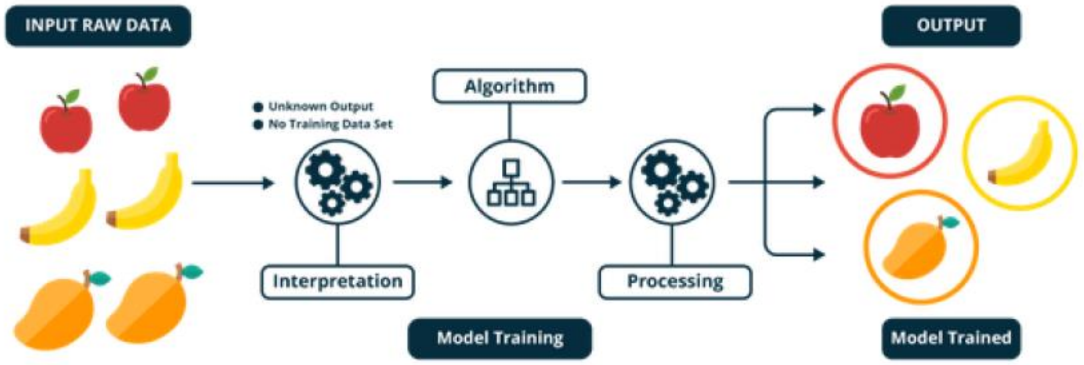
Unsupervised learning

Unsupervised learning

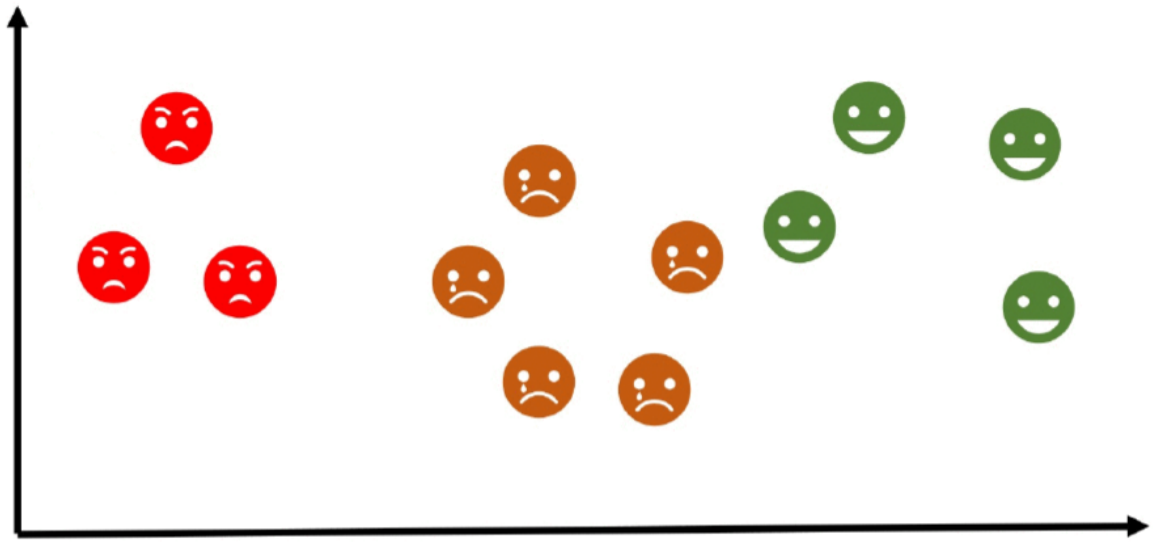
When we have data with no labeled response variable in data set.



Inference is drawn from dataset consisting of only input data.

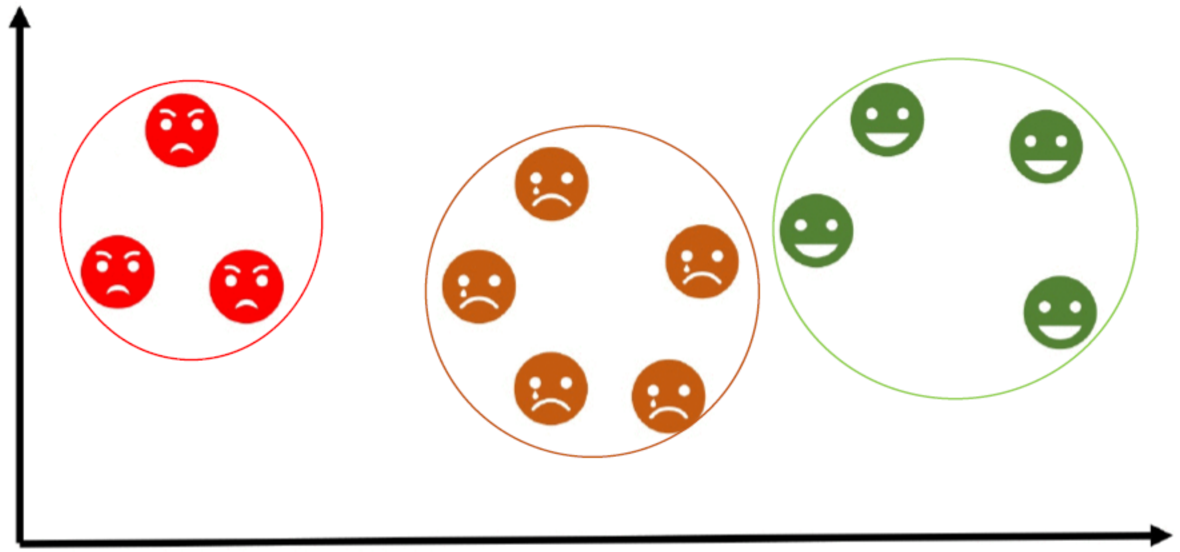
What is Clustering?

Unsupervised learning in which data is organized into distinct groups , based on similarity they exhibit.



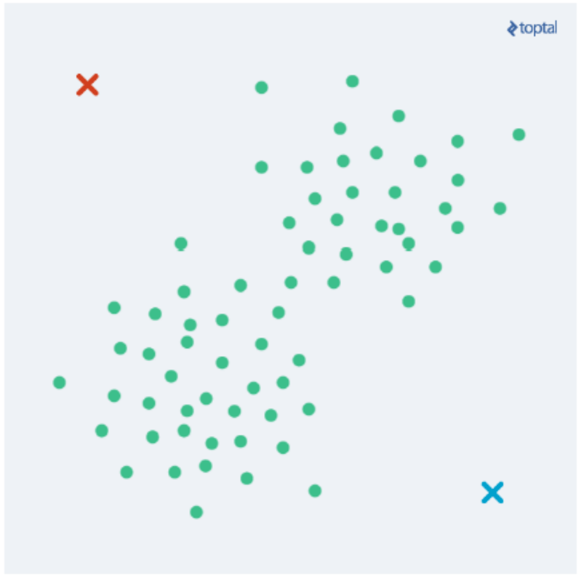
What is Clustering?

After organizing data points into distinct clusters, we observe similarities within the clusters.



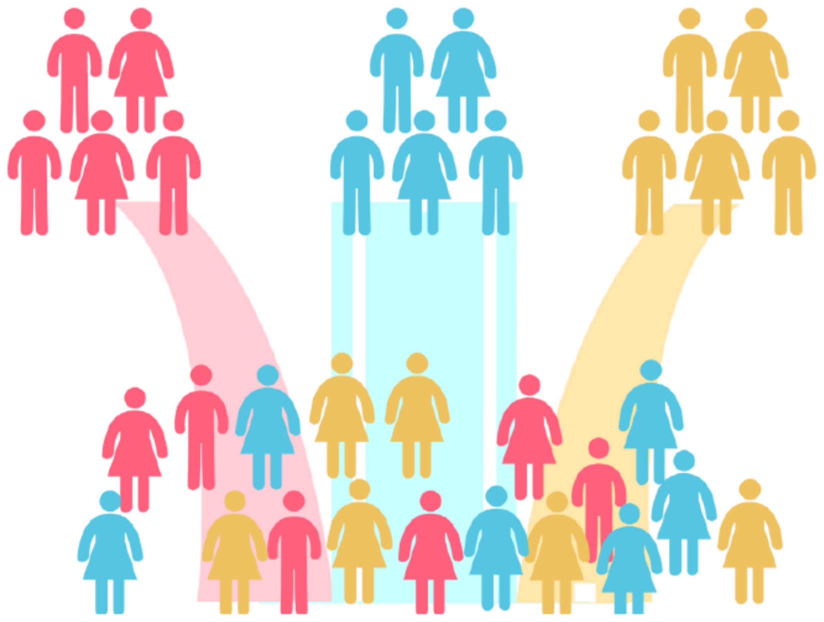
Why Clustering?

- Clustering helps organize unknown observations into groups based on similarity.
- We do clustering when there is no labeled variable(dependent column) in data set.
- It helps us find patterns. Widely used for initial analysis of data.



Customer Segmentation
Segmenting Customers based on purchasing behavior

Segmenting Customers based on purchasing behavior



Example:

1. A car maker has **3 variant models** of car viz. **SUV , Sedan & Hatchback.**

2. Based on purchase of 3 models **offering different specification at different prices** we can **segment** the buyers.

3. Segmenting buyers using clustering helps us **assign next buyers** to those clusters.

Application of Clustering: Medicine

Segmenting Medicines based on constituents, reactions , function & results

Example:

Antibiotic	– Fights bacterial infection
Antacid	– Reduces acid in stomach
Analgesic	– Alleviate pain
Antidepressant	– Reduce depression & anxiety
Beta blocker	– Reduces BP

Application of Clustering: Psychology

Segmenting moms

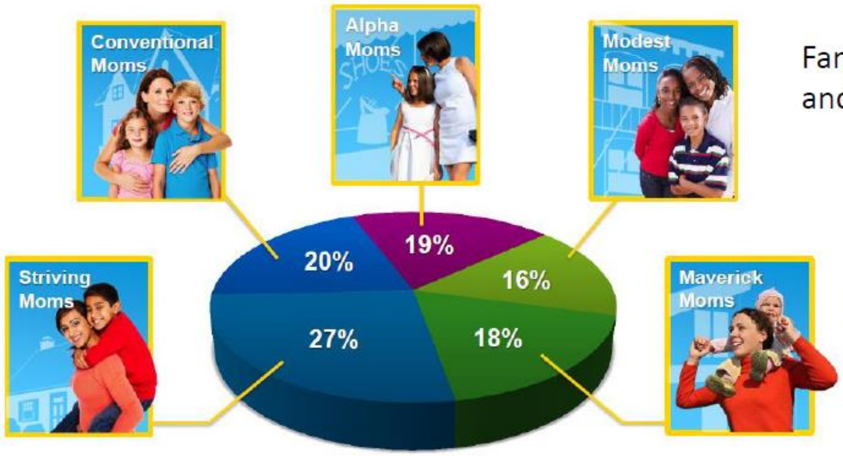
Sophisticated,
Youthful appearance

Full-time job,
Faith plays an important role

Family and faith-oriented. Modest and simple.

