

**Cluster Analysis Introduction:** Types of Data in Cluster Analysis, A Categorization of Major Clustering Methods.

**Partitioning Methods:** Classical Partitioning Methods: k-Means and k-Medoids, Hierarchical Methods, Density-Based Methods, Grid-Based Methods.

## **CLUSTER ANALYSIS:**

The process of grouping a set of physical or abstract objects into classes of similar objects is called **clustering**. A cluster is a collection of data objects that are

- similar to one another within the same cluster
- dissimilar to the objects in other clusters

Clustering is also called **data segmentation** in some applications because clustering partitions large data sets into groups according to their **similarity**.

Cluster analysis is finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters. It is unsupervised learning i.e., no predefined classes.

Applications are Pattern recognition, Spatial data analysis, Image processing, Economic science, World Wide Web and many others.

Examples of cluster applications:

- **Marketing:** Help marketers discover distinct groups in their customer bases, and 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.

A good cluster method will produce high quality clusters with

- high intra-class similarity
- Low inter-class similarity

The quality of a clustering result depends on both the similarity measure used by the method and its implementation. Quality can also be measured by its ability to discover some or all of the hidden patterns.

The following are typical requirements of clustering in data mining:

- Scalability
- Ability to deal with different types of attributes
- Discovery of clusters with arbitrary shape

- Minimal requirements for domain knowledge to determine input parameters
- Ability to deal with noisy data
- Incremental clustering and insensitivity to the order of input records
- High dimensionality
- Constraint-based clustering
- Interpretability and usability

## TYPES OF DATA IN CLUSTER ANALYSIS

Main memory-based clustering algorithms typically operate on either of the following two data structures.

- **Data matrix (or object-by-variable structure):** This represents  $n$  objects, such as persons, with  $p$  variables (also called measurements or attributes), such as age, height, weight, gender, and so on. The structure is in the form of a relational table, or  $n$ -by- $p$  matrix ( $n$  objects  $\times p$  variables)

$$\begin{bmatrix} x_{11} & \dots & x_{1f} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{if} & \dots & x_{ip} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{nf} & \dots & x_{np} \end{bmatrix} \quad (\text{Two modes})$$

- **Dissimilarity matrix (or object-by-object structure):** This stores a collection of proximities that are available for all pairs of  $n$  objects. It is often represented by an  $n$ -by- $n$  table

$$\begin{bmatrix} 0 & & & & & \\ d(2, 1) & 0 & & & & \\ d(3, 1) & d(3, 2) & 0 & & & \\ \vdots & \vdots & \vdots & & & \\ d(n, 1) & d(n, 2) & \dots & \dots & 0 & \end{bmatrix} \quad (\text{One Mode})$$

where  $d(i, j)$  is the measured difference or dissimilarity between objects  $i$  and  $j$ .

### Data matrix versus Dissimilarity matrix is as follows

- | Data matrix  | Dissimilarity matrix                                 |
|--|--|
| 1. The rows and columns in this matrix represent different entities.   | 1. The rows and columns represent the same entity.   |
| 2. This matrix is called <b>two-mode</b> matrix.   | 2. This matrix is called <b>one-mode</b> matrix.     |
| 3. If the data are presented in the form of a data matrix, it can first be transformed into a dissimilarity matrix before applying such clustering algorithms. | 3. Many clustering algorithms operate on this matrix |

## Types of data in clustering analysis are as follows

1. Interval-scaled variables
2. Binary variables
3. Nominal, ordinal and ratio variables.
4. Variables of mixed types.
5. Vector Objects

### 1. Interval-scaled variables:

Interval-scaled variables are continuous measurements of a roughly linear scale.

Examples: weight and height, latitude and longitude coordinates (e.g., when clustering houses), and weather temperature

To standardize measurements, one choice is to convert the original measurements to unitless variables. Given measurements for a variable  $f$ , this can be performed as follows.

1. Calculate the mean absolute deviation,  $s_f$ :

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \dots + |x_{nf} - m_f|),$$

where  $x_{1f}, \dots, x_{nf}$  are  $n$  measurements of  $f$ , and  $m_f$  is the *mean* value of  $f$ , that is,  $m_f = \frac{1}{n}(x_{1f} + x_{2f} + \dots + x_{nf})$ .

2. Calculate the standardized measurement, or z-score:

$$z_{if} = \frac{x_{if} - m_f}{s_f}.$$

Distances are normally used to measure the similarity or dissimilarity between two data objects.

- **Euclidean distance** is defined as

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \dots + (x_{in} - x_{jn})^2},$$

where  $i = (x_{i1}, x_{i2}, \dots, x_{in})$  and  $j = (x_{j1}, x_{j2}, \dots, x_{jn})$  are two  $n$ -dimensional data objects.

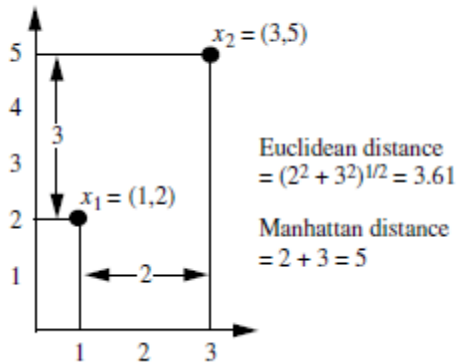
- **Manhattan (or city block) distance** is defined as

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{in} - x_{jn}|.$$

Both the Euclidean distance and Manhattan distance satisfy the following mathematic requirements of a distance function

1.  $d(i, j) \geq 0$ : Distance is a nonnegative number.
2.  $d(i, i) = 0$ : The distance of an object to itself is 0.
3.  $d(i, j) = d(j, i)$ : Distance is a symmetric function.
4.  $d(i, j) \leq d(i, h) + d(h, j)$ : Going directly from object  $i$  to object  $j$  in space is no more than making a detour over any other object  $h$  (*triangular inequality*).

Example: Let  $x_1 = (1, 2)$  and  $x_2 = (3, 5)$  represent two objects as in Figure. The Euclidean distance between the two is  $\sqrt{(2^2 + 3^2)} = 3.61$ . The Manhattan distance between the two is  $2+3 = 5$ .



- **Minkowski distance** is a generalization of both Euclidean distance and Manhattan distance. It is defined as

$$d(i, j) = (|x_{i1} - x_{j1}|^p + |x_{i2} - x_{j2}|^p + \dots + |x_{in} - x_{jn}|^p)^{1/p}$$

Where  $p$  is a positive integer. Such a distance is also called  $L_p$  norm, in some literature. It represents the Manhattan distance when  $p = 1$  (i.e.,  $L_1$  norm) and Euclidean distance when  $p = 2$  (i.e.,  $L_2$  norm).

- If each variable is assigned a weight according to its perceived importance, the weighted Euclidean distance can be computed as

$$d(i, j) = \sqrt{w_1|x_{i1} - x_{j1}|^2 + w_2|x_{i2} - x_{j2}|^2 + \dots + w_m|x_{in} - x_{jn}|^2}$$

Weighting can also be applied to the Manhattan and Minkowski distances.

## 2. Binary variables:

A binary variable has only two states: 0 or 1, where 0 means that the variable is absent, and 1 means that it is present. If binary variables have interval scaled, then it lead to misleading clustering results. Therefore methods specific to binary data are necessary for computing dissimilarities.

		object <i>j</i>		
		1	0	sum
object <i>i</i>	1	<i>q</i>	<i>r</i>	<i>q+r</i>
	0	<i>s</i>	<i>t</i>	<i>s+t</i>
	sum	<i>q+s</i>	<i>r+t</i>	<i>p</i>

Binary variables are as follows

- **Symmetric binary variable:** A binary variable is symmetric if both of its states are equally valuable and carry the same weight; that is, there is no preference on which outcome should be coded as 0 or 1.  
Example: attribute gender → male and female
- **Symmetric binary dissimilarity** based on symmetric binary variables and assess the dissimilarity between objects *i* and *j*.

$$d(i, j) = \frac{r+s}{q+r+s+t}.$$

- **Asymmetric binary variable:** A binary variable is asymmetric if the outcomes of the states are not equally.  
Example: Disease test → positive and negative
- **Asymmetric binary dissimilarity:** It has number of negative matches, *t*, is considered unimportant and thus is ignored in computation as

$$d(i, j) = \frac{r+s}{q+r+s}.$$

Distance can be measured between two binary variables based on the notion of **similarity** instead of **dissimilarity**.

For example, the asymmetric binary similarity between the objects *i* and *j*, or  $sim(i, j)$ , can be computed as,

$$sim(i, j) = \frac{q}{q+r+s} = 1 - d(i, j).$$

The coefficient  $sim(i, j)$  is called the **Jaccard coefficient**.

