MALINENI LAKSHMAIAH WOMEN'S ENGINEERING COLLEGE

(Approved by AICTE, Affiliated to JNTUK)(An ISO9001:2008 Certified Institution)

IV B.Tech (Common to CSE and IT), IV-I Semester, R19, Machine Learning Notes, UNIT-V

Prepared by Dr.M.BHEEMALINGAIAH

UNIT V

Bayesian Learning: Probability theory and Bayes rule. Naive Bayes learning algorithm. Parameter smoothing. Generative vs. discriminative training. Logistic regression. Bayes nets and Markov nets for representing dependencies.

Instance-Based Learning: Constructing explicit generalizations versus comparing to past specific examples. K-Nearest-neighbor algorithm. Case-based learning

Content

5.1 Bayesian learning

Features of Bayesian learning methods*:*

Each observed training example can incrementally decrease or increase the estimatedprobability that a hypothesis is correct.

- This provides a more flexible approach to learning than algorithms that completelyeliminate a hypothesis if it is found to be inconsistent with any single example.

Prior knowledge can be combined with observed data to determine the final probability of a hypothesis. In Bayesian learning, prior knowledge is provided byasserting

- a prior probability for each candidate hypothesis, and
- a probability distribution over observed data for each possible hypothesis.

Bayesian methods can accommodate hypotheses that make probabilistic predictions

New instances can be classified by combining the predictions of multiple hypotheses,weighted by their probabilities.

•Even in cases where Bayesian methods prove computationally intractable, they can provide a standard of optimal decision making against which other practical methodscan be measured.

Bayesian learning

- In machine learning, we try to determine the **best hypothesis** from somehypothesis space H, given the observed training data D.
- In Bayesian learning, the **best hypothesis** means the **most probable** hypothesis, given the data D plus any initial knowledge about the priorprobabilities of the various hypotheses in H.
- Bayes theorem provides a way to calculate the probability of a hypothesis based on its prior probability, the probabilities of observing various data given the hypothesis, and the observed data itself.

In Bayesian learning the following four notations are used

1. P(h) is Prior probability of hypothesis h

- P(h) to denote the initial probability that hypothesis h holds, before observing training data.
- P(h) may reflect any background knowledge we have about the chance that h is correct. If we

have no such prior knowledge, then each candidate hypothesis might simply get the same prior probability.

2. P(D) is Prior probability of training data D

- The probability of D given no knowledge about which hypothesis holds

3. P(h | D) is Posterior probability of h given D

- P(h|D) is called the **posterior probability** of **h,** because it reflects our confidence that **h** holds after we have seen the training data **D.**
- The posterior probability P(h|D) reflects the influence of the training data **D,** in contrast tothe prior probability P(h), which is independent of D.

4. P(D | h) is Posterior probability of D given h

- The probability of observing data **D** given some world in which hypothesis **h** holds. Generally, we write **P(x | y)** to denote the probability of **event x** given **event y.**

In machine learning problems, we are interested in the probability P(h|D) that h holds given the observed training data D.

Bayes theorem provides a way to calculate the posterior probability P(h|D), from the prior probability P(h), together with P(D) and P(D|h).

Bayes Theorem:

$$\boxed{P(h \mid D) = \frac{P(D \mid h) P(h)}{P(D)}}$$

- P(h|D) increases with P(h) and P(D|h) according to Bayes theorem.
- P(h|D) decreases as P(D) increases, because the more probable it is that D will be observed independent of h, the less evidence D providesin support of h

Example : Sample Space for events A and B

A holds	T	T	F	F	T	F	T
B holds	T	F	T	F	T	F	F

P(A) = 4/7 , P(B) = 3/ 7, P(B|A) = 2/4 , P(A|B) = 2/3

P(B|A) = P(A|B)P(B) / P(A) = (2/3 * 3/7) / 4/7 = 2/4

P(B|A) = P(A|B)P(B) / P(A) = (2/3 * 3/7) / 4/7 = 2/4

Maximum A Posteriori (MAP) Hypothesis H_{map}.

•The learner considers some set of candidate hypotheses H and it is interested in finding the **most probable hypothesis** h Є H given theobserved data D

•Any such maximally probable hypothesis is called a **maximum aposteriori (MAP) hypothesis h_{MAP}.**

•We can determine the MAP hypotheses by using Bayes theorem to calculate the posterior probability of each candidate hypothesis.

$$\boxed{\begin{aligned} h_{MAP} &\equiv \underset{h\in H}{\operatorname{argmax}}\, P(h|D) \\ &= \underset{h\in H}{\operatorname{argmax}}\, \frac{P(D|h)P(h)}{P(D)} \\ &= \underset{h\in H}{\operatorname{argmax}}\, P(D|h)P(h) \end{aligned}}$$

Maximum Likelihood(ML) Hypothesis, H_{ML}•

- If we assume that every hypothesis in H is equally probable i.e. $P(h_i) = P(h_j)$ for all h_i and h_j in H

We can only consider P(D|h) to find the most probable hypothesis.

• P(D|h) is often called the likelihood of the data D given h

• Any hypothesis that maximizes P(D|h) is called a maximum likelihood (ML) hypothesis, h_{ML}.

$$h_{ML} \equiv \underset{h \in H}{\operatorname{argmax}} P(D|h)$$

Example - Does patient have cancer or not?

The test returns a correct positive result in only 98% of the cases in which the disease is actually present, and a correct negative result in only 97% of the cases in which thedisease is not present.

Furthermore, 0.008 of the entire population has cancer.

- P(cancer) = 0.008, P(notcancer) =0 .992
- P(+|cancer) =0 .98, P(-|cancer) =0 .02
- P(+|notcancer) =0.03, P(-|notcancer) =0.97

A patient takes a lab test and the result comes back positive

- P(+|cancer) P(cancer) = 0.98 *0.008 = 0.0078
- P(+|notcancer) P(notcancer) =0 .03 * 0.992 =0 .0298 -.> **h_{MAP} is notcancer**

Since P(cancer|+) + P(notcancer|+) must be 1

- P(cancer|+) = 0.0078 / (0.0078+0.0298) =0.21
- P(notcancer|+) = 0.0298 / (0.0078+0.0298) =0.79

Difficulties with Bayesian Methods

Require initial knowledge of many probabilities

- When these probabilities are not known in advance they are often estimated basedon background knowledge, previously available data, and assumptions about the form of the underlying distributions.

Significant computational cost is required to determine the Bayesoptimal hypothesis in the general case (linear in the number of candidate hypotheses).

In certain specialized situations, this computational cost can be significantlyreduced.

Application of Bayesian learning

5.2 Probability theory and Baye's rule

Bayes rule (Baye's Theorem)

$$P(A/B) = \frac{P(B/A) \times P(A)}{P(B)}$$

- $P(A/B)$ is Posterior Probability (Probability of A is being true given B is true)
- $P(B/A)$ is likelihood (Probability of B is being true given A is true) .
- P(A) is Prior Probability (Probability of A is being true)
- P(B) is Probability of B is being true)

Some Problems on Baye's Theorem

Problem -1: Consider there are three baskets, Basket I, Basket II and Basket III with each basket containing rings of red color and green color. Basket I contains 6red rings and 5 green rings. Basket II contains 3green rings and 2red rings while Basket III contains 6 rings which are all red. A person chooses a ring randomly from a basket. If the ring picked is red, find the probability that it was taken from Basket II.

