# Unit 5
# Cluster Analysis

# Cluster Analysis

- Clustering is a Process of grouping objects which are similar
- Objects of a cluster are similar and objects of different cluster are dissimilar
- The objects can be grouped based on attributes/features or by relationships with other objects (distance or similarity)
- Clustering does not require assumptions about category labels that tag objects with prior identifiers
- Clustering is an unsupervised learning and classification is a supervised learning
- Clustering is subjective (or problem dependent)

- Clustering can summarize data to a manageable level
- Applications
  - Customer relationship management
  - Information retrieval
  - Data compression
  - Image processing
  - Marketing
  - Medicine
  - Pattern recognition
  - Psychology and statistics

# Similarity and its measurement

- Grouping is done based on the closeness or similarity
- Sometimes we are given with perfect features to measure similarity
- But most of times, we need to
  - Generate features
  - Clean features
  - Normalize features
  - Reduce features
- Two useful tricks to measure similarity
  - Feature Projection - ( how similar) s(x,y)
  - Edit distance- ( how dissimilar ) d(x,y)

- Feature Projection
  - We project the data into feature space, the distance in feature space becomes the similarity
  - Suppose given with the different birds we group taking the feature beak length
- Edit distance
  - Grouping done based on the dissimilarities or distance between the objects when forming the clusters
  - The distance can be based on single dimension or multiple dimensions
  - Suppose we want to group the fast food categories, we can start clustering be their calories, taste, price etc.

- Distance measure
  - Euclidean Distance – calculated from raw data
    - Distance($O_i$,$O_j$)= sqrt ($\sum(O_{ik} - O_{jk})^2$

| Object | X1 | X2 | X3 | X4 |
|---|---|---|---|---|
| O1 | 5 | 6 | 4 | 9 |
| O2 | 8 | 9 | 3 | 2 |
| O3 | 3 | 4 | 5 | 3 |

Distance $(O_1,O_2)$ = sqrt($(5-8)^2+(6-9)^2+(4-3)^2+(9-2)^2$ = 8.25

Distance $(O_1,O_3)$ = sqrt($(5-3)^2+(6-4)^2+(4-5)^2+(9-3)^2$ = 6.7

Distance $(O_1,O_2)$ = sqrt($(8-3)^2+(9-4)^2+(3-5)^2+(2-3)^2$ = 7.4

- Distance measure
  - Manhattan Distance – Simply the average difference across dimensions
    - Distance($O_i$,$O_j$)= $1/n \left(\sum(|O_{ik} - O_{jk}|\right)$
    - n represents the number of features

| Object | X1 | X2 | X3 | X4 |
|---|---|---|---|---|
| O1 | 5 | 6 | 4 | 9 |
| O2 | 8 | 9 | 3 | 2 |
| O3 | 3 | 4 | 5 | 3 |

Distance $(O_1,O_2) = 1/4(|5\text{-}8|+|6\text{-}9|+|4\text{-}3|+|9\text{-}2| = 14/4 = 3.5$

Distance $(O_1,O_2) = 1/4(|5\text{-}3|+|6\text{-}4|+|4\text{-}5|+|9\text{-}3| = 2.75$

Distance $(O_1,O_2) = 1/4(|8\text{-}3|+|9\text{-}4|+|3\text{-}5|+|2\text{-}3| = 3.25$

- Distance measure
  - Chebychev Distance – Simply the average difference across dimensions
    - $\text{Distance}(O_i, O_j) = \text{Max}(|O_{ik} - O_{jk}|)$

| Object | X1 | X2 | X3 | X4 |
|---|---|---|---|---|
| O1 | 5 | 6 | 4 | 9 |
| O2 | 8 | 9 | 3 | 2 |
| O3 | 3 | 4 | 5 | 3 |

$\text{Distance}(O_1, O_2) = \text{Max}(|5-8|, |6-9|, |4-3|, |9-2| \; = 7$

$\text{Distance}(O_1, O_2) = \text{Max}(|5-3|, |6-4|, |4-5|, |9-3| \; = 6$

$\text{Distance}(O_1, O_2) = \text{Max}(|8-3|, |9-4|, |3-5|, |2-3| \; = 5$

- Distance measure
  - Percent Disagreement– suited for features which is categorical in nature
    - Distance($O_i$,$O_j$)= 100* [ Number of ($O_{ik}$ <>$O_{jk}$ )] div n
    - N represents the number of features

| Object | Gender | Age bracket | Income level | BP |
|---|---|---|---|---|
| O1 | M | 20-30 | Low | Normal |
| O2 | M | 30-40 | Low | Normal |
| O3 | F | 20-30 | Medium | Normal |

