Subject: ML
Faculty: S. Sandhya Rani
Topic: Introduction to clustering

Class Notes

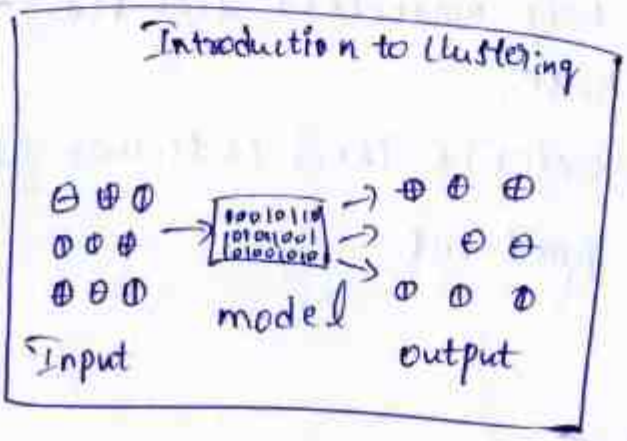
Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No.... of SP
Book Reference:
Date Conducted:
Page No: 1

Introduction: problem Definition & clustering Overview

clustering is the task of dividing the population (or) data points into a no. of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups.

In simple words, the aim is to segregate groups with similar traits and assign them into clusters.

Example:
Suppose, you are the head of a rental store and wish to understand preferences of your customers to scale up your business. is it possible for you to look at details of each customer and devise a unique business strategy for each one of them?



what is good clustering

A good clustering method will produce high quality clusters with
- hig intra-class similarity
- Low intra-class similarity

- The quality of a clustering result depends on both the similarity measure used by the methodand its implementation

Applications of clustering

- clustering analysis is broadly used in many applications such as market research, patternrecognition, data analysis and image processing.
- clustering also helps in classifying documents on the web for information discovery.
- clustering is also used in outlier detection applications such as detection of credit card fraud.
- clustering can also help marketers discover distinct groups in their customer base.
  And they can characterize their customer groups based on the purchasing patterns.

Subject: ML
Faculty: S. Sandhya Rani
Topic: Hierarchical clustering

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No.... of SP
Book Reference:
Date Conducted:
Page No: 2

Hierarchical Clustering:

HC is another unsupervised ML algorithm, which is used to group the unlabeled dataset into a cluster and also known as hierarchical cluster analysis (or) HCA.

In this algorithm, we develop the hierarchy of clusters in the form of a tree, and this tree-shaped structure is known as the dendrogram.

Sometimes the result of k-means clustering and hierarchical clustering may look similar, but they both differ depending on how they work. As there is no requirement to predetermine the no. of clusters as we did in the k-means algorithm.

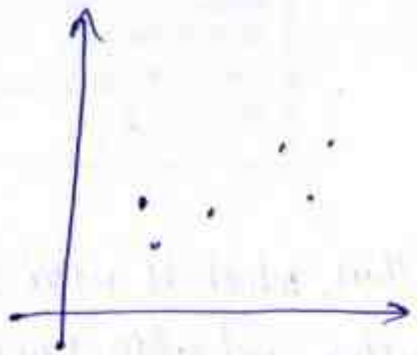
1. Agglomerative:

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

2. Divisive:

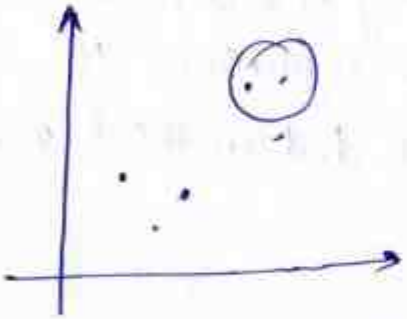
Divisive algorithm is the reverse of the agglomerative algorithm as it is a top-down approach.

How the Agglomerative Hierarchical clustering work:

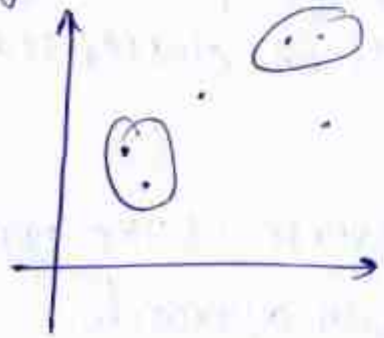


Step1: create each data point as a single cluster. Let's say there are N data points, so the no. of clusters will also be N.

Step 2: Take two closest data points (or) clusters and merge them to form one cluster. So, there will now be N-1 clusters.



Step3: Again, take the two closest clusters and merge them together to form one cluster. There will be N-2 clusters.

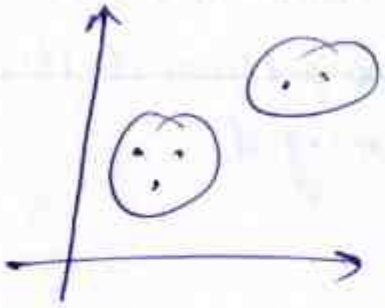


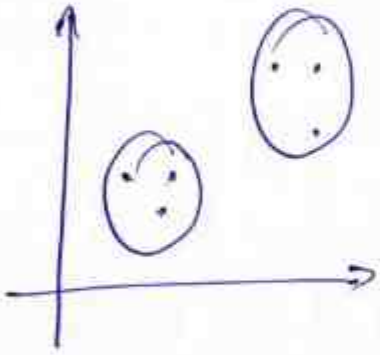
Subject: ML
Faculty: S. Sandhya Rani
Topic: ~~AGNES~~ AGNES

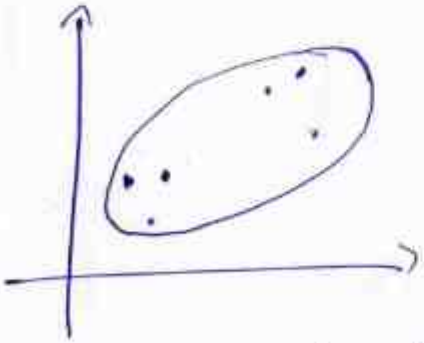
Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S No. of SP
Book Reference:
Date Conducted:
Page No: 3

Step 4: Repeat step 3 until only one cluster left, so, will get the following clusters. consider the







Step 5: once all the clusters are combined into one big cluster, develop the dendrogram to divide the clusters as per the problem.

Aglomerative Nesting hierarchical Clustering algorithm

AGNES

AGNES algorithm is all about hierarchical clustering it treats every data as a cluster and Gradually merges those clusters according to some certain criteria.

ex: two different clusters are merged.

Steps in AGNES clustering

1. Initialization
2. Distance calculation
3. cluster merging
4. Itration
5. Dendogram Interpretation

Advantages of AGNES:

1. Intuitive visualization
2. NO Assumption of cluster shape

Limitations:

1. Computational complexity
2. sensitivity to Noise

Subject: ML
Faculty: S. Sandhya Rani
Topic: DIANA, K-means clustering

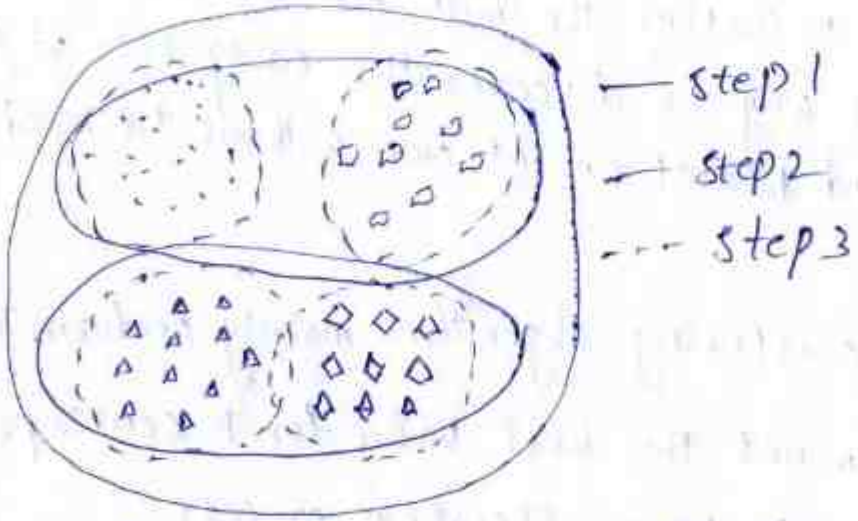
Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No of SP
Book Reference:
Date Conducted:
Page No: 9

DIANA (Divise ANALysis clustering Algorithm)

DIANA is also known as Divise Analysis clustering algorithm. it is the top-down approach form of hierarchical clustering where all data points are initially assigned a single cluster.

Further, the clusters are split into two least similar clusters.



National Representation of DIANA:

K-Means clustering

It is a unsupervised learning algorithm that is used to solve the clustering problems in ML (or) data science.

K-means clustering is a unsupervised learning algorithm which groups the unlabeled dataset into different clusters. Here K defines the no.of predefined 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.

* It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

The K-means clustering algorithm mainly perform two tasks. * Determines the best value for K centerpoints (or) centroids by an iterative process

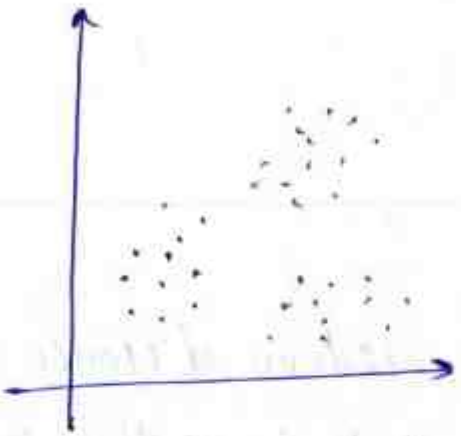
* Assigns each data point to its closest k-center. those datapoints which are near to the particular k-center, create a cluster.

Subject: ML
Faculty: S. Sandhya Rani
Topic: K-means clustering

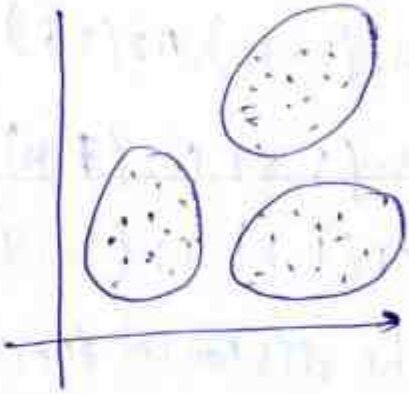
Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No ... of SP
Book Reference:
Date Conducted:
Page No: 5

Before K-means



After K-means



How does the K-means Algorithm work.

Step1: select the number K to decide the number of clusters.

Step2: select random K points (or) centroids.

Step3: Assign each data point to their closest centroid, which will form the predefined K-clusters.

Step4: calculate the variance and place a new centroid of each cluster.

