Unsupervised Learning

Dr. Wedad Hussein
wedad.hussein@cis.asu.edu.eg

Dr. Mahmoud Mounir
mahmoud.mounir@cis.asu.edu.eg

CLUSTERING

INTRODUCTION-

- Given the following dataset of objects

	Attribute 1	Attribute 2	Attribute 3
Objects	**X**	**Y**	**Z**
OB-1	1	4	1
OB-2	1	2	2
OB-3	1	4	2
OB-4	2	1	2
OB-5	1	1	1
OB-6	2	4	2
OB-7	1	1	2
OB-8	2	1	1

INTRODUCTION-

- Given the following dataset of objects

	Attribute 1	Attribute 2	Attribute 3	
Objects	**X**	**Y**	**Z**	**Class**
OB-1	1	4	1	**A**
OB-2	1	2	2	**B**
OB-3	1	4	2	**B**
OB-4	2	1	2	**A**
OB-5	1	1	1	**A**
OB-6	2	4	2	**B**
OB-7	1	1	2	**A**
OB-8	2	1	1	**A**

INTRODUCTION

- The types of **Learning** can broadly be classified into two types
  - **Supervised Learning**
  - **Unsupervised Learning and**
    - No variable to predict tied to the data. Instead of having an output, the data only has an input which would be multiple variables that describe the data.
    - This is where clustering comes in.

INTRODUCTION-
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.

INTRODUCTION-
What is clustering?

Characteristics of Clustering:

- Objects in the same cluster are more similar to each other than to objects in other clusters.
- Similarity is a metric that reflects the strength of relationship between two data objects.
- Clustering is mainly used for exploratory data mining. It has manifold usage in many fields such as machine learning, pattern recognition, image analysis, information retrieval, bio-informatics, data compression, and computer graphics.

Types of Clustering:

Clustering algorithms can be categorized based on their cluster model, that is based on how they form clusters or groups.

- Connectivity-based Clustering
- Centroid-based Clustering
- Distribution-based Clustering
- Density-based Clustering

Types of Clustering:

- **Connectivity-based Clustering**

- Data points that are closer in the data space are more related (similar) than to data points farther away.
- The clusters are formed by connecting data points according to their distance. At different distances, different clusters will form and can be represented using a dendrogram, which gives away why they are also commonly called "hierarchical clustering".
- These methods do not produce a unique partitioning of the dataset, rather a hierarchy from which the user still needs to choose appropriate clusters by choosing the level where they want to cluster.
- They are also not very robust towards outliers, which might show up as additional clusters or even cause other clusters to merge.

Types of Clustering:

- **Connectivity-based Clustering**
- **Hierarchical algorithms**:
  - Find successive clusters using previously established clusters.
  - Agglomerative ("bottom-up"): Begins with each element as a separate cluster and merge them into successively larger clusters.
  - Divisive ("top-down"): Begin with the whole set and proceed to divide it into successively smaller clusters.

Types of Clustering:

- **Centroid-based Clustering**
  - Clusters are represented by a central vector or a centroid.
  - This centroid might not necessarily be a member of the dataset.
  - This is an iterative clustering algorithms in which the similarity is derived by how close a data point is to the centroid of the cluster
  - It is called partitional clustering
  - Examples are K-means and fuzzy k-means clustering

Types of Clustering:

- **Distribution-based Clustering**
  - This clustering is very closely related to statistics: distributional modeling.
  - Clustering is based on the notion of how probable is it for a data point to belong to a certain distribution, such as the Gaussian distribution, for example. Data points in a cluster belong to the same distribution.
  - Example: Gaussian mixture models, using the expectation-maximization algorithm is a famous distribution based clustering method.

Types of Clustering:

- **Density-based Clustering**
  - Search the data space for areas of varied density of data points.
  - Clusters are defined as areas of higher density within the data space compared to other regions.
  - Examples: DBSCAN and OPTICS.

Common Distance measures:

Objects	X	Y	Z
OB-1	1	4	1
OB-2	1	2	2

- *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:

$$\boldsymbol{d} = \sqrt{(\boldsymbol{X_2}-\boldsymbol{X_1})^2 + (\boldsymbol{Y_2}-\boldsymbol{Y_1})^2 + (\boldsymbol{Z_2}-\boldsymbol{Z_1})^2}$$

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

$$\boldsymbol{d} = |\boldsymbol{X_2} - \boldsymbol{X_1}| + |\boldsymbol{Y_2} - \boldsymbol{Y_1}| + |\boldsymbol{Z_2} - \boldsymbol{Z_1}|$$

Common Distance measures:

3.The maximum norm is given by:

$$d = Max\ (|X_2 - X_1|, |Y_2 - Y_1|, |Z_2 - Z_1|))$$

4. Inner product space: The angle between two vectors can be used as a distance measure when clustering high dimensional data

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

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.

