K-MEANS CLUSTERING

Dr. Deepak Ranjan Nayak

What is clustering?

- Unsupervised learning –
- Requires data, but no labels –
- Detect patterns e.g. in
  - Group emails or search results
  - Customer shopping patterns
  - Regions of images
- Useful when don’t know what you’re looking for

What is clustering?

- **Input:**
  - **Training samples** $\{\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_n}\} \epsilon \mathbb{R}^{\boldsymbol{n}}$
  - **No labels** $\boldsymbol{y_i}$ **are given**
- **Goal: group input samples into classes of similar objects –cohesive “clusters.”**
  - **high intra-group similarity**
  - **low inter-group similarity**
  - **It is the commonest form of unsupervised learning**

What is clustering?

- **Clustering** is the classification of objects into different groups, or
- more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often according to some defined distance measure.

Aspects of clustering:

1. **A clustering algorithm**:
   1. Partitional clustering (e.g., k-means)
   2. Hierarchical clustering
   3. Model based clustering
   4. Density based clustering
   5. Graph based clustering
2. **A distance or similarity function:**

   - such as Euclidean, Minkowski, cosine, etc.
3. **Clustering quality**
   1. Inter-clusters distance ⇒ maximized
   2. Intra-clusters distance ⇒ minimized

Types of clustering:

1. **Hierarchical algorithms**: these find successive clusters using previously established clusters.
   1. Agglomerative ("bottom-up"): Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters.
   2. Divisive ("top-down"): Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.

2. **Partitional clustering:** Partitional algorithms determine all clusters at once. They include:

- ***K*-means and derivatives**
- Fuzzy *c*-means clustering
- QT clustering algorithm

Common Distance measures:

- *Distance measure* will determine how the *similarity* of two elements is calculated and it will influence the shape of the clusters.

  They include:

1. The **Euclidean distance** (also called 2-norm distance) is given by:

$$d(x, y) = \sqrt[2]{\sum_{i=1}^{p} |x_i - y_i|^2}$$

2. The **Manhattan distance** (also called taxicab norm or 1-norm) is given by:

$$d(x, y) = \sum_{i=1}^{p} |x_i - y_i|$$

3. The **Cosine distance** is given by: (used for text similarity)

$$\text{cosine}(x_i, x_j) = \frac{x_i . x_j}{\|x_i\|.\|x_j\|.}$$

4. The **maximum norm** is given by:

$$d(x,y) = \max_{1 \leq i \leq p} |x_i - y_i|$$

5. The **Mahalanobis distance** corrects data for different scales and correlations in the variables.

$$d(x,y) = \sqrt{(x_i - x_j)C^{-1}(x_i - x_j)}$$

6. **Hamming distance** (sometimes edit distance) measures the minimum number of substitutions required to change one member into another.

Quality of clustering:

Two types of measures of quality

- ***Internal evaluation:*** assign the best score to the algorithm that produces clusters with high similarity within a cluster and low similarity between clusters

  e.g., Davies-Bouldin index

$$DB = \frac{1}{n}\sum_{i=1}^{k} \max_{j\neq i} \frac{\sigma_i+\sigma_j}{d(c_i,c_j)}$$

Cluster quality is evaluated by the clusters and the data

- ***External evaluation:*** evaluated based on data such as known class labels and external benchmarks e.g., Rand index, Jaccard index, f-measure, etc.

$$RI = \frac{TP+TN}{TP+FP+FN+TN}$$

$$J(A,B) = \frac{|A\cap B|}{|A\cup B|} = \frac{TP}{TP+FP+FN}$$

Cluster quality is evaluated by additional data

K-MEANS CLUSTERING

- The **k-means algorithm** is an algorithm to cluster *n* objects based on attributes into *k* partitions, where *k* < *n*.
- It is similar to the expectation-maximization algorithm for mixtures of Gaussians in that they both attempt to find the centers of natural clusters in the data.
- It assumes that the object attributes form a vector space.

