

Partitioning Methods ***(k -Medoids)***

- The simplest and most fundamental version of cluster analysis is ***partitioning***, which organizes the objects of a set into several exclusive groups or clusters.
- Formally, given a data set, D , of n objects, and k , the number of clusters to form, a partitioning algorithm organizes the objects into k partitions ($k \leq n$), where each partition represents a cluster.
- The *most commonly* used partitioning methods are ***k-means*** and ***k-medoids***.

***k-medoid* Clustering Algorithm**

- ***K-Medoids*** is an unsupervised clustering algorithm using the ***partition method*** of clustering. It is an improvised version of the K-Means clustering algorithm especially used to handle outlier data. It requires unlabeled data to work with.
- The partitioning will be carried on such that:
 - Each cluster must have ***at least one*** object
 - An object must belong to ***only one*** cluster
- In the ***K-Medoids*** algorithm, each data point is called a ***medoid***. The ***medoids*** serve as cluster centers. The medoid is a point such that its sum of the distance from all other points in the same cluster is minimum.

K-Medoids Algorithm:

Step-1: Choose k number of random points from the data and consider these k points are the *initial medoids*.

Step-2: For all the remaining data points, *calculate the distance* from *each medoid* and assign it to the cluster with the nearest medoid.

Step-3: Calculate the total cost (Sum of all the distances from all the data points to the medoids).

Step-4: Select k random points as the new medoids and swap the previous medoids with them. Repeat 2 and 3 steps.

K-Medoids Algorithm:

Step-5: If the total cost of the new medoid is *less than* that of the previous medoid, make the new medoid permanent and repeat step 4.

Step-6: If the total cost of the new medoid is *greater than* the cost of the previous medoid, undo the swap and repeat step 4.

Step-7: The Repetitions have to continue until **no change** is encountered with new medoids to classify data points.

How to Compute the Distance?

The k-means algorithm enables us to clusters or groups the data points by ***calculating the distance*** from centroids to existing data points.

- To calculate the distance we use any of the following are ***Distance Measures***.
 - ***Euclidean distance***
 - ***Manhattan distance***

Euclidean distance:

- Euclidean distance can be simply explained as the ***ordinary distance*** between two points.
- Mathematically it computes the ***root of squared differences*** between the coordinates between two objects.
- The ***Euclidean distance*** between two points or tuples, say, **$\mathbf{X} = (x_1, x_2, \dots, x_n)$** and **$\mathbf{Y} = (y_1, y_2, \dots, y_n)$** is

$$dist(X, Y) = dist(Y, X) = \sqrt{\sum_{i=1}^n (X_i - Y_i)^2}$$

Manhattan distance:

- Manhattan Distance metric is generally used when we are interested in the total distance travelled by the object instead of the displacement.
- This distance is calculated by **summing the absolute difference** between the coordinates of the points in n-dimensions.
- The **Manhattan distance** between two points or tuples, say, $\mathbf{X} = (x_1, x_2, \dots, x_n)$ and $\mathbf{Y} = (y_1, y_2, \dots, y_n)$ is

$$\text{dist}(\mathbf{X}, \mathbf{Y}) = \text{dist}(\mathbf{Y}, \mathbf{X}) = \sum_{i=1}^n |X_i - Y_i|$$

Example: *K-mediod algorithm*

Suppose we want to group the visitors to a website using just their *age* as follows:

16, 17, 21, 23, 29, 36, 41, 43

Solution:

Select random **K=2** points or *medoids*. i.e.

medoid (m1) = 17

medoid (m2) = 23

- Now, calculate the distance from each data point to each ***medoid***.