Step5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

Step6: if any reassignment occurs, then go to step-4

Example:

- Suppose that the data mining task is to cluster points into three clusters.
- where the points are

$A_1(2, 10), A_2(2, 5), A_3(8, 4)$

$B_1(5, 8), B_2(7, 5), B_3(6, 4)$

$C_1(1, 2), C_2(4, 9)$

- The distance function is Euclidean distance
- Suppose initially we assign A_1, B_1, C_1 as the center of each cluster, respectively.

Data points			Distance to A_1 (2, 10)	Distance to B_1 (5, 8)	Distance to C_1 (1, 2)	cluster	NEW cluster
A_1	2	10	0.00	3.61	8.06	1	
A_2	2	5	5.00	4.24	3.16	3	
A_3	8	4	8.49	5.00	7.28	2	
B_1	5	8	3.61	0.00	7.21	2	
B_2	7	5	7.07	3.61	6.71	2	
B_3	6	4	7.21	4.12	5.39	2	
C_1	1	2	8.06	7.21	0.00	3	
C_2	4	9	2.24	1.41	7.62	2	

Subject: ML
Faculty: S. Sandhya Rani
Topic: K-means Example

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No ... of SP
Book Reference:
Date Conducted:
Page No: 6

Initial centroids:

$A_1 (2, 10)$

$B_1 (5, 8)$

$C_1 (1, 2)$

A1: $d(\vec{P_1}, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ (points)

$= \sqrt{(2-2)^2 + (10-10)^2}$

$= 0$

x-values data points

$x_1 = 2$
$y_1 = 10$
$x_2 = 2$
$y_2 = 10$

A2: $d(\vec{P_1}, P_3) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

$= \sqrt{(2-2)^2 + (10-5)^2}$

$= \sqrt{0 + (5)^2}$

$= 5$

$A_2 = (2, 5)$
$x_1 = 2$
$y_1 = 5$
$x_2 = 2$
$y_2 = 10$

A3: $= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

$= \sqrt{(2-8)^2 + (10-4)^2}$

$= \sqrt{(6)^2 + (6)^2} = \sqrt{72} = 8.4$

$A_3 (8, 4)$
$x_1 = 8$
$y_1 = 4$
$x_2 = 2$ $y_2 = 10$

$B_1 = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$

$\sqrt{(5-5)^2 + (8-8)^2}$

$\sqrt{0} = 0.$

$B_1(5,8)$ centroid $= (5,8)$

$x_1 = 5$

$y_1 = 8$

$x_2 = 5$

$y_2 = 8$

$B_2 = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$

$\sqrt{(5-7)^2 + (8-5)^2}$

$\sqrt{(2)^2 + (3)^2}$

$\sqrt{4+9} \Rightarrow \sqrt{13} \Rightarrow 3.61$

$B_2(7,5)$

$x_1 = 7$

$y_1 = 5$

$x_2 = 5$

$y_2 = 8$

centroid $(5,8)$

$B_3 = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$

$= \sqrt{(5-6)^2 + (8-4)^2}$

$= \sqrt{(1)^2 + (4)^2}$

$= \sqrt{1+16}$

$= \sqrt{17} = 4.12$

$B_3 = 6,4$

$x_1 = 6$

$y_1 = 4$

$x_2 = 5$

$y_2 = 8$

centroid $(5,8)$

Subject: ML
Faculty: S. Sandhya Rani
Topic: k-means Example

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No. of SP
Book Reference:
Date Conducted:
Page No: 7

$$C_1 = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$$

$$\sqrt{(1-1)^2 + (2-2)^2}$$

$$= 0$$

$$C_2 = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$$

$$= \sqrt{(1-4)^2 + (2-9)^2}$$

$$= \sqrt{(-3)^2 + (-7)^2}$$

$$= \sqrt{9+49}$$

$$= 7.62$$

$C_1 = (1,2)$ centroid $(1,2)$

$x_1 = 1$
$y_1 = 2$
$x_2 = 1$
$y_2 = 2$

$C_2 = (4,9)$ centroid $(1,2)$

$x_1 = 4$
$y_1 = 9$
$x_2 = 1$
$y_2 = 2$

=) Now we can take low value for cluster column
we have 3 iteration like 1, 2, 3
compare both then collect low value.

1, 3, 2, 2, 2, 2, 3, 2

For New clusters:

Given Data points

cluster 1 = $A1 = (2, 10)$

cluster 2 = $A3 = (8, 4)$

$B_1 = (5, 8)$

$B_2 = (7, 5)$

$B_3 = (6, 4)$

$C_2 = (4, 9)$

cluster 3 = $A_2 = (2, 5)$

$C_1 = (1, 2)$

Calculate the centroid:

cluster 1: A1 (2, 10)

cluster-1 only has one point so, centroid will be that point itself.

centroid of cluster 1 = (2, 10)

cluster 2: A3 - (8, 4), B1 - (5, 8), B2 (7, 5), B3 (6, 4)

C2 (4, 9)

For cluster 2, we have the following points:

$A_3 - (8,4)$

$B_1 - (5,8)$

$B_2 - (7,5)$

$B_3 - (6,4)$

$C_2 - (4,9)$

Calculate the centroid Of Cluster-2

Sum of x-coordinates:

$$8+5+7+6+4 = 30$$

sum of y-coordinates:

$$4+8+5+4+9 = 30$$

$$\left(\frac{30}{5}, \frac{30}{5}\right) = (6,6)$$

centroid of cluster 2 is (6,6)

cluster 3: $A_2(2,5)$ $C_1(1,2)$

centroid of cluster 3

Sum of x-coordinates

$(2+1) = 3$

Sum of y-coordinates

$(5+2) = 7$

centroid of cluster: $\left(\frac{3}{2}, \frac{7}{2}\right) = (1.5, 3.5)$

New centroids

centroid of cluster 1 = (2, 10)

2 = (6, 6)

3 = (1.5, 3.5)

Subject: ML
Faculty: S. Sandhya Rani
Topic: K-means

Class Notes

Unit No:
Lecture No:
Link to Session
Planner (SP): S.No of SP
Book Reference:
Date Conducted:
Page No: 9

Current centroid $A1 = 2,10$ $C_1 = 1.5, 3.5$
$B_1 = 6,6$

Data points			Distance to						cluster	new cluster
			2	10	6	6	1.5	3.5		
A_1	2	10	0.00		5.66		6.52		1	1
A_2	2	5	5.00		4.12		1.58		3	3
A_3	8	4	8.49		2.83		6.52		2	2
B_1	5	8	3.61		2.24		5.70		2	2
B_2	7	5	7.07		1.41		5.70		2	2
B_3	6	4	7.21		2.00		4.53		2	2
C_1	1	2	8.06		6.40		1.58		3	3
C_2	4	9	2.24		3.61		6.04		2	1

new centroid $A_1 - A1(3, 9.5)$

$B_1 - (6.5, 5.25)$

$C_1 - (1.5, 3.5)$

Current centroid:

A_1: (3, 9.5)

B_1: (6.5, 5.25)

C_1: (1.5, 3.5)

Datapoints			Distance to						Cluster	New cluster
			3	9.5	6.5	5.25	1.5	3.5		
A_1	2	10	1.12		6.54		6.52		1	1
A_2	2	5	4.61		4.51		1.58		3	3
A_3	8	4	7.43		1.95		6.52		2	2
B_1	5	8	2.50		3.13		5.70		2	1
B_2	7	5	6.02		0.56		5.70		2	2
B_3	6	4	6.26		1.35		4.53		2	2
C_1	1	2	7.76		6.39		1.58		3	3
C_2	4	9	1.12		4.51		6.04		1	1

New centroid:

A_1 (3.67, 9)

B_1 (7, 4.33)

C_1 (1.5, 3.5)

Subject: ML
Faculty: S. Sandhya Rani
Topic: k-means, k-mode clustering

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No. of SP
Book Reference:
Date Conducted:
Page No: 10

Current centroid:

$A_1 : (3.67, 9)$

$B_1 : (7, 4.33)$

$C_1 : (1.5, 3.5)$

Datapoints			Distance to 3.67, 9	Distance to 7, 4.33	Distance to 1.5, 3.5	cluster	new cluster
A1	2	10	1.94	7.56	6.52	1	1
A2	2	5	4.33	5.04	1.58	3	3
A3	8	4	6.62	1.05	6.52	2	2
B1	5	8	1.67	4.18	5.70	1	1
B2	7	5	5.21	0.67	5.70	2	2
B3	6	4	5.52	1.05	4.53	2	2
C1	1	2	7.49	6.44	1.58	3	3
C2	4	9	0.33	5.55	6.04	1	1

$$d(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

K-mode Clustering

The difference between K-mode and K-means methods

K-means method is applied to numerical data

↓

even, odd, scores of students
company profits in each quarter,
No. of days in year...

K-modes method is usually applied to categorical data.

↓

Names.

eg: Gender, Name, education level...

K-mode clustering is an unsupervised ML technique used to group a set of data objects into a specified no. of clusters. based on their categorical attributes. The algorithm is called "K-mode" because it uses modes (i.e the most frequent values) instead of means (or) medians to represent the clusters.

How does the KModes Algorithm work

1. Pick K observations at random and use them as clusters.
2. calculate the dissimilarities and assign each observation to it's closest cluster

Subject: ML
Faculty: S. Sandhya Rani
Topic: K-mode clustering

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S No of SP
Book Reference:
Date Conducted:
Page No: 11

Step 3: Define new modes for the clusters

Step 4: Repeat 2-3 steps until there are is no re-assignment required.

clusters:

Individual	Q_1	Q_2	Q_3	Q_4	Q_5	C_1	C_2	C_3
1	A	B	A	B	C	0	4	2
2	A	A	A	B	B	2	4	4
3	C	A	B	B	A	4	2	4
4	A	B	B	A	C	2	5	0
5	C	C	C	B	A	4	0	5
6	A	A	A	A	B	3	5	4
7	A	C	A	C	C	2	4	3
8	C	A	B	B	C	3	3	3
9	A	A	B	C	A	4	4	3
10	A	B	B	A	C	2	5	0

clusters	Q_1	Q_2	Q_3	Q_4	Q_5
1 (1)	A	B	A	B	C
2 (5)	C	C	C	B	A
3 (10)	A	B	B	A	C

Assign individuals to clusters

Cluster 1: (1), (2), (6), (7), (8)

Cluster 2: (3), (5)

Cluster 3: (4), (9), (10)

⇒ Cluster 1:

Q_1	Q_2	Q_3	Q_4	Q_5
A	B	A	B	C
A	A	A	B	B
A	A	A	A	B
A	C	A	C	C
C	A	B	B	C
Repeated → A	A	A	B	C

we took cluster-1 from those data points maximum value is assigned as a cluste

So, cluster1 : A A A B C

⇒ cluster 2

Q_1	Q_2	Q_3	Q_4	Q_5	
C	A	B	B	A	
C	C	C	B	A	
C	A	B	B	A	⇐ cluster 2

⇒ cluster 3:

Q_1	Q_2	Q_3	Q_4	Q_5	
A	B	B	A	C	
A	A	B	C	A	
A	B	B	A	C	
A	B	B	A	C	⇐ cluster 3

Subject: ML
Faculty: S. Sandhya Rani
Topic: E

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S No of SP
Book Reference:
Date Conducted:
Page No: 12

Cluster1: A A A B C

Cluster2: C A B B A

Cluster3: A B B A C

Individual	Q_1	Q_2	Q_3	Q_4	Q_5	C_1	C_2	C_3
1	A	B	A	B	C	1	4	2
2	A	A	A	B	B	1	3	4
3	C	A	B	B	A	3	0	4
4	A	B	B	A	C	3	4	0
5	C	C	C	B	A	4	2	5
6	A	A	A	A	B	2	4	3
7	A	C	A	C	C	2	5	3
8	C	A	B	B	C	2	1	3
9	A	A	B	C	A	3	2	3
10	A	B	B	A	C	3	4	0

Assign individuals to clusters

Cluster1 : (1), (2), (6), (7), (8)

Cluster2 : (3), (5), (8), (9)

Cluster 3 : (4), (10)

Expectation - Maximization

In the real world applications of ML, it is very common that there are many relevant features available for learning but only a small subset of them are observable.