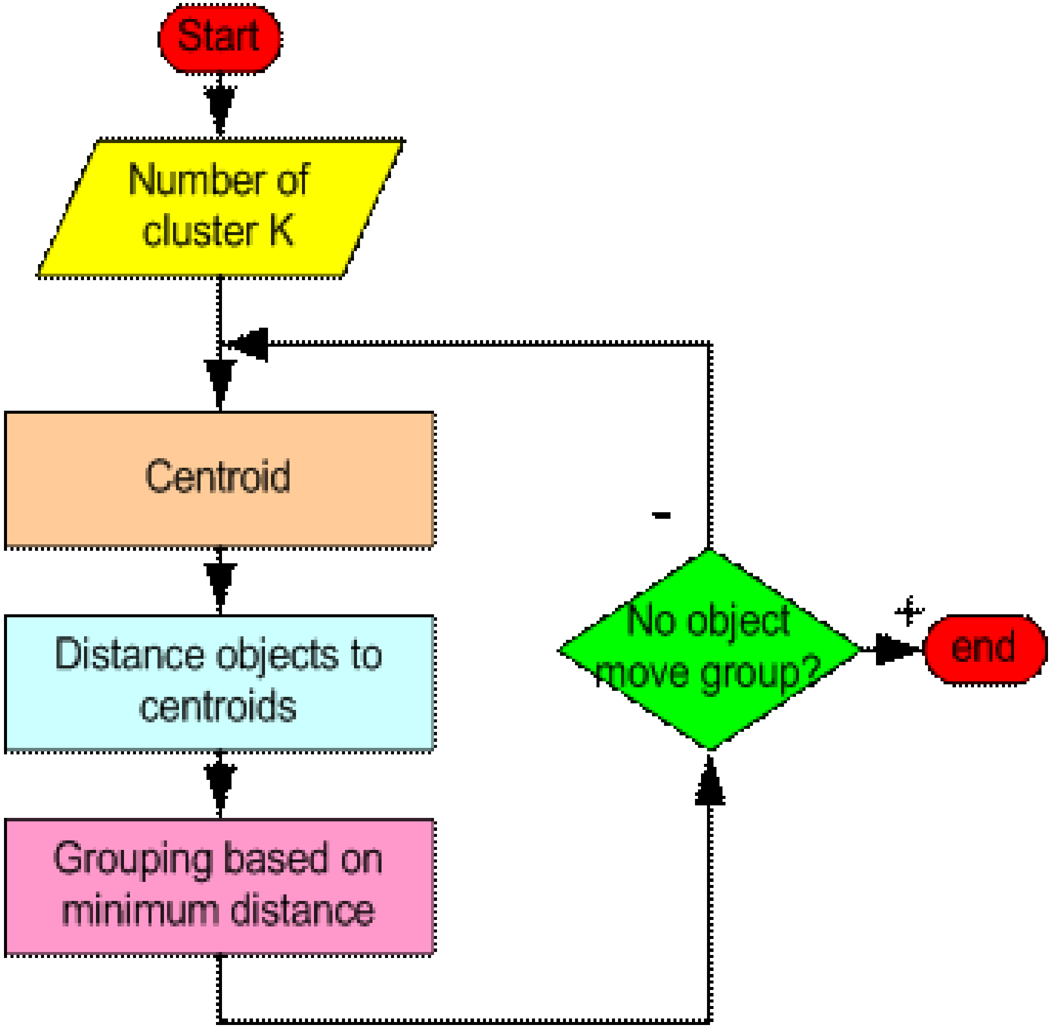
- An algorithm for partitioning (or clustering) N data points into K disjoint subsets S_j containing data points so as to minimize the sum-of-squares criterion

$$J = \sum_{j=1}^{K} \sum_{n \in S_j} \left| x_n - \mu_j \right|^2,$$

where x_n is a vector representing the the nth data point and u_j is the geometric centroid of the data points in S_j.

- Simply speaking k-means clustering is an algorithm to classify or to group the objects based on attributes/features into K number of group.
- K is positive integer number.
- The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid.

How the K-Mean Clustering algorithm works?


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

- **Step 1:** Begin with a decision on the value of k = number of clusters .
- **Step 2**: Put any initial partition that classifies the data into k clusters. You may assign the training samples randomly,or systematically as the following:

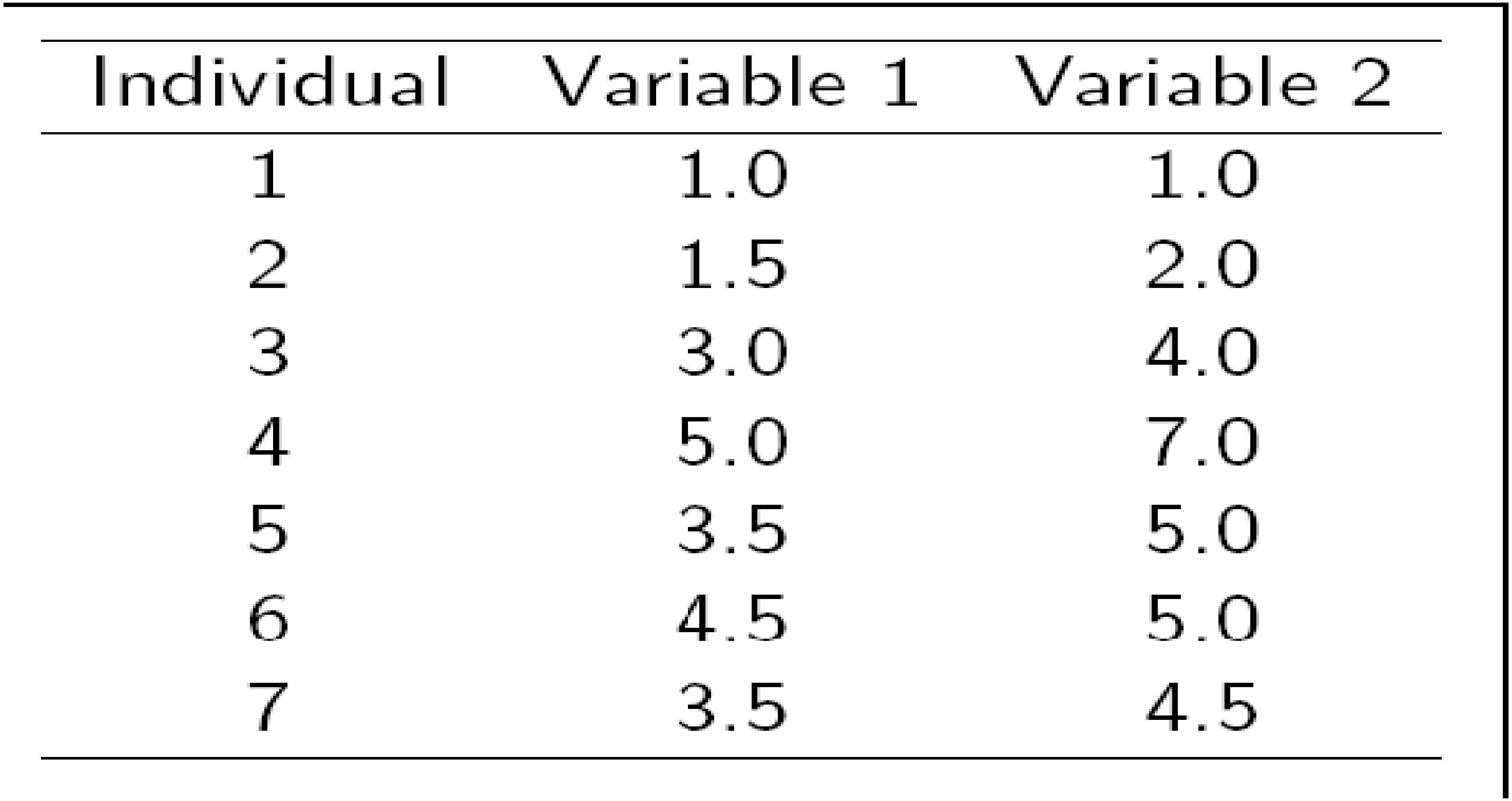
1. Take the first k training sample as single-element clusters
2. Assign each of the remaining (N-k) training sample to the cluster with the nearest centroid. After each assignment, recompute the centroid of the gaining cluster.

- **Step 3:** Take each sample in sequence and compute its distance from the centroid of each of the clusters. If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the cluster gaining the new sample and the cluster losing the sample.
- **Step 4 .** Repeat step 3 until convergence is achieved, that is until a pass through the training sample causes no new assignments.

K-means convergence (stopping) criterion

- no (or minimum) re-assignments of data points to different clusters, or
- no (or minimum) change of centroids, or
- minimum decrease in the sum of squared error

A Simple example showing the implementation of k-means algorithm (using K=2)

