DATA WAREHOUSING AND MINING

T.E. CSE-DS, Sem V

Clustering: K-mean Clustering,K-medoids example Exercise

Poonam Pangarkar

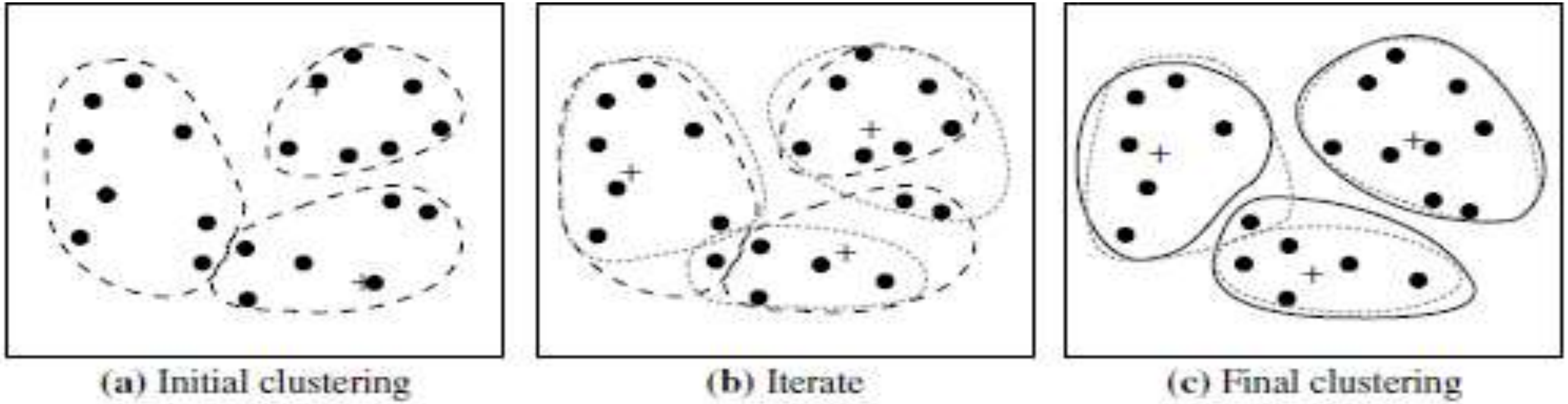
K-means Clustering

- Simple unsupervised learning algorithm developed by J MacQueen in 1967.
- It partitions x data points into the st of k clusters where each data point is assigned to its closets cluster.
- This method is defined by objective function which tries to minimize the sum of all squared distances within a cluster, for all clusters.
- An objective function is used to assess the partitioning quality so that objects within a cluster are similar to one another but dissimilar to objects in other clusters.

The objective function used is Euclidean Distance

- The k-means algorithm defines the centroid of a cluster as the mean value of the points within the cluster.
- First, it randomly selects k of the objects in D, each of which initially represents a cluster center(mean).
- Remaining objects, are assigned to the cluster to which it is the most similar, based on the Euclidean distance between the object and the cluster mean.

- The k-means algorithm then iteratively improves the within-cluster variation.
- For each cluster, it computes the new mean using the objects assigned to the cluster in the previous iteration.
- All the objects are then reassigned using the updated means as the new cluster centers.
- The iterations continue until the clusters formed in the current round are the same as those formed in the previous round.



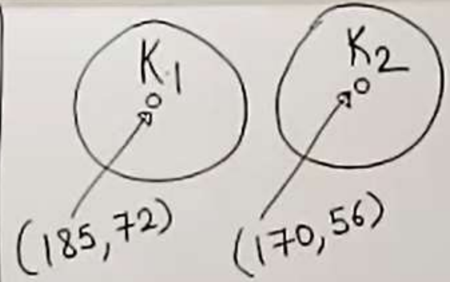
Clustering of a set of objects using the k-means method; for (b) update cluster centers and reassign objects accordingly (the mean of each cluster is marked by a $+$).