The Expectation - Maximization algorithm can be used for the latent variables (variables that are not directly observable and are actually inferred from the values of the other observed variables)

This algorithm is actually the base for many unsupervised clustering algorithms in the field of ML.

Let us understand the EM algorithm in detail.

- 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.

- The next step is known as "Expectation" - step (or) E-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.

Subject: ML
Faculty: S. Sandhya Rani
Topic: Expectation - Maximization

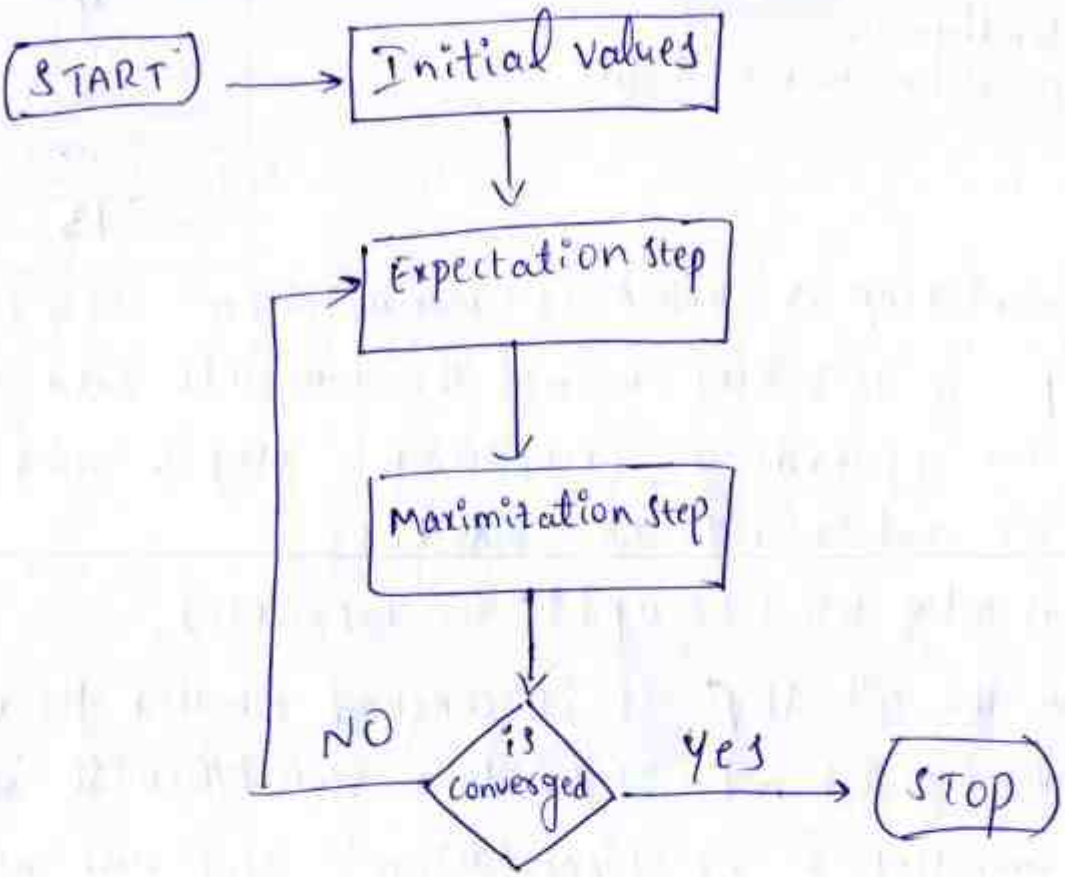
Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S No of SP
Book Reference:
Date Conducted:
Page No: 13

- The Next step is known as "maximization" - step (or) M-step. in this step, we use the complete data generated in the preceding "Exceptation" - step in order to update the values of the parameters.
it is basically used to update the hypothesis.
- Now, in the 4th 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" - stop untill the convergence occurs.

Algorithm:

1. Given a set of incomplete data, consider a set of starting parameter.
2. Expectation step (E-step): using the observed available data of the dataset, estimate (guess) the values of the missing data.
3. Maximization step (M-step): complete data generated after the expectation (E) step is used in order to update the parameters.
4. Repeat step 2 and step 3 until convergence.



Usage of EM algorithm

- it can be used to fill 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 Hidden Markov mode (HMM)
- it can be used for discovering the values of latent variables.

Subject: ML
Faculty: S. Sandhya Rani
Topic: EM Algorithm

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S No ... of SP
Book Reference:
Date Conducted:
Page No: 14

Advantages:

- it is always guaranteed that likelihood will increase with each iteration.
- The E-step and M-step are often pretty easy for many problems in terms of implementation.
- solutions to the M-steps often exist in the closed form.

dis-advantages

- it has slow convergence.
- it makes convergence to the local optima only.
- it requires both the probabilities, forward and backward

Example:

- Assume that we have two coins, C1 and C2
- Assume the bias of C_1 is θ_1 (i.e probability of getting heads with C1)
- Assume the bias of C_2 is θ_2
- we want to find θ_1, θ_2 by performing a no. of trials (i.e, coin tosses)

- we choose 5 times one of the coins
- we toss the chosen coin 10 times.

(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

$$\theta_1 = \frac{\text{no. of heads using } C_1}{\text{Total no. of flips using } C_1}$$

$$\theta_2 = \frac{\text{no. of heads using } C_2}{\text{Total no. of flips using } C_2}$$

Subject: ML
Faculty: S. Sandhya Rani
Topic: Example of EM

Class Notes

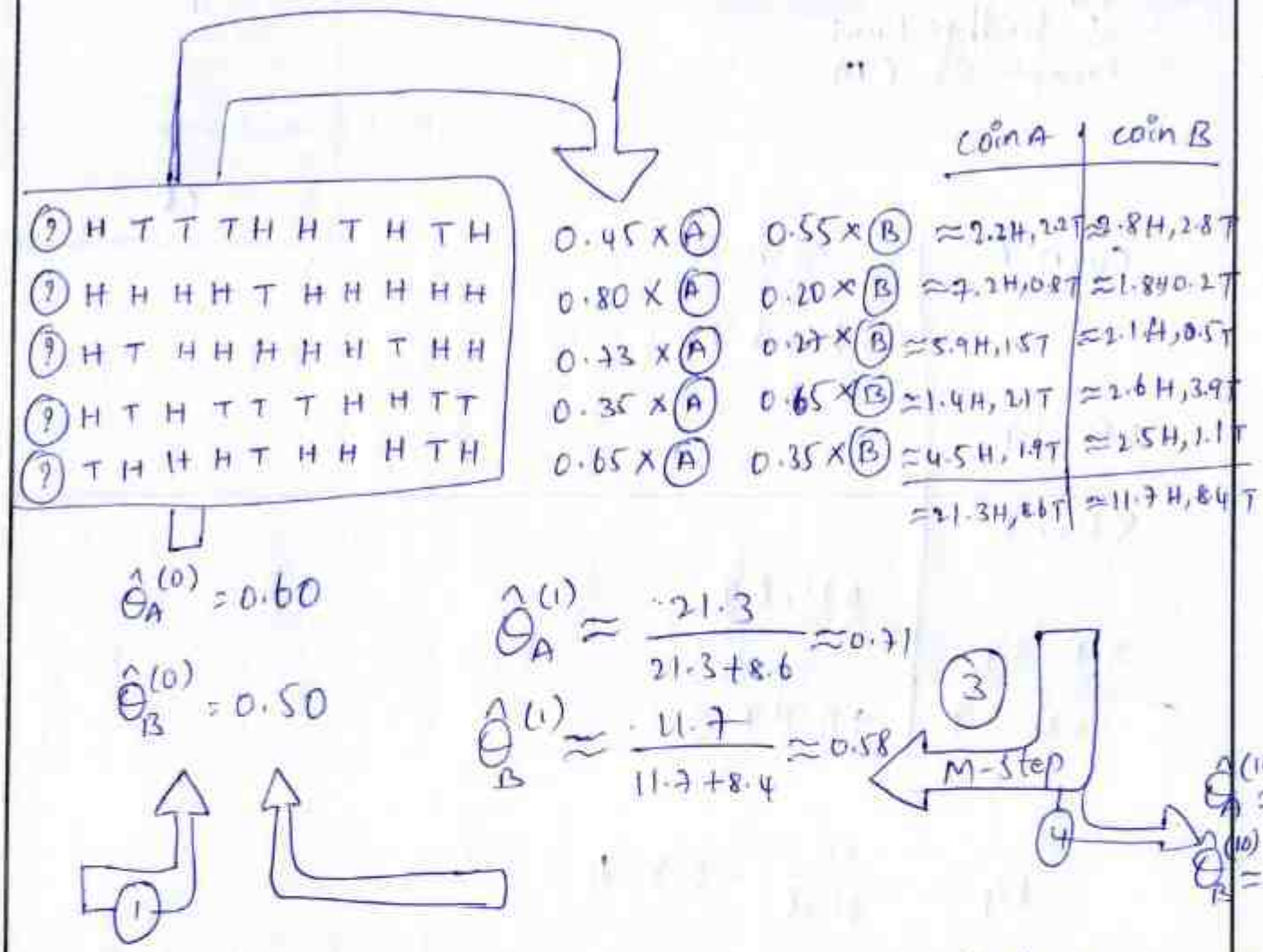
Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No of SP
Book Reference:
Date Conducted:
Page No: 15

Coin A	Coin B
	5H, 5T
9H, 1T	
8H, 2T	
	4H, 6T
7H, 3T	
24H, 6T	9H, 11T

$$\theta_1 = \frac{24}{24+6} = 0.8 \text{ H}$$

$$\theta_2 = \frac{9}{9+11} = 0.45 \text{ H}$$

- Assume a more challenging problem
  - we do not know the identities of the coins used for each set of tosses (we treat them as hidden variables)



$$P(E|Z_A) = P(HHHHHHHHHT \mid A \text{ chosen}) = \binom{n}{x} \theta_A^x (1-\theta_A)^n$$

$$P(E|Z_B) = P(HHHHHHHHHT \mid B \text{ chosen}) = \binom{n}{x} \theta_B^x (1-\theta_B)^n$$

$$P(E|Z_A) = \binom{9}{1} * (0.6)^9 * (0.4)^1 = 0.036$$

$$P(E|Z_B) = \binom{9}{1} * (0.5)^9 * (0.5)^1 = 0.009$$

$$P(Z_A|E) = \frac{0.036}{0.036+0.009} = 0.80$$

$$P(Z_B|E) = \frac{0.009}{0.036+0.009} = 0.20$$

Subject: ML
Faculty: S. Sandhya Rani
Topic: Self-Organizing Map (SOM)

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No of SP
Book Reference:
Date Conducted:
Page No: 16

Self-organizing map

SOM is an a unsupervised ML technique used to produce a low-dimensional representation of a higher-dimensional data set while preserving the topological structure of the data.

