DataScience Homework - 5.

Phani Teja Kesha

KP38691

1. The given Points are

A_1 (2,10)

A_2 (2,5)

A_3 (8,4)

B_1 (5,8)

B_2 (7,5)

B_3 (6,4)

C_1 (1,2)

C_2 (4,9)

Here A_1, B_1, C_1 are assigned as the initial centers of the cluster.

→ Eucledian distance gives you the distance between two Points.

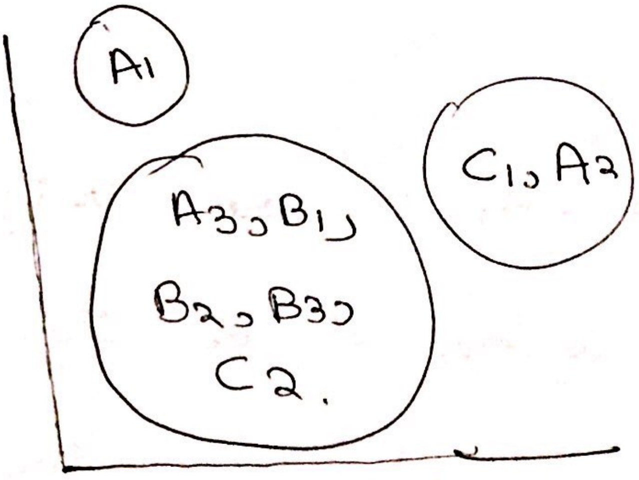
If (x_1,y_1) & (x_2,y_2), The distance is

$$\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$$

After a single iteration.

	A_1	~~A2~~	~~B3~~ C1 ~~B1~~ ~~B2~~ ~~B3~~ ~~C1~~ ~~C3~~
A_1	$\boxed{0}$	3.6	8.06
A_2	5	4.2	$\boxed{3.16}$
A_3	8.4	$\boxed{5}$	7.28
B_1	3.6	$\boxed{0}$	7.21
B_2	7.07	$\boxed{2.6}$	6.7
B_3	7.2	$\boxed{4.1}$	5.3
C_1	8.06	7.2	$\boxed{0}$
C_2	2.2	$\boxed{1.41}$	7.6

} Picking the nearest centers from given values.



→ Let's check for the Second iteration. For the next iteration centers are updated by doing the average of all the Points wo the given cluster.

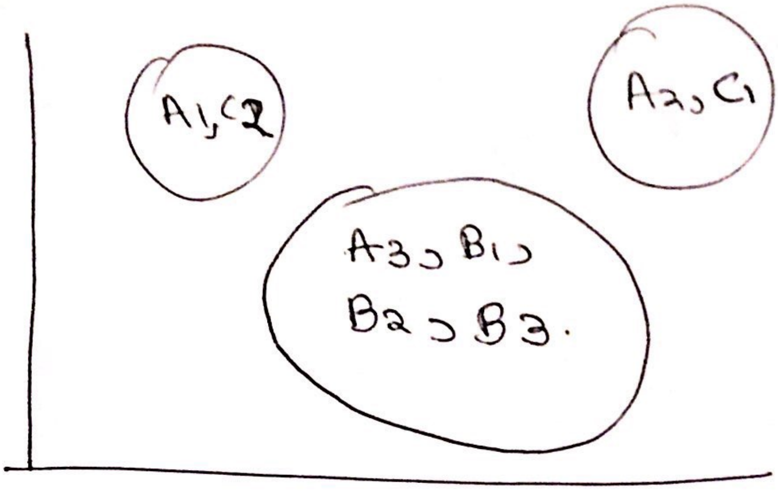
$C_1 = (2, 10)$

$C_2 = (6, 6)$

$C_3 = (1.5, 3.5)$

Second Iteration:-

	C_1	C_2	C_3
A_1	$\boxed{0}$	5.65	6.5
A_2	5	4.12	$\boxed{1.58}$
A_3	8.4	$\boxed{2.82}$	6.51
B_1	3.6	$\boxed{2.23}$	5.7
B_2	7.07	$\boxed{1.41}$	5.7
B_3	7.2	$\boxed{2}$	4.52
C_1	8.06	6.4	$\boxed{1.58}$
C_2	$\boxed{2.2}$	3.6	6.04



→ Again updating the Centers.

