K Medoids

K-Medoids Clustering

1. What is K-Medoids?

K-Medoids is a partitioning-based clustering algorithm in which the center of each cluster is an **actual data point** from the dataset.

- **K-Means:** cluster center = mean/centroid, which may not be an actual data point.
- **K-Medoids:** cluster center = actual data point called a **medoid**.
- A medoid is the point that gives relatively small total dissimilarity to the other points in its cluster.

- A common implementation is **PAM (Partitioning Around Medoids)**.

Objective

For a point $P_i = (x_i, y_i)$ and medoid $C_j = (x_j, y_j)$:

$$d(P_i, C_j) = |x_i - x_j| + |y_i - y_j|$$

This is the **Manhattan distance** used in this example.

The objective is to minimize:

$$\boxed{\text{Total Cost} = \sum \text{distance of each point to its assigned medoid}}$$

---

2. Given Dataset

We have **10 data points** and need **2 clusters**, therefore:

$$\boxed{k = 2}$$

Point	X	Y
P0	8	7
P1	3	7
P2	4	9
P3	9	6
P4	8	5
P5	5	8
P6	7	3
P7	8	4
P8	7	5
P9	4	5

---

3. Step 1 — Select Initial Medoids

The example randomly selects:

$$\boxed{C_1 = (4, 5) = P9}$$

$$\boxed{C_2 = (8, 5) = P4}$$

Important: The medoids are actual points in the dataset.

So initially:

- **P9 = C1**
- **P4 = C2**

These two points are therefore not calculated as ordinary non-medoid points in the cost table.

---

4. Step 2 — Calculate Dissimilarity

For every **non-medoid point**, calculate its Manhattan distance from both medoids.

Formula

$$d(P, C) = |X_P - X_C| + |Y_P - Y_C|$$

Example 1: P0 = (8,7)

Distance from $C_1 = (4, 5)$:

$$|8 - 4| + |7 - 5| = 4 + 2 = \boxed{6}$$

Distance from $C_2 = (8, 5)$:

$$|8 - 8| + |7 - 5| = 0 + 2 = \boxed{2}$$

Therefore, P0 goes to **C2** because 2 < 6.

Example 2: P1 = (3,7)

$$d(P1, C_1) = |3 - 4| + |7 - 5| = 1 + 2 = \boxed{3}$$

$$d(P1, C_2) = |3 - 8| + |7 - 5| = 5 + 2 = \boxed{7}$$

Therefore, P1 → **C1**.

---

5. Complete Initial Distance Table

Point	X	Y	Distance from C1 (4,5)	Distance from C2 (8,5)	Assigned Cluster	Minim C
P0	8	7	6	2	C2	2
P1	3	7	3	7	C1	3
P2	4	9	4	8	C1	4
P3	9	6	6	2	C2	2
P4	**8**	**5**	–	–	**C2 (Medoid)**	0
P5	5	8	4	6	C1	4
P6	7	3	5	3	C2	3
P7	8	4	5	1	C2	1
P8	7	5	3	1	C2	1
P9	**4**	**5**	–	–	**C1 (Medoid)**	0

Why are P4 and P9 shown as "–"?

Because they are currently the **medoids**.

Their contribution to the objective function is effectively:

$$d(C_1, C_1) = 0, \qquad d(C_2, C_2) = 0$$

---

6. Assign Each Point to the Nearest Medoid

We compare the two distances for every non-medoid point.

Point	C1 Distance	C2 Distance	Decision
P0	6	**2**	C2
P1	**3**	7	C1
P2	**4**	8	C1
P3	6	**2**	C2
P5	**4**	6	C1
P6	5	**3**	C2
P7	5	**1**	C2
P8	3	**1**	C2

Therefore:

$$\boxed{C_1 = \{P1, P2, P5, P9\}}$$

$$\boxed{C_2 = \{P0, P3, P4, P6, P7, P8\}}$$

where P9 and P4 are the medoids.

---

7. Calculate Initial Total Cost

Only the **minimum distance** for each point contributes to the cost.

Cluster C1

P1 = 3

P2 = 4

P5 = 4

P9 = 0

Therefore:

$$Cost(C_1) = 3 + 4 + 4 = \boxed{11}$$

Cluster C2

P0 = 2

P3 = 2

P4 = 0

P6 = 3

P7 = 1

P8 = 1

Therefore:

$$Cost(C_2) = 2 + 2 + 3 + 1 + 1 = \boxed{9}$$

Hence:

$$\boxed{Total\ Cost = 11 + 9 = 20}$$

So the **initial configuration has cost = 20**.

---

8. Step 3 — Perform a Swap

K-Medoids now asks:

> Can we obtain a lower cost by replacing an existing medoid with a non-medoid point?

The example selects:

$$\boxed{P7 = (8, 4)}$$

as the candidate for swapping with the existing medoid:

$$\boxed{C_2 = (8, 5) = P4}$$

Thus:

Before swap

$$C_1 = (4, 5)$$

$$C_2 = (8, 5)$$

After proposed swap

$$C_1 = (4, 5)$$

$$\boxed{C_2' = (8, 4)}$$

Notice that **P4 is no longer a medoid**, while **P7 becomes the new medoid**.

