K-MEANS CLUSTERING

What is clustering?

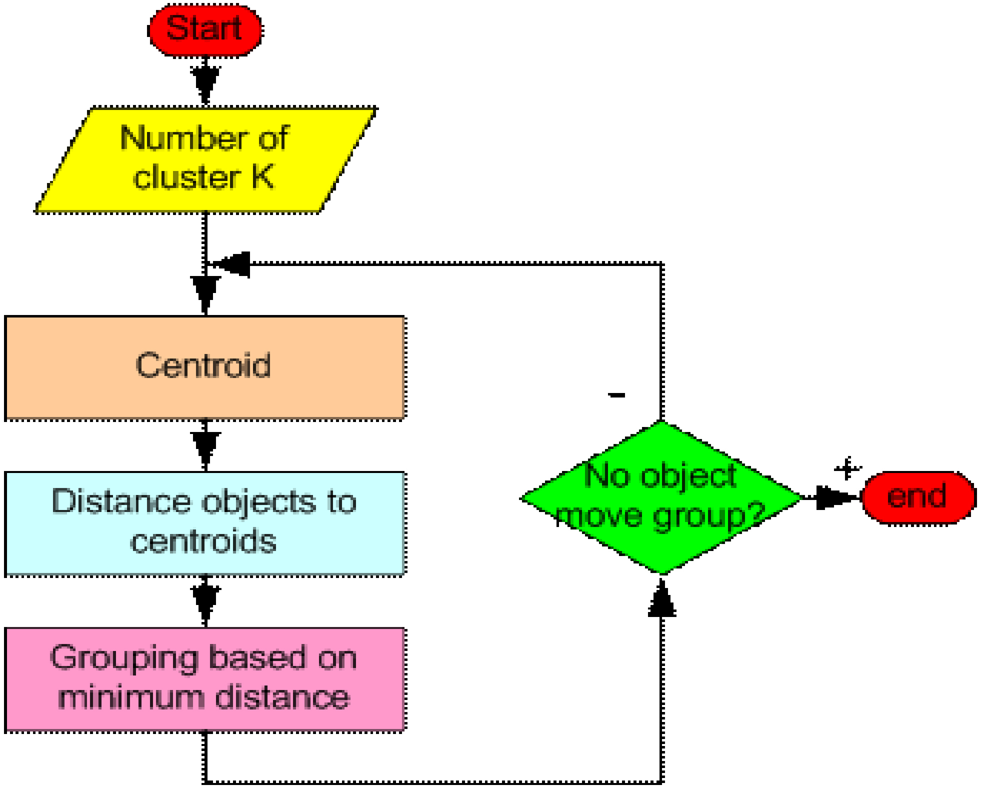
- **Clustering** is the classification of objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often according to some defined distance measure.

K-MEANS CLUSTERING

- The **k-means algorithm** is an algorithm to cluster *n* objects based on attributes into *k* partitions, where $k < n$.
- It is similar to the expectation-maximization algorithm for mixtures of Gaussians in that they both attempt to find the centers of natural clusters in the data.

- Simply speaking k-means clustering is an algorithm to classify or to group the objects based on attributes/features into K number of group.
- K is positive integer number.
- The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid.

How the K-Mean Clustering algorithm works?



Steps

- **Step 1:** Begin with a decision on the value of k = number of clusters .
- **Step 2**: Put any initial partition that classifies the data into k clusters. You may assign the training samples randomly,or systematically as the following:

  1. Take the first k training sample as single-element clusters
  2. Assign each of the remaining (N-k) training sample to the cluster with the nearest centroid. After each assignment, recompute the centroid of the gaining cluster.

- **<u>Step 3:</u>** Take each sample in sequence and compute its [distance] from the centroid of each of the clusters. If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the cluster gaining the new sample and the cluster losing the sample.
- **<u>Step 4 .</u>** Repeat step 3 until convergence is achieved, that is until a pass through the training sample causes no new assignments.

How to choose the value of "K number of clusters" in K-means Clustering?