Young, diverse.
Confident & aspiring.

Independent trailblazers,
Brave.



Source: Experian Marketing Services

Main Topics

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
   - Partitioning Methods
   - Hierarchical Methods

More on Clustering

- In clustering class label of each object is not known. (unsupervised learning)
- The process of grouping a set of 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 and are *dissimilar* to the objects in other clusters.
- Object within a cluster have high similarity in comparison to one another but are very dissimilar to objects in other clusters.

Clustering

- Clustering is also called **data segmentation** in some applications because clustering partitions large data sets into groups according to their *similarity*.
- It can be used for outlier detection where outliers (values that are “far away” from any cluster) may be more interesting than common cases.

What is Cluster Analysis?

- Cluster?: a collection of data objects
  - Similar to one another within the same cluster.
  - Dissimilar to the objects in other clusters.

- Cluster analysis
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters.
- Unsupervised learning: no predefined classes

More Examples of Clustering Applications

- Marketing: Customer groups based on purchasing patterns.
- In biology: it can be used to derive animal taxonomies, categorize genes with similar functionality.
- 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.

Quality: What is good clustering?

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

Requirements of Clustering

The following are the typical requirements of clustering in Machine Learning:-

- Scalability
  - **highly scalable clustering algorithms are needed to handle the millions of data object**
- Ability to deal with different types of attributes
  - **data may be numeric or other type of data**
- Discovery of clusters with arbitrary shape
  - **Many clustering algorithms determine clusters based on Euclidean or Manhattan distance measures**
  - **Algorithms based on such distance measures tend to find spherical clusters with similar size and density**
  - **A cluster could be of any shape**
  - **It is important to develop algorithms that can detect clusters of arbitrary shape**
- Minimal requirements for domain knowledge to determine input parameters
- Ability to deal with noise and outliers
  - **Most real-world databases contain outliers or missing, unknown, or erroneous data**
  - **Some clustering algorithms are sensitive to such data**
  - **and may lead to clusters of poor quality**
- Incremental clustering and insensitivity to the order of input records
  - **It is important to develop incremental clustering algorithms and algorithms that are insensitive to the order of input**
- High dimensionality
- Constraint-based clustering
- Interpretability and usability
  - **clustering results to be interpretable and usable**

Types of Data in Cluster Analysis Data Structures

- What types of data that often occur in cluster analysis and how to preprocess them for such an analysis?
- Suppose that a data set to be clustered contains *n objects, which may represent persons, houses, documents, countries, and so on.*
- Main clustering algorithms typically operate on either of the following two data structures.
  - Data matrix (or *object-by-variable structure)*
  - Dissimilarity matrix (or *object-by-object structure)*

Types of Data in Cluster Analysis 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 p variables):*

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

Types of Data in Cluster Analysis Data Structures

- 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:

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

- where *d(i, j) is the measured difference or dissimilarity between objects i and j. In* general, *d(i, j) is a nonnegative number.*
- 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.

Similarity and Dissimilarity Between Objects

- Distances are normally used to measure the similarity or dissimilarity between two data objects
- Some popular ones include: *Minkowski distance*:

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

where $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ are two q -dimensional data objects, and p is a positive integer

- If p = *1*, *d* is Manhattan distance

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

Similarity and Dissimilarity Between Objects (Cont.)

- *If p = 2, d* is Euclidean distance:

$$d(i,j) = (|x_{i1}-x_{j1}|^2 + |x_{i2}-x_{j2}|^2 + \dots\dots\dots + |x_{in}-x_{jn}|^2)^{1/2}$$

  - Properties
    - $d(i,j) \geq 0$
    - $d(i,i) = 0$
    - $d(i,j) = d(j,i)$
    - $d(i,j) \leq d(i,k) + d(k,j)$

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

- A contingency table for binary data

		Object *j*		
		1	0	*sum*
Object *i*	1	a	b	$a+b$
	0	c	d	$c+d$
	sum	$a+c$	$b+d$	p

- Distance measure for symmetric binary variables:

$$d(i,j) = \frac{b+c}{a+b+c+d}$$

- Distance measure for asymmetric binary variables:

$$d(i,j) = \frac{b+c}{a+b+c}$$

- 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. One such example could be the attribute *gender having the states male and female.*
- A binary variable is asymmetric if the outcomes of the states are not equally important, such as the *positive and negative outcomes of a disease test. By convention,* we shall code the most important outcome, which is usually the rarest one, by 1 (e.g., *HIV positive) and the other by 0 (e.g., HIV negative).*

Dissimilarity between Binary Variables

		Object j		
		1	0	sum
Object i	1	a	b	$a+b$
	0	c	d	$c+d$
	sum	$a+c$	$b+d$	p

- Example

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

  - gender is a symmetric attribute
  - the remaining attributes are asymmetric binary
  - let the values Y and P be set to 1, and the value N be set to 0

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(i,j) = \frac{b+c}{a+b+c}$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

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

Major Clustering Approaches

- Partitioning approach:
  - Given a database of n objects or data tuples, a partitioning method constructs k partitions of the data, where each partition represents a cluster and k ≤ n.
  - It classifies the data into k groups, which together satisfy the following requirements.
  1. Each group must contains at least on object.
  2. Each object must belong to exactly one group.
  - Typical methods: k-means, k-medoids, CLARANS

Partitioning Algorithms: Basic Concept

- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters.

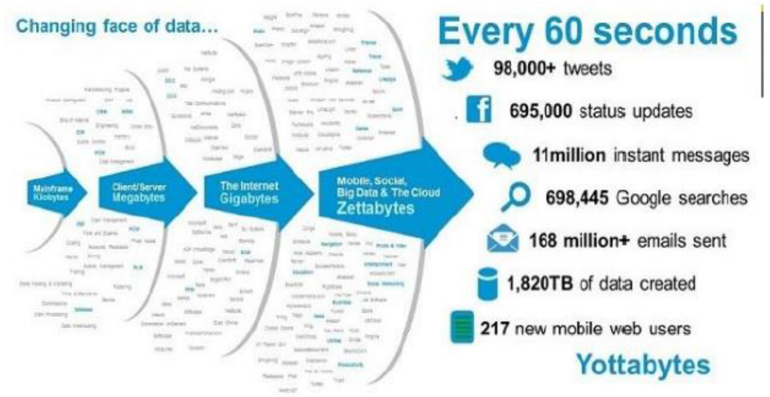
- Most applications adopt one of a few popular Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means* (MacQueen'67): where each cluster is represented by the mean value of the objects in the cluster.
  - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): where each cluster is represented by one of the objects located near the centre of the cluster.

Why K-means clustering?

- K-means clustering performs better than Hierarchical clustering(discussed in reference slides) in large data set.
- Because as **data size increases computational time** for Hierarchical clustering increases.

In Hierarchical clustering:

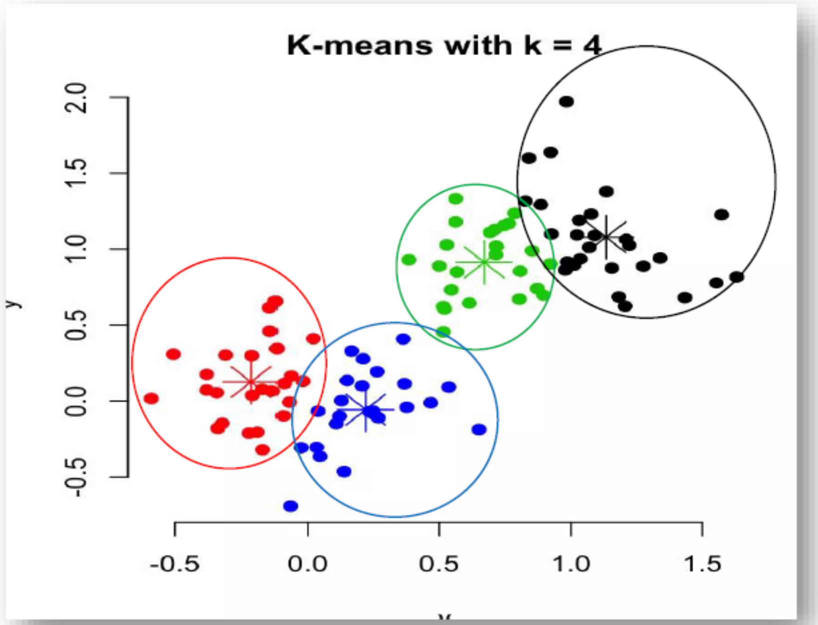
Data size increases



Computational time increases

What is K-means clustering?

- An unsupervised learning technique
- In which data is organized into distinct groups having centroids(Mean values).
- K denotes number of clusters or groups.



A simple walkthrough of K-means clustering

Step 1: Choose the value of K– where K indicates number of clusters(we'll discuss later how to choose optimal K)

Step 2: Initialize mean or centroid of each cluster taking random values

Step 3: Calculate Euclidean distance of each attribute for each observation from each centroid.

Step 4: Based on nearest distance from centroids, assign observations to clusters.

Step 5: After each assignment recalculate mean of each attribute across all clusters.

Step 6: Repeat step **3**, **4** & **5** until convergence of centroids i.e. centroids don't change significantly.

The *K-Means* Clustering Method

- Given *k*, the *k-means* algorithm is implemented in four steps:
  - Partition objects into *k* nonempty subsets
  - Compute seed points as the centroids of the clusters of the current partition (the centroid is the center, i.e., *mean point*, of the cluster)
  - An object is assigned to the cluster to which it is the most similar, based on the distance between the object and the cluster mean.
  - It then computes the new mean for each cluster.
  - Stop when no more new assignment

Algorithm: *k-means*

Algorithm: *k***-means.** The *k*-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

Input:

