology, the maximum edge b/w the nodes that belong to different nodes subsets is a MAX.

Group Average: This technique defines cluster proximity to be the average pair wise proximities of all pairs from different clusters. These 3 techniques are illustrated in the following figure:



6.9.1: Time and Space Complexity: The basic agglomerative hierarchical clustering algorithm requires the storage of ½ m^2 proximate.

Where m is the number of data points.

- The space needed to keep track of the clusters is proportional to the number of clusters, as m – 1. Total space complexity is $O(m^2)$.
- The overall time required for a hierarchical clustering based on Algorithm is $O(m^2 \log m)$.
- The space and time complexity of hierarchical clustering severely limits the size of data sets that can be processed.

6.9.2: Specific Techniques of Agglomerative hierarchical clustering:

The specific techniques are illustrated for the behavior of the various hierarchical clustering algorithms. This uses simple data that consists of 6 two-dimensional points, shown in fig.

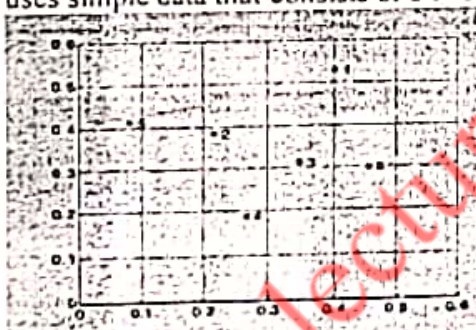


Figure 8.15. Set of 6 two-dimensional points.

Point	x Coordinate	y Coordinate
p1	0.40	0.53
p2	0.22	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

Table 8.3. xy coordinates of 6 points.

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Table 8.4. Euclidean distance matrix for 6 points.

Single Link or MIN: For Single line or MIN technique, the proximity of two clusters is defined as the minimum of the distance b/w any two points in the two different clusters. The single link technique is good at handling non-elliptical shapes, but is sensitive to noise and outliers.

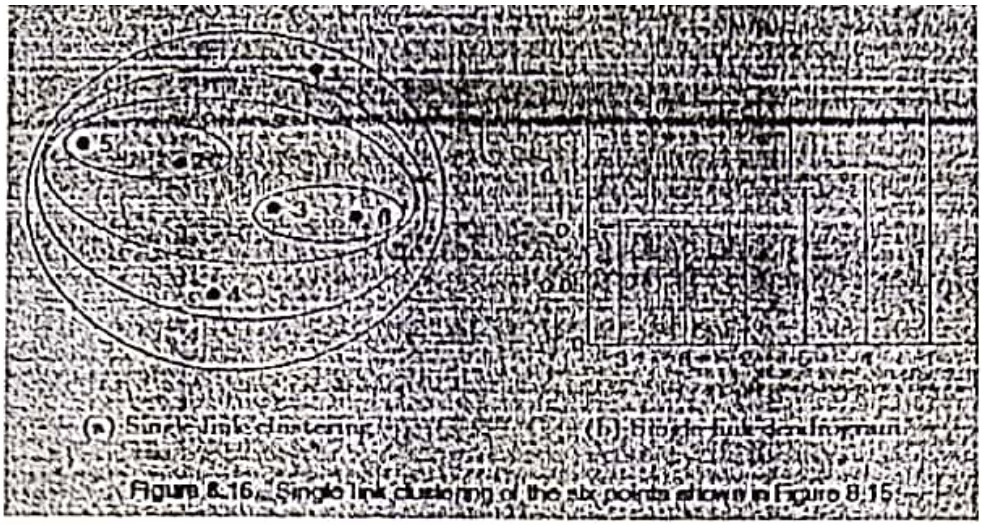


Figure 8.16. Single link clustering of the six points shown in Figure 8.15.

For example: The following figure shows the result of applying the single link technique to data set of six points.

→Figure (a) shows the nested clusters as a sequence of nested ellipses, where the numbers associated with the ellipses indicate the order of the clustering.

→Figure (b) shows the same information, but as a dendrogram. The height at which two clusters are merged in the dendrogram reflects the distance of the two clusters. From table 8.4, the distance b/w points 3 and 6 is 0.11, and that is the height at which they are joined into one cluster in the dendrogram.

Another example, the distance b/w clusters {3,6} and {2,5} is given by

dist {3,6}, {2,5} = min (dist(3,2), dist(6,2), dis(3,5), dist(6,5))
= min (0.15,0.25,0.28,0.39)
= 0.15

Complete Link or MAX or CLIQUE: For complete link or MAX technique of hierarchical clustering, the proximity of two clusters is defined as the maximum of the distance b/w any two points in the two different clusters.

For example: The following figure shows the result of applying the single link technique to data set of six points.