- The Elbow method is one of the most popular ways to find the optimal number of clusters.
- This method uses the concept of WCSS value. **WCSS** stands for **Within Cluster Sum of Squares**, which defines the total variations within a cluster. The formula to calculate the value of WCSS (for 3 clusters) is given below:

$$WCSS= \sum_{P_{i \text{ in Cluster1}}} \text{distance}(P_i\ C_1)^2 + \sum_{P_{i \text{ in Cluster2}}} \text{distance}(P_i\ C_2)^2 + \sum_{P_{i \text{ in CLuster3}}} \text{distance}(P_i\ C_3)^2$$

In the above formula of WCSS,

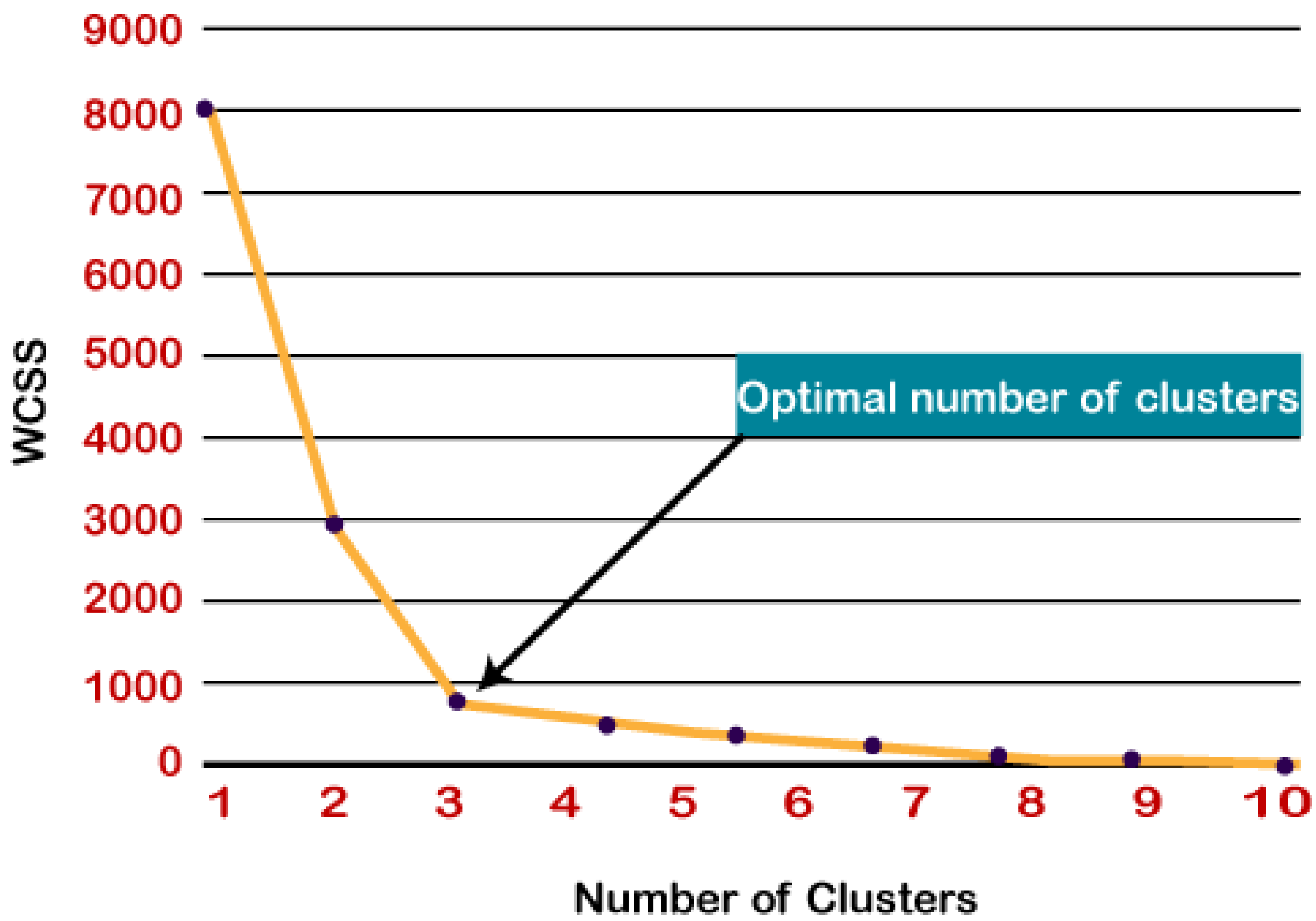
$\sum_{P_{i \text{ in Cluster1}}} \text{distance}(P_i\ C_1)^2$: It is the sum of the square of the distances between each data point and its centroid within a cluster1 and the same for the other two terms.