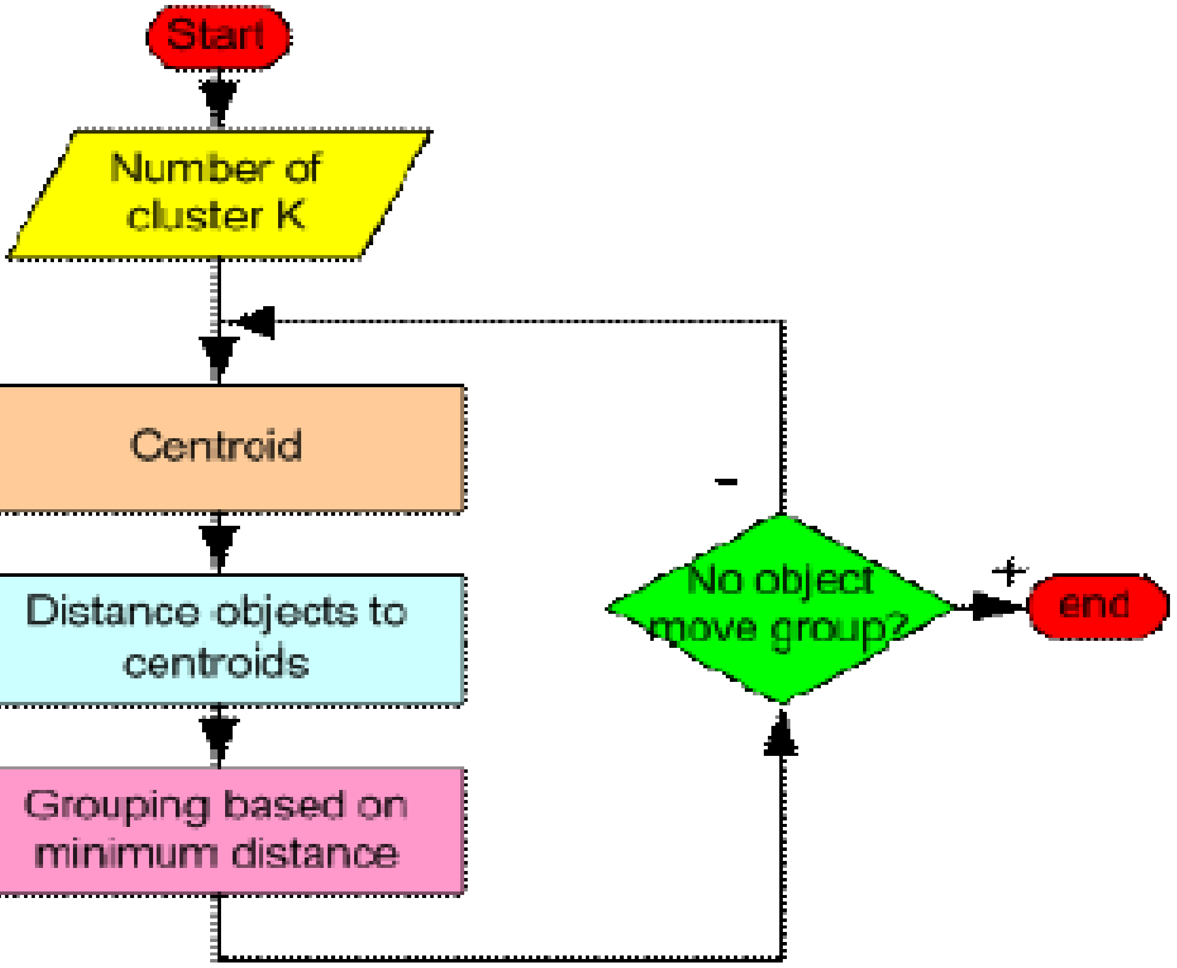
- *k*: the number of clusters,
- *D*: a data set containing *n* objects.

Output: A set of *k* clusters.

Method:

```
(1) arbitrarily choose $k$ objects from $D$ as the initial cluster centers;
(2) repeat
(3)       (re)assign each object to the cluster to which the object is the most similar,
                based on the mean value of the objects in the cluster;
(4)       update the cluster means, i.e., calculate the mean value of the objects for
                each cluster;
(5) until no change;
```

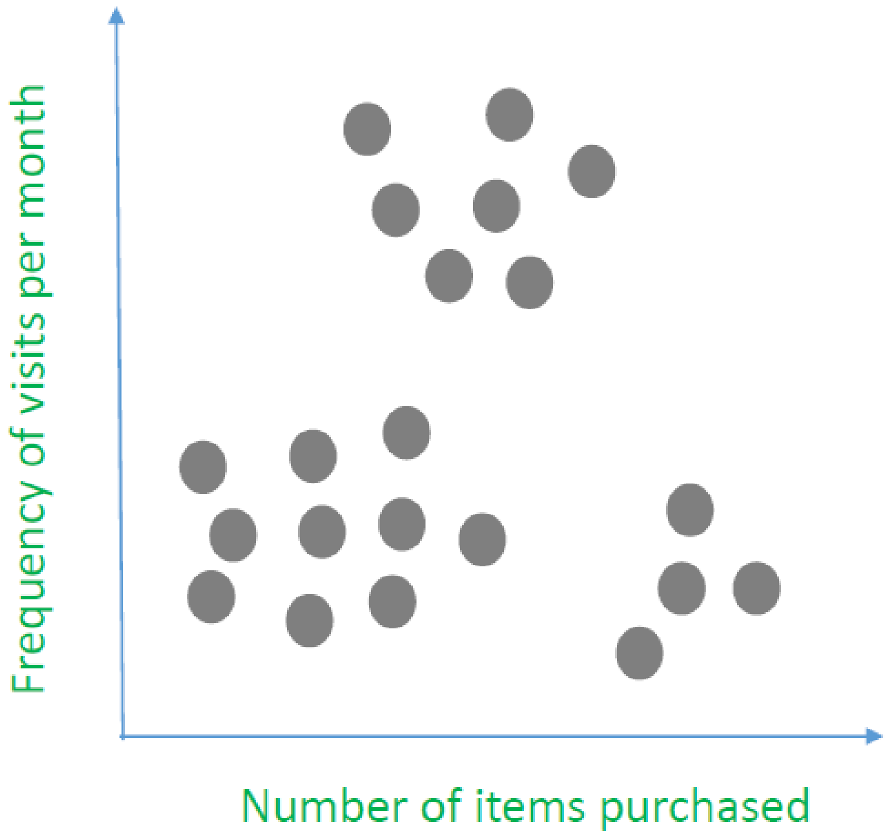
The *K-Means* Clustering Method



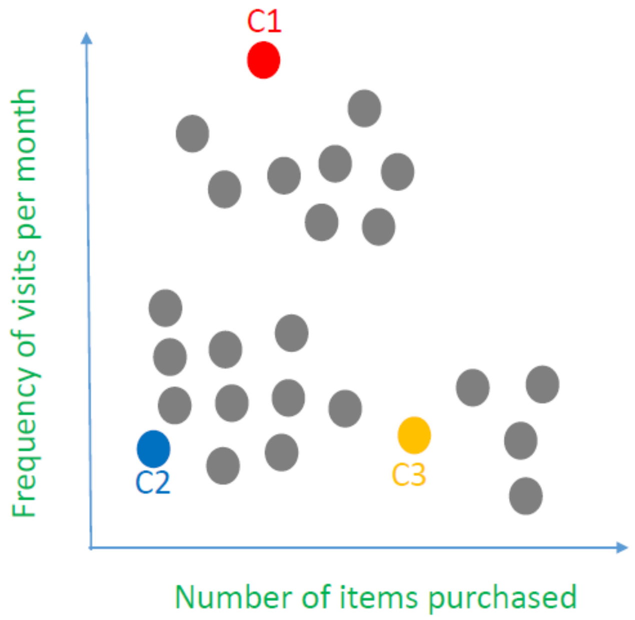
Start
Number of cluster K
Centroid
Distance objects to centroids
Grouping based on minimum distance
No object move group?
-
+
end

Step 1: Pick the value of K(Number of clusters)

We arbitrarily take K=3

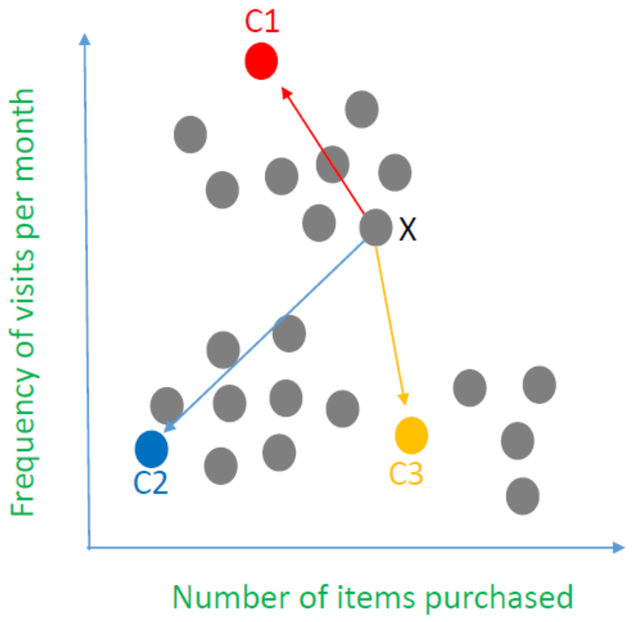


Step 2: Initialize Centroids or Means randomly



Randomly initialize Centroids for 3 clusters with C1, C2 & C3

distance of each data point fromall centroids

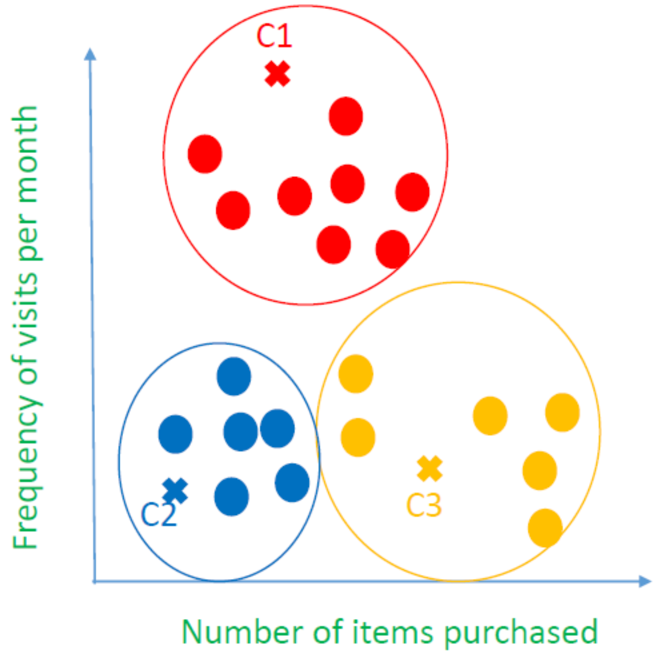


- Let’s say
  X is a data point &
  C1 is centroid of a cluster

- Now calculate Euclidean distance of X from C1.
  Euclidean distance(X,C1) = $\sqrt{(X - C1)^2}$

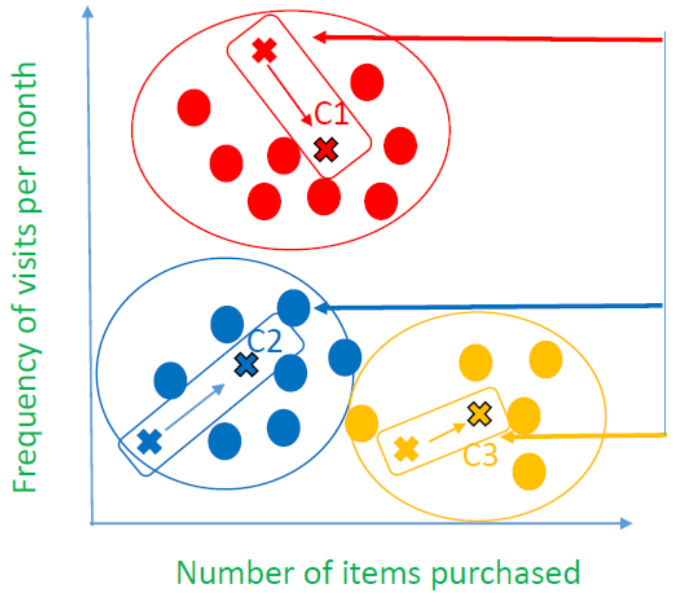
- This way we can **calculate** Euclidean distance of all data points from each centroid.