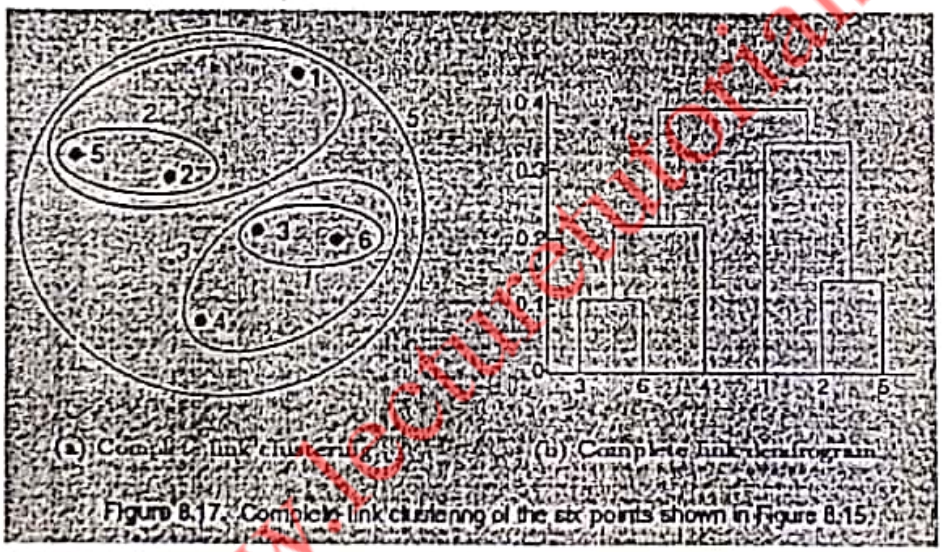


Figure 8.17. Complete link clustering of the six points shown in figure 8.15.

From table 8.4, the points 3 and 6 are merged first. For example,

Points {3,6} is merged with {4}, instead of {2,5} or {1} because:

Dist({3,6},{4}) = max(dist(3,4), dist(6,4)
= max(0.15, 0.22)
= 0.22

Dist({3,6},{2,5})=max(dist(3,6),dist(6,2)
,dist(3,5),dist(6,5)
= max(0.15,
0.25,0.28,
0.39)
= 0.39

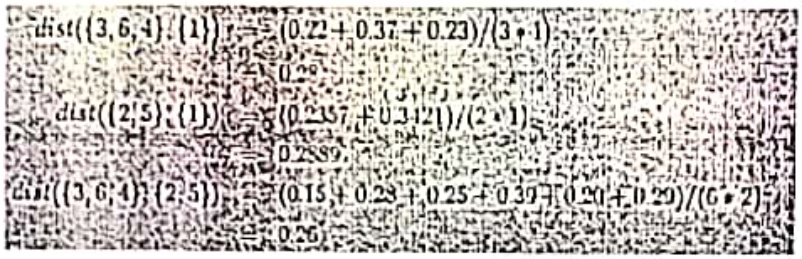
dist({3,6}, {1} = max(dist(3,1), dist(6,1)
= max(0.22,0.23)
= 0.23

Group Average: For the group average of hierarchical clustering, the proximity of two clusters is defined as the average pairwise proximity among all pairs of points in the

$$proximity(C_i, C_j) = \frac{\sum_{x \in C_i, y \in C_j} proximity(x, y)}{m_i \times m_j} \quad (8.6)$$

different clusters. This is an intermediate approach b/w the single and complex link approaches. Thus, for group average, the cluster proximity (C_i , C_j) of clusters C_i and C_j.

For example: The following figure shows the result of applying the complex link technique to data set of six points. The group average is illustrated as follows.



Because dist({3,6,4}, {2,5} is smaller than dist({3,6,4}, {1}) and dist({2,5}, {1}), clusters {3,6,4} and {2,5} are merged at the fourth stage.

6.10: DBSCAN: Density-based clustering locates regions of high density that are separated from one another by regions of low density. It is a simple and effective density-based clustering technique for density based clustering.

Traditional Density for Center-based Approach:
In the center-based approach, density is estimated for a particular point in the data set by counting the number of points within a specified radius, *Eps*, of that point. This includes the point itself. This technique is graphically illustrated. The number of points within a radius of *Eps* of point A is 7, including A itself.

This method is simple to implement, but the density of any point will depend on the specified radius. For example, if the radius is large enough, then all points will have a density of m, the number of points in the data set. Similarly If the radius is too small, then all points will have a density of 1.

Classification of points According to Center-Based Density: The center-based approach to density allows us to classify a point as being

1) In the interior of a dense region (a core point).
2) On the edge of a dense region (a border point).
3) In a sparsely occupied region (a noise or background point).

The figure graphically illustrates the concepts of core, border, and noise points using a collection of two-dimensional points.

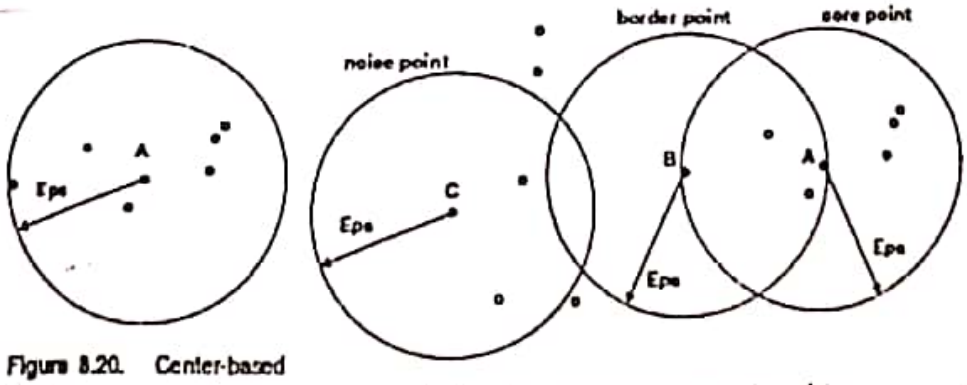


Figure 8.20. Center-based density.

Figure 8.21. Core, border, and noise points.