Example: Consider patient record table contain the attributes name, gender, fever, cough, test-1, test-2, test-3 and test-4, where name is an object identifier, gender is a symmetric attribute, and the remaining attributes are asymmetric binary

<i>name</i>	<i>gender</i>	<i>fever</i>	<i>cough</i>	<i>test-1</i>	<i>test-2</i>	<i>test-3</i>	<i>test-4</i>
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	Y	N	N	N	N
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

For asymmetric attribute values, let the values Y (yes) and P (positive) be set to 1, and the value N (no or negative) be set to 0.

$$d(i, j) = \frac{r+s}{q+r+s}.$$

Then

$$d(\text{Jack}, \text{Mary}) = \frac{0+1}{2+0+1} = 0.33$$

$$d(\text{Jack}, \text{Jim}) = \frac{1+1}{1+1+1} = 0.67$$

$$d(\text{Mary}, \text{Jim}) = \frac{1+2}{1+1+2} = 0.75$$

These measurements suggest that Mary and Jim are unlikely to have a similar disease because they have the highest dissimilarity value among to three pairs of the three patients, Jack and Mary are the most likely to have a similar disease.

### 3. Categorical, Ordinal, and Ratio-Scaled Variables:

Objects can be described by categorical, ordinal, and ratio-scaled variables are as follows

**Categorical Variables:** A categorical variable is a generalization of the binary variable in that it can take on more than two states.

Example: map color is a categorical variable that may have, say, five states: red, yellow, green, pink, and blue.

The dissimilarity between two objects  $i$  and  $j$  can be computed based on the ratio of mismatches

$$d(i, j) = \frac{p-m}{p},$$

Where  $m$  is the number of matches (i.e., the number of variables for which  $i$  and  $j$  are in the same state), and  $p$  is the total number of variables. Weights can be assigned to increase the effect of  $m$  or to assign greater weight to the matches in variables having a larger number of states.

Example: Suppose sample data of table, except that only object-identifier and the variable test-1 are available, where test-1 is categorical

<b>object Identifier</b>	<b>test-1 (categorical)</b>	<b>test-2 (ordinal)</b>	<b>test-3 (ratio-scaled)</b>
1	code-A	excellent	445
2	code-B	fair	22
3	code-C	good	164
4	code-A	excellent	1,210

→ Table- I

Dissimilarity matrix is as follows

$$\begin{bmatrix} 0 & & & \\ d(2,1) & 0 & & \\ d(3,1) & d(3,2) & 0 & \\ d(4,1) & d(4,2) & d(4,3) & 0 \end{bmatrix}$$

Here we have one categorical variable, test-1, set  $p=1$  so that  $d(i,j)$  evaluates to 0 if objects I and j match, and 1 if the objects differ. So

$$\begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ 1 & 1 & 0 & \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

**Ordinal Variables:** A **discrete ordinal variable** resembles a categorical variable, except that the  $M$  states of the ordinal value are ordered in a meaningful sequence. An **ordinal variable** can be discrete or continuous. Order is important.

Example: Professional ranks enumerated in a sequential order, such as assistant, associate and professors.

It can be treated like interval-scaled as follows

- Replace  $x_{if}$  by their rank i.e.,  $r_{if} \in \{1, \dots, M_f\}$
- Map the range of each variable onto  $[0,1]$  by replacing  $i$ th object in the  $f$ th variable by  $z_{if} = \frac{r_{if} - 1}{M_f - 1}$ .
- Compute the dissimilarity using methods for interval-scaled variables.

Example: Suppose sample data taken from Table-I, except that this time only the object-identifier and the continuous ordinal variable, test-2 are available. There are three states for test-2, namely fair, good and excellent, ie.,  $M_f = 3$

- Replace each value for test-2 by its rank, the objects are assigned the ranks 3, 1, 2 and 3 respectively.
- Normalizes the ranking by mapping rank 1 to 0.0, rank 2 to 0.5 and rank 3 to 1.0.
- Use, Euclidean distances

$$d(i, j) = \sqrt{w_1|x_{i1} - x_{j1}|^2 + w_2|x_{i2} - x_{j2}|^2 + \dots + w_m|x_{in} - x_{jn}|^2}$$

Which results in the following dissimilarity matrix

$$\begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ 0.5 & 0.5 & 0 & \\ 0 & 1.0 & 0.5 & 0 \end{bmatrix}$$

**Ratio-Scaled Variables:** A positive measurement on a nonlinear scale, approximately at exponential scale, such as

$$Ae^{Bt} \quad \text{or} \quad Ae^{-Bt}$$

where  $A$  and  $B$  are positive constants, and  $t$  typically represents time.

Example: Growth of a bacteria population or decay of a radioactive element.

There are three methods to handle ratio-scaled variables for computing the dissimilarity between objects as follows

- Treat ratio-scaled variables like interval-scaled variables
- Apply **logarithmic transformation** to a ratio-scaled variable  $f$  having value  $x_{if}$  for object  $i$  by using the formula  $y_{if} = \log(x_{if})$ .
- Treat  $x_{if}$  as continuous ordinal data and treat their ranks as interval-valued.

Example: Consider Table-I, except that only object identifier and the ratio-scaled variable, test-3, are available. Take the log of test-3 results in the values 2.65, 1.34, 2.21, and 3.08 for the objects 1 to 4, respectively. Using the Euclidean distance on the transformed values, obtain the following dissimilarity matrix:

$$\begin{bmatrix} 0 & & & \\ 1.31 & 0 & & \\ 0.44 & 0.87 & 0 & \\ 0.43 & 1.74 & 0.87 & 0 \end{bmatrix}$$

#### 4. Variables of Mixed Types:

A database may contain all the six types of variables as symmetric binary, asymmetric binary, nominal, ordinal, interval and ratio.

Suppose that the data set contains  $p$  variables of mixed type. The dissimilarity  $d(i, j)$  between objects  $i$  and  $j$  is defined as



$$d(i, j) = \frac{\sum_{f=1}^p \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^p \delta_{ij}^{(f)}}$$

where the indicator  $\delta_{ij}^{(f)} = 0$  if either (1)  $x_{if}$  or  $x_{jf}$  is missing (i.e., there is no measurement of variable  $f$  for object  $i$  or object  $j$ ), or (2)  $x_{if} = x_{jf} = 0$  and variable  $f$  is asymmetric binary; otherwise,  $\delta_{ij}^{(f)} = 1$ . The contribution of variable  $f$  to the dissimilarity between  $i$  and  $j$ , that is,  $d_{ij}^{(f)}$ , is computed dependent on its type:

- If  $f$  is interval-based:  $d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{\max_h x_{hf} - \min_h x_{hf}}$ , where  $h$  runs over all nonmissing objects for variable  $f$ .
- If  $f$  is binary or categorical:  $d_{ij}^{(f)} = 0$  if  $x_{if} = x_{jf}$ ; otherwise  $d_{ij}^{(f)} = 1$ .
- If  $f$  is ordinal: compute the ranks  $r_{if}$  and  $z_{if} = \frac{r_{if} - 1}{M_f - 1}$ , and treat  $z_{if}$  as interval-scaled.
- If  $f$  is ratio-scaled: either perform logarithmic transformation and treat the transformed data as interval-scaled; or treat  $f$  as continuous ordinal data, compute  $r_{if}$  and  $z_{if}$ , and then treat  $z_{if}$  as interval-scaled.

Example: Consider Table-I

$$d(2, 1) = \frac{1(1) + 1(1) + 1(0.75)}{3} = 0.92.$$

Consider

$$\begin{bmatrix} 0 & & & \\ 0.75 & 0 & & \\ 0.25 & 0.50 & 0 & \\ 0.25 & 1.00 & 0.50 & 0 \end{bmatrix}$$

Resulting dissimilarity matrix obtained for the data described by the three variables of mixed type is

$$\begin{bmatrix} 0 & & & \\ 0.92 & 0 & & \\ 0.58 & 0.67 & 0 & \\ 0.08 & 1.00 & 0.67 & 0 \end{bmatrix}$$

**5. Vector Objects:** Keywords in documents, general features in micro-arrays etc. Applications are information retrieval, biologic taxonomy etc..

There are several ways to define such a similarity function,  $s(x, y)$ , to compare two vectors  $x$  and  $y$ . One popular way is to define the similarity function as a cosine measure as follows

$$s(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^t \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}$$

where  $\mathbf{x}^t$  is a transposition of vector  $\mathbf{x}$ ,  $\|\mathbf{x}\|$  is the Euclidean norm of vector  $\mathbf{x}$ ,  $\|\mathbf{y}\|$  is the Euclidean norm of vector  $\mathbf{y}$ , and  $s$  is essentially the cosine of the angle between vectors  $\mathbf{x}$  and  $\mathbf{y}$ .

