Tutorial exercises
Clustering – K-means, Nearest Neighbor and Hierarchical.

<u>Exercise 1.</u> K-means clustering

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).
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

Suppose that the initial seeds (centers of each cluster) are A1, A4 and A7. Run the k-means algorithm for 1 epoch only. At the end of this epoch show:
a) The new clusters (i.e. the examples belonging to each cluster)
b) The centers of the new clusters
c) Draw a 10 by 10 space with all the 8 points and show the clusters after the first epoch and the new centroids.
d) How many more iterations are needed to converge? Draw the result for each epoch.

Solution:
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)

epoch1 – start:

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

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

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

A4:
d(A4, seed1)= $\sqrt{13}$
d(A4, seed2)=0 as A4 is seed2
d(A4, seed3)= $\sqrt{52}$ >0
➔ A4 $\in$ cluster2

A5:
d(A5, seed1)= $\sqrt{50}$ = 7.07

A6:
d(A6, seed1)= $\sqrt{52}$ = 7.21

d(A5, seed2)= $\sqrt{13}$ = 3.60 ← smaller
d(A5, seed3)= $\sqrt{45}$ = 6.70
→ A5 $\in$ cluster2

d(A6, seed2)= $\sqrt{17}$ = 4.12 ← smaller
d(A6, seed3)= $\sqrt{29}$ = 5.38
→ 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

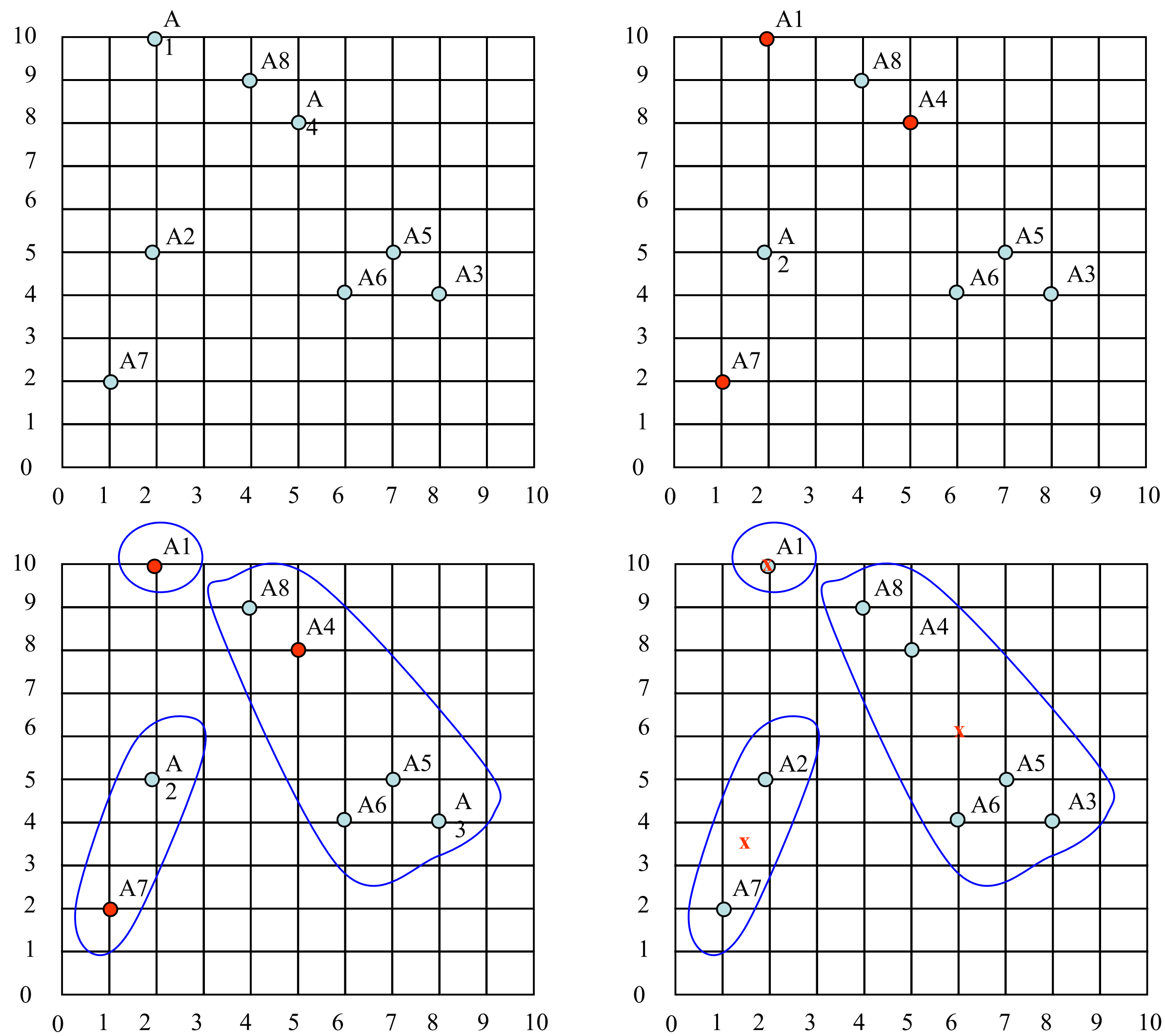
A8:
d(A8, seed1)= $\sqrt{5}$
d(A8, seed2)= $\sqrt{2}$ ← smaller
d(A8, seed3)= $\sqrt{58}$
→ A8 $\in$ cluster2

end of epoch1

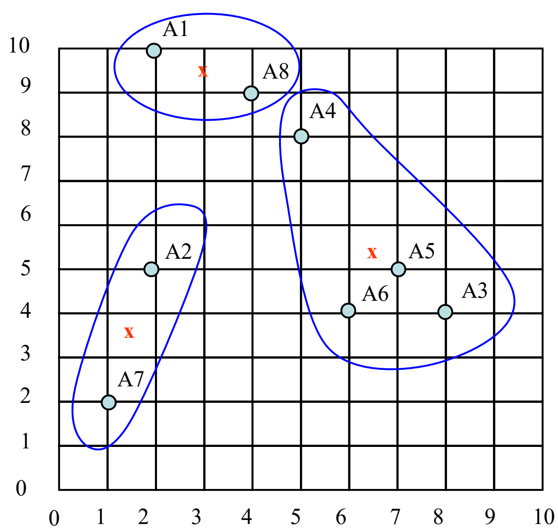
new clusters: 1: {A1}, 2: {A3, A4, A5, A6, A8}, 3: {A2, A7}

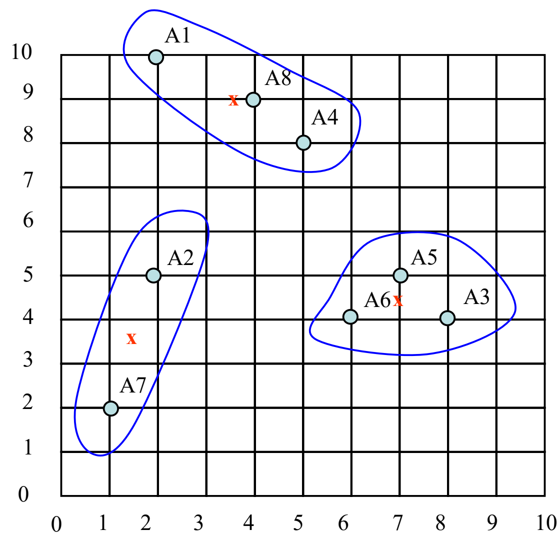
b) centers of the new clusters:
C1= (2, 10), C2= ((8+5+7+6+4)/5, (4+8+5+4+9)/5) = (6, 6), C3= ((2+1)/2, (5+2)/2) = (1.5, 3.5)

c)



d)
We would need two more epochs. 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).





<u>Exercise 2.</u> 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.

Solution:
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}

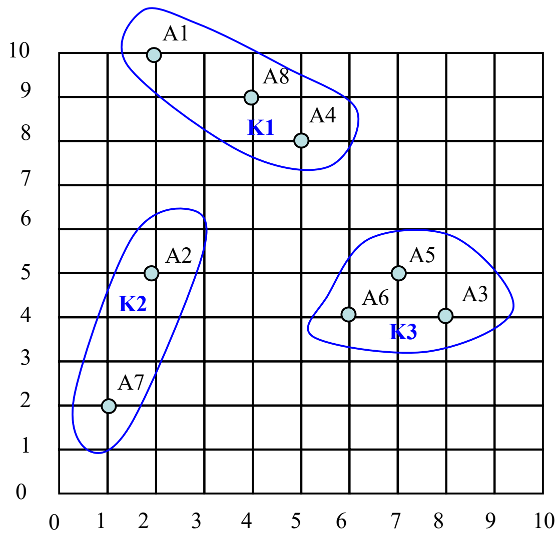
A6: We compare the distances from A6 to A1, A2, A3, A4 and A5.
A3 is the closest object and d(A6,A3)=$\sqrt{2} < t$ ➔ K3={A3, A5, A6}

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)

Yes, it is the same result as with K-means.



<u>Exercise 3.</u> Hierarchical clustering

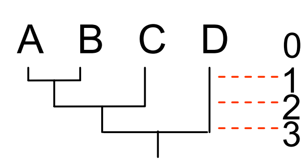
Use single and complete link agglomerative clustering to group the data described by the following distance matrix. Show the dendrograms.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0

Solution:
Agglomerative ➔ initially every point is a cluster of its own and we merge cluster until we end-up with one unique cluster containing all points.

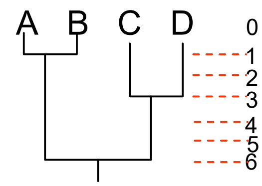
a) single link: distance between two clusters is the shortest distance between a pair of elements from the two clusters.

d	**k**	**K**	**Comments**
0	4	{A}, {B}, {C}, {D}	We start with each point = cluster
1	3	{A, B}, {C}, {D}	Merge {A} and {B} since A & B are the closest: d(A, B)=1
2	2	{A, B, C}, {D}	Merge {A, B} and {C} since B & C are the closest: d(B, C)=2
3	1	{A, B, C, D}	Merge D



b) complete link: distance between two clusters is the longest distance between a pair of elements from the two clusters.

d	k	K	Comments
0	4	{A}, {B}, {C}, {D}	We start with each point = cluster
1	3	{A, B}, {C}, {D}	d(A,B)=1<=1 ➔ merge {A} and {B}
2	3	{A, B}, {C}, {D}	d(A,C)=4>2 so we can't merge C with {A,B} d(A,D)=5>2 and d(B,D)=6>2 so we can't merge D with {A, B} d(C,D)=3>2 so we can't merge C and D
3	2	{A, B}, {C, D}	- d(A,C)=4>3 so we can't merge C with {A,B} - d(A,D)=5>3 and d(B,D)=6>3 so we can't merge D with {A, B} - d(C,D)=3 <=3 so merge C and D
4	2	{A, B}, {C, D}	{C,D} cannot be merged with {A, B} as d(A,D)= 5 >4 (and also d(B,D)= 6 >4) although d(A,C)= 4 <= 4, d(B,C)= 2<=4)
5	2	{A, B}, {C, D}	{C,D} cannot be merged with {A, B} as d(B,D)= 6 > 5
6	1	{A, B, C, D}	{C, D} can be merged with {A, B} since d(B,D)= 6 <= 6, d(A,D)= 5 <= 6, d(A,C)= 4 <= 6, d(B,C)= 2 <= 6



Exercise 4: **Hierarchical clustering <u>(to be done at your own time, not in class)</u>**

Use single-link, complete-link, average-link agglomerative clustering as well as medoid and centroid to cluster the following 8 examples:

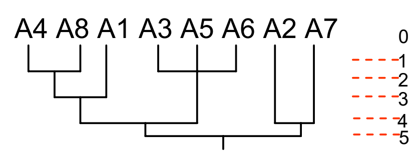
A1=(2,10), A2=(2,5), A3=(8,4), A4=(5,8), A5=(7,5), A6=(6,4), A7=(1,2), A8=(4,9).

The distance matrix is the same as the one in Exercise 1. Show the dendrograms.

Solution:

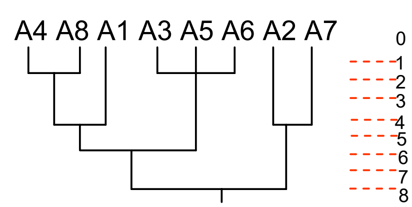
Single Link:

d	k	K
0	8	{A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}
1	8	{A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}
2	5	{A4, A8}, {A1}, {A3, A5, A6}, {A2}, {A7}
3	4	{A4, A8, A1}, {A3, A5, A6}, {A2}, {A7}
4	2	{A1, A3, A4, A5, A6, A8}, {A2, A7}
5	1	{A1, A3, A4, A5, A6, A8, A2, A7}



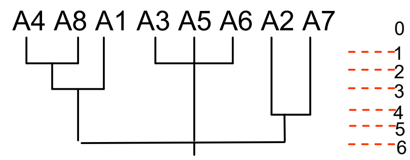
Complete Link

d	k	K
0	8	{A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}
1	8	{A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}
2	5	{A4, A8}, {A1}, {A3, A5, A6}, {A2}, {A7}
3	5	{A4, A8}, {A1}, {A3, A5, A6}, {A2}, {A7}
4	3	{A4, A8, A1}, {A3, A5, A6}, {A2, A7}
5	3	{A4, A8, A1}, {A3, A5, A6}, {A2, A7}
6	2	{A4, A8, A1, A3, A5, A6}, {A2, A7}
7	2	{A4, A8, A1, A3, A5, A6}, {A2, A7}
8	1	{A4, A8, A1, A3, A5, A6, A2, A7}



Average Link

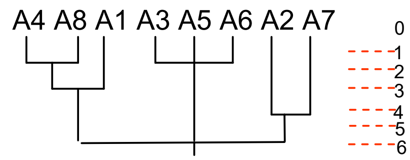
d	k	K
0	8	{A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}
1	8	{A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}
2	5	{A4, A8}, {A1}, {A3, A5, A6}, {A2}, {A7}
3	4	{A4, A8, A1}, {A3, A5, A6}, {A2}, {A7}
4	3	{A4, A8, A1}, {A3, A5, A6}, {A2, A7}
5	3	{A4, A8, A1}, {A3, A5, A6}, {A2, A7}
6	1	{A4, A8, A1, A3, A5, A6, A2, A7}



Average distance from {A3, A5, A6} to {A1, A4, A8} is 5.53 and is 5.75 to {A2, A7}

Centroid

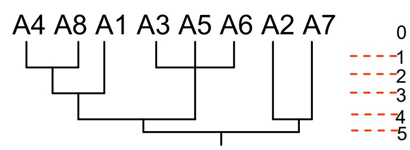
D	k	K
0	8	{A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}
1	8	{A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}
2	5	{A4, A8}, {A1}, {A3, A5, A6}, {A2}, {A7}
3	5	{A4, A8}, {A1}, {A3, A5, A6}, {A2}, {A7}
4	3	{A4, A8, A1}, {A3, A5, A6}, {A2, A7}
5	3	{A4, A8, A1}, {A3, A5, A6}, {A2, A7}
6	1	{A4, A8, A1, A3, A5, A6, A2, A7}



Centroid of {A4, A8} is B=(4.5, 8.5) and centroid of {A3, A5, A6} is C=(7, 4.33)
distance(A1, B) = 2.91 Centroid of {A1, A4, A8} is D=(3.66, 9) and of {A2, A7} is E=(1.5, 3.5)
distance(D,C)= 5.74 distance(D,E)= 5.90

Medoid
This is not deterministic. It can be different depending upon which medoid in a cluster we chose.

d	k	K
0	8	{A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}
1	8	{A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}
2	5	{A4, A8}, {A1}, {A3, A5, A6}, {A2}, {A7}
3	4	{A4, A8, A1}, {A3, A5, A6}, {A2}, {A7}
4	2	{A1, A3, A4, A5, A6, A8}, {A2, A7}
5	1	{A1, A3, A4, A5, A6, A8, A2, A7}



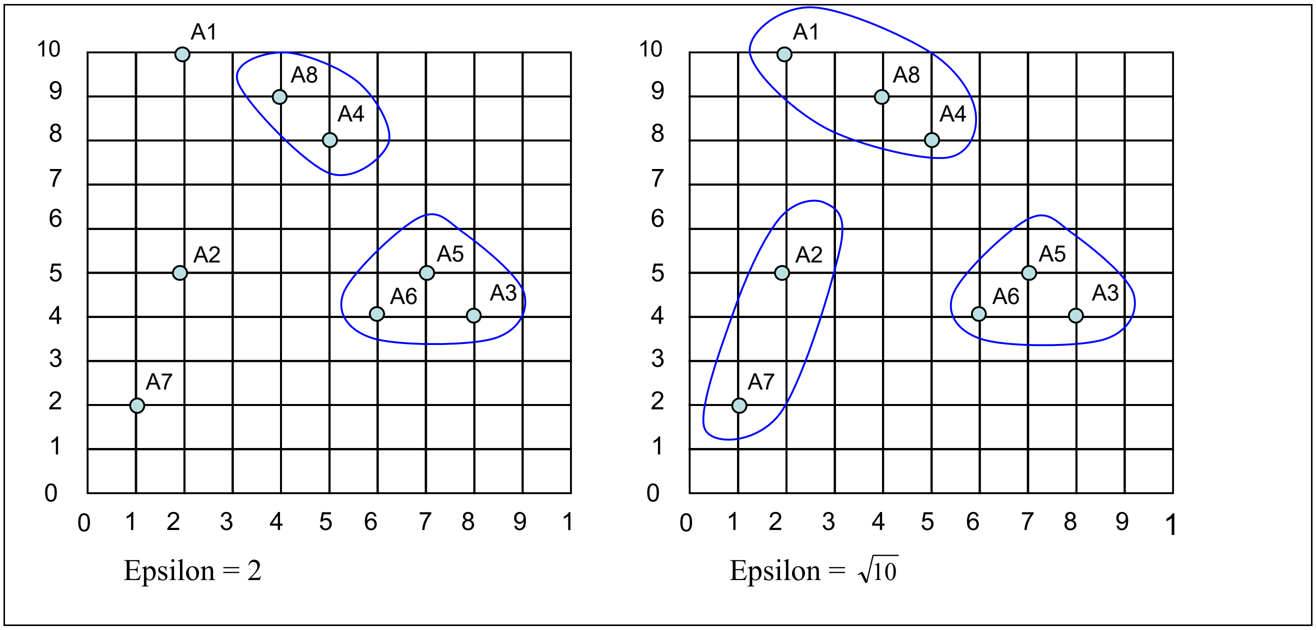
***Exercise 5:* DBScan**

If Epsilon is 2 and minpoint is 2, what are the clusters that DBScan would discover with the following 8 examples: A1=(2,10), A2=(2,5), A3=(8,4), A4=(5,8), A5=(7,5), A6=(6,4), A7=(1,2), A8=(4,9).
The distance matrix is the same as the one in Exercise 1. Draw the 10 by 10 space and illustrate the discovered clusters. What if Epsilon is increased to $\sqrt{10}$?

Solution:
What is the Epsilon neighborhood of each point?
N_2(A1)={}; N_2(A2)={}; N_2(A3)={A5, A6}; N_2(A4)={A8}; N_2(A5)={A3, A6};
N_2(A6)={A3, A5}; N_2(A7)={}; N_2(A8)={A4}

So A1, A2, and A7 are outliers, while we have two clusters C1={A4, A8} and C2={A3, A5, A6}

If Epsilon is $\sqrt{10}$ then the neighborhood of some points will increase:
A1 would join the cluster C1 and A2 would joint with A7 to form cluster C3={A2, A7}.


A1
A8
A4
A2
A5
A6
A3
A7
Epsilon = 2
Epsilon = $\sqrt{10}$

Lecture on

Unsupervised Learning

Introduction

- Unsupervised learning is a type of machine learning in which models are trained using unlabeled dataset and are allowed to act on that data without any supervision.

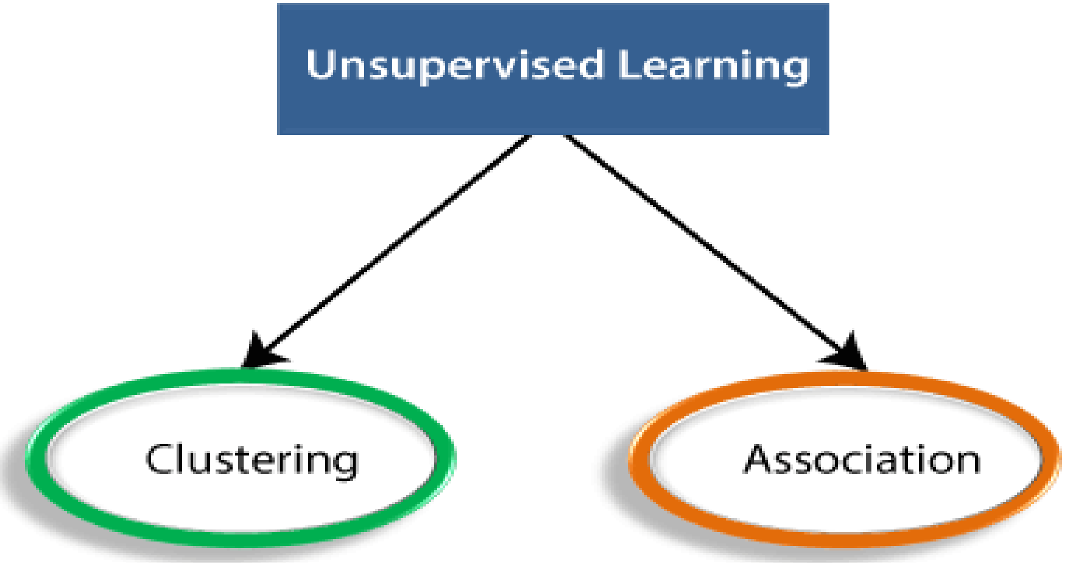
- Instead, models itself find the hidden patterns and insights from the given data.

- It can be compared to learning which takes place in the human brain while learning new things.

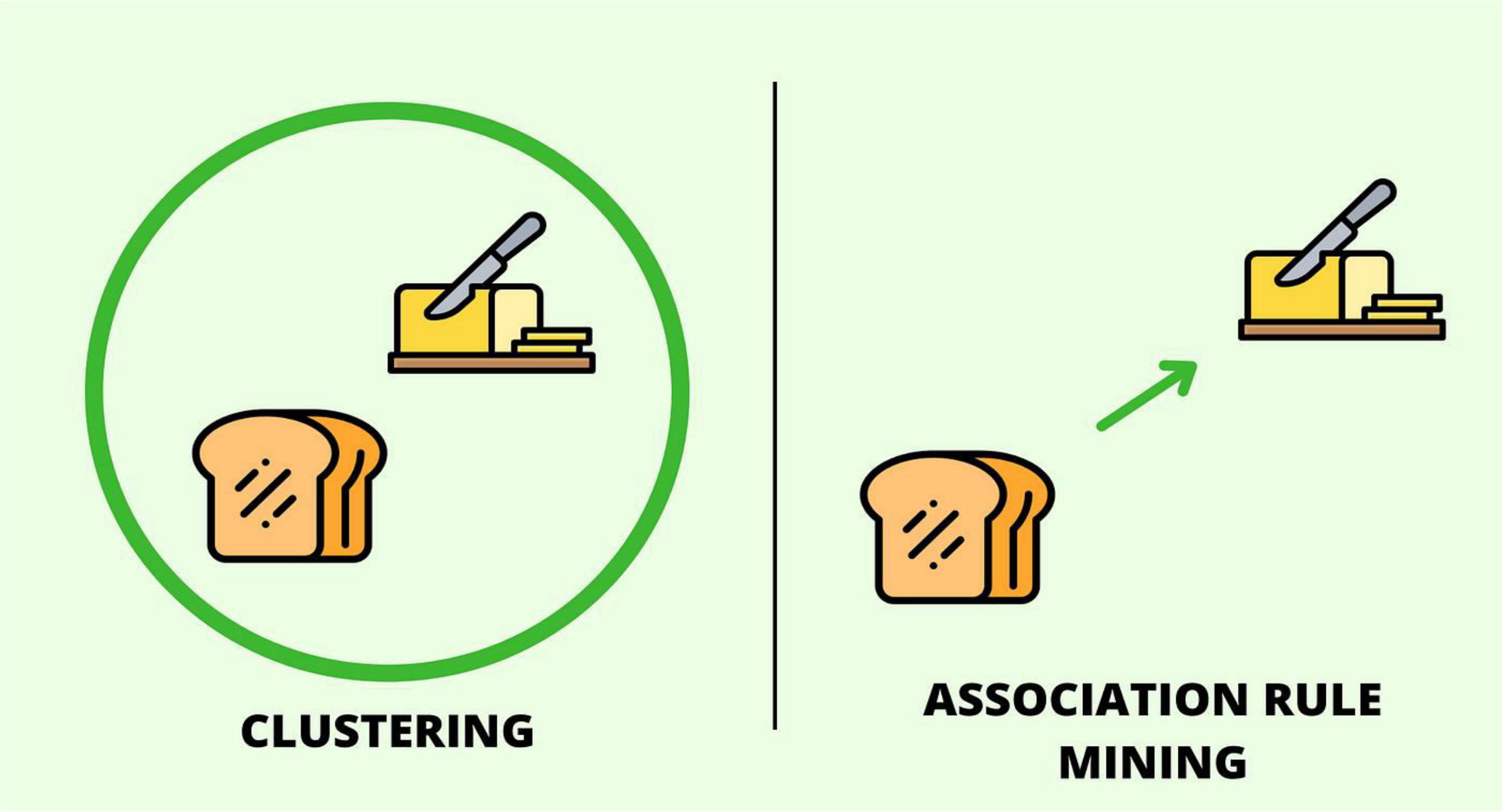
Why to use Unsupervised Learning

- Unsupervised learning is helpful for finding useful insights from the data.
- Unsupervised learning is much similar as a human learns to think by their own experiences, which makes it closer to the real AI.
- Unsupervised learning works on unlabeled and uncategorized data which make unsupervised learning more important.
- In real-world, we do not always have input data with the corresponding output so to solve such cases, we need unsupervised learning.

