**K-Medoids or Partitioning Around Medoid(PAM) Clustering**

**Medoid:** A Medoid is a point in the cluster from which the sum of distances to other data points is minimal.

**(or)**

A Medoid is a point in the cluster from which dissimilarities with all the other points in the clusters are minimal.

Instead of centroids as reference points in K-Means algorithms, the K-Medoids algorithm takes a Medoid as a reference point.

**Given the value of k and unlabelled data:**

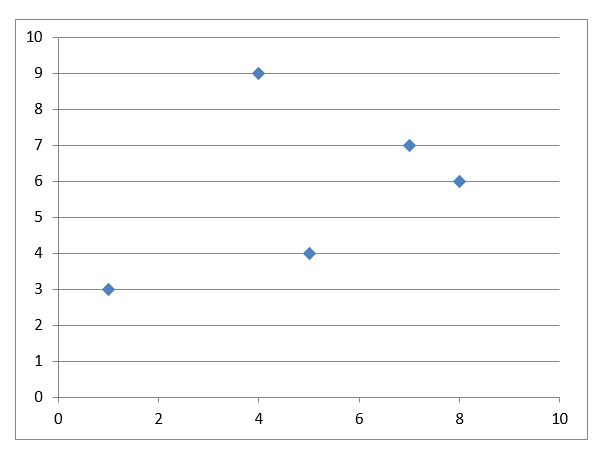
1. Choose k number of random points from the data and assign these k points to k number of clusters. These are the initial medoids.
2. For all the remaining data points, calculate the distance from each medoid and assign it to the cluster with the nearest medoid.
3. Calculate the total cost (Sum of all the distances from all the data points to the medoids)
4. Select a random point as the new medoid and swap it with the previous medoid. Repeat 2 and 3 steps.
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.
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.
7. The Repetitions have to continue until no change is encountered with new medoids to classify data points.

**Here is an example to make the theory clear:**

**Data set:**

|  |  |  |
| --- | --- | --- |
|  | x | y |
| 0 | 5 | 4 |
| 1 | 7 | 7 |
| 2 | 1 | 3 |
| 3 | 8 | 6 |
| 4 | 4 | 9 |

**Scatter plot:**



If k is given as 2, we need to break down the data points into 2 clusters.

1. **Initial medoids: M1(1, 3) and M2(4, 9)**
2. Calculation of distances

**Manhattan Distance: |x1 - x2| + |y1 - y2|**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **x** | **y** | **From M1(1, 3)** | **From M2(4, 9)** |
| 0 | 5 | 4 | 5 | 6 |
| 1 | 7 | 7 | 10 | 5 |
| 2 | 1 | 3 | - | - |
| 3 | 8 | 6 | 10 | 7 |
| 4 | 4 | 9 | - | - |

**Cluster 1:** 0

**Cluster 2:** 1, 3

1. Calculation of total cost:   
   (5) + (5 + 7) = 17
2. Random medoid: (5, 4)

**M1(5, 4) and M2(4, 9):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **x** | **y** | **From M1(5, 4)** | **From M2(4, 9)** |
| 0 | 5 | 4 | - | - |
| 1 | 7 | 7 | 5 | 5 |
| 2 | 1 | 3 | 5 | 9 |
| 3 | 8 | 6 | 5 | 7 |
| 4 | 4 | 9 | - | - |

**Cluster 1:** 2, 3

**Cluster 2:** 1

1. Calculation of total cost:   
   (5 + 5) + 5 = 15   
   Less than the previous cost   
   New medoid: (5, 4).
2. Random medoid: (7, 7)

**M1(5, 4) and M2(7, 7)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **x** | **y** | **From M1(5, 4)** | **From M2(7, 7)** |
| 0 | 5 | 4 | - | - |
| 1 | 7 | 7 | - | - |
| 2 | 1 | 3 | 5 | 10 |
| 3 | 8 | 6 | 5 | 2 |
| 4 | 4 | 9 | 6 | 5 |

**Cluster 1:** 2

**Cluster 2:** 3, 4

1. Calculation of total cost:   
   (5) + (2 + 5) = 12   
   Less than the previous cost   
   New medoid: (7, 7).
2. Random medoid: (8, 6)