Solution:

Consider there are three bags Bag I, Bag II and Bag III with each bag containing balls of blue color and yellow color. In Bag I, there are 4 blue balls and 7 yellow balls. Bag II contains 5 blue balls and 4 yellow balls while Bag III contains 3 blue balls and 6 yellow balls. A person chooses a ball randomly from a bag. If the ball he has picked is yellow, find the probability that it was taken from Bag III.

Let the events A_1, A_2, and A_3 are choosing from Basket I, Basket II and Basket III. Let E be the event of picking a red ring.

$P(A_1) = P(A_2) = P(A_3) = 1/3$

P (Picking a red ring from Basket I) = P(E | A_1) = 6 /11

P (Picking a red ring from Basket II) = P(E | A_2) = 2 /5

P (Picking a red ring from Basket III) = P(E | A_3) = 6 /6

Now to find the probability of picking a ring from Basket II given that it is red i.e., P(A_2 | E)

Applying Bayes theorem,

$$P(A_2 \mid E) = \frac{P(E|A2)P(A2)}{P(A1)P(E|A1) + P(A2)P(E|A2) + P(A3)P(E|A3)}$$

$$= \frac{\frac{2}{5}*\frac{1}{3}}{\frac{1}{3}*\frac{6}{11}+\frac{1}{3}*\frac{2}{5}+\frac{1}{3}*\frac{6}{6}} = \frac{\frac{2}{15}}{\frac{107}{165}} = 22/107 = 0.2056$$

Problem-2 : Assume the following probabilities, the probability of a person having Malaria to be 0.02%, the probability of the test to be positive on detecting Malaria, given that the person has Malaria is 98% and similarly the probability of the test to be negative on detecting Malaria, given that the person doesn’t have malaria to be 95%. Find the probability of a person having Malaria; given that, the test result is positive.

Solution:

P (Hypothesis h = ‘Person having Malaria’) = 0.02

P(Evidence E = ‘Test is Positive’ | Hypothesis h = ‘Person has Malaria’) = 98%

P(Evidence E = ‘Test is Negative’ | Hypothesis h = ‘Person doesn’t have Malaria’) = 95%

P(Evidence E = ‘Test is Positive’) = [P(Evidence E = ‘Test is Positive’)* Sensitivity] + [P(Evidence E = ‘Test is Negative’)* (1-Specificity)]

P(Evidence E = ‘Test is Positive’) = [0.02 * 0.98] + [0.98 * (1-0.95)] = 0.0686

P(Hypothesis h = ‘Person has Malaria’ | Evidence E = ‘Test is Positive’) =

P(Hypothesis h = ‘Person having Malaria’) * P(Evidence E = ‘Test is Positive’ | Hypothesis h = ‘Person has Malaria’) / P(Evidence E = ‘Test is Positive’)= $\frac{0.02 * 0.98}{0.0686}$ = 0.2857

5.3 Naive Bayes learning algorithm

Principle of Naive Bayes Classifier:

A Naive Bayes classifier is a probabilistic machine learning model that’s used for classification task. The classifier is based on the Bayes theorem.

Bayes Theorem:

$$\boxed{P(A|B) = \frac{P(B|A)P(A)}{P(B)}}$$

Using Bayes theorem, we can find the probability of **A** happening, given that **B** has occurred. Here, **B** is the evidence and **A** is the hypothesis. The assumption made here is that the predictors/features are independent. That is presence of one particular feature does not affect the other. Hence it is called naive.

According to this example, Bayes theorem can be rewritten as:

$$\boxed{P(y|X) = \frac{P(X|y)P(y)}{P(X)}}$$

The variable **y** is the class variable (label) which represents output. For example Playtennis training dataset, there are two class labels yes or no given the conditions. Variable **X** represents the attributes /features. **X** is given as,

$$X = (x_1, x_2, x_3, \ldots.., x_n)$$

Here x_1, x_2….x_n represent the features/attributes of training dataset,

For example in Playtennis training dataset, (Outlook, Temp, Humidity and Wind) .

. By substituting for **X** and expanding using the chain rule we get,

$$\boxed{P(y|x_1, ..., x_n) = \frac{P(x_1|y)P(x_2|y)...P(x_n|y)P(y)}{P(x_1)P(x_2)...P(x_n)}}$$

Now, you can obtain the values for each by looking at the dataset and substitute them into the equation. For all entries in the dataset, the denominator does not change, it remain static. Therefore, the denominator can be removed and proportionality can be introduced.

$$\boxed{P(y|x_1, ..., x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|y)}$$

In our case, the class variable(**y**) has only two outcomes, yes or no. There could be cases where the classification could be multivariate. Therefore, we need to find the class **y** with maximum probability.

$$\boxed{y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)}$$

Using the above function, we can obtain the class, given the predictors.

5.3.1 Advantages of Naïve Bayes Classifier:

- Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.
- It can be used for Binary as well as Multi-class Classifications.
- It performs well in Multi-class predictions as compared to the other Algorithms.
- It is the most popular choice for text classification problems.

- When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
- It performs well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption.

5.3.2 Disadvantages of Naïve Bayes Classifier:

- Naive Bayes assumes that all features are independent or unrelated, so it cannot learn the relationship between features.
- If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as "Zero Frequency". To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.
- On the other side naive Bayes is also known as a bad estimator, so the probability outputs from predict_proba are not to be taken too seriously.
- Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

5.3.3 Applications of Naïve Bayes Classifier:

- It is used for Credit Scoring.
- It is used in medical data classification.
- It can be used in real-time predictions because Naïve Bayes Classifier is an eager learner.
- It is used in Text classification such as Spam filtering and Sentiment analysis.
- **Real time Prediction:** Naive Bayes is an eager Learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.
- **Multi class Prediction:** This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.
- **Text classification/ Spam Filtering/ Sentiment Analysis:** Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)

- **Recommendation System:** Naive Bayes Classifier and Collaborative Filtering together builds a Recommendation System that uses machine Learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not.

5.3.4 Types of Naïve Bayes Model:

There are three types of Naive Bayes Model, which are given below:

- **Gaussian**: The Gaussian model assumes that features follow a normal distribution. This means if predictors take continuous values instead of discrete, then the model assumes that these values are sampled from the Gaussian distribution.
- **Multinomial**: The Multinomial Naïve Bayes classifier is used when the data is multinomial distributed. It is primarily used for document classification problems; it means a particular document belongs to which category such as Sports, Politics, education, etc. The classifier uses the frequency of words for the predictors.
- **Bernoulli**: The Bernoulli classifier works similar to the Multinomial classifier, but the predictor variables are the independent Booleans variables. Such as if a particular word is present or not in a document. This model is also famous for document classification ta

Problems on Naïve Bayes Model:

5.3.5 Problem -1 .Take a real time example of predicting the result of a student using Naïve Bayes algorithm. The training dataset T consists of 8 data instances with attributes such as 'Assessment', 'Assignment', 'Project' and ‘Seminar' as shown in the below Table 8.17. The target variable is Result which is classified as Pass or Fail for a candidate student. Given a test data to be (Assessment = Average, Assignment = Yes, Project = No and Seminar = Good), predict the result of the student. Apply Laplace Correction if Zero probability problem occurs.

Table 1: Training Dataset

S.No.	Assessment	Assignment	Project	Seminar	Result
1.	Good	Yes	Yes	Good	Pass
2.	Average	Yes	No	Poor	Fail
3.	Good	No	Yes	Good	Pass
4.	Average	No	No	Poor	Fail
5.	Average	No	Yes	Good	Pass
6.	Good	No	No	Poor	Pass
7.	Average	No	Yes	Good	Fail

8.	Good	Yes	Yes	Poor	Pass

Solution:

Step 1: Compute the prior probability for the target feature ‘Result’. It has two classes ‘Pass’ and ‘Fail’.