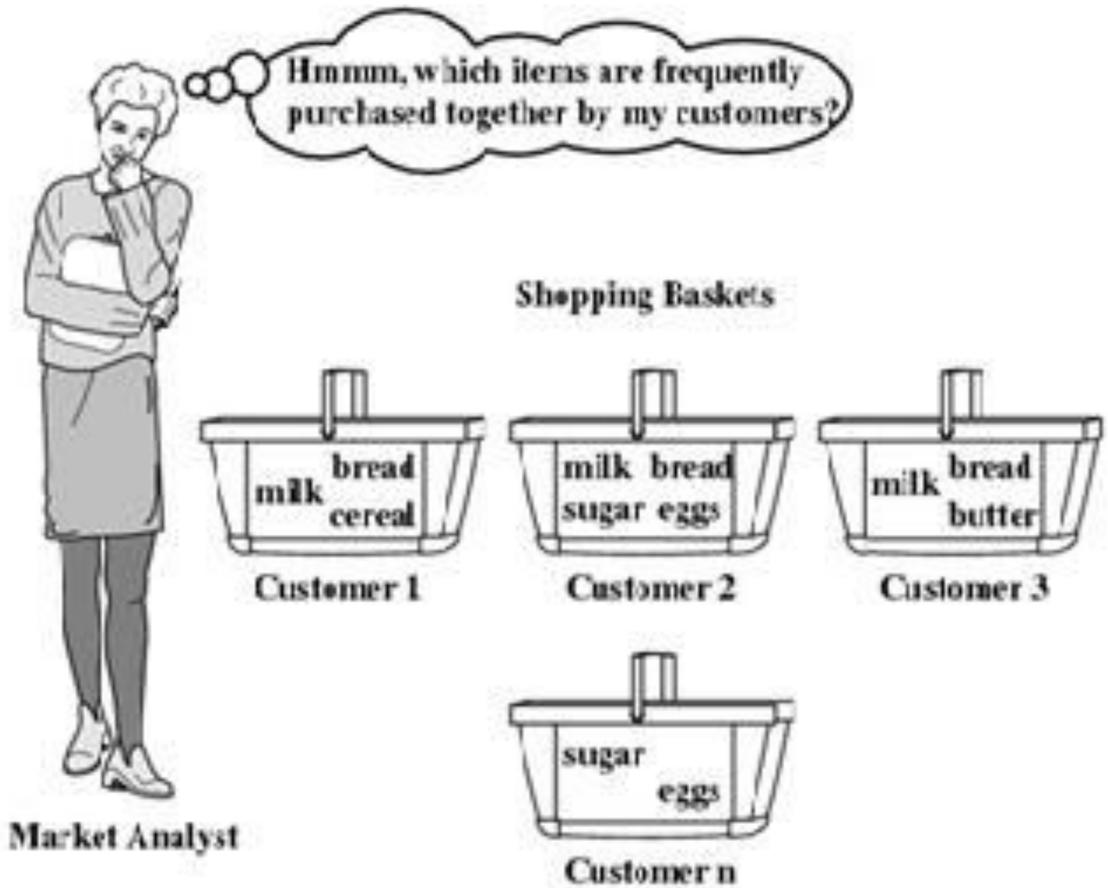
Types



Contd..



Market Basket Analysis



Contd..

- **Clustering:**
  - Clustering is a method of grouping the objects into clusters such that objects with most similarities remains into a group and has less or no similarities with the objects of another group.

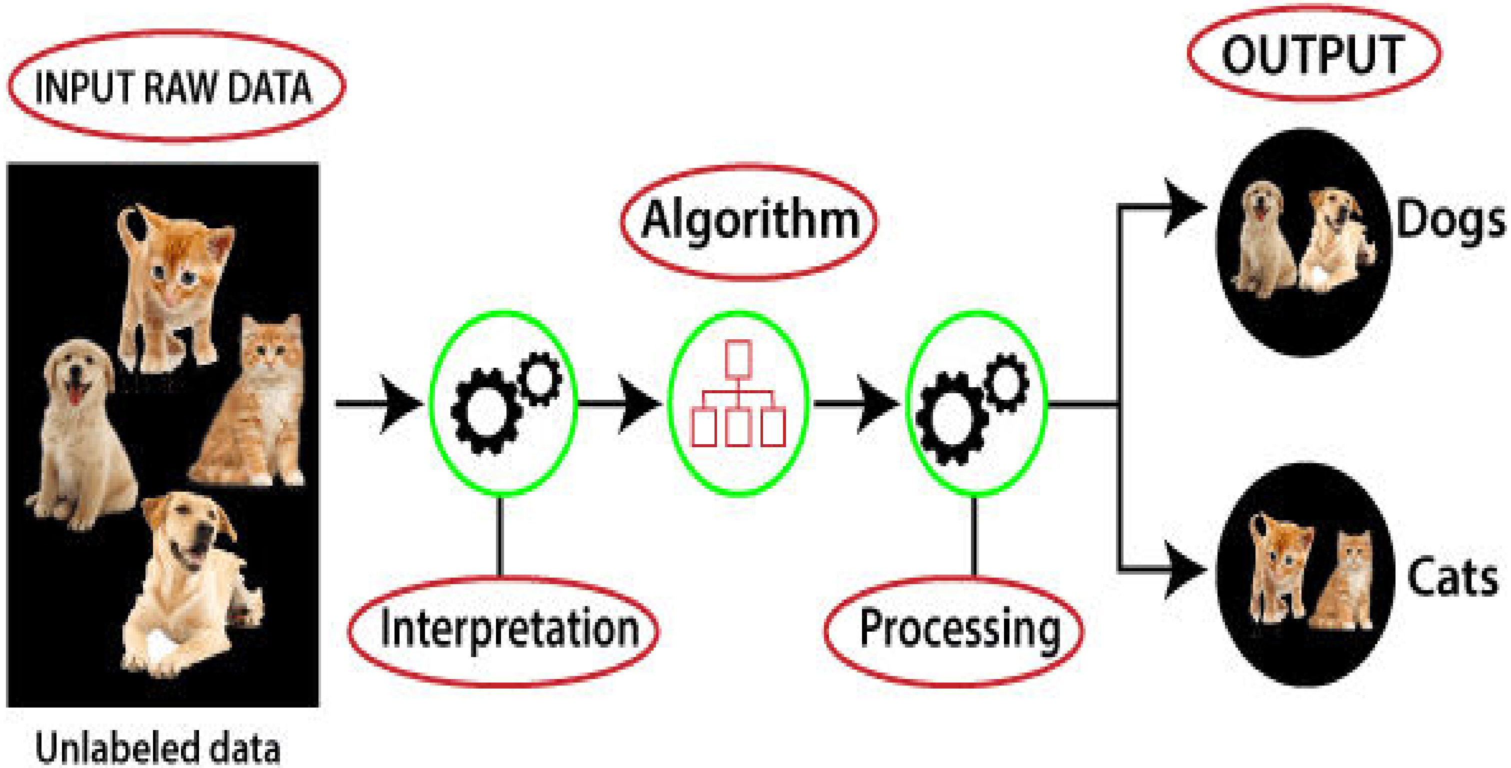
  - Cluster analysis finds the commonalities between the data objects and categorizes them as per the presence and absence of those commonalities.

- **Association:**
  - An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database

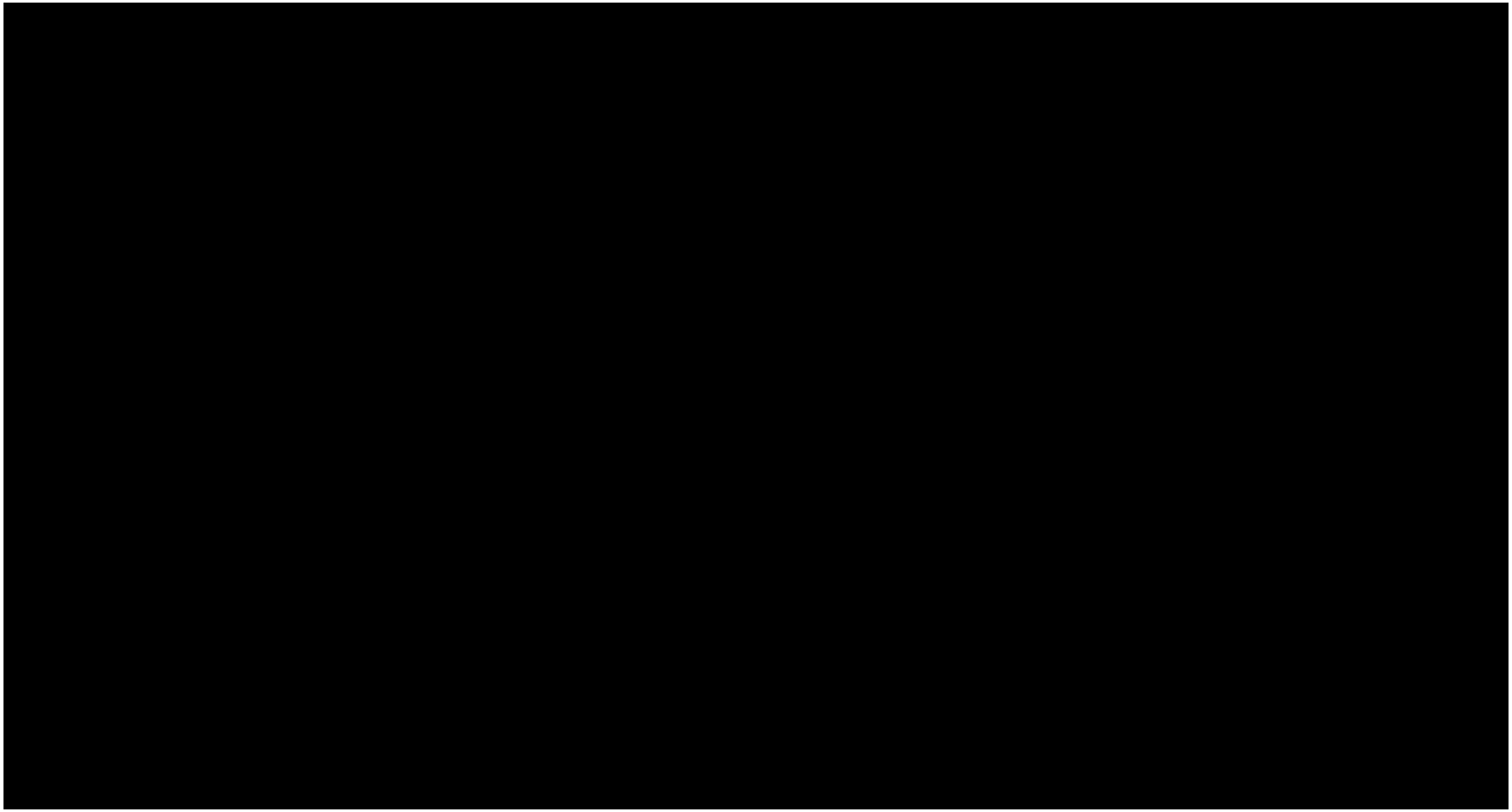
  - It determines the set of items that occurs together in the dataset. Association rule makes marketing strategy more effective.

  - Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item. A typical example of Association rule is Market Basket Analysis.

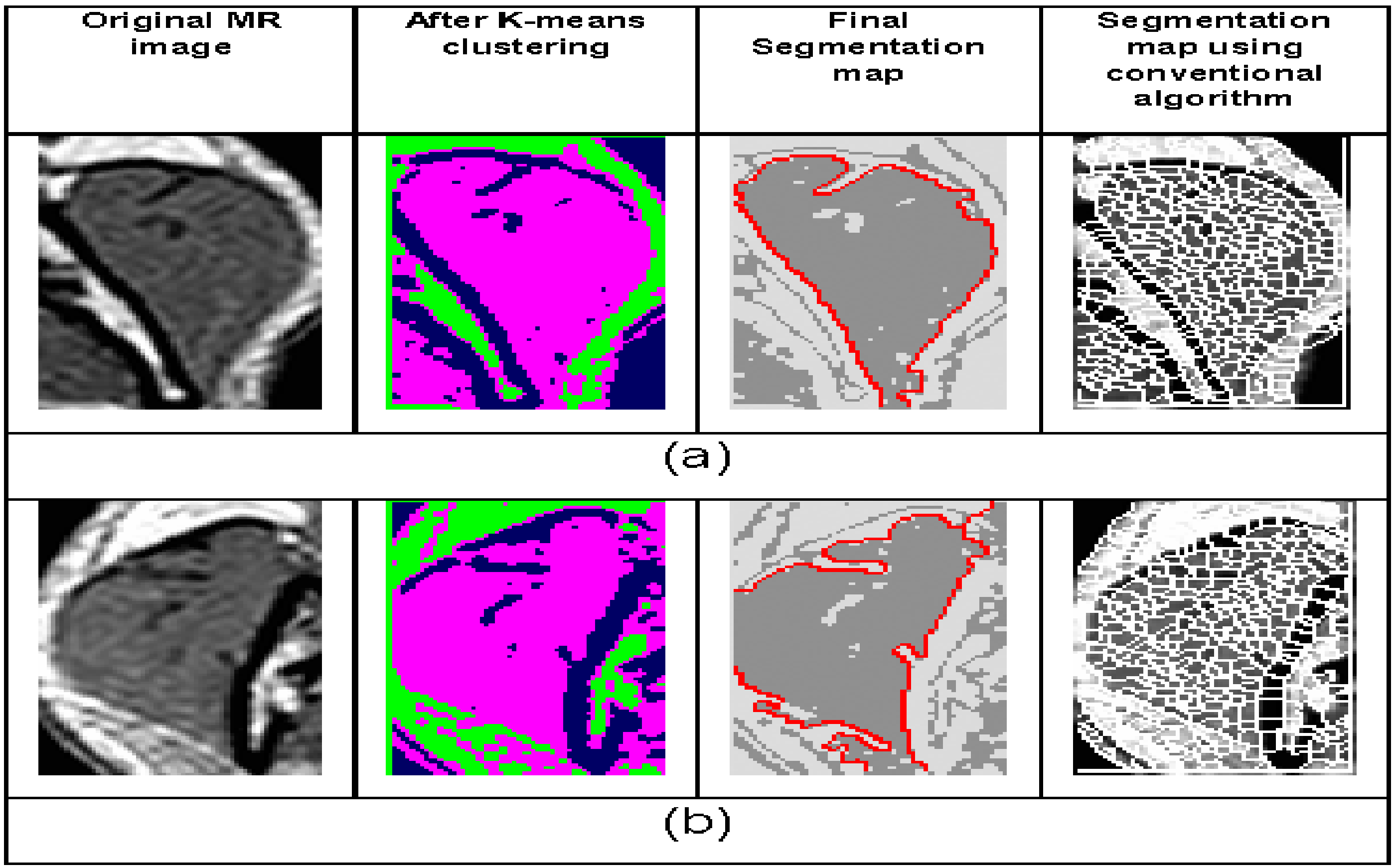
Working


INPUT RAW DATA
Unlabeled data
Interpretation
Algorithm
Processing
OUTPUT
Dogs
Cats

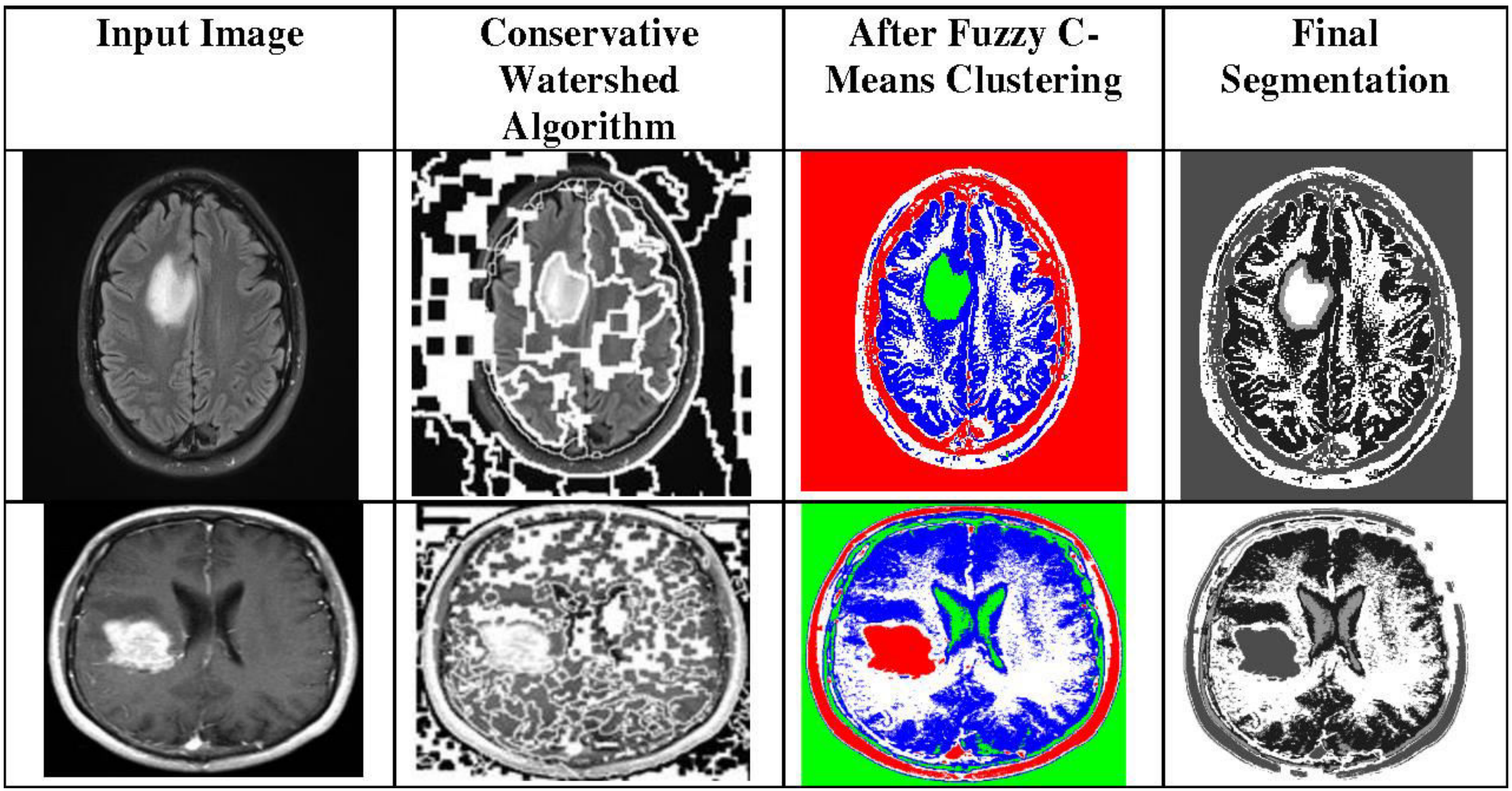
Clustering



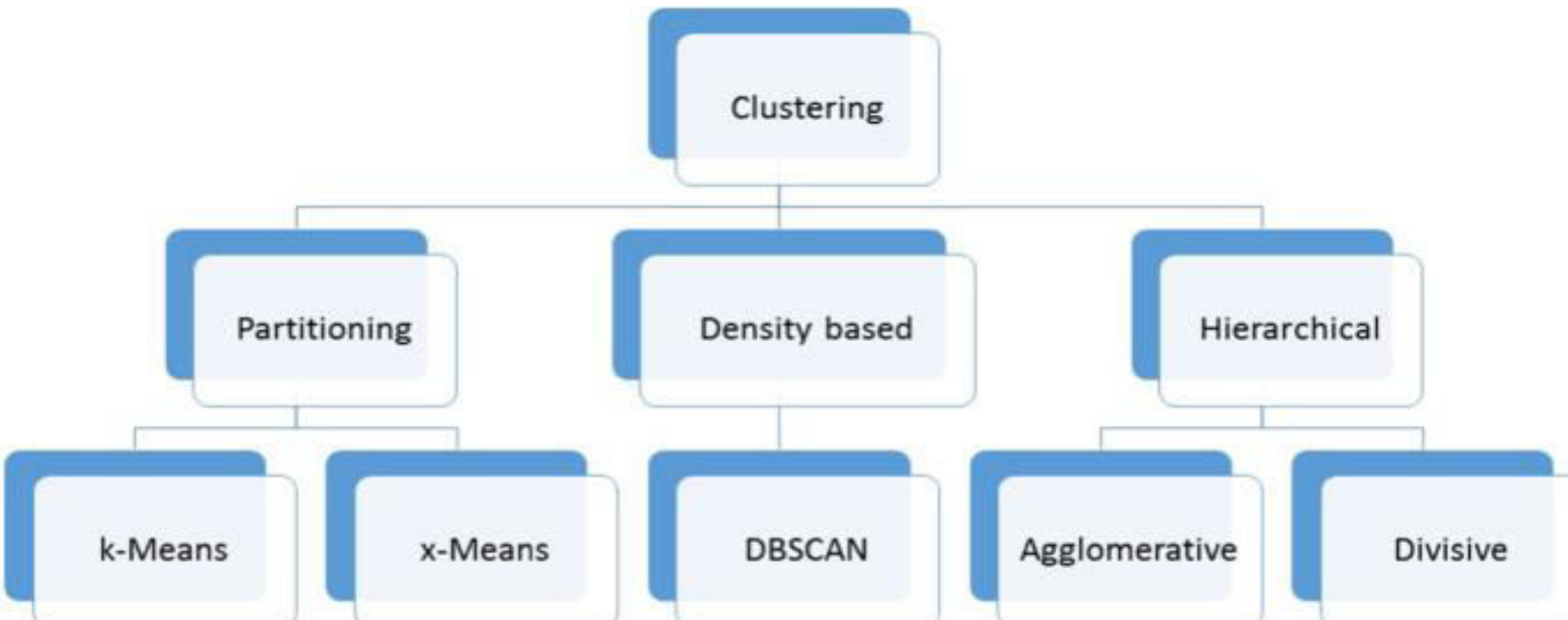
Eg 1: Medical Image Segmentation



Eg 2: Medical Image Segmentation



Contd..



Clustering
Partitioning
Density based
Hierarchical
k-Means
x-Means
DBSCAN
Agglomerative
Divisive

K- Means clustering

- K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters.

- Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

- It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.

K- means algorithm

Step- 1: Select the number K to decide the number of clusters.

Step-2: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-3: Calculate the variance and place a new centroid of each cluster.

Step-4: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

Step-5: If any reassignment occurs, then go to step-3 else go to FINISH.

Step-6: The model is ready.

K-means problem

- Consider the following data points, and implement k- means clustering algorithms?

(2,4) (2, 6), (5,6), (4, 7), (8,3), (6, 6), (5, 2), (5, 7), (6, 3), (4, 4)

- The initial centroids are (1, 5), (4, 1), (8, 4)

Lecture on

Unsupervised Learning: Hierarchical Clustering

Shesh Narayan Sahu, CSE

Issue with k- means

- Choosing value of k manually.
- Clustering data of varying sizes and density.
- Clustering outliers.
- Scaling with number of dimensions.
- Traditional K-means clustering algorithms have the drawback of getting stuck at local optima

Introduction

- The hierarchical clustering Technique is one of the popular Clustering techniques in Machine Learning.

- So, as we have seen in the K-means clustering that there are some challenges with this algorithm, which are a predetermined number of clusters, and it always tries to create the clusters of the same size.

- To solve these challenges, we can opt for the hierarchical clustering algorithm because, in this algorithm, we don't need to have knowledge about the predefined number of clusters.

Types of hierarchical clustering

- The hierarchical clustering technique has two approaches:

    1. Agglomerative: Agglomerative is a bottom-up approach, in which the algorithm starts with taking all data points as single clusters and merging them until one cluster is left.

    2. Divisive: Divisive algorithm is the reverse of the agglomerative algorithm as it is a top-down approach.

Agglomerative Hierarchical clustering

- The agglomerative hierarchical clustering algorithm is a popular example of HCA.
- To group the datasets into clusters, it follows the bottom-up approach.
- It means, this algorithm considers each dataset as a single cluster at the beginning, and then start combining the closest pair of clusters together.
- It does this until all the clusters are merged into a single cluster that contains all the datasets.

Contd..

- In this technique, initially each data point is considered as an individual cluster.

- At each iteration, the similar clusters merge with other clusters until one cluster or K clusters are formed.

- The basic algorithm of Agglomerative is straight forward.

  1. Initially each data point forms a cluster.
  2. Compute the distance matrix between the cluster
  3. Repeat: Merge the two closest clusters and update the distance matrix.
  4. Until only a single cluster remains

Divisive Hierarchical clustering Technique

- Divisive Hierarchical clustering Technique is not much used in the real world.

- In simple words, we can say that the Divisive Hierarchical clustering is exactly the opposite of the Agglomerative Hierarchical clustering.

