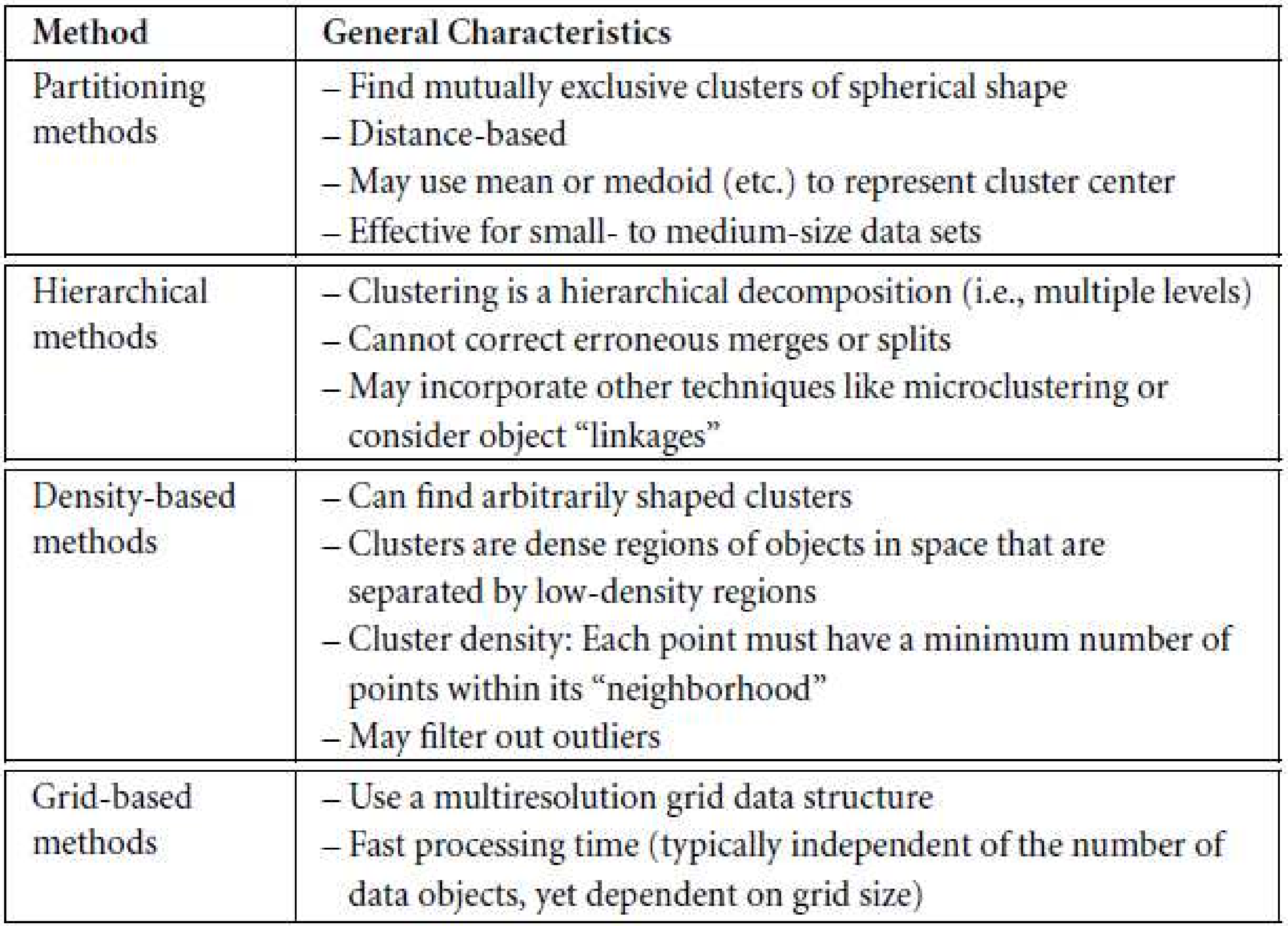
Partitions Methods

Major Clustering Approaches

- **Partitioning algorithms:** Construct various partitions and then evaluate them by some criterion
- **Hierarchy algorithms:** Create a hierarchical decomposition of the set of data (or objects) using some criterion.
  - Can be agglomerative and divisive approach
- **Density-based:** based on connectivity and density functions
  - Grow the cluster as long as the density in the neighborhood exceeds some threshold
- **Grid-based:** based on a multiple-level granularity structure
- **Model-based:** A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other

Major Clustering Approaches



Method	General Characteristics
Partitioning methods	– Find mutually exclusive clusters of spherical shape – Distance-based – May use mean or medoid (etc.) to represent cluster center – Effective for small- to medium-size data sets
Hierarchical methods	– Clustering is a hierarchical decomposition (i.e., multiple levels) – Cannot correct erroneous merges or splits – May incorporate other techniques like microclustering or consider object "linkages"
Density-based methods	– Can find arbitrarily shaped clusters – Clusters are dense regions of objects in space that are separated by low-density regions – Cluster density: Each point must have a minimum number of points within its "neighborhood" – May filter out outliers
Grid-based methods	– Use a multiresolution grid data structure – Fast processing time (typically independent of the number of data objects, yet dependent on grid size)

Partitioning Algorithms: Basic Concepts

- **Partitioning method:** Discovering the groupings in the data by optimizing a specific objective function and iteratively improving the quality of partitions

- ***K*-partitioning method:** Partitioning a dataset ***D*** of ***n*** objects into a set of ***K*** clusters so that an objective function is optimized (e.g., the sum of squared distances is minimized, where c_k is the centroid or medoid of cluster C_k)
  - A typical objective function: **Sum of Squared Errors (SSE)**

$$SSE(C) = \sum_{k=1}^{K} \sum_{x_{i \in C_k}} \| x_i - c_k \|^2$$

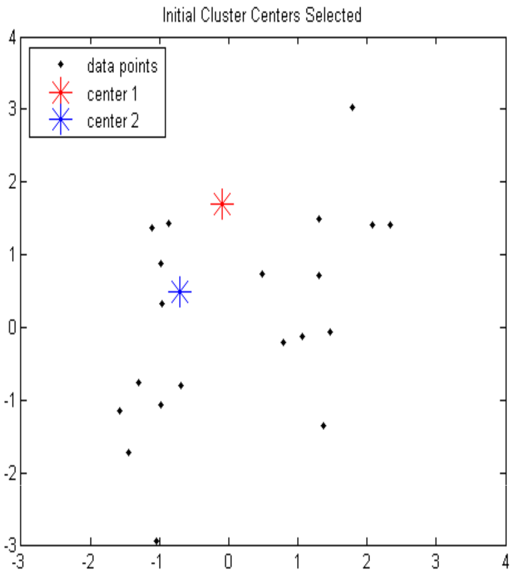
Partitioning Algorithms: Basic Concepts

- Problem definition: Given K, find a partition of K clusters that optimizes the chosen partitioning criterion
  - **Global optimal:** Needs to exhaustively enumerate all partitions
  - Heuristic methods (i.e., greedy algorithms): K-Means, K-Medians, K-Medoids, etc.