From the training data set, we observe that the frequency or the number of instances with ‘Result = Pass’ is 5 and ‘Result = Fail’ is 3.

The prior probability for ‘Result = Pass’ is 5/8 and ‘Result = Fail’ is 3/8.

Step 2: Compute Frequency matrix and Likelihood Probability for each of the feature.

Step 2a: Feature - Assessment

Table 1 shows the frequency matrix for the feature Assessment.

Table 1 Frequency Matrix of Assessment

Assessment	Result = Pass	Result = Fail
Good	4	0
Average	2	2
Total	6	2

Table 2 shows how the likelihood probability is calculated for Assessment using conditional probability.

Table 2 Likelihood Probability of Assessment

Assessment	P(Result = Pass)	P(Result = Fail)
Good	4/6	0/2
Average	2/6	2/2

Step 2b: Feature - Assignment

Table 3 shows the frequency matrix for the feature Assignment.

Table 3 Frequency Matrix of Assignment

Assignment	Result = Pass	Result = Fail
YES	2	2
NO	3	1
Total	5	3

Table 4 shows how the likelihood probability is calculated for Assignment using conditional probability.

Table 4 Likelihood Probability of Assignment

Assignment	P(Result = Pass)	P(Result = Fail)
YES	2/5	**2/3**
NO	3/5	**1/3**

Step 2c: Feature – Project

Table 5 shows the frequency matrix for the feature Project.

Table 5 Frequency Matrix of Project

Project	Result = Pass	Result = Fail
YES	4	2
NO	1	1
Total	5	3

Table 6 shows how the likelihood probability is calculated for Project using conditional probability.

Table 6 Likelihood Probability of Project

Project	P(Result = Pass)	P(Result = Fail)
Yes	4/5	**2/3**
No	1/5	**1/3**

Step 2d: Feature – Seminar

Table 7 shows the frequency matrix for the feature Seminar.

Table 7 Frequency Matrix of Seminar

Seminar	Result = Pass	Result = Fail
Good	3	2
Poor	1	2
Total	4	4

Table 8 shows how the likelihood probability is calculated for Seminar using conditional probability.

Table 8 Likelihood Probability of Seminar

Seminar	P(Result = Pass)	P(Result = Fail)
Good	¾	2/4
Poor	¼	2/4

Step 3: Use Bayes theorem to calculate the probability of all hypotheses.

Given the test data = (Assessment = Average, Assignment = Yes, Project = No and Seminar = Good)) apply the Bayes theorem to classify whether the given student Result is Pass or Fail.

P (Result= Pass | Test data) = (P (Assessment = Average |Result = Pass) P(Assignment = Yes | Result = Pass) P(Project = No | Result = Pass) P(Seminar = Good | Result = Pass) P(Result = Pass)))/((P(Test Data))

We can ignore P (Test Data) in the denominator since it is common for all cases to be considered.

Hence P (Result = Pass | Test data) = $\left(\frac{2}{6}\times\frac{2}{5}\times\frac{1}{5}\times\frac{3}{5}\right)\times\left(\frac{5}{8}\right) = \mathbf{0.0125}$

(2/6 * 2/5 * 1/5 * 3/4 *5/8= **0.0125)**

Similarly, for the other case 'Result = Fail',

We compute the probability,

P(Result = Fail | Test data) =(P (Assessment = Average |Result = Fail) P(Assignment = Yes | Result = Fail) P(Project = No | Result = Fail) P(Seminar = Good | Result = Fail) P(Result = Fail))/(P(Test Data))= $\left(\frac{2}{2}\times\frac{2}{3}\times\frac{1}{3}\times\frac{2}{4}\right)\times\left(\frac{3}{8}\right)$== **0.0417**

Step 4: Use Maximum A Posteriori (MAP) Hypothesis, $\boldsymbol{h_{MAP}}$ to classify the test object to the hypothesis with the highest probability.

Since P(Result = Fail | Test data) has the highest probability value, the test data is classified as '**Result= Fail'**.

There is no Zero Probability Error, so the algorithm stops.

5.3. 6 Problem -2 : Let us take an example to get some better intuition. Consider the problem of playing tennis. The dataset is represented as below

Table 2 : Training Data Set

Day	Outlook	Temp	Humidity	Wind	Pay Tennis (decision)
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Step 1: Estimate frequencies and probabilities for each feature of all their instances

Total 4 features outlook, temperature, Humidity and Wind with total 14 instances

Output column is classification column that is Playtennis

Playtennis			
Classification	Yes	No	Total
	9	5	14

1. Outlook				
Instances		Yes	No	Total
1	sunny	2	3	5
2	overcast	4	0	4
3	rainy	3	2	5
Total		9	5	14

2.Temperate				
Instances		Yes	No	Total
1	Hot	2	2	4
2	Mild	4	2	6
3	Cool	3	1	4
Total		9	5	14

3.Humidity				
Instances		Yes	No	Total
1	High	3	4	7
2	Normal	6	1	7
Total		9	5	14

4. wind				
Instances		Yes	No	Total
1	Strong	3	3	6
2	Week	6	2	8
Total		9	5	14

Step 2: Estimate Probability for each feature

Playtennis			
Classification	Yes	No	Total
	9	5	14

1. Outlook				
Instances		Yes	No	
1	sunny	2/9	3/5	
2	overcast	4/9	0/5	
3	rainy	3/9	2/5	

2.Temperate				
Instances		Yes	No	
1	Hot	2/9	2/5	
2	Mild	4/9	2/5	
3	Cool	3/9	1/5	

2. Humidity				
Instances		Yes	No	
1	High	3/9	4/5	
2	Normal	6/9	1/5	

4. wind				
Instances		Yes	No	
1	Strong	3/9	3/5	
2	Week	6/9	2/5	

Step 3: Estimate probability of new given instance using Naive Bayes

Outlook	Temperature	Humidity	Wind	Play Tennis
Sunny	Cool	High	Strong	?

N’= (Outlook=Sunny, Temperature=Cool, Humidity=High, Wind=Strong)

Using step2, compute the probabilities to Play=Yes

- P (Outlook=Sunny| Play=Yes) = 2/9
- P (Temperature=Cool | Play=Yes) = 3/9
- P (Humanity=High | Play=Yes) = 3/9
- P (Wind=Strong | Play=Yes) = 3/9
- P (Play=Yes) = 9/14

Using step2, compute the probabilities to Play=No

- P (Outlook=Sunny | Play=No) = 3/5
- P (Temperature=Cool | Play==No) = 1/5
- P (Humanity=High | Play=No) = 4/5
- P (Wind=Strong | Play=No) = 3/5
- P (Play=No) = 5/14