Step 4: Assigning data points to clusters



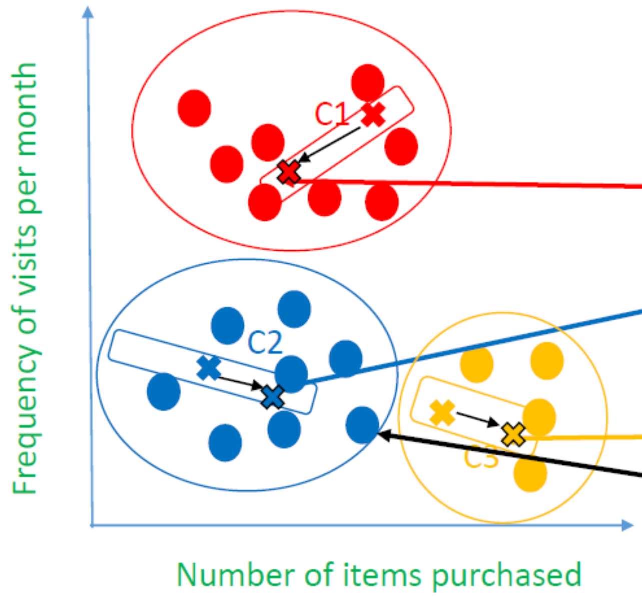
- **Euclidean distance** is measured between each data point & all centroids
- Based on nearest Euclidean distance between centroids & data points assign data points to respective clusters.

Step 5: Recalculating mean &assigning data points to clusters



Centroids changed after recalculation of mean
With all data points

Step 6: Recalculate New Centroids till they change



Centroids changed after recalculation of mean
With all data points

This data point was previously assigned to Cluster 3 but
after centroid update it's been assigned to Cluster 2

Mathematical aspect of K-means

Step 1: Pick the value of K(Number of clusters)

x_i
15
15
16
19
19
20
20
21
22
28
35
40
41
42
43
44
60
61
65

For the time being we take K=2 i.e. observations will be organized into 2 clusters namely K1 & K2.

Step 2: Initialize Centroids or Means

x_i
15
15
16
19
19
20
20
21
22
28
35
40
41
42
43
44
60
61
65

The centroids(C1 & C2) are initialized arbitrarily.

C1= 16,

C2 = 22.

Step 3: Calculate Euclidean distance between centroids & data points

x_i	C1	C2	Distance 1	Distance 2
15	16	22	1	7
15	16	22	1	7
16	16	22	0	6
19	16	22	3	3
19	16	22	3	3
20	16	22	4	2
20	16	22	4	2
21	16	22	5	1
22	16	22	6	0
28	16	22	12	6
35	16	22	19	13
40	16	22	24	18
41	16	22	25	19
42	16	22	26	20
43	16	22	27	21
44	16	22	28	22
60	16	22	44	38
61	16	22	45	39
65	16	22	49	43

$$Euclidean\ distance(\underbrace{15}_{\text{First observation}},\underbrace{16}_{\text{Centroid of K1}}) = \sqrt{(15-16)^2} = 1$$

$$Euclidean\ distance(\underbrace{15}_{\text{First observation}},\underbrace{22}_{\text{Centroid of K2}}) = \sqrt{(15-22)^2} = 7$$

Step 4: Assigning data points to clusters

x_i	C1	C2	Distance 1	Distance 2	Nearest Cluster
15	16	22	1	7	1
15	16	22	1	7	1
16	16	22	0	6	1
19	16	22	3	3	2
19	16	22	3	3	2
20	16	22	4	2	2
20	16	22	4	2	2
21	16	22	5	1	2
22	16	22	6	0	2
28	16	22	12	6	2
35	16	22	19	13	2
40	16	22	24	18	2
41	16	22	25	19	2
42	16	22	26	20	2
43	16	22	27	21	2
44	16	22	28	22	2
60	16	22	44	38	2
61	16	22	45	39	2
65	16	22	49	43	2

$$Euclidean\ distance(\underset{\text{First observation}}{15},\underset{\text{Centroids of K1}}{16}) = \sqrt{(15-16)^2}$$

$$= 1$$

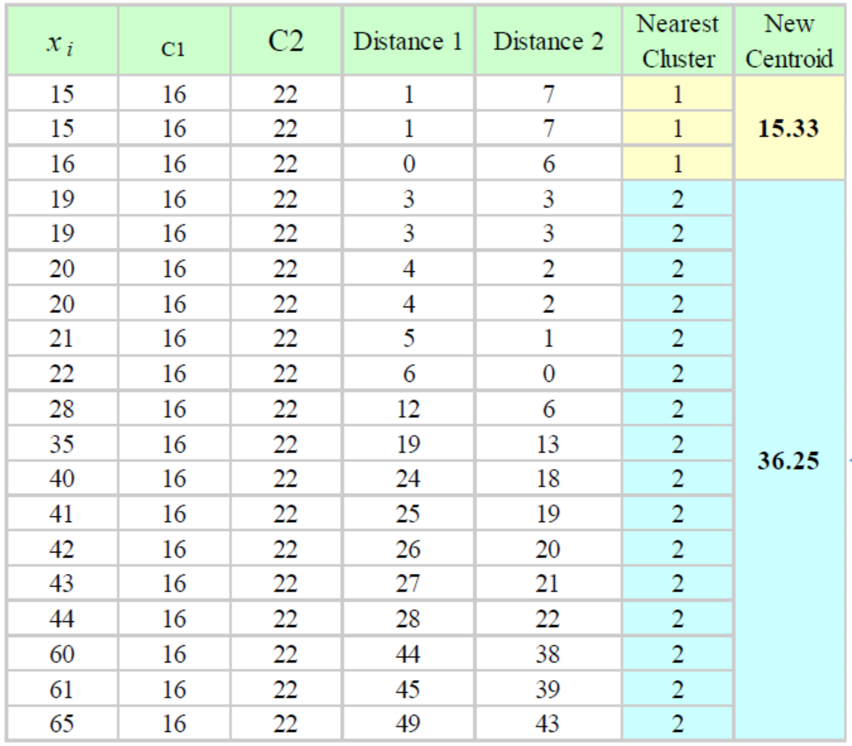
$$Euclidean\ distance(\underset{\text{First observation}}{15},\underset{\text{Centroids of K2}}{22}) = \sqrt{(15-22)^2}$$

$$= 7$$

Based on **nearest Euclidean distance** measure we assign **first** observation to Cluster 1 or K1 because 1 < 7.

Thus we've assigned for other data points

Step 5: Recalculating mean or updating centroids



x_i	C1	C2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	16	22	1	7	1	
15	16	22	1	7	1	**15.33**
16	16	22	0	6	1	
19	16	22	3	3	2	
19	16	22	3	3	2	
20	16	22	4	2	2	
20	16	22	4	2	2	
21	16	22	5	1	2	
22	16	22	6	0	2	
28	16	22	12	6	2	
35	16	22	19	13	2	
40	16	22	24	18	2	**36.25**
41	16	22	25	19	2	
42	16	22	26	20	2	
43	16	22	27	21	2	
44	16	22	28	22	2	
60	16	22	44	38	2	
61	16	22	45	39	2	
65	16	22	49	43	2	

$$\text{New Centroid} = \frac{15+15+16}{3} = 15.33$$

$$\text{New Centroid} = \frac{19+19+20+20+21.....+65}{16} = 36.25$$

Step 5: Recalculating mean or updating centroids

x_i	C1	C2	Distance 1	Distance 2	Nearest Cluster
15	15.3	36.2	0.33	21.25	1
15	15.3	36.2	0.33	21.25	1
16	15.3	36.2	0.67	20.25	1
19	15.3	36.2	3.67	17.25	2
19	15.3	36.2	3.67	17.25	2
20	15.3	36.2	4.67	16.25	2
20	15.3	36.2	4.67	16.25	2
21	15.3	36.2	5.67	15.25	2
22	15.3	36.2	6.67	14.25	2
28	15.3	36.2	12.67	8.25	2
35	15.3	36.2	19.67	1.25	2
40	15.3	36.2	24.67	3.75	2
41	15.3	36.2	25.67	4.75	2
42	15.3	36.2	26.67	5.75	2
43	15.3	36.2	27.67	6.75	2
44	15.3	36.2	28.67	7.75	2
60	15.3	36.2	44.67	23.75	2
61	15.3	36.2	45.67	24.75	2
65	15.3	36.2	49.67	28.75	2

Data points were previously assigned to cluster 2.
But after updating centroids they will be assigned to cluster 1.

Step 6: Recalculate New Centroids till they change

X_i	C_1	C_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	15.33	36.25	0.33	21.25	1	18.56
15	15.33	36.25	0.33	21.25	1	
16	15.33	36.25	0.67	20.25	1	
19	15.33	36.25	3.67	17.25	1	
19	15.33	36.25	3.67	17.25	1	
20	15.33	36.25	4.67	16.25	1	
20	15.33	36.25	4.67	16.25	1	
21	15.33	36.25	5.67	15.25	1	
22	15.33	36.25	6.67	14.25	1	
28	15.33	36.25	12.67	8.25	2	45.9
35	15.33	36.25	19.67	1.25	2	
40	15.33	36.25	24.67	3.75	2	
41	15.33	36.25	25.67	4.75	2	
42	15.33	36.25	26.67	5.75	2	
43	15.33	36.25	27.67	6.75	2	
44	15.33	36.25	28.67	7.75	2	
60	15.33	36.25	44.67	23.75	2	
61	15.33	36.25	45.67	24.75	2	
65	15.33	36.25	49.67	28.75	2	

This step is repetition of step 3, step 4 & step 5.

After updating centroids some data points have been assigned to cluster 1.