The *K*-Means Clustering Method

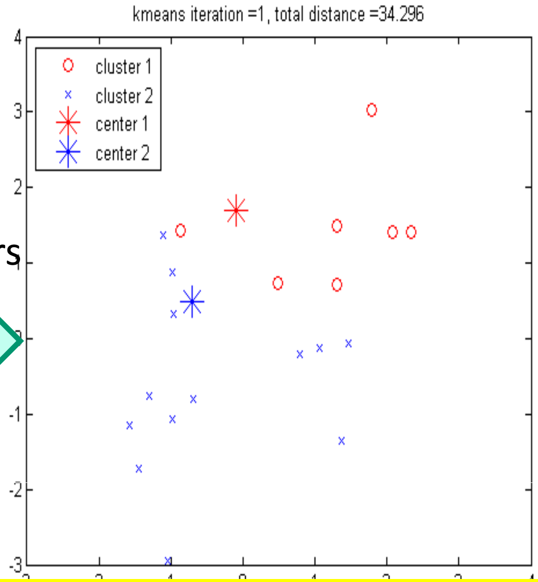
- *K-Means* :Each cluster is represented by the center of the cluster
- Given K, the number of clusters, the *K*-Means clustering algorithm is outlined as follows
  - Select *K* points as initial centroids
  - **Repeat**
    - Form *K* clusters by assigning each point to its closest centroid based on the distance measures
    - Re-compute the centroids (i.e., *mean point*) of each cluster
  - **Until** convergence criterion is satisfied
- Different kinds of measures can be used
  - Manhattan distance (L_1 norm), Euclidean distance (L_2 norm), Cosine similarity

Example: K-Means Clustering

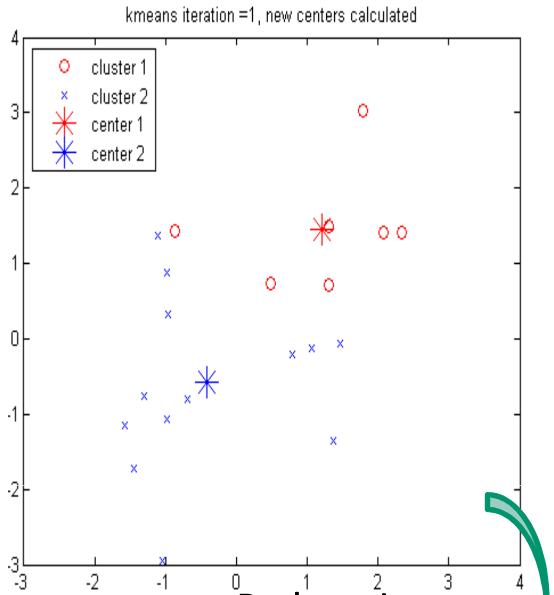


points & randomly select K = 2 centroids

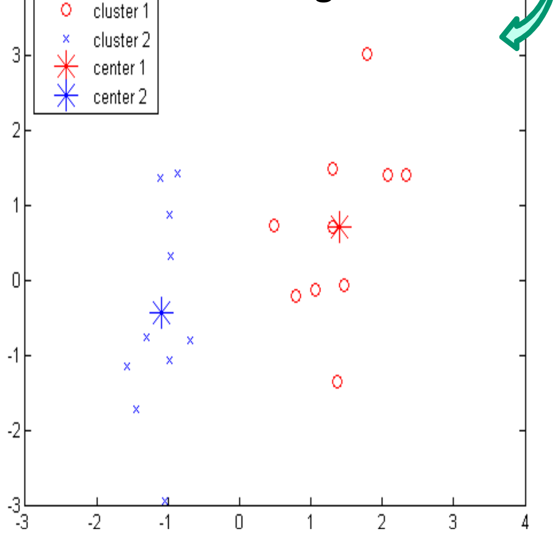
Assign points to clusters



Recompute cluster centers



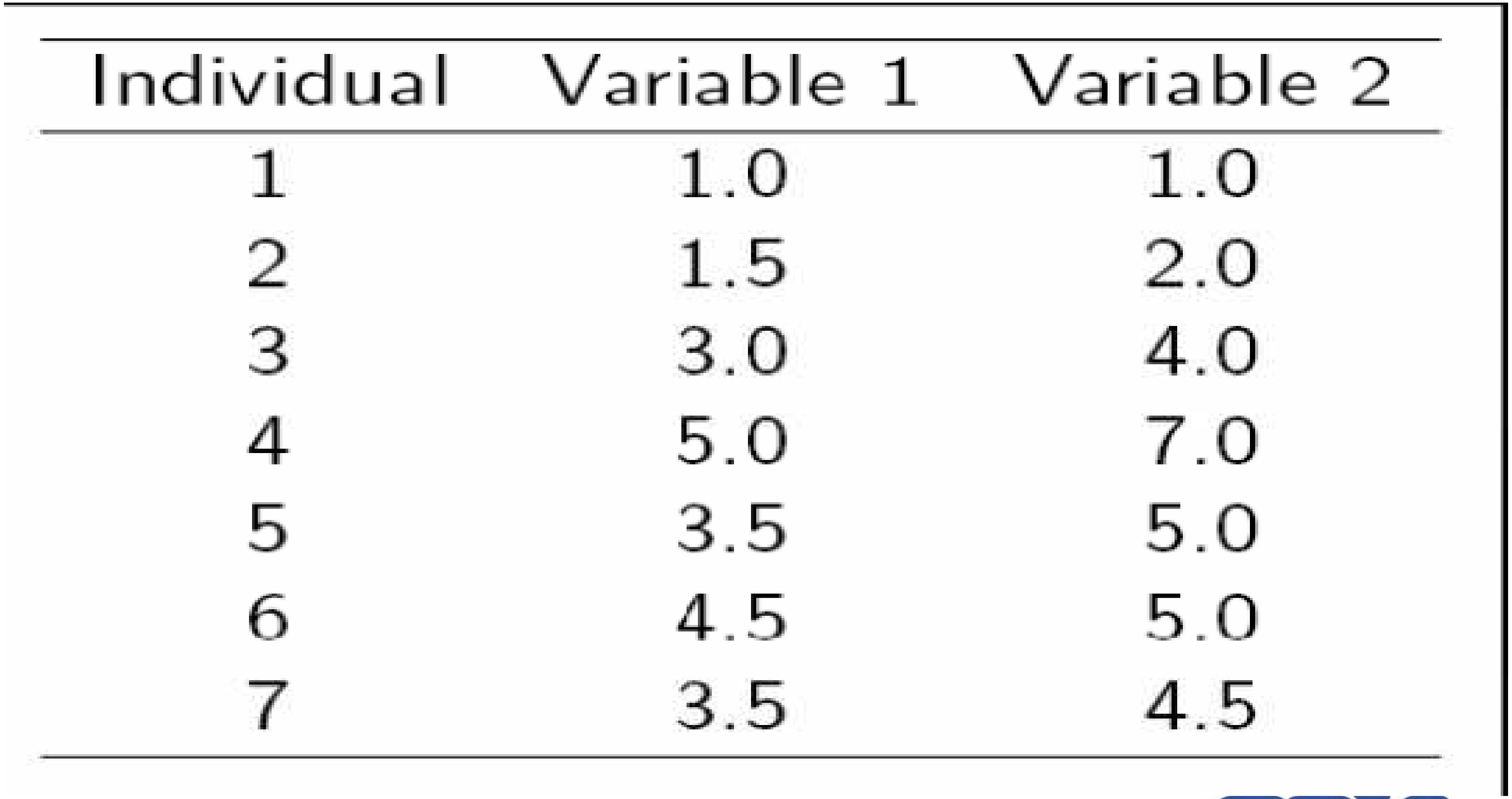
Redo point assignment



Execution of the K-Means Clustering Algorithm

```
Select $K$ points as initial centroids
Repeat
    • Form $K$ clusters by assigning each point to its closest centroid
    • Re-compute the centroids (i.e., mean point) of each cluster
Until convergence criterion is satisfied
```