To find the optimal value of clusters, the elbow method follows the below steps:

- It executes the K-means clustering on a given dataset for different K values (ranges from 1-10).
- For each value of K, calculates the WCSS value.

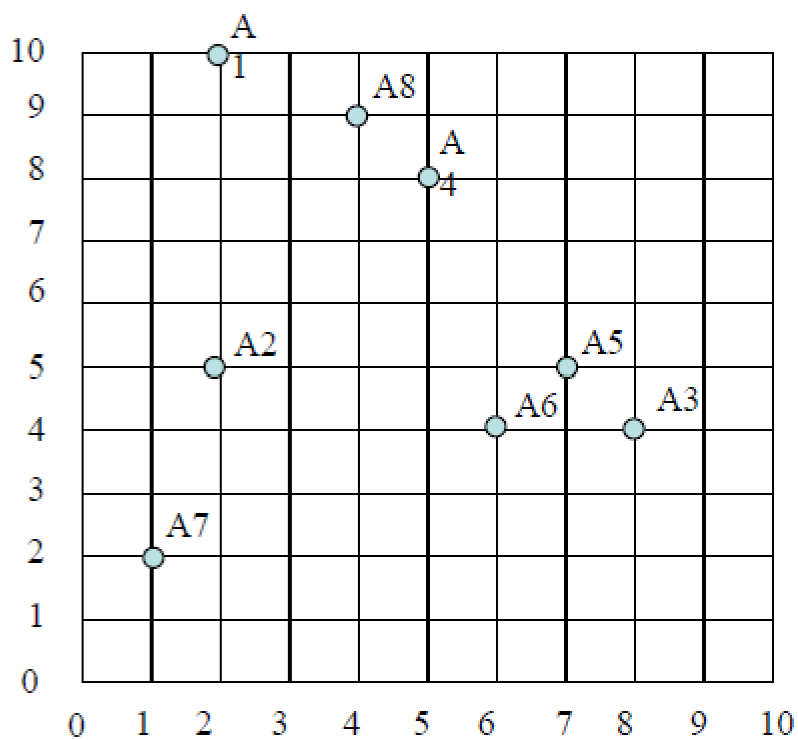
Plots a curve between calculated WCSS values and the number of clusters K.

- The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value of K.


9000
8000
7000
6000
5000
4000
3000
2000
1000
0
WCSS
Optimal number of clusters
1
2
3
4
5
6
7
8
9
10
Number of Clusters

Example

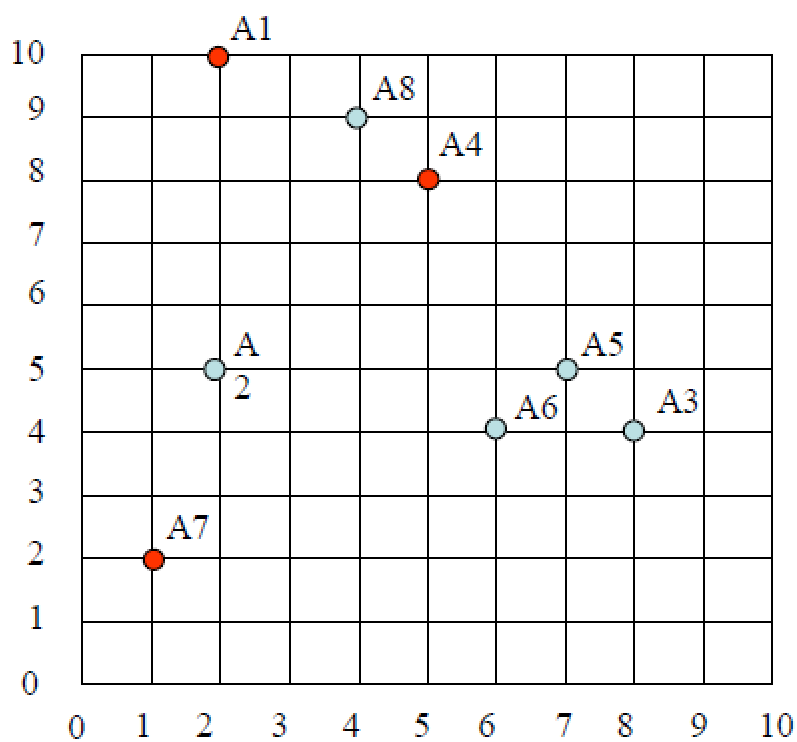
Use the k-means algorithm and Euclidean distance to cluster the following 8 examples into 3 clusters:
A1=(2,10), A2=(2,5), A3=(8,4), A4=(5,8), A5=(7,5), A6=(6,4), A7=(1,2), A8=(4,9).