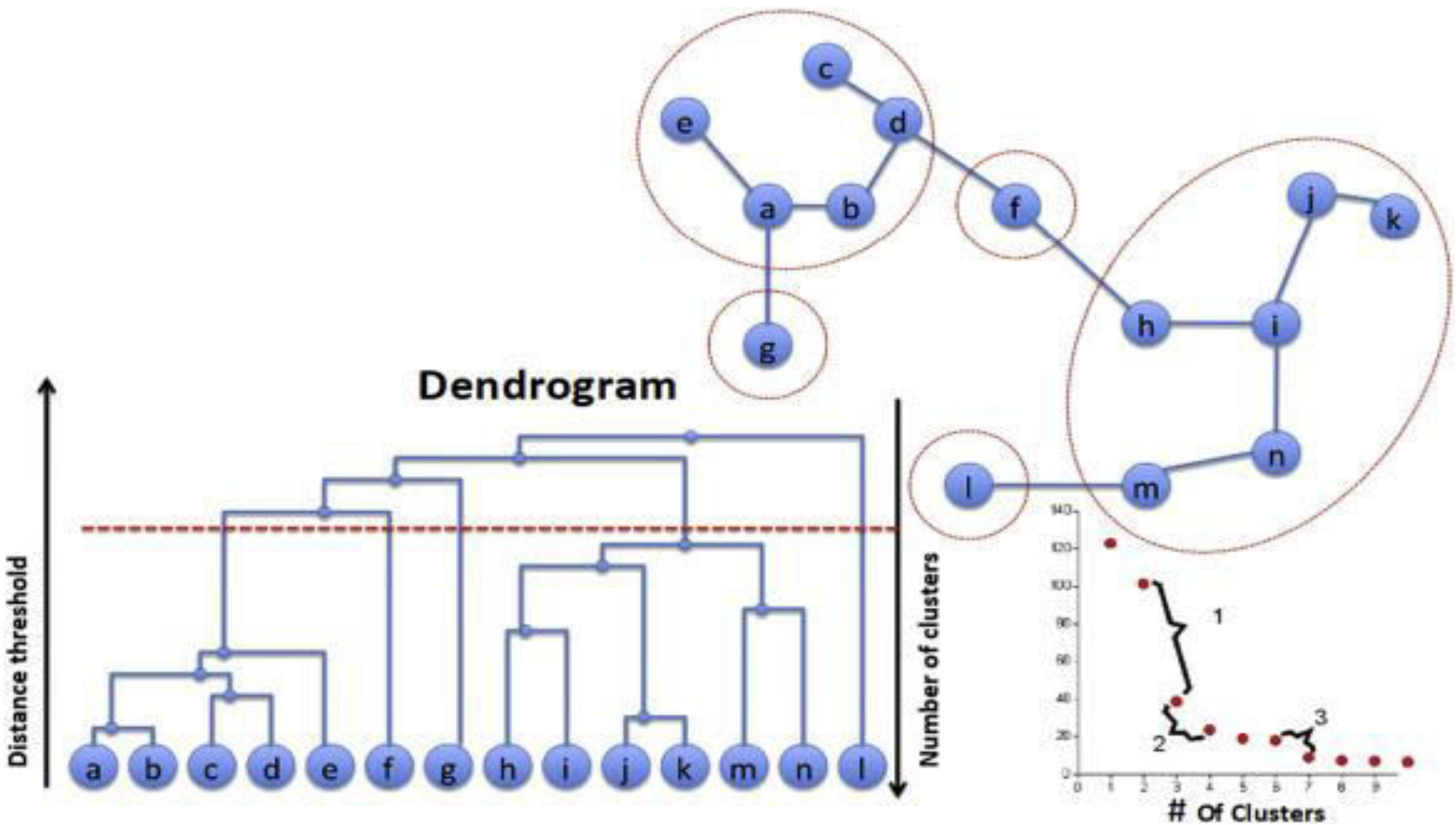
- In Divisive Hierarchical clustering, we consider all the data points as a single cluster and in each iteration, we separate the data points from the cluster which are not similar.

- Each data point which is separated is considered as an individual cluster. In the end, we'll be left with n clusters.

Example: {A, M, N, S, T, V}

Dendrograms

Contd..


Dendrogram
Distance threshold
Number of clusters
a
b
c
d
e
f
g
h
i
j
k
m
n
l
1
2
3
Of Clusters

Closest pair

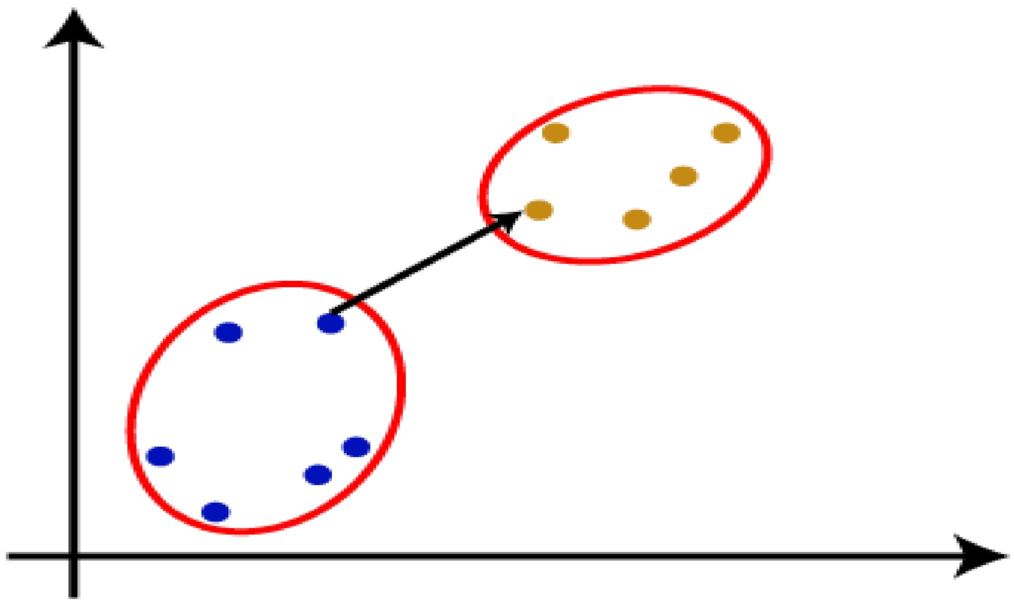
- As we have seen, the closest distance between the two clusters is crucial for the hierarchical clustering.

- There are various ways to calculate the distance between two clusters, and these ways decide the rule for clustering.

- These measures are called Linkage methods. Some of the popular linkage methods are given below-
- 1. Single linkage
- 2. Complete linkage
- 3. Centroid linkage
- 4. Average linkage

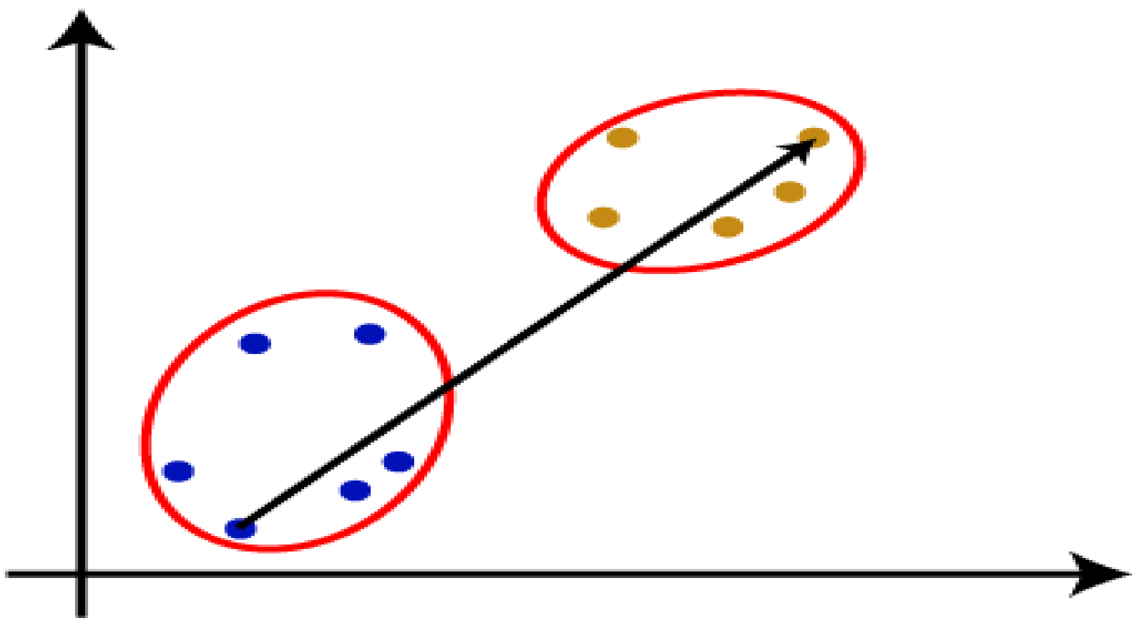
Single linkage

- It is the Shortest Distance between the closest points of the clusters. Consider the below image-



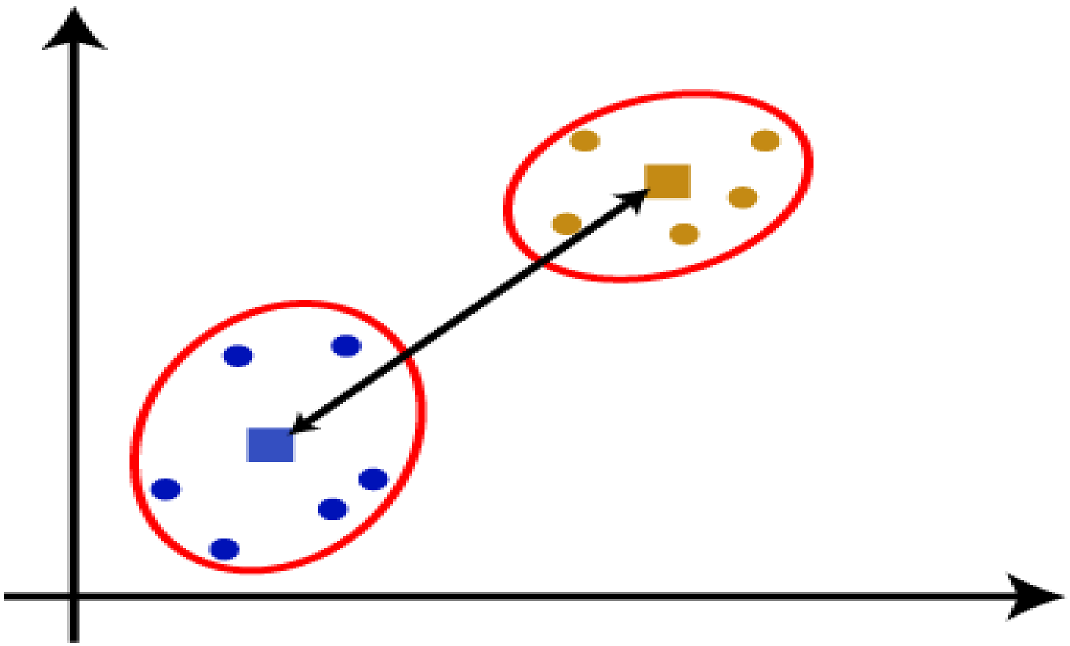
Complete linkage

- It is the farthest distance between the two points of two different clusters. It is one of the popular linkage methods as it forms tighter clusters than single-linkage.



Centroid Linkage:

- It is the linkage method in which the distance between the centroid of the clusters is calculated. Consider the below image:



Average Linkage:

- It is the linkage method in which the distance between each pair of datasets is added up and then divided by the total number of datasets to calculate the average distance between two clusters. It is also one of the most popular linkage methods.

Problem: 1

- Consider the following data points, and implement single linkage hierarchical clustering algorithms?

	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

Similarity Matrix

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0

Contd..

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.22	0.15	0		
P4	0.37	0.20	0.15	0	
P5	0.34	0.14	0.28	0.29	0

Contd..

- To update the distance matrix MIN[dist(P3,P6),P1)]
- MIN(dist(P3,P1), (P6,P1))

  = min[(0.22,0.23)]

  = 0.22
- To update the distance matrix MIN[dist(P3,P6),P2)]
- MIN(dist(P3,P2), (P6,P2))

  = min[(0.15,0.25)]

  = 0.15

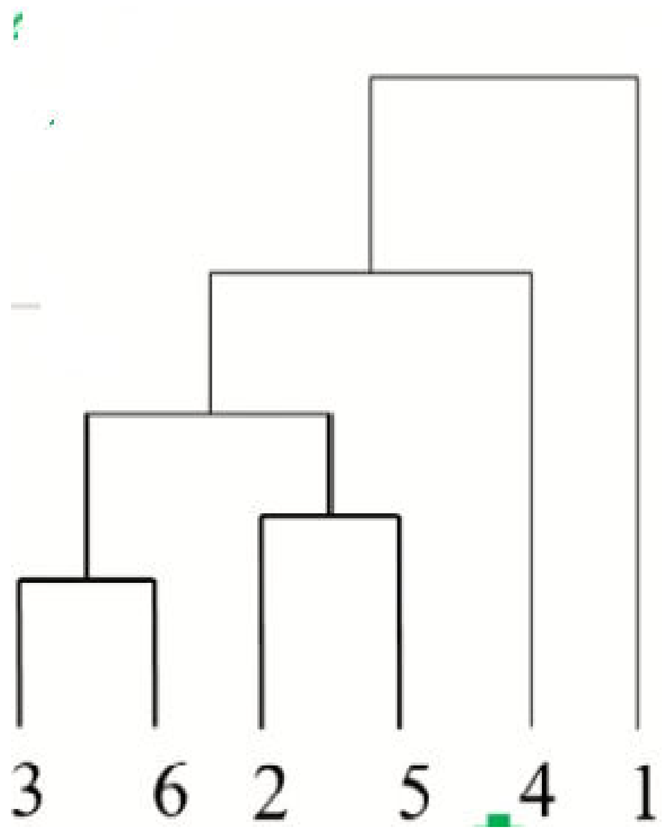
Updated proximity

	P1	P2,P5	P3,P6	P4
P1	0			
P2,P5	0.23	0		
P3,P6	0.22	0.15	0	
P4	0.37	0.20	0.15	0

Contd..

	P1	P2,P5,P3,P6,P4
P1	0	
P2,P5,P3,P6,P4	0.22	0

Dendrogram



Lecture on

Clustering: Density-Based Clustering

Shesh Narayan Sahu, CSE

Introduction: Density based clustering

- To find clusters of arbitrary shape, alternatively, we can model clusters as dense regions in the data space, separated by sparse regions.

- A cluster is defined as maximal set of density connected points.

- This is the main strategy behind density-based clustering methods, which can discover clusters of non spherical shape.

- Today, we will learn the basic techniques of density-based clustering by studying three representative methods, namely:

  1) DBSCAN
  2) OPTICS (Ordering Points to Identify the Clustering Structure)
  3) DENCLUE (DENsity-based CLUstEring)

DBSCAN Clustering

- DBSCAN stands for **Density-Based Spatial Clustering of Applications with Noise.**

- It was proposed by **Martin Ester et al. in 1996**. DBSCAN is a density-based clustering algorithm that works on the assumption that clusters are dense regions in space separated by regions of lower density.

- The DBSCAN algorithm is based on this intuitive notion of “clusters” and “noise”.

- The key idea is that for each point of a cluster, the neighborhood of a given radius has to contain at least a minimum number of points.

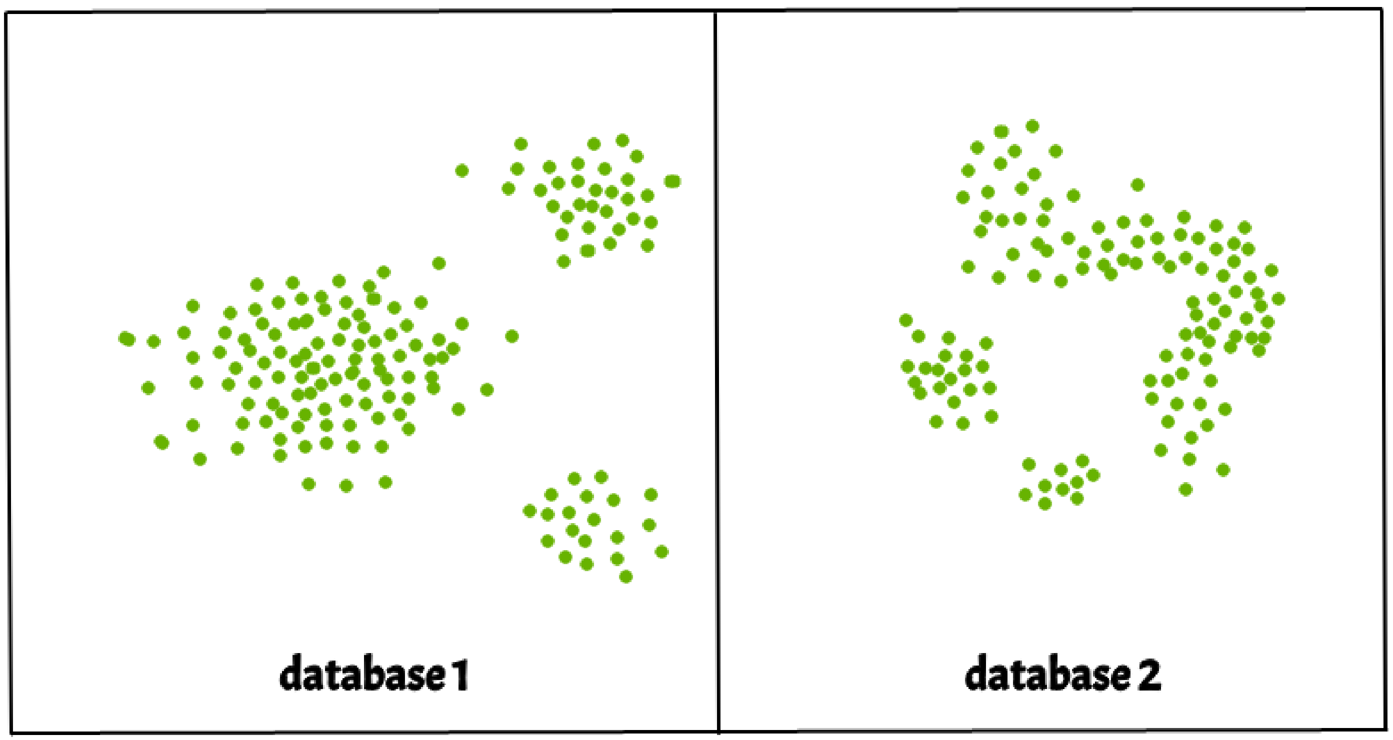
Contd..

- It groups ‘densely grouped’ data points into a single cluster.
- It can identify clusters in large spatial datasets by looking at the local density of the data points.
- The most exciting feature of DBSCAN clustering is that it is robust to outliers.
- It also does not require the number of clusters to be told beforehand, unlike K-Means, where we have to specify the number of centroids.

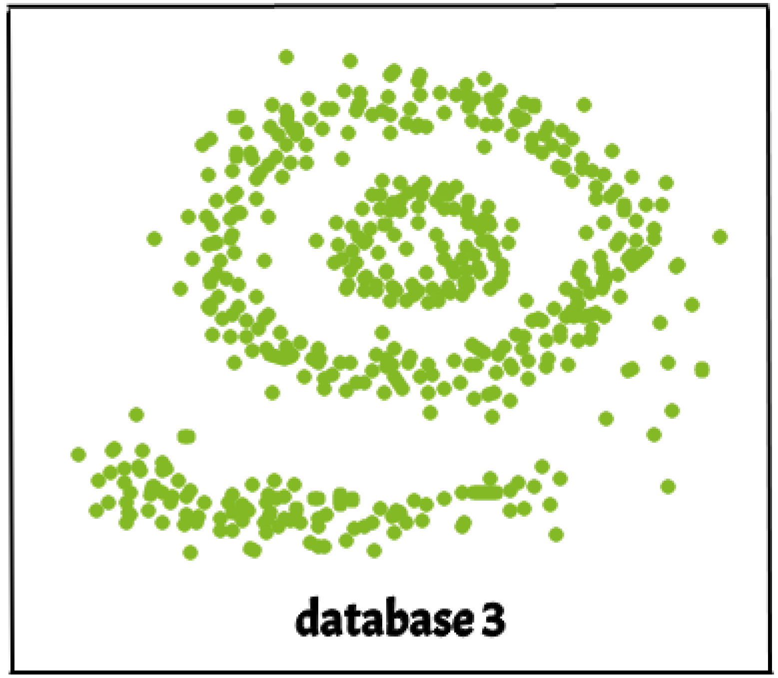
Why DBSCAN Clustering

- K-Means and Hierarchical Clustering both fail in creating clusters of arbitrary shapes.
- The above mentioned clustering algorithms lack this property and make spherical clusters only and are very sensitive to outliers.
- By sensitivity, mean the sphere-shaped clusters made through K-Means can easily get influenced by the introduction of a single outlier as they are included too
- They are not able to form clusters based on varying densities. That's why we need DBSCAN clustering.

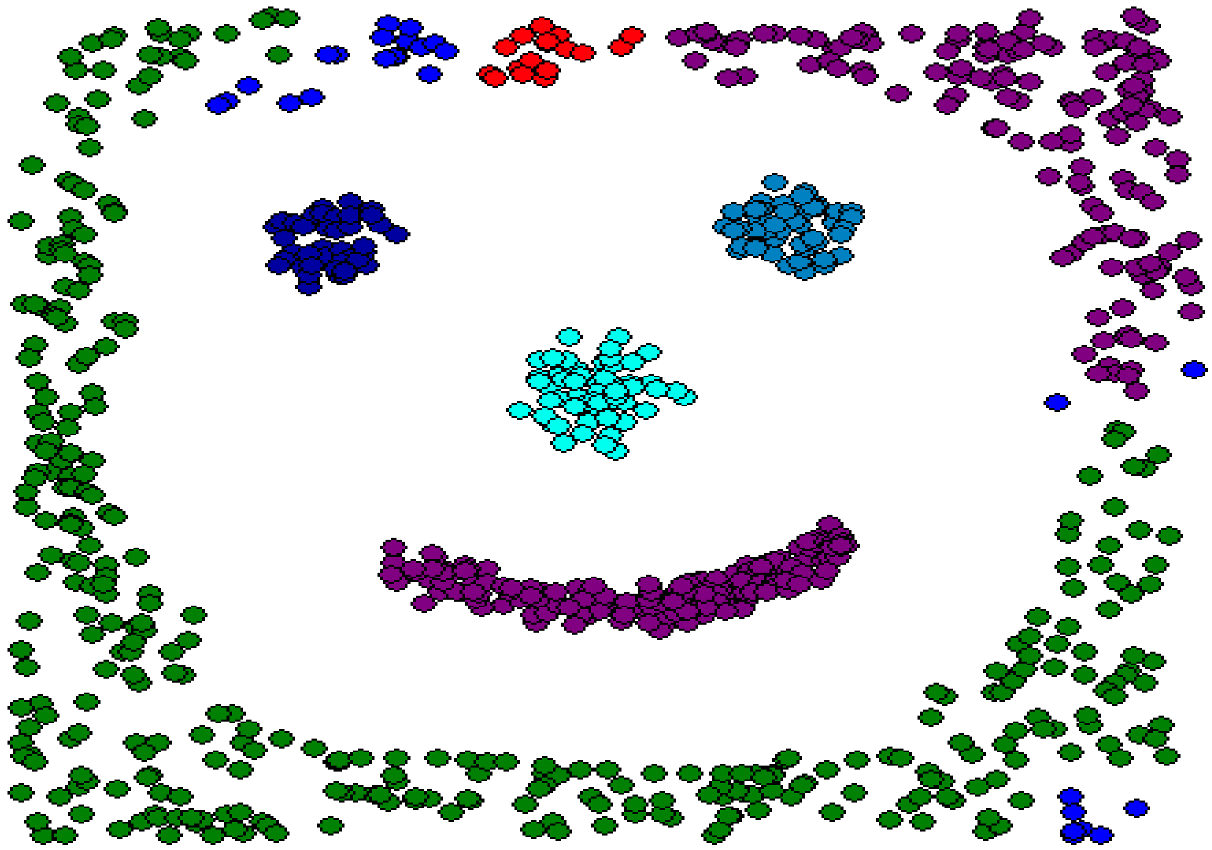
Arbitrary shape:



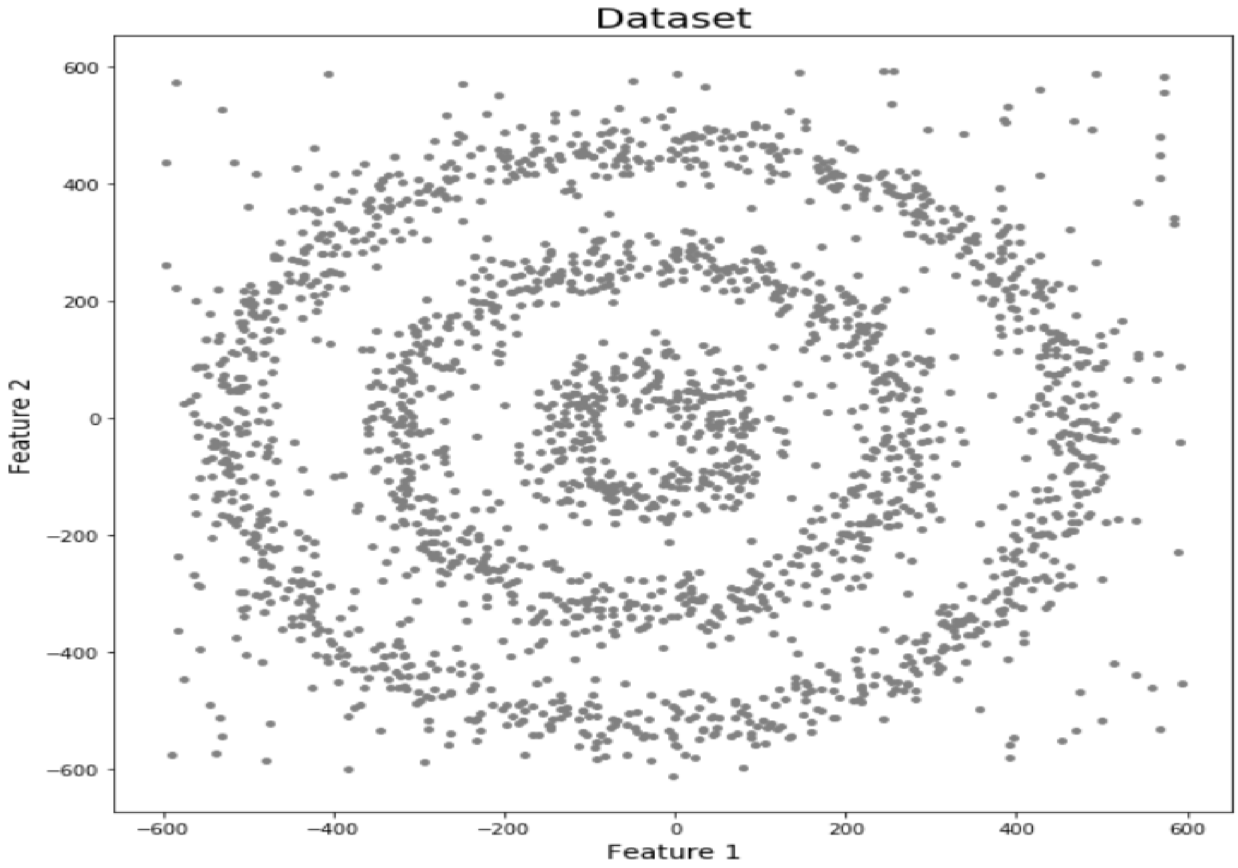
Contd..