---

9. Recalculate All Distances After the Swap

Now calculate distances from:

$$C_1 = (4, 5)$$

and

$$C_2' = (8, 4)$$

Point	X	Y	Distance from C1 (4,5)	Distance from C2′ (8,4)	Assigned Cluster	Minim C
P0	8	7	6	3	C2	3
P1	3	7	3	8	C1	3
P2	4	9	4	8	C1	4
P3	9	6	6	3	C2	3
P4	8	5	4	**1**	C2	1
P5	5	8	4	7	C1	4
P6	7	3	5	2	C2	2
P7	**8**	**4**	–	–	**C2′ Medoid**	0
P8	7	5	3	2	C2	2
P9	**4**	**5**	–	—	**C1 Medoid**	0

Important observation

P4 was previously a medoid, but after the swap it becomes an ordinary point.

Therefore, we now calculate:

$$d(P4, C_1) = |8-4| + |5-5| = 4$$

$$d(P4, C_2') = |8-8| + |5-4| = 1$$

So P4 moves to C2 with cost **1**.

---

10. New Cluster Assignments

After the proposed swap:

$$\boxed{C_1 = \{P1, P2, P5, P9\}}$$

$$\boxed{C_2 = \{P0, P3, P4, P6, P7, P8\}}$$

The membership happens to remain the same, but the **medoid of C2 has changed**:

$$(8,5) \rightarrow (8,4)$$

---

11. Calculate New Cost

C1

$$3 + 4 + 4 = \boxed{11}$$

C2

$$3 + 3 + 1 + 2 + 0 + 2 = \boxed{11}$$

Therefore:

$$\boxed{New\ Cost = 11 + 11 = 22}$$

---

12. Calculate Swap Cost

The key decision in K-Medoids is based on whether the swap reduces the cost.

$$Swap\ Cost = New\ Cost - Old\ Cost$$

$$= 22 - 20$$

$$\boxed{Swap\ Cost = +2}$$

Interpretation

Swap Cost	Decision
Negative (< 0)	Accept swap
Zero (= 0)	No improvement
Positive (> 0)	Reject/undo swap

Here:

$$\boxed{+2 > 0}$$

Therefore, the proposed swap **increases the cost**.

Hence:

$$\boxed{\text{Reject the swap}}$$

---

13. Final Result of This Example

The original medoids are retained:

$$\boxed{C_1 = (4, 5)}$$

$$\boxed{C_2 = (8, 5)}$$

Final clusters

Cluster	Medoid	Points
Cluster 1	**(4,5)**	P1=(3,7), P2=(4,9), P5=(5,8), P9=(4,5)
Cluster 2	**(8,5)**	P0=(8,7), P3=(9,6), P4=(8,5), P6=(7,3), P7=(8,4), P8=(7,5)

Final objective value

$$\boxed{Total\ Cost = 20}$$

---

14. The Entire K-Medoids Procedure in One Table

Step	What happens	Result
1. Select K	Choose number of clusters	$K = 2$
2. Select medoids	Randomly choose actual data points	C1=(4,5), C2=(8,5)
3. Calculate distance	Use Manhattan distance	(d=
4. Assign points	Choose smaller distance	C1: P1,P2,P5,P9; C2: P0,P3,P4,P6,P7,P8
5. Calculate cost	Add minimum distances	20
6. Select swap candidate	Choose non-medoid P7=(8,4)	Replace C2=(8,5)
7. Recalculate	Compute all distances again	New cost = 22
8. Compare costs	$22 - 20 = +2$	Cost increased
9. Decision	Reject positive-cost swap	Retain original medoids
10. Final	Report clusters and medoids	Cost = **20**

---

15. Points

Medoid vs Centroid

K-Means	K-Medoids
Uses centroid	Uses medoid
Centroid may not be an actual data point	Medoid is always an actual data point
Usually uses Euclidean distance	Can use Manhattan, Euclidean, or other dissimilarities
More affected by outliers	Generally more robust to outliers
Mean-based	Representative-point-based

Three formulas to remember

1. Manhattan distance

$$\boxed{d = |X_1 - X_2| + |Y_1 - Y_2|}$$

2. Total cost

$$\boxed{Cost = \sum_i \min_j d(P_i, C_j)}$$

3. Swap cost

$$\boxed{Swap\ Cost = New\ Cost - Old\ Cost}$$

---

16. Exam-Friendly Logic

Remember the sequence:

$$\boxed{\text{Select Medoids}}$$

↓

$$\boxed{\text{Calculate Distances}}$$

↓

$$\boxed{\text{Assign to Nearest Medoid}}$$

↓

$$\boxed{\text{Calculate Total Cost}}$$

↓

$$\boxed{\text{Try a Medoid–Non-Medoid Swap}}$$

↓

$$\boxed{\text{Recalculate Cost}}$$

↓

$$\boxed{\text{If Cost Decreases} \rightarrow \text{Accept}}$$

$$\boxed{\text{If Cost Increases} \rightarrow \text{Reject}}$$

↓

$$\boxed{\text{Repeat Until No Further Improvement}}$$

One important clarification

The example demonstrates **one particular swap**. Since its cost changes from **20 to 22**, that swap is rejected and the original medoids are retained. Strictly speaking, to establish a **globally converged PAM solution**, all relevant medoid/non-medoid swaps (or the prescribed PAM search procedure) should be evaluated; merely testing one randomly selected swap does not mathematically prove global optimality.

after 12:53 PM.