Example: Nonmetric similarity between two objects using cosine

Given two vectors  $\mathbf{x} = (1, 1, 0, 0)$  and  $\mathbf{y} = (0, 1, 1, 0)$

Similarity between  $\mathbf{x}$  and  $\mathbf{y}$  is

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

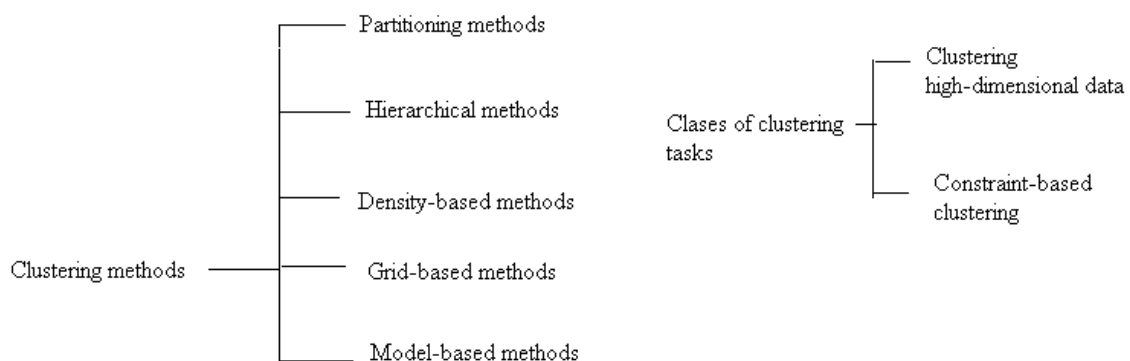
A simple variation of above measure is

$$s(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^t \cdot \mathbf{y}}{\mathbf{x}^t \cdot \mathbf{x} + \mathbf{y}^t \cdot \mathbf{y} - \mathbf{x}^t \cdot \mathbf{y}}$$

Which is the ratio of the number of attributes shared by  $\mathbf{x}$  and  $\mathbf{y}$  to the number of attributes possessed by  $\mathbf{x}$  or  $\mathbf{y}$ . This function, known as the **Tanimoto coefficient** or **Tanimoto distance**, is frequently used in information retrieval and biology taxonomy.

## A CATEGORIZATION OF MAJOR CLUSTERING METHODS

The major clustering methods can be classified into the following categories.



- **Partitioning methods:** Constructs various partitions and then evaluate them by some criterion. Example: Minimizing the sum of square errors. Typical methods are k-Means, k-Medoids, CLARANS.
- **Hierarchical methods:** Create a hierarchical decomposition of the set of data (or objects) using some criterion. This method can be classified based on
  - **Agglomerative approach (Bottom-up):** 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.

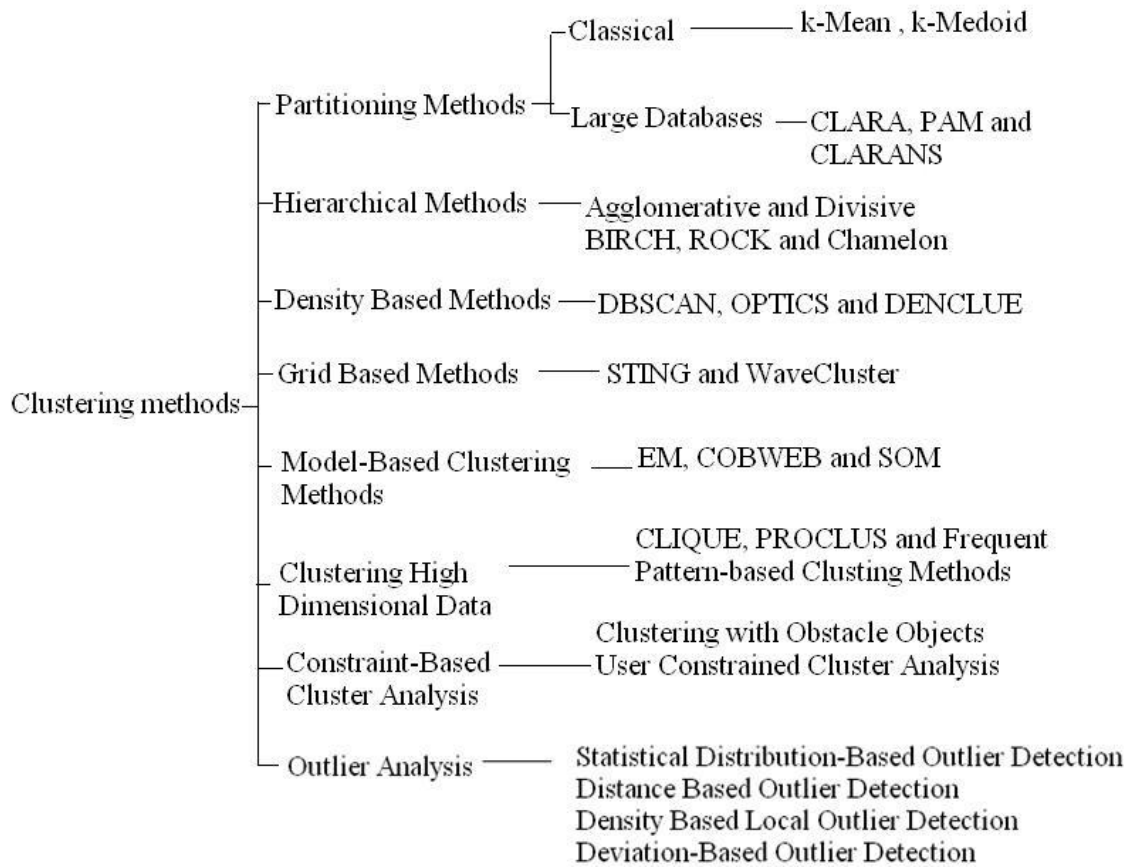
- **Divisive approach (Top-down):** starts with all of the objects in the same cluster. In each successive iteration, a cluster is split up into smaller clusters, until eventually each object is in one cluster, or until a termination condition holds.

Typical methods are BIRCH, ROCK, Chameleon and so on

- **Density-based methods:** It is based on connectivity and density functions. Typical methods are DBSCAN, OPTICS, DENCLUE etc.
- **Grid-based methods:** It is based on a multiple-level granularity structure. Typical methods are STING, WaveCluster, and CLIQUE etc.
- **Model-based methods:** This model is hypothesized for each of the clusters and tries to find the best fit of that model to each other. Typical methods are EM, SOM and COBWEB etc.

Classes of clustering tasks are as follows

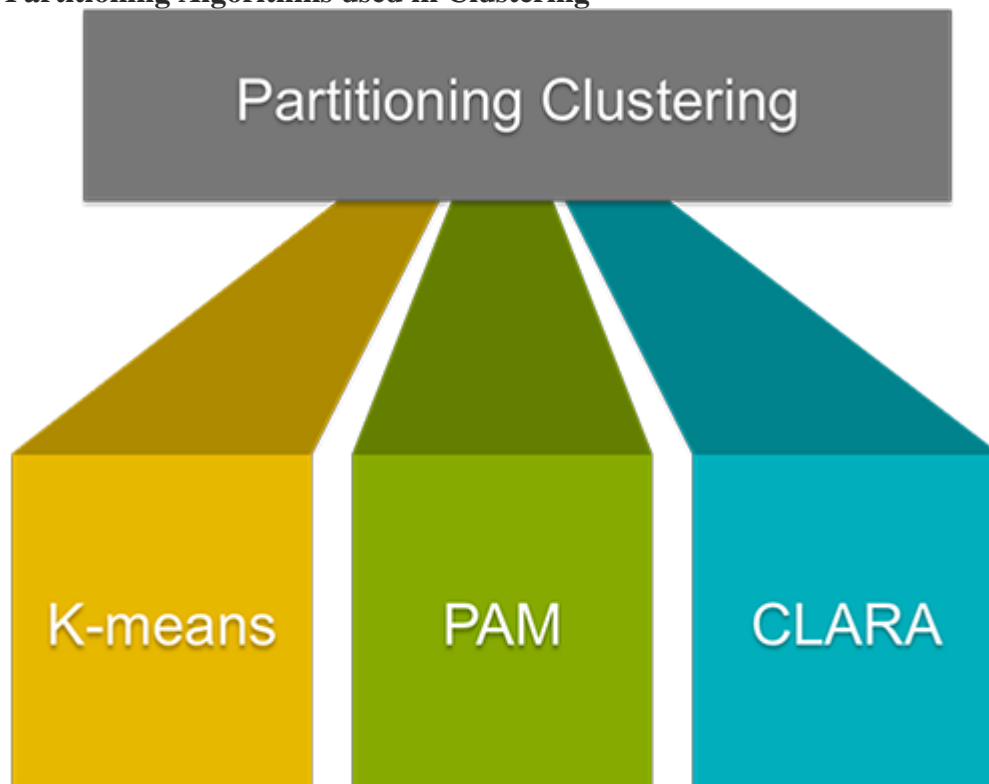
- **Clustering high-dimensional data** is a particularly important task in cluster analysis because many applications require the analysis of objects containing a large number of features or dimensions. Frequent pattern-based clustering, another clustering methodology, and extracts distinct frequent patterns among subsets of dimensions that occur frequently.
- **Constraint-based clustering** is a clustering approach that performs clustering by incorporation of user-specified or application-oriented constraints. It focus mainly on
  - Spatial clustering
  - Semi-supervised clustering



## Partitional Clustering

The most popular class of clustering algorithms that we have is the iterative relocation algorithms. These algorithms minimize a given clustering criterion by iteratively relocating data points between clusters until a (locally) optimal partition is attained. There are many algorithms that come under partitioning method some of the popular ones are K-Means, PAM(k-Medoid), CLARA algorithm (Clustering Large Applications) etc.