A Simple example showing the implementation of k-means algorithm



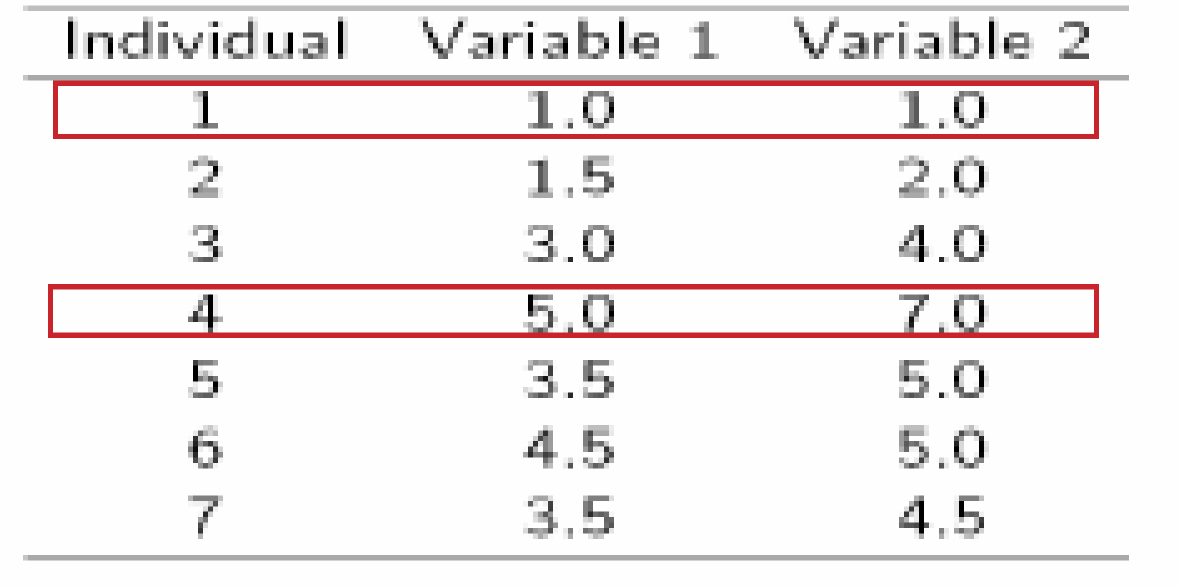
Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Example: *K-Means* Clustering

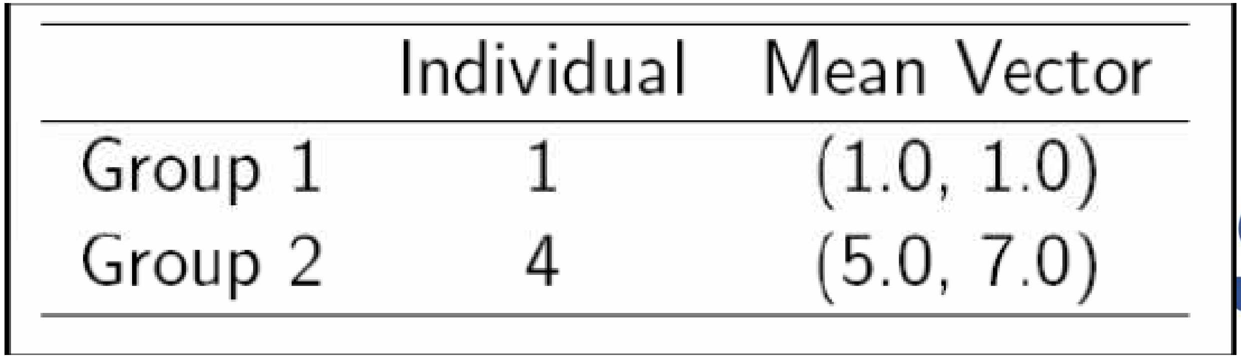
Step 1:

Initialization: Randomly we choose following two centroids (k=2) for two clusters.

In this ca 0,7.0).



Individual	Variable 1	Variable 2
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



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

Step 2:

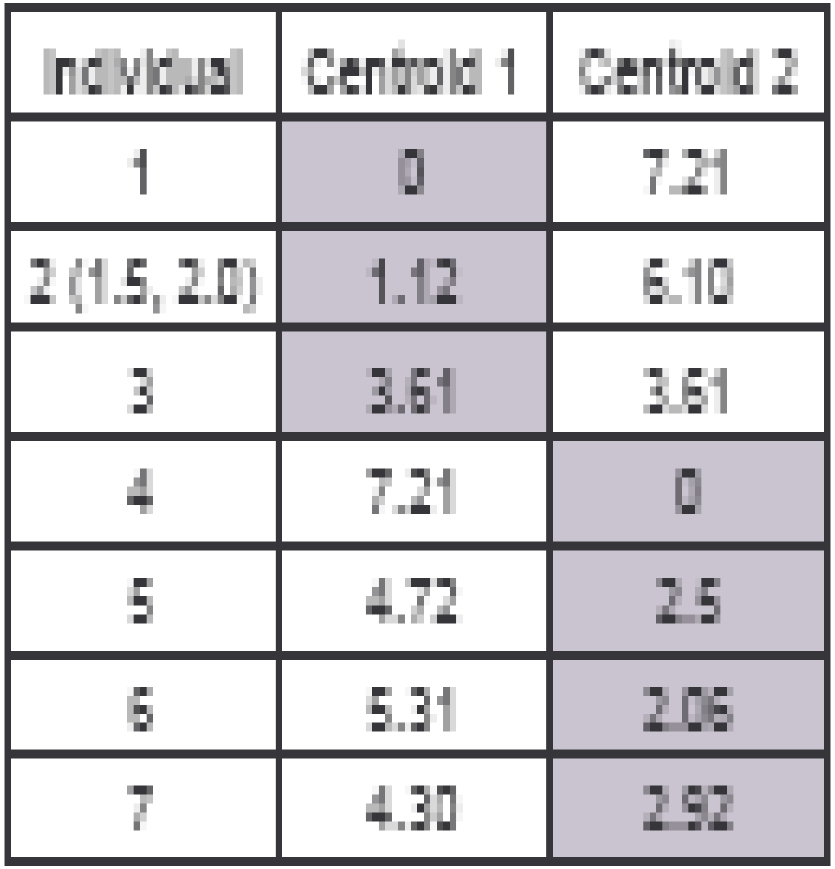
- Thus, we obtain two clusters containing:

$$m_1 = \left(\frac{1}{3}(1.0+1.5+3.0), \frac{1}{3}(1.0+2.0+4.0)\right) = (1.83, 2.33)$$

$$m_2 = \left(\frac{1}{4}(5.0+3.5+4.5+3.5), \frac{1}{4}(7.0+5.0+5.0+4.5)\right)$$

- Their new centroids are:

$$= (4.12, 5.38)$$



Individual	Centroid 1	Centroid 2
1	0	7.21
2 (1.5, 2.0)	1.12	6.10
3	3.61	3.61
4	7.21	0
5	4.72	2.5
6	5.31	2.06
7	4.30	2.92