a)

d(a,b) denotes the Eucledian distance between a and b. It is obtained directly from the distance matrix or calculated as follows: $d(a,b)=sqrt((x_b-x_a)^2+(y_b-y_a)^2)$

seed1=A1=(2,10), seed2=A4=(5,8), seed3=A7=(1,2)



The distance matrix based on the Euclidean distance is given below:

	A1	A2	A3	A4	A5	A6	A7	A8
A1	0	$\sqrt{25}$	$\sqrt{36}$	$\sqrt{13}$	$\sqrt{50}$	$\sqrt{52}$	$\sqrt{65}$	$\sqrt{5}$
A2		0	$\sqrt{37}$	$\sqrt{18}$	$\sqrt{25}$	$\sqrt{17}$	$\sqrt{10}$	$\sqrt{20}$
A3			0	$\sqrt{25}$	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{53}$	$\sqrt{41}$
A4				0	$\sqrt{13}$	$\sqrt{17}$	$\sqrt{52}$	$\sqrt{2}$
A5					0	$\sqrt{2}$	$\sqrt{45}$	$\sqrt{25}$
A6						0	$\sqrt{29}$	$\sqrt{29}$
A7							0	$\sqrt{58}$
A8								0

seed1=A1=(2,10), seed2=A4=(5,8), seed3=A7=(1,2)

A1=(2,10), A2=(2,5), A3=(8,4), A4=(5,8), A5=(7,5), A6=(6,4), A7=(1,2), A8=(4,9).

seed1=A1=(2,10), seed2=A4=(5,8), seed3=A7=(1,2)

epoch1 – start:

A1:
d(A1, seed1)=0 as A1 is seed1
d(A1, seed2)= $\sqrt{13} > 0$
d(A1, seed3)= $\sqrt{65} > 0$
$\rightarrow$ A1 $\in$ cluster1

A2:
d(A2,seed1)= $\sqrt{25} = 5$
d(A2, seed2)= $\sqrt{18} = 4.24$
d(A2, seed3)= $\sqrt{10} = 3.16$
$\rightarrow$ A2 $\in$ cluster3

A3:
d(A3, seed1)= $\sqrt{36} = 6$
d(A3, seed2)= $\sqrt{25} = 5$
d(A3, seed3)= $\sqrt{53} = 7.28$
$\rightarrow$ A3 $\in$ cluster2

A4:
d(A4, seed1)= $\sqrt{13}$
d(A4, seed2)=0 as A4 is seed2
d(A4, seed3)= $\sqrt{52} > 0$

A5:
d(A5, seed1)= $\sqrt{50} = 7.07$
d(A5, seed2)= $\sqrt{13} = 3.60$
d(A5, seed3)= $\sqrt{45} = 6.70$
$\rightarrow$ A5 $\in$ cluster2

A6:
d(A6, seed1)= $\sqrt{52} = 7.21$
d(A6, seed2)= $\sqrt{17} = 4.12$
d(A6, seed3)= $\sqrt{29} = 5.38$
$\rightarrow$ A6 $\in$ cluster2

A7:

d(A7, seed1)= $\sqrt{65} > 0$

d(A7, seed2)= $\sqrt{52} > 0$

d(A7, seed3)=0 as A7 is seed3

➔ A7 $\in$ cluster3

A8:

d(A8, seed1)= $\sqrt{5}$

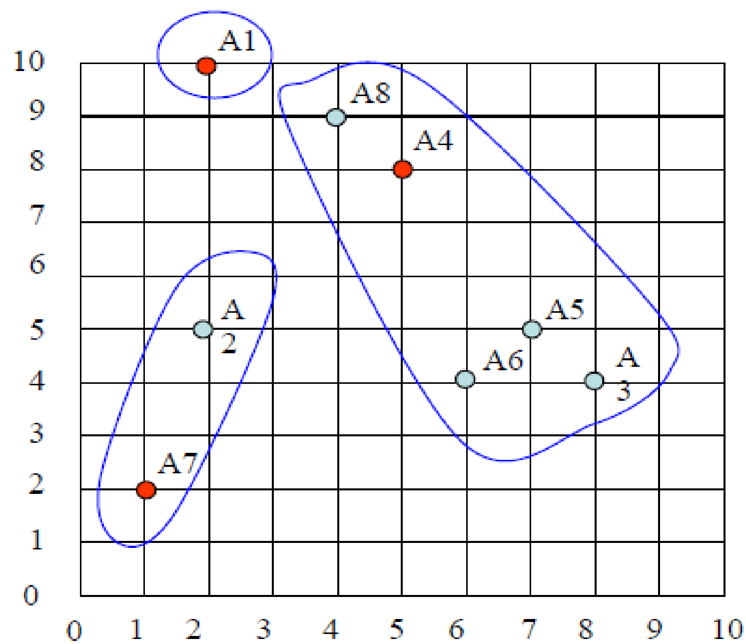
d(A8, seed2)= $\sqrt{2}$

d(A8, seed3)= $\sqrt{58}$

➔ A8 $\in$ cluster2

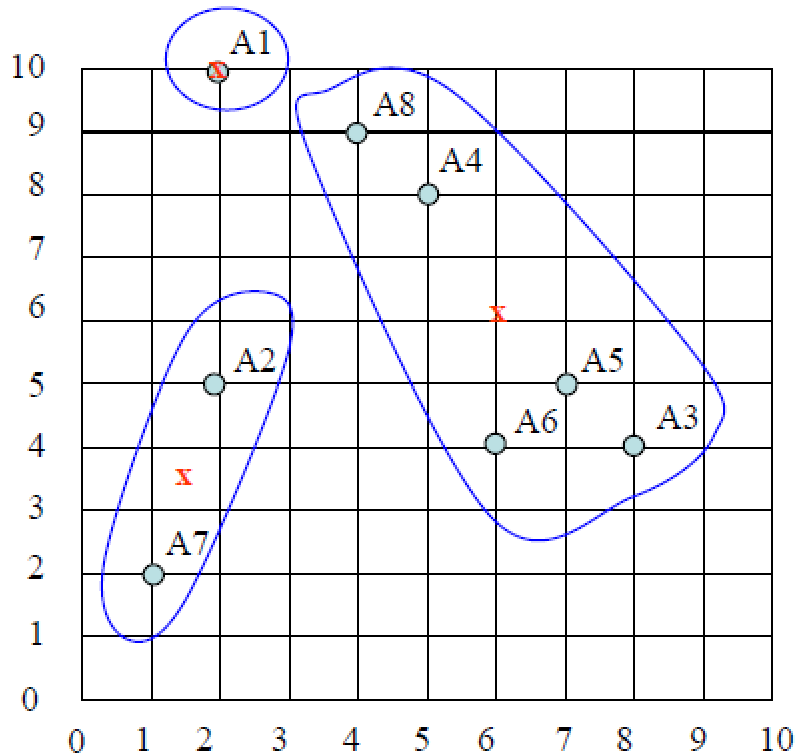
end of epoch1

new clusters: 1: {A1}, 2: {A3, A4, A5, A6, A8}, 3: {A2, A7}



centers of the new clusters:

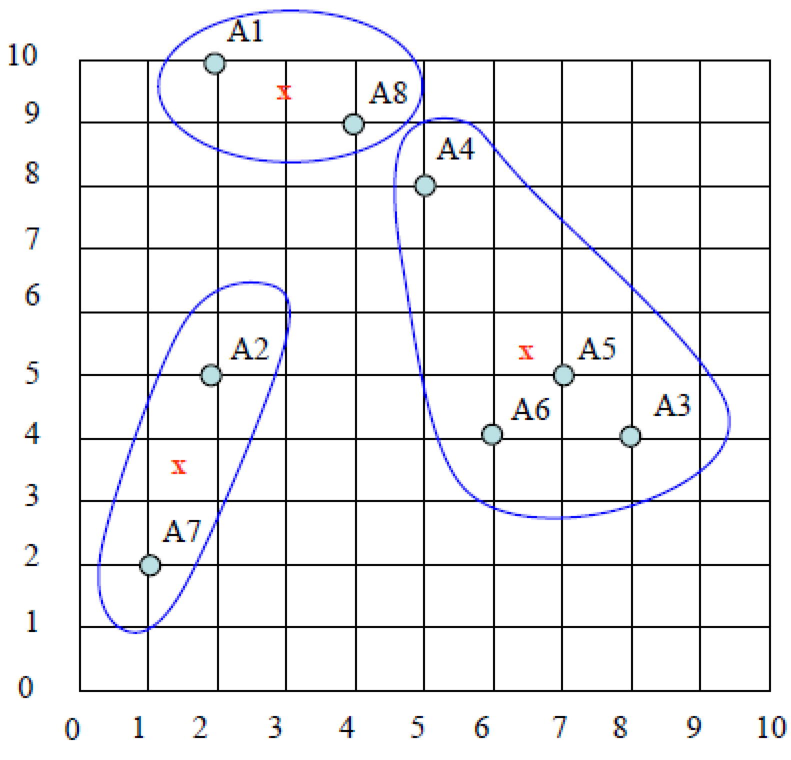
C1= (2, 10), C2= ((8+5+7+6+4)/5, (4+8+5+4+9)/5) = (6, 6), C3= ((2+1)/2, (5+2)/2) = (1.5, 3.5)



After the 2nd epoch the results would be:

1: {A1, A8}, 2: {A3, A4, A5, A6}, 3: {A2, A7}

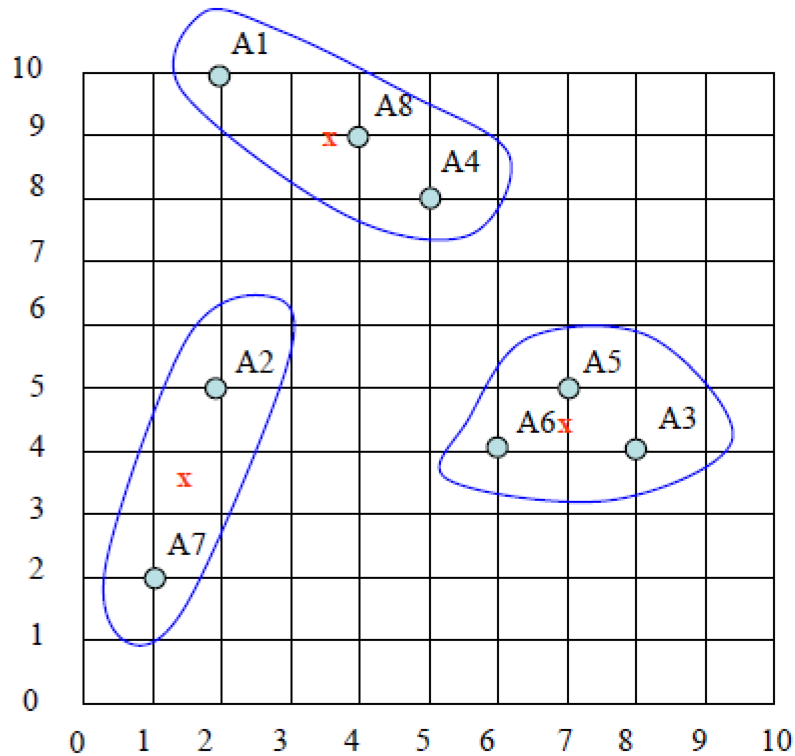
with centers C1=(3, 9.5), C2=(6.5, 5.25) and C3=(1.5, 3.5).



After the 3rd epoch, the results would be:

1: {A1, A4, A8}, 2: {A3, A5, A6}, 3: {A2, A7}

with centers C1=(3.66, 9), C2=(7, 4.33) and C3=(1.5, 3.5).



Exercise 2. *Nearest Neighbor clustering*

Use the Nearest Neighbor clustering algorithm and Euclidean distance to cluster the examples from the previous exercise: A1=(2,10), A2=(2,5), A3=(8,4), A4=(5,8), A5=(7,5), A6=(6,4), A7=(1,2), A8=(4,9). Suppose that the threshold t is 4.

A1 is placed in a cluster by itself, so we have K1={A1}.