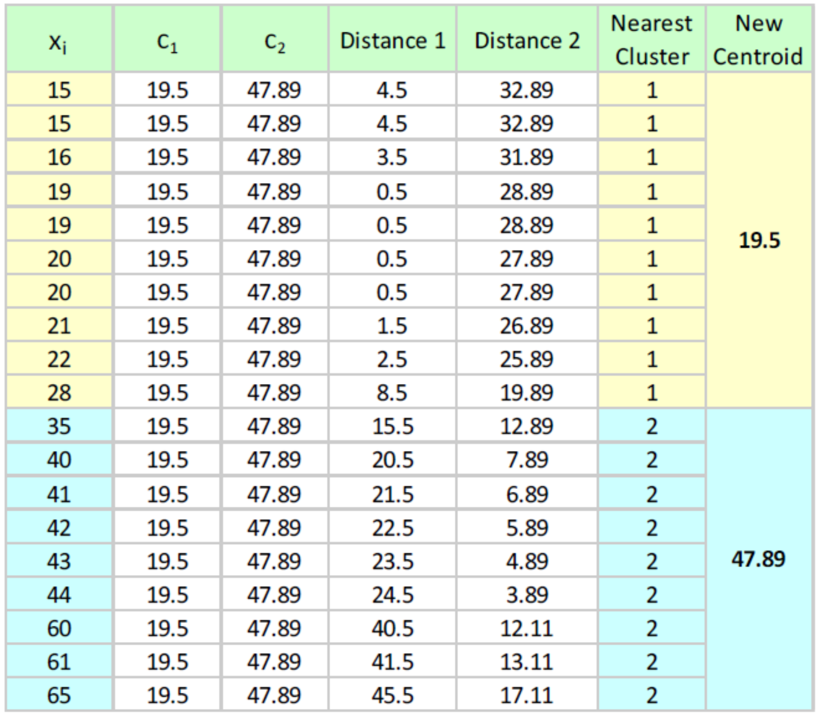
Step 6: Recalculate New Centroids till they change

x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	18.56	45.9	3.56	30.9	1	19.5
15	18.56	45.9	3.56	30.9	1	
16	18.56	45.9	2.56	29.9	1	
19	18.56	45.9	0.44	26.9	1	
19	18.56	45.9	0.44	26.9	1	
20	18.56	45.9	1.44	25.9	1	
20	18.56	45.9	1.44	25.9	1	
21	18.56	45.9	2.44	24.9	1	
22	18.56	45.9	3.44	23.9	1	
28	18.56	45.9	9.44	17.9	1	
35	18.56	45.9	16.44	10.9	2	47.89
40	18.56	45.9	21.44	5.9	2	
41	18.56	45.9	22.44	4.9	2	
42	18.56	45.9	23.44	3.9	2	
43	18.56	45.9	24.44	2.9	2	
44	18.56	45.9	25.44	1.9	2	
60	18.56	45.9	41.44	14.1	2	
61	18.56	45.9	42.44	15.1	2	
65	18.56	45.9	46.44	19.1	2	

This step is repetition of step 3, step 4 & step 5.

After updating centroid, one data point has been assigned to cluster 1.

Step 6: Recalculate New Centroids till they change



x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	19.5	47.89	4.5	32.89	1	
15	19.5	47.89	4.5	32.89	1	
16	19.5	47.89	3.5	31.89	1	
19	19.5	47.89	0.5	28.89	1	
19	19.5	47.89	0.5	28.89	1	**19.5**
20	19.5	47.89	0.5	27.89	1	
20	19.5	47.89	0.5	27.89	1	
21	19.5	47.89	1.5	26.89	1	
22	19.5	47.89	2.5	25.89	1	
28	19.5	47.89	8.5	19.89	1	
35	19.5	47.89	15.5	12.89	2	
40	19.5	47.89	20.5	7.89	2	
41	19.5	47.89	21.5	6.89	2	
42	19.5	47.89	22.5	5.89	2	
43	19.5	47.89	23.5	4.89	2	**47.89**
44	19.5	47.89	24.5	3.89	2	
60	19.5	47.89	40.5	12.11	2	
61	19.5	47.89	41.5	13.11	2	
65	19.5	47.89	45.5	17.11	2	

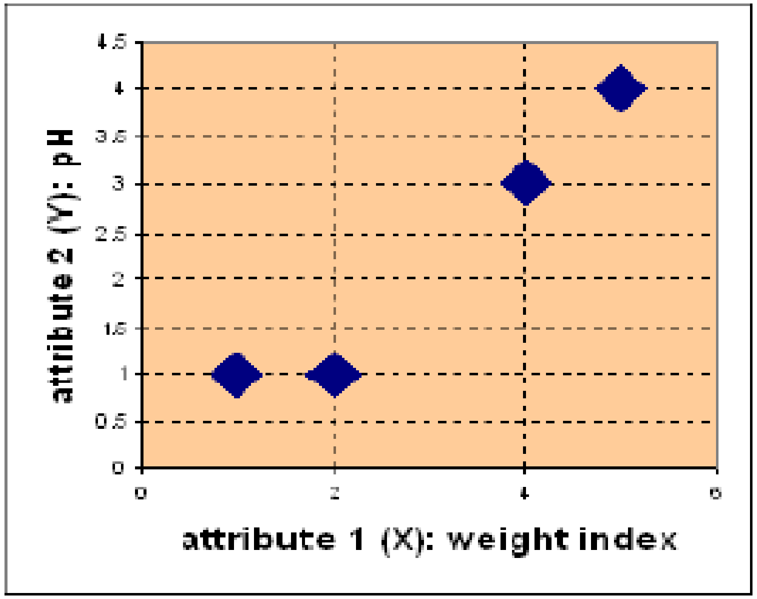
This step is repetition of step 3, step 4 & step 5.

We should stop proceeding here because **centroids don't change further** after repetition of step 3, step 4 & step 5.

Demonstration of *K-Means* Clustering algorithm

- Suppose we have objects (4 types of medicines) and each object have two attributes or features as shown in table below.
- Our goal is to group these objects into K=2 group of medicine based on the two features (pH and weight index)

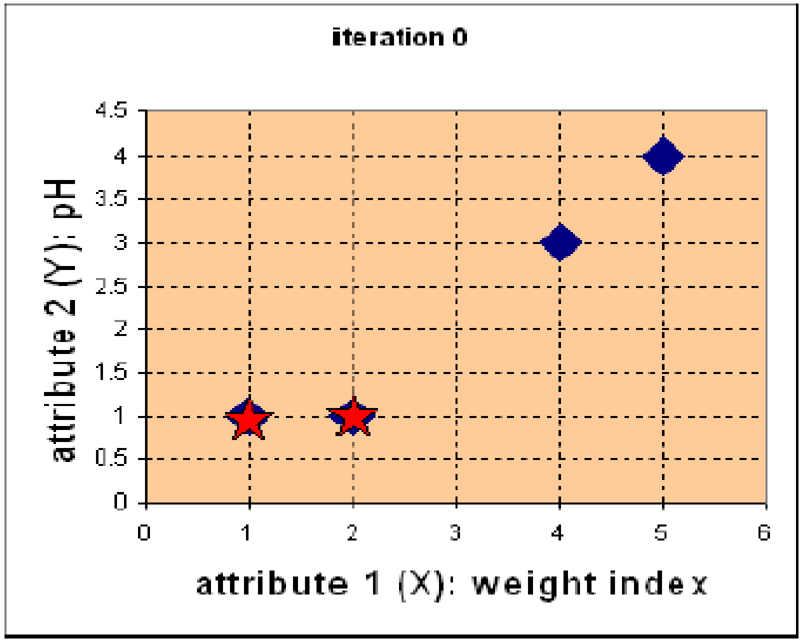
Object	attribute 1 (X): weight index	attribute 2 (Y): pH
Medicine A	1	1
Medicine B	2	1
Medicine C	4	3
Medicine D	5	4



- Each medicine represents one point with two attributes (X, Y) that we can represent it as coordinate in an attribute space as shown in above figure.

Demonstration of *K-Means* Clustering algorithm

1. *Initial value of centroids* : Suppose we use medicine A and medicine B as the first centroids. Let and denote the coordinate of the centroids, then C_1= (1,1) and C_2=(2,1)



Demonstration of *K-Means* Clustering algorithm

2. *Objects-Centroids distance* : we calculate the distance between cluster centroid to each object. Let us use Euclidean distance, then we have distance matrix at iteration 0 is

$$\mathbf{D}^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \quad \begin{array}{ll} \mathbf{c}_1 = (1,1) & group-1 \\ \mathbf{c}_2 = (2,1) & group-2 \end{array}$$

$$\begin{array}{cccc} A & B & C & D \end{array}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \quad \begin{array}{l} X \\ Y \end{array}$$

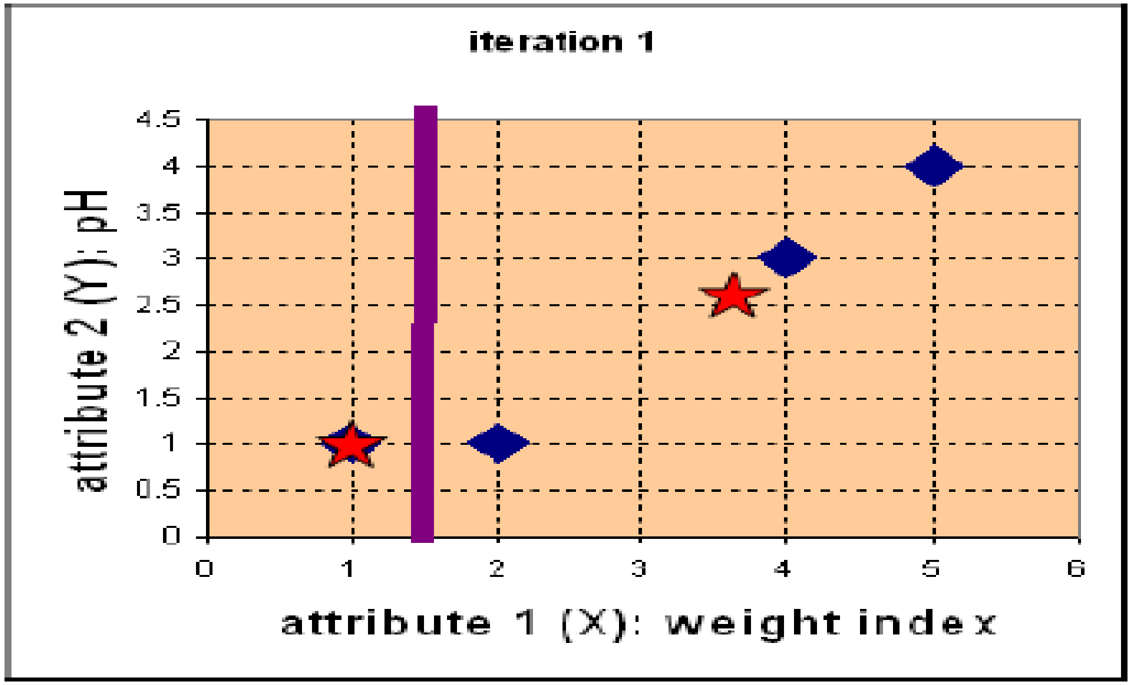
Each column in the distance matrix symbolizes the object. The first row of the distance matrix corresponds to the distance of each object to the first centroid and the second row is the distance of each object to the second centroid. For example, distance from medicine C = (4, 3) to the first centroid $\mathbf{c}_1 = (1,1)$ is $\sqrt{(4-1)^2+(3-1)^2} = 3.61$, and its distance to the second centroid $\mathbf{c}_2 = (2,1)$ is $\sqrt{(4-2)^2+(3-1)^2} = 2.83$, etc.