Hence total probabilities

$$\mathbf{P}(\mathbf{Yes}\,|\,\mathbf{x}') = \left[\mathbf{P}(\mathbf{Sunny}\,|\,\mathbf{Yes}) \times \mathbf{P}(\mathbf{Cool}\,|\,\mathbf{Yes}) \times \mathbf{P}(\mathbf{High}\,|\,\mathbf{Yes}) \times \mathbf{P}(\mathbf{Strong}\,|\,\mathbf{Yes})\right] \times \mathbf{P}(\mathbf{Play} = \mathbf{Yes})$$

$$P(Yes\,|\,\mathbf{x}') = \left(\frac{2}{9}\times\frac{3}{9}\times\frac{3}{9}\times\frac{3}{9}\right)\times\frac{9}{14} = 0.0082\times 0.6428 = 0.0053$$

$$\mathbf{P}(\mathbf{No}\,|\,\mathbf{x}') = \left[\mathbf{P}(\mathbf{Sunny}\,|\,\mathbf{No}) \times \mathbf{P}(\mathbf{Cool}\,|\,\mathbf{No}) \times \mathbf{P}(\mathbf{High}\,|\,\mathbf{No}) \times \mathbf{P}(\mathbf{Strong}\,|\,\mathbf{No})\right] \times \mathbf{P}(\mathbf{Play} = \mathbf{No})$$

$$\mathbf{P}(\mathbf{No}\,|\,\mathbf{x}') = \left(\frac{3}{5}\times\frac{1}{5}\times\frac{4}{5}\times\frac{3}{5}\right)\times\frac{5}{14} = 0.077\times 0.3571 = 0.0275$$

Conclusions: Given the fact P (Yes’) < P (No | x’), we label x’ to be “No”. As per computed probabilities of both Yes and No **.**Probability of play tennis no is higher than probability of play tennis yes .Given day weather conditions don’t support to pay tennis

5.3.4 Parameter smoothing

Laplace Smoothing

Laplace smoothing is a smoothing technique that handles the problem of zero probability in Naïve Bayes. Using Laplace smoothing, we can represent P(w’|positive) as

$$P(w'|positive) = \frac{\text{number of reviews with w' and y} = \text{positive} \; + \; \alpha}{\text{N} \; + \; \alpha * K}$$

Here,

alpha represents the smoothing parameter,

K represents the number of dimensions (features) in the data, and

N represents the number of reviews with y=positive

If we choose a value of alpha!=0 (not equal to 0), the probability will no longer be zero even if a word is not present in the training dataset.

Interpretation of changing alpha

Let’s say the occurrence of word w is 3 with y=positive in training data. Assuming we have 2 features in our dataset, i.e., K=2 and N=100 (total number of positive reviews).

$$P(w|positive) = \frac{3 + \alpha}{100 + 2 * \alpha}$$

Case 1- when alpha=1

P(w’|positive) = 3/102

Case 2- when alpha = 100

P(w’|positive) = 103/300

Case 3- when alpha=1000

P(w’|positive) = 1003/2100

As alpha increases, the likelihood probability moves towards uniform distribution (0.5). Most of the time, alpha = 1 is being used to remove the problem of zero probability.

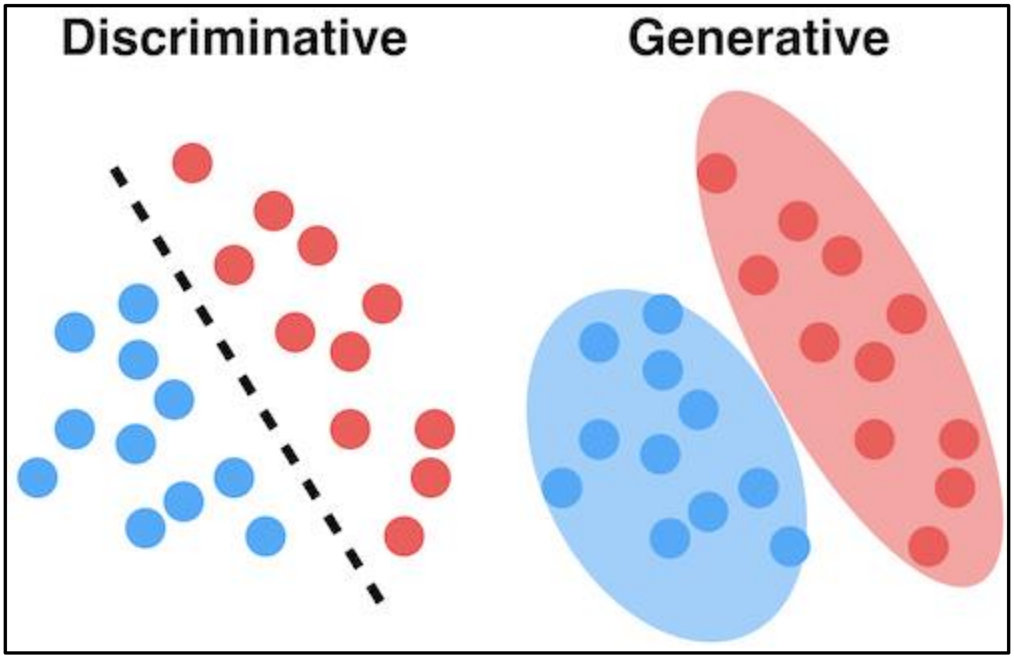
Conclusion

Laplace smoothing is a smoothing technique that helps tackle the problem of zero probability in the Naïve Bayes machine learning algorithm. Using higher alpha values will push the likelihood towards a value of 0.5, i.e., the probability of a word equal to 0.5 for both the positive and negative reviews. Since we are not getting much information from that, it is not preferable. Therefore, it is preferred to use alpha=1.

5.4 Generative vs. discriminative training models

Generative models: Generative models are models where the focus is the distribution of individual classes in a dataset and the learning algorithms tend to model the underlying patterns/distribution of the data points. These models use the intuition of joint probability in theory, creating instances where a given feature (*x*)/input and the desired output/label (*y*) exist at the same time.

Generative models use probability estimates and likelihood to model data points and distinguish between different class labels in a dataset. These models are capable of generating new data instances. However, they also have a major drawback. The presence of outliers affects these models to a significant extent.



Examples of machine learning generative models

- Naive Bayes (and generally Bayesian networks)
- Hidden Markov model
- Linear discriminant analysis (LDA),
- Dimensionality reduction technique

Discriminative models: Discriminative models, also called *conditional models*, tend to learn the boundary between classes/labels in a dataset. Unlike generative models, the goal here is to find the *decision boundary* separating one class from another.
So while a generative model will tend to model the joint probability of data points and is capable of creating new instances using probability estimates and maximum likelihood, discriminative models (just as in the literal meaning) separate classes by rather modeling the conditional probability and do not make any assumptions about the data point. They are also not capable of generating new data instances.
Discriminative models have the advantage of being more robust to outliers, unlike the generative models.
However, one major drawback is a *misclassification problem*, i.e., wrongly classifying a data point.