$C_1 = (3, 9.5)$

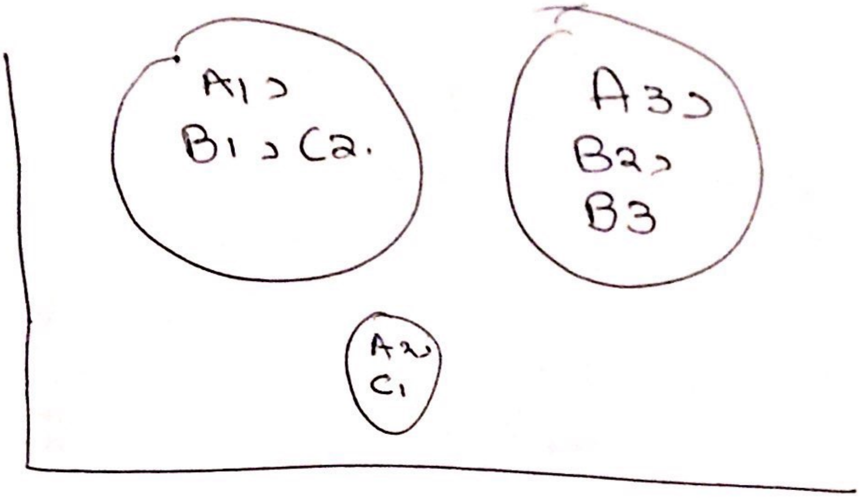
$C_2 = (6.5, 5.25)$

$C_3 = (1.5, 3.5)$

Third Iteration:

	C_1	C_2	C_3
A_1	$\boxed{1.11}$	6.5	6.5
A_2	4.6	4.5	$\boxed{1.58}$
A_3	7.4	$\boxed{1.95}$	6.51
B_1	$\boxed{2.5}$	3.13	5.7
B_2	6.02	$\boxed{0.5}$	5.7
B_3	6.26	$\boxed{1.34}$	4.52
C_1	7.7	6.3	$\boxed{1.58}$
C_2	$\boxed{1.11}$	4.5	6.04.

KP38691



→ Calculating Centers

$C_1 = (3.66, 9)$

$C_2 = (7, 4.33)$

$C_3 = (1.5, 3.5.$

Fourth Iteration :-

	C_1	C_2	C_3
A_1	$\boxed{1.93}$	7.55	6.5
A_2	4.33	5.04	$\boxed{1.58}$
A_3	6.62	$\boxed{4.05}$	6.5
B_1	$\boxed{1.67}$	4.17	5.7
B_2	5.2	$\boxed{0.67}$	5.7
B_3	5.52	$\boxed{1.05}$	4.52
C_1	7.4	6.43	$\boxed{1.58}$
C_2	$\boxed{6.34}$	5.5	6.04

// centers are Same, & not any Change.

→ Since the centers are not changed, it has reached the best Possible cluster.

1a)

Center1	Center 2	Center 3
A1	A3, B1, B2 B3, C2	C1, A2.

1b)

Center1	Center 2	Center 3
A1, B1, C2	A3, B2, B3	A2, C1

2)
10.10) why does BIRCH encounter difficulty in finding clusters of arbitary Shape but OPTICS does not? Propose modifications to BIRCH to help it find clusters of arbitary Shape?

BIRCH uses the euclidean distance to find distance between the Points, Due to this the Shape is nearly spherical, the resulting cluster is not in the arbitary shape. whereas OPTICS uses density of the Points as the metric for distance all the Points which are very close enough or within a minimum density measure form a cluster which Can be of any arbitary Shape, which is based on the original Position of the Points.

As a modification to BIRCH, it Can be modified to use dynam density measure to form arbitary shaped Clusters by clustering very low level B+ trees, which have very closely Positioned Points. This will turn take density measure and not distance measure, So that form a CF tree and will result in arbitary Shaped clusters.

3)

→ Partition based and hierarchial Clustering uses distance as a measure for creating clusters of the given Points.

→ Due to this distance measures they end up having spherical shaped clusters.

→ But in the case of density based clustering, takes uses of the fact that the every cluster formed has a density different than other formed clusters.

→ Density clustering method takes the clusters as the dense region of the data Points

→ In this way they arbitarily form arbitary shaped cluster dense regions and are very suitable than other clustering methods in a way that they can separate those from less density regions in the data Point space.

4) Basically, we are given the datapoints and need to build the clustering depending on the Contstraints and Kmeans or other clustering techniques.

→ Lets choose some random ATM's as Centroids and for each point we assign it with the nearest Centroid. Satisfying the given Constraints which we have enforced on the data Points.

→ If a datapoint there is no Centroid Satisfying the Constraint then the datapoint is not assigned to any centroid. and the datapoint is then updated to the Centroid list.

→ This Process is repeated untill any Convergence is found.

→ The algorithm goes as follows

1) Define Constraints function which has Constraints like → 10000 households per cluster

→ No obstacle objects

2) Randomly choose Centroid Datapoints & assign it to all Datapoints

3) If a Datapoint has no nearest centroid with given requirements then add the datapoint to Centroid list.

4) Repeat This untill Convergence.

5) Then the Points with no Clusters assigned ~~by~~ to ATM ~~are~~ ~~to~~ by dropping all the Contstraints.