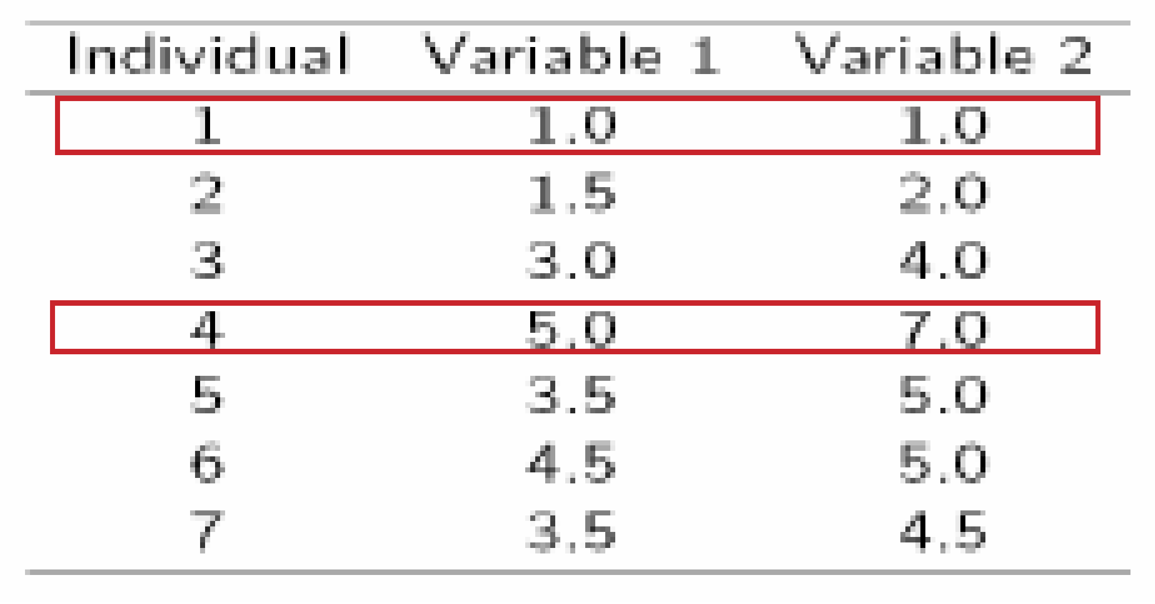


Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

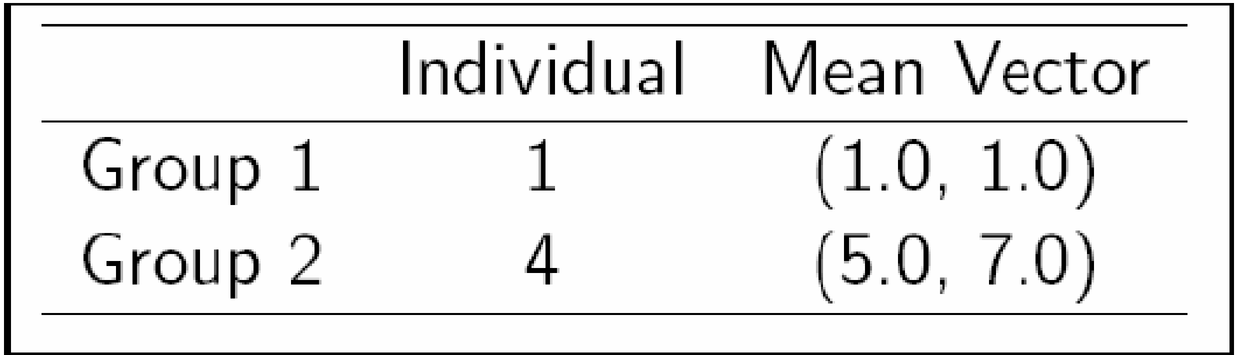
Step 1:

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

In this case the 2 centroid are: m1=(1.0,1.0) and m2=(5.0,7.0).



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:

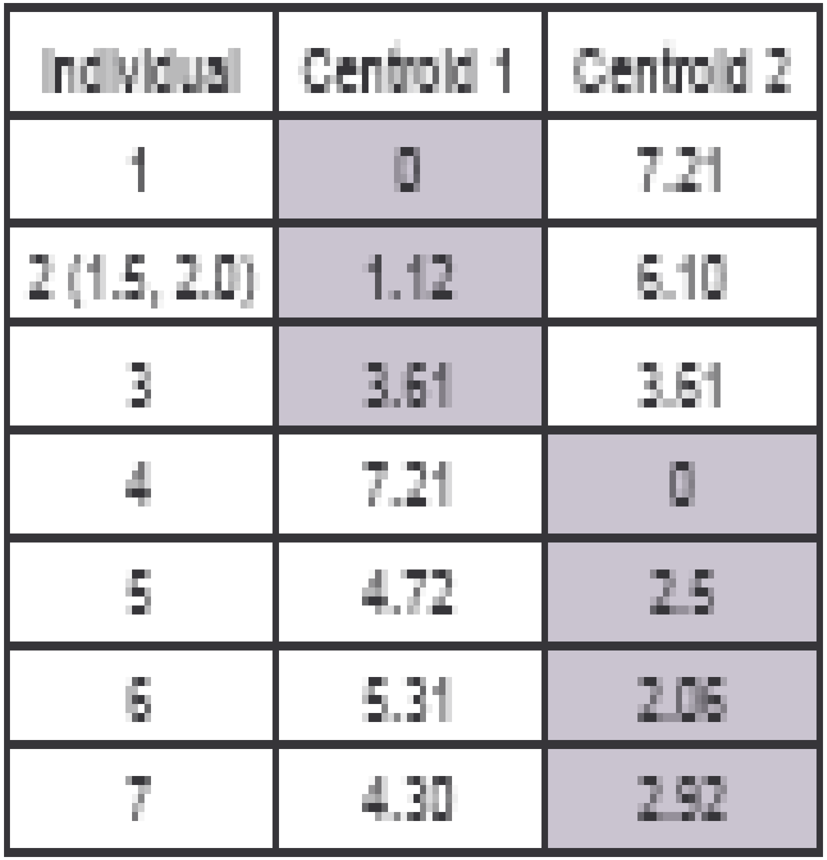
- Thus, we obtain two clusters containing:

  {1,2,3} and {4,5,6,7}.
- Their new centroids are:

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

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

$$= (4.12, 5.38)$$



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

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

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

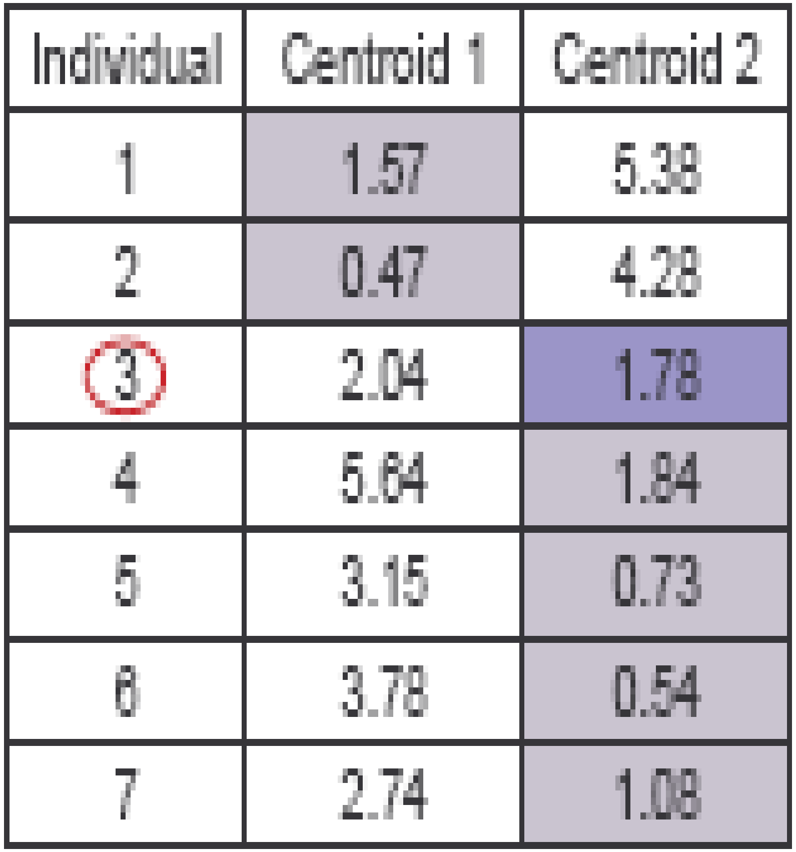
Step 3:

- 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)



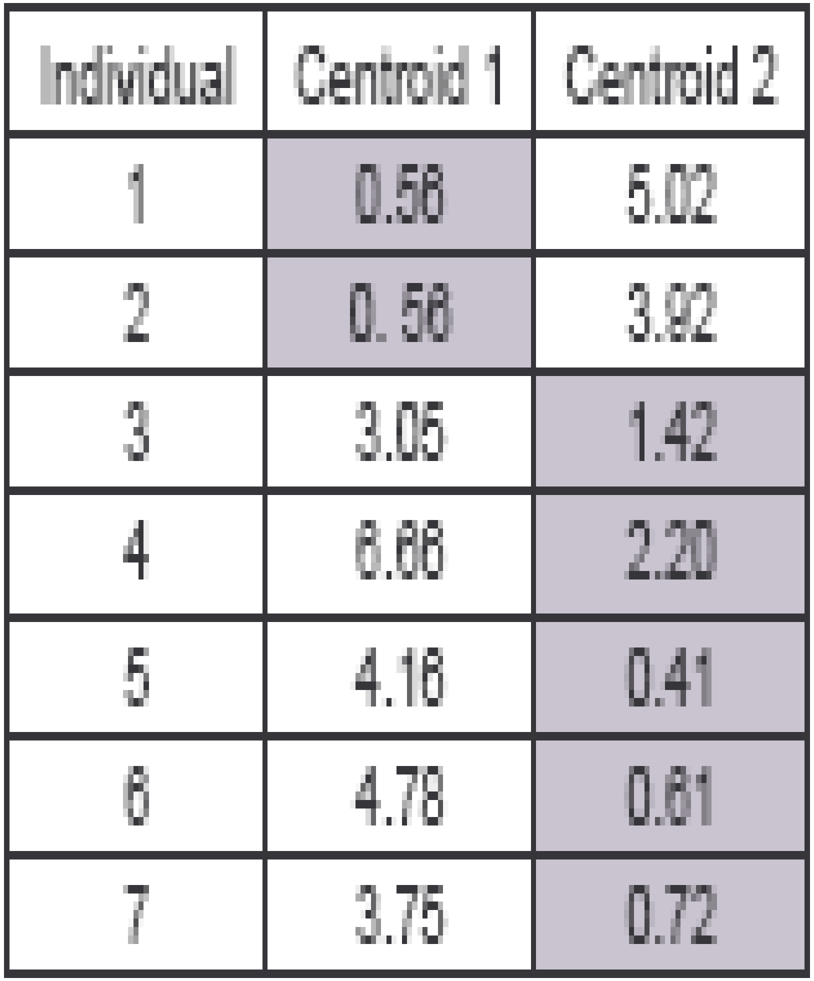
Individual	Centroid 1	Centroid 2
1	1.57	5.38
2	0.47	4.28
3	2.04	1.78
4	5.64	1.84
5	3.15	0.73
6	3.78	0.54
7	2.74	1.08

- Step 4 :

  The clusters obtained are:

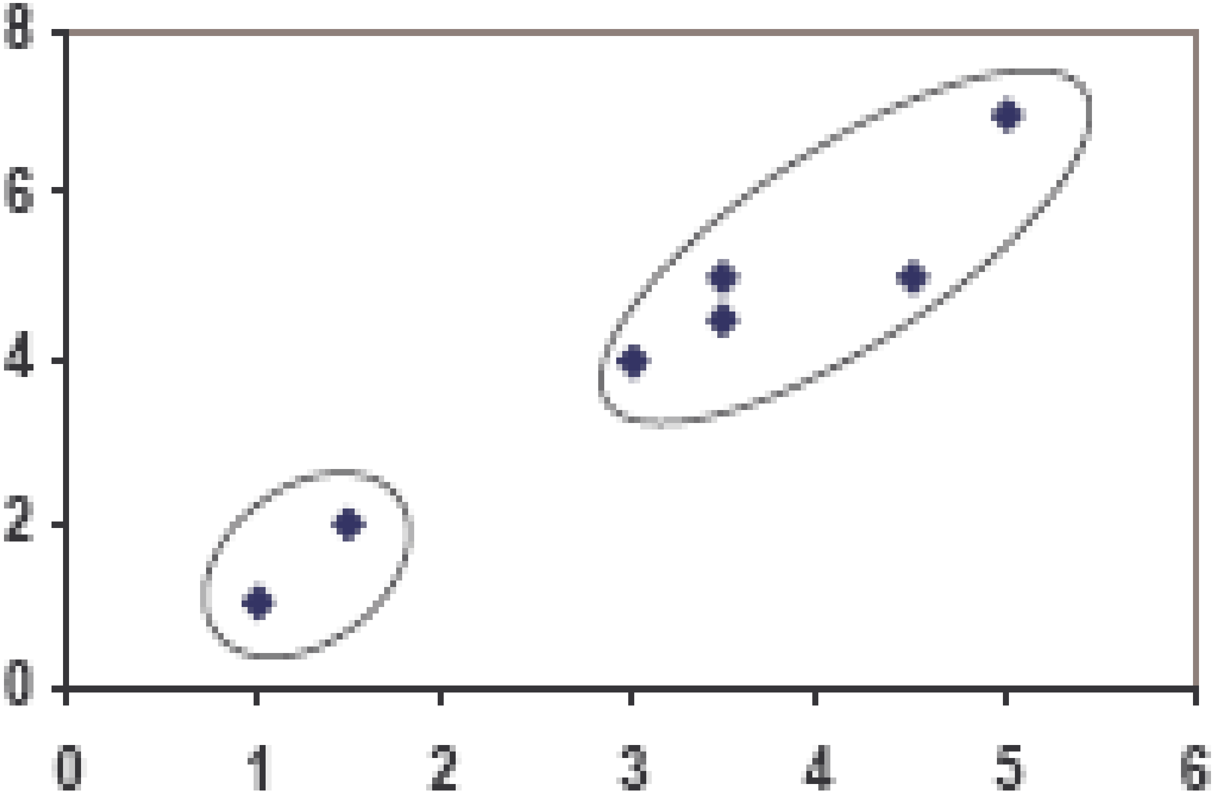
  {1,2} and {3,4,5,6,7}

- 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}.

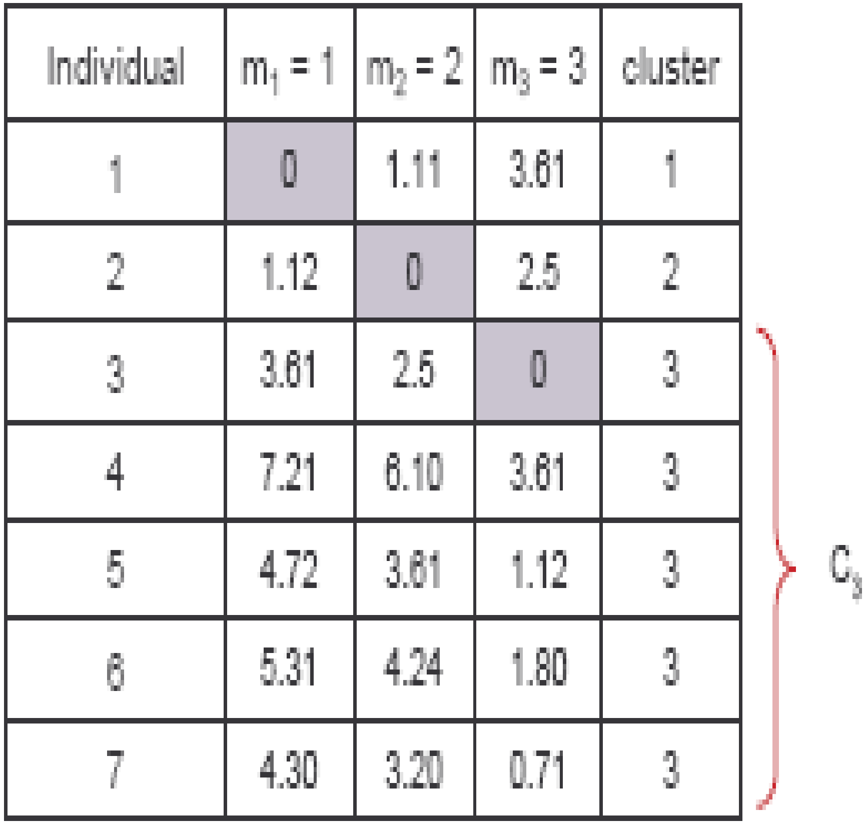


Individual	Centroid 1	Centroid 2
1	0.56	5.02
2	0. 56	3.92
3	3.05	1.42
4	6.66	2.20
5	4.16	0.41
6	4.78	0.61
7	3.75	0.72