K-means Algorithm

	Height	weight
1	185	72
2	170	56
3	168	60
4	179	68
5	182	72
6	188	77
7	180	71
8	180	70
9	183	84
10	180	88
11	180	67
12	177	76

Euclidean Distance

$$\sqrt{(X_0 - X_c)^2 + (Y_0 - Y_c)^2}$$



ED for 3 $\rightarrow K_1 \rightarrow \sqrt{(168-185)^2+(60-72)^2}$

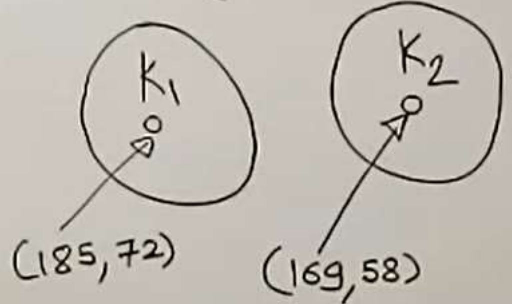
20.80

$\rightarrow K_2 \rightarrow \sqrt{(168-170)^2+(60-56)^2}$

$= 4.48$

New centroid Calculation :-

for $K_2 = \left(\frac{170+168}{2}, \frac{60+56}{2}\right) = (169, 58)$



E.D for 4 $\rightarrow K_1 = \sqrt{(179-185)^2+(68-72)^2}$

$= (6.32)$

$\rightarrow K_2 = \sqrt{(179-169)^2+(68-58)^2}$

$= 14.14$

$K_1 \rightarrow \{1, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$

$K_2 \rightarrow \{2, 3\}$

K-Means Clustering Algorithm

It is a centroid-based partitioning technique.

It uses the centroid of a cluster, Ci , to represent that cluster.

Conceptually, the centroid of a cluster is its center point.

The k-means algorithm defines the centroid of a cluster as the mean value of the points within the cluster

Input :

k: the number of clusters

D: a data set containing n objects

Output : A set of k clusters.

K-means Clustering

- Arbitrarily choose k objects from D as the initial cluster centres.
- Repeat
  - (re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;
  - update the cluster means, that is, calculate the mean value of the objects for each cluster;
- until no change;

Limitations of K-Means Clustering

- ✓ There are several variants of the k-means method.
- ✓ These can differ in the **selection of the initial k-means**, the **calculation of dissimilarity**, and the strategies for **calculating cluster means.**
- ✓ Can be applied only when the mean of a set of objects is defined.
- ✓ This may not be the case in some applications such as when data with nominal attributes are involved.
- ✓ The k-modes method is a variant of k-means, which extends the k-means paradigm to cluster nominal data by replacing the means of clusters with modes.
- ✓ It uses new dissimilarity measures to deal with nominal objects and a frequency-based method to update modes of clusters

Limitations of K-Means Clustering

✓User needs to specify k, the number of clusters in advanced.

✓Various models with different values of k can be generated and then the best model can be chosen.

✓Not suitable for nonconvex cluster shapes.

✓Sensitive to outliers and noise in the data as they can substantially influence the mean.

K –Medoid Algorithm

- K- means algorithm is sensitive to outliers
- This could affect the assignment of the other objects to the cluster
- Instead of taking mean as the reference point, actual objects can be picked to represent the clusters.
- Each remaining object is assigned to the cluster of which the representative object is the most similar.
- These objects are called Medoids.
- A medoid the point in the cluster, whose dissimilarities with all the other points in the cluster are minimum.
- Medoid – least dissimilar object/ most similar Object
- Such partitioning method is also called as Partitioning Around Medoids (PAM).

K-Medoid Algorithm

The dissimilarity of the medoid(Ci) and object(Pi)

is calculated by using **E = |Pi – Ci|**, also called **absolute-error criterion**

Algorithm:

1. Initialize: select *k* random points out of the *n* data points as the medoids.
2. Associate each data point to the closest medoid by using any common distance metric methods.
3. While the cost decreases: For each medoid m, for each data o point which is not a medoid:
    1. Swap m and o, associate each data point to the closest medoid, recompute the cost.
    2. If the total cost is more than that in the previous step, undo the swap.