$$d(m_1, 2) = \sqrt{|1.0-1.5|^2 + |1.0-2.0|^2} = 1.12$$

$$d(m_2, 2) = \sqrt{|5.0-1.5|^2 + |7.0-2.0|^2} = 6.10$$

Example: *K-Means* Clustering

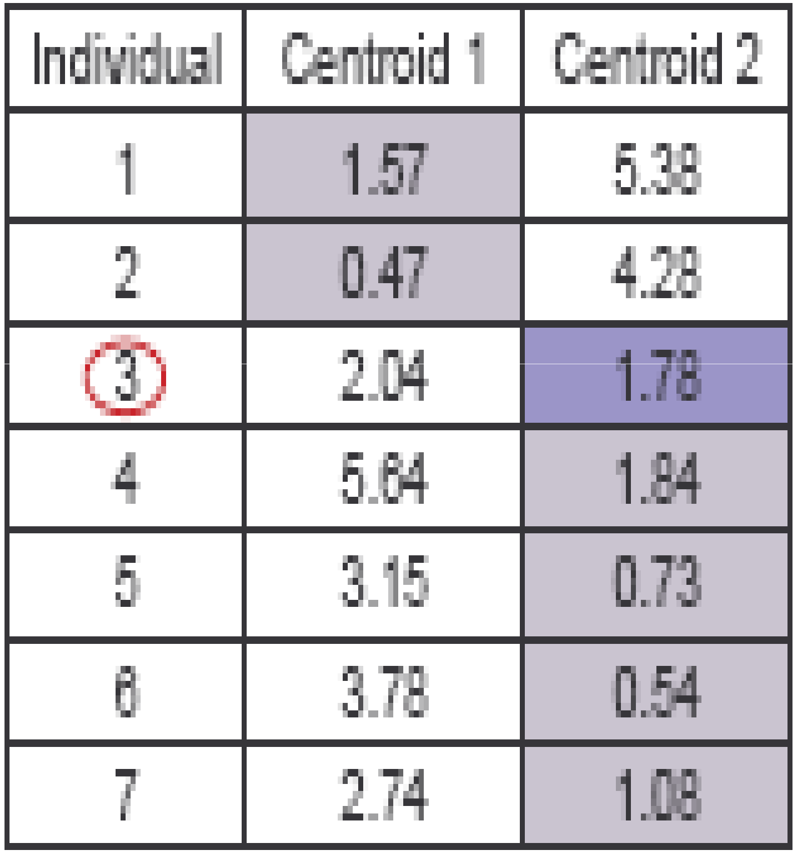
Step 3:

- *Now using these centroids we compute the Euclidean distance of each object, as shown in table.*

- Therefore, the new clusters are:

  {1,2} and {3,4,5,6,7}

- Next centroids are: m1=(1.25,1.5) and m2 = (3.9,5.1)



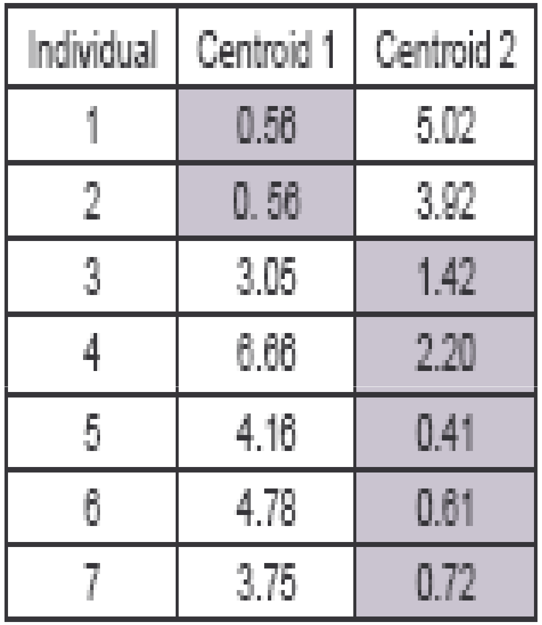
Individual	Centroid 1	Centroid 2
1	1.57	5.38
2	0.47	4.28
3	2.04	1.78
4	5.64	1.84
5	3.15	0.73
6	3.78	0.54
7	2.74	1.08

Example: *K-Means* Clustering

- Step 4 :

  The clusters obtained are:

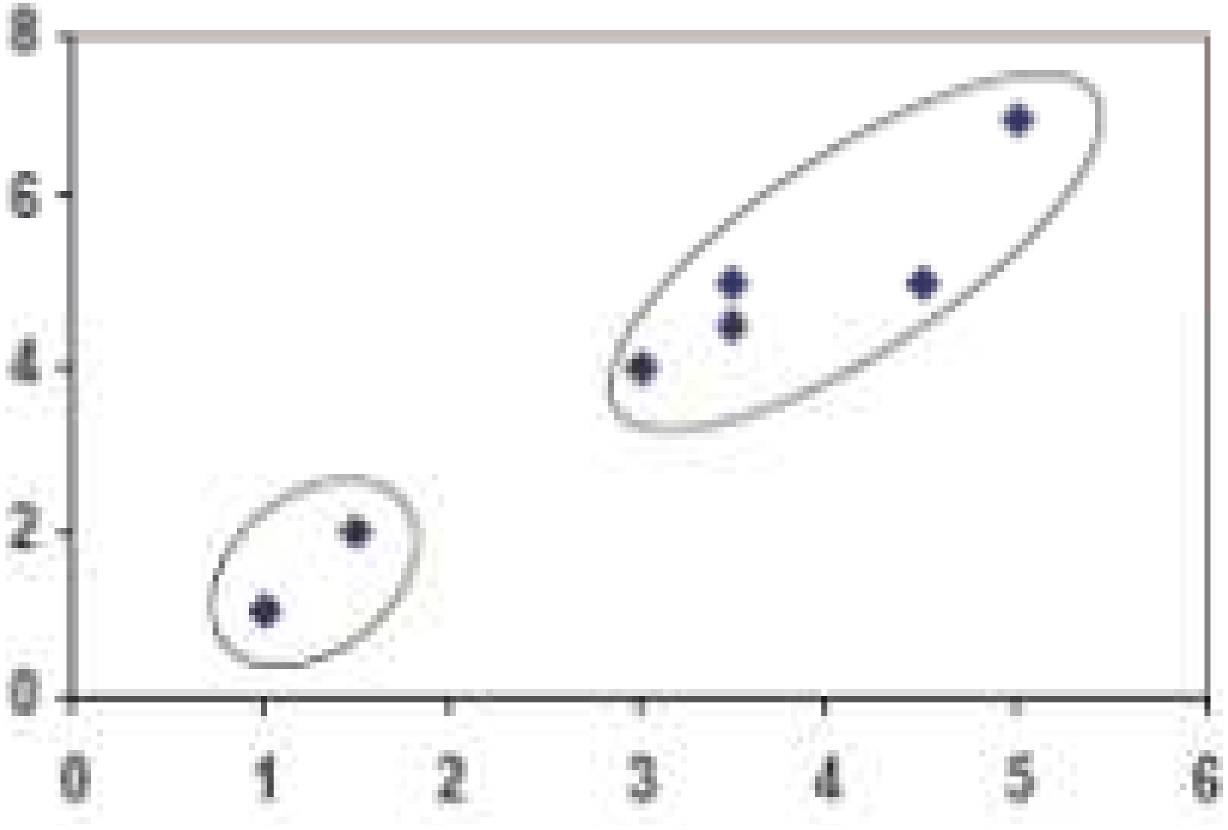
  {1,2} and {3,4,5,6,7}



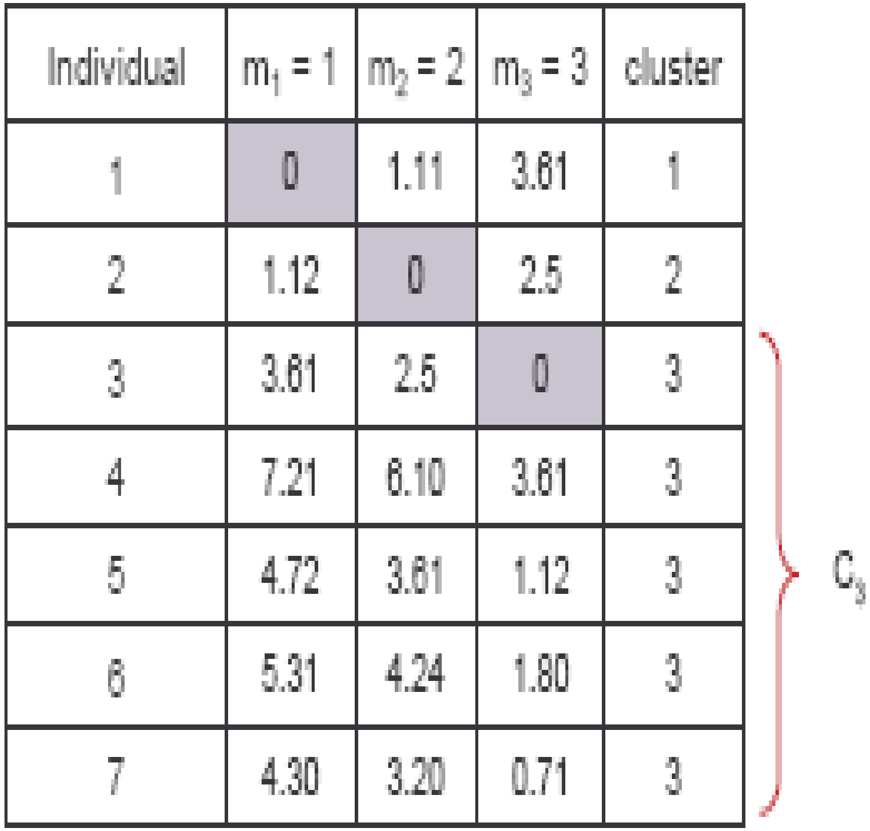
Individual	Centroid 1	Centroid 2
1	0.56	5.02
2	0. 56	3.92
3	3.05	1.42
4	6.66	2.20
5	4.16	0.41
6	4.78	0.61
7	3.75	0.72

- Therefore, there is no change in the cluster.
- Thus, the algorithm comes to a halt here and final result consist of 2 clusters {1,2} and {3,4,5,6,7}.

PLOT



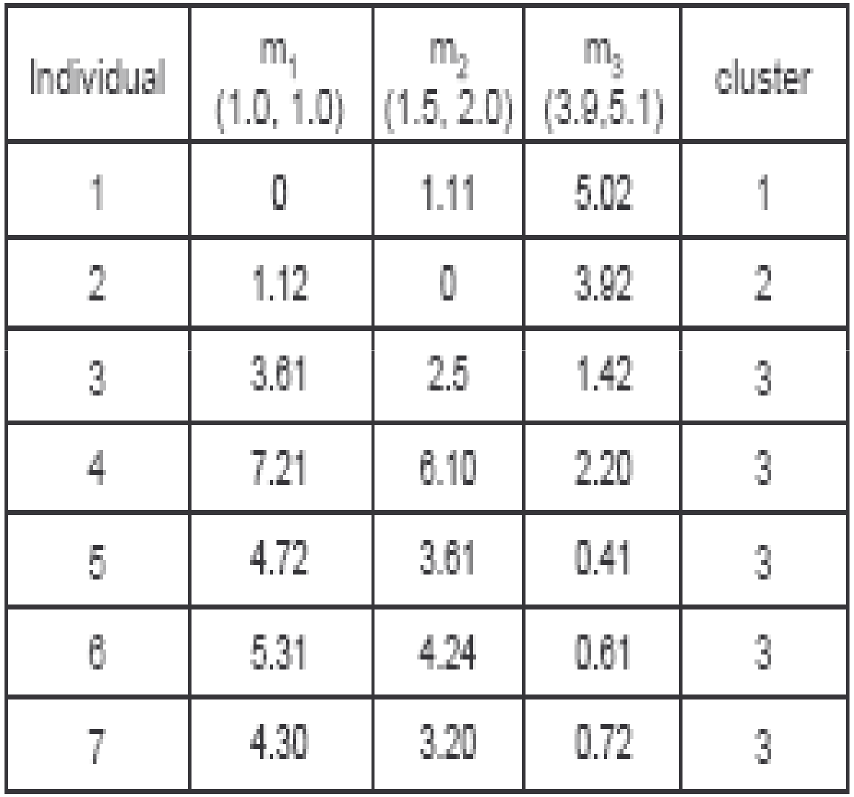
(with K=3)



Individual	$m_1 = 1$	$m_2 = 2$	$m_3 = 3$	cluster
1	0	1.11	3.61	1
2	1.12	0	2.5	2
3	3.61	2.5	0	3
4	7.21	6.10	3.61	3
5	4.72	3.61	1.12	3
6	5.31	4.24	1.80	3
7	4.30	3.20	0.71	3

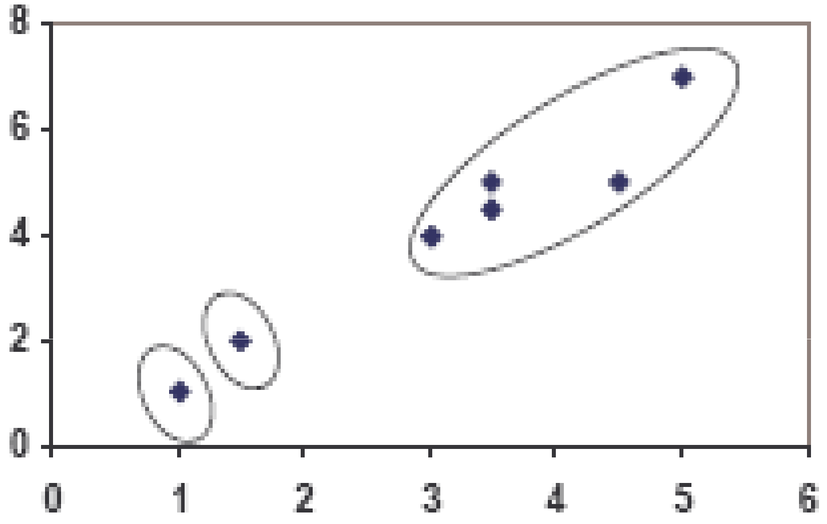
C_3

clustering with initial centroids (1, 2, 3)



Individual	m_1 (1.0, 1.0)	m_2 (1.5, 2.0)	m_3 (3.9,5.1)	cluster
1	0	1.11	5.02	1
2	1.12	0	3.92	2
3	3.61	2.5	1.42	3
4	7.21	6.10	2.20	3
5	4.72	3.61	0.41	3
6	5.31	4.24	0.61	3
7	4.30	3.20	0.72	3