K-MEANS CLUSTERING

- 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

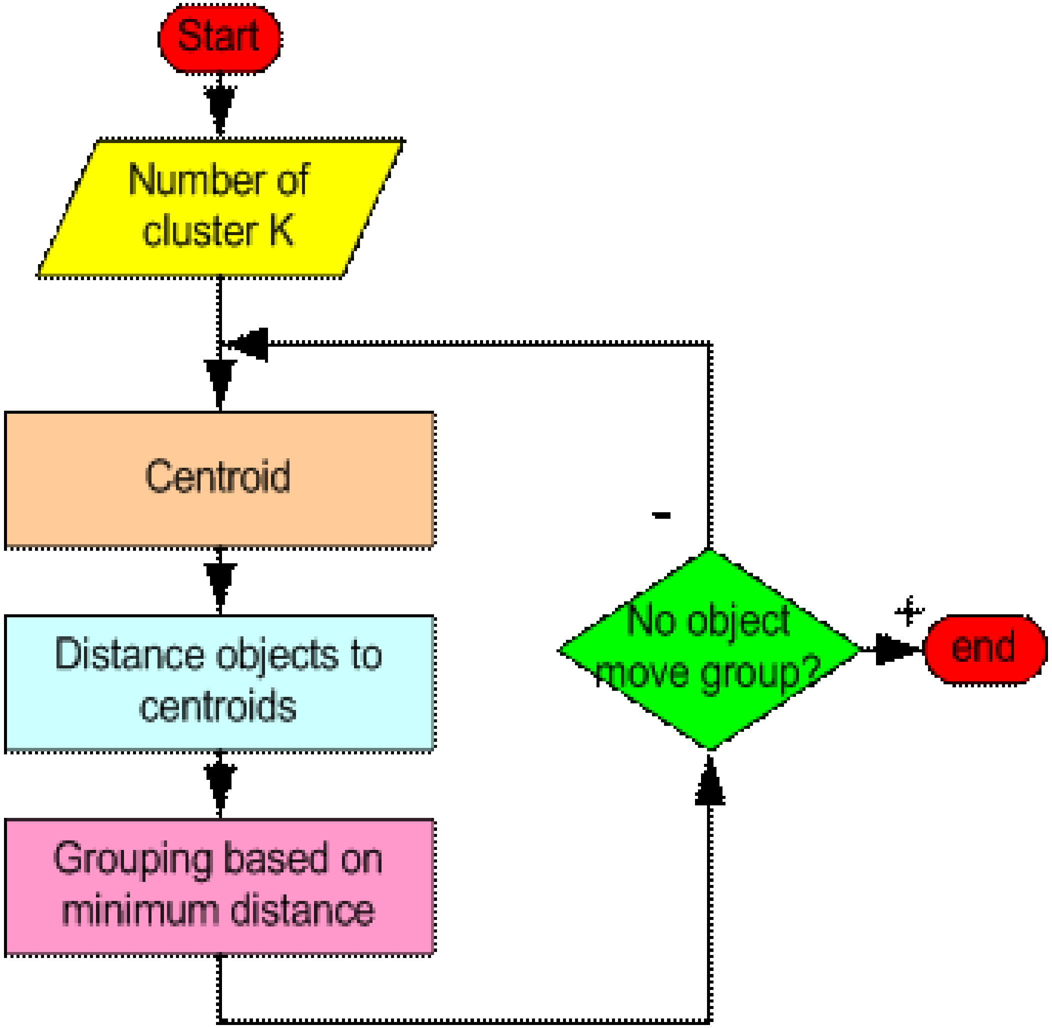
$$J = \sum_{j=1}^{K} \sum_{n \in S_j} |x_n - \mu_j|^2,$$

where x_n is a vector representing the the n^{th} data point and u_j is the geometric centroid of the data points in S_j.

K-MEANS CLUSTERING

- 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

How the K-Mean Clustering algorithm works?

- **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.

How the K-Mean Clustering algorithm works?

- **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.

A Simple example showing the k-means algorithm (using K=2) and the Manhattan distance as a similarity measure

- Given the points **A = (0, 3), B = (1, 3), C = (3, 1), D = (3, 0.5), E = (5, 0), F = (6, 0**), Starting from initial clusters Cluster1 = {A} which contains only the point **A** and Cluster2 = {D} which contains only the point **D**, apply the K-means clustering algorithm and report the final clusters. Use the distance between points which is given by d((x1, y1), (x2, y2)) = (| x1 – x2 | + | y1 – y2 |).

A Simple example showing the k-means algorithm (using K=2) and the Manhattan distance as a similarity measure

Objects	X	Y
A	0	3
B	1	3
C	3	1
D	3	0.5
E	5	0
F	6	0

A Simple example showing the k-means algorithm (using K=2) and the Manhattan distance as a similarity measure

First Iteration K = 2

Initial Centroids

C1 = A = (0,3), C2 = D = (3, 0.5)

Objects	Distance from C1= (0,3)	Distance from C2 = (3,0.5)	Cluster
A = (0, 3)			
B = (1, 3)			
C = (3, 1)			
D = (3 , 0.5)			
E = (5 , 0)			
F = (6, 0)			

A Simple example showing the k-means algorithm (using K=2) and the Manhattan distance as a similarity measure

First Iteration K = 2

Initial Centroids

C1 = A = (0,3), C2 = D = (3, 0.5)

Objects	Distance from C1= (0,3)	Distance from C2 = (3,0.5)	Cluster
A = (0, 3)	**0**	**5.5**	
B = (1, 3)	**1**	**4.5**	
C = (3, 1)	**5**	**0.5**	
D = (3 , 0.5)	**5.5**	**0**	
E = (5 , 0)	**8**	**2.5**	
F = (6, 0)	**9**	**3.5**	

A Simple example showing the k-means algorithm (using K=2) and the Manhattan distance as a similarity measure

First Iteration K = 2

Initial Centroids

C1 = A = (0,3), C2 = D = (3, 0.5)

Objects	Distance from C1= (0,3)	Distance from C2 = (3,0.5)	Cluster
A = (0, 3)	**0**	**5.5**	1
B = (1, 3)	**1**	**4.5**	1
C = (3, 1)	**5**	**0.5**	2
D = (3 , 0.5)	**5.5**	**0**	2
E = (5 , 0)	**8**	**2.5**	2
F = (6, 0)	**9**	**3.5**	2

A Simple example showing the k-means algorithm (using K=2) and the Manhattan distance as a similarity measure

First Iteration

Cluster (1)
A = (0, 3)
B = (1, 3)

Cluster (2)
C = (3, 1)
D = (3 , 0.5)
E = (5 , 0)
F = (6, 0)

New Centroids

$$\left(\frac{\sum_1^n X_i}{n} , \frac{\sum_1^n Y_i}{n}\right)$$

$$C1 = \left(\frac{0+1}{2} , \frac{3+3}{2}\right) = \left(\frac{1}{2} , \frac{6}{2}\right)$$
$$(0.5 , 3)$$

$$C2 = \left(\frac{3+3+5+6}{4} , \frac{1+0.5+0+0}{4}\right) =$$
$$\left(\frac{17}{4} , \frac{1.5}{4}\right) = (4.25 , 0.375)$$

A Simple example showing the k-means algorithm (using K=2) and the Manhattan distance as a similarity measure

Second Iteration

C1 = (0.5,3), C2 = (4.25, 0.375)

Objects	Distance from C1= (0.5,3)	Distance from C2 = (4.25,0.375)	Cluster
A = (0, 3)			
B = (1, 3)			
C = (3, 1)			
D = (3 , 0.5)			
E = (5 , 0)			
F = (6, 0)			

A Simple example showing the k-means algorithm (using K=2) and the Manhattan distance as a similarity measure

Second Iteration

C1 = (0.5,3), C2 = (4.25, 0.375)