Contd..

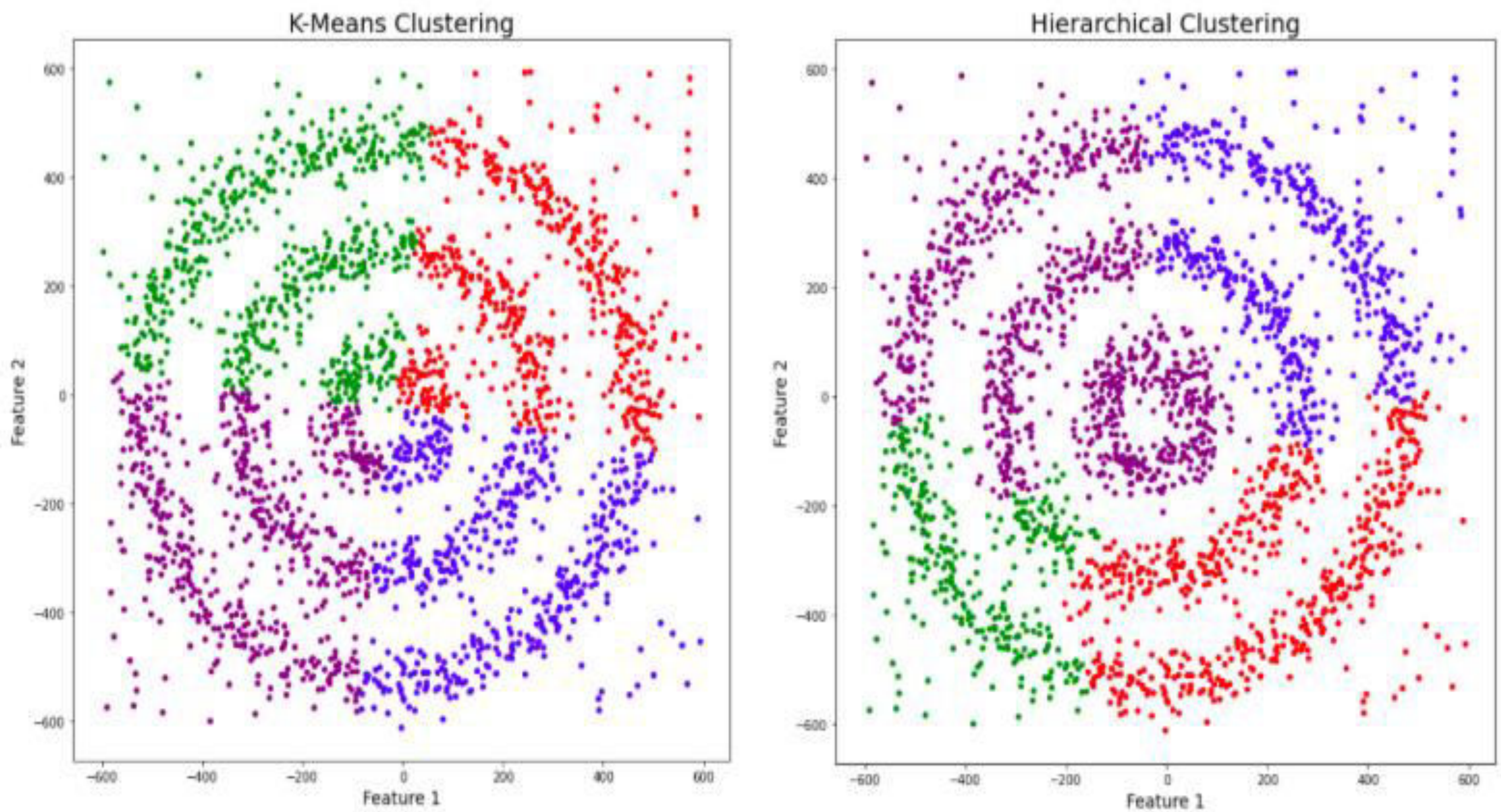


Contd..



Dataset
600
400
200
0
−200
−400
−600
Feature 2
−600
−400
−200
0
200
400
600
Feature 1

After k-means & Hierarchical clustering

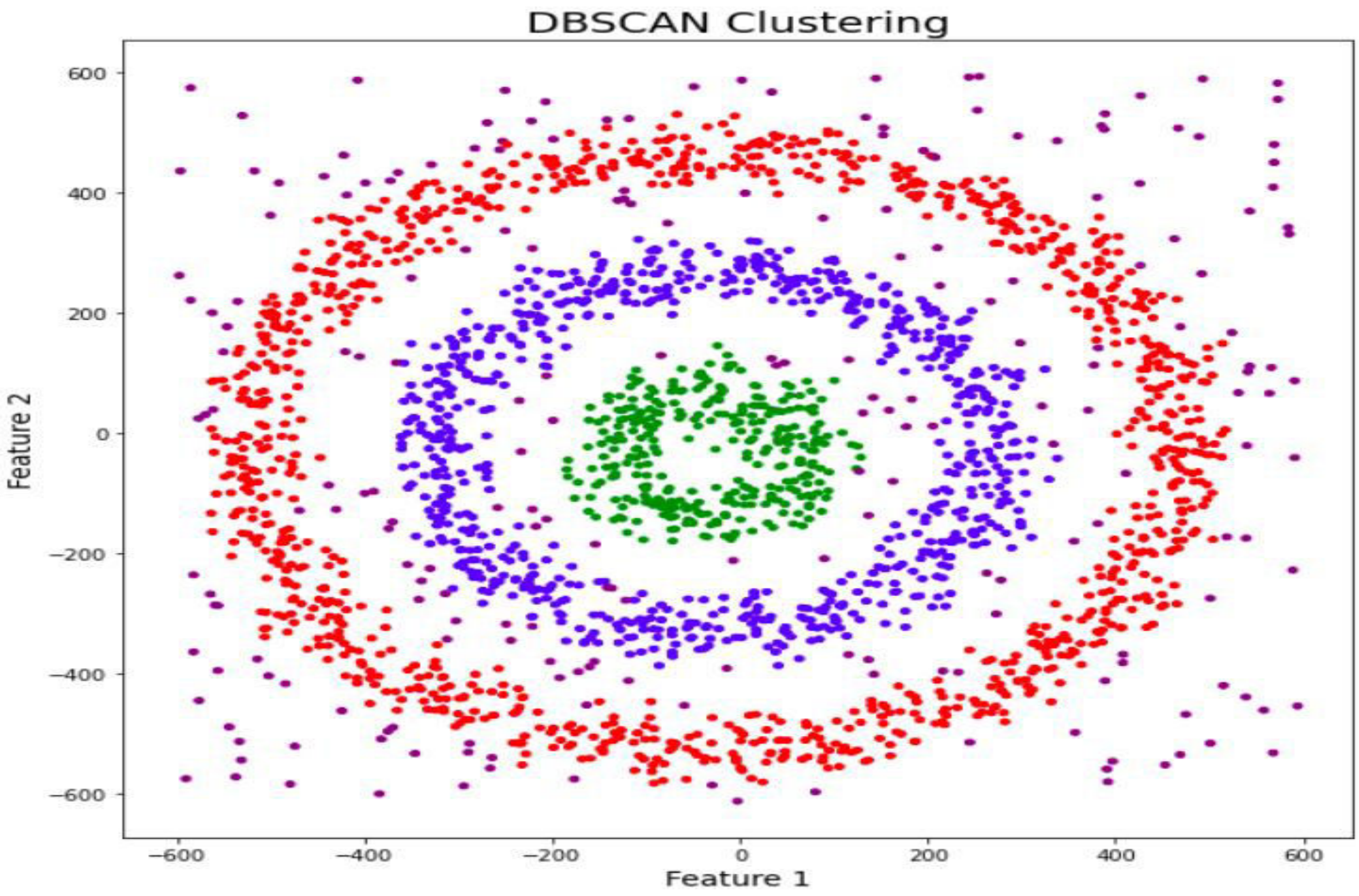


Observations

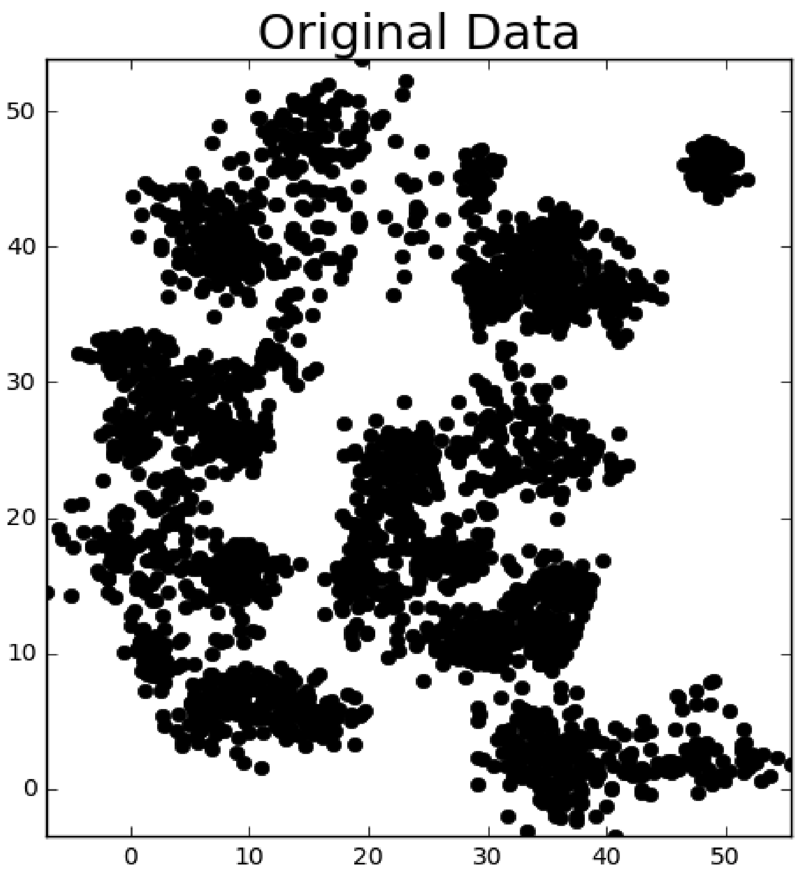
Observations

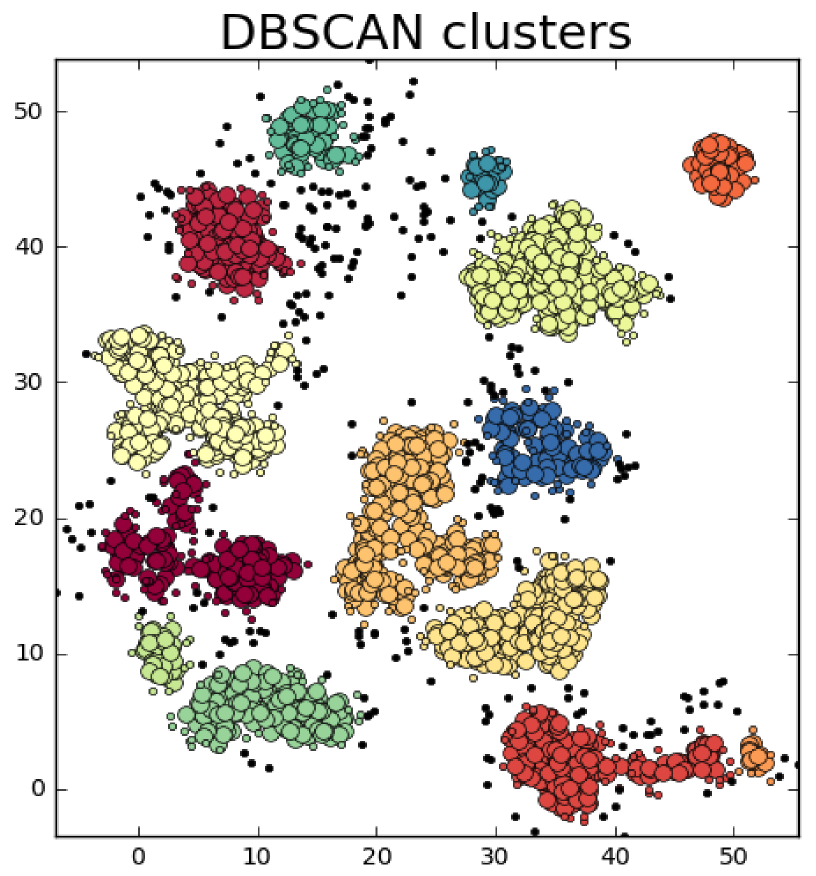
- You might be wondering why there are four colors in the graph?
- As I said earlier, this data contains noise too, therefore, we have taken noise as a different cluster which is represented by the purple color.
- Sadly, both of them failed to cluster the data points.
- Also, they were not able to properly detect the noise present in the dataset.

After DBSCAN



Contd..


Original Data
50
40
30
20
10
0
0
10
20
30
40
50


DBSCAN clusters
50
40
30
20
10
0
0
10
20
30
40
50

DBSCAN Clustering: Parameter

- DBSCAN requires only two parameters: epsilon and minPoints.
- Epsilon is the radius of the circle to be created around each data point to check the density and minPoints is the minimum number of data points required inside that circle for that data point to be classified as a Core point.
- In higher dimensions the circle becomes hypersphere, epsilon becomes the radius of that hypersphere, and minPoints is the minimum number of data points required inside that hypersphere.

Contd..

- **eps:**

1) It defines the neighborhood around a data point i.e. if the distance between two points is lower or equal to ‘eps’ then they are considered neighbors.

2) If the eps value is chosen too small then a large part of the data will be considered as an outlier. If it is chosen very large then the clusters will merge and the majority of the data points will be in the same clusters. One way to find the eps value is based on the k-distance graph.

- **MinPts:**

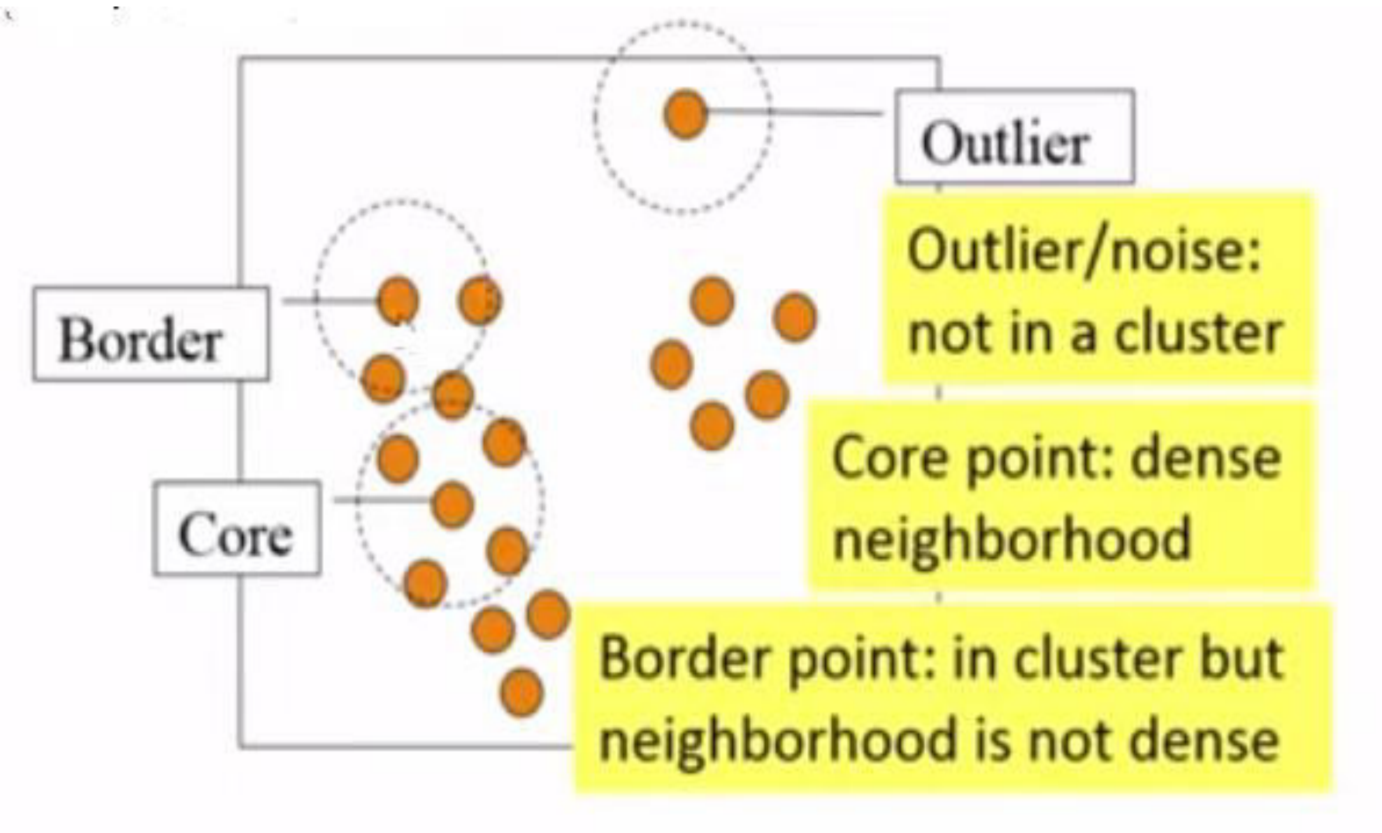
1) Minimum number of neighbors (data points) within eps radius. The larger the dataset, the larger value of MinPts must be chosen.

2) As a general rule, the minimum MinPts can be derived from the number of dimensions D in the dataset as, MinPts >= D+1. The minimum value of MinPts must be chosen at least 3.

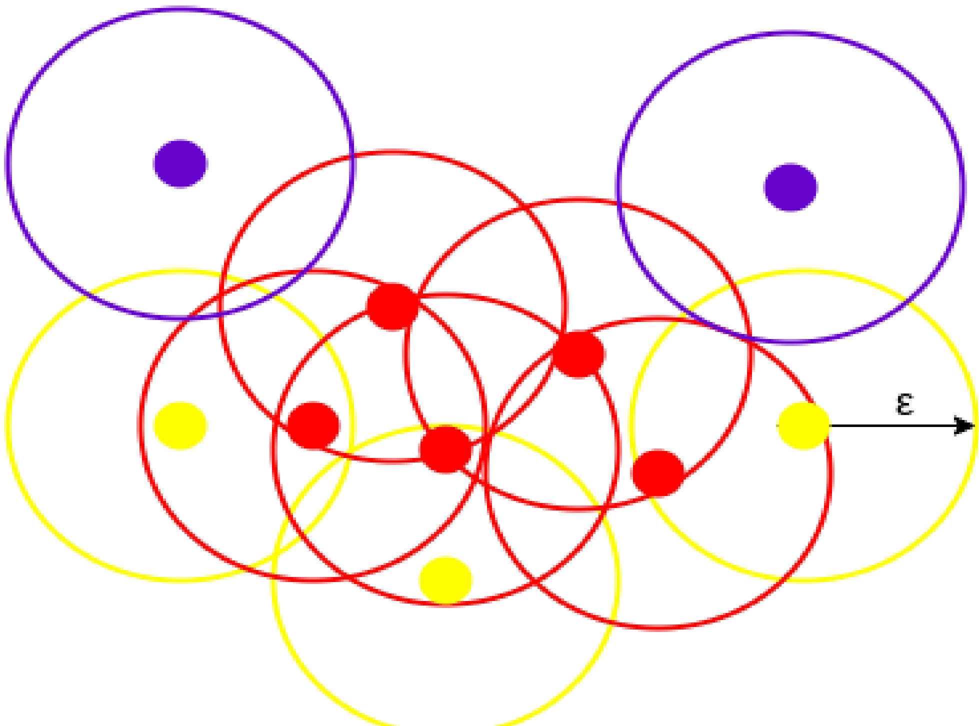
Types of data points

- **Core Point**: A point is a core point if it has more than MinPts points within eps.
- **Border Point:** A point which has fewer than MinPts within eps but it is in the neighborhood of a core point.
- **Noise or outlier**: A point which is not a core point or border point.

Contd..


Outlier
Outlier/noise: not in a cluster
Border
Core point: dense neighborhood
Core
Border point: in cluster but neighborhood is not dense

Contd..



Steps in DBSCAN

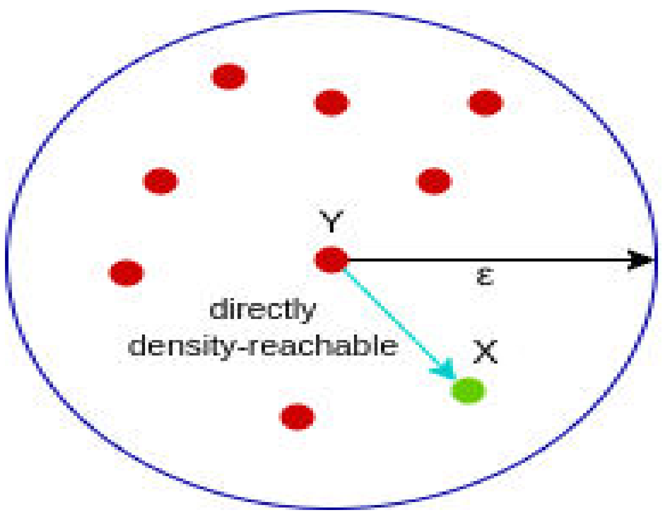
1. Find all the neighbor points within eps and identify the core points or visited with more than MinPts neighbors.
2. For each core point if it is not already assigned to a cluster, create a new cluster.
3. Find recursively all its density-connected points and assign them to the same cluster as the core point.
4. Iterate through the remaining unvisited points in the dataset. Those points that do not belong to any cluster are noise.

DBSCAN: Reachability and Connectivity

- These are the two concepts that you need to understand before moving further.
- Reachability states if a data point can be accessed from another data point directly or indirectly, whereas Connectivity states whether two data points belong to the same cluster or not.
- In terms of reachability and connectivity, few points in DBSCAN can be referred to as:
  1. Directly Density-Reachable
  2. Density-Reachable
  3. Density-Connected

Directly Density-Reachable

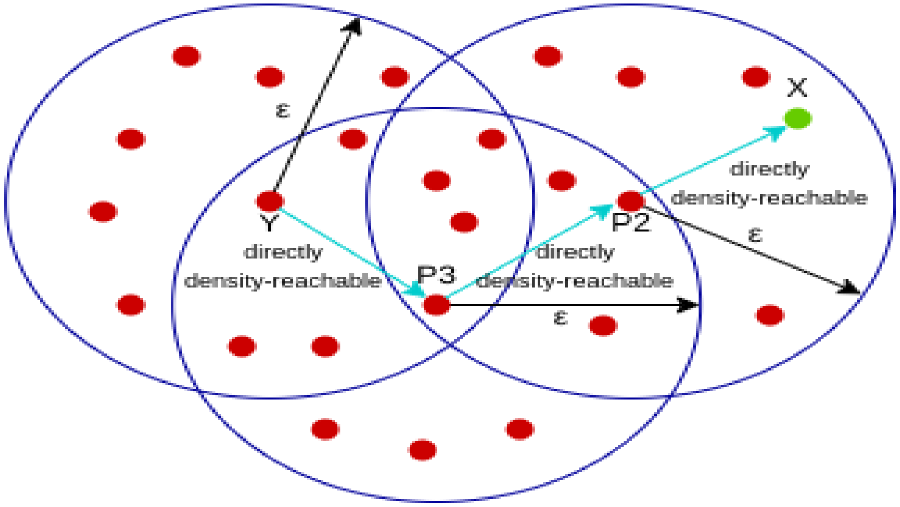
- A point X is directly density-reachable from point Y w.r.t epsilon, minPoints if,
  1. X belongs to the neighborhood of Y, i.e, dist(X, Y) <= epsilon
  2. Y is a core point
- Here, X is directly density-reachable from Y, but vice versa is not valid.



Density-reachable

- A point X is density-reachable from point Y w.r.t epsilon, minPoints if there is a chain of points p1, p2, p3, …, pn and p1=X and pn=Y such that pi+1 is directly density-reachable from pi.