Distance ($O_1$,$O_2$) = 100*1 div 4 =25%
Distance ($O_1$,$O_3$) = 100*2 div 4= 50%
Distance ($O_2$,$O_3$) = 100*3 div 4 = 75%

# Types of Clustering

- Partitional – we construct various partitions and then evaluate them by some criteria
  - K-means
  - K- medoids
- Hierarchical – we create hierarchical decomposition of the set of objects using some criterion
  - Bottom up – agglomerative
    - Initially, each point is a cluster
    - Repeatedly combine the two nearest clusters into one
  - Top-down – divisive
    - Start with one cluster and recursively split it

# K-means

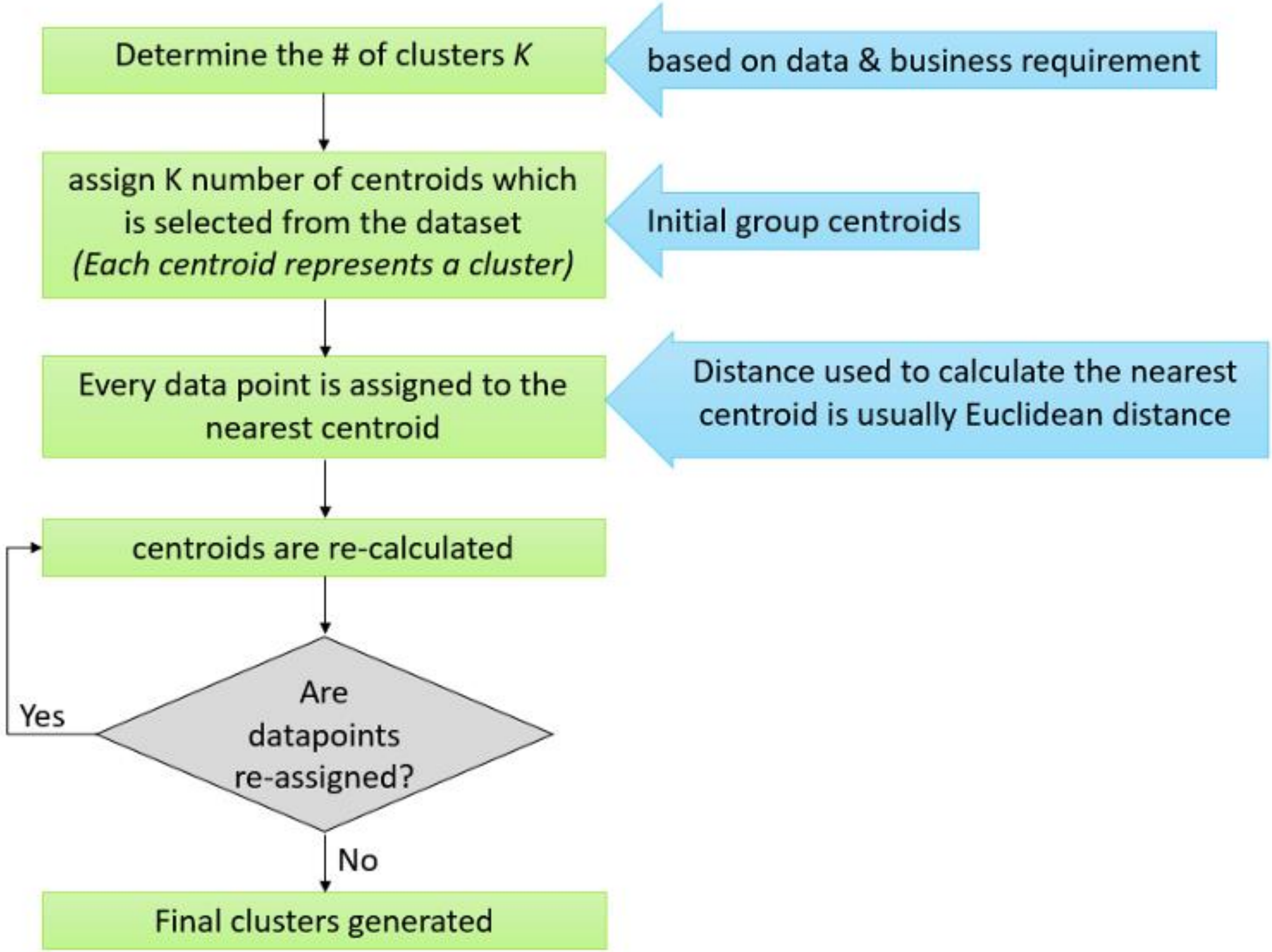
**Step 1**: Choose k objects arbitrarily from D as in initial cluster centers