K-MEDOID EXAMPLE

K=2

i	x	y
x_1	2	6
x_2	3	4
x_3	3	8
x_4	4	7
x_5	6	2
x_6	6	4
x_7	7	3
x_8	7	4
x_9	8	5
x_{10}	7	6

Step 1

we select two random representative objects:

$c_1(3, 4)$, $c_2(7, 4)$

1 [illegible] + 2

i	x	y	c 1		Distance / cost	C
x_1	2	6	3	4	$\|2-3\| + \|6-4\|$ =	3
x_3	3	8	3	4	0 + 4	4
x_4	4	7	3	4	1 + 3	4
x_5	6	2	3	4	3 + 2	5
x_6	6	4	3	4	3 + 0	3
x_7	7	3	3	4	4 + 1	5
x_9	8	5	3	4	5 + 1	6
x_{10}	7	6	3	4	4 + 2	6

$m = (a, b)$

$n = (c, d)$

Distance = $|a-c| + |b-d|$

i	x	y	c_2		Distance / cost	c
x_1	2	6	7	4	$\|2-7\| + \|6-4\|$	7
x_3	3	8	7	4	4 + 4	8
x_4	4	7	7	4	3 + 3	6
x_5	6	2	7	4	1 + 2	3
x_6	6	4	7	4	1 + 0	1
x_7	7	3	7	4	0 + 1	1
x_9	8	5	7	4	1 + 1	2
x_{10}	7	6	7	4	0 + 2	2

Compare cost of Cost(c_1) and Cost(c_2) for every i & Select the minimum one

1 [illegible] + 2

i	x	y	c1		Distance / cost	C	C
x_1	2	6	3	4	$\lvert 2-3\rvert + \lvert 6-4\rvert$ =	3	7
x_3	3	8	3	4	0 + 4	4	8
x_4	4	7	3	4	1 + 3	4	6
x_5	6	2	3	4	3 + 2	5	3
x_6	6	4	3	4	3 + 0	3	1
x_7	7	3	3	4	4 + 1	5	1
x_9	8	5	3	4	5 + 1	6	2
x_{10}	7	6	3	4	4 + 2	6	2

elect

$$m = (a, b)$$

$$n = (c, d)$$

$$\text{Distance} = \lvert a-c\rvert + \lvert b-d\rvert$$

Step II) then cluster are

cluster 1: $\{(2,6), (3,8), (4,7), (3,4)\}$

cluster 2: $\{(7,4), (6,2), (6,4), (7,3), (8,5), (7,6)\}$

Calculate total cost

$$T\ cost(x,c) = \sum_{i=1}^{d} |x_i - c_i|$$

Total cost = $\{cost((3,4),(2,6)), cost((3,4),(3,8)),$
$cost((3,4),(4,7)), cost((7,4),(8,5)),$
$cost((7,4),(6,2)), cost((7,4),(6,4)),$
$cost((7,4),(7,3)), cost((7,4),(7,6)),$

$= (3+4+4) + (3+1+1+2+2)$

$= \underline{20}$

Step 3) Select one of non-Medoids O'

Let's $O' = (7,3)$ i.e x_7

So now medoid's are $c_1(3,4)$ & $O'(7,3)$

i	x	y	O'		Distance/cost =	C
x_1	2	6	7	3	$\lvert 2-7\rvert + \lvert 6-3\rvert$	8
x_3	3	8	7	3		
x_4	4	7	7	3		
x_5	6	2	7	3		
x_6	6	4	7	3		
x_8	7	4	7	3		
x_9	8	5	7	3		
x_{10}	7	6	7	3		

Compare the cost of cost(c_1) and cost(O') every i & Select the minimum one

i	x	y	c_1		Distance / cost	C
x_1	2	6	3	4	$\|2-3\| + \|6-4\|$	$= 3$
x_3	3	8	3	4	$0 + 4$	$= 4$
x_4	4	7	3	4	$1 + 3$	$= 4$
x_5	6	2	3	4	$3 + 2$	$= 5$
x_6	6	4	3	4	$3 + 0$	$= 3$
x_8	7	4	3	4	$4 + 0$	$= 4$
x_9	8	5	3	4	$5 + 1$	$= 6$
x_{10}	7	6	3	4	$4 + 2$	$= 6$

Again create the cluster

cluster 1: {(3,4), (2,6), (3,8), (4,7)}

cluster 2: {(7,3), (6,2), (5,4), (7,4), (8,5), (7,6)}

current total cost = (3+4+4) + (2+2+1+3+3)

= 11 + 11

= 22

step 4) So cost of swapping medoid from C_2 to O''s

S = current total cost − past total cost

= 22 − 20

= 2 > 0

so, moving O' would be a bad idea so previous choice was Good

Example

	X	Y
0	8	7
1	3	7
2	4	9
3	9	6
4	**8**	**5**
5	5	8
6	7	3
7	8	4
8	7	5
9	**4**	**5**

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.