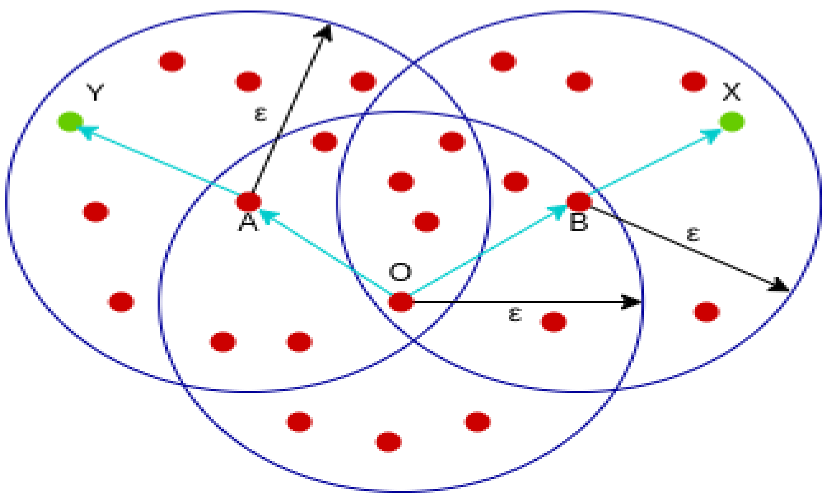
- Here, X is density-reachable from Y with X being directly density-reachable from P2, P2 from P3, and P3 from Y. But, the inverse of this is not valid.



Density-connected

- A point X is density-connected from point Y w.r.t epsilon and min_Points if there exists a point O such that both X and Y are density-reachable from O w.r.t to epsilon and min_Points.

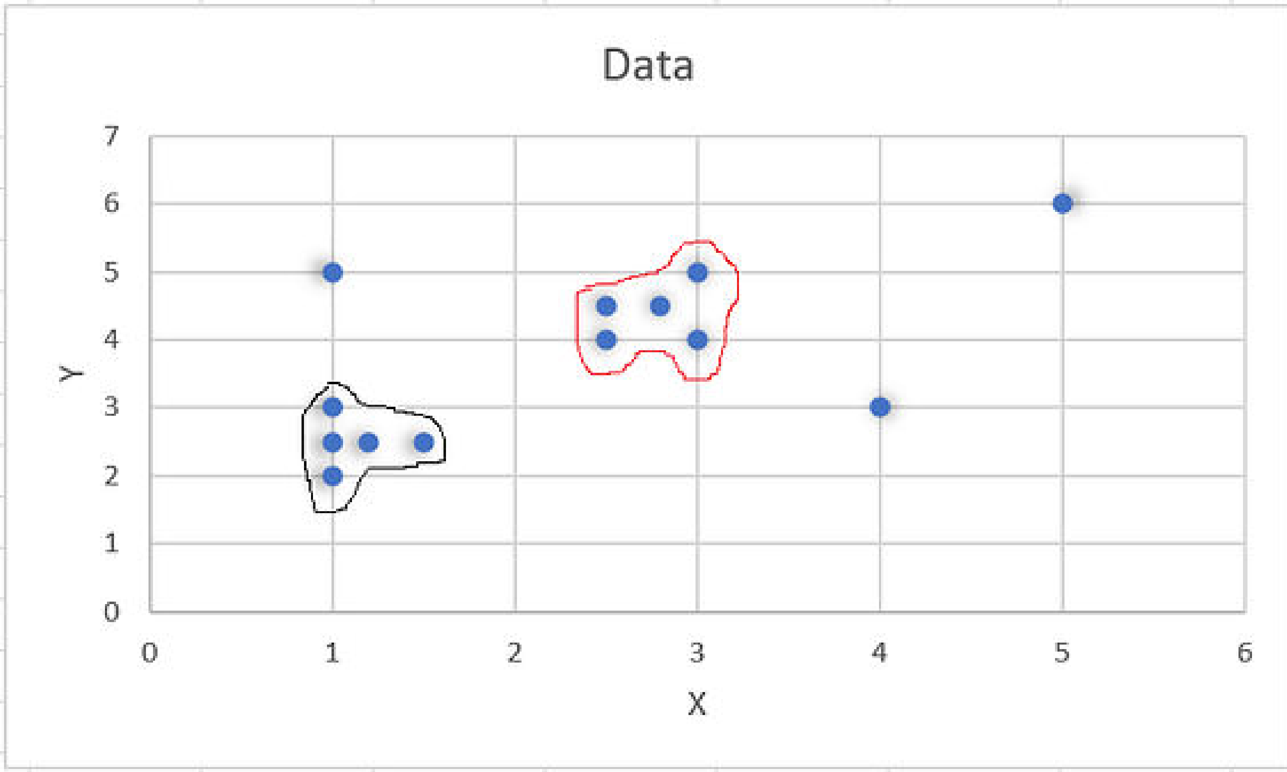
- Here, both X and Y are density-reachable from O, therefore, we can say that X is density-connected from Y



Numerical problem:

- For example, consider eps = 0.6 and MinPts =4 then implement DBSCAN?

X	Y
1	2
3	4
2.5	4
1.5	2.5
3	5
2.8	4.5
2.5	4.5
1.2	2.5
1	3
1	5
1	2.5
5	6
4	3



Solution

Point	Neighbourhood Points				
(1,2)	(1.2, 2.5)	(1, 2.5)			
(3, 4)	(2.5, 4)	(2.8, 4.5)			
(2.5, 4)	(3, 4)	(2.8, 4.5)	(2.5, 4.5)		
(1.5, 2.5)	(1.2, 2.5)	(1, 2.5)			
(3, 5)	(2.8, 4.5)				
(2.8, 4.5)	(3, 4)	(2.5, 4)	(3, 5)	(2.5, 4.5)	Cluster 1
(2.5, 4.5)	(2.5, 4)	(2.8, 4.5)			
(1.2, 2.5)	**(1, 2)**	(1.5, 2.5)	(1, 3)	(1, 2.5)	Cluster 2
(1, 3)	(1.2, 2.5)	(1, 2.5)			
(1, 5)					
(1, 2.5)	**(1, 2)**	(1.5, 2.5)	(1.2, 2.5)	(1, 3)	Cluster 2
(5, 6)					
(4, 3)					

Contd..

Point	Neighbourhood Points					
(1,2)	(1.2, 2.5)	(1, 2.5)			Border Point	
(3, 4)	(2.5, 4)	(2.8, 4.5)			Border Point	
(2.5, 4)	(3, 4)	(2.8, 4.5)	(2.5, 4.5)		Border Point	
(1.5, 2.5)	(1.2, 2.5)	(1, 2.5)			Border Point	
(3, 5)	(2.8, 4.5)				Border Point	
(2.8, 4.5)	(3, 4)	(2.5, 4)	(3, 5)	(2.5, 4.5)	**Core Point**	Cluster 1
(2.5, 4.5)	(2.5, 4)	(2.8, 4.5)			Border Point	
(1.2, 2.5)	**(1, 2)**	(1.5, 2.5)	(1, 3)	(1, 2.5)	**Core Point**	Cluster 2
(1, 3)	(1.2, 2.5)	(1, 2.5)			Border Point	
(1, 5)					Outlier	
(1, 2.5)	**(1, 2)**	(1.5, 2.5)	(1.2, 2.5)	(1, 3)	**Core Point**	Cluster 2
(5, 6)					Outlier	
(4, 3)					Outlier	

Contd..

Cluster 1	Cluster 2	Outliers
(3,4)	(1, 2)	(1, 5)
(2.5, 4)	(1.5, 2.5)	(5, 6)
(3,5)	(1.2, 2.5)	(4, 3)
(2.8, 4.5)	(1, 3)	
(2.5, 4.5)	(1, 2.5)	

Contd..

- **Direct density reachable**: A point is called direct density reachable if it has a core point in its neighborhood. Consider the **point (1, 2),** it has a core point (1.2, 2.5) in its neighborhood, hence, it will be a direct density reachable point.

- **Density Reachable:** A point is called density reachable from another point if they are connected through a series of core points. For example, consider the points (1, 3) and (1.5, 2.5), since they are connected through a core **point (1.2, 2.5),** they are called density reachable from each other.

- **Density Connected**: Two points are called density connected if there is a core point which is density reachable from both the points.

Lecture on

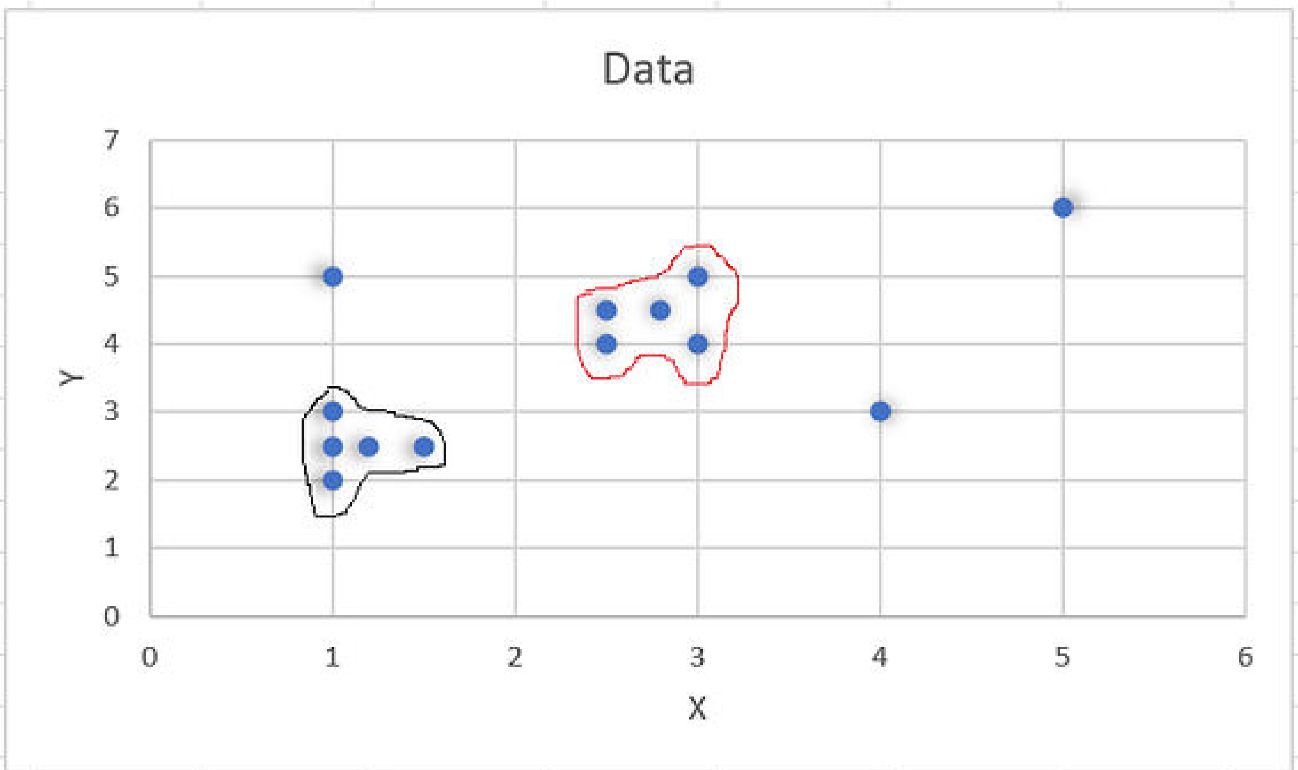
Clustering: Density-Based Clustering

Shesh Narayan Sahu, CSE

Numerical problem:

- For example, consider eps = 0.6 and MinPts =4 then implement DBSCAN?

X	Y
1	2
3	4
2.5	4
1.5	2.5
3	5
2.8	4.5
2.5	4.5
1.2	2.5
1	3
1	5
1	2.5
5	6
4	3



Solution

Point	Neighbourhood Points				
(1,2)	(1.2, 2.5)	(1, 2.5)			
(3, 4)	(2.5, 4)	(2.8, 4.5)			
(2.5, 4)	(3, 4)	(2.8, 4.5)	(2.5, 4.5)		
(1.5, 2.5)	(1.2, 2.5)	(1, 2.5)			
(3, 5)	(2.8, 4.5)				
(2.8, 4.5)	(3, 4)	(2.5, 4)	(3, 5)	(2.5, 4.5)	Cluster 1
(2.5, 4.5)	(2.5, 4)	(2.8, 4.5)			
(1.2, 2.5)	**(1, 2)**	(1.5, 2.5)	(1, 3)	(1, 2.5)	Cluster 2
(1, 3)	(1.2, 2.5)	(1, 2.5)			
(1, 5)					
(1, 2.5)	**(1, 2)**	(1.5, 2.5)	(1.2, 2.5)	(1, 3)	Cluster 2
(5, 6)					
(4, 3)					

Contd..

Point	Neighbourhood Points					
(1,2)	(1.2, 2.5)	(1, 2.5)			Border Point	
(3, 4)	(2.5, 4)	(2.8, 4.5)			Border Point	
(2.5, 4)	(3, 4)	(2.8, 4.5)	(2.5, 4.5)		Border Point	
(1.5, 2.5)	(1.2, 2.5)	(1, 2.5)			Border Point	
(3, 5)	(2.8, 4.5)				Border Point	
(2.8, 4.5)	(3, 4)	(2.5, 4)	(3, 5)	(2.5, 4.5)	**Core Point**	Cluster 1
(2.5, 4.5)	(2.5, 4)	(2.8, 4.5)			Border Point	
(1.2, 2.5)	**(1, 2)**	(1.5, 2.5)	(1, 3)	(1, 2.5)	**Core Point**	Cluster 2
(1, 3)	(1.2, 2.5)	(1, 2.5)			Border Point	
(1, 5)					Outlier	
(1, 2.5)	**(1, 2)**	(1.5, 2.5)	(1.2, 2.5)	(1, 3)	**Core Point**	Cluster 2
(5, 6)					Outlier	
(4, 3)					Outlier	

Contd..

Cluster 1	Cluster 2	Outliers
(3,4)	(1, 2)	(1, 5)
(2.5, 4)	(1.5, 2.5)	(5, 6)
(3,5)	(1.2, 2.5)	(4, 3)
(2.8, 4.5)	(1, 3)	
(2.5, 4.5)	(1, 2.5)	

Contd..

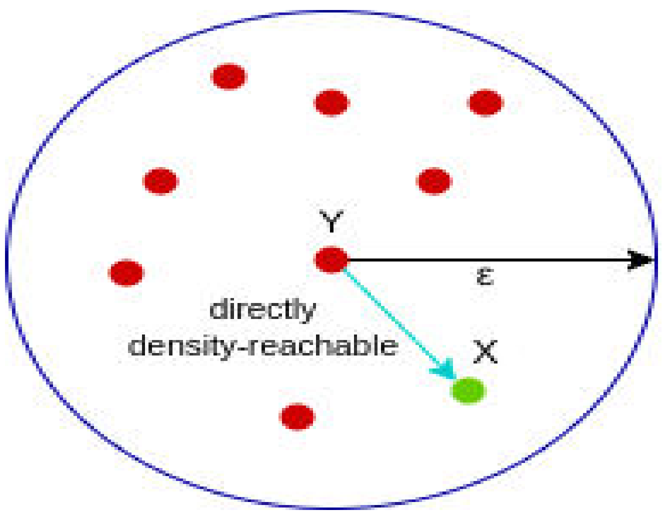
- If Epsilon is 2 and min point is 2, what are the clusters that DBScan would discover with the following 8
- examples: A1=(2,10), A2=(2,5), A3=(8,4), A4=(5,8), A5=(7,5), A6=(6,4), A7=(1,2), A8=(4,9).

Q. 1. Consider above points and draw the 10 by 10 space and illustrate the discovered clusters.

Q. 2. What if Epsilon is increased to 5 and then 10 ?

Directly Density-Reachable

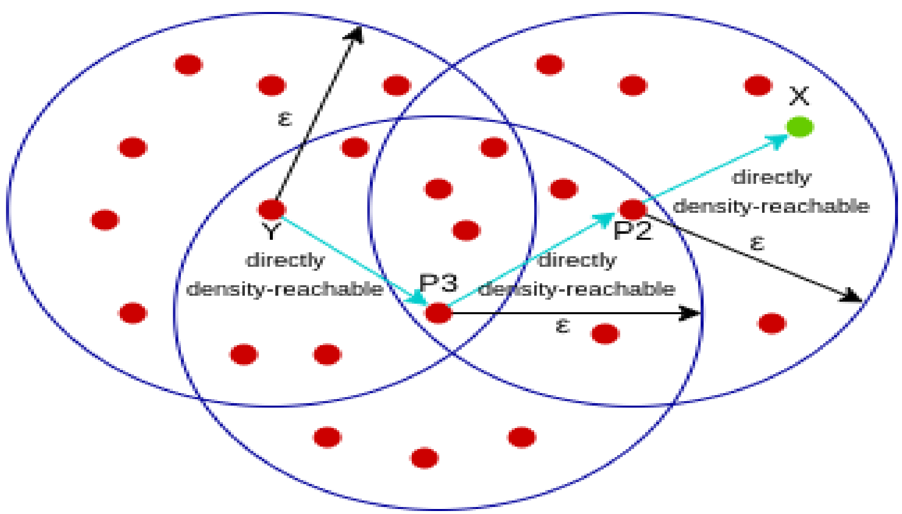
- A point X is directly density-reachable from point Y w.r.t epsilon, minPoints if,
  1. X belongs to the neighborhood of Y, i.e, dist(X, Y) <= epsilon
  2. Y is a core point
- Here, X is directly density-reachable from Y, but vice versa is not valid.



Density-reachable

- A point X is density-reachable from point Y w.r.t epsilon, minPoints if there is a chain of points p1, p2, p3, …, pn and p1=X and pn=Y such that pi+1 is directly density-reachable from pi.

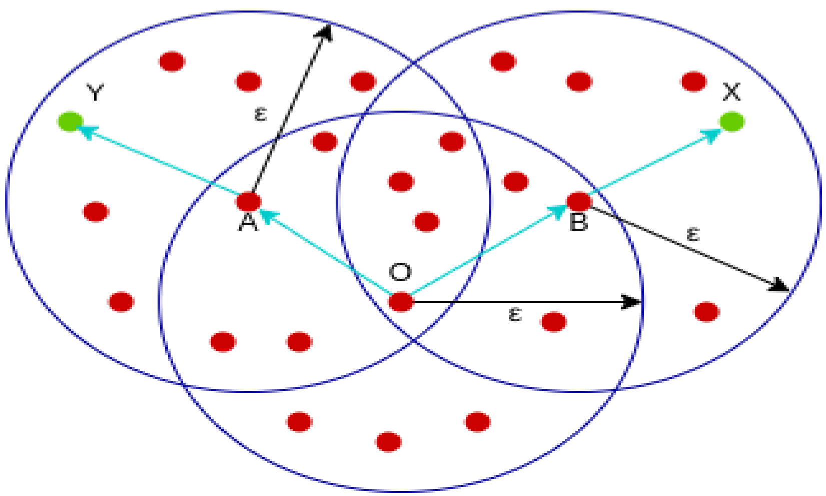
- Here, X is density-reachable from Y with X being directly density-reachable from P2, P2 from P3, and P3 from Y. But, the inverse of this is not valid.



Density-connected

- A point X is density-connected from point Y w.r.t epsilon and min_Points if there exists a point O such that both X and Y are density-reachable from O w.r.t to epsilon and min_Points.

- Here, both X and Y are density-reachable from O, therefore, we can say that X is density-connected from Y



Contd..

- **Direct density reachable**: A point is called direct density reachable if it has a core point in its neighborhood. Consider the **point (1, 2),** it has a core point (1.2, 2.5) in its neighborhood, hence, it will be a direct density reachable point.

- **Density Reachable:** A point is called density reachable from another point if they are connected through a series of core points. For example, consider the points (1, 3) and (1.5, 2.5), since they are connected through a core **point (1.2, 2.5),** they are called density reachable from each other.

- **Density Connected**: Two points are called density connected if there is a core point which is density reachable from both the points.

Lecture on

Rough Set Theory: Rough c-means

Introduction

- Video link- https://www.youtube.com/watch?v=mOx86M0LSPQ&t=634s
- https://www.sciencedirect.com/science/article/pii/S2468232216300786

Introduction: Rough C-Means

- Rough set C-means have been proposed as clustering models on an approximation space considering the granularity of the universe.

- It consider three types of memberships, that is, the lower, upper, and boundary areas of each cluster, to represent a certain, possible, and uncertain belonging of object to cluster, respectively.

- Note that RCM-type methods do not consider binary relations and the granularity of the object space, so the lower, upper, and boundary areas are different concepts from the lower approximations, upper approximations, and boundary regions in rough set theory.

Contd..

- It constitutes sound basis for knowledge discovery in databases, it offers mathematical tools to discover patterns hidden in data.
- It can be used for feature selection, feature extraction, data reduction, decision rule generation, and pattern extraction (templates, association rules), clustering, etc.
- Identifies partial or total dependencies in data, eliminates redundant data, gives approach to null values, missing data, dynamic data and others.

Example: Text-Graphics Segmentation Using HCM

Example: Text-Graphics Segmentation Using FCM

Example: Text-Graphics Segmentation Using PCM

Example: Text-Graphics Segmentation Using FPCM

Example: Text-Graphics Segmentation Using RFCM

Example: Text-Graphics Segmentation Using RPCM

Shesh Narayan Sahu, CSE

Example: Text-Graphics Segmentation Using RFPCM

Rough Set

- Rough set theory is a new paradigm to deal with uncertainty, vagueness, and incompleteness. It is proposed for indiscernibility in classification or clustering according to some similarity.

- Rough set theory is based on the establishment of equivalence classes within the given training data.

- All the data tuples forming an equivalence class are indiscernible, that is, the samples are identical with respect to the attributes describing the data.

- Given real-world data, it is common that some classes cannot be distinguished in terms of the available attributes. Rough sets can be used to approximately or “roughly” define such classes.

Concepts

- Information/Decision system
- Indiscernibility
- Set Approximation
- Reduct and core
- Rough Membership

Information Systems/Tables

- Consider Information system IS is a pair of (U, A).
- U is a non empty finite set of objects.
- A is non empty finite set of attribute such that a: U---> V_a

U	A1	A2	Walk
x1	16-30	50	yes
x2	16-30	0	no
x3	31-45	1-25	no
x4	31-45	1-25	yes
x5	46-60	26-49	no
x6	16-30	26-49	yes
x7	46-60	26-49	no

Rough set: Indiscernibility

- Indiscernibility Relation is a central concept in Rough Set Theory, and is considered as a relation between two objects or more, where all the values are identical in relation to a subset of considered attributes.

- Indiscernibility relation is an equivalence relation, where all identical objects of set are considered as elementary (Pawlak, 1998).

- It can be expressed as-

$$\mathrm{IND}(P) = \{(x, y) \in \mathbb{U}^2 \mid \forall a \in P, a(x) = a(y)\}$$

Contd..

- IND({A1}) = {{x1,x2,x6}, {x3,x4}, {x5,x7}}
- IND({A2})= ?
- IND({A1, A2})= ?

U	A1	A2	Walk
x1	16-30	50	yes
x2	16-30	0	no
x3	31-45	1-25	no
x4	31-45	1-25	yes
x5	46-60	26-49	no
x6	16-30	26-49	yes
x7	46-60	26-49	no

Contd..

- IND({A1}) = {{x1,x2,x6}, {x3,x4}, {x5,x7}}
- IND({A2})= {{x1}, {x2},{x3 x4}, {x5,x6,x7}}
- IND({A1, A2})= {{ x1}, {x2}, {x3,x4}, {x5,x7}, {x6}}.

Observation

- An equivalence relation induces a partitioning of the universe.
- The partitions can be used to build new subsets of the universe.
- Subsets that are most often of interest have the same value of the decision attribute.
- It may happen, however, that a concept such as "Walk" cannot be defined in a crisp manner.

Set Approximation

- The indiscernibility relation is intended to express the fact that due to the lack of knowledge it is unable to discern some objects employing the available information Approximations is also an important concept in Rough Sets.

- The lower and the upper approximations of a set are interior and closure operations in a topology generated by the indiscernibility relation.

Contd..

- Lower approximation is the set of objects that positively belong to the target set.

Lower Approximation:

$$\underline{R}X = \bigcup\{Y \in U/R : Y \subseteq X\}$$

- Upper approximation is the set of objects which possibly belong to the target set

Upper Approximation:

$$\overline{R}X = \bigcup\{Y \in U/R : Y \cap X \neq \phi\}$$

Contd..

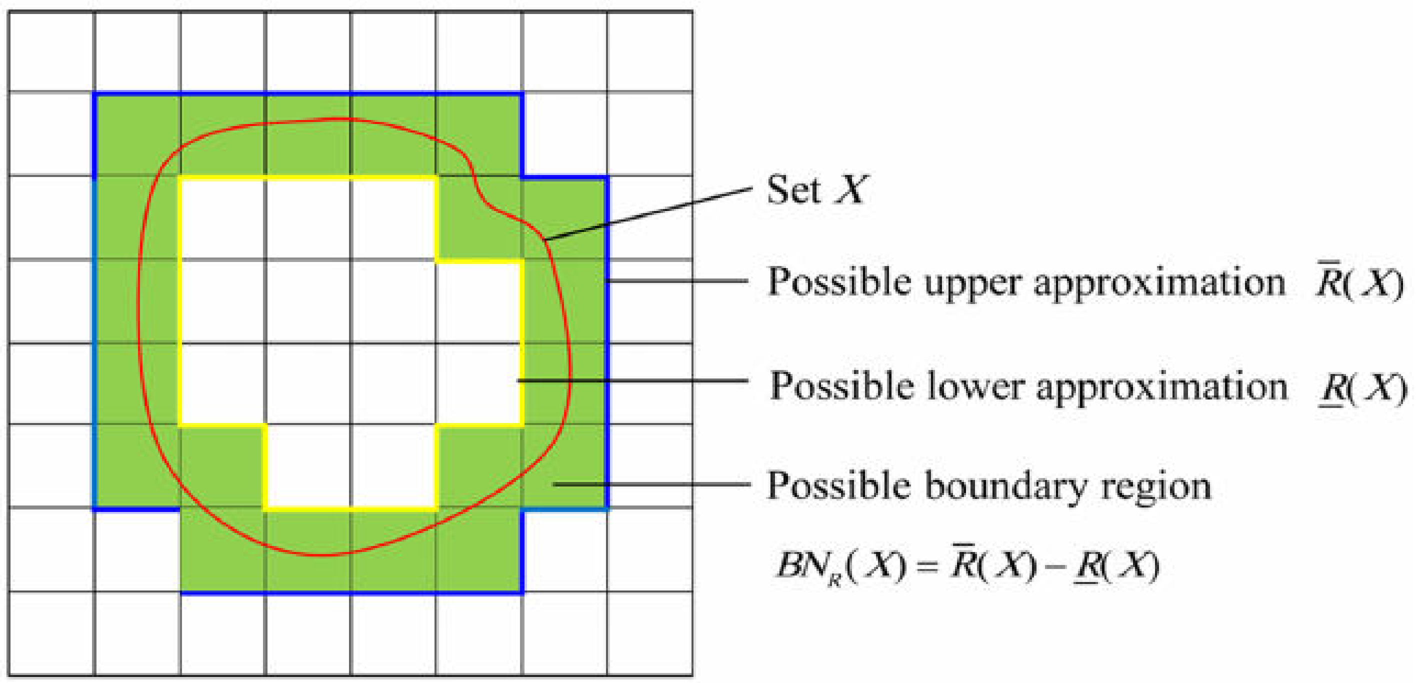
- Boundary region-
  - boundary region of X, comprises that can not be classified with certainty to be neither inside x nor outside x using attribute set B.