Demonstration of *K-Means* Clustering algorithm

3. *Objects clustering* : We assign each object based on the minimum distance. Thus, medicine A is assigned to group 1, medicine B to group 2, medicine C to group 2 and medicine D to group 2. The element of Group matrix below is 1 if and only if the object is assigned to that group.

$$\mathbf{G}^0 = \begin{array}{cccc} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} & \begin{array}{l} group-1 \\ group-2 \end{array} \\ \begin{array}{cccc} A & B & C & D \end{array} & \end{array}$$

4. *Iteration-1, determine centroids* : Knowing the members of each group, now we compute the new centroid of each group based on these new memberships. Group 1 only has one member thus the centroid remains in $\mathbf{c}_1 = (1,1)$. Group 2 now has three members, thus the centroid is the average coordinate among the three members: $\mathbf{c}_2 = (\frac{2+4+5}{3}, \frac{1+3+4}{3}) = (\frac{11}{3}, \frac{8}{3})$.



Demonstration of *K-Means* Clustering algorithm

5. *Iteration-1, Objects-Centroids distances* : The next step is to compute the distance of all objects to the new centroids. Similar to step 2, we have distance matrix at iteration 1 is

$$\mathbf{D}^1 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} \begin{matrix} \mathbf{c}_1 = (1,1) & group-1 \\ \mathbf{c}_2 = (\frac{11}{3}, \frac{8}{3}) & group-2 \end{matrix}$$

$$\begin{array}{cccc} A & B & C & D \end{array}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \begin{matrix} X \\ Y \end{matrix}$$

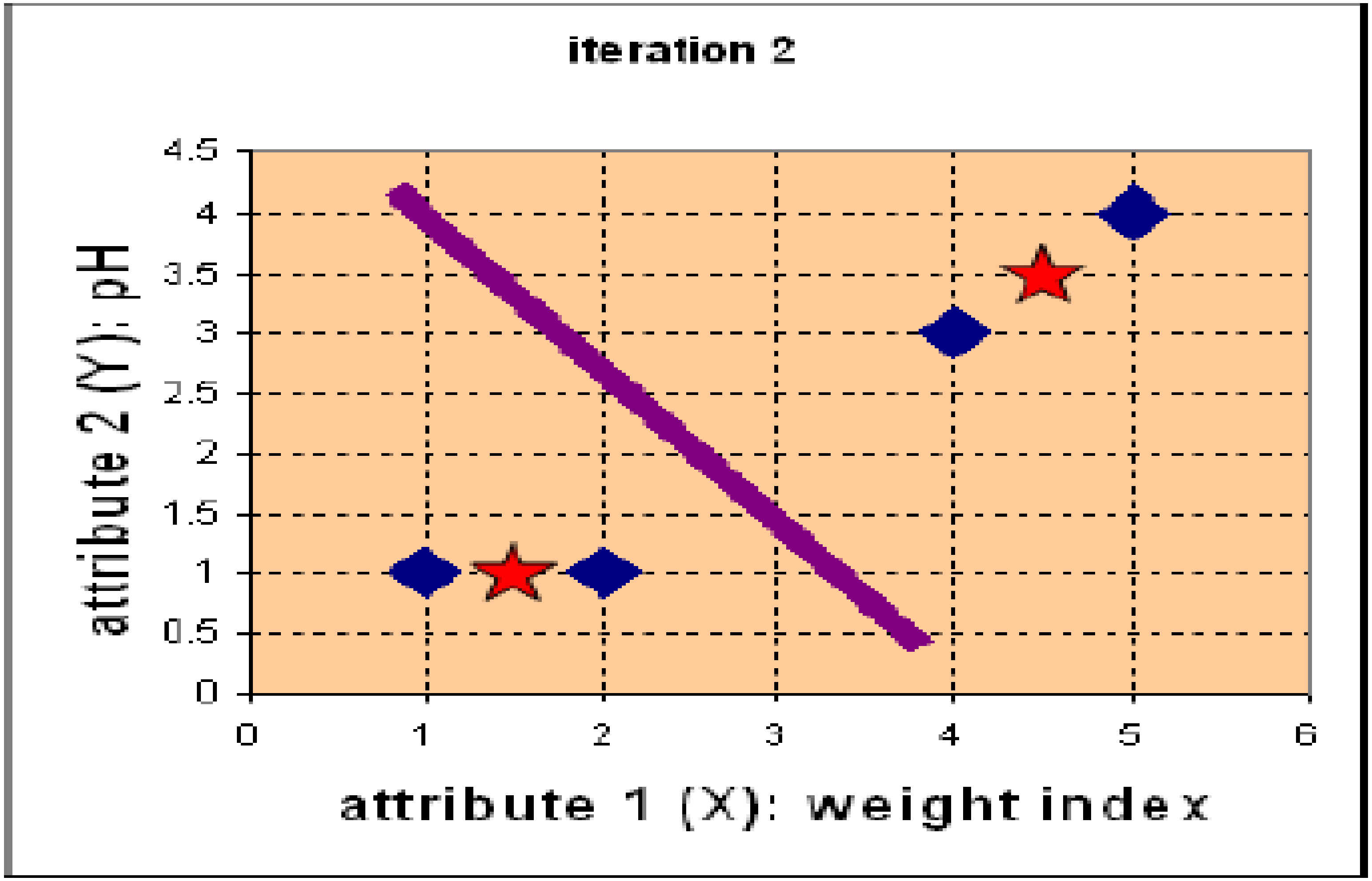
6. *Iteration-1, Objects clustering:* Similar to step 3, we assign each object based on the minimum distance. Based on the new distance matrix, we move the medicine B to Group 1 while all the other objects remain. The Group matrix is shown below

$$\mathbf{G}^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{matrix} group-1 \\ group-2 \end{matrix}$$

$$\begin{array}{cccc} A & B & C & D \end{array}$$

7. *Iteration 2, determine centroids:* Now we repeat step 4 to calculate the new centroids coordinate based on the clustering of previous iteration. Group1 and group 2 both has two members, thus the new centroids are $\mathbf{c}_1 = (\frac{1+2}{2}, \frac{1+1}{2}) = (1\frac{1}{2}, 1)$ and $\mathbf{c}_2 = (\frac{4+5}{2}, \frac{3+4}{2}) = (4\frac{1}{2}, 3\frac{1}{2})$

Demonstration of *K-Means* Clustering algorithm



New Centroids

Demonstration of *K-Means* Clustering algorithm

8. *Iteration-2, Objects-Centroids distances* : Repeat step 2 again, we have new distance matrix at iteration 2 as

$$\mathbf{D}^2 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \begin{array}{ll} \mathbf{c}_1 = (1\frac{1}{2}, 1) & group-1 \\ \mathbf{c}_2 = (4\frac{1}{2}, 3\frac{1}{2}) & group-2 \end{array}$$

$$\begin{array}{cccc|c} A & B & C & D & \\ 1 & 2 & 4 & 5 & X \\ 1 & 1 & 3 & 4 & Y \end{array}$$

9. *Iteration-2, Objects clustering:* Again, we assign each object based on the minimum distance.

Demonstration of *K-Means* Clustering algorithm

$$G^2 = \begin{array}{cccc|l} & & & & \\ \end{array}$$

$$\begin{array}{c} G^2 = \\ \\ \end{array} \begin{array}{l} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{array}{l} group-1 \\ group-2 \end{array} \\ \;\; A \quad B \quad C \quad D \end{array}$$

We obtain result that $G^2 = G^1$. Comparing the grouping of last iteration and this iteration reveals that the objects does not move group anymore. Thus, the computation of the k-mean clustering has reached its stability and no more iteration is needed. We get the final grouping as the results

Object	attribute 1 (X): weight index	attribute 2 (Y): pH	Group (result)
Medicine A	1	1	**1**
Medicine B	2	1	**1**
Medicine C	4	3	**2**
Medicine D	5	4	**2**

Comments on the *K-Means* Method

- Strength: The method is r*elatively scalable and efficient in processing large data sets because the computational complexity of the algorithm is $O(tkn)$*, where n is no. objects, k is no. clusters, and t is no. iterations. Normally, $k, t << n$.
- Weakness
  - Applicable only when *mean* is defined.
  - Need to specify *k,* the *number* of clusters, in advance can be seen as a disadvantage.
  - Unable to handle noisy data and *outliers*
  - Not suitable to discover clusters of very different size.

Hierarchical Clustering

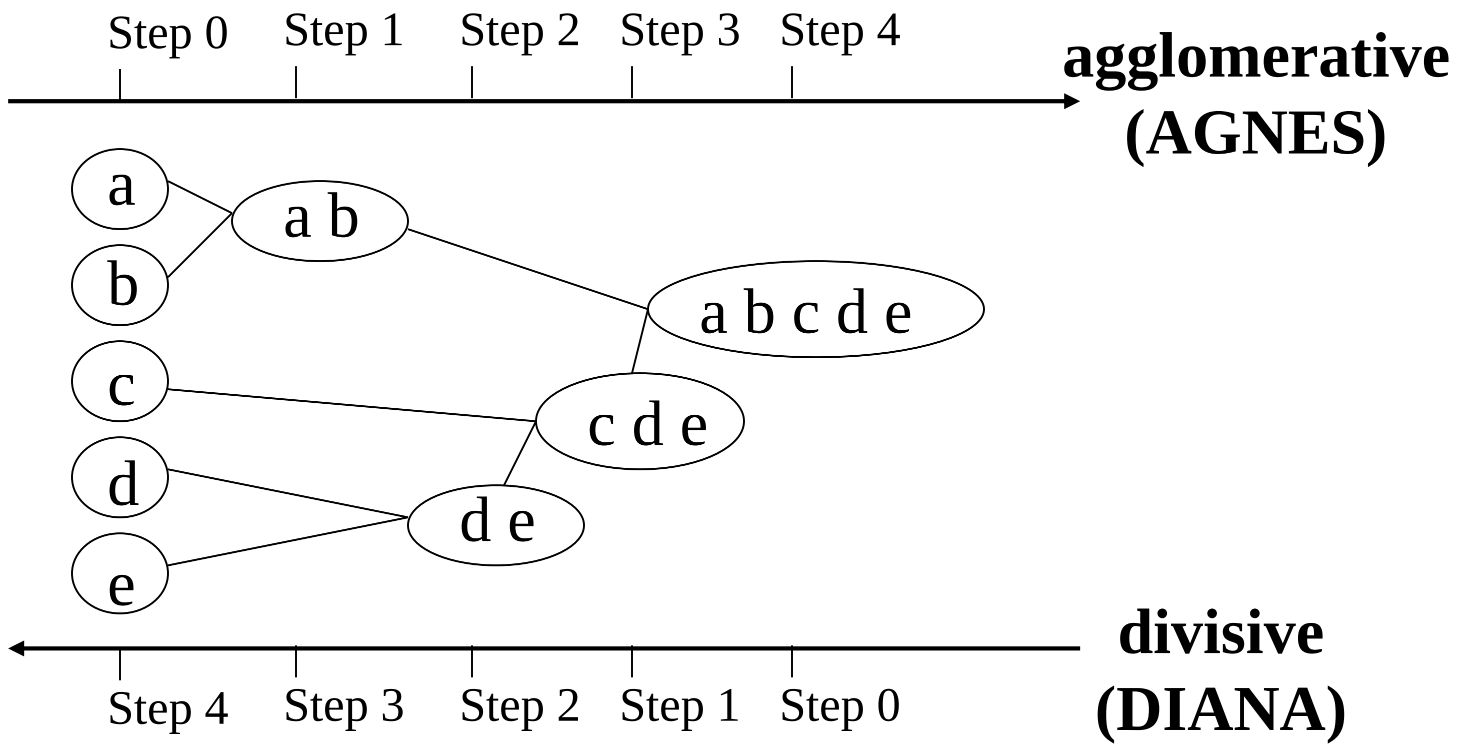
- A hierarchical clustering method works by grouping data objects into a tree of clusters.
- Hierarchical clustering methods can be further classified as either *agglomerative* or *divisive*, depending on whether the hierarchical decomposition is formed in a *bottom-up (merging)* or *top-down (splitting)* fashion.
- The method can not backtrack.