Objects	Distance from C1= (0.5,3)	Distance from C2 = (4.25,0.375)	Cluster
A = (0, 3)	**0.5**	**6.875**	
B = (1, 3)	**0.5**	**5.875**	
C = (3, 1)	**4.5**	**1.875**	
D = (3 , 0.5)	**5**	**1.375**	
E = (5 , 0)	**7.5**	**1.125**	
F = (6, 0)	**8.5**	**2.125**	

A Simple example showing the k-means algorithm (using K=2) and the Manhattan distance as a similarity measure

Second Iteration

C1 = (0.5,3), C2 = (4.25, 0.375)

Objects	Distance from C1= (0.5,3)	Distance from C2 = (4.25,0.375)	Cluster
A = (0, 3)	**0.5**	**6.875**	**1**
B = (1, 3)	**0.5**	**5.875**	**1**
C = (3, 1)	**4.5**	**1.875**	**2**
D = (3 , 0.5)	**5**	**1.375**	**2**
E = (5 , 0)	**7.5**	**1.125**	**2**
F = (6, 0)	**8.5**	**2.125**	**2**

A Simple example showing the k-means algorithm (using K=2) and the Manhattan distance as a similarity measure

Second Iteration

Cluster (1)
A = (0, 3)
B = (1, 3)

Cluster (2)
C = (3, 1)
D = (3 , 0.5)
E = (5 , 0)
F = (6, 0)

New Centroids

$$\left(\frac{\sum_1^n X_i}{n}, \frac{\sum_1^n Y_i}{n}\right)$$

$$C1 = \left(\frac{0+1}{2}, \frac{3+3}{2}\right) = \left(\frac{1}{2}, \frac{6}{2}\right)$$
$$(0.5 , 3)$$

$$C2 = \left(\frac{3+3+5+6}{4}, \frac{1+0.5+0+0}{4}\right) =$$
$$\left(\frac{17}{4}, \frac{1.5}{4}\right) = (4.25 , 0.375)$$

No Change in Centroids

A Simple example showing the k-means algorithm (using K=2) and the Manhattan distance as a similarity measure

Final Clusters

Cluster (1)
A = (0, 3)
B = (1, 3)

Cluster (2)
C = (3, 1)
D = (3 , 0.5)
E = (5 , 0)
F = (6, 0)

Final Centroids

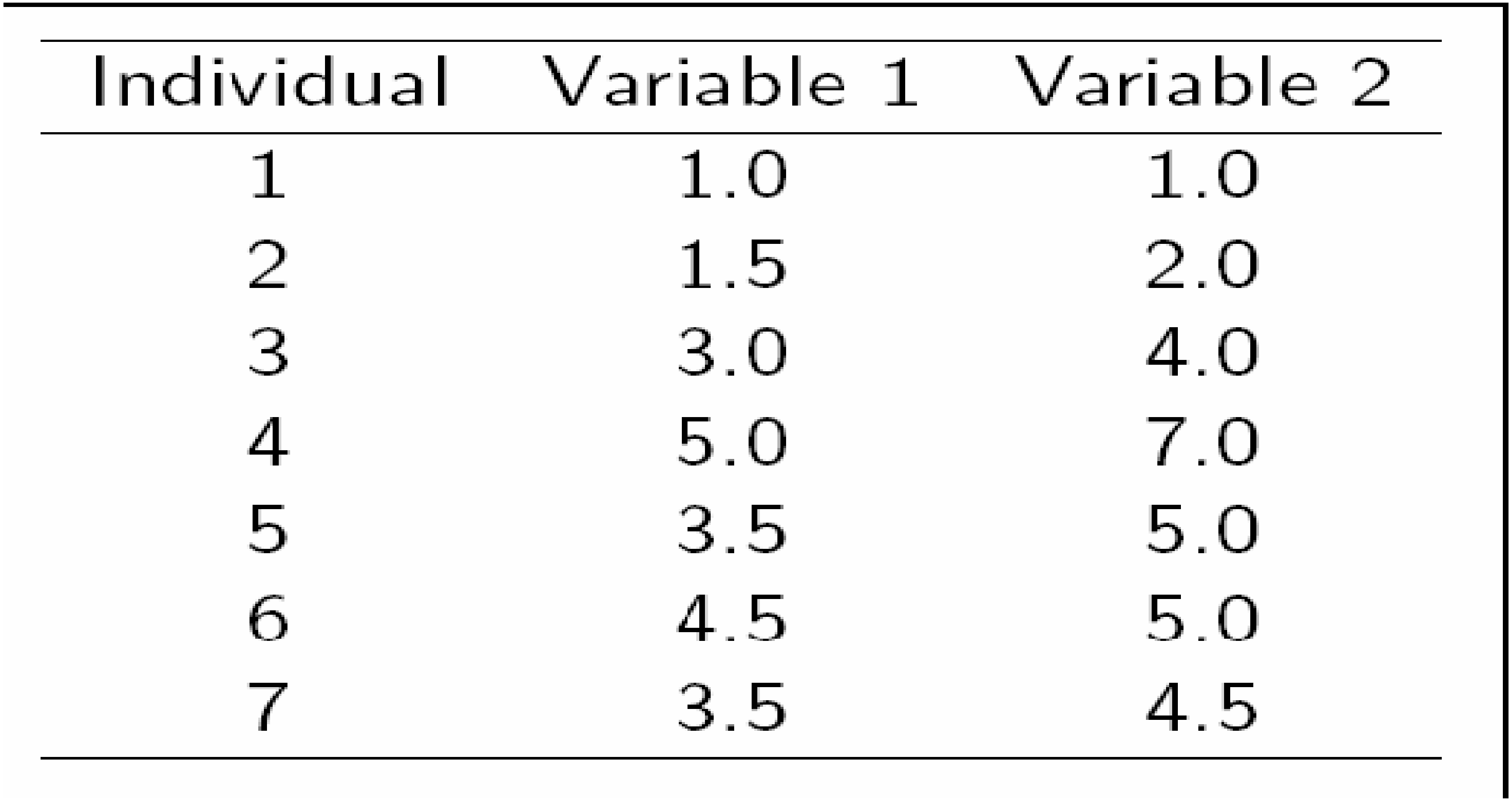
$$C1 = \left(\frac{0+1}{2}, \frac{3+3}{2}\right) = \left(\frac{1}{2}, \frac{6}{2}\right)$$

$$(0.5 , 3)$$

$$C2 = \left(\frac{3+3+5+6}{4}, \frac{1+0.5+0+0}{4}\right) =$$

$$\left(\frac{17}{4}, \frac{1.5}{4}\right) = (4.25 , 0.375)$$

A Simple example showing the k-means algorithm (using K=2) and the Euclidean distance as a similarity measure