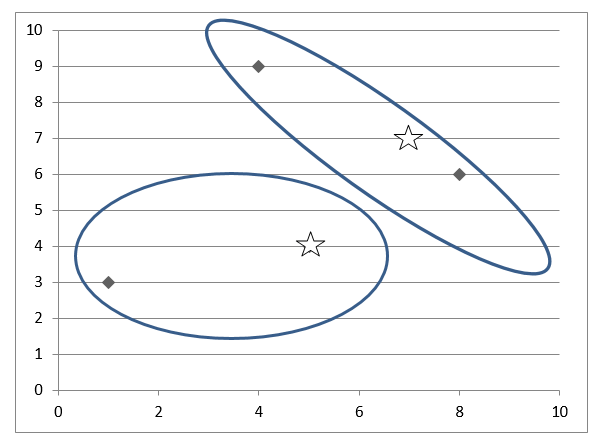
**M1(7, 7) and M2(8, 6)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **x** | **y** | **From M1(7, 7)** | **From M2(8, 6)** |
| 0 | 5 | 4 | 5 | 5 |
| 1 | 7 | 7 | - | - |
| 2 | 1 | 3 | 10 | 10 |
| 3 | 8 | 6 | - | - |
| 4 | 4 | 9 | 5 | 7 |

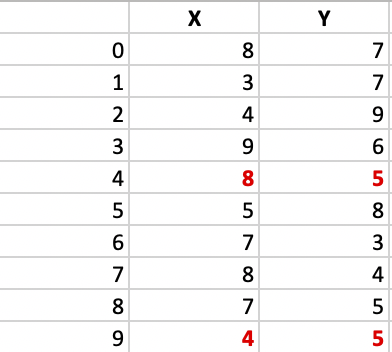
**Cluster 1:** 4

**Cluster 2:** 0, 2

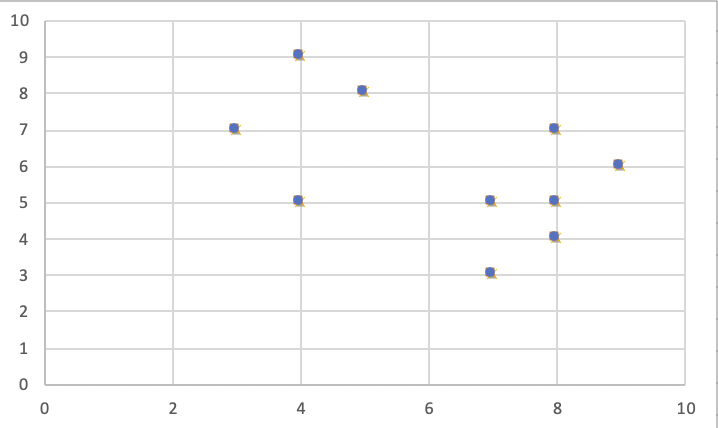
1. Calculation of total cost:   
   (5) + (5 + 10) = 20   
   Greater than the previous cost   
   **UNDO**   
   Hence, the final medoids: **M1(5, 4) and M2(7, 7)**   
   **Cluster 1:** 2   
   **Cluster 2:** 3, 4   
   Total cost: 12   
   **Clusters:**



**Example:**

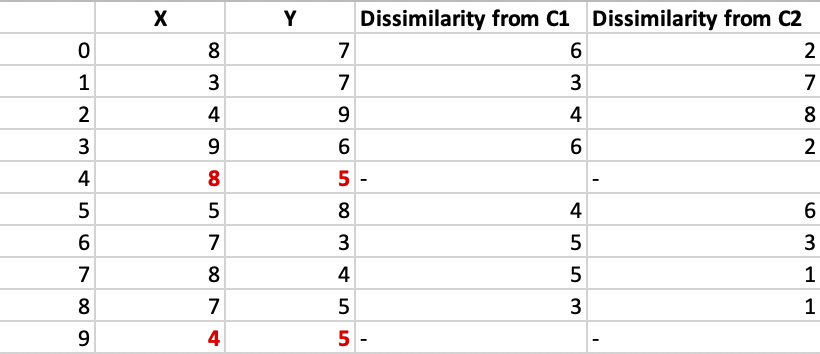


Let’s consider the following example:  If a graph is drawn using the above data points, we obtain the following:



**Step 1:** Let the randomly selected 2 medoids, so select k = 2, and let **C1 -(4, 5)** and **C2 -(8, 5)** are the two medoids.