Some Facts about Hierarchical Clustering

- Hierarchical methods suffer from the fact that once we have performed either merge or split step, it can never be undone.
- This inflexibility is useful in that it leads to smaller computation costs by not having to worry about a combinatorial number of different choices.
- However, such techniques cannot correct mistaken decisions that once have taken.
- There are two approaches that can help in improving the quality of hierarchical clustering:
  - Firstly to perform careful analysis of object linkages at each hierarchical partitioning or
  - By integrating hierarchical agglomeration and other approaches by first using a hierarchical agglomerative algorithm to group objects into micro-clusters, and then performing macro-clustering on the micro-clusters using another clustering method.

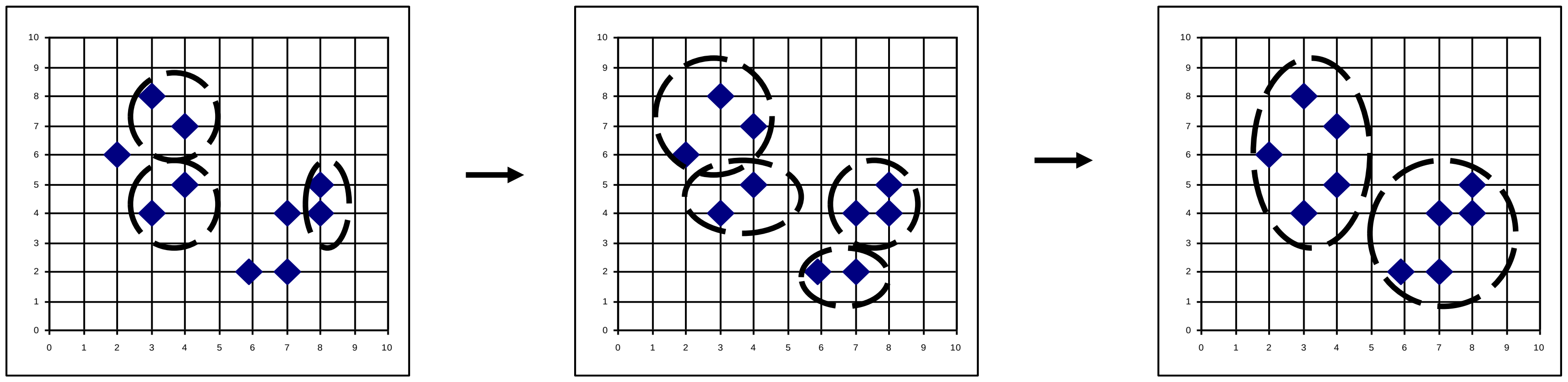
Hierarchical Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition

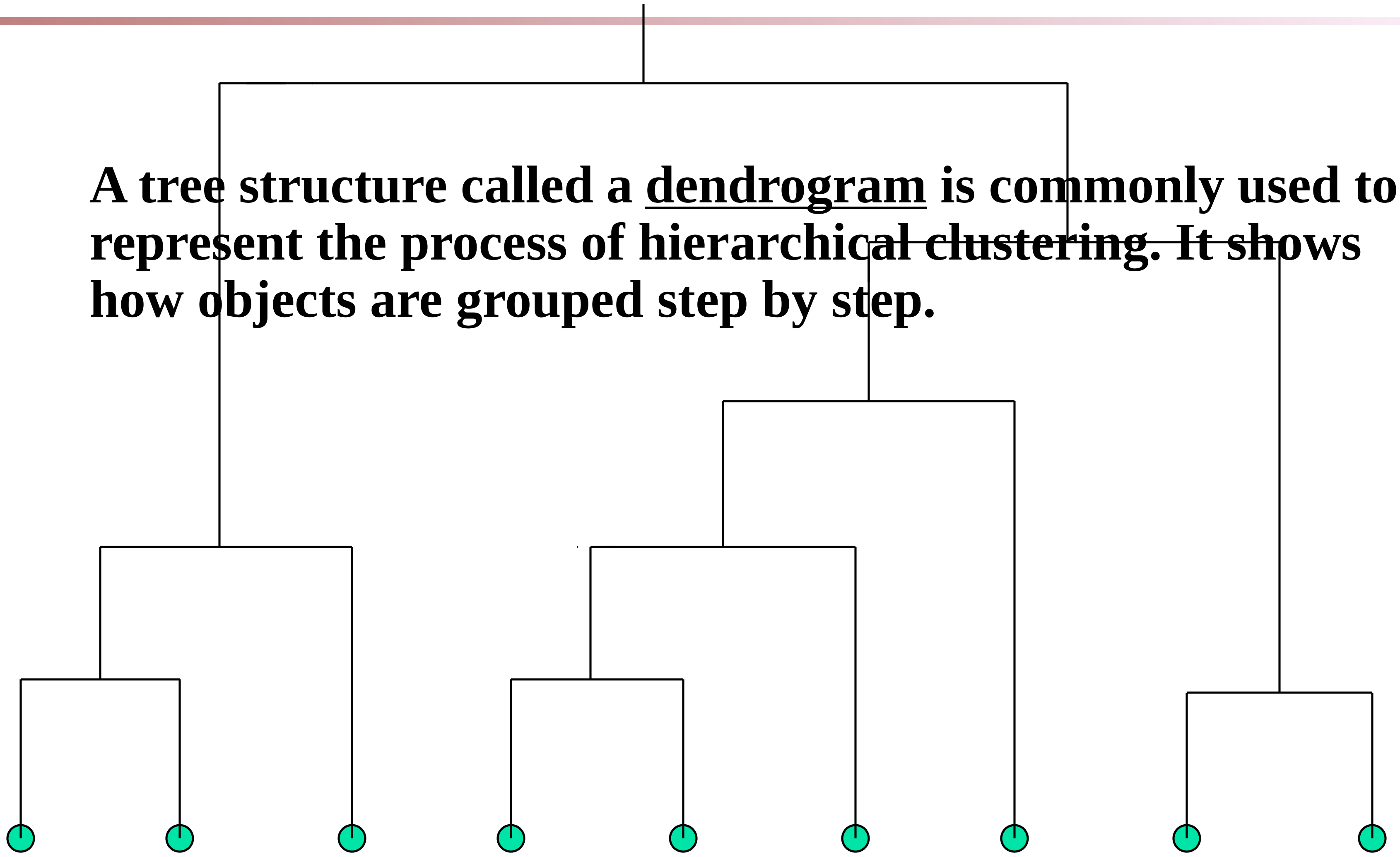


AGNES (Agglomerative Nesting)

- This *bottom-up* strategy starts by placing object in its own cluster and then merges these atomic clusters into larger and larger clusters.
- Use of dissimilarity matrix.
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster or until it satisfies certain termination conditions.



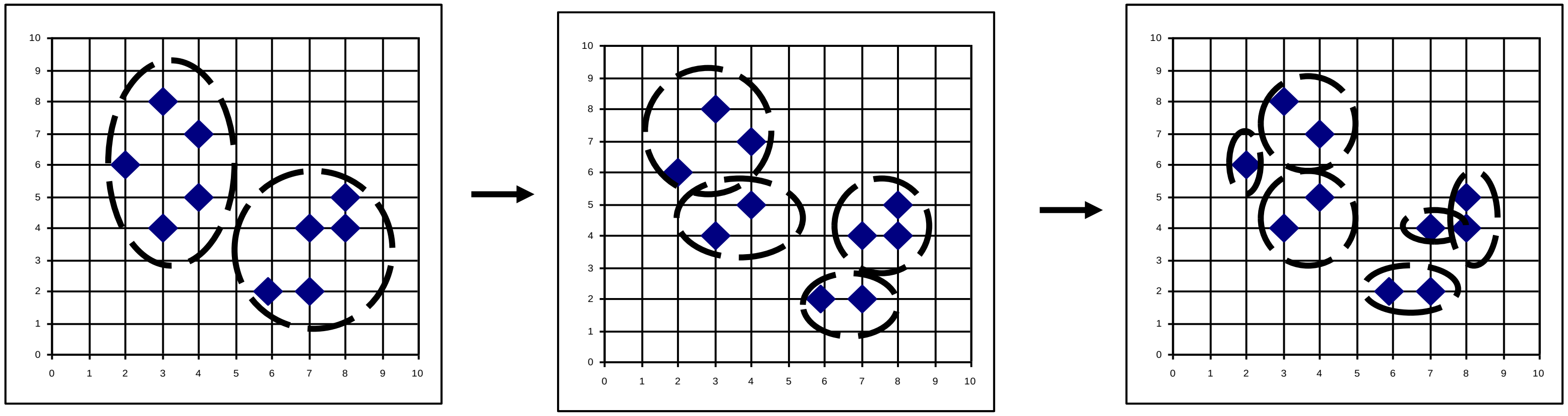
Dendrogram: Shows How the Clusters are Merged



A tree structure called a dendrogram is commonly used to represent the process of hierarchical clustering. It shows how objects are grouped step by step.