**Step 2** : Repeat

**Step 3**: Reassign each object to the most similar cluster based on the mean value of the objects in the cluster

**Step 4**: Update the cluster means

**Step 5**: Until no change


Determine the # of clusters K
based on data & business requirement
assign K number of centroids which is selected from the dataset
(Each centroid represents a cluster)
Initial group centroids
Every data point is assigned to the nearest centroid
Distance used to calculate the nearest centroid is usually Euclidean distance
centroids are re-calculated
Are datapoints re-assigned?
Yes
No
Final clusters generated

# Example

- As a simple illustration of a k-means algorithm, consider the following data set consisting of the scores of two variables on each of Five individuals. This data set is to be grouped into two clusters.

| Subject | X1 | X2 |
|---|---|---|
| A | 1 | 1 |
| B | 1 | 0 |
| C | 0 | 2 |
| D | 2 | 4 |
| E | 3 | 5 |

- Choose the cluster centroids

| | Individual | Mean Vector (centroid) |
|---|---|---|
| **Group 1** | A | (1, 1) |
| **Group 2** | D | (2, 4) |

- Calculate the distance (using euclidean/ manhattan/ chebychev) of each individual to the chosen centroid. Assign 1 to the min distance of the cluster. Eg: for obj A min(0,3) = 0 , so put 1 in cluster A. Rearrange the cluster and reassign the centroid.

| Object | Cluster 1 (1,1) | Cluster 2 (2,4) |
|---|---|---|
| A | 0 | 3 |
| B | 1 | 3 |
| C | 2 | 2 |
| D | 3 | 0 |
| E | 4 | 1 |

| Object | Cluster 1 (1,1) | Cluster 2 (2,4) |
|---|---|---|
| A | 1 | 0 |
| B | 1 | 0 |
| C | 1 | 0 |
| D | 0 | 1 |
| E | 0 | 1 |

New centroid,

Cluster 1 (1+1+0/3 , 1+0+2/3) = (2/3,3/3) = (0.6,1)

Cluster 2 ( 2+3/2 , 4+5/2) = (5/2,9/2)=(2.5,4.5)

- Repeat until no change in the centroids

| Object | Cluster 1 (0.6,1) | Cluster 2 (2.5,4.5) |
|---|---|---|
| A | 0.4 | 3.5 |
| B | 1 | 4.5 |
| C | 1 | 2.5 |
| D | 3 | 0.5 |
| E | 4 | 0.5 |

| Object | Cluster 1 (0.6,1) | Cluster 2 (2.5,4.5) |
|---|---|---|
| A | 1 | 0 |
| B | 1 | 0 |
| C | 1 | 0 |
| D | 0 | 1 |
| E | 0 | 1 |

New centroid,

Cluster 1 (1+1+0/3 , 1+0+2/3) = (2/3,3/3) = (0.6,1)

Cluster 2 ( 2+3/2 , 4+5/2) = (5/2,9/2)=(2.5,4.5)

# Practice Problem

| 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 |

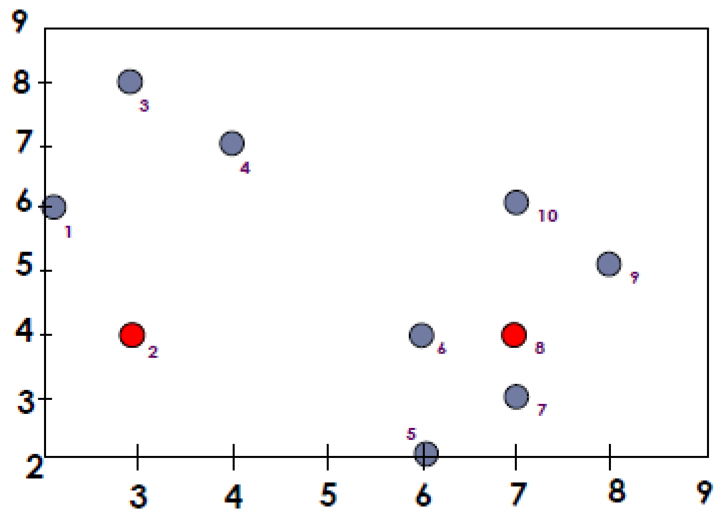
# K-medoids

- K-means is sensitive to outliers
- k-medoids – instead of taking the mean value of the object in the cluster as a reference point, medoids can be used which is more centrally located object in a cluster
- The k-medoids clustering algorithm:
  - Select k points as the initial representative of the objects
  - Repeat
    - Assigning each point closest to the medoid
    - Randomly select a non-representative object Oi
    - Compute the total cost S of swapping the medoid m with Oi
    - If S<0, then swap m with Oi to form the new set of medoids
  - Until convergence criterion is satisfied