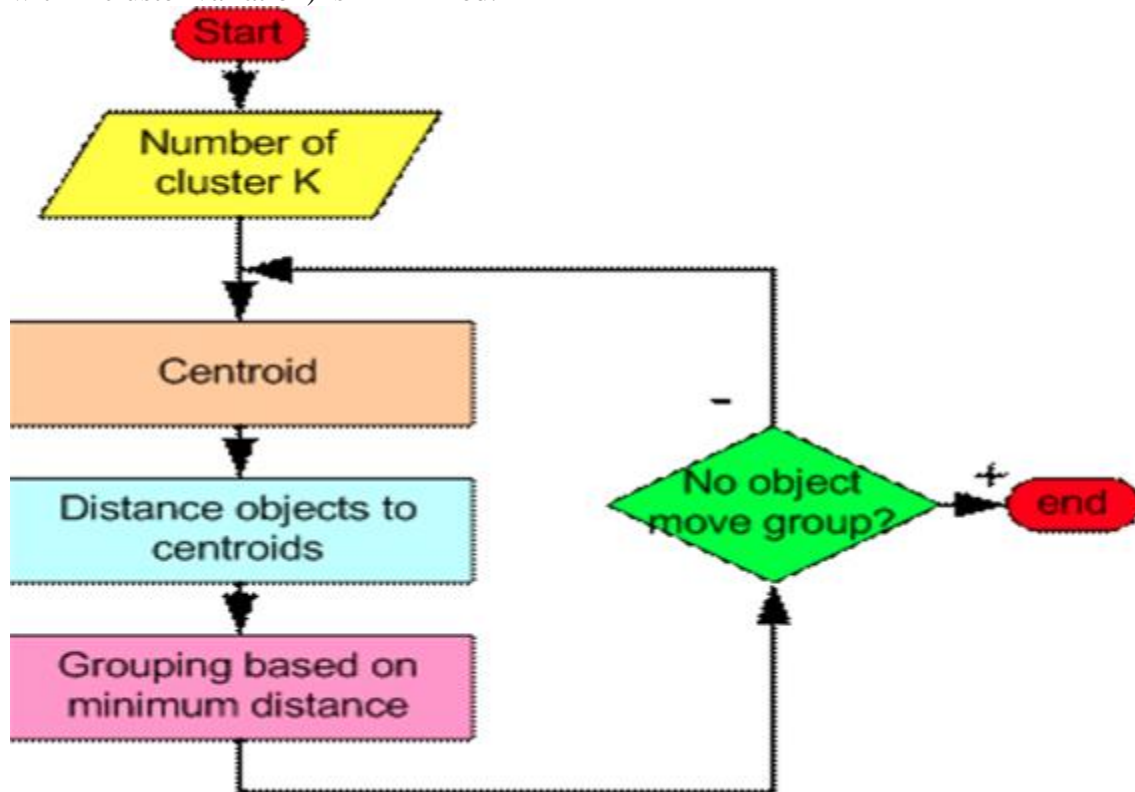
### Partitioning Algorithms used in Clustering -



### Types of Partitional Clustering

**K-Means Algorithm (A centroid based Technique):** It is one of the most commonly used algorithm for partitioning a given data set into a set of  $k$  groups (i.e.  $k$  clusters), where  $k$  represents the number of groups. It classifies objects in multiple groups (i.e., clusters), such that objects within the same cluster are as similar as possible (i.e., high *intra-class similarity*),

whereas objects from different clusters are as dissimilar as possible (i.e., low *inter-class similarity*). In k-means clustering, each cluster is represented by its center (i.e, *centroid*) which corresponds to the mean of points assigned to the cluster. The basic idea behind k-means clustering consists of defining clusters so that the total intra-cluster variation (known as total within-cluster variation) is minimized.



Process for K-means Algorithm

Steps involved in K-Means Clustering :

1. The first step when using k-means clustering is to indicate the number of clusters (k) that will be generated in the final solution.
2. The algorithm starts by randomly selecting k objects from the data set to serve as the initial centers for the clusters. The selected objects are also known as cluster means or centroids.
3. Next, each of the remaining objects is assigned to its closest centroid, where closest is defined using the [Euclidean distance](#) between the object and the cluster mean. This step is called “cluster assignment step”.

- After the assignment step, the algorithm computes the new mean value of each cluster. The term cluster “centroid update” is used to design this step. Now that the centers have been recalculated, every observation is checked again to see if it might be closer to a different cluster. All the objects are reassigned again using the updated cluster means.
- The cluster assignment and centroid update steps are iteratively repeated until the cluster assignments stop changing (i.e until *convergence* is achieved). That is, the clusters formed in the current iteration are the same as those obtained in the previous iteration.

**Step 1 ->**

Subject	A	B
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

**Step 2 ->**

	Individual	Mean Vector (centroid)
Group 1	1	(1.0, 1.0)
Group 2	4	(5.0, 7.0)

**Step 3 ->**

Step	Cluster 1		Cluster 2	
	Individual	Mean Vector (centroid)	Individual	Mean Vector (centroid)
1	1	(1.0, 1.0)	4	(5.0, 7.0)
2	1, 2	(1.2, 1.5)	4	(5.0, 7.0)
3	1, 2, 3	(1.8, 2.3)	4	(5.0, 7.0)
4	1, 2, 3	(1.8, 2.3)	4, 5	(4.2, 6.0)
5	1, 2, 3	(1.8, 2.3)	4, 5, 6	(4.3, 5.7)
6	1, 2, 3	(1.8, 2.3)	4, 5, 6, 7	(4.1, 5.4)

**Step 4 ->**

	Individual	Mean Vector (centroid)
Cluster 1	1, 2, 3	(1.8, 2.3)
Cluster 2	4, 5, 6, 7	(4.1, 5.4)

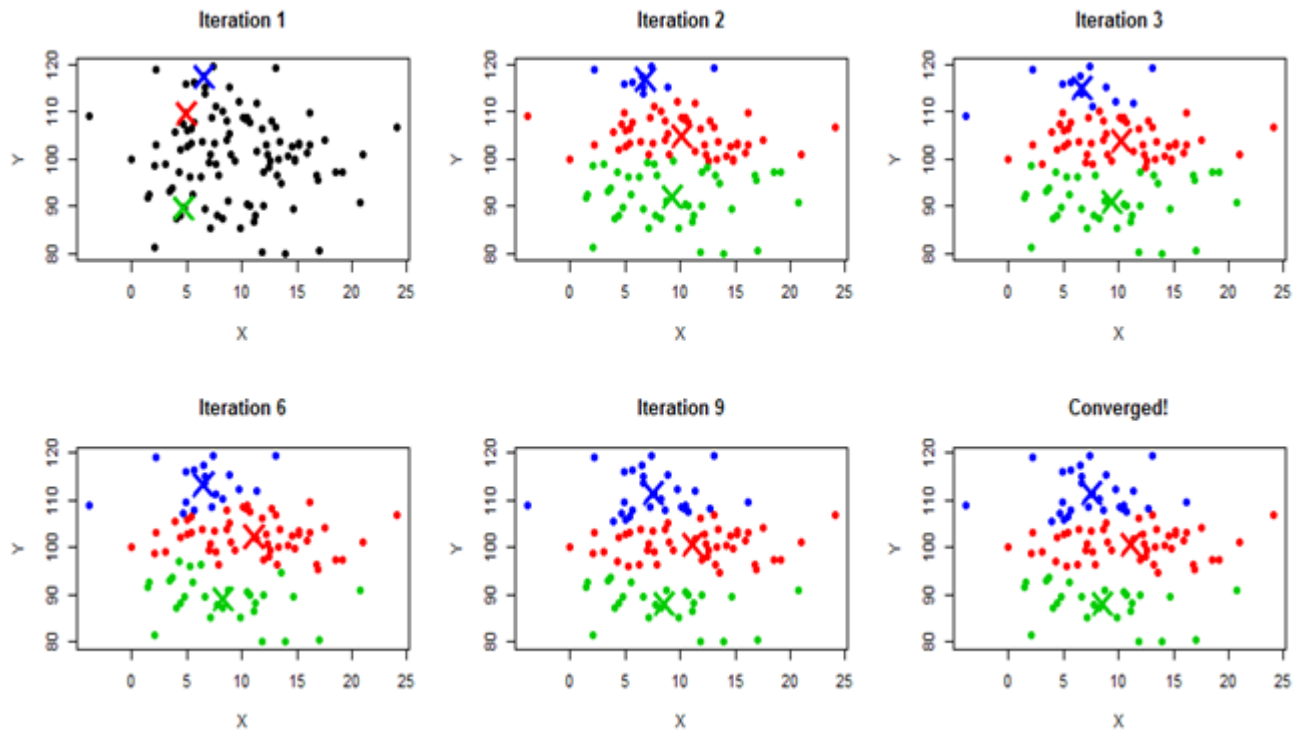
**Step 5 ->**

Individual	Distance to mean (centroid) of Cluster 1	Distance to mean (centroid) of Cluster 2
1	1.5	5.4
2	0.4	4.3
3	2.1	1.8
4	5.7	1.8
5	3.2	0.7
6	3.8	0.6
7	2.8	1.1

**Step 6 ->**

	Individual	Mean Vector (centroid)
Cluster 1	1, 2	(1.3, 1.5)
Cluster 2	3, 4, 5, 6, 7	(3.9, 5.1)