PLOT



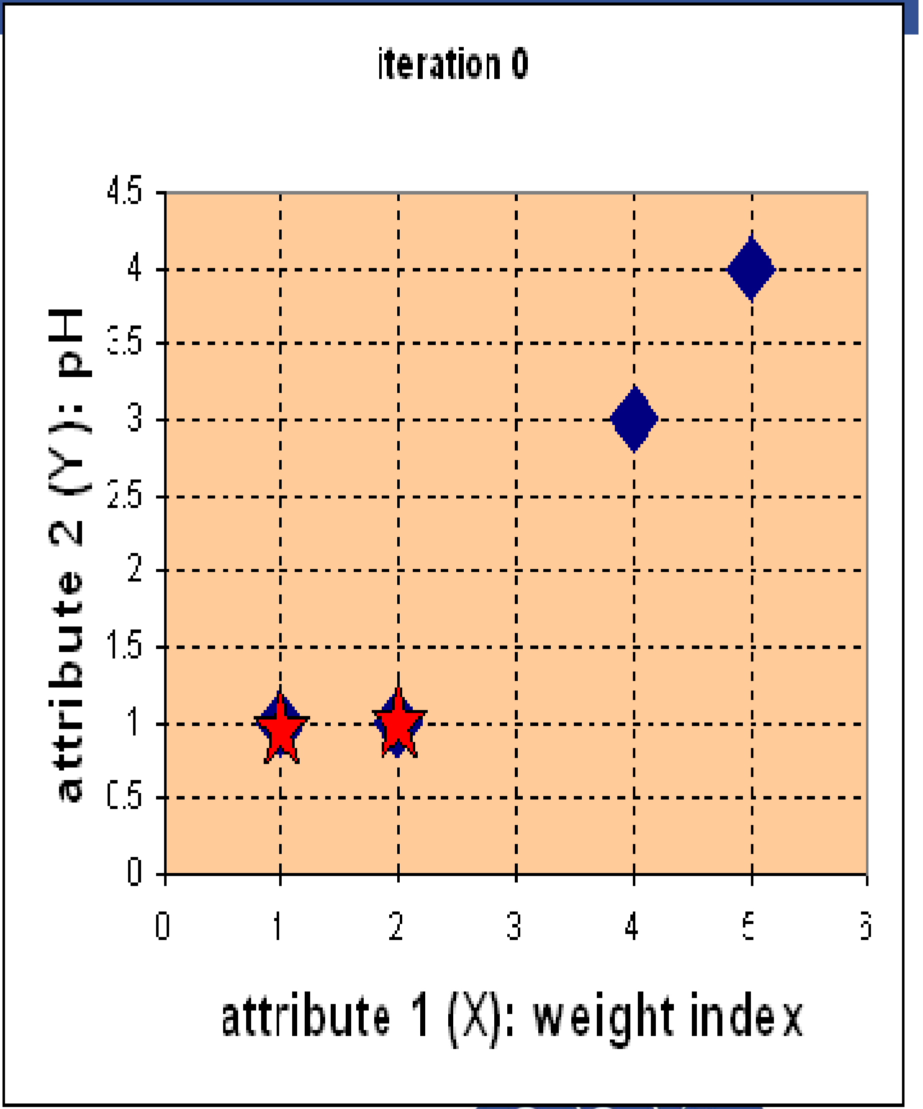
Real-Life Numerical Example of K-Means Clustering

Object	Attribute1 (X): weight index	Attribute 2 (Y): pH
Medicine A	1	1
Medicine B	2	1
Medicine C	4	3
Medicine D	5	4

Real-Life Numerical Example of K-Means Clustering

Step 1:

- **Initial value of centroids** : Suppose we use medicine A and medicine B as the first centroids.
- Let c_1 and c_2 denote the coordinate of the centroids, then c_1=(1,1) and c_2=(2,1)



Real-Life Numerical Example of K-Means Clustering

- ***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

$$\mathbf{D}^0 = \begin{bmatrix} 0 & 1 & 3.51 & 5 \\ 1 & 0 & 2.33 & 4.24 \end{bmatrix} \quad \begin{array}{ll} \mathbf{c}_1 = (1,1) & group-1 \\ \mathbf{c}_2 = (2,1) & group-2 \end{array}$$

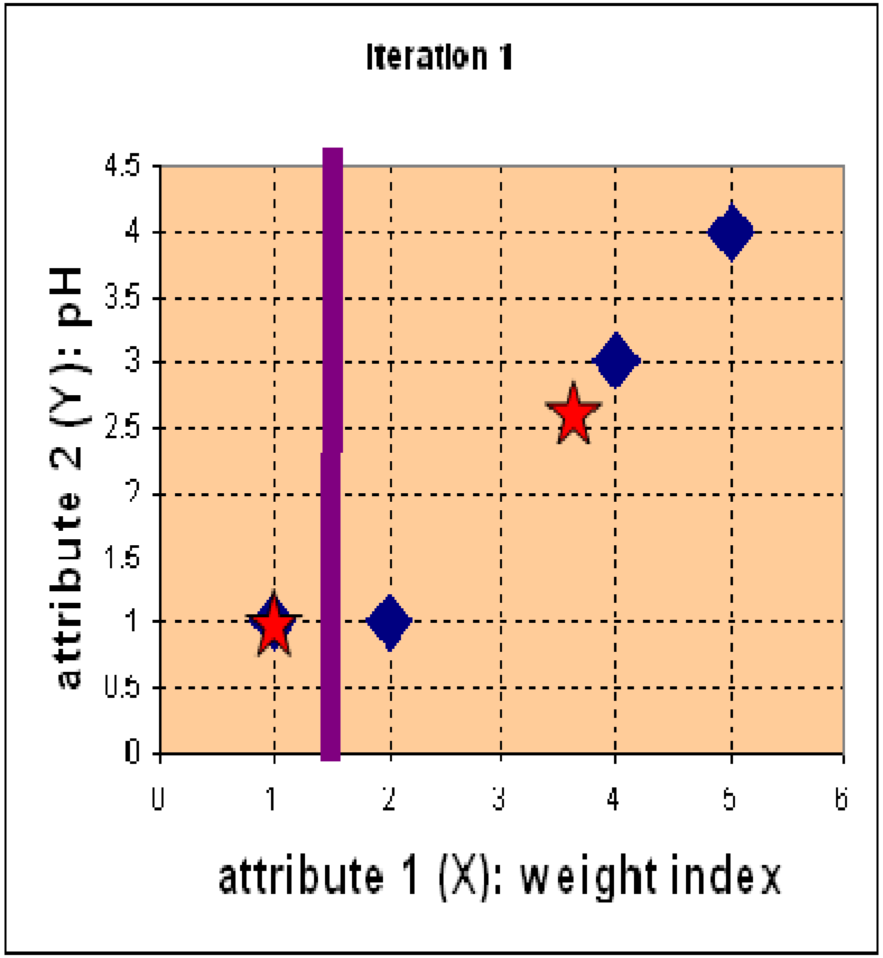
$$\begin{array}{ccccc} A & B & C & D & \\ 1 & 2 & 4 & 5 & X \\ 1 & 1 & 3 & 4 & Y \end{array}$$

- 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 is $\mathbf{c}_2 = (2,1)$ is $\sqrt{(4-2)^2 + (3-1)^2} = 2.83$ etc.

Step 2:

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

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



Real-Life Numerical Example of K-Means Clustering

- **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

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

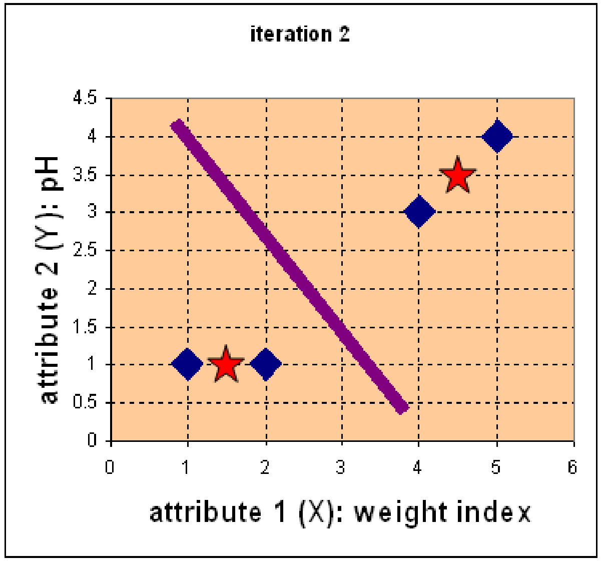
$$\begin{array}{cccc|c} A & B & C & D & \\ 1 & 2 & 4 & 5 & X \\ 1 & 1 & 3 & 4 & Y \end{array}$$

Real-Life Numerical Example of K-Means Clustering

- **Iteration-1, Objects clustering:** 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

$$\mathbf{G}^1 = \begin{array}{cccccc} \left[\begin{matrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{matrix}\right] & \begin{matrix} group-1 \\ group-2 \end{matrix} \\ \begin{matrix} A & \;B & \;C & \;D \end{matrix} & \end{array}$$

- **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 $\mathbf{c}_2 = (\frac{4+5}{2}, \frac{3+4}{2}) = (4\frac{1}{2}, 3\frac{1}{2})$



Real-Life Numerical Example of K-Means Clustering

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

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

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

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

Real-Life Numerical Example of K-Means Clustering

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

$$\mathbf{G}^2 = \begin{array}{c} \\ \\ \end{array}\begin{array}{cccc|l} \left[\begin{matrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{matrix}\right] & \begin{matrix} group-1 \\ group-2 \end{matrix} \\ \begin{matrix} A & B & C & D \end{matrix} & \end{array}$$

- We obtain result that $\mathbf{G}^2 = \mathbf{G}^1$. 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..