# K-medoids example

**Data Objects**

| | $A_1$ | $A_2$ |
|---|---|---|
| $O_1$ | 2 | 6 |
| $O_2$ | 3 | 4 |
| $O_3$ | 3 | 8 |
| $O_4$ | 4 | 7 |
| $O_5$ | 6 | 2 |
| $O_6$ | 6 | 4 |
| $O_7$ | 7 | 3 |
| $O_8$ | 7 | 4 |
| $O_9$ | 8 | 5 |
| $O_{10}$ | 7 | 6 |



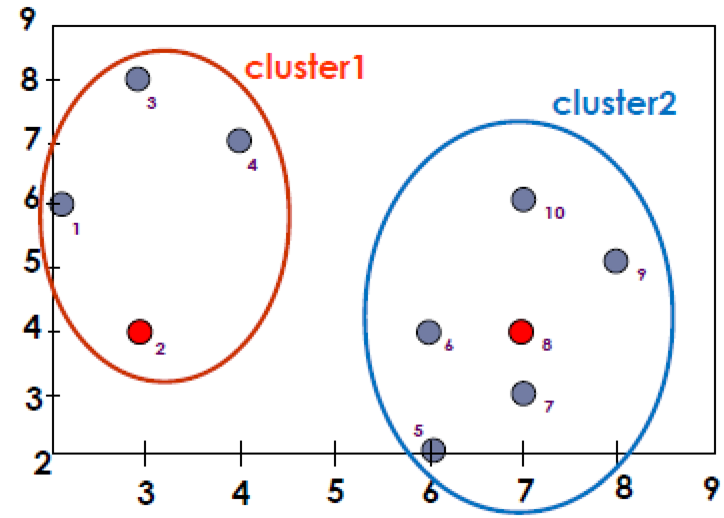

**Goal: create two clusters**

Choose randmly two medoids

$O_2 = (3,4)$
$O_8 = (7,4)$

## Data Objects

| | $A_1$ | $A_2$ |
|---|---|---|
| $O_1$ | 2 | 6 |
| $O_2$ | 3 | 4 |
| $O_3$ | 3 | 8 |
| $O_4$ | 4 | 7 |
| $O_5$ | 6 | 2 |
| $O_6$ | 6 | 4 |
| $O_7$ | 7 | 3 |
| $O_8$ | 7 | 4 |
| $O_9$ | 8 | 5 |
| $O_{10}$ | 7 | 6 |




→Assign each object to the closest representative object

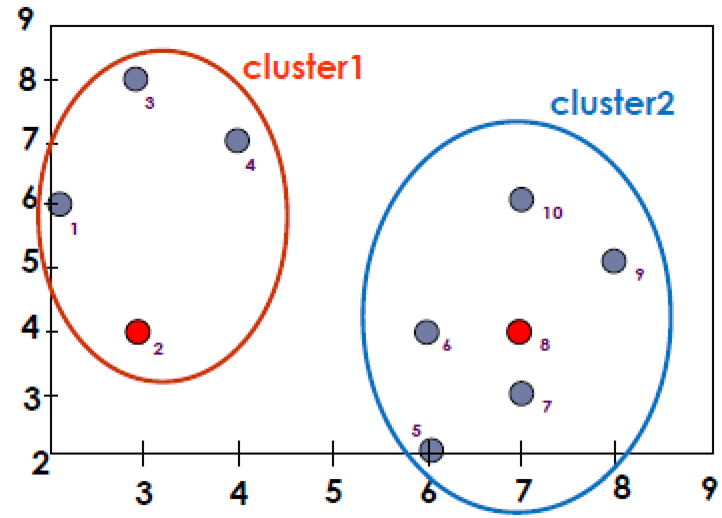
→Using L1 Metric (Manhattan), we form the following clusters