For example, a data set with p variables measured in n observations could be represented as clusters of observations with similar values for the variables.

These clusters then could be visualized as a two-dimensional "map" such that observations in proximal clusters have more similar values than observations in ~~dig~~disital clusters.

This can make high-dimensional data easier to visualize and analyze.

Five stages in self-organizing map

i) Initialization ii) Sampling iii) matching

iv) updating v) continuation.

Initialization:

- Set map size and shape
- Initialize neurons with random weights.
- Define learning rate and radius.

Sampling:

- Select an input data point from the dataset
- present it to the N/w.

matching:

- calculate distance b/w i/p data and neuron weights
- Determine the Best Matching unit (BMU)

Updating:

- update weights of BMU and its neighbors
- Adjust learning rate and radius

continuation:

- Repeat step2-4 until convergence (or) stopping criteria
- Refine map topology and weights.

These are also known as kohonen maps (or) SOM's are a type of Artificial neural network used for unsupervised learning and data visualization.

input data into lower to dimensional grid.

Subject: ML
Faculty: S. Sandhya Rani
Topic: Self-organizing feature maps

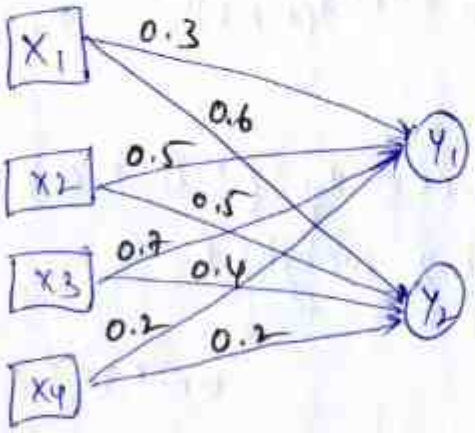
Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No. of SP
Book Reference:
Date Conducted:
Page No: 17

- consider the N/w shown in Figure which consider four training samples each vector of length 4 and two output units.
- Train the SOFM Network by determining the class memberships of the input data.
- Training samples:

X_1: (1, 0, 1, 0) X_2: (1, 0, 0, 0)

X_3: (1, 1, 1, 1) X_4: (0, 1, 1, 0)



- output units: unit1, unit2
- Learning rate $\eta(t) = 0.6$ → control how quickly the model [The weight amount updated during the training)
- Initial weight matrix

$$\begin{bmatrix} \text{unit 1} \\ \text{unit 2} \end{bmatrix} = \begin{bmatrix} 0.3 & 0.5 & 0.7 & 0.2 \\ 0.6 & 0.5 & 0.4 & 0.3 \end{bmatrix}$$

Iteration 1:

Training sample $x_1 = (1, 0, 1, 0)$

weight matrix:

$$\begin{bmatrix} \text{unit 1} \\ \text{unit 2} \end{bmatrix} : \begin{bmatrix} 0.3 & 0.5 & 0.7 & 0.2 \\ 0.6 & 0.7 & 0.4 & 0.3 \end{bmatrix}$$

compute Euclidean distance between $x_1 : (1, 0, 1, 0)$ and unit 1 weights.

$$d(P_1 P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

x_2

unit 1 = 0.3 0.5 0.7 0.2

$x_1 : (1, 0, 1, 0)$

x_1

$$d^2 = (0.3-1)^2 + (0.5-0)^2 + (0.7-1)^2 + (0.2-0)^2$$

$$= 0.87$$

compute Euclidean distance between $x_1 : (1, 0, 1, 0)$ and unit 2 weights

x_2

unit 2 = 0.6 0.7 0.4 0.3

$x_1 : 1, 0, 1, 0$

$$d^2 = (0.6-1)^2 + (0.7-0)^2 + (0.4-1)^2 + (0.3-0)^2$$

$$= 1.1$$

unit 1 wins Because smaller than unit 2 of sample 1

Subject: ML
Faculty: S. Sandhya Rani
Topic: Self organization map

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S No. of SP
Book Reference:
Date Conducted:
Page No: 18

update weights:

$$w_j(t+1) = w_j(t) + \eta(t)\,(x_s - w_j(t))$$

↓ new weight ↓ old weight ↓ learning rate ↓ i/p ↓ old weights

update the weights of the winning unit

New unit 1 weights $= [0.3\ 0.5\ 0.7\ 0.2] + 0.6\,([1\ 0\ 1\ 0] - [0.3\ 0.5\ 0.7\ 0.2])$

$= [0.3\ 0.5\ 0.7\ 0.2] + 0.6\,[0.7\ -0.5\ 0.3\ -0.2]$

$= [0.3\ 0.5\ 0.7\ 0.2] + [0.42\ -0.30\ 0.18\ -0.12]$

$= [0.72\ 0.2\ 0.88\ 0.08]$

$$\begin{bmatrix}\text{unit 1}\\ \text{unit 2}\end{bmatrix} : \begin{bmatrix}0.72 & 0.2 & 0.88 & 0.08\\ 0.6 & 0.7 & 0.4 & 0.3\end{bmatrix}$$

Iteration 2:

Training Sample $x_2 : (1, 0, 0, 0)$

weight matrix

$$\begin{bmatrix}\text{unit 1}\\ \text{unit 2}\end{bmatrix} : \begin{bmatrix}0.72 & 0.2 & 0.88 & 0.08\\ 0.6 & 0.7 & 0.4 & 0.3\end{bmatrix}$$

Comput Euclidean distance between x_1: (1, 0, 0, 0) and unit 1 weights.

$$d^2 = (0.7-1)^2 + (0.2-0)^2 + (0.88-0)^2 + (0.08-0)^2$$

$$= 0.74$$

Comput Euclidean distance between x_1: (1, 0, 0, 0) and unit 2 weights.

$$d^2 = (0.6-1)^2 + (0.7-0)^2 + (0.4-0)^2 + (0.3-0)^2$$

$$= 0.9$$

unit 1 wins

update weights:

$$w_j(t+1) = w_j(t) + n(t)(x_s - w_j(t))$$

update the weights of the winning unit:

New unit 1 weights $= [0.72\ 0.2\ 0.88\ 0.08] + 0.6([1\ 0\ 0\ 0] - [0.72\ 0.2\ 0.88\ 0.08])$

$= [0.72\ 0.2\ 0.88\ 0.08] + 0.6[0.28\ -0.2\ -0.88\ -0.08]$

$= [0.72\ 0.2\ 0.88\ 0.08] + [0.17\ -0.12\ -0.53\ -0.05]$

$= [0.8\ 0.08\ 0.35\ 0.03]$

$$\begin{bmatrix} \text{unit 1} \\ \text{unit 2} \end{bmatrix} = \begin{bmatrix} 0.89 & 0.08 & 0.35 & 0.03 \\ 0.6 & 0.7 & 0.4 & 03 \end{bmatrix}$$

Subject: ML
Faculty: S Sandhya Rani
Topic: SOM

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No.... of SP
Book Reference:
Date Conducted:
Page No: 19

Iteration 3:

Training sample x_3: (1, 1, 1, 1)

weight matrix:

$$\begin{bmatrix} \text{unit 1} \\ \text{unit 2} \end{bmatrix} : \begin{bmatrix} 0.89 & 0.08 & 0.35 & 0.03 \\ 0.6 & 0.7 & 0.4 & 0.3 \end{bmatrix}$$

compute Euclidean distance between x_3: (1, 1, 1, 1) and unit 1 weights.

$$d^2 = (0.89-1)^2 + (0.08-1)^2 + (0.35-1)^2 + (0.03-1)^2$$

$$= 2.2$$

compute Euclidean distance between x_3: (1, 1, 1, 1) and unit 2 weights

$$d^2 = (0.6-1)^2 + (0.7-1)^2 + (0.4-1)^2 + (0.3-1)^2$$

$$= 1.1$$

Unit 2 wins

update weights:

$$w_j(t+1) = w_j(t) + \eta(t)\left(x_s - w_j(t)\right)$$

update the weights of the winning unit:

New unit 2 weights = $[0.6\ 0.7\ 0.4\ 0.3] + 0.6([1\ 1\ 1\ 1] - [0.6\ 0.7\ 0.4\ 0.3])$

$= [0.6\ 0.7\ 0.4\ 0.3] + 0.6[0.4\ 0.3\ 0.6\ 0.7]$

$= [0.6\ 0.7\ 0.4\ 0.3] + [0.24\ 0.18\ 0.36\ 0.42]$

$= [0.84\ 0.88\ 0.76\ 0.72]$

$$\begin{bmatrix} \text{unit 1} \\ \text{unit 2} \end{bmatrix} : \begin{bmatrix} 0.89 & 0.08 & 0.35 & 0.03 \\ 0.84 & 0.88 & 0.76 & 0.72 \end{bmatrix}$$

Iteration 4:

Training sample x_4: (0, 1, 1, 0)

weight matrix:

$$\begin{bmatrix} \text{unit 1} \\ \text{unit 2} \end{bmatrix} : \begin{bmatrix} 0.89 & 0.08 & 0.35 & 0.03 \\ 0.84 & 0.88 & 0.76 & 0.72 \end{bmatrix}$$

Compute Euclidean distance b/w x_4: (0, 1, 1, 0) and unit 1 weights.

$d^2 = (0.89-0)^2 + (0.08-1)^2 + (0.35-1)^2 + (0.03-0)^2$

$= 2.06$ ✓

compute Euclidean distance b/w x_1: (0, 1, 1, 0) and unit 2 weights.