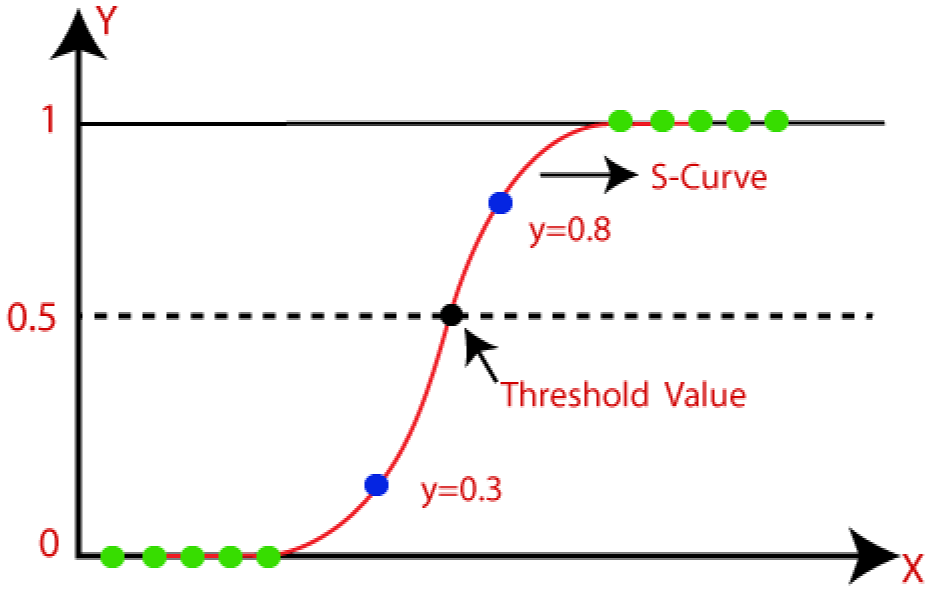
Another key difference between these two types of models is that while a generative model focuses on explaining how the data was generated, a discriminative model focuses on predicting labels of the data.

Examples of discriminative models in machine learning are:

- Logistic regression
- Support vector machine
- Decision tree
- Random forest

5.5 Logistic regression

- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.
- Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
- The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
- Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.
- Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:



Note: Logistic regression uses the concept of predictive modeling as regression; therefore, it is called logistic regression, but is used to classify samples; Therefore, it falls under the classification algorithm.

Logistic Function (Sigmoid Function):

The sigmoid function is a mathematical function used to map the predicted values to probabilities.

$$y = f(x) = \frac{1}{1+e^{-x}}$$

It maps any real value into another value within a range of 0 and 1.

The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.

In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

Assumptions for Logistic Regression:

The dependent variable must be categorical in nature.

The independent variable should not have multi-collinearity.

Logistic Regression Equation:

The Logistic regression equation can be obtained from the Linear Regression equation. The mathematical steps to get Logistic Regression equations are given below:

We know the equation of the straight line can be written as:

$$y= b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y):

$$\frac{\mathbf{y}}{\mathbf{1-y}}\text{; 0 for y= 0, and infinity for y=1}$$

But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will become:

$$\boldsymbol{log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n}$$

The above equation is the final equation for Logistic Regression.

Type of Logistic Regression:

On the basis of the categories, Logistic Regression can be classified into three types:

Binomial: In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.

Multinomial: In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"

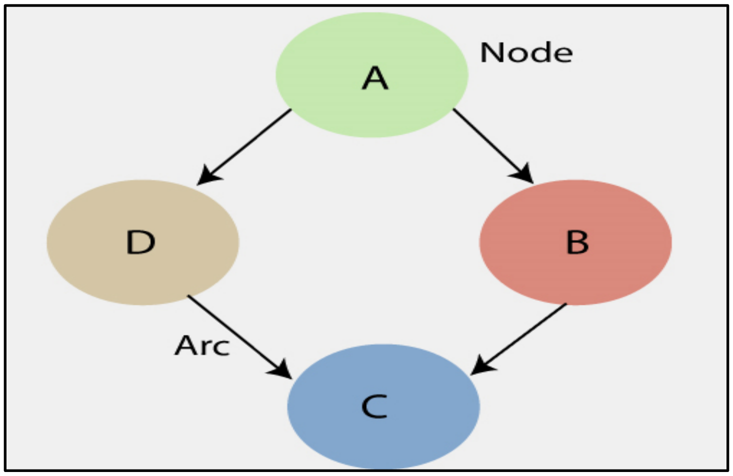
Ordinal: In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

5.6 Bayesian Belief Network

- Bayesian belief network is key computer technology for dealing with probabilistic events and to solve a problem which has uncertainty. We can define a Bayesian network as:
- "A Bayesian network is a probabilistic graphical model which represents a set of variables and their conditional dependencies using a directed acyclic graph."
- It is also called a Bayes network, belief network, decision network, or Bayesian model.
- Bayesian networks are probabilistic, because these networks are built from a probability distribution, and also use probability theory for prediction and anomaly detection.
- Real world applications are probabilistic in nature, and to represent the relationship between multiple events, we need a Bayesian network. It can also be used in various tasks including prediction, anomaly detection, diagnostics, automated insight, reasoning, time series prediction, and decision making under uncertainty.
- Bayesian Network can be used for building models from data and experts opinions, and it consists of two parts:

  - **Directed Acyclic Graph**
  - **Table of conditional probabilities.**

The generalized form of Bayesian network that represents and solve decision problems under uncertain knowledge is known as an **Influence diagram**. A Bayesian network graph is made up of nodes and Arcs (directed links), where:



- Each **node** corresponds to the random variables, and a variable can be **continuous** or **discrete**.
- **Arc or directed arrows** represent the causal relationship or conditional probabilities between random variables. These directed links or arrows connect the pair of nodes in the graph. These links represent that one node directly influence the other node, and if there is no directed link that means that nodes are independent with each other

In the above diagram, A, B, C, and D are random variables represented by the nodes of the network graph.

- If we are considering node B, which is connected with node A by a directed arrow, then node A is called the parent of Node B. Node C is independent of node A.

- Note: The Bayesian network graph does not contain any cyclic graph. Hence, it is known as a directed acyclic graph or DAG.

- The Bayesian network has mainly two components:
    - **Causal Component**
    - **Actual numbers**
  - Each node in the Bayesian network has condition probability distribution **P(X_i |Parent(X_i))**, which determines the effect of the parent on that node.
  - Bayesian network is based on Joint probability distribution and conditional probability. So let's first understand the joint probability distribution:

Joint probability distribution:

If we have variables x1, x2, x3,....., xn, then the probabilities of a different combination of x1, x2, x3.. xn, are known as Joint probability distribution.

P[x_1, x_2, x_3,....., x_n], it can be written as the following way in terms of the joint probability distribution.

= P[x_1| x_2, x_3,....., x_n]P[x_2, x_3,....., x_n]

= P[x_1| x_2, x_3,....., x_n]P[x_2|x_3,....., x_n]....P[x_{n-1}|x_n]P[x_n].

In general for each variable Xi, we can write the equation as:

$P(X_i|X_{i-1},........., X_1) = P(X_i |Parents(X_i))$

Explanation of Bayesian network:

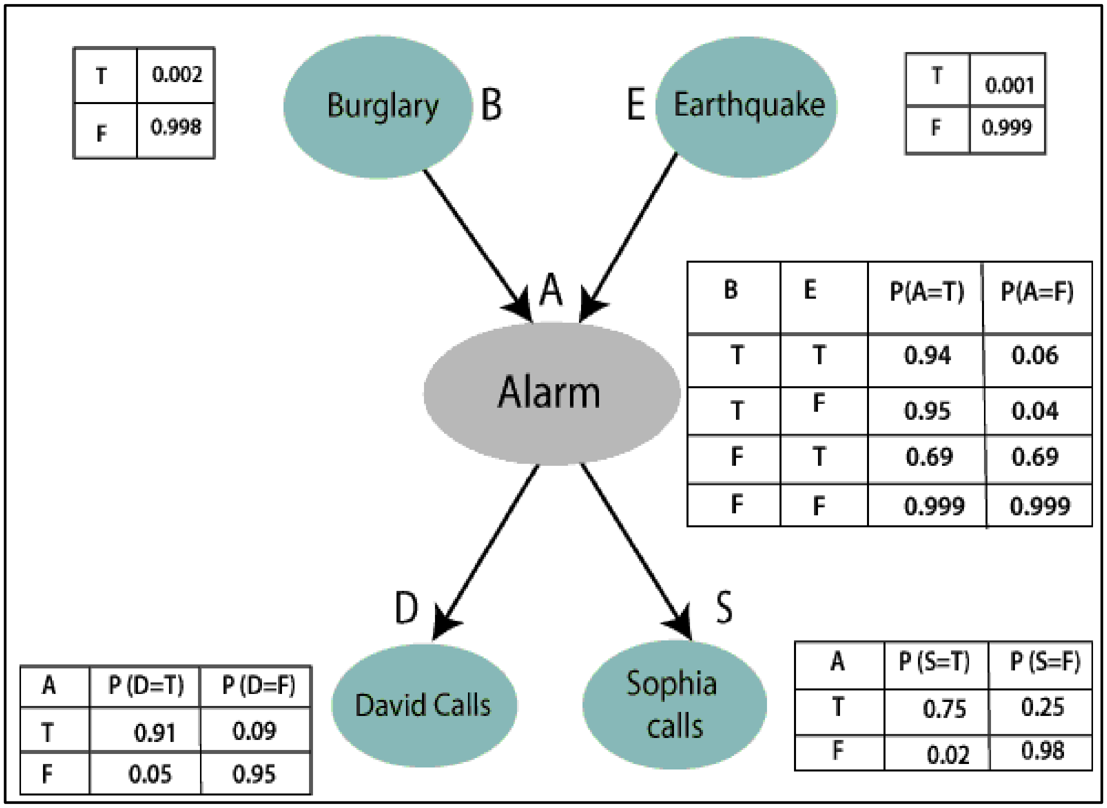
Let's understand the Bayesian network through an example by creating a directed acyclic graph:

Problems on Bayesian Belief Network

5.6.1 Problem-1: Example: Harry installed a new burglar alarm at his home to detect burglary. The alarm reliably responds at detecting a burglary but also responds for minor earthquakes. Harry has two neighbors David and Sophia, who have taken a responsibility to inform Harry at work when they hear the alarm. David always calls Harry when he hears the alarm, but sometimes he got confused with the phone ringing and calls at that time too. On the other hand, Sophia likes to listen to high music, so sometimes she misses to hear the alarm. Here we would like to compute the probability of Burglary Alarm.

Calculate the probability that alarm has sounded, but there is neither a burglary, nor an earthquake occurred, and David and Sophia both called the Harry**.**

Solution:



T	0.002
F	0.998

T	0.001
F	0.999

B	E	P(A=T)	P(A=F)
T	T	0.94	0.06
T	F	0.95	0.04
F	T	0.69	0.69
F	F	0.999	0.999

A	P (D=T)	P (D=F)
T	0.91	0.09
F	0.05	0.95

A	P (S=T)	P (S=F)
T	0.75	0.25
F	0.02	0.98

- The Bayesian network for the above problem is given below. The network structure is showing that burglary and earthquake is the parent node of the alarm and directly affecting the probability of alarm's going off, but David and Sophia's calls depend on alarm probability.

- The network is representing that our assumptions do not directly perceive the burglary and also do not notice the minor earthquake, and they also not confer before calling.
- The conditional distributions for each node are given as conditional probabilities table or CPT.
- Each row in the CPT must be sum to 1 because all the entries in the table represent an exhaustive set of cases for the variable.
- In CPT, a boolean variable with k boolean parents contains 2^K probabilities. Hence, if there are two parents, then CPT will contain 4 probability values

List of all events occurring in this network:

- Burglary (B)
- Earthquake(E)
- Alarm(A)
- David Calls(D)
- Sophia calls(S)

We can write the events of problem statement in the form of probability: **P[D, S, A, B, E]**, can rewrite the above probability statement using joint probability distribution:

P[D, S, A, B, E]= P[D | S, A, B, E]. P[S, A, B, E]

=P[D | S, A, B, E]. P[S | A, B, E]. P[A, B, E]

= P [D| A]. P [S| A, B, E]. P[A, B, E]

= P[D | A]. P[S | A]. P[A| B, E]. P[B, E]

= P[D | A]. P[S | A]. P[A| B, E]. P[B |E]. P[E]

Let's take the observed probability for the Burglary and earthquake component:

- P(B= True) = 0.002, which is the probability of burglary.
- P(B= False)= 0.998, which is the probability of no burglary.
- P(E= True)= 0.001, which is the probability of a minor earthquake
- P(E= False)= 0.999, Which is the probability that an earthquake not occurred.

We can provide the conditional probabilities as per the below tables:

Conditional probability table for Alarm A:

The Conditional probability of Alarm A depends on Burglar and earthquake:

B	E	P(A= True)	P(A= False)
True	True	0.94	0.06
True	False	0.95	0.04
False	True	0.31	0.69
False	False	0.001	0.999

Conditional probability table for David Calls:

The Conditional probability of David that he will call depends on the probability of Alarm.

A	P(D= True)	P(D= False)
True	0.91	0.09
False	0.05	0.95

Conditional probability table for Sophia Calls:

The Conditional probability of Sophia that she calls is depending on its Parent Node "Alarm."

A	P(S= True)	P(S= False)
True	0.75	0.25
False	0.02	0.98

From the formula of joint distribution, we can write the problem statement in the form of probability distribution:

$P(S, D, A, \neg B, \neg E) = P(S|A) *P(D|A)*P(A|\neg B \wedge \neg E) *P(\neg B) *P(\neg E)$.

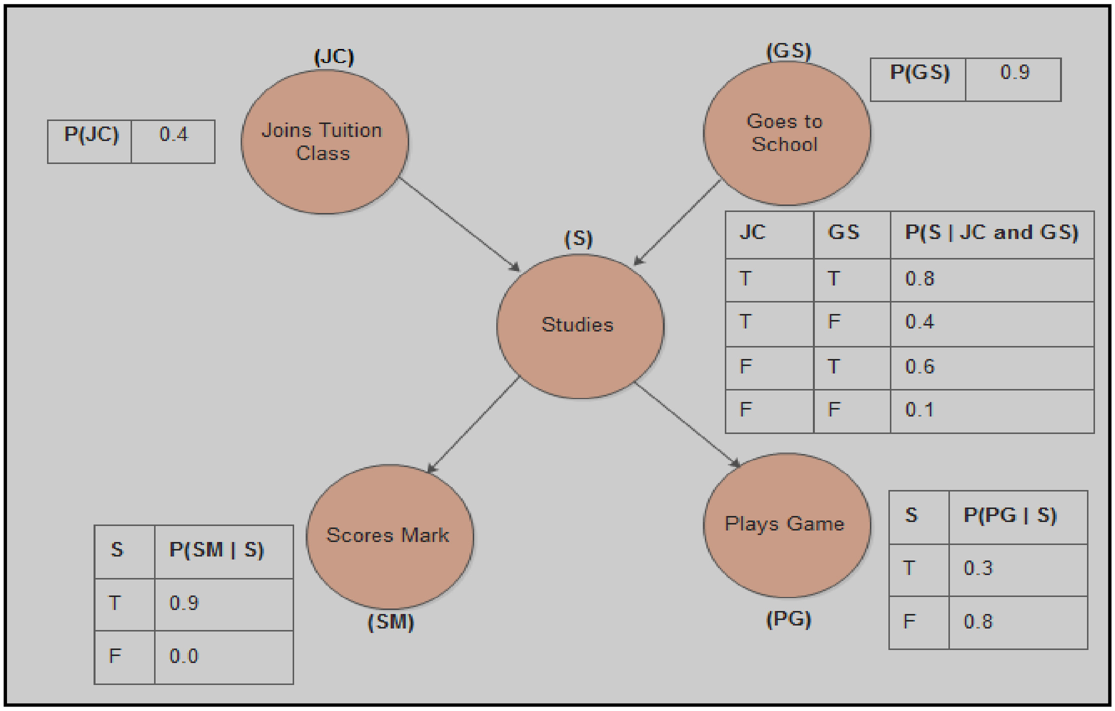
$= 0.75* 0.91* 0.001* 0.998*0.999$

= 0.00068045.