Let us assume,

- ***Distance 1*** is the distances from each data point (X_i) to ***medoid 1***(m_1)

$$\text{Distance 1} = |X_i - m_1|$$

- ***Distance 2*** is the distances from each data point (X_i) to ***medoid 2***(m_2)

$$\text{Distance 2} = |X_i - m_2|$$

Iteration 1:

The Initial ***medoids*** are

$$\text{medoid1 (m1)} = \mathbf{17}$$

$$\text{medoid2 (m2)} = \mathbf{23}$$

x_i	m_1	m_2	Distance 1	Distance 2	Nearest Cluster	Cost of Cluster	Total Cost
16	17	23	1	7	1	1	60
17	17	23	0	6	1		
21	17	23	4	2	2	59	
23	17	23	6	0	2		
29	17	23	12	6	2		
36	17	23	19	13	2		
41	17	23	24	18	2		
43	17	23	26	20	2		

- Calculation of ***Total Cost***:

$$\text{Cost of cluster1} = (1+0) = 1$$

$$\text{Cost of cluster2} = (2+0+6+13+18+20) = 59$$

$$\text{Total Cost} = 1+59 = \mathbf{60}$$

Iteration-2:

Select random medoids

$$\mathbf{medoid1} \text{ (m1)} = \mathbf{21}$$

$$\mathbf{medoid2} \text{ (m2)} = \mathbf{36}$$

x_i	m_1	m_2	Distance 1	Distance 2	Nearest Cluster	Cost of Cluster	Total Cost
16	21	36	5	20	1	11	30
17	21	36	4	19	1		
21	21	36	0	15	2		
23	21	36	2	13	2		
29	21	36	8	7	2	19	
36	21	36	15	0	2		
41	21	36	20	5	2		
43	21	36	22	7	2		

- Calculation of ***Total Cost***:
Cost of cluster1 = $(5+4+0+2) = 11$
Cost of cluster2 = $(7+0+5+7) = 19$
Total Cost = $11+19 = 30$
- The ***total cost*** of the *new medoids* is **less than** that of the *previous medoids*, make the new medoids ***permanent***.

Iteration-3:

Select random medoids

$$\textbf{medoid1 (m1) = 21}$$

$$\textbf{medoid2 (m2) = 41}$$

x_i	m_1	m_2	Distance 1	Distance 2	Nearest Cluster	Cost of Cluster	Total Cost
16	21	41	5	25	1	19	26
17	21	41	4	24	1		
21	21	41	0	20	2		
23	21	41	2	18	2		
29	21	41	8	12	2		
36	21	41	15	5	2	7	
41	21	41	20	0	2		
43	21	41	22	2	2		

- Calculation of ***Total Cost***:
Cost of cluster1 = $(5+4+0+2+8) = 19$
Cost of cluster2 = $(5+0+2) = 7$
Total Cost = $19+7 = 26$
- The ***total cost*** of the *new medoids* is **less than** that of the *previous medoids*, make the new medoids ***permanent***.

Iteration-4:

Select random medoids

$$\mathbf{medoid1} (m1) = 23$$

$$\mathbf{medoid2} (m2) = 41$$

x_i	m_1	m_2	Distance 1	Distance 2	Nearest Cluster	Cost of Cluster	Total Cost
16	23	41	7	25	1	21	28
17	23	41	6	24	1		
21	23	41	2	20	2		
23	23	41	0	18	2		
29	23	41	6	12	2		
36	23	41	13	5	2	7	
41	23	41	18	0	2		
43	23	41	20	2	2		

- Calculation of ***Total Cost***:

Cost of cluster1 = $(7+6+2+0+6) = 21$

Cost of cluster2 = $(5+0+2) = 7$

Total Cost = $21 + 7 = \mathbf{28}$

- The total cost of the new medoid is ***greater than*** the cost of the previous medoid, ***undo the swap***.
- Repeat the same process until ***no change***

- After completion of the above process, the ***finalized / best medoids*** are:

$$\text{medoid1 (m1)} = 21$$

$$\text{medoid2 (m2)} = 41$$

Hence, the ***resultant clusters*** are:

$$\text{Cluster 1} = (16, 17, 21, 23, 29)$$

$$\text{Cluster 2} = (36, 41, 43)$$