$d^2 = (0.84-0)^2 + (0.88-1)^2 + (0.76-1)^2 + (0.72-0)^2 = 1.3$ ✓

Subject: ML
Faculty: S Sandhya Rani
Topic:

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No.... of SP
Book Reference:
Date Conducted:
Page No: 20

update weights:

$$w_j(t+1) = w_j(t) + \eta(t)(x_s - w_j(t))$$

unit 2 wins

update the weights of the winning unit:

New unit 2 weights:

$$= [0.84\ 0.88\ 0.76\ 0.72] + 0.6([0\ 1\ 1\ 0] - [0.8\ 0.88\ 0.76\ 0.72])$$

$$= [0.84\ 0.88\ 0.76\ 0.72] + 0.6[-0.84\ 0.12\ 0.24\ -0.72]$$

$$= [0.84\ 0.88\ 0.76\ 0.72] + [-0.5\ 0.07\ 0.14\ -0.43]$$

$$= [0.34\ 0.95\ 0.9\ 0.29]$$

$$\begin{bmatrix} \text{unit 1} \\ \text{unit 2} \end{bmatrix} : \begin{bmatrix} 0.89 & 0.08 & 0.35 & 0.03 \\ 0.34 & 0.95 & 0.9 & 0.29 \end{bmatrix}$$

Best mapping units each of the sample taken are:

x_1: (1, 0, 1, 0) → unit 1

x_2: (1, 0, 0, 0) → unit 1

x_3: (1, 1, 1, 1) → unit 2

x_4: (0, 1, 1, 0) → unit 2

This process is continued for many epochs unit the feature map does not change.

SOFM is not typically used to map low-dimensional data to high-dimensional data.

However, if we consider the opposite process.

Low dimensional data:

Imagine a dataset of 2D points

Mapping to high-dimensional data

we can use a technique like polynomial feature expansion to map these 2D points to a higher-dimensional space, say 5D. (x, y, x^2, xy, y^2)

5D (X, Y, Z)

λ → wave length dimension

T - Time dimension

2D (X, Y)

multi dimensional arrays containing values (or) data represented in a grid (or) table style with rows and columns.

Subject: ML
Faculty: S. Sandhya Rani
Topic: Gaussian Mixture Model

Class Notes

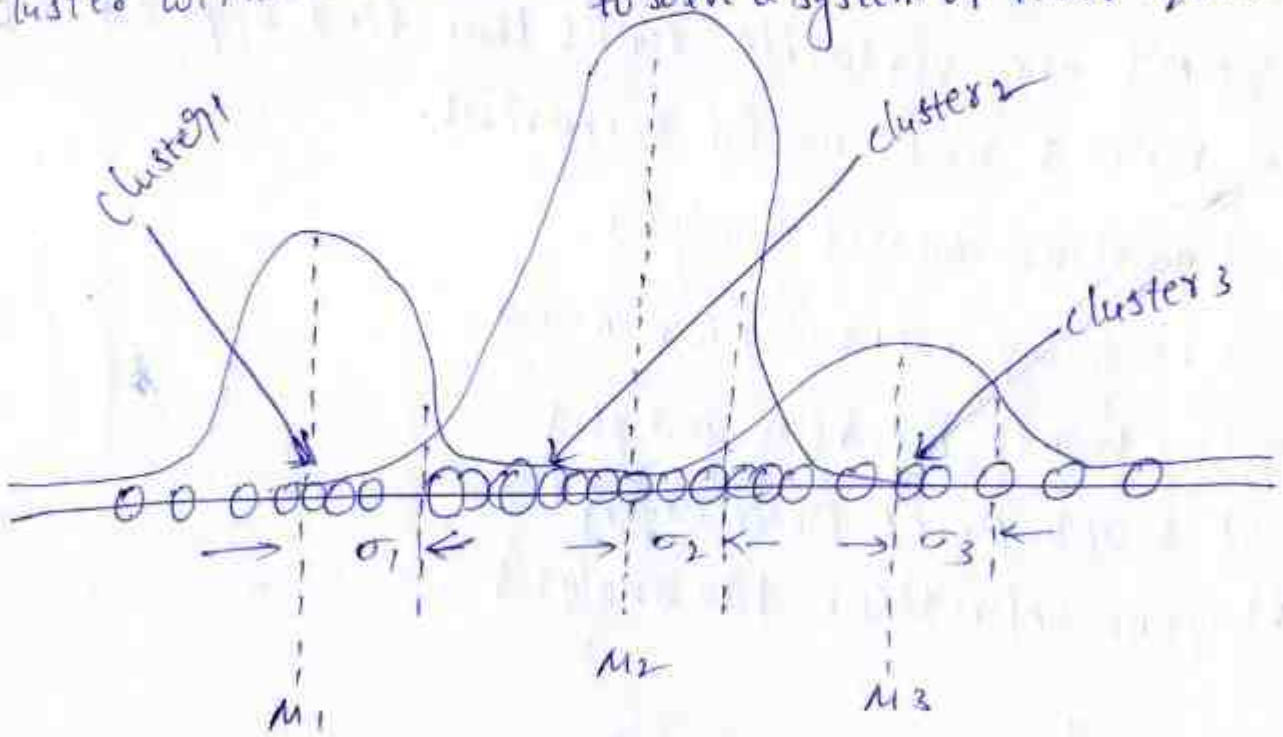
Unit No: 4
Lecture No:
Link to Session
Planner (SP): S No ... of SP
Book Reference:
Date Conducted:
Page No: 21

Gaussian Mixture Model:

A GMM is a probabilistic model representing data as a mixture of multiple Gaussian distributions.

- Each Gaussian distribution represents a component or cluster within the data. # to describe and classify features.

to solve a system of linear equations



The key component of a GMM are:

- No. of components: The GMM assumes that the data is a mixture of a specific no. of Gaussian distributions, also known as components (or) clusters.

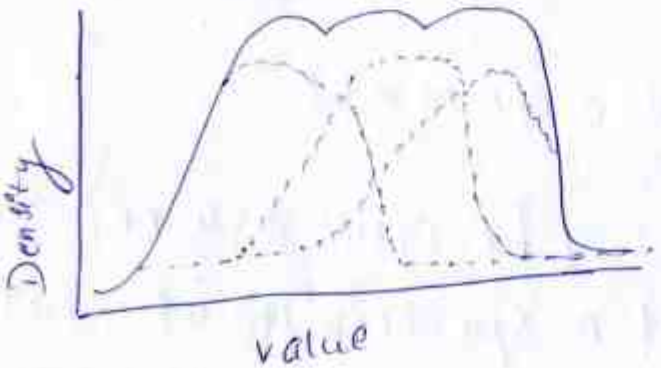
Gaussian Distributions: each component in the GMM is represented by a Gaussian distribution.

Mixture weights: The GMM assigns mixture weights to each component, representing the probability of selecting that component when generating a data.

Real-world examples:

GMM's are versatile tools that find applications in various real-world scenarios.

i) medical dataset analysis
ii) modeling natural phenomena
iii) customer behavior analysis
iv) stock price prediction
v) Gene expression data analysis



Subject: ML
Faculty: S. Sandhya Rani
Topic:

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No. of SP
Book Reference:
Date Conducted:
Page No: 22

Advantages:

- Flexibility
- Robustness
- speed
- To Handle missing data
- Interpretability.

Dis - Advantages

- Sensitivity to initialization
- Assumption of Normality
- No. of components
- High-dimensional data
- Limited expressive power

Principal component Analysis (PCA)

It is an unsupervised learning algorithm that is used for the dimensionality reduction in ML.
it is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation.
These new transformed features are called the PC.
it is one of the popular tools that is used for exploratory data analysis and predictive modeling.
it is a technique to draw strong patterns from the given dataset by reducing the variances.

PCA generally tries to find ~~the~~ lower-dimensional surface to project ~~the~~ high-dimensional data.

Real world Applications:

i) Image processing
ii) Movie Recommendation system
iii) optimizing the power allocation in various communication channels.

Subject: ML
Faculty: S. Sandhya Rani
Topic: PCA

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No. of SP
Book Reference:
Date Conducted:
Page No: 23

The PCA algorithm is based on some mathematical concepts such as:

- variance and covariance
- Eigenvalues and Eigen factors

Some common terms used in PCA Algorithm:

- Dimensionality
- correlation
- orthogonal
- Eigenvectors
- covariance matrix

Applications

1. computer vission
2. Image compression.

Steps for PCA algorithm

1. Getting the dataset
2. Representing data into a structure
3. Standardizing the data
4. calculating the covariance of Z
5. calculating the Eigen values and Eigen vectors
6. Sorting the eigen vectors
7. calculating the new features (or) principal components.
8. Remove less (or) unimportant features from the new dataset.

Locally Linear Embedding in ML

LLE is an unsupervised approach designed to transform data from its original high-dimensional space into a lower-dimensional representation.

LLE operates in several key steps:

- Firstly, it constructs a nearest neighbors graph to capture these local relationships. Then, it optimizes weight values for each data point, aiming to minimize the reconstruction error when expressing a point as a linear combination of it's neighbors. This weight matrix reflects the strength of connections b/w points.
- NEXT, LLE computes a low dimensional representation of the data by finding eigenvectors of a matrix derived from the weight matrix.

Square matrix is a non-zero vector

These eigenvectors represent the most relevant directions in the reduced space.

users can specify the desired dimensionality for the output space.

Subject: ML
Faculty: S Sandhya Rani
Topic: Locally Linear Embedding in ML

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No.... of SP
Book Reference:
Date Conducted:
Page No: 24

Mathematical Implementation of LLE algo:

The key idea of LLE is that locally, in the vicinity of each data point, the data lies approximately on a linear subspace. LLE attempts to unfold (or) unroll the data while preserving these local linear relationships.

Here is a mathematical overview of the LLE algorithm,

minimize : $\sum_i | x_i - \sum_j w_{ij} x_j |$

Subject to : $\sum_j w_{ij} = 1$

where x_i - represents the i^{th} data point

w_{ij} are the weights that minimize the reconstruction error for data point x_i using it's neighbors.

It aims to find a lower-dimensional representation of data while preserving local relationships.

The mathematical expression for LLE involves minimizing the reconstruction error of each data point by expressing it as a weighted sum of it's k-nearest neighbors contributions.

LLE Algorithm:

- Neighboorhood selection:

for each data point in the high-dimensional space, LLE identifies it's k-nearest neighbors. This step is crucial because LLE assumes that each data point can be well approximated by a linear combination of it's neighbors.

- weight matrix construction:

LLE computes a set of weights for each data point to express it as a linear combination of it's neighbors. These weights are determined in such a way that the reconstruction error is minimized. Linear regression is often used to find these weights.

- Global structure preservation:

After constructing the weight matrix, LLE aims to find a lower-dimensional representation of the data that best preserves the local linear relationship.

- output Embedding:

once the optimization process is complete, LLE provides the final lower-dimensional representation of the data. This representation caputre the essential structure of the data while reducing its dimensionality.

Subject: ML
Faculty: S Sandhya Rani
Topic: Factor Analysis

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No.... of SP
Book Reference:
Date Conducted:
Page No: 25

Factor Analysis:

Factor analysis is a statistical method used to analyze the relationships among a set of observed variables by explaining the correlations (or) covariances b/w them in terms of a smaller number of unobserved variables called factors.