Or

  - B-boundary region of X, consists of those objects that we cannot decisively classify into X in B.

$$BN_B(X) = \overline{B}X - \underline{B}X$$

Contd..



Set X
Possible upper approximation $\overline{R}(X)$
Possible lower approximation $\underline{R}(X)$
Possible boundary region
$BN_R(X) = \overline{R}(X) - \underline{R}(X)$

Finding approximation

Consider

X= {x: walk(x)= yes }

Attribute set A= {A1, A2}

Find lower, upper approximation and boundary ?

Finding approximation

Consider

X= {x: walk(x)= yes }

Attribute set A= {A1, A2}

- X= {x1, x4, x6 }
- IND(A)= {{ x1}, {x2}, {x3,x4}, {x5,x7}, {x6}}.

- Lower approximation of A:

  A_L= {x1, x6}
- Upper approximation of A:

  A_U= {x1, x3, x4, x6}

- B boundary region- It consists of those objects that we can not decisively classify into X in B.

$$BN_B(X) = \overline{B}X - \underline{B}X$$

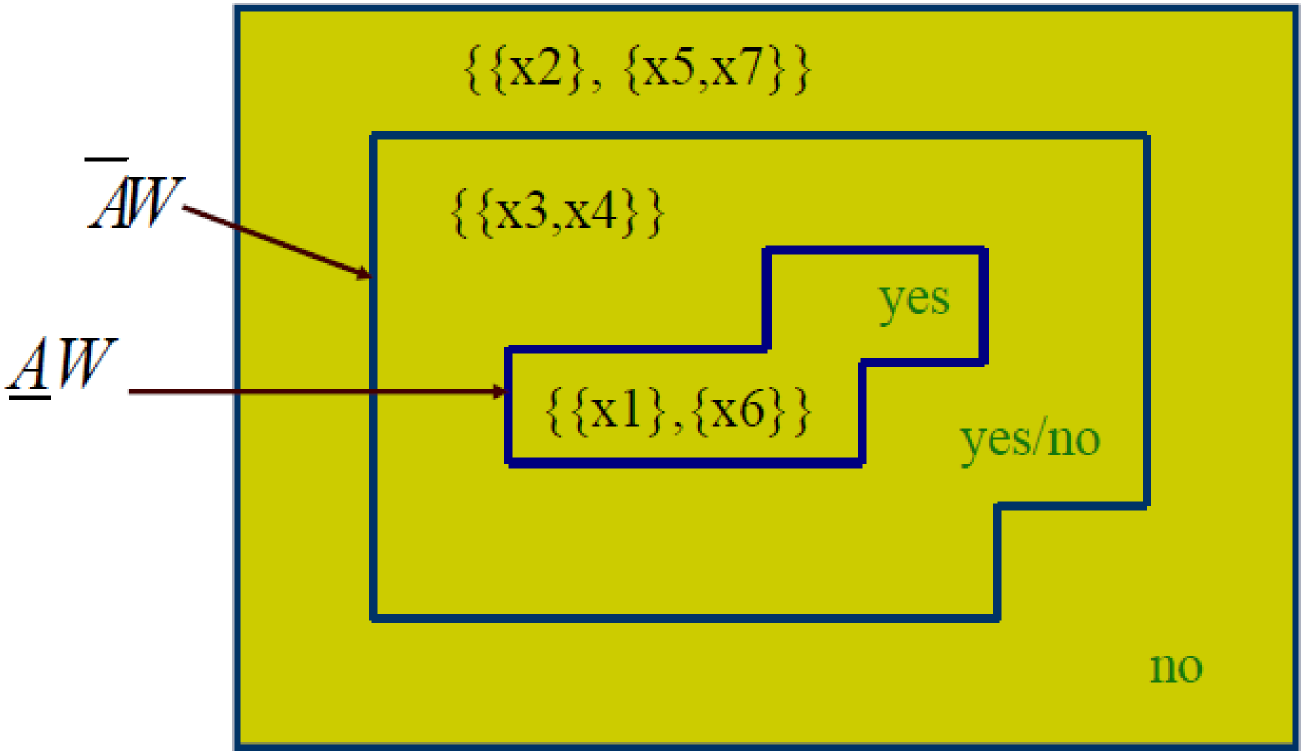
BN (W)= {x3, x4}

- B outside boundary- It consists of those objects that can be witg certainty classified as not belonging to X.

$$U - \overline{B}X$$

- U- A_U (W)= {x2, x5, x7}

Contd..



- A set is said to be rough if its boundary region is non-empty, otherwise the set is crisp.

Steps in Rough c means

- Input: Data set, number of cluster k, parameter value (w_lower, w_upper, threshold value)

- Output: Lower approximation of K and upper approximation of K.

Contd..

Step1: Randomly assign each data object one lower approximation $\underline{U}(K)$. By definition (property 2) the data object also belongs to upper approximation $\overline{U}K$ of the same Cluster.

Step 2: Compute Cluster Centroids C_j

If $\underline{U}(K) \neq \emptyset$ and $\overline{U}(K)$ - $\underline{U}(K) = \emptyset$

$$C_j = \sum_{x \in \underline{U}(K)} \frac{x_i}{|\underline{U}(K)|}$$

Else If $\underline{U}(K) = \emptyset$ and $\overline{U}(K)$ - $\underline{U}(K) \neq \emptyset$

$$C_j = \sum_{x \in \overline{U}(K) - \underline{U}(K)} \frac{x_i}{|\overline{U}(K) - \underline{U}(K)|}$$

Else

$$C_j = W_{lower} \times \sum_{x \in \underline{U}(K)} \frac{x_i}{|\underline{U}(K)|} + W_{upper} \times \sum_{x \in \overline{U}(K) - \underline{U}(K)} \frac{x_i}{|\overline{U}(K) - \underline{U}(K)|}$$

Step 3: Assign each object to the lower approximation $\underline{U}(K)$ or upper approximation $\overline{U}(K)$ of cluster i respectively. For each object vector x, let $d(X, C_j)$ be the distance between itself and the centroid of cluster C_j.

$$d(X, C_j) = min_{1 \leq j \leq K}\, d(X, C_j).$$

The ratio $d(X, C_i) / d(X, C_j)$, $1 \leq i, j \leq K$ is used to determine the membership of x as follow: If $d(X, C_i) / d(X, C_j) \leq$ epsilon, for any pair (i, j), the $x \in \overline{U}(C_i)$ and $x \in \overline{U}(C_j)$ and x will not be a part of any lower approximation. Otherwise, $x \in \underline{U}(C_i)$, such that $d(X, C_i)$ is the minimum of $1 \leq i \leq K$. In addition $x \in \overline{U}(C_i)$.

Step 4: Repeat Steps 2 and 3 until convergence.

Example

Table 1 shows example information system with real-valued conditional attributes. It consists of six objects/genes, and two features/samples. k = 2, which is the number of clusters. Weight of the lower approximation W_{lower}=0.7, Weight of the upper approximation W_{upper} = 0.3 and Relative threshold = 2.

Table 1 Example dataset for Rough K-Means

U	X	Y
1	0	3
2	1	3
3	3	1
4	3	0.5
5	5	0
6	6	0

Solution

Step1: Randomly assign each data objects to exactly one lower approximation

$\underline{K}_1 = \{(0, 3), (1, 3), (3, 1)\}$

$\underline{K}_2 = \{(3, 0.5), (5, 0), (6, 0)\}$

Step 2: In this case $\underline{U}(K) \neq \emptyset$ and $\overline{U}(K)$ - $\underline{U}(K) = \emptyset$, so we compute the centroid

using $C_j = \sum_{x \in \underline{U}(K)} \frac{x_i}{|\underline{U}(K)|}$,

$$C_1 = \left(\frac{0+1+3}{3}, \frac{3+3+1}{3}\right) = (1.33, 2.33)$$

$$C_2 = \left(\frac{3+5+6}{3}, \frac{0.5+0+0}{3}\right) = (4.67, 0.17)$$

Contd..

$d_1(X, C_1)$:

$$(0, 3)(1.33, 2.33) \Rightarrow \sqrt{(1.33-0)^2 + (2.33-3)^2} = 1.49$$

$$(1, 3)(1.33, 2.33) \Rightarrow \sqrt{(1.33-1)^2 + (2.33-3)^2} = 0.75$$

$$(3, 1)(1.33, 2.33) \Rightarrow \sqrt{(1.33-3)^2 + (2.33-1)^2} = 2.13$$

$$(3, 0.5)(1.33, 2.33) \Rightarrow \sqrt{(1.33-3)^2 + (2.33-0.5)^2} = 2.48$$

$$(5, 0)(1.33, 2.33) \Rightarrow \sqrt{(1.33-5)^2 + (2.33-0)^2} = 4.45$$

$$(6, 0)(1.33, 2.33) \Rightarrow \sqrt{(1.33-6)^2 + (2.33-0)^2} = 5.22$$

Contd..

$d_2(X, C_2)$:

$$(0, 3)(4.67, 0.17) \Rightarrow \sqrt{(4.67-0)^2 + (0.17-3)^2} = 5.46$$

$$(1, 3)(4.67, 0.17) \Rightarrow \sqrt{(4.67-1)^2 + (0.17-3)^2} = 4.63$$

$$(3, 1)(4.67, 0.17) \Rightarrow \sqrt{(4.67-3)^2 + (0.17-1)^2} = 1.86$$

$$(3, 0.5)(4.67, 0.17) \Rightarrow \sqrt{(4.67-3)^2 + (0.17-0.5)^2} = 1.70$$

$$(5, 0)(4.67, 0.17) \Rightarrow \sqrt{(4.67-5)^2 + (0.17-0)^2} = 0.37$$

$$(6, 0)(4.67, 0.17) \Rightarrow \sqrt{(4.67-6)^2 + (0.17-0)^2} = 1.34$$

Contd..

Step 3: Assign each object to the lower approximation $\underline{U}(K)$ or upper approximation $\overline{U}(K)$ of cluster i respectively. Check If d(X, C_i) /d(X, C_j) $\leq$ epsilon.

1. $(0, 3) \Rightarrow d_2 / d_1 = 5.46/1.49 = 3.66443 \nleq 2$. So, x_1 will be a part of $\underline{K_1}$
2. $(1, 3) \Rightarrow 4.63/0.75 = 6.173 \nleq 2$. So, x_2 will be a part of $\underline{K_1}$
3. $(3, 1) \Rightarrow 2.13/1.86 = 1.145 < 2$, so x_3 will not be a part of $\underline{K_1}$ & $\underline{K_2}$
4. $(3, 0.5) \Rightarrow 2.48/1.70 = 1.458 < 2$, so x_4 will not be a part of $\underline{K_1}$ & $\underline{K_2}$
5. $(5, 0) \Rightarrow 4.35/0.37 = 11.756 \nleq 2$. So, x_5 will be a part of $\underline{K_2}$
6. $(6, 0) \Rightarrow 5.22/1.34 = 3.895 \nleq 2$. So, x_6 will be a part of $\underline{K_2}$

Contd..

Now, we have clusters

$\underline{K_1}$= {(0, 3), (1, 3)} $\overline{K}_1$ = {(0, 3), (1, 3), (3, 1), (3, 0.5)}

$\underline{K_2}$= {(5, 0), (6, 0)} $\overline{K}_2$ = {(5, 0), (6, 0), (3, 1), (3, 0.5)}

Here, $\underline{U}(K) \neq \emptyset$ and $\overline{U}(K)$ - $\underline{U}(K) \neq \emptyset$ then find out the new centroid by using below equation,

$$C_j = W_{lower} \times \sum_{x \in \underline{U}(K)} \frac{x_i}{|\underline{U}(K)|} + W_{upper} \times \sum_{x \in \overline{U}(K) - \underline{U}(K)} \frac{x_i}{|\overline{U}(K) - \underline{U}(K)|}$$

$$C_1 = \quad 0.7 \times \left(\frac{0+1}{2}, \frac{3+3}{2}\right) + 0.3 \times \left(\frac{3+3}{2}, \frac{1+0.5}{2}\right) \quad = (1.25, 2.325)$$

$$C_2 = \quad 0.7 \times \left(\frac{5+6}{2}, \frac{0+0}{2}\right) + 0.3 \times \left(\frac{3+3}{2}, \frac{1+0.5}{2}\right) \quad = (4.75, 0.225)$$

Step 4: Repeat Steps 2 and 3 until convergence (Old Centroid = New Centroid).

References

- https://www.youtube.com/watch?v=mOx86M0LSPQ&t=634s

- Qinghua Zhang a b, Qin Xie a, Guoyin Wang, “A survey on rough set theory and its applications”, CAAI Transactions on Intelligence Technology, Volume 1, Issue 4, October 2016, Pages 323-333.

Lecture on

Clustering: Fuzzy c-means clustering

Shesh Narayan Sahu, CSE

Introduction

- The unsupervised k-means clustering algorithm gives the values of any point lying in some particular cluster to be either as 0 or 1 i.e., either true or false.
- But the fuzzy logic gives the fuzzy values of any particular data point to be lying in either of the clusters.
- Here, in fuzzy c-means clustering, we find out the centroid of the data points and then calculate the distance of each data point from the given centroids until the clusters formed become constant.

Contd..

- Fuzzy Clustering is a type of clustering algorithm in machine learning that allows a data point to belong to more than one cluster with different degrees of membership.

- Unlike traditional clustering algorithms, such as k-means or hierarchical clustering, which assign each data point to a single cluster, fuzzy clustering assigns a membership degree between 0 and 1 for each data point for each cluster.

Example

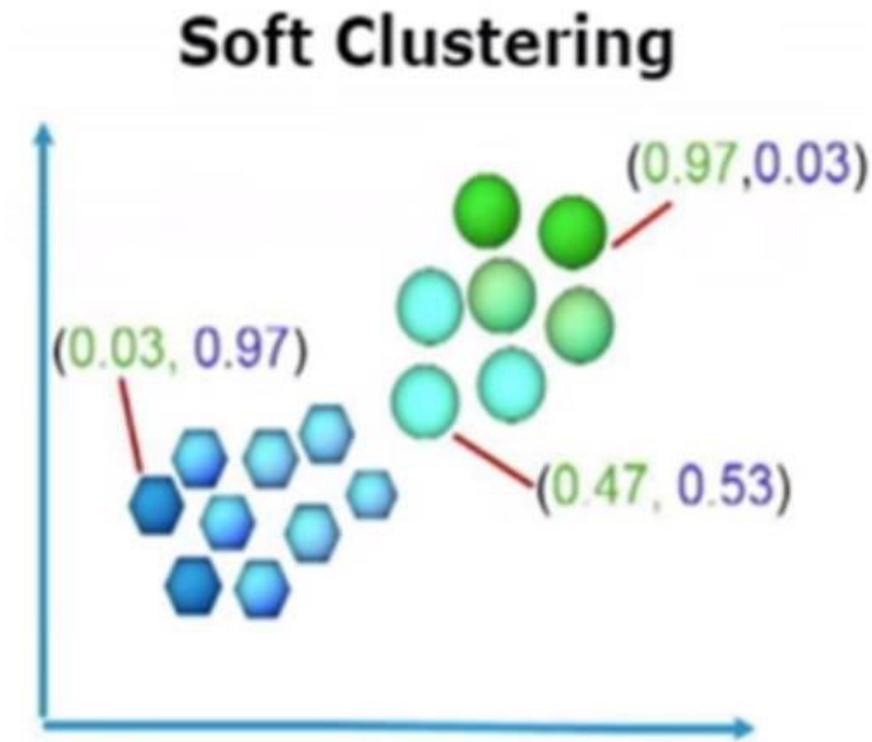
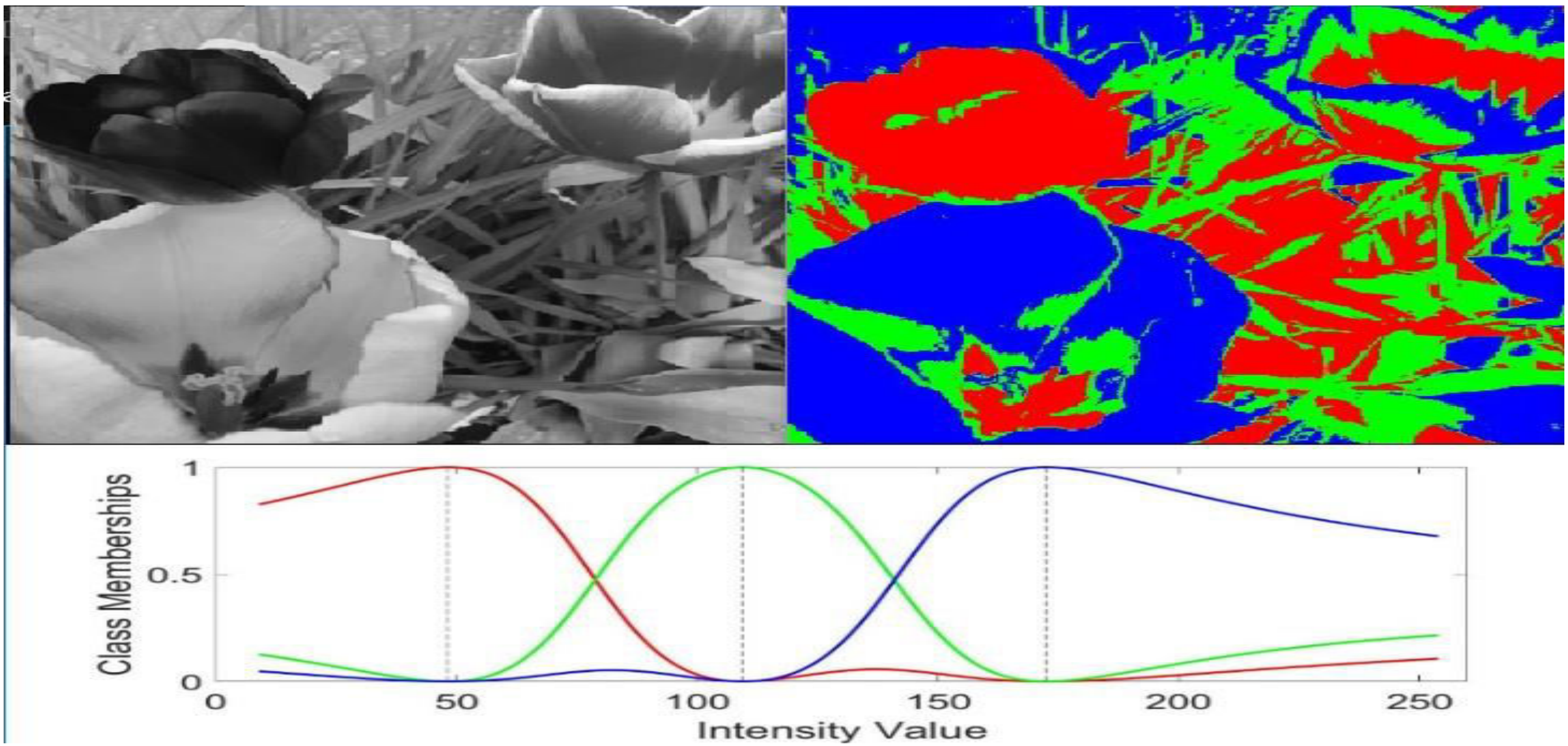
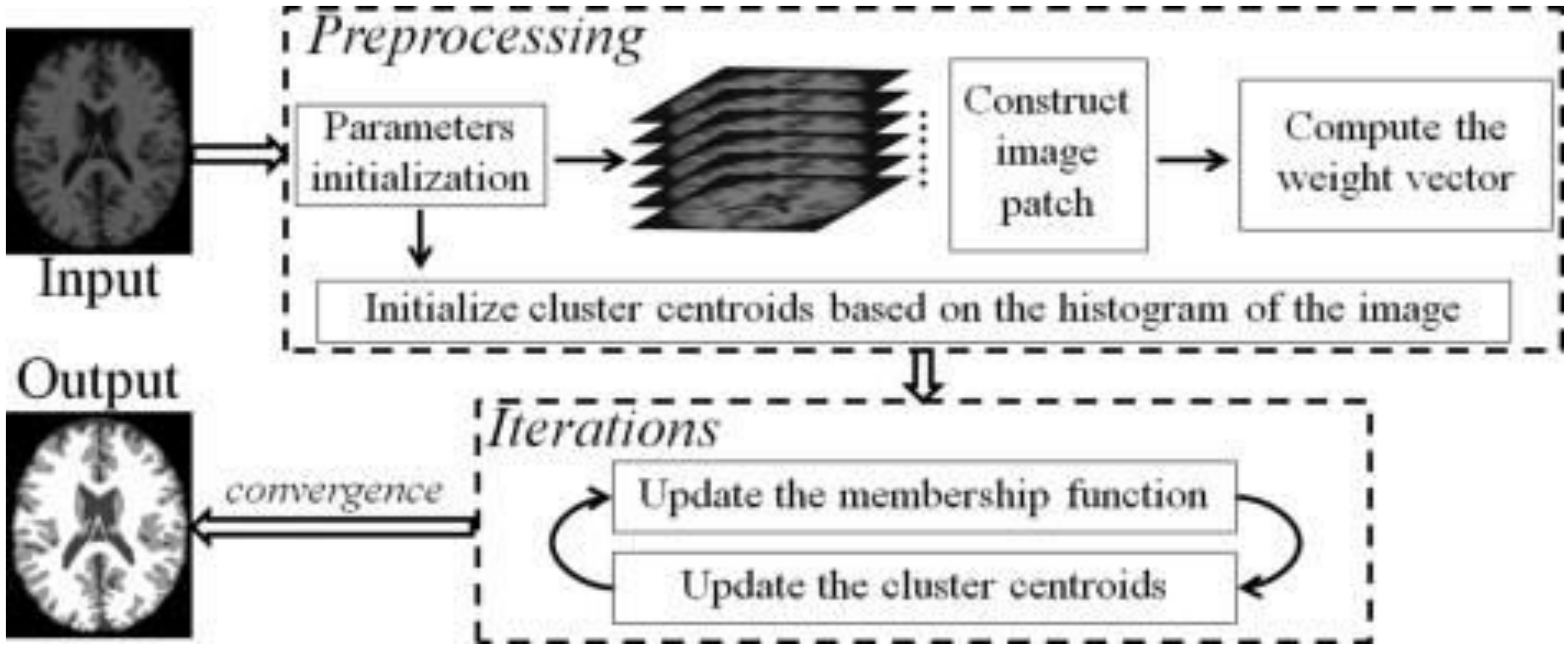


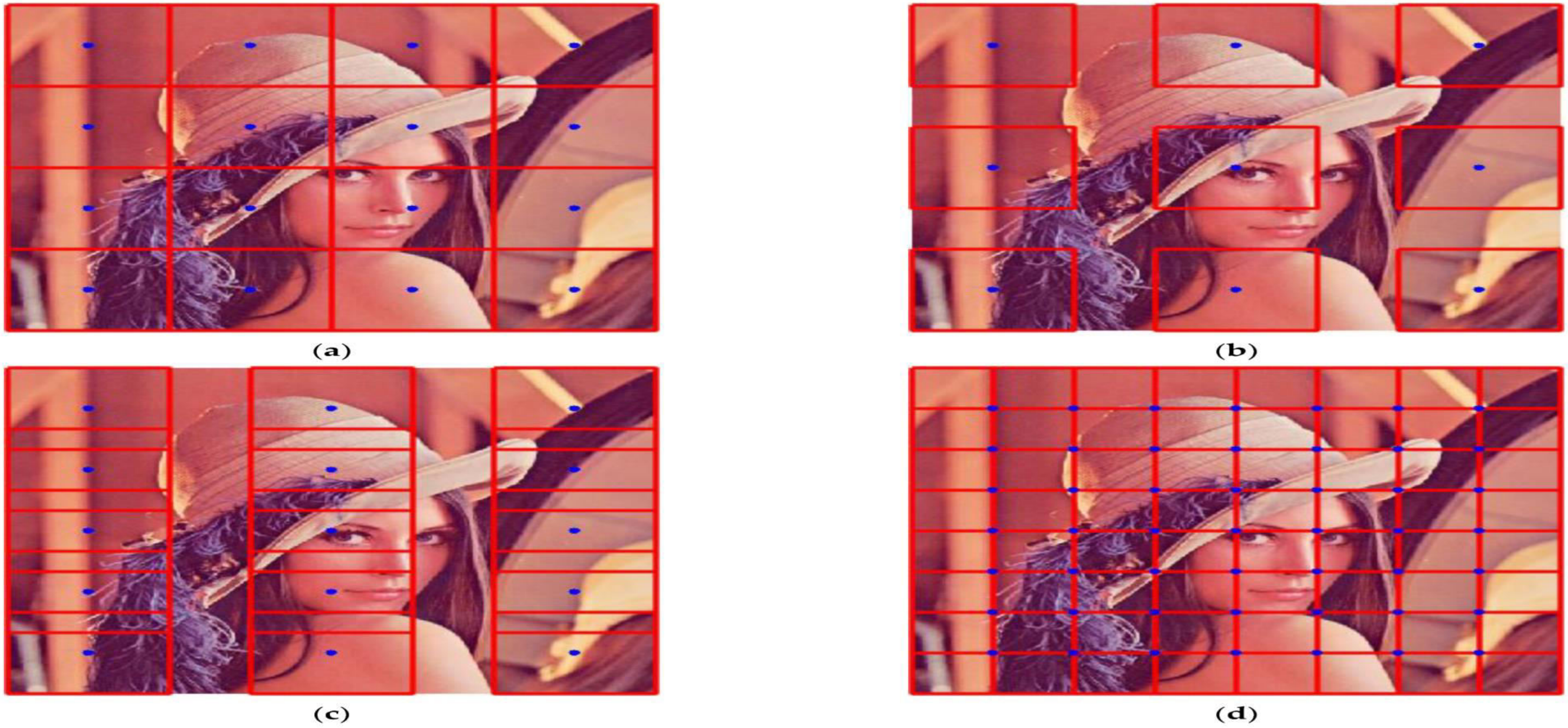
Image segmented by fuzzy clustering, with the original (top left), clustered (top right), and membership map (bottom)



Fuzzy c-means clustering with weighted image patch for image segmentation



Patch extraction examples for image



Fuzzy logic: Basics

- The word “fuzzy” means “vagueness (ambiguity)”.
- Fuzziness occurs when the boundary of a piece of information is not clear-cut.
- In 1965 Lotfi Zadeh, published his famous paper “Fuzzy sets”. This new logic for representing and manipulating fuzzy terms was called fuzzy logic, and Zadeh became the Master/Father of fuzzy logic.