Example for K-Means Clustering

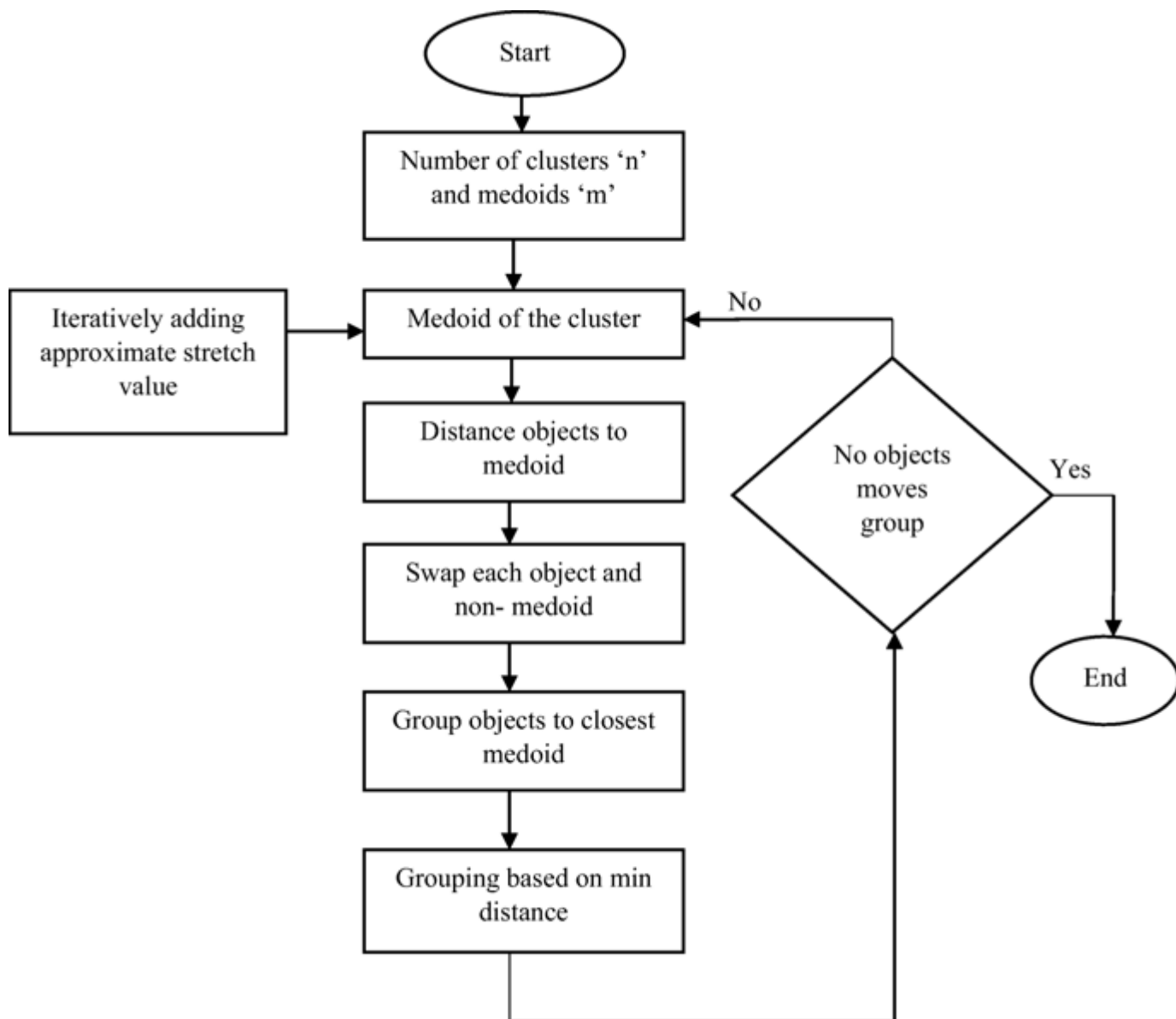


Plotting of K-means Clustering

### **K-Medoids Algorithm (Partitioning Around Medoid) :**

1. A medoid can be defined as the point in the cluster, whose similarities with all the other points in the cluster is maximum.
2. In k-medoids clustering, each cluster is represented by one of the data point in the cluster. These points are named cluster medoids. The term medoid refers to an object within a cluster for which average dissimilarity between it and all the other the members of the cluster is minimal. It corresponds to the most centrally located point in the cluster.
3. These objects (one per cluster) can be considered as a representative example of the members of that cluster which may be useful in some situations. Recall that, in k-means clustering, the center of a given cluster is calculated as the mean value of all the data points in the cluster.
4. K-medoid is a robust alternative to k-means clustering. This means that, the algorithm is less sensitive to noise and outliers, compared to k-means, because it uses medoids as cluster centers instead of means (used in k-means).





Steps involved in K-Medoid Clustering

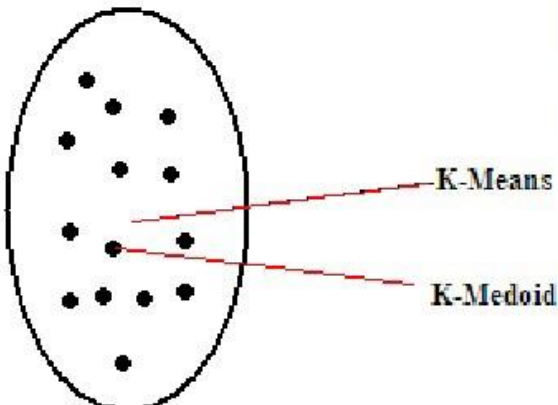
Steps involved in K-Medoids Clustering :

1. The PAM algorithm is based on the search for k representative objects or medoids among the observations of the data set.
2. After finding a set of k medoids, clusters are constructed by assigning each observation to the nearest medoid.
3. Next, each selected medoid m and each non-medoid data point are swapped and the objective function is computed. The objective function corresponds to the sum of the dissimilarities of all objects to their nearest medoid.

- The SWAP step attempts to improve the quality of the clustering by exchanging selected objects (medoids) and non-selected objects. If the objective function can be reduced by interchanging a selected object with an unselected object, then the swap is carried out. This is continued until the objective function can no longer be decreased. The goal is to find  $k$  representative objects which minimize the sum of the dissimilarities of the observations to their closest representative object.

#### Difference between K-Means & K-Medoids Clustering -

## K-Means & K-medoid



Cluster ==> 2 5 6 9 13

**K-mean's Centroid = 7**

**K-medoid's Centroid = 6**

K-means clustering use the exact center of a cluster (means or the center of gravity) while K-medoid uses the most centrally located object in a cluster (medoid).

K-medoid is less sensitive to outliers Compared to K-means.

K value (number of clusters) has to be determined a-priori.

- $K$ -means attempts to minimize the total [squared error](#), while  $k$ -medoids minimizes the sum of dissimilarities between points labeled to be in a cluster and a point designated as the center of that cluster. In contrast to the  $k$ -means algorithm,  $k$ -medoids chooses data points as centers ( [medoids](#) or exemplars).
- $K$ -medoid is more robust to noise and outliers as compared to  $K$ -means because it minimizes a sum of general pairwise dissimilarities instead of a sum of squared Euclidean distances.

3. K-medoids has the advantage of working on **distances other than numerical** and lends itself well to analyse mixed-type data that include both numerical and categorical features.

k-means	k-medoids
Complexity is $O(ikn)$	Complexity is $O(i k(n-k)^2)$
More efficient	Comparatively less efficient
Sensitive to outliers	Not Sensitive to outliers
Convex shape is required	Convex shape is not must
Number of clusters need to be specified in advance	Number of clusters need to be specified in advance
Efficient for separated clusters	Efficient for separated clusters and small data sets

Difference between k-means and k-medoids

### **Hierarchical Clustering**

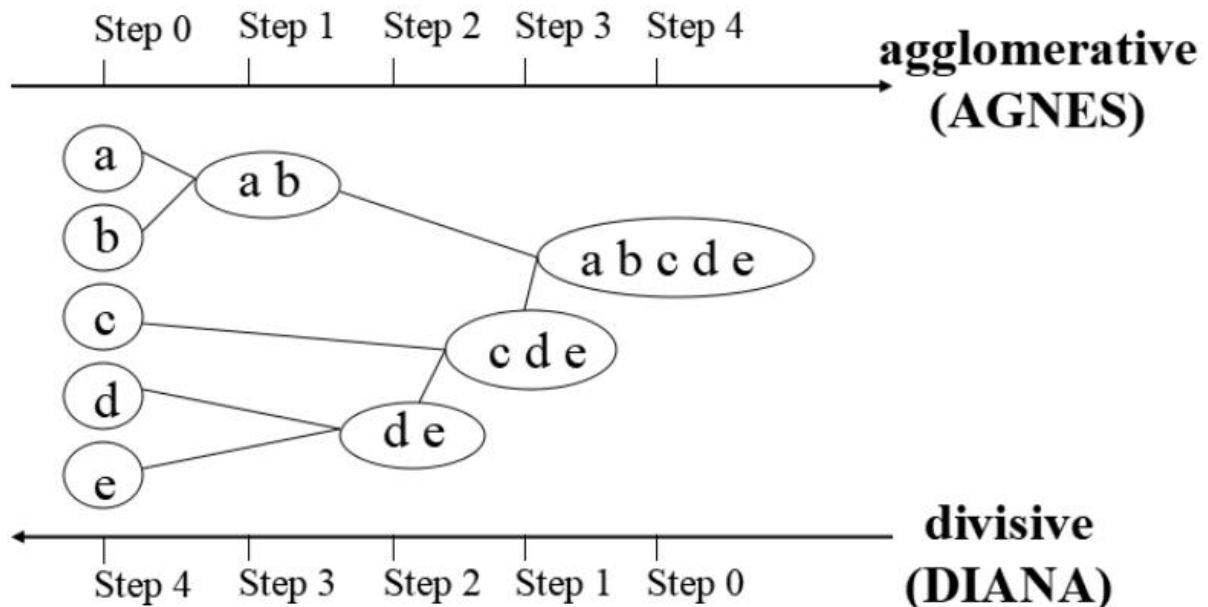
A hierarchical [clustering method](#) works by grouping data objects into a tree of clusters. Hierarchical clustering methods can be further classified into agglomerative and divisive hierarchical clustering, depending on whether the hierarchical decomposition is formed in a bottom-up or top-down fashion.