Types of FA:

1. Exploratory FA:

It is a powerfull tool to uncover latent factors within a dataset.

EFA does not impose any predetermined structure (or) assume pre-existing relationships among variables.

2. confirmatory FA:

CFA is a technique used to evaluate predetermined hypotheses regarding the relationships b/w variables and Factors.

Principles of FA:

1. Latent variables:

a latent variable is a random variable which you can't observe neither in training nor in test phase.

2. Variance Explanation:

ML refers to the changes in the model when using different portions of the training data set.

3. Factor Loadings:

The correlation coefficient for the variable and factor.

Subject: ML
Faculty: S Sandhya Rani
Topic: Neural Networks

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No.... of SP
Book Reference:
Date Conducted:
Page No: 26

Neural Networks:

Neural Networks are computational model that mimic the complex functions of the human brain.

The neural network consist of interconnected nodes (or) neurons that process and learn from data, enabling tasks such as pattern recognition and decision making in ML.

Evolution of NN:

Since the 1940's there have been a no. of noteworthy advancements in the field of neural networks.

1940s - 1950s Early concept by Mcculloch and pitts

1960s - 1970s by the work of Rosenblatt on perceptrons

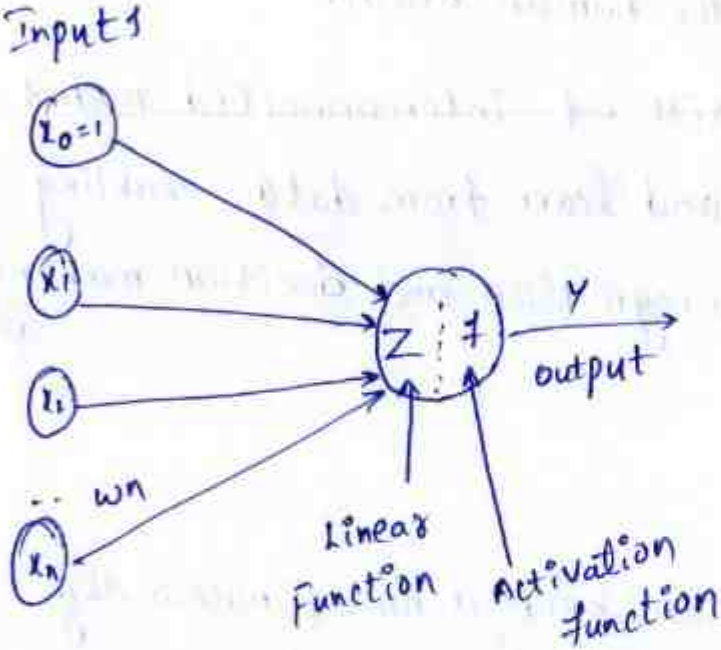
1980s - Backpropagation and connectionism

1990s - Boom and winter

2000s - Resurgence and deep learning

2010s - present - Deep learning Domainance.

- The neural network is simulated by a new environment.
- Then the free parameters of the neural network are changed as a result of this simulation.
- The neural network then responds in a new way to the environment because of the changes in it's free parameters.



What are neural networks used for?

Neural networks have several use cases across many industries, such as the following:

* Medical diagnosis by medical image classification
* Electrical load and energy demand forecasting
* Process and quality control
* Financial predictions by processing historical data of financial instruments.

Subject: ML
Faculty: S Sandhya Rani
Topic: Neural Networks

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No.... of SP
Book Reference:
Date Conducted:
Page No: 27

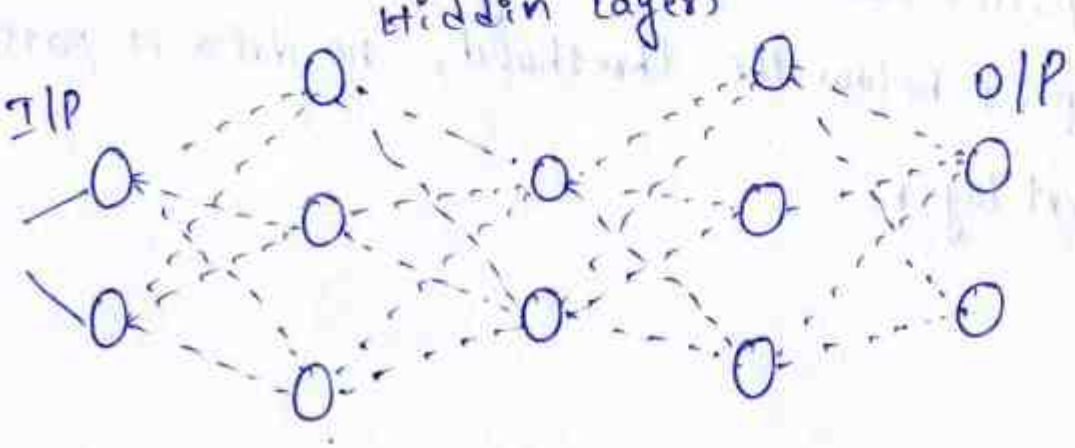
Applications:

- Computer vision
- speech recognition
- Natural language processing
- Recommendation engines

Types of neural Networks

1. feedforward neural Networks
2. Backpropagation algorithm
3. convolutional neural networks.

Neural Network



How do neural networks work?

neural networks are composed of a collection of nodes. The nodes are spread out across at least three layers.

An "i/p" layer

A "hidden" layer

An "o/p" layer

These 3 layers are the minimum. Neural networks can have more than one hidden layer, in addition to the input layer and output layer.

- each node performs some sort of processing task (or) function on whatever i/p it receives from the previous node.
- each node variable within the formula weighted differently.
- if the o/p of applying that mathematical formula to the i/p exceeds a certain threshold,
- The node passes data to the next layer in the neural network. if the o/p is below the threshold, no data is passed to the next layer.

Subject: ML
Faculty: S Sandhya Rani
Topic: perceptron

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No.... of SP
Book Reference:
Date Conducted:
Page No: 28

Types of neural networks?

1. shallow neural networks:
   usually have only one hidden layer
2. Deep neural networks:
   have multiple hidden layers.

shallow NN:

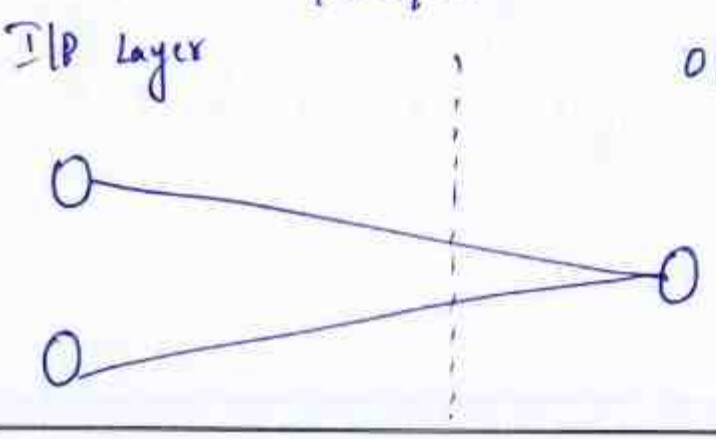
This N/w fast and require less processing power than deep neural networks.

But they cannot perform as many complex tasks as deep neural networks.

perceptron:

Perceptron neural networks are simple, shallow networks with an i/p layer and an o/p layer

Perceptron



perceptron is a building block of an Artificial neural Network.

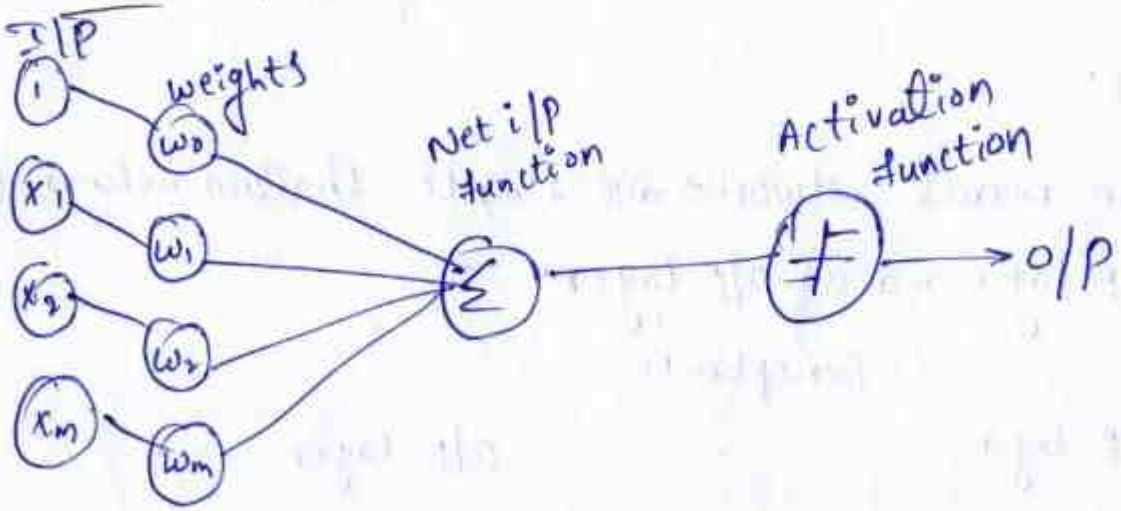
Perceptron is also understood as an ANN (or) NN (neural N/w) unit that helps to detect certain i/p data computations in business intelligence.

Perceptron model is also treated as one of the best and simplest types of ANN.

However, it is a supervised learning algorithm of binary classifiers. Hence, we can consider it as a single-layer neural network with four main parameters.

i.e., i/p values, weights and Bias, net sum, and an activation function.

Basic components of perceptron



Subject: ML
Faculty: S Sandhya Rani
Topic:

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No.... of SP
Book Reference:
Date Conducted:
Page No: 29

Input Nodes (or) Input layer:

This is the primary component of perceptron which accepts initial data into the system for further processing.
each i/p node contains real numerical value.

Weight and Bias:

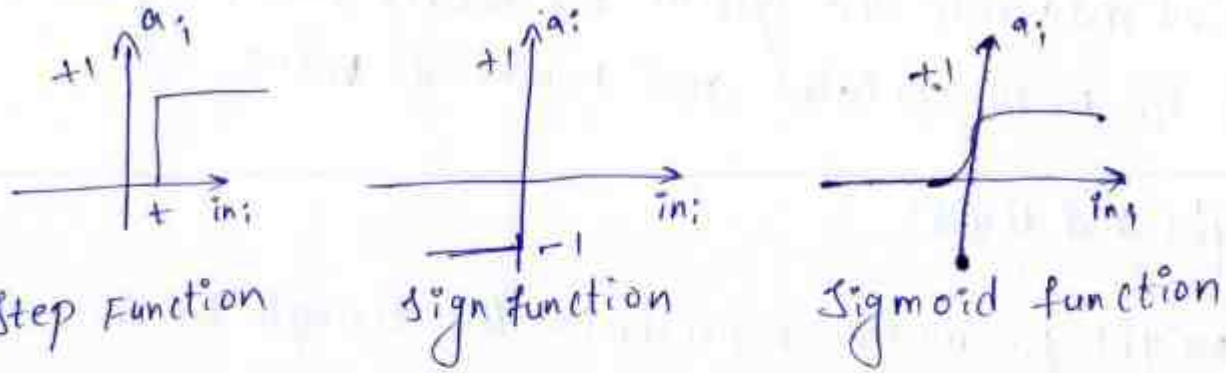
weight parameter represents the strength of the connection b/w units.
This is another most imp parameter of perceptron components.
weight is directly proportional to the strength of the associated i/p neuron in deciding the o/p.
Further, Bias can be considered as the line of intercept in a linear equation.