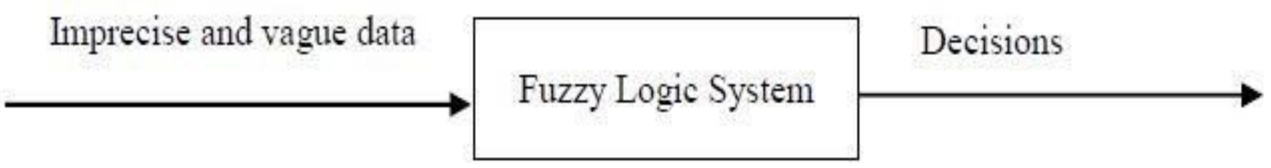


Figure 1: A fuzzy logic system accepting imprecise data and providing a decision

Contd..

- Fuzzy logic is the logic underlying approximate, rather than exact, modes of reasoning. It operates on the concept of membership. The membership was extended to possess various "degrees of membership" on the real continuous interval [0, l].

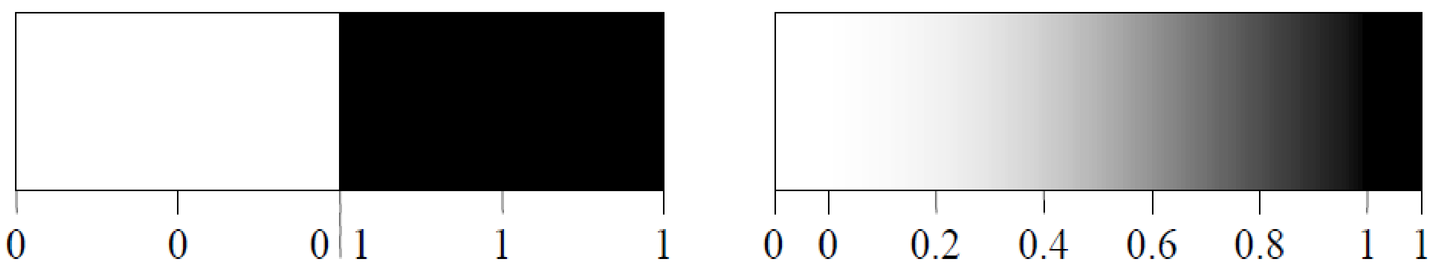


Figure 2: Crisp set vs fuzzy set

Contd..

- Fuzzy logic is the logic underlying approximate, rather than exact, modes of reasoning. It operates on the concept of membership.

- The membership was extended to possess various "degrees of membership" on the real continuous interval [0, l].

- In fuzzy systems, values are indicated by a number (called a truth value) ranging from 0 to l, where 0.0 represents absolute falseness and 1.0 represents absolute truth.

Example

- Words like young, tall, good or high are fuzzy.

· There is no single quantitative value which defines the term young.

· For some people, age 25 is young, and for others, age 35 is young.
· The concept young has no clean boundary.

· Age 35 has some possibility of being young and usually depends on the context in which it is being considered.

Crisp Vs Fuzzy

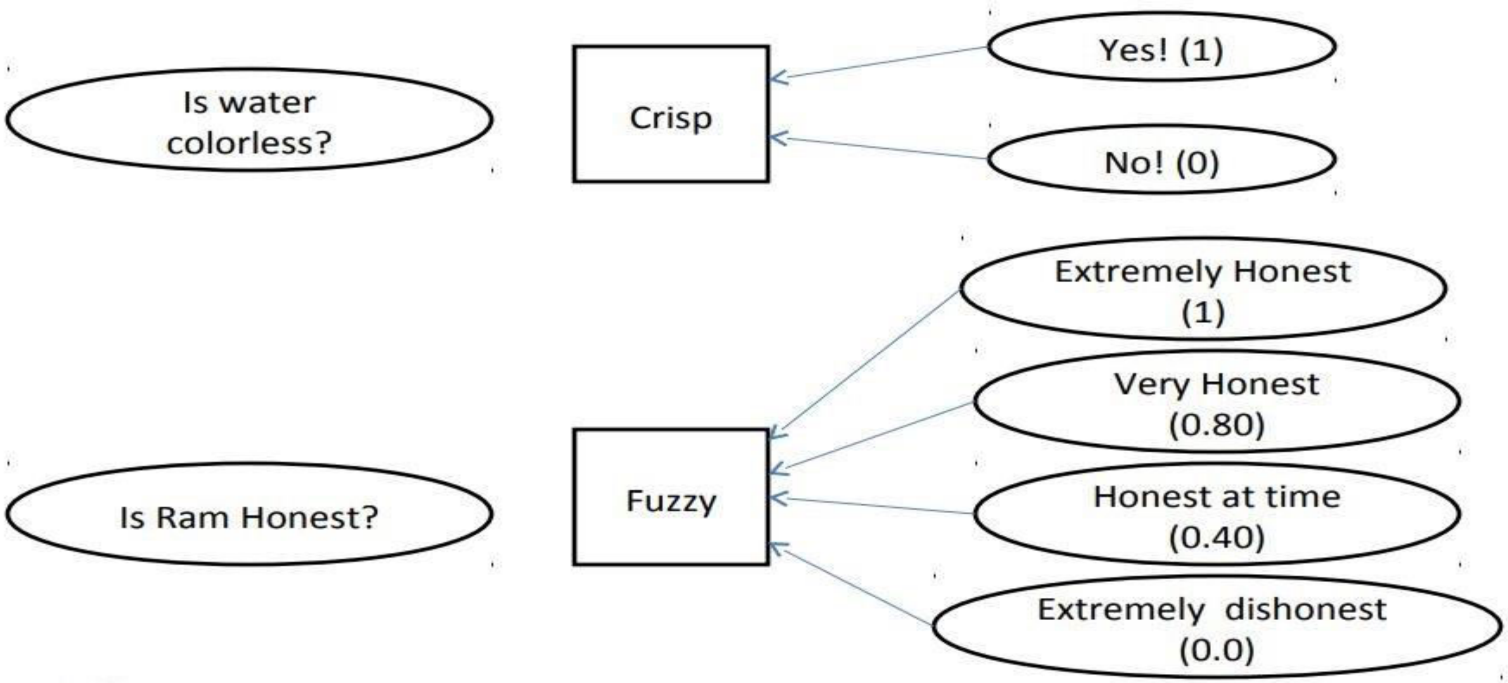


Figure 3: crisp vs fuzzy

fuzzy set

- A fuzzy set A in the universe of discourse U can be defined as

$$\underset{\sim}{A} = \left\{ \left(x, \mu_{\underset{\sim}{A}}(x) \right) \mid x \in X \right\}$$

where $\mu_{\underset{\sim}{A}}(x)$ is the degree of membership of x in $\underset{\sim}{A}$ and it indicates the degree that x belongs to $\underset{\sim}{A}$. In the fuzzy theory, fuzzy set A of universe X is defined by function $\mu_{\underset{\sim}{A}}(x)$ called the membership function of set A.

Contd..

$$\mu_{\underset{\sim}{A}}(x): X \rightarrow [0, 1], \qquad \text{where} \quad \begin{array}{ll} \mu_{\underset{\sim}{A}}(x) = 1 & \text{if } x \text{ is totally in } A; \\ \mu_{\underset{\sim}{A}}(x) = 0 & \text{if } x \text{ is not in } A; \\ 0 < \mu_{\underset{\sim}{A}}(x) < 1 & \text{if } x \text{ is partly in } A. \end{array}$$

Contd..

- Let X = {g1, g2, g3, g4, g5} be the reference set of students.

- Let Ã be the fuzzy set of "smart" students, where "smart" is fuzzy term.

  Ã = {(g1, 0.4), (g2, 0.5), (g3, 1), (g4, 0.9), (g5, 0.8)}

  Here Ã indicates that the smartness of g1 is 0.4 and so on

Membership Function

- A membership function provides a measure of the degree of similarity of an element to a fuzzy set.

- Membership function defines the fuzziness in a fuzzy set irrespective of the elements in the set, which are discrete or continuous.

- A fuzzy set '*A*' in the universe of discourse X can be defined as a set of ordered pairs:

$$\underset{\sim}{A} = \left\{ \left(x, \mu_{\underset{\sim}{A}}(x) \right) \mid x \in X \right\}$$

Contd..

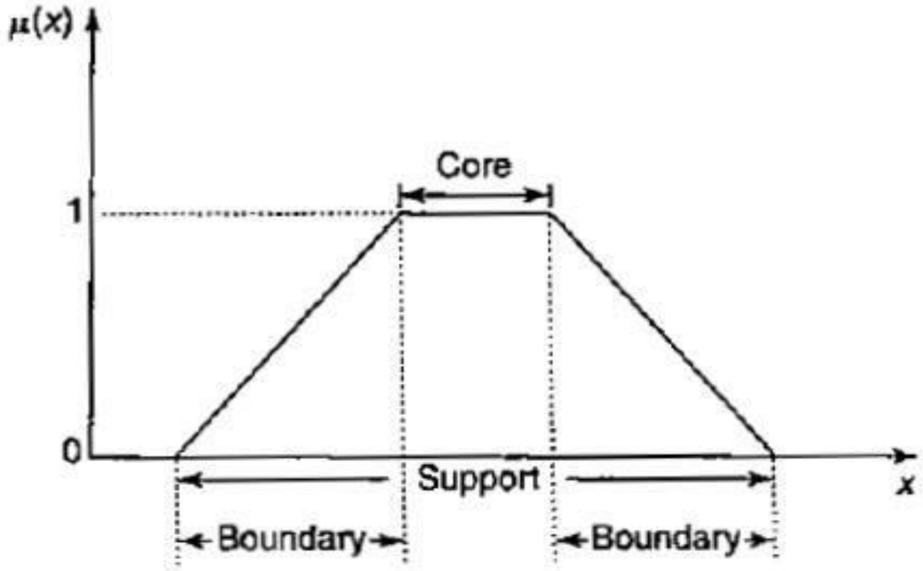


Figure 4. : Features of membership functions

Contd..

- **Core**

1) The core of a membership function for some fuzzy set *A* is defined as that region of universe that is characterized by complete membership in the set *A*

2) The core has elements x of the universe such that

$$\mu_A(x)=1$$

- **Support**

The support of a membership function for a fuzzy set *A*~ is defined as that region of universe that is characterized by a non zero membership in the set *A*.

$$\mu_A(x)>0$$

- Boundary

1) The support of a membership functions as the region of universe containing elements that have a non zero but not complete membership.

2) The boundary comprises those elements of x of the universe such that

$$0<\mu_A(x)<1$$

1) The boundary elements are those which possess partial membership in the fuzzy set $A\sim$

Steps: Fuzzy c means clustering

- Step 1: Initialize the data points into the desired number of clusters randomly.
- Step 2: Find out the centroid.
- Step 4: Updating membership values.
- Repeat the steps(2-4) until the constant values are obtained for the membership values or the difference is less than the tolerance value.

Problem 1

	(1, 3)	**(2, 5)**	**(4, 8)**	**(7, 9)**
Cluster 1	0.8	0.7	0.2	0.1
Cluster 2	0.2	0.3	0.8	0.9

Steps in Fuzzy c-means

Step 1: Initialize the data points into the desired number of clusters randomly.

- Consider, there are 2 clusters in which the data is to be divided, initializing the data point randomly.

- Each data point lies in both clusters with some membership value which can be assumed anything in the initial state.

Step 2: Find out the centroid.

The formula for finding out the centroid (V) is:

$$V_{ij} = \left(\sum_{1}^{n}(\gamma_{ik}^{m} * x_k) / \sum_{1}^{n} \gamma_{ik}^{m}\right.$$

Contd..

- C11 = (0.8^2 *1 + 0.7^2 * 2 + 0.2^2 * 4 + 0.1^2 * 7) / ((0.8^2 + 0.7^2 + 0.2^2 + 0.1^2) = **1.568**
- C12 = (0.8^2 *3 + 0.7^2 * 5 + 0.2^2 * 8 + 0.1^2 * 9) / ((0.8^2 + 0.7^2 + 0.2^2 + 0.1^2) = **4.051**
- C21 = (0.2^2 *1 + 0.3^2 * 2 + 0.8^2 * 4 + 0.9^2 * 7) / ((0.2^2 + 0.3^2 + 0.8^2 + 0.9^2) = **5.35**
- C22 = (0.2^2 *3 + 0.3^2 * 5 + 0.8^2 * 8 + 0.9^2 * 9) / ((0.2^2 + 0.3^2 + 0.8^2 + 0.9^2) = **8.215**
- **Centroid are:**

Step 3: Find out the distance of each point from the centroid.

- X11 = ((1 - 1.568) ^2 + (3 - 4.051) ^2) ^0.5 = 1.2
- X12 = ((1 - 5.35) ^2 + (3 - 8.215) ^2) ^0.5 = 6.79
- X21= 1.04
- X22= 4.64
- X31= 4.63
- X32= 1.36
- X41= 7.34
- X42= 1.82

Step 4: Updating membership values.

$$\gamma = \sum_{1}^{n} (d_{ki}^2 / d_{kj}^2)^{1/m-1}]^{-1}$$

Contd..

μ_{x11} = [{ [(1.2)2 / (1.2)2] + [(1.2)2 / (6.79)2]} ^ {(1 / (2 – 1))}] -1 = 0.96

μ_{x12} = [{ [(6.79)2 / (6.79)2] + [(6.79)2 / (1.2)2]} ^ {(1 / (2 – 1))}] -1 = 0.04

Final step

	(1, 3)	**(2, 5)**	**(4, 8)**	**(7, 9)**
Cluster 1	0.97	0.95	0.08	0.06
Cluster 2	0.03	0.05	0.92	0.94

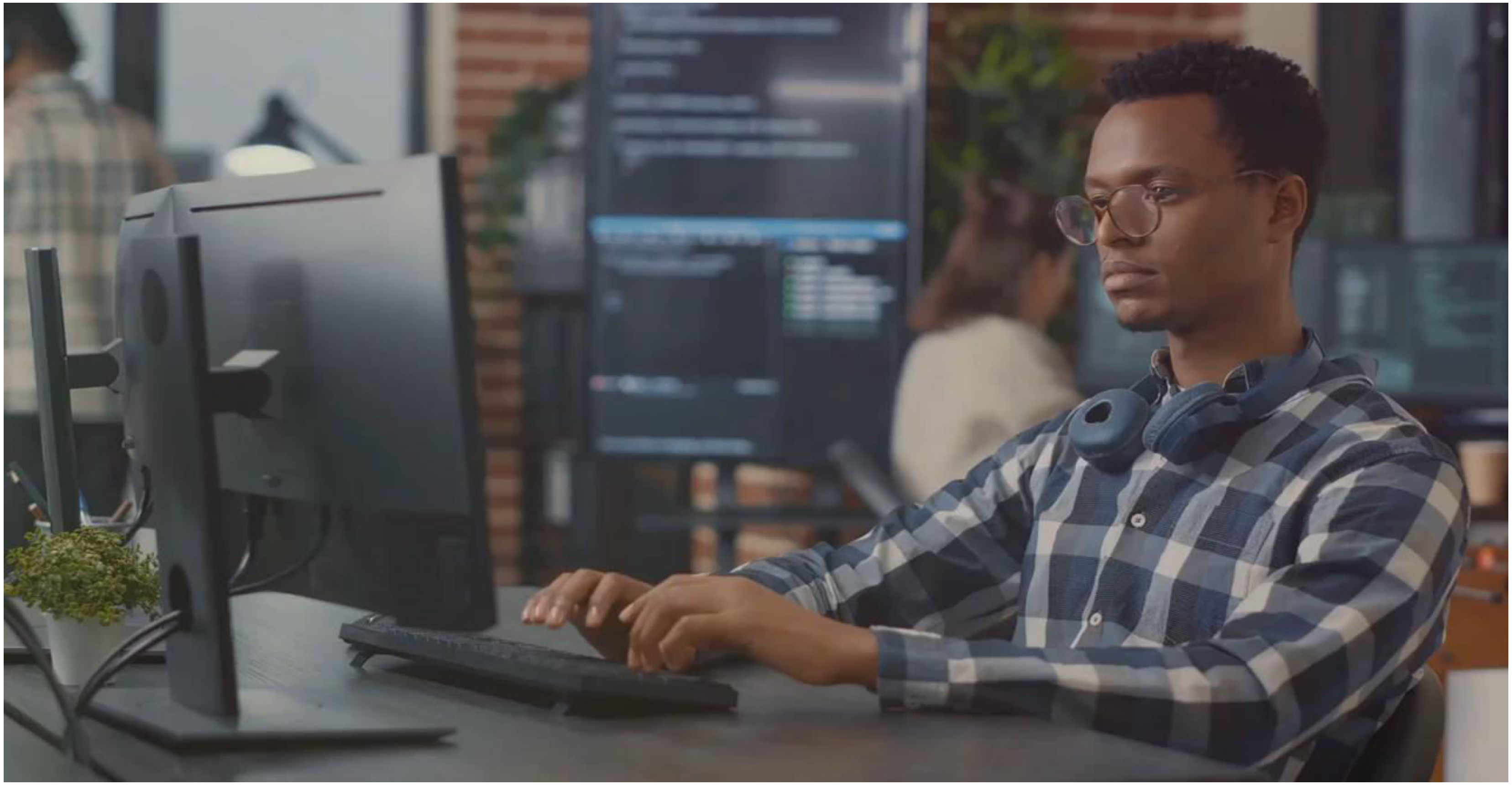
Lecture on

Expectation Maximization Algorithm

Shesh Narayan Sahu, CSE

Introduction

- Video link- https://www.youtube.com/watch?v=1u5RtLs9tK4



Introduction

- The expectation-Maximization algorithm can be used to handle situations where variables are partially observable.

- When certain variables are observable, we can use those instances to learn and estimate their values. Then, we can predict the values of these variables in instances when it is not observable.

- The EM algorithm was proposed and named in a seminal paper published in 1977 by Arthur Dempster, Nan Laird, and Donald Rubin.

- EM algorithm is applicable to latent variables, which are variables that are not directly observable but are inferred from the values of other observed variables.

Contd..

- The EM algorithm serves as the foundation for many unsupervised clustering algorithms in the field of machine learning.

- It provides a framework to find the local maximum likelihood parameters of a statistical model and infer latent variables in cases where data is missing or incomplete.

Application: EM

- It can be used to fill in the missing data in a sample.
- It can be used as the basis of unsupervised learning of clusters.
- It can be used for the purpose of estimating the parameters of the Hidden Markov Model (HMM).
- It can be used for discovering the values of latent variables.
- Parameter of mixtures of Gaussian (MoG).

Definition: EM

- The Expectation-Maximization (EM) algorithm is an iterative optimization method.

- Expectation-Maximization (EM) algorithm is an iterative optimization that can derive the maximum likelihood estimates in the presence of missing or hidden data.

- It combines different unsupervised machine learning algorithms to find maximum likelihood or maximum posterior estimates of parameters in statistical models that involve unobserved latent variables.

Definition: EM

- The Expectation-Maximization (EM) algorithm is an iterative optimization method that combines different unsupervised machine learning algorithms to find maximum likelihood or maximum posterior estimates of parameters in statistical models that involve unobserved latent variables.

- It consists of an estimation step (E-step) and a maximization step (M-step), forming an iterative process to improve model fit.
  - In the E step, the algorithm computes the latent variables i.e. expectation of the log-likelihood using the current parameter estimates.
  - In the M step, the algorithm determines the parameters that maximize the expected log-likelihood obtained in the E step, and corresponding model parameters are updated based on the estimated latent variables.

General steps

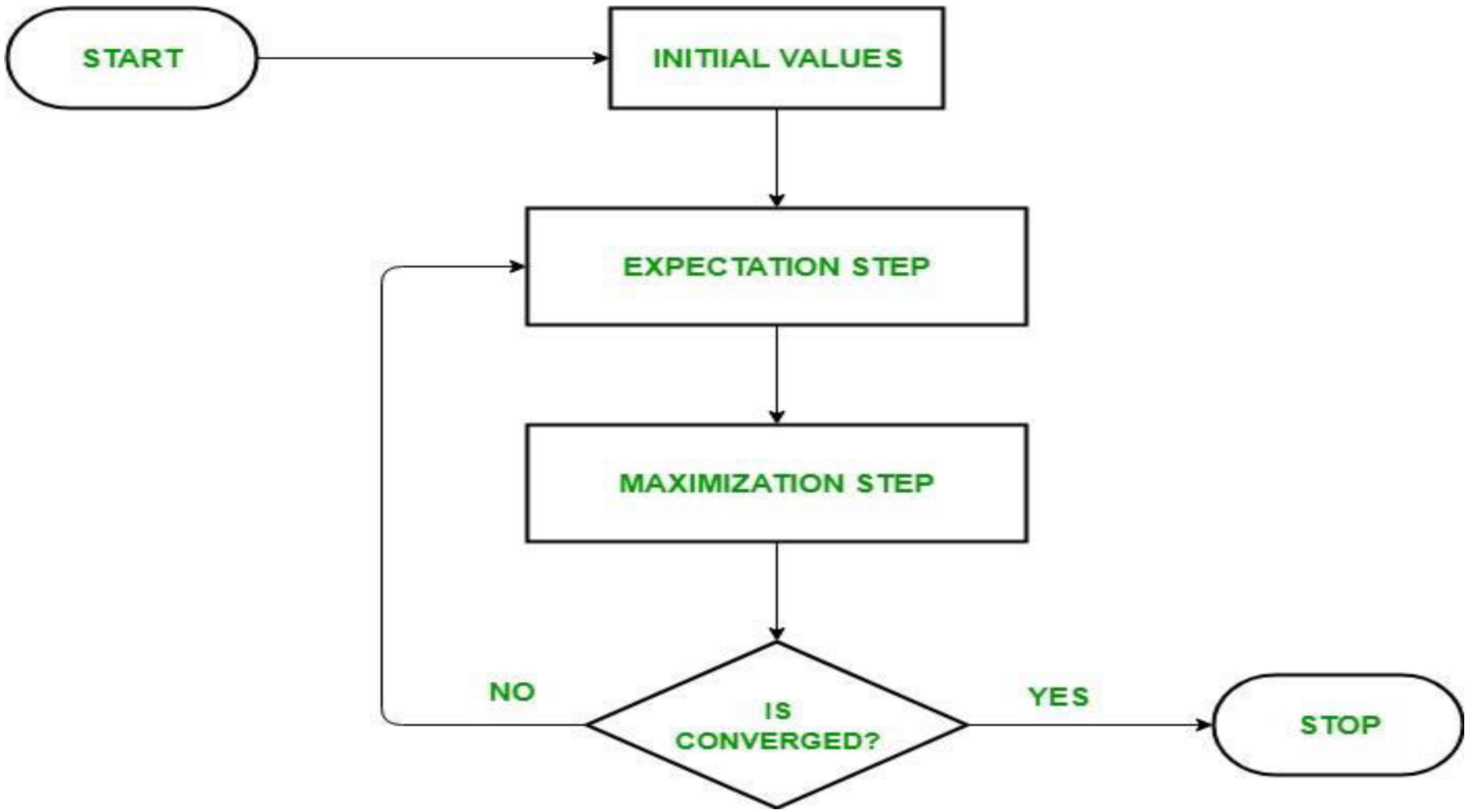
- Consider a set of starting parameter-
  - Given s set of incomplete (observed) data
  - Assume observed data come from a specific model.

- Use these estimates the missing data formulate some parameters for that model, use this guess for the missing value/data (expectation step).

- Use complete data to update parameter form the missing data and observed data, find the most likely parameter (maximization step).

- Repeat step 2 and 3 until convergence.

Working: EM


START
INITIIAL VALUES
EXPECTATION STEP
MAXIMIZATION STEP
IS
CONVERGED?
NO
YES
STOP

EM Algorithm

Step 1: Initialization

Initially, a set of initial values of the parameters are considered. A set of incomplete observed data is given to the system with the assumption that the observed data comes from a specific model.