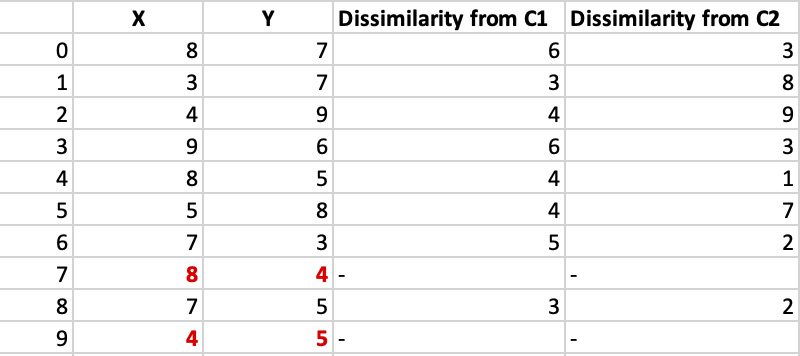
**Step 2: Calculating cost.** The dissimilarity of each non-medoid point with the medoids is calculated and tabulated:



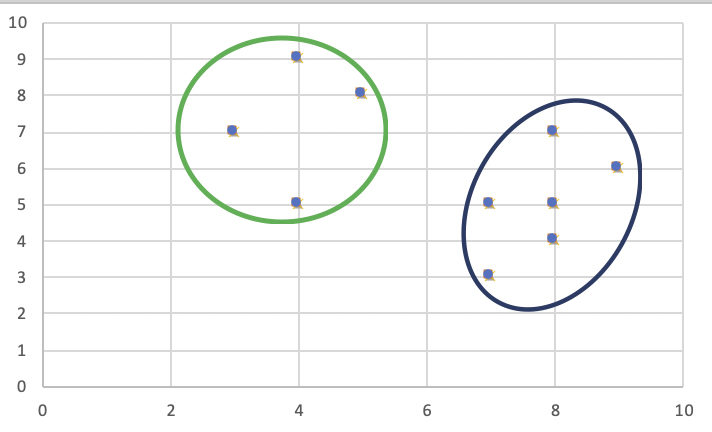
Here we have used Manhattan distance formula to calculate the distance matrices between medoid and non-medoid points. That formula tell that  **Distance = |X1-X2| + |Y1-Y2|.**

 Each point is assigned to the cluster of that medoid whose dissimilarity is less. Points 1, 2, and 5 go to cluster C1 and 0, 3, 6, 7, 8 go to cluster C2. The Cost = (3 + 4 + 4) + (3 + 1 + 1 + 2 + 2) = 20