Example

	X	Y	Dissimilarity from C1	Dissimilarity from C2
0	8	7	6	2
1	3	7	3	7
2	4	9	4	8
3	9	6	6	2
4	8	5	-	-
5	5	8	4	6
6	7	3	5	3
7	8	4	5	1
8	7	5	3	1
9	4	5	-	-

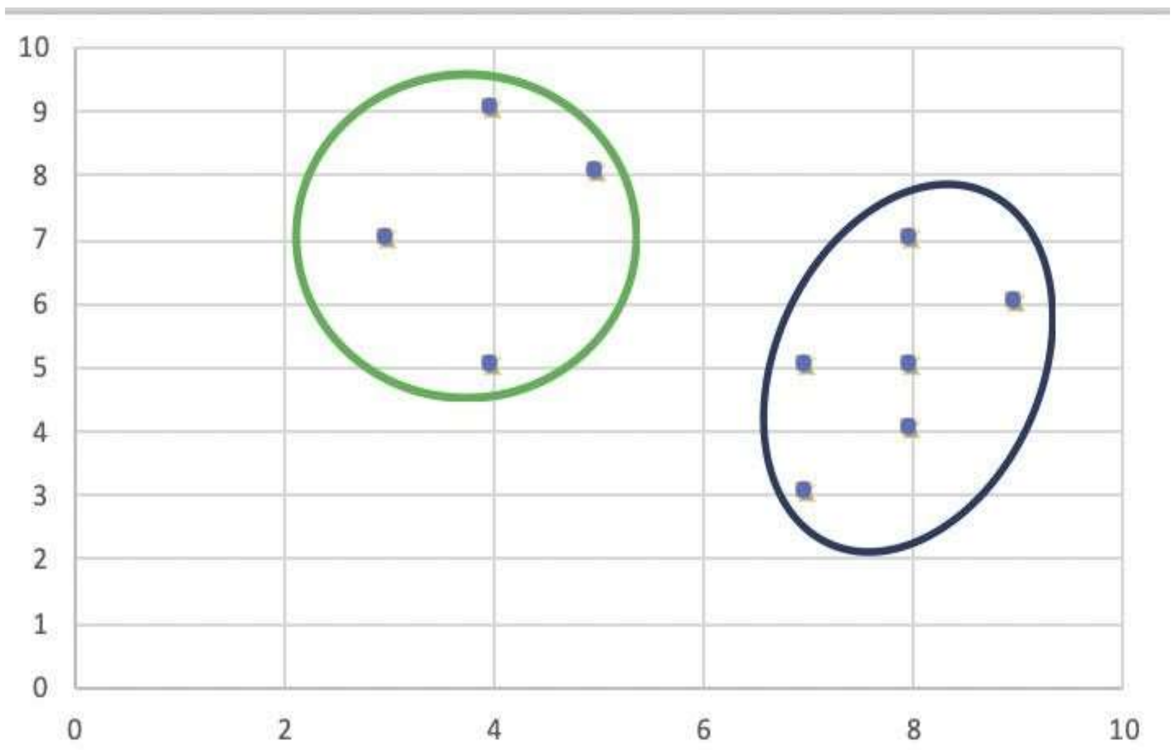
- **Step 2: Calculating cost.** The dissimilarity of each non-medoid point with the medoids is calculated and tabulated.
- Each point is assigned to the cluster of that medoid whose dissimilarity is less.
- The points 1, 2, 5 go to cluster C1 and 0, 3, 6, 7, 8 go to cluster C2.
- The Cost = (3 + 4 + 4) + (3 + 1 + 1 + 2 + 2) = 20

Example

	X	Y	Dissimilarity from C1	Dissimilarity from C2
0	8	7	6	3
1	3	7	3	8
2	4	9	4	9
3	9	6	6	3
4	8	5	4	1
5	5	8	4	7
6	7	3	5	2
7	8	4	-	-
8	7	5	3	2
9	4	5	-	-

- **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, the points 1, 2, 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**

Example



Step 4: Calculate Swap Cost

Swap Cost = New Cost – Previous Cost = 22 – 20

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.