Cluster1 = {$O_1$, $O_2$, $O_3$, $O_4$}

Cluster2 = {$O_5$, $O_6$, $O_7$, $O_8$, $O_9$, $O_{10}$}

Data Objects

| | $A_1$ | $A_2$ |
|---|---|---|
| $O_1$ | 2 | 6 |
| $O_2$ | 3 | 4 |
| $O_3$ | 3 | 8 |
| $O_4$ | 4 | 7 |
| $O_5$ | 6 | 2 |
| $O_6$ | 6 | 4 |
| $O_7$ | 7 | 3 |
| $O_8$ | 7 | 4 |
| $O_9$ | 8 | 5 |
| $O_{10}$ | 7 | 6 |



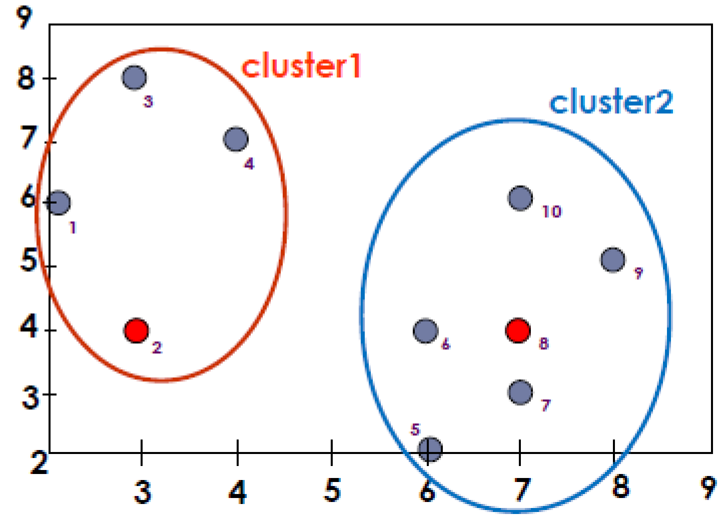

→Compute the absolute error criterion **[for the set of Medoids (O2,O8)]**

$$E=\sum_{i=1}^{k}\sum_{p\in C_i}|p-o_i|=|o_1-o_2|+|o_3-o_2|+|o_4-o_2|$$

$$+|o_5-o_8|+|o_6-o_8|+|o_7-o_8|+|o_9-o_8|+|o_{10}-o_8|$$

**Data Objects**

| | $A_1$ | $A_2$ |
|---|---|---|
| $O_1$ | 2 | 6 |
| $O_2$ | 3 | 4 |
| $O_3$ | 3 | 8 |
| $O_4$ | 4 | 7 |
| $O_5$ | 6 | 2 |
| $O_6$ | 6 | 4 |
| $O_7$ | 7 | 3 |
| $O_8$ | 7 | 4 |
| $O_9$ | 8 | 5 |
| $O_{10}$ | 7 | 6 |

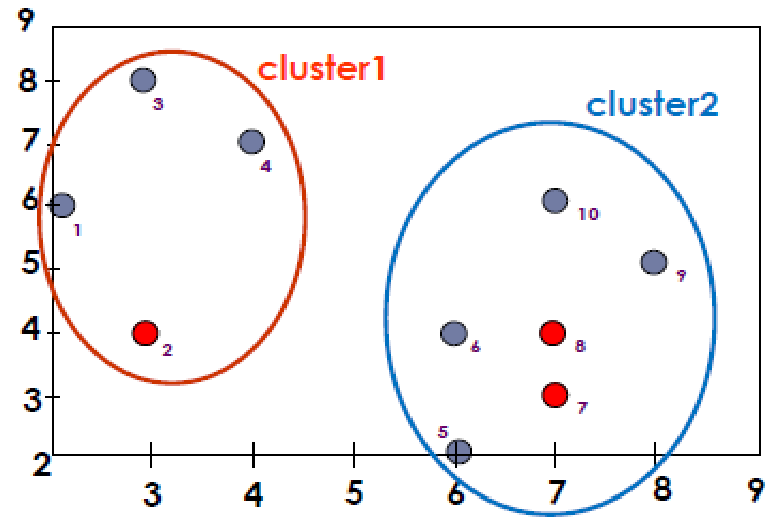



→The absolute error criterion **[for the set of Medoids (O2,O8)]**

$$E = (3+4+4)+(3+1+1+2+2) = 20$$

**Data Objects**

| | $A_1$ | $A_2$ |
|---|---|---|
| $O_1$ | 2 | 6 |
| $O_2$ | 3 | 4 |
| $O_3$ | 3 | 8 |
| $O_4$ | 4 | 7 |
| $O_5$ | 6 | 2 |
| $O_6$ | 6 | 4 |
| $O_7$ | 7 | 3 |
| $O_8$ | 7 | 4 |
| $O_9$ | 8 | 5 |
| $O_{10}$ | 7 | 6 |