.

5.6.2 Problem-2:

Consider the scenario shown in Figure 9.16.Events‘student joins tuition classes’ and ‘goes to school daily’ have a direct effect on how the ‘student studies’. The event that the student studies have a direct effect on his scoring marks or playing games. What is the probability that he does not join tuition class, he goes to school daily, he studies, and he does not score marks?



P(JC)	0.4

P(GS)	0.9

JC	GS	P(S | JC and GS)
T	T	0.8
T	F	0.4
F	T	0.6
F	F	0.1

S	P(SM | S)
T	0.9
F	0.0

S	P(PG | S)
T	0.3
F	0.8

Solution:

$$P(\neg JC) * \ P(GS) * \ P(S|\ \neg JC \ and \ GS) * \ P(\neg SM|S)$$

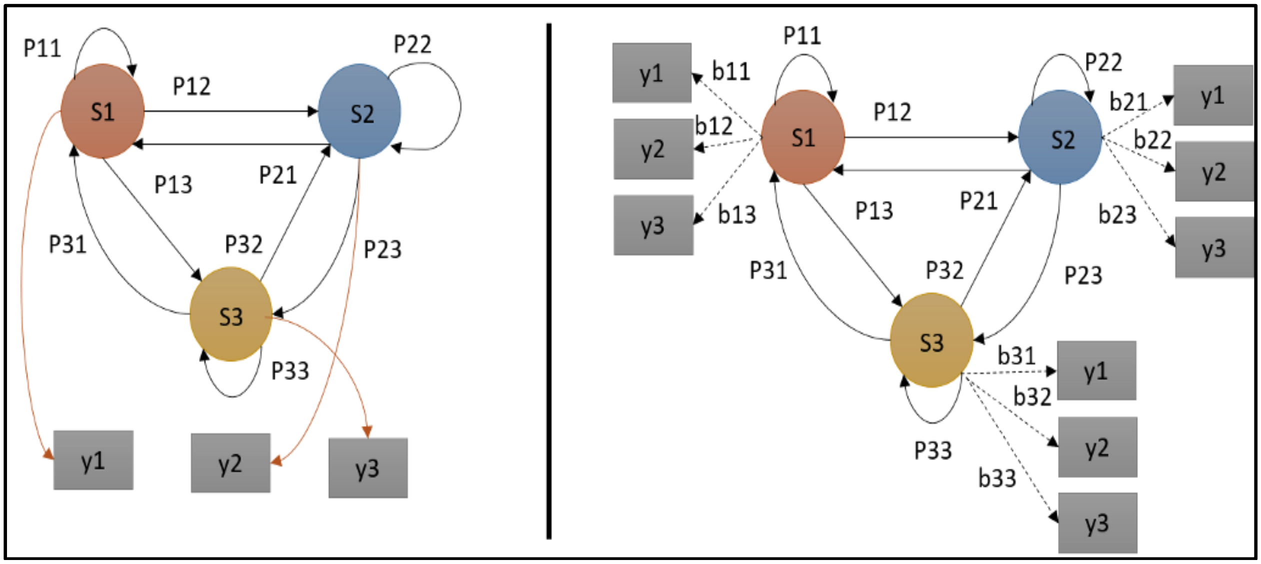
$$= 0.6 * 0.9 * 0.6 * 0.1$$

$$= 0.0324$$

5.7 Hidden Markov Model (Networks)

The Hidden Markov model is a probabilistic model which is used to explain or derive the probabilistic characteristic of any random process. It basically says that an observed event will not be corresponding to its step-by-step status but related to a set of probability distributions. Let’s assume a system that is being modelled is assumed to be a Markov chain and in the process, there are some hidden states. In that case, we can say that hidden states are a process that depends on the main Markov process/chain.

The main goal of HMM is to learn about a Markov chain by observing its hidden states. Considering a Markov process X with hidden states Y here the HMM solidifies that for each time stamp the probability distribution of Y must not depend on the history of X according to that time as shown in figure



Hidden Markov Network

5.7.1 Hidden Markov Model with an Example

To explain it more we can take the example of two friends, Rahul and Ashok. Now Rahul completes his daily life works according to the weather conditions. Major three activities completed by Rahul are- go jogging, go to the office, and cleaning his residence. What Rahul is doing today depends on whether and whatever Rahul does he tells Ashok and Ashok has no proper information about the weather But Ashok can assume the weather condition according to Rahul work.

Ashok believes that the weather operates as a discrete Markov chain, wherein the chain there are only two states whether the weather is Rainy or it is sunny. The condition of the weather cannot be observed by Ashok, here the conditions of the weather are hidden from Ashok. On each day, there is a certain chance that Bob will perform one activity from the set of the following activities {“jog”, “work”,” clean”}, which are depending on the weather. Since Rahul tells Ashok that what he has done, those are the observations. The entire system is that of a hidden Markov model (HMM).

Here we can say that the parameter of HMM is known to Ashok because he has general information about the weather and he also knows what Rahul likes to do on average.

So let’s consider a day where Rahul called Ashok and told him that he has cleaned his residence. In that scenario, Ashok will have a belief that there are more chances of a rainy day and we can say that belief Ashok has is the start probability of HMM let’s say which is like the following.

The states and observation are:

- states = ('Rainy', 'Sunny')
- observations = ('walk', 'shop', 'clean')
- And the start probability is:
- start_probability = {'Rainy': 0.6, 'Sunny': 0.4}

Now the distribution of the probability has the weightage more on the rainy day stateside so we can say there will be more chances for a day to being rainy again and the probabilities for next day weather states are as following

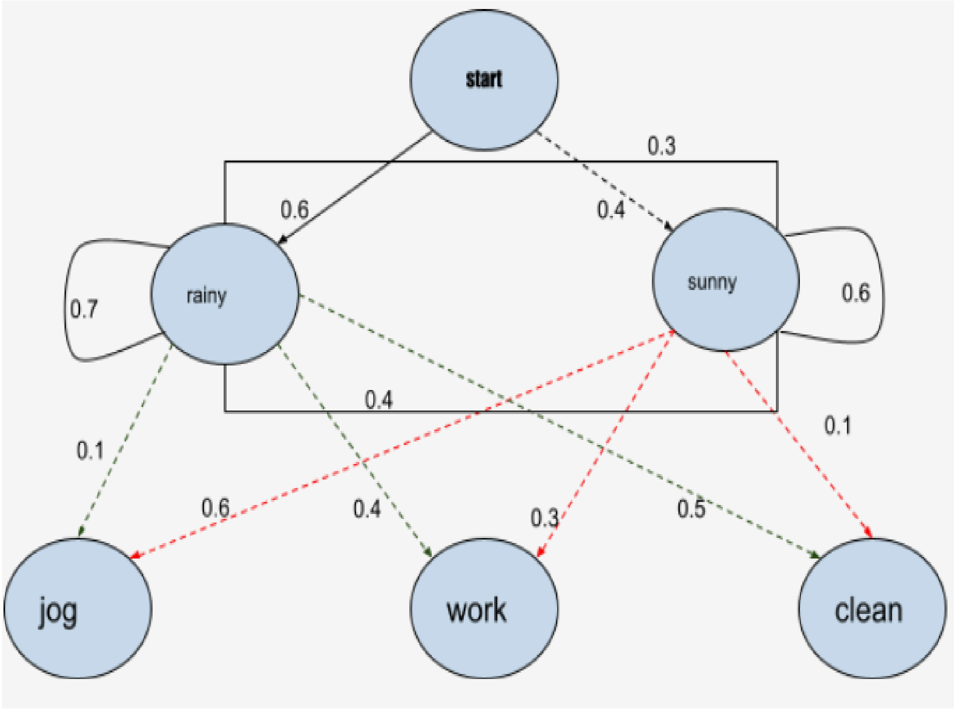
```
transition_probability = {    'Rainy' : {'Rainy': 0.7, 'Sunny': 0.3},
  'Sunny' : {'Rainy': 0.4, 'Sunny': 0.6},

  }
```