Activation Function:

These are the final and imp components that help to determine whether the neuron will fire (or) not.
Activation function can be considered primarily as a step function.

Types of Activation function

1. sign function
2. step function
3. sigmoid function



How does perceptron work?

In ML, perceptron is considered as a single-layer neural network that consists of four main parameters named i/p values, weight and Bias, net sum, and an activation function.

The perceptron model begin with the multiplication of all i/p values and their weights, then adds these values together to create the weighted sum.

Then this weighted sum is applied to the activation function 'f' to obtain the desired o/p.

This activation function is also known as the step function and is represented by 'f'

Subject: ML
Faculty: S Sandhya Rani
Topic: Multi layer perceptron

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No.... of SP
Book Reference:
Date Conducted:
Page No: 30

Types of perceptron models:

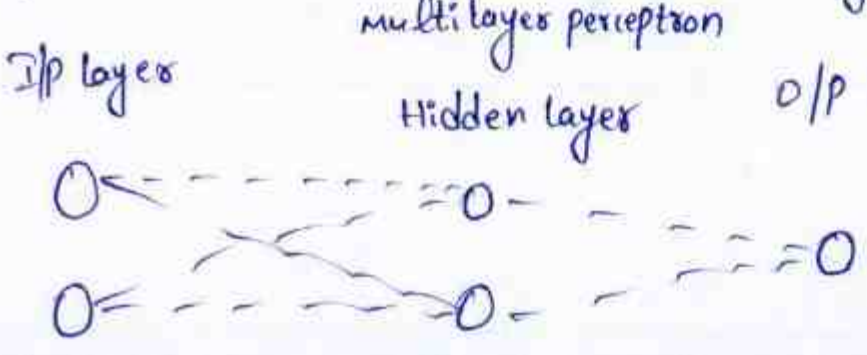
1. Single - Layer perceptron model
2. Multi - Layer perceptron model

Single-layer perceptron model:

This is one of the easiest ANN types. A single-layered perceptron model consists feed-forward network and also include a threshold transfer function inside the model.
The main objetive of the single-layer perceptron model is to analyze the linearly separable objects with binary outcomes.
Single-layer perceptron can learn only linearly separable patterns.

Multi-layer perception:

Multilayer perceptron neural networks add complexity to perceptron n/w, an include a hidden layer.



Like a single-layer perceptron model, a multi-layer perceptron model also has the same model structure but has a greater no. of hidden layers.

The multi-layer perceptron model is also known as the Backpropagation algorithm, which executes in two stages as follows:

Forward stage: Activation functions start from the i/p layer, in the forward stage and terminate on the o/p layer.

Backward stage: In the Backward stage, weight and bias values are modified as per the model's requirement.
In this stage, the error b/w actual o/p and demanded originated backward on the o/p layer and ended on the i/p layer.

A multi-layer perceptron model has greater processing power and can process linear and non-linear patterns.
Further, it can also implement logic gates such as AND, OR, NAND, NOT, XNOR, NOR.

Subject: ML
Faculty: S Sandhya Rani
Topic:

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No.... of SP
Book Reference:
Date Conducted:
Page No: 30+1 : 31

Advantages:

A multi-layered perceptron model can be used to solve complex non-linear problems.

it works well with both small and large i/p data.

it helps us to obtain quick predictions after the training.

it helps to obtain the same accuracy ratio with large as well as small data.

Dis-Advantages:

* in MLP, computations are difficult and time-consuming
* In MLP, it is difficult to predict how much the dependent variable affects each independent variable.
* The model functioning depends on the quality of the quality of the training.

Perceptron function

Perceptron function "f(x)" can be achieved as o/p by multiplying the i/p 'x' with the learned weight coefficient 'w'

Mathematically, we can express it as follows:

$f(x) = 1$; if $w \cdot x + b > 0$

otherwise, $f(x) = 0$

'w' - represents real-valued weights vector

'b' - represents the bias

'x' - represents a vector of i/p x values.

Subject: ML
Faculty: S Sandhya Rani
Topic: Support vector machines

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No ... of SP
Book Reference:
Date Conducted:
Page No: 32

SVM:

SVM is one of the most popular supervised learning algorithms. which is used for classification as well as regression problems.

However, primarly, it is used for classification problems in ML.

The goal of SVM algorithm is to create the best line (or) decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future.

The best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane.

These extreme cases are called as support vectors, and hence algorithm is termed as Support vector machine

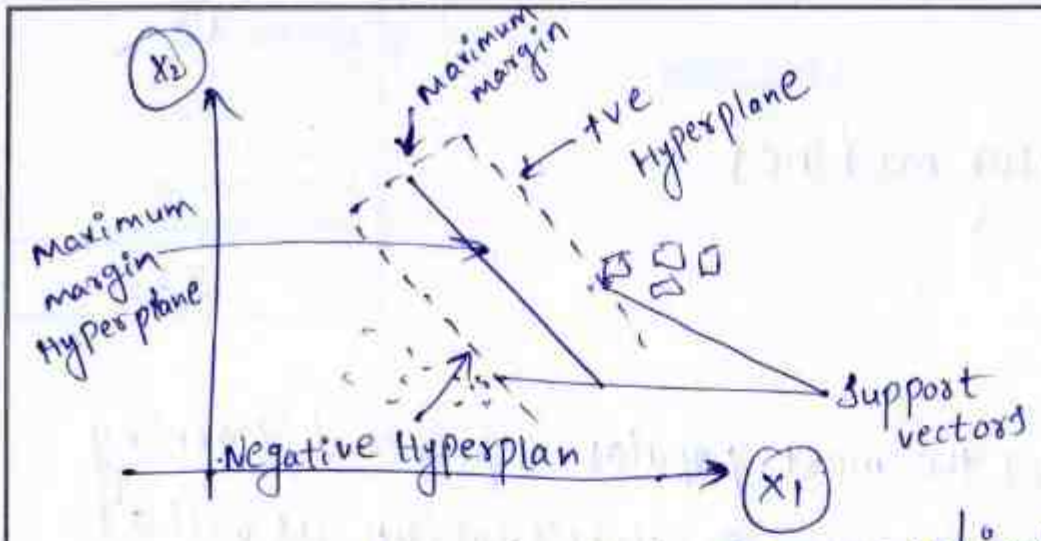


fig: Suppor vector machine.

Subject: ML
Faculty: S Sandhya Rani
Topic: Kernel functions

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No.... of SP
Book Reference:
Date Conducted:
Page No: 33

Kernel function:

A set of techniques known as kernel methods are used in ML to address classification, regression, and other prediction issues.

They are built around the idea of kernels, which are functions that gauge how similar two data points are to one another in a high-dimensional feature space.

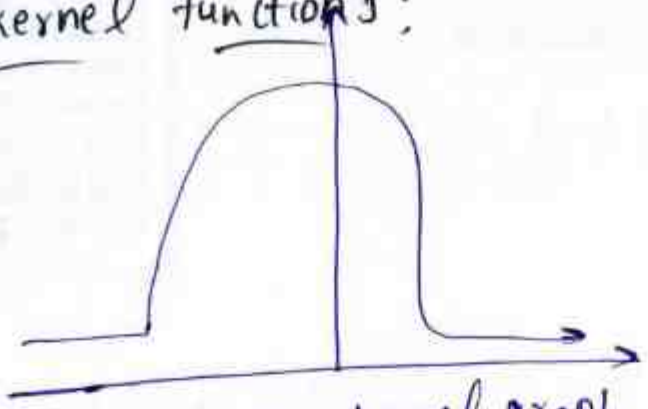
The most popular kind of kernel approach is SVM, a binary classifier that determines the best hyperplane that most effectively divides the two groups.

In order to efficiently locate the ideal hyperplane, SVM map the i/p into a higher-dimensional space using a Kernel function.

Kernel function is a method used to take data as i/p and transform it into the required form of processing data.

"Kernel" is used due to a set of mathematical functions used in SVM providing the window to manipulate the data.

Major kernel functions:



Gaussian kernel graph

Gaussian kernels: it is used to perform transformation when there is no prior knowledge about data.

$$k(x,y) = e^{-\left(\frac{\|x-y\|^2}{2\sigma^2}\right)}$$

Sigmoid kernel:

This function is equivalent to a two-layer, perceptron model of the neural network, which is used as an activation function for artificial neurons.

$$k(x,y) = \tanh(\gamma \cdot x^T y + \gamma)$$

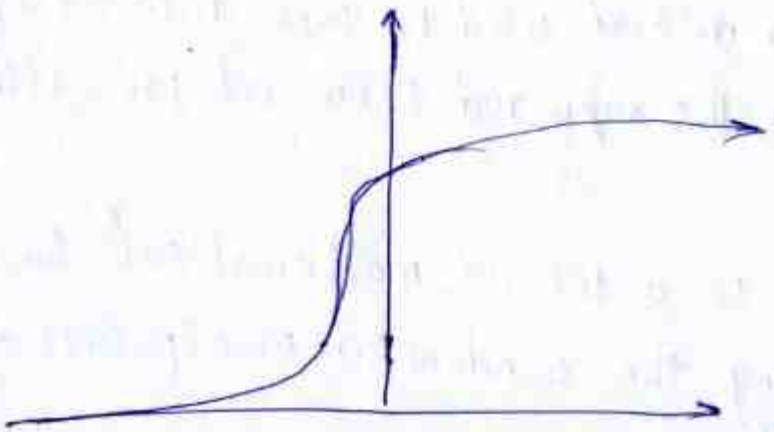


fig: Sigmoid kernel function

Subject: ML
Faculty: S Sandhya Rani
Topic: Kernel function

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No.... of SP
Book Reference:
Date Conducted:
Page No: 34

Polynomial kernel:

It represents the similarity of vectors in the training set of data in a feature space over polynomials of the original variables used in the kernel.

$$K(x, y) = \tanh(\gamma . x^T y + \delta)^d, \gamma > 0$$

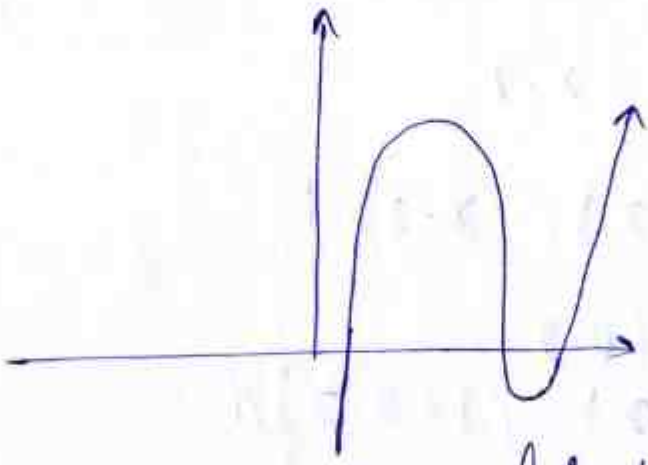


fig: Polynomial kernel function

Popular kernels:

i) Fisher kernel:

it is a function that measures the similarity of two objects on the basis of set of measurments for each oject and a statistical model.

ii) Graph kernel:

It is a kernel function that computes an inner product on graphs.