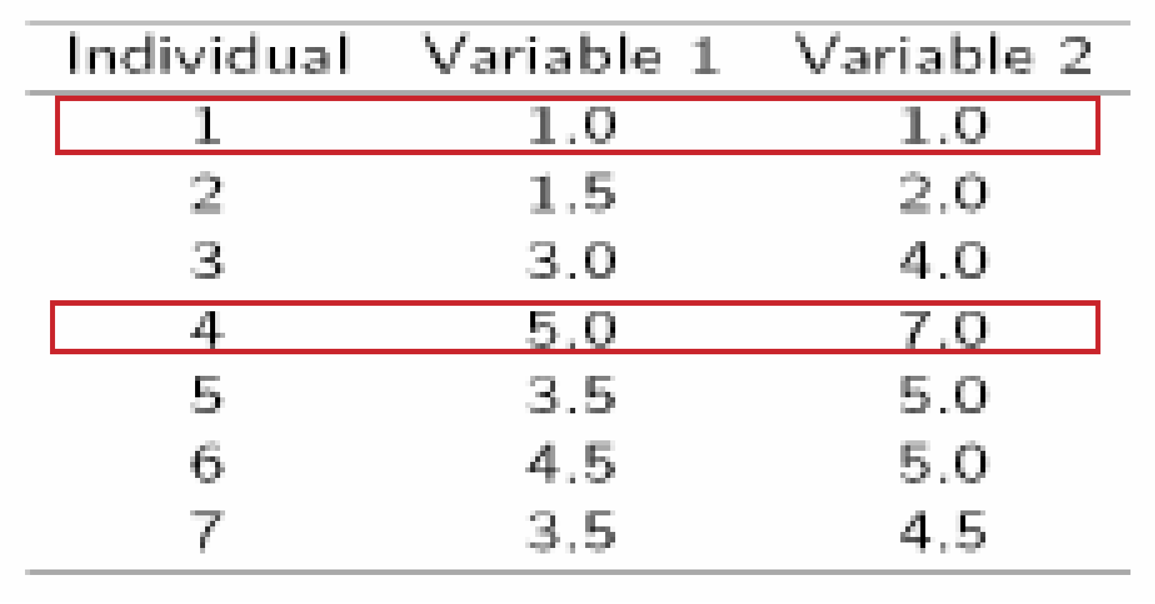


Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

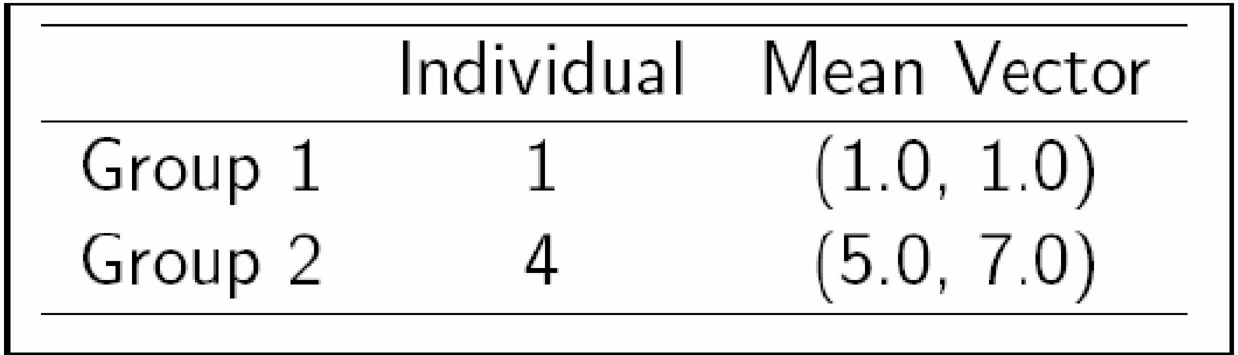
Step 1:

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

In this case the 2 centroid are: C1=(1.0,1.0) and C2=(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:

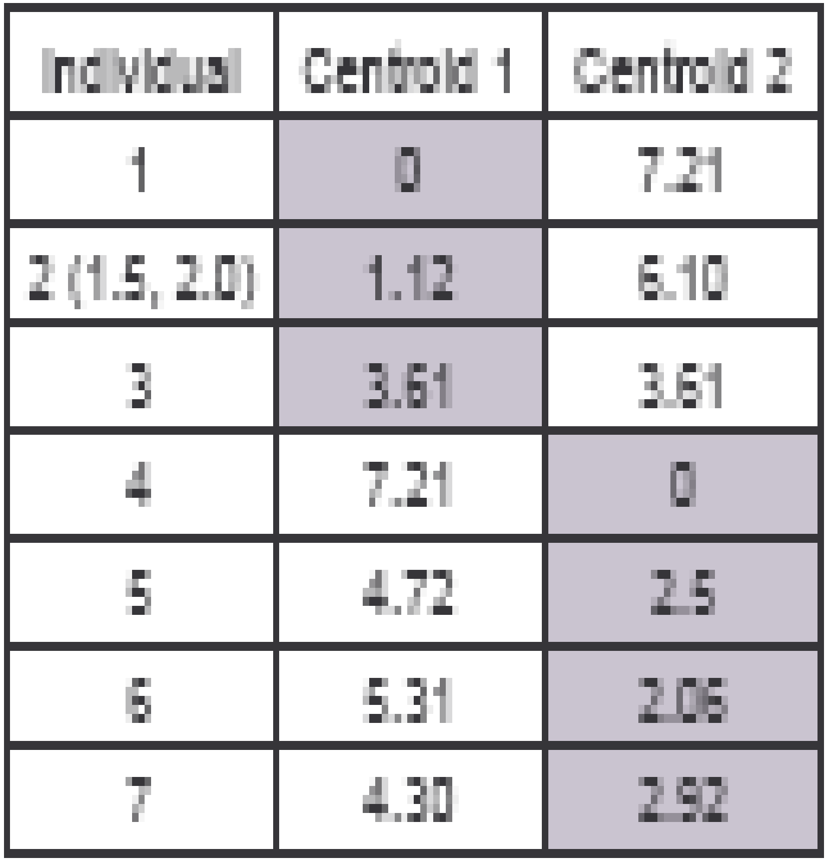
- Thus, we obtain two clusters containing:

  {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

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

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

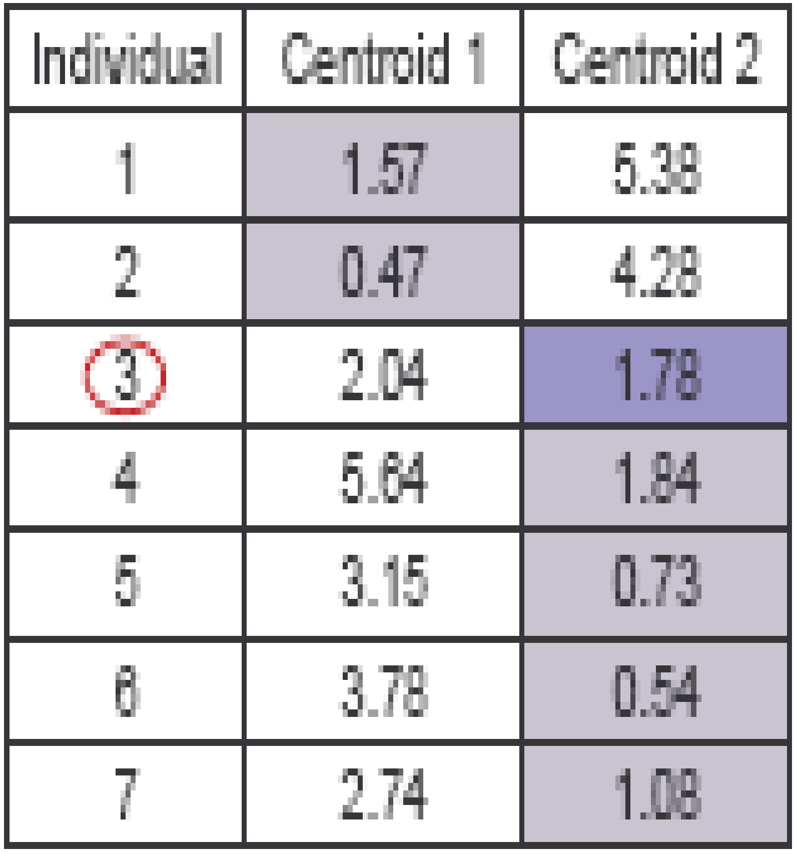
Step 3:

- 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: C1=(1.25,1.5) and C2 = (3.9,5.1)



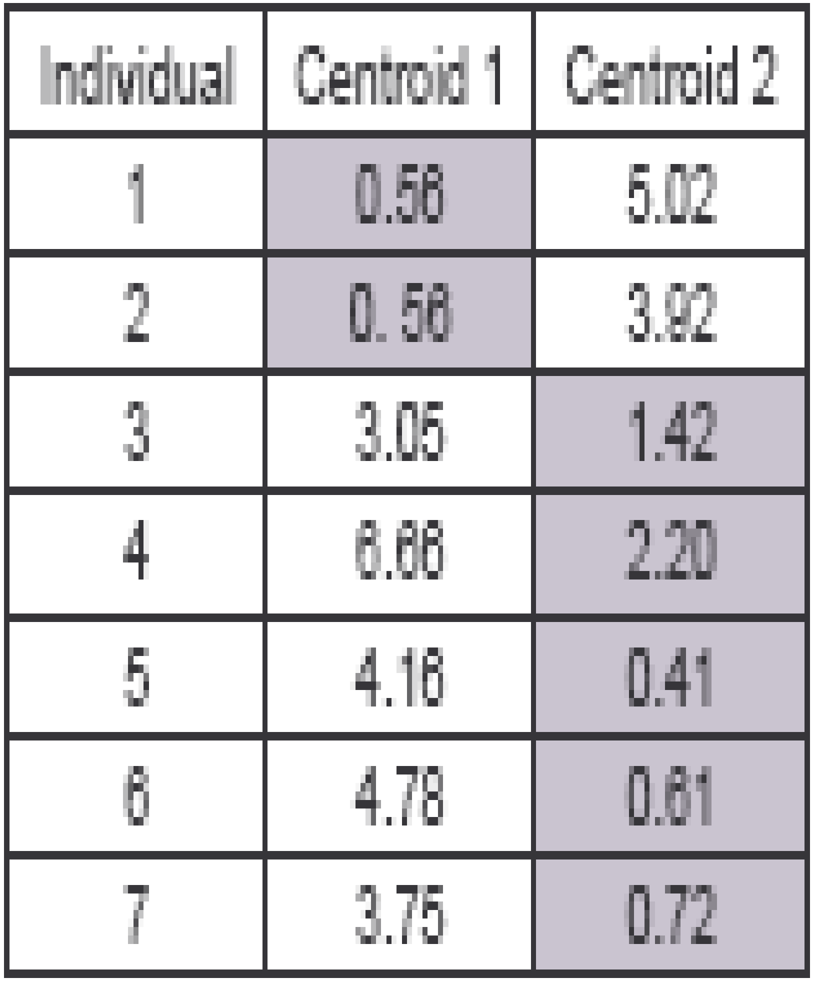
Individual	Centroid 1	Centroid 2
1	1.57	5.38
2	0.47	4.28
3	2.04	1.78
4	5.64	1.84
5	3.15	0.73
6	3.78	0.54
7	2.74	1.08

- Step 4 :

  The clusters obtained are:

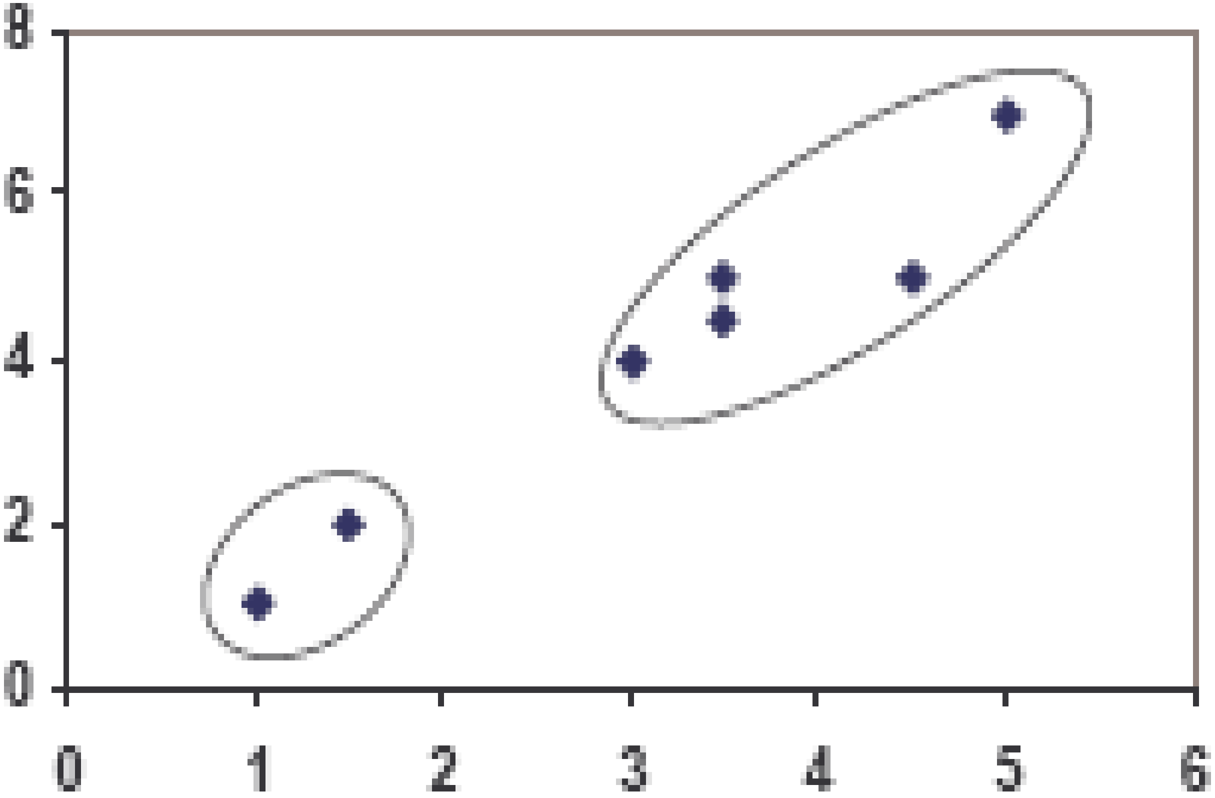
  {1,2} and {3,4,5,6,7}

- 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}.

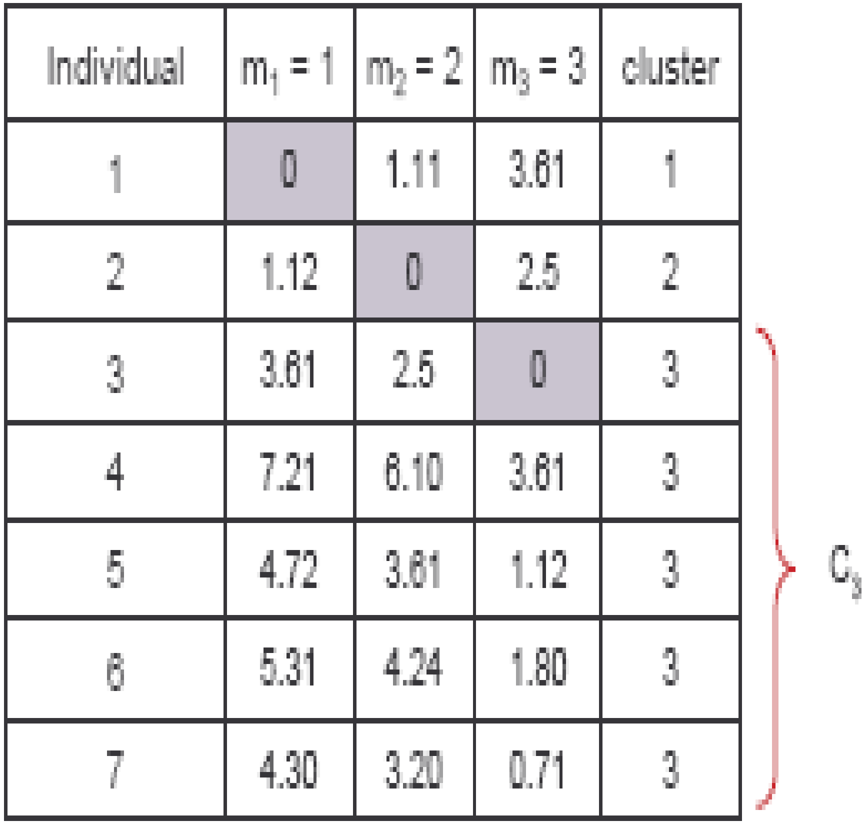


Individual	Centroid 1	Centroid 2
1	0.56	5.02
2	0. 56	3.92
3	3.05	1.42
4	6.66	2.20
5	4.16	0.41
6	4.78	0.61
7	3.75	0.72

PLOT



(with K=3)