#### **(AGNES) Agglomerative Hierarchical Clustering:**

This bottom-up strategy starts by placing each object in its own cluster and then merges these atomic clusters into larger and larger clusters until all of the objects are in a single cluster or until certain termination conditions are satisfied.

#### **(DIANA) Divisive Hierarchical Clustering:**

This top-down strategy does the reverse of agglomerative clustering by starting with all objects in one cluster. It subdivides the cluster into smaller and smaller pieces until each object forms a cluster on its own or until it satisfies certain termination conditions such as a desired number of clusters is obtained or the distance between the two closest clusters is above a certain threshold distance.



Data set of five objects a, b, c, d. Initially, AGNES places each object into a cluster of its own.

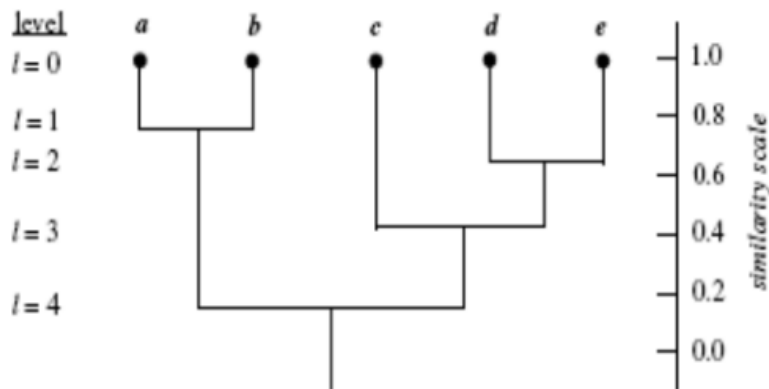
The clusters are then merged step-by-step according to some criterion.

**Decompose data objects into several levels of nested partitioning (tree of clusters), called a dendrogram.**

**A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.** It shows how objects are grouped together step by step.

The figure shows a dendrogram for the five objects presented in previous Fig, where  $l = 0$  shows the five objects as singleton clusters at level 0.

At  $l = 1$ , objects a and b are grouped together to form the first cluster, and they stay together at all subsequent levels. We can also use a vertical axis to show the similarity scale between clusters. For example, when the similarity of two groups of objects, {a, b} and {c, d, e} is roughly 0.16, they are merged together to form a single cluster.



7.7 Dendrogram representation for hierarchical clustering of data objects  $\{a, b, c, d, e\}$ .

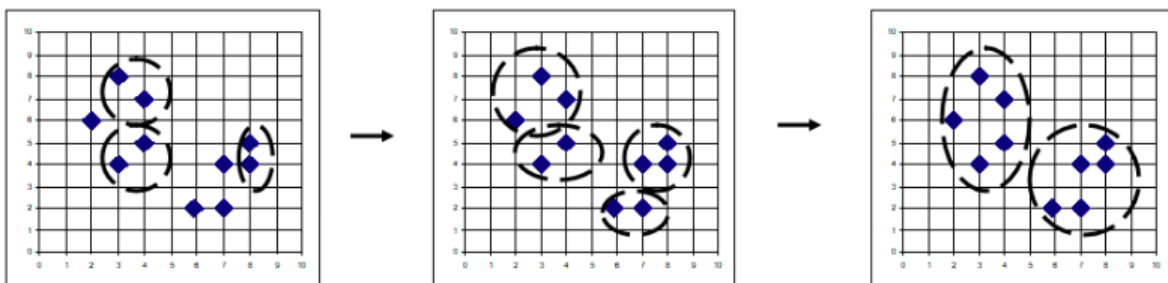
### Agglomerative Nesting(AGNES)

This clustering method was Introduced by Kaufmann and Rousseeuw (1990).

It is generally implemented in statistical analysis packages, e.g., Splus.

It uses the [Single-Link](#) method and the dissimilarity matrix.

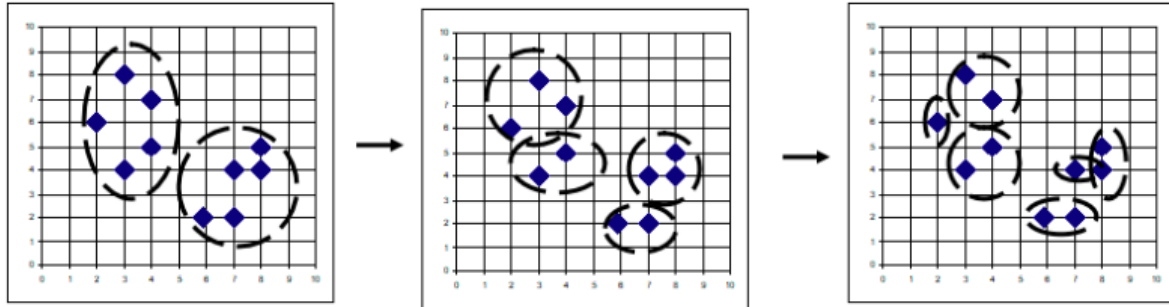
It merge nodes that have the least dissimilarity this process goes on in a non-descending fashion and eventually all nodes belong to the same cluster.



### Divisive Analysis(DIANA)

It was also introduced by Kaufmann and Rousseeuw (1990).

It is also similar to that of Agglomerative Clustering but the way of application is different i.e in the Reverse way of AGNES.It is also implemented in statistical analysis packages, e.g., Splus.



### Disadvantages Of Hierarchical Clustering

The hierarchical clustering method, though simple, often encounters difficulties regarding the selection of merge or split points. Such a decision is critical because once a group of objects is merged or split, the process at the next step will operate on the newly generated clusters.

It will neither undo what was done previously nor perform object swapping between clusters.

Thus merge or split decisions, if not well chosen at some step, may lead to low-quality clusters. Moreover, the method does not scale well, because each decision to merge or split requires the examination and evaluate a good number of objects or clusters.

More On Hierarchical Methods Are

**BIRCH (1996):** Uses CF-tree and incrementally adjusts the quality of sub-clusters.

**CURE (1998):** Selects well-scattered points from the cluster and then shrinks them towards the center of the cluster by a specified fraction.

**CHAMELEON (1999):** Hierarchical clustering using dynamic modeling.

### Density-Based Clustering

Density-Based Clustering method is one of the clustering methods based on density (local cluster criterion), such as density-connected points.

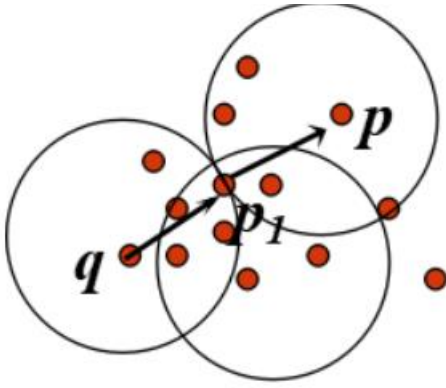
The basic ideas of density-based clustering involve a number of new definitions. We intuitively present these definitions and then follow up with an example.

The neighborhood within a radius  $\epsilon$  of a given object is called the  $\epsilon$ -neighborhood of the object.

If the  $\epsilon$ -neighborhood of an object contains at least a minimum number, MinPts, of objects, then the object is called a core object.

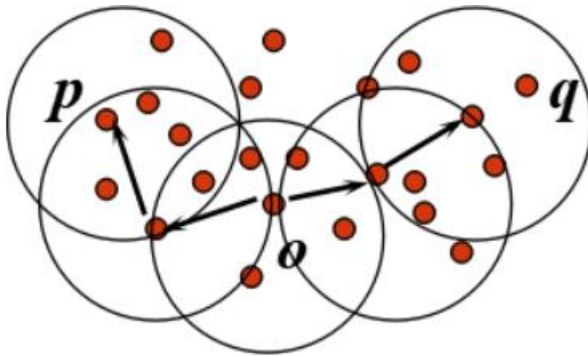
**Density-reachable:**

- A point  $p$  is density-reachable from a point  $q$  wrt.  $\text{Eps}$ ,  $\text{MinPts}$  if there is a chain of points  $p_1, \dots, p_n, p_1 = q, p_n = p$  such that  $p_{i+1}$  is directly density-reachable from  $p_i$



### Density-connected

- A point  $p$  is density-connected to a point  $q$  wrt.  $\text{Eps}$ ,  $\text{MinPts}$  if there is a point  $o$  such that both,  $p$  and  $q$  are density-reachable from  $o$  wrt.  $\text{Eps}$  and  $\text{MinPts}$ .