PLOT



(with K=3)

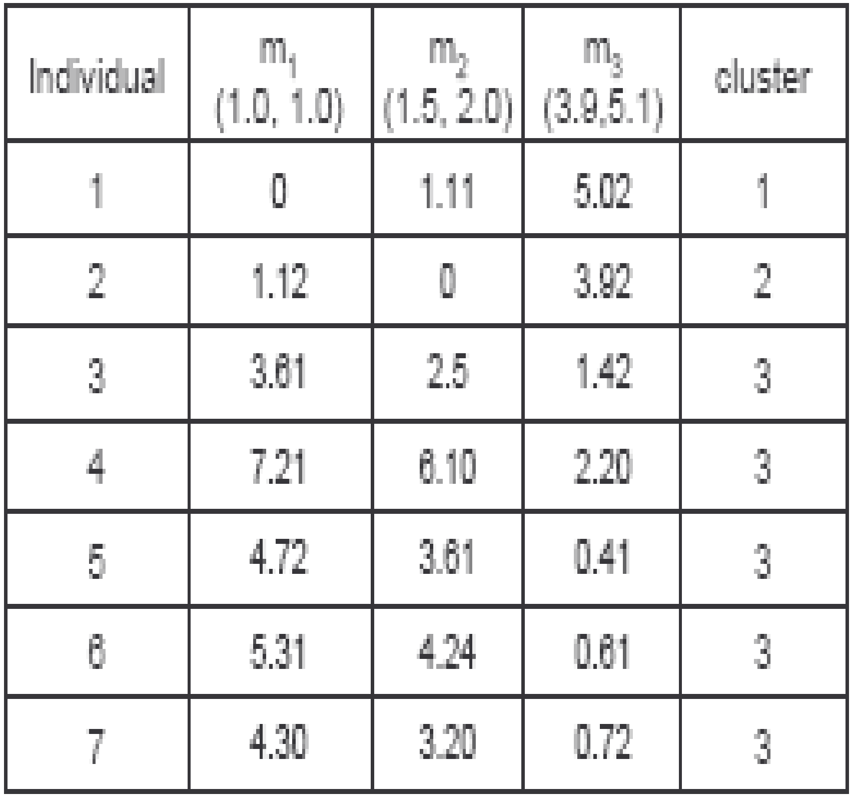


Individual	$m_1 = 1$	$m_2 = 2$	$m_3 = 3$	cluster
1	0	1.11	3.61	1
2	1.12	0	2.5	2
3	3.61	2.5	0	3
4	7.21	6.10	3.61	3
5	4.72	3.61	1.12	3
6	5.31	4.24	1.80	3
7	4.30	3.20	0.71	3

C_3

clustering with initial centroids (1, 2, 3)