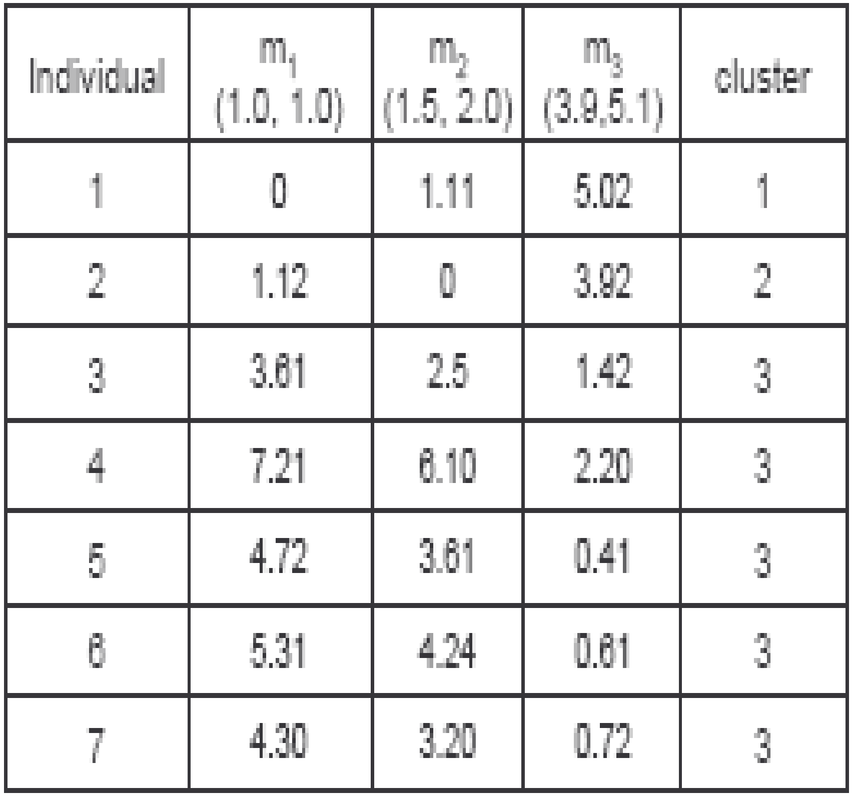


Individual	$m_1 = 1$	$m_2 = 2$	$m_3 = 3$	cluster
1	0	1.11	3.61	1
2	1.12	0	2.5	2
3	3.61	2.5	0	3
4	7.21	6.10	3.61	3
5	4.72	3.61	1.12	3
6	5.31	4.24	1.80	3
7	4.30	3.20	0.71	3

C_3

clustering with initial centroids (1, 2, 3)

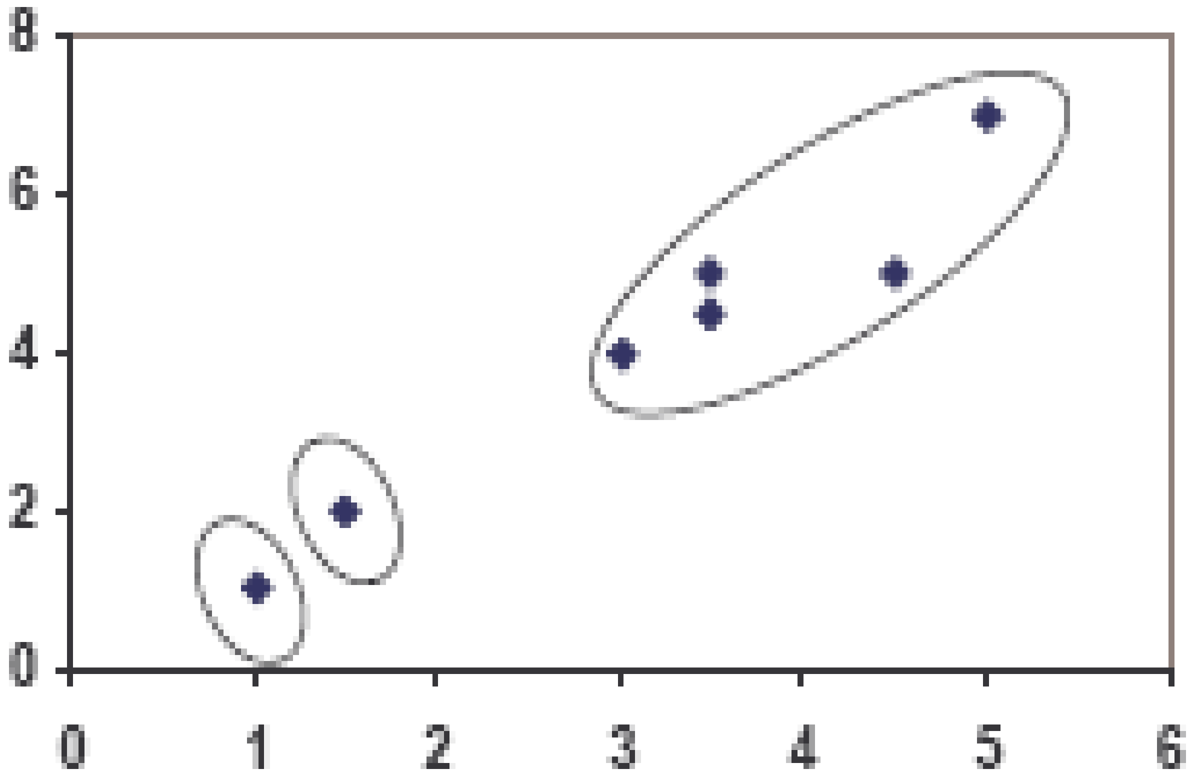
Step 1



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



A Simple example showing the k-means algorithm (using K=3) and the Manhattan distance as a similarity measure

Objects	X	Y	Z
OB-1	1	4	1
OB-2	1	2	2
OB-3	1	4	2
OB-4	2	1	2
OB-5	1	1	1
OB-6	2	4	2
OB-7	1	1	2
OB-8	2	1	1

A Simple example showing the k-means algorithm (using K=2) and the Manhattan distance as a similarity measure

First Iteration K = 2

Initial Centroids

C1 = OB-2 = (1,2,2), C2 = OB-6 = (2,4,2)

Objects	Distance from C1= (1,2,2)	Distance from C2 = (2,4,2)	Cluster
OB-1 = (1,4,1)			
OB-2 = (1,2,2)			
OB-3 = (1,4,2)			
OB-4 = (2,1,2)			
OB-5 = (1,1,1)			
OB-6 = (2,4,2)			
OB-7 = (1,1,2)			
OB-8 = (2,1,1)			

A Simple example showing the k-means algorithm (using K=2) and the Manhattan distance as a similarity measure

First Iteration K = 2

Initial Centroids

C1 = OB-2 = (1,2,2), C2 = OB-6 = (2,4,2)

Objects	Distance from C1= (1,2,2)	Distance from C2 = (2,4,2)	Cluster
OB-1 = (1,4,1)	3	2	
OB-2 = (1,2,2)	0	3	
OB-3 = (1,4,2)	2	1	
OB-4 = (2,1,2)	2	3	
OB-5 = (1,1,1)	2	5	
OB-6 = (2,4,2)	3	0	
OB-7 = (1,1,2)	1	4	
OB-8 = (2,1,1)	3	4	