We then look at A2 if it should be added to K1 or be placed in a new cluster.
d(A1,A2)= $\sqrt{25} = 5 > t$ ➔ K2={A2}

A3: we compare the distances from A3 to A1 and A2.
A3 is closer to A2 and d(A3,A2)= $\sqrt{36} > t$ ➔ K3={A3}

A4: We compare the distances from A4 to A1, A2 and A3.
A1 is the closest object and d(A4,A1)= $\sqrt{13} < t$ ➔ K1={A1, A4}

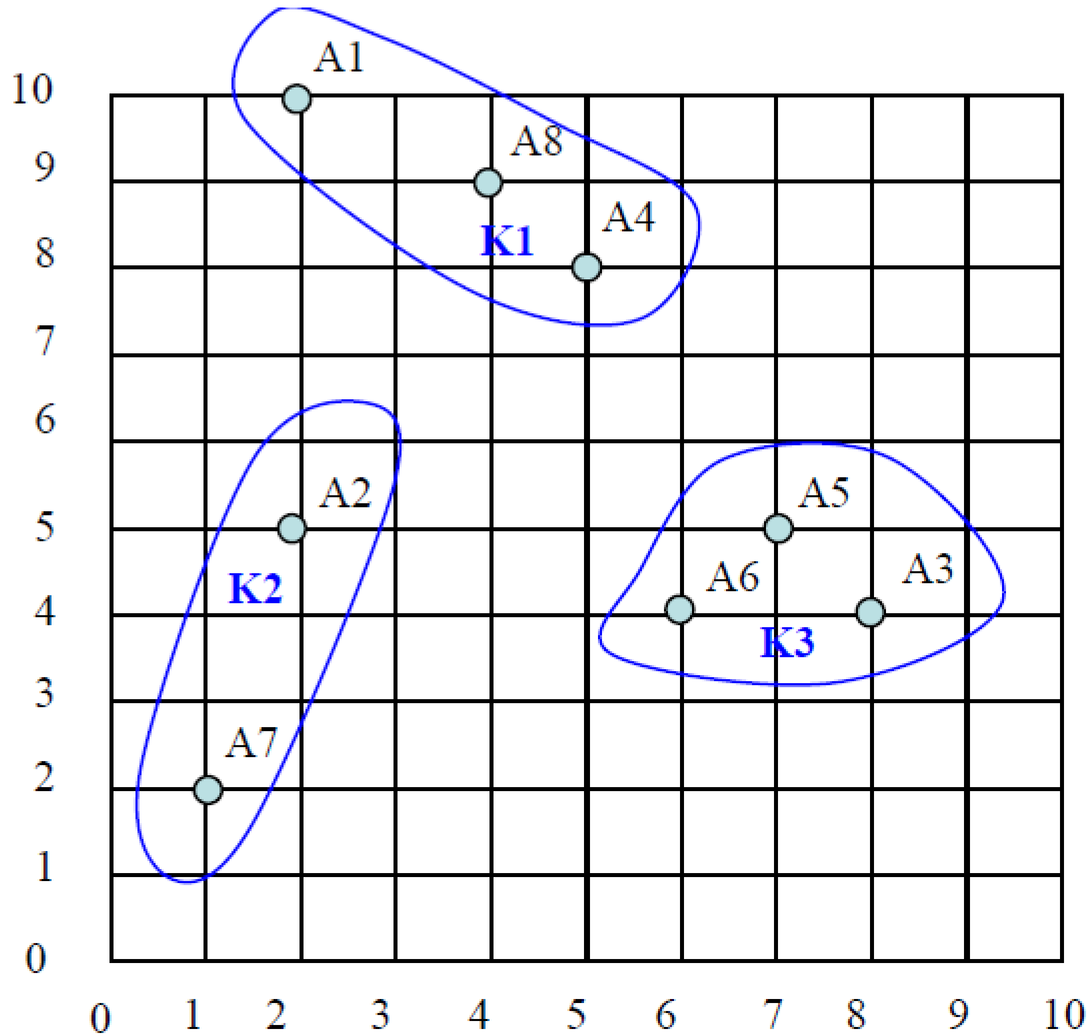
A5: We compare the distances from A5 to A1, A2, A3 and A4.
A3 is the closest object and d(A5,A3)= $\sqrt{2} < t$ ➔ K3={A3, A5}

A6: We compare the distances from A6 to A1, A2, A3, A4 and A5.
A3 is the closest object and d(A6,A3)= $\sqrt{2} < t$ ➔ K3={A3, A5, A6}