### Working Of Density-Based Clustering

Given a set of objects,  $D'$  we say that an object  $p$  is directly density-reachable from object  $q$  if  $p$  is within the  $\epsilon$ -neighborhood of  $q$ , and  $q$  is a core object.

An object  $p$  is density-reachable from object  $q$  with respect to  $\epsilon$  and  $\text{MinPts}$  in a set of objects,  $D'$  if there is a chain of objects  $p_1, \dots, p_n$ , where  $p_1 = q$  and  $p_n = p$  such that  $p_{i+1}$  is directly density-reachable from  $p_i$  with respect to  $\epsilon$  and  $\text{MinPts}$ , for  $1/n, p_i \in D$ .

An object  $p$  is density-connected to object  $q$  with respect to  $\epsilon$  and  $\text{MinPts}$  in a set of objects,  $D'$ , if there is an object  $o$ , belongs  $D$  such that both  $p$  and  $q$  are density-reachable from  $o$  with respect to  $\epsilon$  and  $\text{MinPts}$ .

### Density-Based Clustering - Background

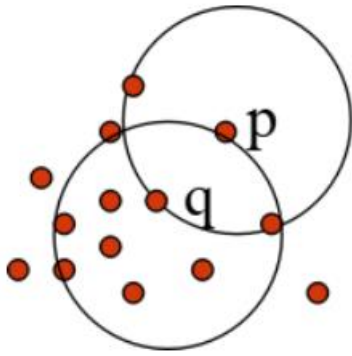
Two parameters:

- $\text{Eps}$ : Maximum radius of the neighborhood.
- $\text{MinPts}$ : Minimum number of points in an  $\text{Eps}$ -neighbourhood of that point.

$\text{NEps}(p)$ :  $\{q \text{ belongs to } D \mid \text{dist}(p,q) \leq \text{Eps}\}$

Directly density-reachable: A point  $p$  is directly density-reachable from a point  $q$  wrt.  $\text{Eps}$ ,  $\text{MinPts}$  if

- $p$  belongs to  $NEps(q)$
- core point condition:  $|NEps(q)| \geq MinPts$



$MinPts = 5$

$Eps = 1 \text{ cm}$

### Major features:

It is used to discover clusters of arbitrary shape.

It is also used to handle noise in the data clusters.

It is a one scan method.

It needs density parameters as a termination condition.

### Density-Based Methods

**DBSCAN:** Ester, et al. (KDD'96)

**OPTICS:** Ankerst, et al (SIGMOD'99).

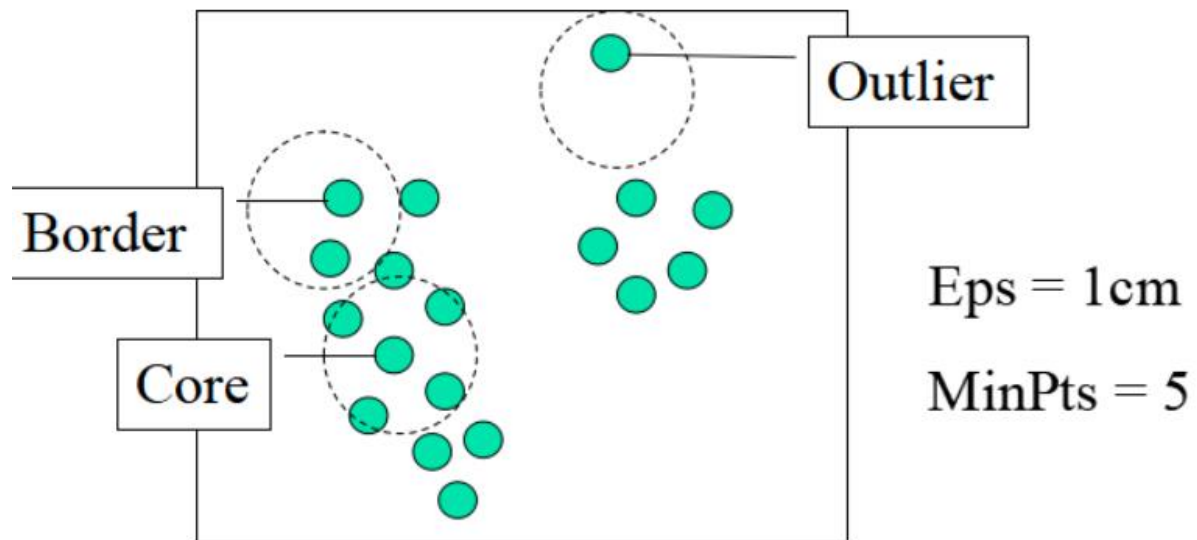
**DENCLUE:** Hinneburg & D. Keim (KDD'98)

**CLIQUE:** Agrawal, et al. (SIGMOD'98)

### DBSCAN(Density-Based Spatial Clustering of Applications with Noise)

It relies on a density-based notion of cluster: A cluster is defined as a maximal set of density-connected points. It discovers clusters of arbitrary shape in spatial databases with noise.





### DBSCAN Algorithm

Arbitrary select a point  $p$ .

Retrieve all points density-reachable from  $p$  wrt  $Eps$  and  $MinPts$ .

If  $p$  is a core point, a cluster is formed.

If  $p$  is a border point, no points are density-reachable from  $p$  and DBSCAN visits the next point of the database.

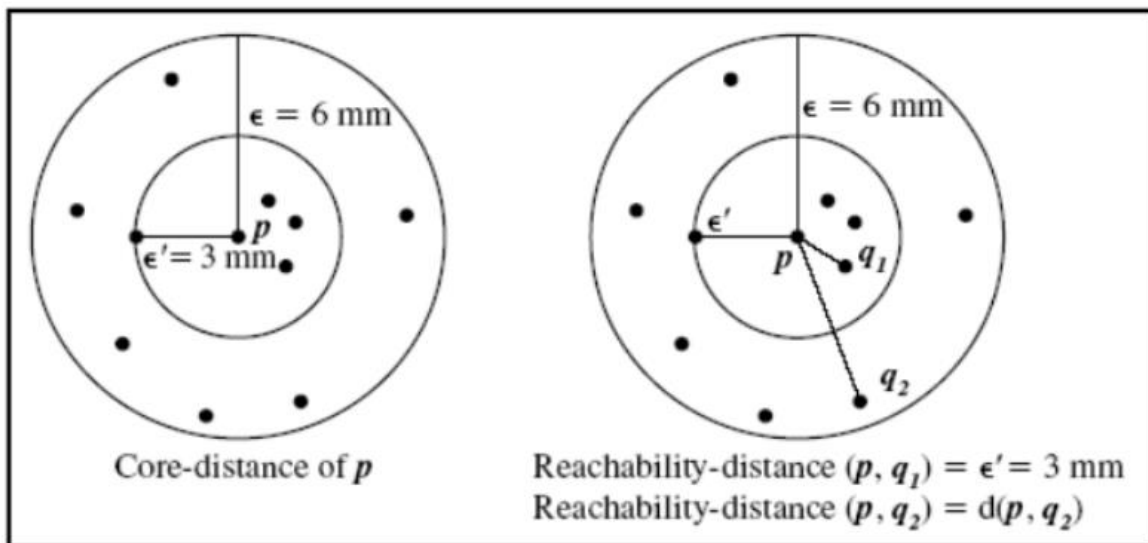
Continue the process until all of the points have been processed.

### OPTICS - A Cluster-Ordering Method

OPTICS: Ordering Points To Identify the Clustering Structure.

- It produces a special order of the database with respect to its density-based clustering structure.
- This cluster-ordering contains info equivalent to the density-based clusterings corresponding to a broad range of parameter settings.

- It is good for both automatic and interactive cluster analysis, including finding an intrinsic clustering structure.
- It can be represented graphically or using visualization techniques.



**Core-distance and reachability-distance:** The figure illustrates the concepts of core-distance and reachability-distance.

Suppose that  $\epsilon = 6 \text{ mm}$  and  $\text{MinPts} = 5$ .

The core distance of  $p$  is the distance,  $\epsilon_0$ , between  $p$  and the fourth closest data object.

The reachability-distance of  $q_1$  with respect to  $p$  is the core-distance of  $p$  (i.e.,  $\epsilon_0 = 3 \text{ mm}$ ) because this is greater than the Euclidean distance from  $p$  to  $q_1$ .

The reachability distance of  $q_2$  with respect to  $p$  is the Euclidean distance from  $p$  to  $q_2$  because this is greater than the core-distance of  $p$ .

DENCLUE - Using Density Functions

DENSity-based CLUStEring by Hinneburg & Keim (KDD'98)

### Major Features

- It has got a solid mathematical foundation.
- It is definitely good for data sets with large amounts of noise.
- It allows a compact mathematical description of arbitrarily shaped clusters in high-dimensional data sets.
- It is significantly faster than the existing algorithm (faster than DBSCAN by a factor of up to 45).
- But it needs a large number of parameters.