Step 2: E-Step (Expectation Step):

In this step, we use the observed data in order to estimate or guess the values of the missing or incomplete data. It is basically used to update the variables.

- Compute the posterior probability or responsibility of each latent variable given the observed data and current parameter estimates.
- Estimate the missing or incomplete data values using the current parameter estimates.

Contd..

Step 3: M-step (Maximization Step):

In this step, we use the complete data generated in the preceding "Expectation" – step in order to update the values of the parameters. It is basically used to update the hypothesis.

- Update the parameters of the model by maximizing the expected complete data log-likelihood obtained from the E-step.
- This typically involves solving optimization problems to find the parameter values that maximize the log-likelihood.
- The specific optimization technique used depends on the nature of the problem and the model being used.

Contd..

Step 4: Convergence:

In this step, it is checked whether the values are converging or not, if yes, then stop otherwise repeat step-2 and step-3 i.e. “Expectation” – step and “Maximization” – step until the convergence occurs.

- Check for convergence by comparing the change in log-likelihood or the parameter values between iterations.
- If the change is below a predefined threshold, stop and consider the algorithm converged.
- Otherwise, go back to the E-step and repeat the process until convergence is achieved.

Key Terms:

- **Latent Variables**: Latent variables are unobserved variables in statistical models that can only be inferred indirectly through their effects on observable variables. They cannot be directly measured but can be detected by their impact on the observable variables.
- **Likelihood**: It is the probability of observing the given data given the parameters of the model. In the EM algorithm, the goal is to find the parameters that maximize the likelihood.
- **Maximum Likelihood Estimation (MLE):** MLE is a method to estimate the parameters of a statistical model by finding the parameter values that maximize the likelihood function, which measures how well the model explains the observed data.
- **Posterior Probability:** In the context of Bayesian inference, the EM algorithm can be extended to estimate the maximum a posteriori (MAP) estimates, where the posterior probability of the parameters is calculated based on the prior distribution and the likelihood function.

Contd..

- **Expectation (E) Step**: The E-step of the EM algorithm computes the expected value or posterior probability of the latent variables given the observed data and current parameter estimates. It involves calculating the probabilities of each latent variable for each data point.
- **Maximization (M) Step**: The M-step of the EM algorithm updates the parameter estimates by maximizing the expected log-likelihood obtained from the E-step. It involves finding the parameter values that optimize the likelihood function, typically through numerical optimization methods.
- **Convergence:** Convergence refers to the condition when the EM algorithm has reached a stable solution. It is typically determined by checking if the change in the log-likelihood or the parameter estimates falls below a predefined threshold.

Example

- Consider two coins A and B, tossed 10 times each

  B: H T T T H H T H T H

  A: H H H H T H H H H H

  A: H T H H H H H T H H

  B: H T H T T T H H T T

  A: T H H H T H H H T H

- θ_1 and θ_2 be probability of getting head in coin A and coin B respectively. Find the value of θ_1 and θ_2 ?

Solution: EM

If we know the coin labels the probability will be as follows:

Coin A	Coin B
	5 H, 5 T
9 H, 1 T	
8 H, 2 T	
	4 H, 6 T
7 H, 3 T	

$$\Theta_1 = 24/(24+6) = 0.8$$

$$\Theta_2 = 9/(9+11) = 0.45$$

Contd..

$$L(C)=\Theta^k (1-\Theta)^{n-k}$$

Likelihood For first coin Flips

$$L(A)= 0.6^5 (1-0.6)^{10-5} = 0.0007963$$

$$L(B)= 0.5^5 (1-0.5)^{10-5} = 0.0009766$$

Contd..

$L(C)=\Theta^k(1-\Theta)^{n-k}$

Likelihood For first coin Flips

$L(A)= 0.6^5 (1- 0.6)^{10-5} = 0.0007963$

$L(B)= 0.5^5 (1- 0.5)^{10-5} = 0.0009766$

$P(A)=L(A)/L(A)+L(B) = 0.0007963/(0.0007963+0.0009766)=0.45$

$P(B)=L(B)/L(A)+L(B) = 0.0009766/(0.0007963+0.0009766) = 0.55$

Contd..

In similar fashion find probability of all coins with all flips. It will be as follows:

L(H): Likely no of heads L(T): Likely no of tails

												Coin A		Coin B	
										Iteration 1->:					
										P(A)	P(B)	L(H)	L(T)	L(H)	L(T)
B	H	T	T	T	H	H	T	H	T	H	0.45	0.55	2.2	2.2	2.8	2.8
A	H	H	H	H	T	H	H	H	H	H	0.80	0.20	7.2	0.8	1.8	0.2
A	H	T	H	H	H	H	H	T	H	H	0.73	0.27	5.9	1.5	2.1	0.5
B	H	T	H	T	T	T	H	H	T	T	0.35	0.65	1.4	2.1	2.6	3.9
A	T	H	H	H	T	H	H	H	T	H	0.65	0.35	4.5	1.9	2.5	1.1

Contd..

For Coin A:

$\sum L(H) = 21.3$

$\sum L(T) = 8.6$

$\Theta_1 = 21.3/(21.3+8.6)$

$= 0.71$

For Coin B:

$\sum L(H) = 11.7$

$\sum L(T) = 8.4$

$\Theta_2 = 11.7/(11.7+8.4)$

$= 0.58$

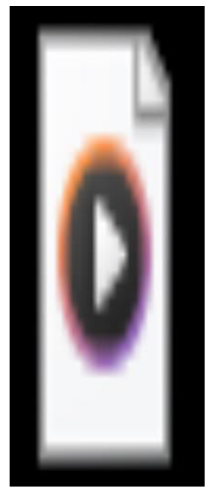
Lecture on

Expectation Maximization Algorithm

Shesh Narayan Sahu, CSE

Introduction

- Video link- [click here to watch]



Semi-supervised Learning explained.mp4

Introduction

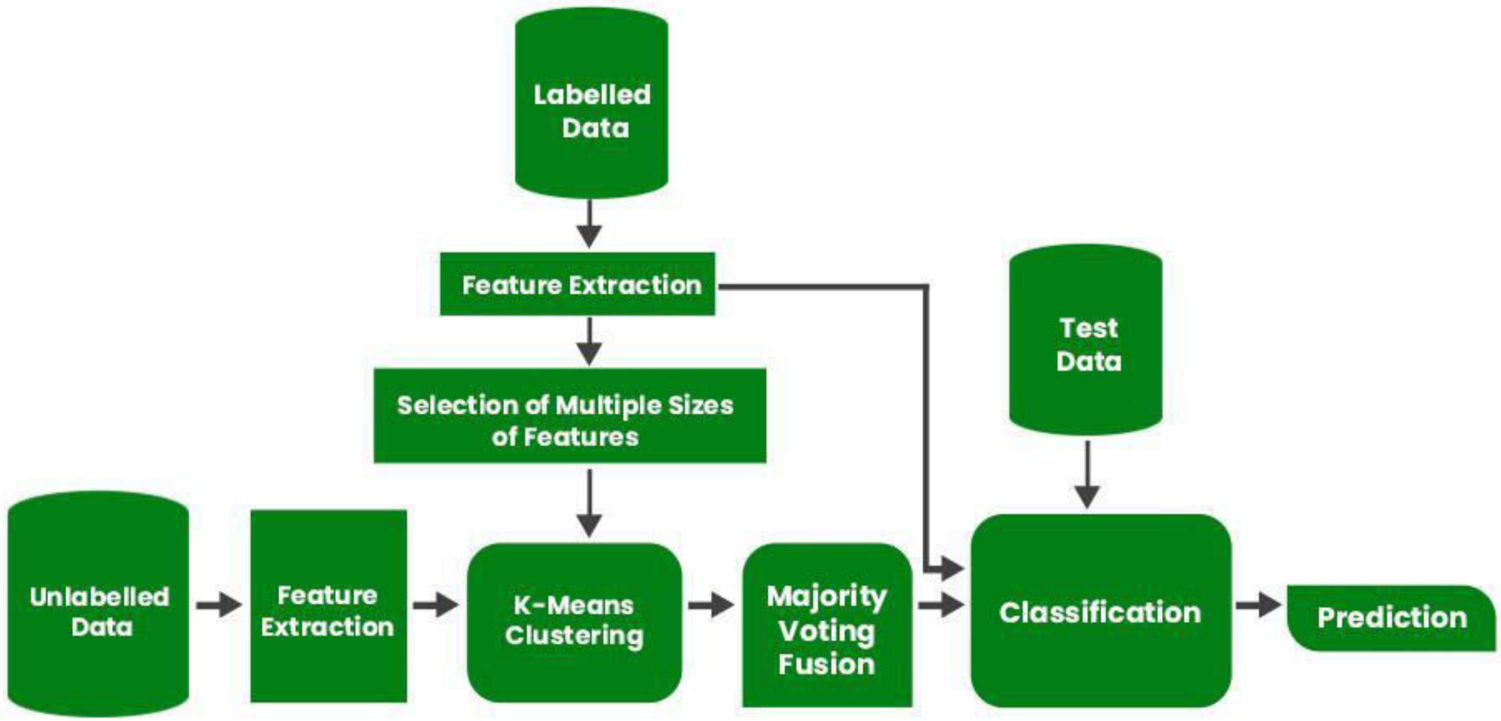
- Semi-supervised learning is a type of machine learning that falls in between supervised and unsupervised learning.

- It is a method that uses a small amount of labeled data and a large amount of unlabeled data to train a model.

- The goal of semi-supervised learning is to learn a function that can accurately predict the output variable based on the input variables, similar to supervised learning.

- Semi-supervised learning is particularly useful when there is a large amount of unlabeled data available, but it's too expensive or difficult to label all of it.

Schematic diagram



Application: Semi-Supervised Learning

- **Text classification**: In text classification, the goal is to classify a given text into one or more predefined categories. Semi-supervised learning can be used to train a text classification model using a small amount of labeled data and a large amount of unlabeled text data.

- **Image classification:** In image classification, the goal is to classify a given image into one or more predefined categories. Semi-supervised learning can be used to train an image classification model using a small amount of labeled data and a large amount of unlabeled image data.

- **Anomaly detection:** In anomaly detection, the goal is to detect patterns or observations that are unusual or different from the norm

Assumption: SSL

- **Continuity Assumption:**
  - As per the continuity assumption, the objects near each other tend to share the same group or label.
  - This assumption is also used in supervised learning, and the datasets are separated by the decision boundaries. But in semi-supervised, the decision boundaries are added with the smoothness assumption in low-density boundaries.

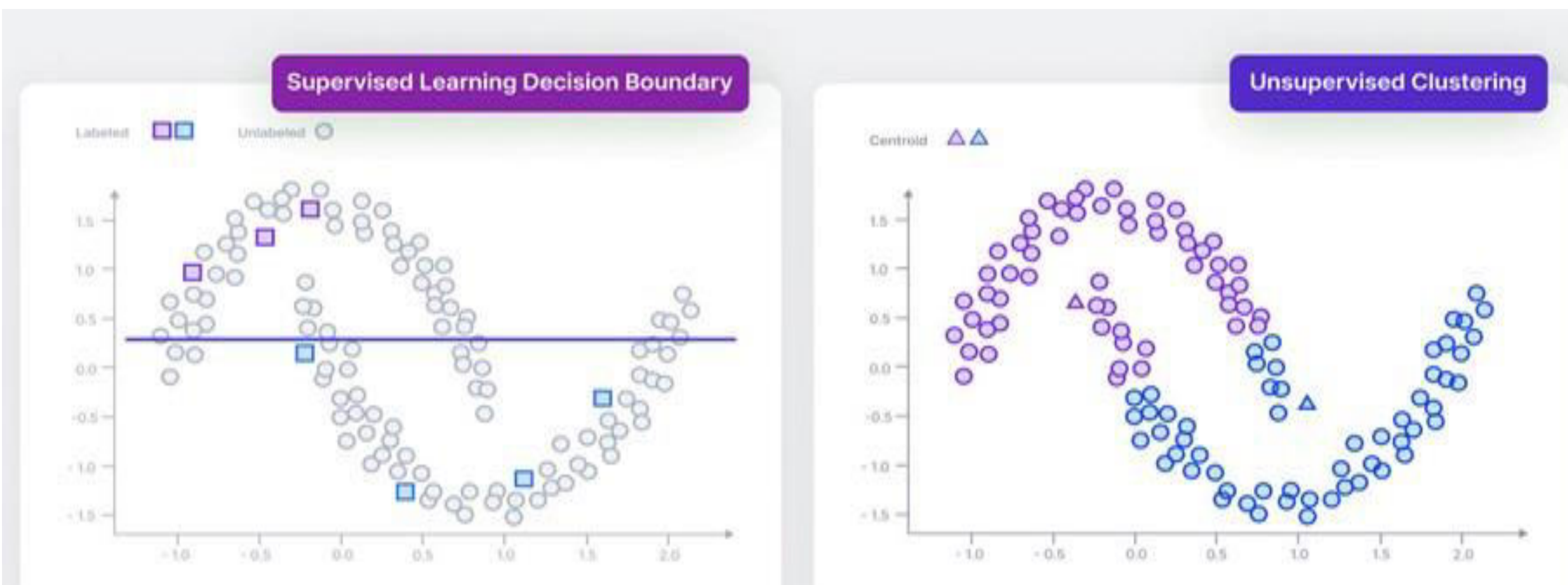
- **Cluster assumptions-**
  - In this assumption, data are divided into different discrete clusters. Further, the points in the same cluster share the output label.

Contd..

- **Manifold assumptions-**
  - It assumes the high-dimensional data distribution can be represented in an embedded low-dimensional space. This low-dimensional space is called the data manifold.
  - This assumption helps to use distances and densities, and this data lie on a manifold of fewer dimensions than input space.

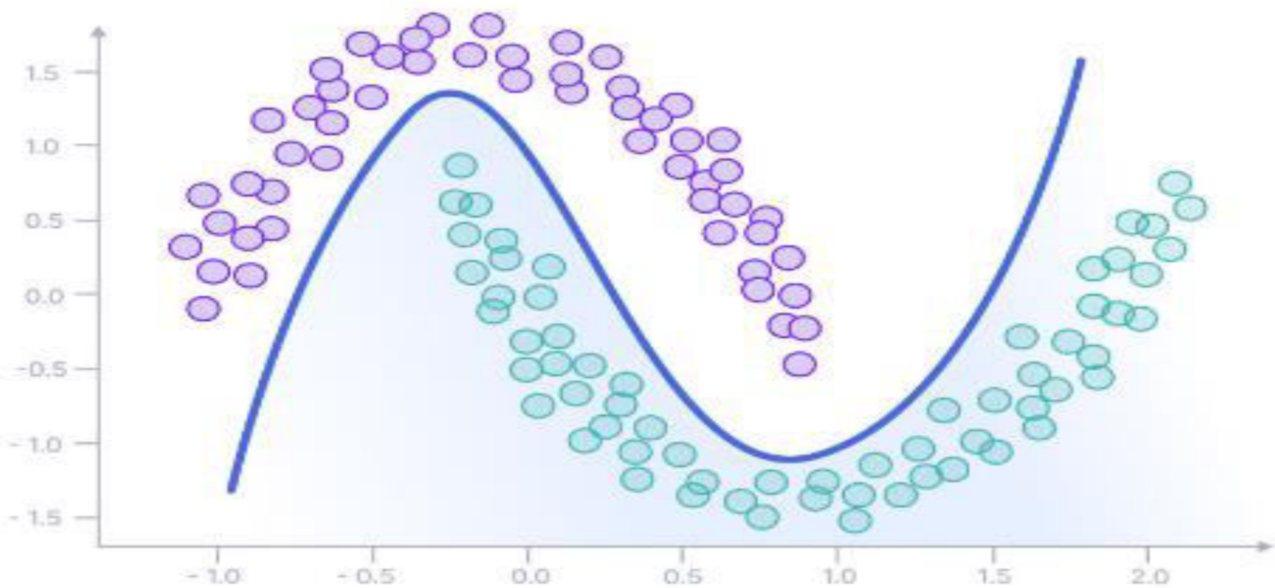
Contd..

- Supervised vs unsupervised learning.
- Without sufficient labeled data, or in difficult clustering settings, supervised and unsupervised techniques can fail to achieve the desired result.



General steps

- In the semi-supervised setting, however, we use both labeled and unlabeled data.

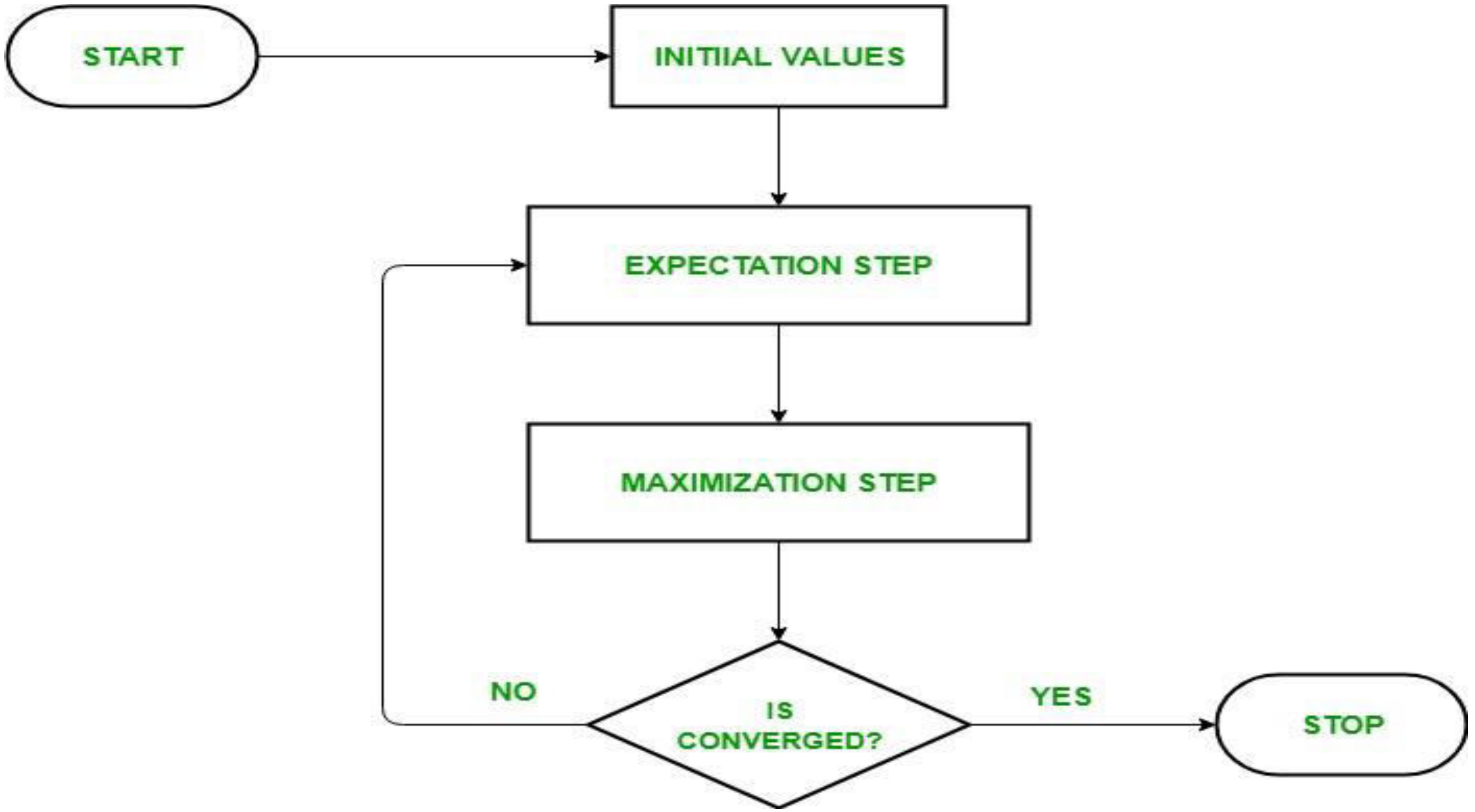


Key takeaways

- Labeled datapoints are handled as in traditional supervised learning; predictions are made, loss is calculated, and network weights are updated by gradient descent.

- Unlabeled datapoints are used to help the model make more consistent and confident predictions. Whether by an added unsupervised loss term or by pseudo-labels, unlabeled examples are used to build upon the progress of labeled examples.

Working: EM


START
INITIIAL VALUES
EXPECTATION STEP
MAXIMIZATION STEP
IS
CONVERGED?
NO
YES
STOP

EM Algorithm

Step 1: Initialization

Initially, a set of initial values of the parameters are considered. A set of incomplete observed data is given to the system with the assumption that the observed data comes from a specific model.

Step 2: E-Step (Expectation Step):

In this step, we use the observed data in order to estimate or guess the values of the missing or incomplete data. It is basically used to update the variables.

- Compute the posterior probability or responsibility of each latent variable given the observed data and current parameter estimates.
- Estimate the missing or incomplete data values using the current parameter estimates.

Contd..

Step 3: M-step (Maximization Step):

In this step, we use the complete data generated in the preceding "Expectation" – step in order to update the values of the parameters. It is basically used to update the hypothesis.

- Update the parameters of the model by maximizing the expected complete data log-likelihood obtained from the E-step.
- This typically involves solving optimization problems to find the parameter values that maximize the log-likelihood.
- The specific optimization technique used depends on the nature of the problem and the model being used.

Contd..

Step 4: Convergence:

In this step, it is checked whether the values are converging or not, if yes, then stop otherwise repeat step-2 and step-3 i.e. "Expectation" – step and "Maximization" – step until the convergence occurs.

- Check for convergence by comparing the change in log-likelihood or the parameter values between iterations.
- If the change is below a predefined threshold, stop and consider the algorithm converged.
- Otherwise, go back to the E-step and repeat the process until convergence is achieved.

Key Terms:

- **Latent Variables**: Latent variables are unobserved variables in statistical models that can only be inferred indirectly through their effects on observable variables. They cannot be directly measured but can be detected by their impact on the observable variables.
- **Likelihood**: It is the probability of observing the given data given the parameters of the model. In the EM algorithm, the goal is to find the parameters that maximize the likelihood.
- **Maximum Likelihood Estimation (MLE):** MLE is a method to estimate the parameters of a statistical model by finding the parameter values that maximize the likelihood function, which measures how well the model explains the observed data.
- **Posterior Probability:** In the context of Bayesian inference, the EM algorithm can be extended to estimate the maximum a posteriori (MAP) estimates, where the posterior probability of the parameters is calculated based on the prior distribution and the likelihood function.

Contd..

- **Expectation (E) Step**: The E-step of the EM algorithm computes the expected value or posterior probability of the latent variables given the observed data and current parameter estimates. It involves calculating the probabilities of each latent variable for each data point.
- **Maximization (M) Step**: The M-step of the EM algorithm updates the parameter estimates by maximizing the expected log-likelihood obtained from the E-step. It involves finding the parameter values that optimize the likelihood function, typically through numerical optimization methods.
- **Convergence:** Convergence refers to the condition when the EM algorithm has reached a stable solution. It is typically determined by checking if the change in the log-likelihood or the parameter estimates falls below a predefined threshold.

Example

- Consider two coins A and B, tossed 10 times each

B: H T T T H H T H T H

A: H H H H T H H H H H

A: H T H H H H H T H H

B: H T H T T T H H T T

A: T H H H T H H H T H

- θ_1 and θ_2 be probability of getting head in coin A and coin B respectively. Find the value of θ_1 and θ_2 ?

Solution: EM

If we know the coin labels the probability will be as follows:

Coin A	Coin B
	5 H, 5 T
9 H, 1 T	
8 H, 2 T	
	4 H, 6 T
7 H, 3 T	

$$\Theta_1 = 24/(24+6) = 0.8$$

$$\Theta_2 = 9/(9+11) = 0.45$$

Contd..

$$L(C) = \Theta^k (1 - \Theta)^{n-k}$$

Likelihood For first coin Flips

$$L(A) = 0.6^5 (1 - 0.6)^{10-5} = 0.0007963$$

$$L(B) = 0.5^5 (1 - 0.5)^{10-5} = 0.0009766$$

$L(C) = \Theta^k (1 - \Theta)^{n-k}$

Likelihood For first coin Flips

$L(A) = 0.6^5 (1 - 0.6)^{10-5} = 0.0007963$

$L(B) = 0.5^5 (1 - 0.5)^{10-5} = 0.0009766$

$P(A) = L(A)/L(A) + L(B) = 0.0007963/(0.0007963 + 0.0009766) = 0.45$

$P(B) = L(B)/L(A) + L(B) = 0.0009766/(0.0007963 + 0.0009766) = 0.55$

Contd..

In similar fashion find probability of all coins with all flips. It will be as follows:

L(H): Likely no of heads L(T): Likely no of tails

													Coin A		Coin B	
										Iteration 1->:						
											P(A)	P(B)	L(H)	L(T)	L(H)	L(T)
B	H	T	T	T	H	H	T	H	T	H	0.45	0.55	2.2	2.2	2.8	2.8
A	H	H	H	H	T	H	H	H	H	H	0.80	0.20	7.2	0.8	1.8	0.2
A	H	T	H	H	H	H	H	T	H	H	0.73	0.27	5.9	1.5	2.1	0.5
B	H	T	H	T	T	T	H	H	T	T	0.35	0.65	1.4	2.1	2.6	3.9
A	T	H	H	H	T	H	H	H	T	H	0.65	0.35	4.5	1.9	2.5	1.1

Contd..

For Coin A:

$\sum L(H) = 21.3$

$\sum L(T) = 8.6$

$\Theta_1 = 21.3/(21.3+8.6)$

$= 0.71$

For Coin B:

$\sum L(H) = 11.7$

$\sum L(T) = 8.4$

$\Theta_2 = 11.7/(11.7+8.4)$

$= 0.58$