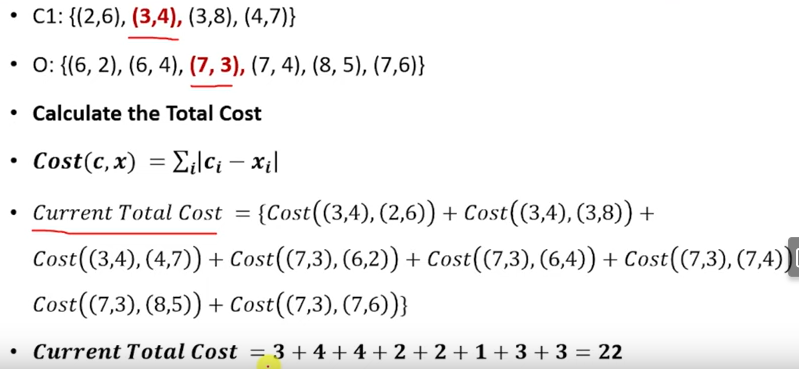
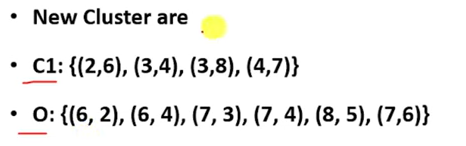
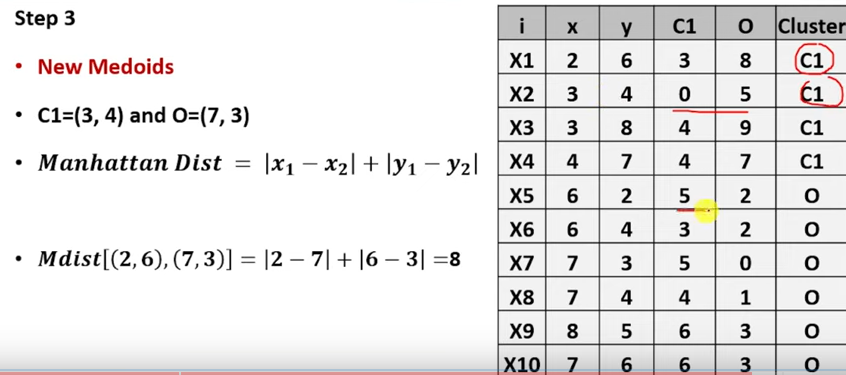
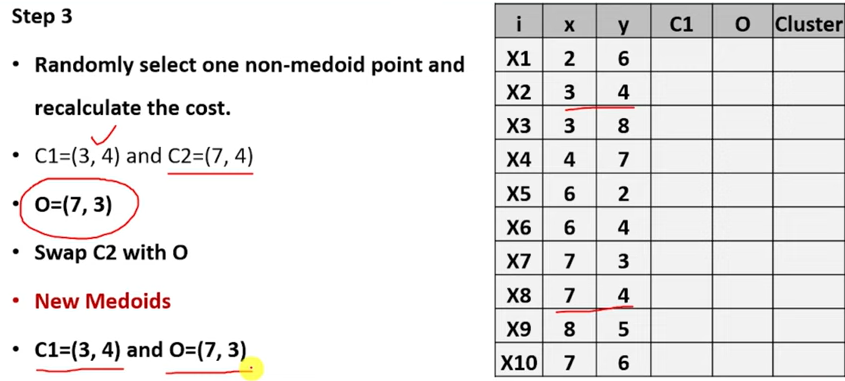
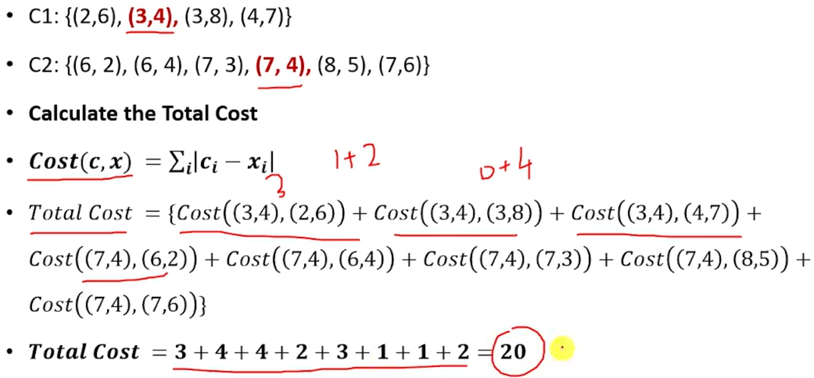
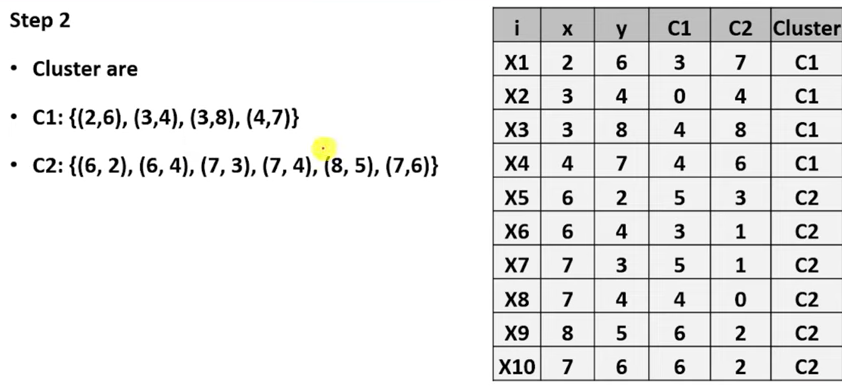
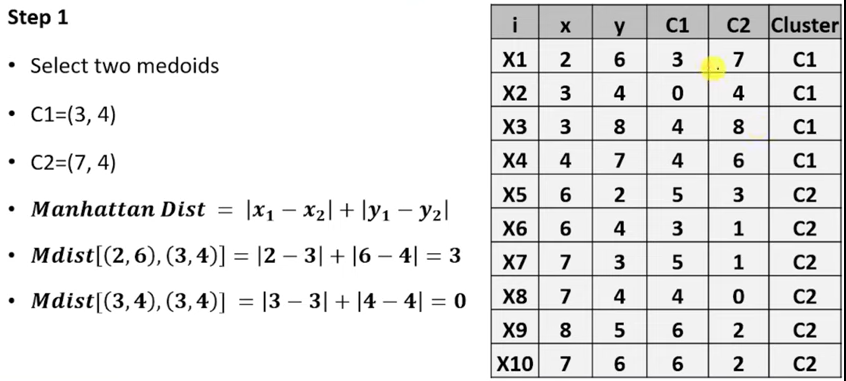
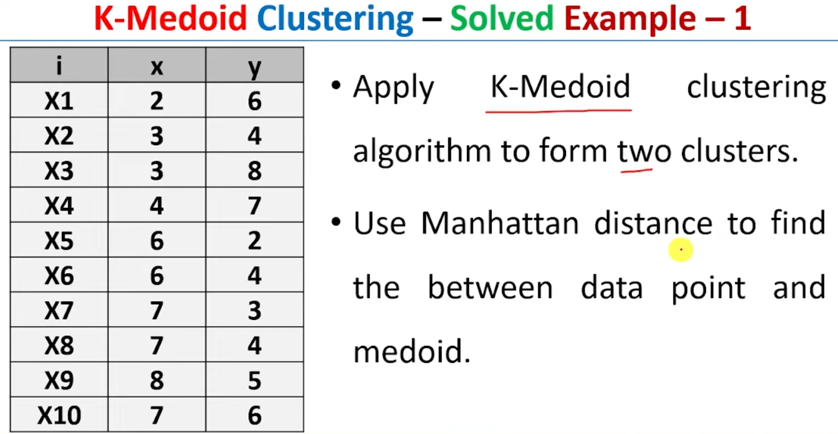
**Step 3: randomly select one non-medoid point and recalculate the cost.** Let the randomly selected point be (8, 4). The dissimilarity of each non-medoid point with the medoids – C1 (4, 5) and C2 (8, 4) is calculated and tabulated.