A Simple example showing the k-means algorithm (using K=2) and the Manhattan distance as a similarity measure

First Iteration K = 2

Initial Centroids

C1 = OB-2 = (1,2,2), C2 = OB-6 = (2,4,2)

Objects	Distance from C1= (1,2,2)	Distance from C2 = (2,4,2)	Cluster
OB-1 = (1,4,1)	3	2	2
OB-2 = (1,2,2)	0	3	1
OB-3 = (1,4,2)	2	1	2
OB-4 = (2,1,2)	2	3	1
OB-5 = (1,1,1)	2	5	1
OB-6 = (2,4,2)	3	0	2
OB-7 = (1,1,2)	1	4	1
OB-8 = (2,1,1)	3	4	1

A Simple example showing the k-means algorithm (using K=2) and the Manhattan distance as a similarity measure

First Iteration

Cluster (1)
OB-2 = (1,2,2)
OB-4 = (2,1,2)
OB-5 = (1,1,1)
OB-7 = (1,1,2)
OB-8 = (2,1,1)

Cluster (2)
OB-1 = (1,4,1)
OB-3 = (1,4,2)
OB-6 = (2,4,2)

New Centroids

$$\left(\frac{\sum_1^n X_i}{n}, \frac{\sum_1^n Y_i}{n}, \frac{\sum_1^n Z_i}{n}\right)$$

$$C1 = \left(\frac{1+2+1+1+2}{5}, \frac{2+1+1+1+1}{5}, \frac{2+2+1+2+1}{5}\right) = (1.4 , 1.2, 1.6)$$

$$C2 = \left(\frac{1+1+2}{3}, \frac{4+4+4}{3}, \frac{1+2+2}{3}\right) = (1.33, 4 , 1.66)$$

A Simple example showing the k-means algorithm (using K=2) and the Manhattan distance as a similarity measure

Second Iteration

C1 = (1.4,1.2,1.6), C2 = (1.33,4,1.66)

Objects	Distance from C1= (1.4,1.2,1.6)	Distance from C2 = (1.33,4,1.66)	Cluster
OB-1 = (1,4,1)	**3.8**	**1**	
OB-2 = (1,2,2)	**1.6**	**2.66**	
OB-3 = (1,4,2)	**3.6**	**0.66**	
OB-4 = (2,1,2)	**1.2**	**4**	
OB-5 = (1,1,1)	**1.2**	**4**	
OB-6 = (2,4,2)	**3.8**	**1**	
OB-7 = (1,1,2)	**1**	**3.66**	
OB-8 = (2,1,1)	**1.4**	**4.33**	

A Simple example showing the k-means algorithm (using K=2) and the Manhattan distance as a similarity measure

Second Iteration

C1 = (1.4,1.2,1.6), C2 = (1.33,4,1.66)

Objects	Distance from C1= (1.4,1.2,1.6)	Distance from C2 = (1.33,4,1.66)	Cluster
OB-1 = (1,4,1)	3.8	1	2
OB-2 = (1,2,2)	1.6	2.66	1
OB-3 = (1,4,2)	3.6	0.66	2
OB-4 = (2,1,2)	1.2	4	1
OB-5 = (1,1,1)	1.2	4	1
OB-6 = (2,4,2)	3.8	1	2
OB-7 = (1,1,2)	1	3.66	1
OB-8 = (2,1,1)	1.4	4.33	1

A Simple example showing the k-means algorithm (using K=2) and the Manhattan distance as a similarity measure

Second Iteration

Cluster (1)
OB-2 = (1,2,2)
OB-4 = (2,1,2)
OB-5 = (1,1,1)
OB-7 = (1,1,2)
OB-8 = (2,1,1)

Cluster (2)
OB-1 = (1,4,1)
OB-3 = (1,4,2)
OB-6 = (2,4,2)

Centroids

$$\left(\frac{\sum_1^n X_i}{n}, \frac{\sum_1^n Y_i}{n}, \frac{\sum_1^n Z_i}{n}\right)$$

C1 = $\left(\frac{1+2+1+1+2}{5}, \frac{2+1+1+1+1}{5}, \frac{2+2+1+2+1}{5}\right)$
= **(1.4 , 1.2, 1.6)**

C2 = $\left(\frac{1+1+2}{3}, \frac{4+4+4}{3}, \frac{1+2+2}{3}\right)$
= **(1.33, 4 , 1.66)**

No Change in Centroids

A Simple example showing the k-means algorithm (using K=2) and the Manhattan distance as a similarity measure

Final Clusters

Cluster (1)
OB-2 = (1,2,2)
OB-4 = (2,1,2)
OB-5 = (1,1,1)
OB-7 = (1,1,2)
OB-8 = (2,1,1)

Cluster (2)
OB-1 = (1,4,1)
OB-3 = (1,4,2)
OB-6 = (2,4,2)

Final Centroids

$$C1 = \left(\frac{1+2+1+1+2}{5}, \frac{2+1+1+1+1}{5}, \frac{2+2+1+2+1}{5}\right) = (1.4 , 1.2, 1.6)$$

$$C2 = \left(\frac{1+1+2}{3}, \frac{4+4+4}{3}, \frac{1+2+2}{3}\right) = (1.33, 4 , 1.66)$$

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.*

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.

References

- Tutorial - Tutorial with introduction of Clustering Algorithms (k-means, fuzzy-c-means, hierarchical, mixture of gaussians) + some interactive demos (java applets).
- Digital Image Processing and Analysis-byB.Chanda and D.Dutta Majumdar.
- H. Zha, C. Ding, M. Gu, X. He and H.D. Simon. "Spectral Relaxation for K-means Clustering", Neural Information Processing Systems vol.14 (NIPS 2001). pp. 1057-1064, Vancouver, Canada. Dec. 2001.
- J. A. Hartigan (1975) "Clustering Algorithms". Wiley.
- J. A. Hartigan and M. A. Wong (1979) "A K-Means Clustering Algorithm", Applied Statistics, Vol. 28, No. 1, p100-108.
- D. Arthur, S. Vassilvitskii (2006): "How Slow is the k-means Method?,"
- D. Arthur, S. Vassilvitskii: "k-means++ The Advantages of Careful Seeding" 2007 Symposium on Discrete Algorithms (SODA).
- www.wikipedia.com

Thank You