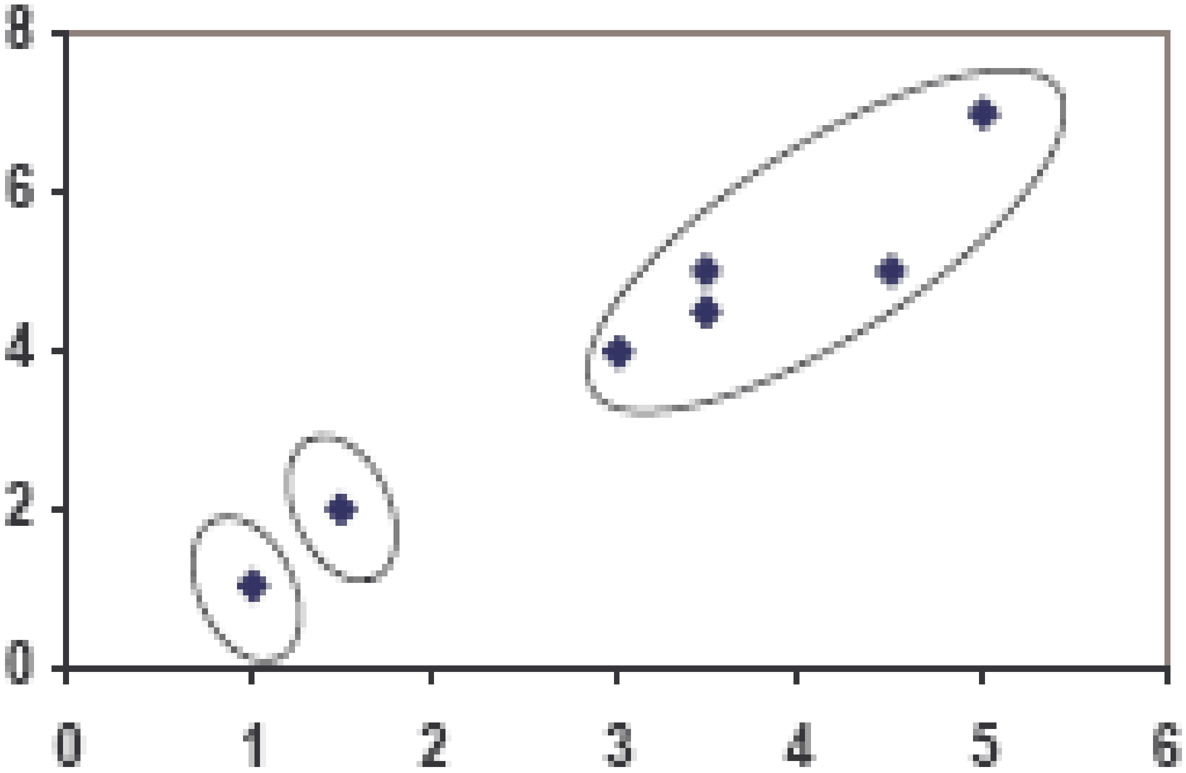
Step 1



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

Step 2

PLOT



Weaknesses of K-Mean Clustering

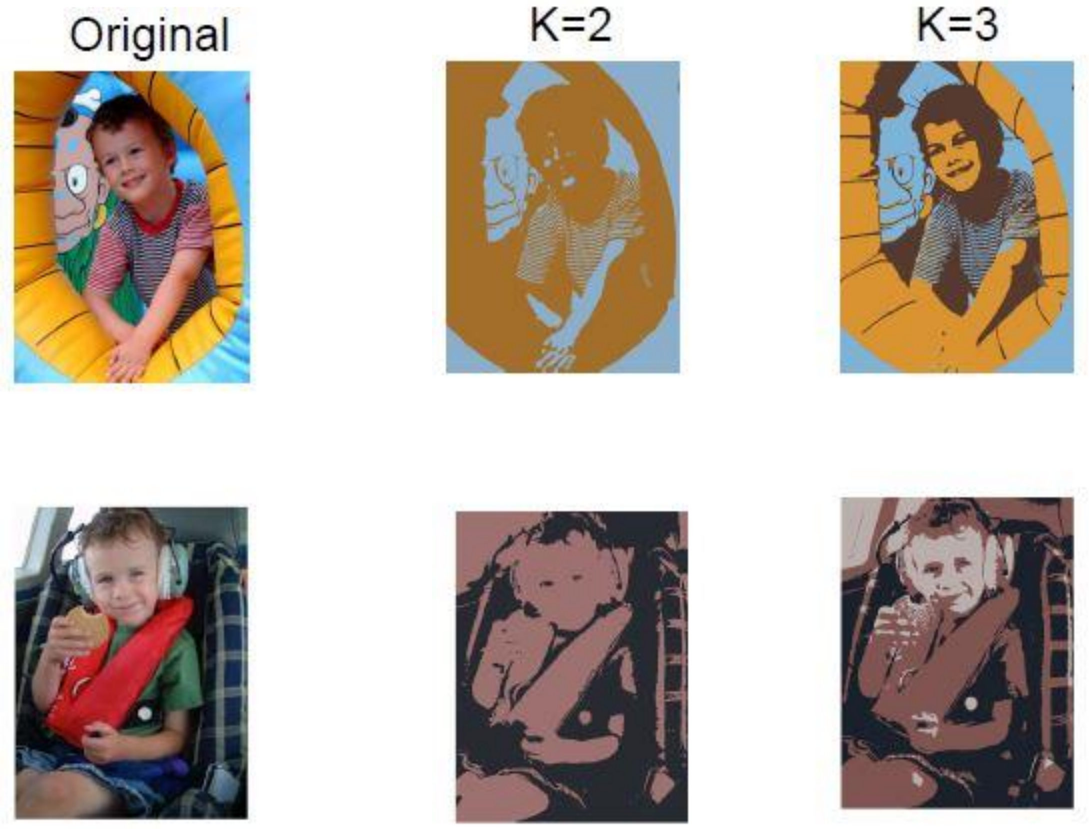
1. When the numbers of data are not so many, initial grouping will determine the cluster significantly.
2. The number of cluster, K, must be determined before hand. Its disadvantage is that it does not yield the same result with each run, since the resulting clusters depend on the initial random assignments.
3. We never know the real cluster, using the same data, because if it is inputted in a different order it may produce different cluster if the number of data is few.
4. It is sensitive to initial condition. Different initial condition may produce different result of cluster. The algorithm may be trapped in the *local optimum*.

Applications of K-Mean Clustering

- It is relatively *efficient and fast.* It computes result at **O(tkn),** where n is number of objects or points, k is number of clusters and t is number of iterations.
- k-means clustering can be applied to *machine learning or data mining*
- *Used on acoustic data in speech understanding to convert waveforms into one of k categories (known as Vector Quantization or Image Segmentation).*
- *Also used for choosing color palettes on old fashioned graphical display devices and Image Quantization.*

Applications of K-Mean Clustering

- **Image Segmentation:** Break up the images into meaningful or perceptually similar regions



CONCLUSION

- *K-means algorithm is* useful for undirected knowledge discovery and is relatively simple.
- K-means has found wide spread usage in lot of fields, ranging from unsupervised learning of neural network, Pattern recognitions, Classification analysis, Artificial intelligence, image processing, machine vision, and many others.