→Choose a random object $O_7$

→Swap O8 and O7

→Compute the absolute error criterion **[for the set of Medoids (O2,O7)]**

$$E = (3+4+4)+(2+2+1+3+3) = 22$$

→Compute the cost function

Absolute error [for $O_2$,$O_7$] – Absolute error [$O_2$,$O_8$]

$$S = 22 - 20$$

S> 0 ⇒ it is a bad idea to replace $O_8$ by $O_7$

# Agglomerative clustering

- Idea: ensure nearby points ends up in the same cluster
- Start with a collection  of n singleton clusters
  - Each cluster contains one data point
- Repeatedly only one cluster is left:
  - Find a pair of clusters that is closest: min $D(c_i,c_j)$
  - Merge the clusters ci, cj into a new cluster $c_{ij}$
  - Remove $c_i$ and $c_j$ from collection C and add $c_{ij}$
- Produce a dendrogram: hierarchical tree of clusters

# Example

- For a given dataset, Form the distance matrix

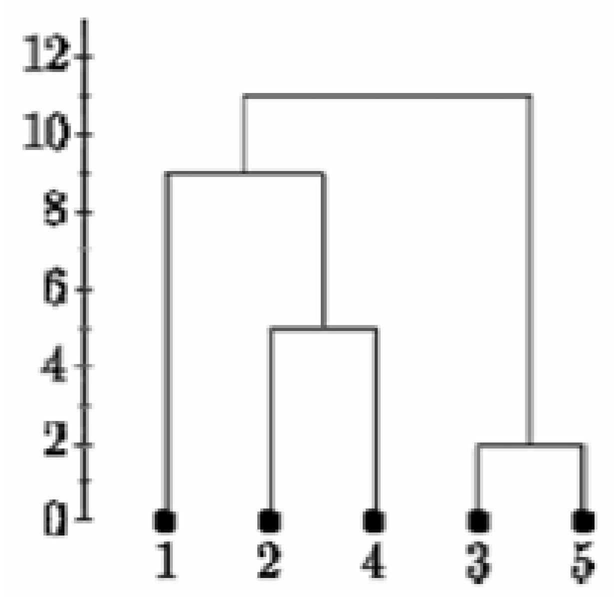
| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | | | | |
| 2 | 9 | 0 | | | |
| 3 | 3 | 7 | 0 | | |
| 4 | 6 | 5 | 9 | 0 | |
| 5 | 11 | 10 | $\boxed{2}$ | 8 | 0 |

- Merge col 3 and 5 . For example, d(1,3)= 3 and d(1,5)=11. So, D(1,"35")=11. This gives us the new distance matrix. The items with the smallest distance get clustered next. This will be 2 and 4.

| | 35 | 1 | 2 | 4 |
|---|---|---|---|---|
| 35 | 0 | | | |
| 1 | 11 | 0 | | |
| 2 | 10 | 9 | 0 | |
| 4 | 9 | 6 | 5 | 0 |

| | 35 | 24 | 1 |
|---|---|---|---|
| 35 | 0 | | |
| 24 | 10 | 0 | |
| 1 | 11 | 9 | 0 |

Form the dendrogram.



# Practice

$$\begin{array}{cc} & \begin{array}{cc} X1 & X2 \end{array} \\ \begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \end{array} & \left\{ \begin{array}{cc} 1 & 1 \\ 1.5 & 1.5 \\ 5 & 5 \\ 3 & 4 \\ 4 & 4 \\ 3 & 3.5 \end{array} \right\} \end{array}$$

# Modern Clustering methods

- Hierarchical clustering
  - BIRCH – Balanced Iterative reducing and clustering using hierarchies
  - CURE – clustering using Representation
  - ROCK – Robust clustering for categorical data
- Partitive clustering
  - CLARA – Clustering Large Application
  - CLARANS – clustering large application on randomized search
  - K-mode

# Other Clustering methods

- Density based clustering
  - DENCLUE – Density based clustering
  - DBSCAN- Density based spatial clustering of application with noise
  - Optics – ordering points to identify the clustering structure
- Grid based methods
  - STING – statistical information Grid based method
  - Wave cluster
- Model based methods
  - COBWEB
  - CLASSIT

THE END