Real-Life Numerical Example of K-Means Clustering

We get the final grouping as the results as:

Object	Feature1(X): weight index	Feature2 (Y): pH	Group (result)
Medicine A	1	1	1
Medicine B	2	1	1
Medicine C	4	3	2
Medicine D	5	4	2

Discussion on the K-Means Method

- **Efficiency**: O(tKn) where n: # of objects, K: # of clusters, and t: # of iterations
  - Normally, K, t << n; thus, an efficient method
- K-means clustering often **terminates at a local optimal**
  - Initialization can be important to find high-quality clusters
- **Need to specify K**, the number of clusters, in advance
  - There are ways to automatically determine the "best" K
  - In practice, one often runs a range of values and selected the "best" K value

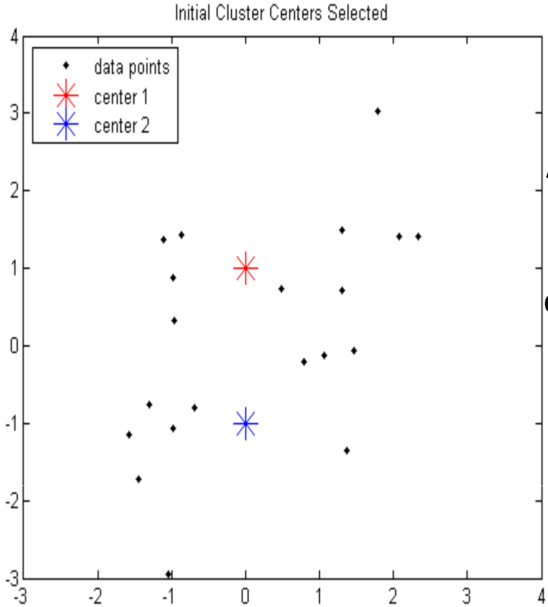
Discussion on the K-Means Method

- **Sensitive to noisy data and outliers**
  - Variations: Using K-medians, K-medoids, etc.
- K-means is applicable only to objects in a continuous n-dimensional space
  - Using the K-modes for **categorical data**
- Not suitable to discover clusters with **non-convex shapes**
  - Using density-based clustering, kernel K-means, etc.

Initialization of K-Means

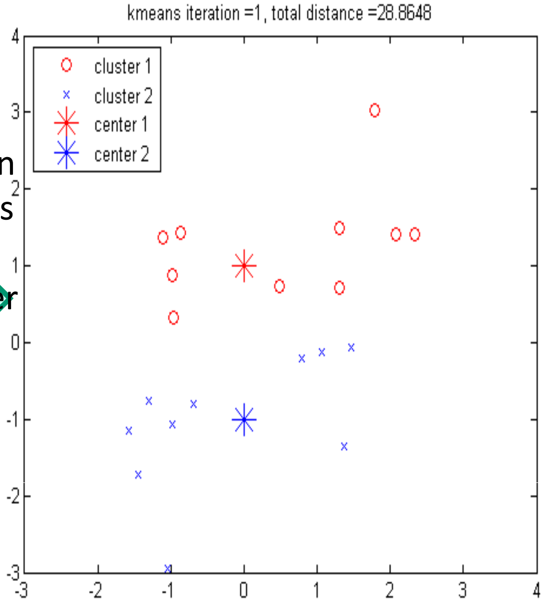
- Different initializations may generate rather different clustering results (some could be far from optimal)
- Original proposal : Select *K* seeds randomly
  - Need to run the algorithm multiple times using different seeds
- There are many methods proposed for better initialization of *k* seeds
  - The first centroid is selected at random
  - **K-Means++,k-median,k-mediods**

Example: Poor Initialization May Lead to Poor Clustering

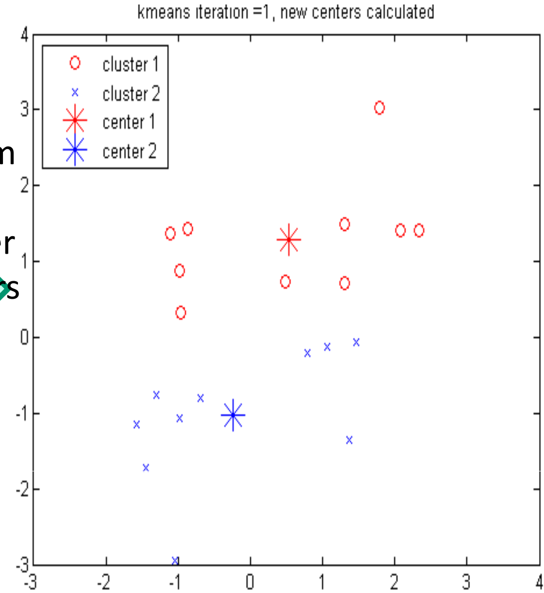


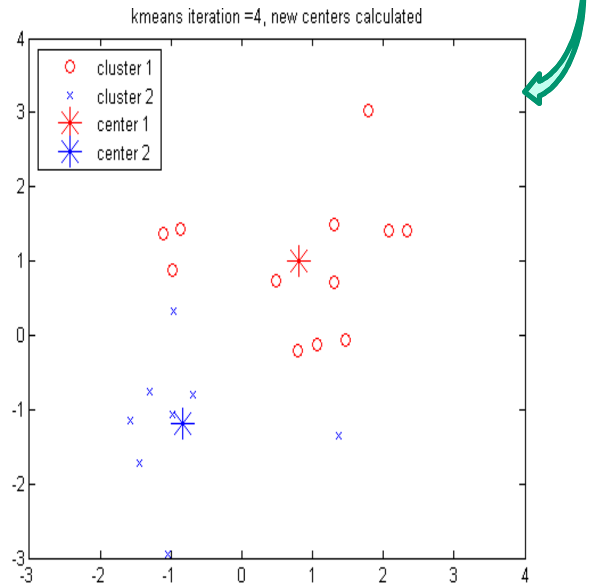
Another random selection of k centroids for the same data points

Assign points to clusters



Recompute cluster centers





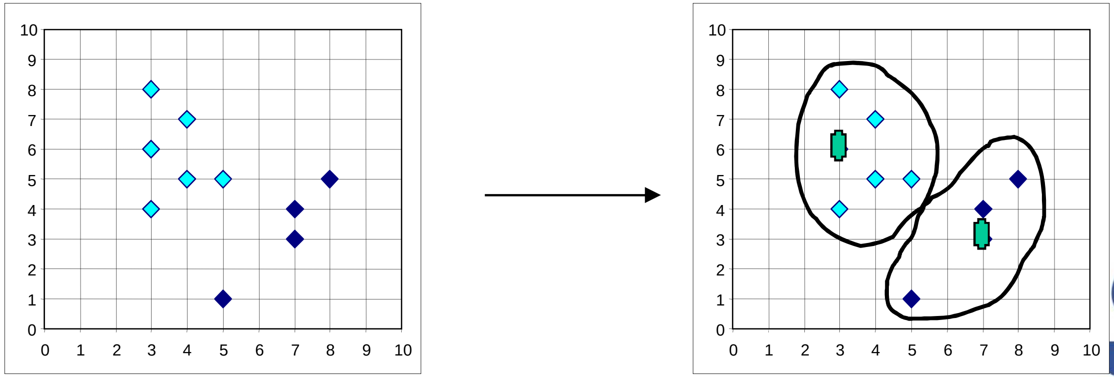
- ❑ Rerun of the K-Means using another random K seeds
- ❑ This run of K-Means generates a poor quality clustering