Graph kernels can be intuitively understood as functions measuring the similarity of pairs of graphs.

They allow kernelized learning algorithm such as SVM to work directly on graphs.

Examples:

Linear : $K(x, z) = x \cdot z$

Polynomial: $K(x, z) = (x \cdot z)^d$

(or)

$K(x, z) = (1 + x \cdot z)^d$

Gaussian: $K(x, z) = e^{-\|x-z\|^2/(2\sigma^2)}$

Subject: ML
Faculty: S Sandhya Rani
Topic: K-Nearest Neighbors

Class Notes

Unit No: 4
Lecture No:
Link to Session Planner (SP): S.No... of SP
Book Reference:
Date Conducted:
Page No: 35

K-Nearest Neighbors:

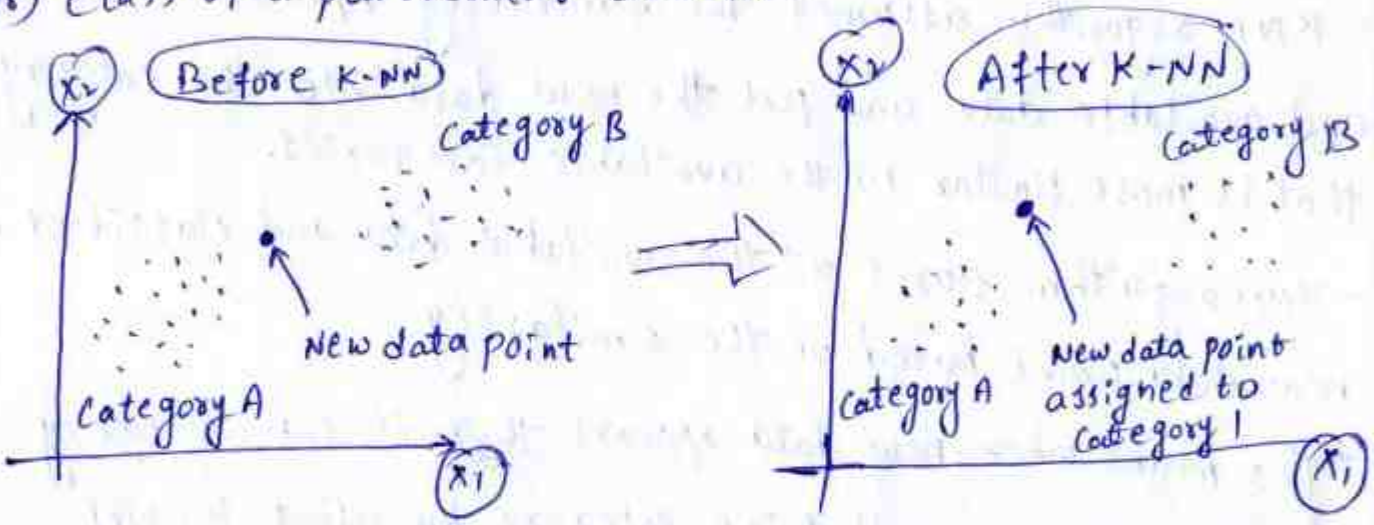
- K-Nearest Neighbor is one of the simplest ML Algorithm based on supervised learning technique.
- KNN algorithm assumes the similarity b/w the new data and available data and put the new data into the category that is most similar to the available categories.
- KNN algorithm stores all the available data and classifies a new data point based on the similarity.
- This means when new data appears then it can be easily classified into a well suite category by using K-NN algorithm.
- K-NN algorithm can be used for Regression as well as for classification but mostly it is used for the classification problems.
- K-NN is a Non-parametric algorithm, which means it does not make any assumption on underlying data.
- it is also called a lazy learner algorithm because it does not learn from the training set* immediately instead it stores the dataset and at the time of classification, it perform an action on the dataset.

* But it memorize the training data

why do we need a KNN Algorithm?

Suppose there are two categories, i.e., category A and category B, and we have a new data point x1, so this data point will lie in which of these categories.

To solve this type of problem, we need a k-NN algorithm. with the help of k-NN, we can easily identify the category (or) class of a particular dataset.



Step 1: Select the number k of the neighbors

Step 2: calculate the Euclidean distance of k no. of neighbors

Step 3: Take the k-nearest neighbors as per the calculated Euclidean distance.

Step 4: Among these k neighbors, count the no. of the data points in each category.

Step 5: Assign the new data points to that category for which the no. of the neighbor is maximum.

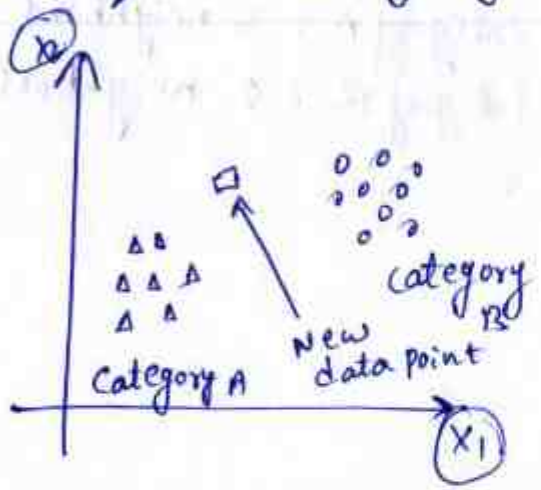
Subject: ML
Faculty: S Sandhya Rani
Topic: KNN

Class Notes

Unit No: 4
Lecture No:
Link to Session Planner (SP): S.No.... of SP
Book Reference:
Date Conducted:
Page No: 36

Step 6: our model is ready

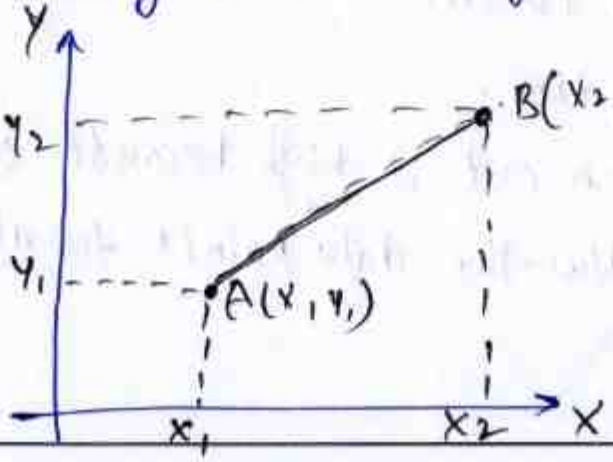
Suppose we have a new data point and we need to put it in the required category.



Firstly we will choose the no. of neighbors, so we will choose the K = 5.

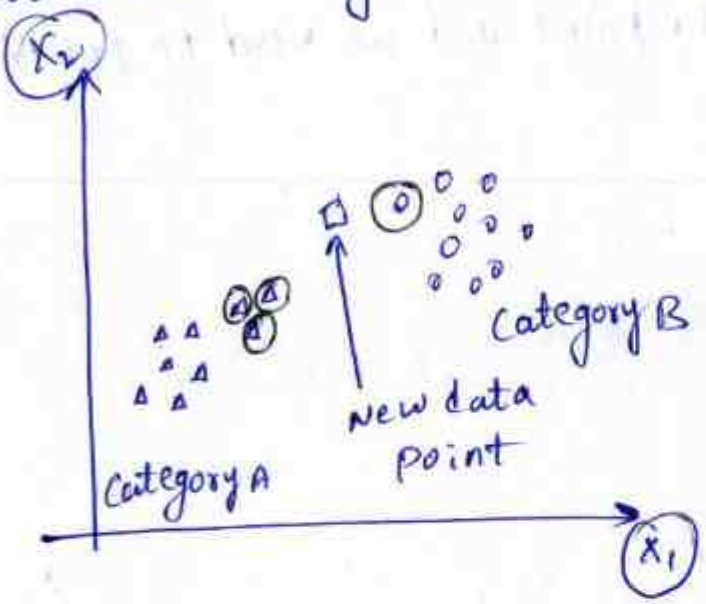
Next, we will calculate the Euclidean distance b/w the data points.

The Euclidean distance is the distance b/w two points, which we have already studied in geometry. it can be calculated as:



Euclidean distance b/w A_1 and $B_2 = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$

- By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B.



Category A : 3 neighbors
Category B : 2 neighbors

Advantages:

* it is simple to implement
* it is robust to the noisy training data
* it can be more effective if the training data is large.

Dis - Advantages:

* always needs to determine the value of K which may be complex some time.
* The computation cost is high because of calculating the distance b/w the data points for all the training samples.

Subject: ML
Faculty: S. Sandhya Rani
Topic: KNN example

Class Notes

Unit No: 4
Lecture No:
Link to Session
Planner (SP): S.No ... of SP
Book Reference:
Date Conducted:
Page No: 37

Example for Lazy learning :

Given data Query => x = (maths = 6, CS = 8)

and K = 3 (nearest neighbors)

classification - pass / Fail

	Maths	CS	Result
1)	4	3	F
2)	6	7	P
3)	7	8	P
4)	5	5	F
5)	8	8	P

Euclidean distance (d)

$$d = \sqrt{|x_{o_1} - x_{A_1}|^2 + |x_{o_2} - x_{A_2}|^2}$$

O - observed value
a - actual value.

1) calculate $d_1 = \sqrt{(6-4)^2 + (8-3)^2}$

$= \sqrt{2^2 + 5^2} = \sqrt{29} = 5.38$

2) $d_2 = \sqrt{(6-6)^2 + (8-7)^2} = \sqrt{0+1} = 1$

3) $d_3 = \sqrt{(6-7)^2 + (8-8)^2} = \sqrt{1+0} = 1$

$$d_4 = \sqrt{(6-5)^2+(8-5)^2} = \sqrt{1+9} = \sqrt{10} = 3.16$$

$$d_5 = \sqrt{(6-8)^2+(8-8)^2} = \sqrt{4+0} = \sqrt{4} = 2$$