A7: We compare the distances from A7 to A1, A2, A3, A4, A5, and A6.
A2 is the closest object and d(A7,A2)= $\sqrt{10} < t$ ➔ K2={A2, A7)

A8: We compare the distances from A8 to A1, A2, A3, A4, A5, A6 and A7.
A4 is the closest object and d(A8,A4)= $\sqrt{2} < t$ ➔ K1={A1, A4, A8)

Thus: K1={A1, A4, A8), K2={A2, A7), K3={A3, A5, A6)


A1
A8
A4
K1
A2
K2
A7
A5
A6
A3
K3
10
9
8
7
6
5
4
3
2
1
0
0
1
2
3
4
5
6
7
8
9
10

Advantages and Disadvantages

Advantages

- It is very easy to understand and implement.
- If we have large number of variables then, K-means would be faster than Hierarchical clustering.
- On re-computation of centroids, an instance can change the cluster.
- Tighter clusters are formed with K-means as compared to Hierarchical clustering.

Disadvantages

- It is a bit difficult to predict the number of clusters i.e. the value of k.
- Output is strongly impacted by initial inputs like number of clusters (value of k).
- Order of data will have strong impact on the final output.
- It is very sensitive to rescaling. If we will rescale our data by means of normalization or standardization, then the output will completely change final output.

Applications

- Market segmentation
- Document Clustering
- Image segmentation
- Customer segmentation
- Analyzing the trend on dynamic data

Apriori Algorithm

What Is An Itemset?

- A set of items together is called an itemset. If any itemset has k-items it is called a k-itemset. An itemset consists of two or more items. An itemset that occurs frequently is called a frequent itemset. **Thus frequent itemset mining is a data mining technique to identify the items that often occur together.**
- **For Example**, Bread and butter, Laptop and Antivirus software, etc.

What Is A Frequent Itemset?

- A set of items is called frequent if it satisfies a minimum threshold value for support and confidence.

Apriori Algorithm

Minimum Support :2

Tid	Items Bought
1	Milk, Tea, Cake
2	Eggs, Tea, Cold Drink
3	Milk, Eggs, Tea, Cold Drink
4	Eggs, Cold Drink
5	Juice

Items Bought	Support
Milk	2
Eggs	3
Tea	3
Cold Drink	3
Juice	1
Cake	1

Items Bought	Support
Milk	2
Eggs	3
Tea	3
Cold Drink	3

Items Bought
Milk, Eggs
Milk, Tea
Milk, Cold Drink
Eggs, Tea
Eggs, Cold Drink
Tea, Cold Drink

Items Bought	Support
Milk, Eggs	1
Milk, Tea	2
Milk, Cold Drink	1
Eggs, Tea	2
Eggs, Cold Drink	3
Tea, Cold Drink	2

Items Bought	Support
Milk, Tea	2
Eggs, Tea	2
Eggs, Cold Drink	3
Tea, Cold Drink	2

Items Bought	Support
Eggs, Tea, Cold Drink	2

There is only one itemset with minimum support 2.
So only one itemset is frequent

Minimum Support :3

Tid	Items Bought
1	Milk, Tea, Cake
2	Eggs, Tea, Cold Drink
3	Milk, Eggs, Tea, Cold Drink
4	Eggs, Cold Drink
5	Juice

Items Bought	Support
Milk	2
Eggs	3
Tea	3
Cold Drink	3
Juice	1
Cake	1

Items Bought	Support
Eggs	3
Tea	3
Cold Drink	3

Items Bought
Eggs, Tea
Eggs, Cold Drink
Tea, Cold Drink

Items Bought	Support
Eggs, Tea	2
Eggs, Cold Drink	3
Tea, Cold Drink	2

Items Bought	Support
Eggs, Cold Drink	3

There is no itemset with minimum support 3, so there is no frequent itemset because there are 0 itemset that have minimum support 3 *but One itemset is frequent itemset and that is Eggs, ColdDrink.*

Advantages of Apriori Algorithm

- It is used to calculate large itemsets.
- Simple to understand and apply.

Disadvantages of Apriori Algorithms

- Apriori algorithm is an expensive method to find support since the calculation has to pass through the whole database.
- Sometimes, you need a huge number of candidate rules, so it becomes computationally more expensive.