### DENCLUE - Technical Essence

It uses grid cells but only keeps information about grid cells that do actually contain data points and manages these cells in a tree-based access structure.

### Influence function:

This describes the impact of a data point within its neighborhood.

The Overall density of the data space can be calculated as the sum of the influence function of all data points. The Clusters can be determined mathematically by identifying density attractors. The Density attractors are local maxima of the overall density function.

### Grid-Based Clustering

In Grid-Based Methods, the space of instance is divided into a grid structure. Clustering techniques are then applied using the Cells of the grid, instead of individual data points, as the base units. The biggest advantage of this method is to improve the processing time. Grid-Based Clustering method uses a multi-resolution grid data structure.

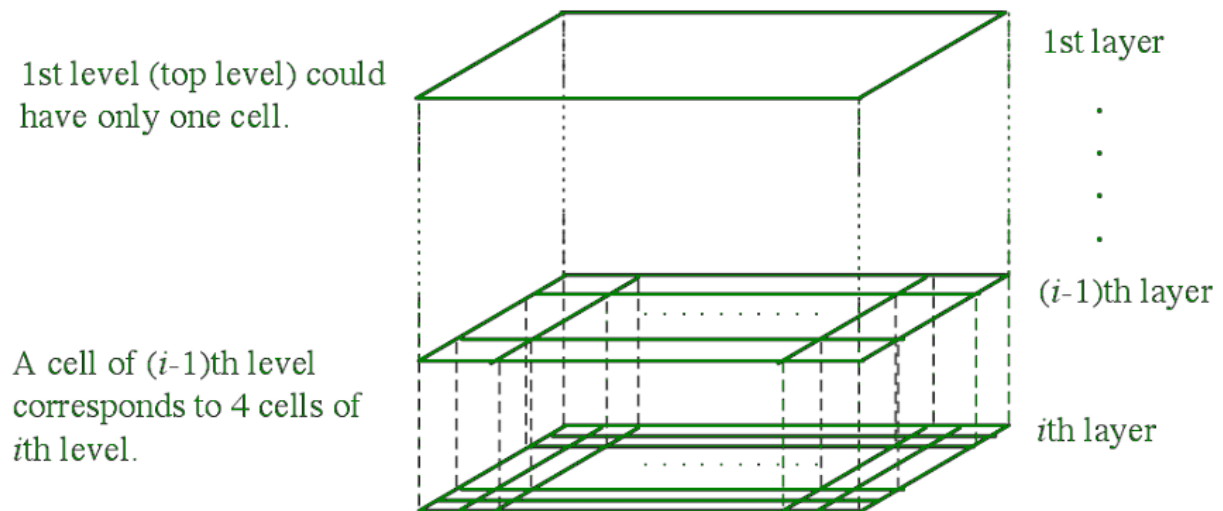
#### Several interesting methods

- **STING** (a **ST**atistical **IN**formation **Grid** approach) by Wang, Yang, and Muntz (1997)
- **WaveCluster** by Sheikholeslami, Chatterjee, and Zhang (VLDB'98) - A multi-resolution clustering approach using wavelet method
- **CLIQUE** - Agrawal, et al. (SIGMOD'98)
- **STING - A Statistical Information Grid Approach**

STING was proposed by Wang, Yang, and Muntz (VLDB'97).

In this method, the spatial area is divided into rectangular cells

There are several levels of cells corresponding to different levels of resolution.



For each cell, the high level is partitioned into several smaller cells in the next lower level.

The statistical info of each cell is calculated and stored beforehand and is used to answer queries.

The parameters of higher-level cells can be easily calculated from parameters of lower-level cell

- Count, mean, s, min, max
- Type of distribution—normal, uniform, etc.

Then using a top-down approach we need to answer spatial data queries. Then start from a pre-selected layer—typically with a small number of cells. For each cell in the current level compute the confidence interval. Now remove the irrelevant cells from further consideration. When finishing examining the current layer, proceed to the next lower level. Repeat this process until the bottom layer is reached.

**Advantages:**

It is Query-independent, easy to parallelize, incremental update.

$O(K)$ , where  $K$  is the number of grid cells at the lowest level.

**Disadvantages:**

All the cluster boundaries are either horizontal or vertical, and no diagonal boundary is detected.

**WaveCluster**

It was proposed by Sheikholeslami, Chatterjee, and Zhang (VLDB'98).

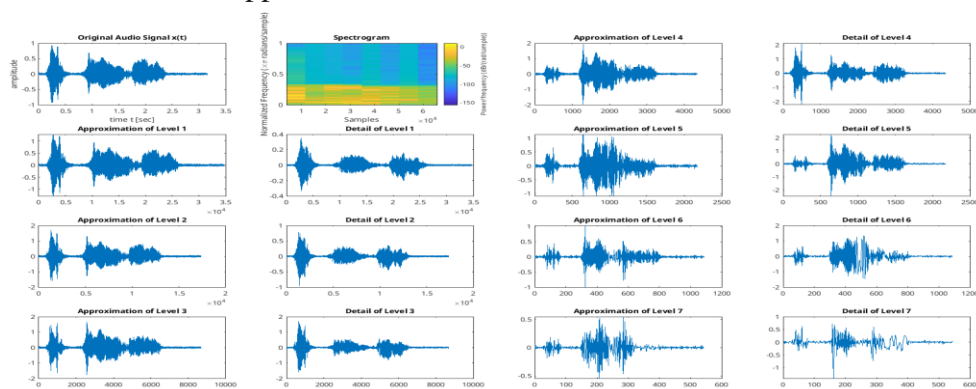
It is a multi-resolution clustering approach which applies wavelet transform to the feature space

- A wavelet transform is a signal processing technique that decomposes a signal into different frequency sub-band.

It can be both grid-based and density-based method.

**Input parameters:**

- No of grid cells for each dimension
- The wavelet, and the no of applications of wavelet transform.



**How to apply the wavelet transform to find clusters**

- It summarizes the data by imposing a multidimensional grid structure onto data space.
- These multidimensional spatial data objects are represented in an n-dimensional feature space.
- Now apply wavelet transform on feature space to find the dense regions in the feature space.
- Then apply wavelet transform multiple times which results in clusters at different scales from fine to coarse.

## Why is wavelet transformation useful for clustering

- It uses hat-shape filters to emphasize region where points cluster, but simultaneously to suppress weaker information in their boundary.
- It is an effective removal method for outliers.
- It is of Multi-resolution method.
- It is cost-efficiency.

## Major features:

- The time complexity of this method is  $O(N)$ .
- It detects arbitrary shaped clusters at different scales.
- It is not sensitive to noise, not sensitive to input order.
- It only applicable to low dimensional data.

## CLIQUE - Clustering In QUES

It was proposed by Agrawal, Gehrke, Gunopulos, Raghavan (SIGMOD'98).

It is based on automatically identifying the subspaces of high dimensional data space that allow better clustering than original space.

CLIQUE can be considered as both density-based and grid-based:

- It partitions each dimension into the same number of equal-length intervals.
- It partitions an  $m$ -dimensional data space into non-overlapping rectangular units.
- A unit is dense if the fraction of the total data points contained in the unit exceeds the input model parameter.
- A cluster is a maximal set of connected dense units within a subspace.

**Partition the data space and find the number of points that lie inside each cell of the partition.**

**Identify the subspaces that contain clusters using the [Apriori](#) principle.**

## Identify clusters:

- Determine dense units in all subspaces of interests.
- Determine connected dense units in all subspaces of interests.

## Generate minimal description for the clusters:

- Determine maximal regions that cover a cluster of connected dense units for each cluster.
- Determination of minimal cover for each cluster.

## Advantages

It automatically finds subspaces of the highest dimensionality such that high-density clusters exist in those subspaces.

It is insensitive to the order of records in input and does not presume some canonical data distribution.

It scales linearly with the size of input and has good scalability as the number of dimensions in the data increases.

### **Disadvantages**

The accuracy of the clustering result may be degraded at the expense of the simplicity of the method.

<b>Method</b>	<b>General Characteristics</b>
Partitioning methods	<ul style="list-style-type: none"><li>– Find mutually exclusive clusters of spherical shape</li><li>– Distance-based</li><li>– May use mean or medoid (etc.) to represent cluster center</li><li>– Effective for small- to medium-size data sets</li></ul>
Hierarchical methods	<ul style="list-style-type: none"><li>– Clustering is a hierarchical decomposition (i.e., multiple levels)</li><li>– Cannot correct erroneous merges or splits</li><li>– May incorporate other techniques like microclustering or consider object “linkages”</li></ul>
Density-based methods	<ul style="list-style-type: none"><li>– Can find arbitrarily shaped clusters</li><li>– Clusters are dense regions of objects in space that are separated by low-density regions</li><li>– Cluster density: Each point must have a minimum number of points within its “neighborhood”</li><li>– May filter out outliers</li></ul>
Grid-based methods	<ul style="list-style-type: none"><li>– Use a multiresolution grid data structure</li><li>– Fast processing time (typically independent of the number of data objects, yet dependent on grid size)</li></ul>