K-Mediods

- Consider six points in 1-D space having the values
- 1, 2, 3, 8, 9, 10, and 25, respectively.
- By visual inspection we may partition the points into the clusters {1,2,3} and {8, 9,10} where point 25 is excluded which is an outlier.
- How would k-means partition the values?
- If we apply k-means using mean 2 and 9 and c1{1,2,3} {8,9,10,25} with cluster variation as 196
- With mean as 3.5 and 14.67 for c1{1,2,3,8} and c2{9,10,25} the cluster variation as 189.67
- Assigns 8 to different cluster due to the prsence of outliers

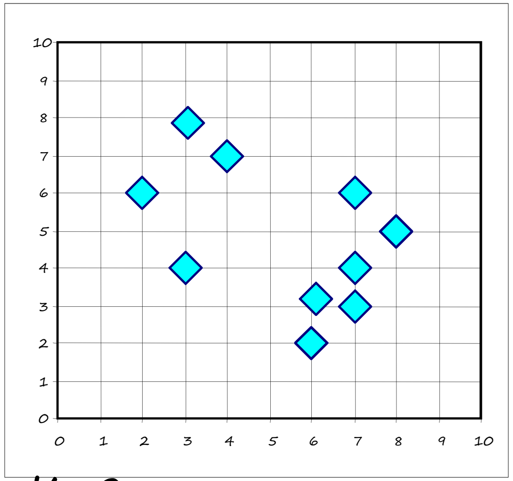
K-Mediods

- Avoid taking the mean value of the object as reference point.
- Actual objects to represent the clusters **medoids** can be used, which is the **most centrally located** object in a cluster
- Assign other similar objects as representative object to the cluster.
- The absolute-error criterion is defined as $E = \sum_{i=1}^{k} \sum_{p \in C_i} dist(p, o_i),$
- O_i representative object and P all objects in the data set



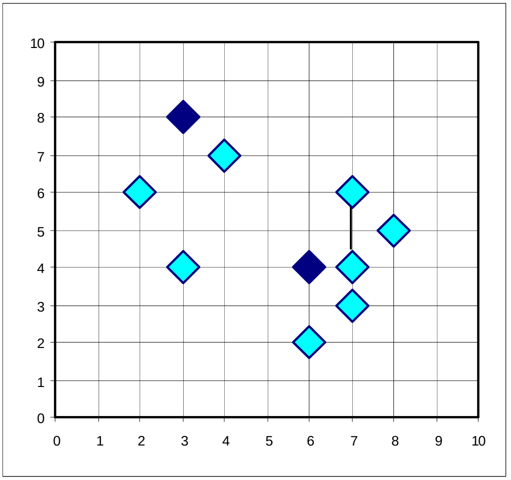
PAM: A Typical K-Medoids Algorithm

Total Cost = 2

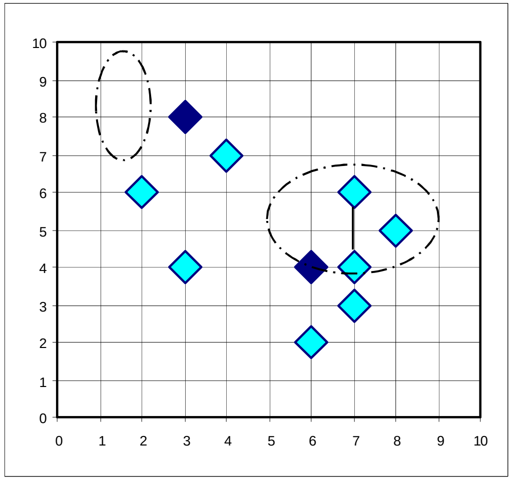


K=2

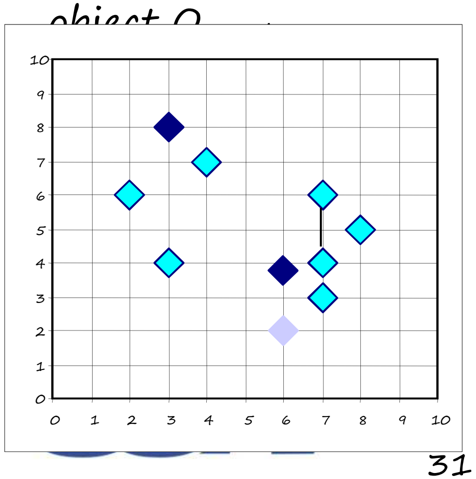
Arbitrary choose k object as initial medoids



Assign each remaining object to nearest medoids

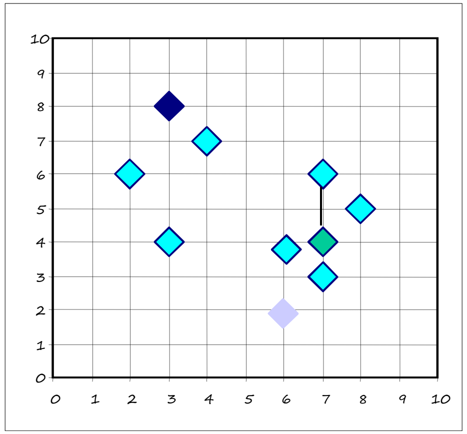


Randomly select a nonmedoid object O



Compute total cost of swapping

Total Cost = 26



Swapping O and O_{ramdom}

If quality is improved.

Do loop

Until no change

Algorithm for PAM

- The *K-Medoids* clustering algorithm:
  - Select *K* points as the initial representative objects (i.e., as initial *K medoids*)
  - **Repeat**
    - Assigning each point to the cluster with the closest medoid
    - Randomly select a non-representative object o_i
    - Compute the total cost S of swapping the medoid m with o_i
    - If $S < 0$, then swap m with o_i to form the new set of medoids
  - **Until** convergence criterion is satisfied

Discussion on *K-Medoids* Clustering

- *K-Medoids* Clustering: Find *representative* objects (medoids) in clusters
- *PAM* (Partitioning Around Medoids:
  - Starts from an initial set of medoids
  - Iteratively replaces one of the medoids by one of the non-medoids if it improves the total sum of the squared errors (SSE) of the resulting clustering
  - *PAM* works effectively for small data sets but does not scale well for large data sets (due to the computational complexity)
  - Computational complexity: PAM: $O(K(n - K)^2)$ (quite expensive!)

Discussion on *K-Medoids* Clustering

- Efficiency improvements on PAM
  - *CLARA* (Kaufmann & Rousseeuw, 1990):
    - PAM on samples; $O(Ks^2 + K(n - K))$, s is the sample size
  - *CLARANS* (Ng & Han, 1994): Randomized re-sampling, ensuring efficiency + quality

K-Medians: Handling Outliers by Computing Medians

- Medians are less sensitive to outliers than means
  - Think of the median salary vs. mean salary of a large firm when adding a few top executives!
- ***K-Medians***: Instead of taking the **mean** value of the object in a cluster as a reference point, **medians** are used (L_1-norm as the distance measure)
- The criterion function for the *K-Medians* algorithm:

$$S = \sum_{k=1}^{K} \sum_{x_{i \in C_k}} | x_{ij} - med_{kj} |$$