DIANA (Divisive Analysis)

- Inverse order of AGNES
- 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.
- Eventually each node forms a cluster on its own



Recent Hierarchical Clustering Methods

- BIRCH (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters
- ROCK (1999): clustering categorical data by neighbour and link analysis
- CHAMELEON (1999): hierarchical clustering using dynamic modeling

Clustering Categorical Data: The ROCK Algorithm

- ROCK: (Robust Clustering using links) is a hierarchical clustering algorithm that explores the concept of *links* (the no. of common neighbors between two objects) for the data with categorical attributes.
- Traditional clustering algorithms for clustering data with Boolean and categorical attributes use distance functions.
- Such distance measures can not lead to high quality clusters when clustering categorical data.
- ROCK takes a more global approach to clustering by considering the *neighborhoods* of individual pairs of points.
- Major ideas
  - Use links to measure similarity/proximity
  - Not distance-based

Similarity Measure in ROCK

- Traditional measures for categorical data may not work well, e.g., Jaccard coefficient, it uses point similarity by distance-based in place of neighborhood *link* information.
- Example: Two groups (clusters) of transactions
  - Suppose that a market basket database contains transactions regarding the items *a, b,… ,g. Consider* two clusters of transactions,*C1 andC2.C1, which references the items <a, b, c, d, e> and C2 references the items <a, b, f , g>*
  - C_1. <a, b, c, d, e>: {a, b, c}, {a, b, d}, {a, b, e}, {a, c, d}, {a, c, e}, {a, d, e}, {b, c, d}, {b, c, e}, {b, d, e}, {c, d, e}
  - C_2. <a, b, f, g>: {a, b, f}, {a, b, g}, {a, f, g}, {b, f, g}
- Jaccard co-efficient may lead to wrong clustering result
  - C_1: 0.2 ({a, b, c}, {b, d, e}) to 0.5 ({a, b, c}, {a, b, d})
  - C_1 & C_2: could be as high as 0.5 ({a, b, c}, {a, b, f})
  - Clearly, by using the Jaccard coefficient on its own, we cannot obtain the desired clusters.
- Jaccard co-efficient-based similarity function:
  - Ex. Let T_1 = {a, b, c}, T_2 = {b, d, e}

$$Sim(T_1, T_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|}$$

$$Sim(T_1, T_2) = \frac{|\{b\}|}{|\{a, b, c, d, e\}|} = \frac{1}{5} = 0.2$$

Link Measure in ROCK

- The link-based approach of ROCK can successfully separate the transactions into appropriate clusters.
- Links: no. of common neighbors
  - C_1 <a, b, c, d, e>: {a, b, c}, {a, b, d}, {a, b, e}, {a, c, d}, {a, c, e}, {a, d, e}, {b, c, d}, {b, c, e}, {b, d, e}, {c, d, e}
  - C_2 <a, b, f, g>: {a, b, f}, {a, b, g}, {a, f, g}, {b, f, g}
  - Let T {a, b, f} of C2 has five links with transaction {a, b, g} of the same cluster (due to common neighbors {a, b, c}, {a, b, d}, {a, b, e}, {a, f, g}, {b, f, g}.
  - However, transaction {a, b, f} of C2 has only three links with {a, b, c} of C1 (due to {a, b, d}, {a, b, e}, and {a, b, g}).
  - Similarly, transaction {a, f, g} of C2 has two links with every other transaction in C2, and zero links with each transaction in C1.
  - Thus link is a better measure than Jaccard coefficient to get the desired cluster

BIRCH (1996)

- Birch: Balanced Iterative Reducing and Clustering using Hierarchies.
- BIRCH is designed for clustering a large amount of numerical data by integration of hierarchical clustering.
- It overcomes the two difficulties of agglomerative clustering methods: (1) *scalability* and (2) *the inability to undo what was done in the previous step.*
- BIRCH introduces two concepts, *clustering features* and *clustering feature tree (CF tree),* which are used to summarize cluster representations.
- These structures help the clustering method achieve good speed and scalability in large databases and also make it effective for incremental and dynamic clustering of incoming objects.
- *Weakness:* handles only numeric data.

BIRCH (1996)

Let's look closer at the above-mentioned structures. Given *n d-dimensional data* objects or points in a cluster, we can define the centroid ***x0****, radius* ***R****, and diameter* ***D*** of the cluster as follows:

$$x_0 = \frac{\sum_{i=1}^{n} x_i}{n}$$

$$R = \sqrt{\frac{\sum_{i=1}^{n} (x_i - x_0)^2}{n}}$$

$$D = \sqrt{\frac{\sum_{i=1}^{n} \sum_{j=1}^{n} (x_i - x_j)^2}{n(n-1)}}$$

- R is the average distance from member objects to the centroid,
- D is the average pairwise distance within a cluster.
- Both R and D reflect the tightness of the cluster around the centroid. A clustering feature (CF) is a three-dimensional vector summarizing information about clusters of objects.

Clustering Feature Vector in BIRCH

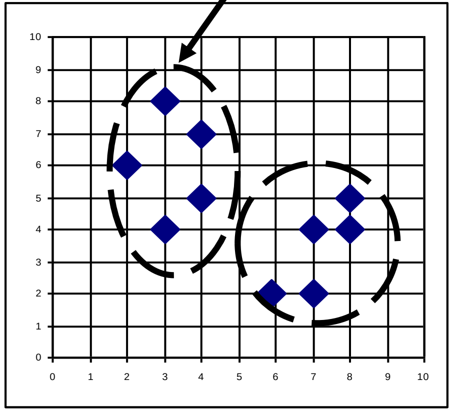
•Given n *d-dimensional* objects or points in cluster, {xi}, then the CF of the cluster is defined as ***CF* = *(N, LS, SS)*,** *N*: Number of data points in the cluster.

LS is the linear sum of the n points. ***LS:*** $\sum_{i=1}^{N} X_i$

SS is the square sum of the data points. ***SS:*** $\sum_{i=1}^{N} X_i^2$

• Clustering features are *additive*. For example suppose that we have two disjoint clusters, C1 and C2, having the cluster features, CF1 and CF2, respectively. The clustering feature for the cluster that is formed by merging C1 and C2 is simply CF1+CF2.

CF = (5, (16,30),(54,190))



(3,4)

(2,6)

(4,5)

(4,7)

(3,8)

CF-Tree in BIRCH

- A CF tree is a height-balanced tree that stores the clustering features for a hierarchical clustering
  - A nonleaf node in a tree has descendants or "children"
  - The nonleaf nodes store sums of the CFs of their children and thus summarize clustering information about their children.
- A CF tree has two parameters
  - Branching factor: specify the maximum number of children per nonleaf node.
  - threshold: max diameter of sub-clusters stored at the leaf nodes of the tree.
- The primary phases are:
  - Phase 1: scan DB to build an initial in-memory CF tree (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data)
  - Phase 2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree

The CF Tree Structure

- for phase 1, the CF tree is built dynamically as objects are inserted. Thus the method is incremental.
- An object is inserted into the closest leaf entry (subcluster).
- If the diameter of the subcluster stored in the leaf node after insertion is larger than the threshold value, then the leaf node and possibly other nodes are split.
- After the insertion of the new object, information about it is passed toward the root of the tree.

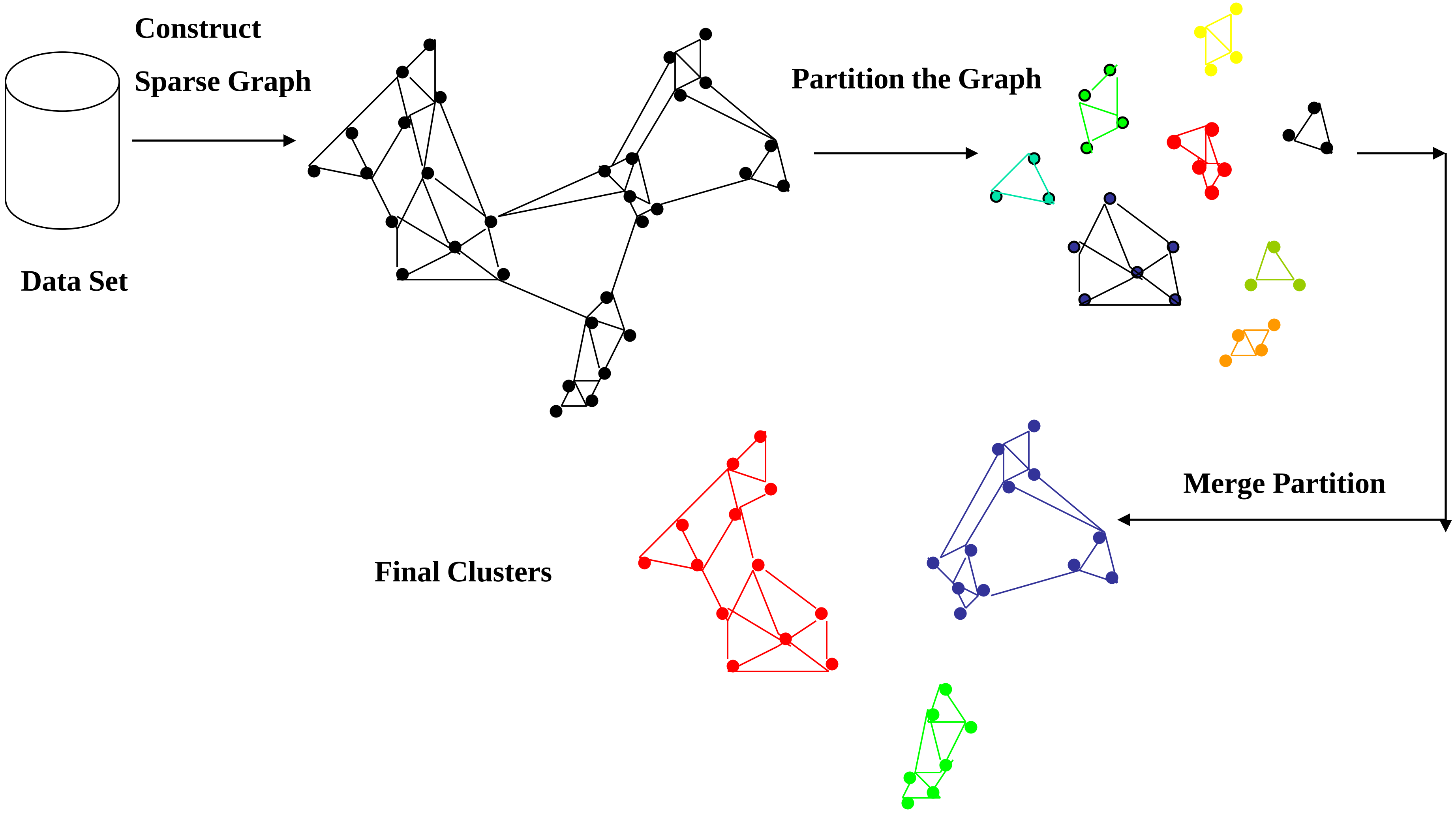
The CF Tree Structure

- The size of the CF tree can be changed by modifying the threshold.
- If the size of the memory that is needed for storing the CF tree is larger than the size of the main memory, then a smaller threshold value can be specified and the CF tree is rebuilt.
- Once the CF tree is built, any clustering algorithm, such as a typical partitioning algorithm, can be used with the CF tree in phase 2.
- NOTE- BIRCH does not perform well, because it uses the notion of radius or diameter to control the boundary of a cluster.

CHAMELEON: Hierarchical Clustering Using Dynamic Modeling (1999)

- Measures the similarity based on a dynamic model
  - Two clusters are merged only if the *interconnectivity* and *closeness (proximity)* between two clusters are high *relative to* the internal interconnectivity of the clusters and closeness of items within the clusters
  - **Rock** ignores information about the **closeness** of two clusters
- A two-phase algorithm
  1. Use a graph partitioning algorithm: cluster objects into a large number of relatively small sub-clusters
  2. Use an agglomerative hierarchical clustering algorithm: find the genuine clusters by repeatedly combining these sub-clusters

Overall Framework of CHAMELEON



Density-Based Clustering Methods

- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98) (more grid-based)