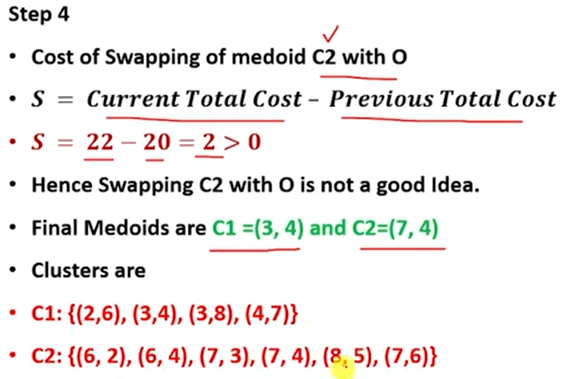


Each point is assigned to that cluster whose dissimilarity is less. So, points 1, 2, and 5 go to cluster C1 and 0, 3, 6, 7, 8 go to cluster C2. The New cost = (3 + 4 + 4) + (2 + 2 + 1 + 3 + 3) = 22 Swap Cost = New Cost – Previous Cost = 22 – 20 and **2 >0** As the swap cost is not less than zero, we undo the swap. Hence (4, 5) and (8, 5) are the final medoids.



**Solved Example**





**Advantages:**

1. It is simple to understand and easy to implement.
2. K-Medoid Algorithm is fast and converges in a fixed number of steps.
3. PAM is less sensitive to outliers than other partitioning algorithms. Deals with noise and outlier data effectively.

**Disadvantages:**

1. The main disadvantage of K-Medoid algorithms is that it is not suitable for clustering non-spherical (arbitrarily shaped) groups of objects. This is because it relies on minimizing the distances between the non-medoid objects and the medoid (the cluster center) – briefly, it uses compactness as clustering criteria instead of connectivity.
2. It may obtain different results for different runs on the same dataset because the first k medoids are chosen randomly.

**K-Means and K-Medoids:**

|  |  |
| --- | --- |
| **K-Means** | **K-Medoids** |
| Both methods are types of Partition Clustering. | |
| Unsupervised iterative algorithms | |
| Have to deal with unlabelled data | |
| Both algorithms group n objects into k clusters based on similar traits where k is pre-defined. | |
| **Inputs:** Unlabelled data and the value of k | |
| **Metric of similarity:** Euclidian Distance | **Metric of similarity:** Manhattan Distance |
| Clustering is done based on distance from **centroids.** | Clustering is done based on distance from **medoids.** |
| A centroid can be a data point or some other point in the cluster | A medoid is always a data point in the cluster. |
| Can't cope with outlier data | Can manage outlier data too |
| Sometimes, outlier sensitivity can turn out to be useful | Tendency to ignore meaningful clusters in outlier data |

**Useful Outlier Clusters:**

For suppose, a data set with data on people's income is being clustered to analyze and understand individuals' purchasing and investing behavior within each cluster. Here outlier data will be people with high incomes-billionaires. All such people tend to purchase and invest more. Hence, a separate cluster for billionaires would be useful in this scenario. In K-Medoids, It merges this data into the upper-class cluster, which loses the meaningful outlier data in Clustering and is one of the disadvantages of K-Medoids in special situations.

**Reference**

1. <https://www.geeksforgeeks.org/ml-k-medoids-clustering-with-example/>
2. <https://www.youtube.com/watch?v=ChBxx4aR-bY>
3. https://www.javatpoint.com/k-medoids-clustering-theoretical-explanation