From the above we can say the changes in the probability for a day is transition probabilities and according to the transition probability the emitted results for the probability of work that Rhul will perform is

```
emission_probability = {     'Rainy' : {'jog': 0.1, 'work': 0.4, 'clean': 0.5},
  'Sunny' : {'jog': 0.6, 'work: 0.3, 'clean': 0.1},    }
```

This probability can be considered as the emission probability. Using the emission probability Ashok can predict the states of the weather or using the transition probabilities Ashok can predict the work which Rahul is going to perform the next day.

Below image shown the HMM process for making probabilities



So here from the above intuition and the example we can understand how we can use this probabilistic model to make a prediction. Now let’s just discuss the applications where it can be used.

5.7.2 Application of Hidden Markov Model

An application, where HMM is used, aims to recover the data sequence where the next sequence of the data cannot be observed immediately but the next data depends on the old sequences. Taking the above intuition into account the HMM can be used in the following applications:

- Computational finance
- speed analysis
- Speech recognition
- Speech synthesis
- Part-of-speech tagging
- Document separation in scanning solutions
- Machine translation
- Handwriting recognition
- Time series analysis
- Activity recognition
- Sequence classification
- Transportation forecasting

Hidden Markov Models in NLP

From the above application of HMM, we can understand that the applications where the HMM can be used have sequential data like time series data, audio, and video data, and text data or NLP data. In this article, our main focus is on those applications of NLP where we can use the HMM for better performance of the model, and here in the above-given list, we can see that one of the applications of the HMM is that we can use it in the Part-of-Speech tagging. Next in the article, we will see how we can use the HMM for POS-tagging**.**

5. 8 Instance-based learning

- The Machine Learning systems which are categorized as **instance-based learning** are the systems that learn the training examples by heart and then generalize to new instances based on some similarity measure. It is called instance-based because it builds the hypotheses from the training instances.
- It is also known as memory-based learning or lazy-learning. The time complexity of this algorithm depends upon the size of training data. The worst-case time complexity of this algorithm is **O (n)**, where n is the number of training instances.
- For example, If we were to create a spam filter with an instance-based learning algorithm, instead of just flagging emails that are already marked as spam emails, our spam filter would be programmed to also flag emails that are very similar to them. This requires a measure of resemblance between two emails.
- A similarity measure between two emails could be the same sender or the repetitive use of the same keywords or something else.

Advantages:

- Instead of estimating for the entire instance set, local approximations can be made to the target function.
- This algorithm can adapt to new data easily, one which is collected as we go .

Disadvantages:

- Classification costs are high
- Large amount of memory required to store the data, and each query involves starting the identification of a local model from scratch.

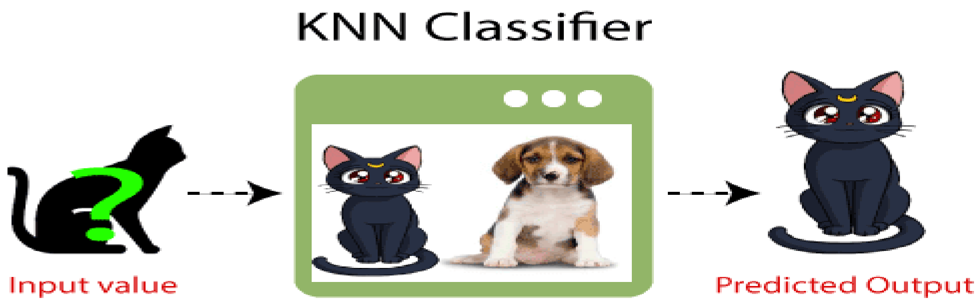
Some of the instance-based learning algorithms are :

- K- Nearest Neighbor (KNN)
- Self-Organizing Map (SOM)
- Learning Vector Quantization (LVQ)
- Locally Weighted Learning (LWL)

5. 9 K-Nearest Neighbor Algorithm

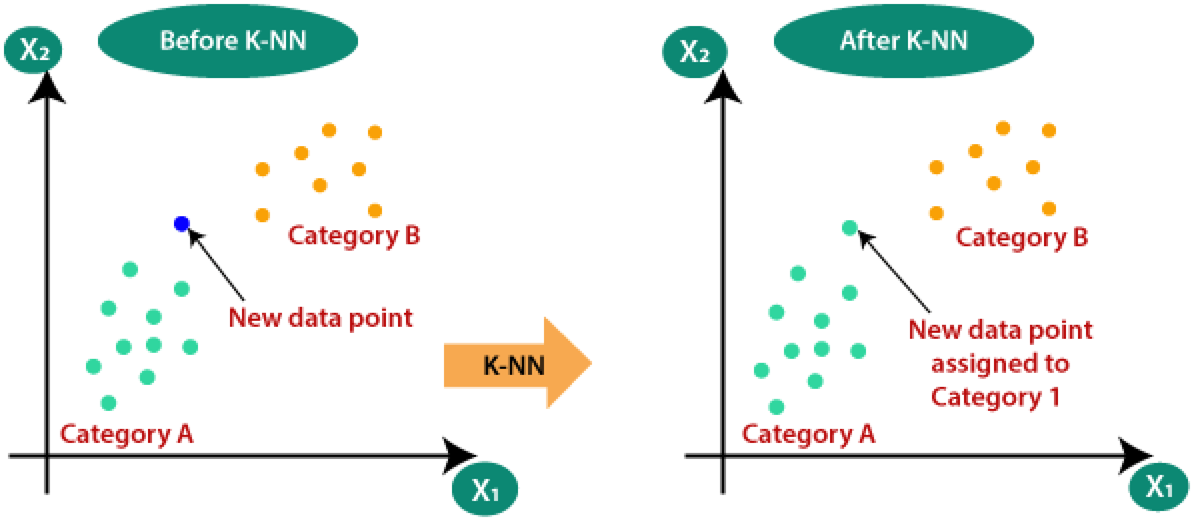
- K-Nearest Neighbor(K-NN) algorithm is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN 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 performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

Example: Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.



5.9.1 Why do we need a K-NN 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. Consider the below diagram:



5.9.2 How does K-NN work?

The K-NN working can be explained on the basis of the below algorithm:

Step-1: Select the number K of the neighbors

Step-2: Calculate the Euclidean distance of K number of neighbors

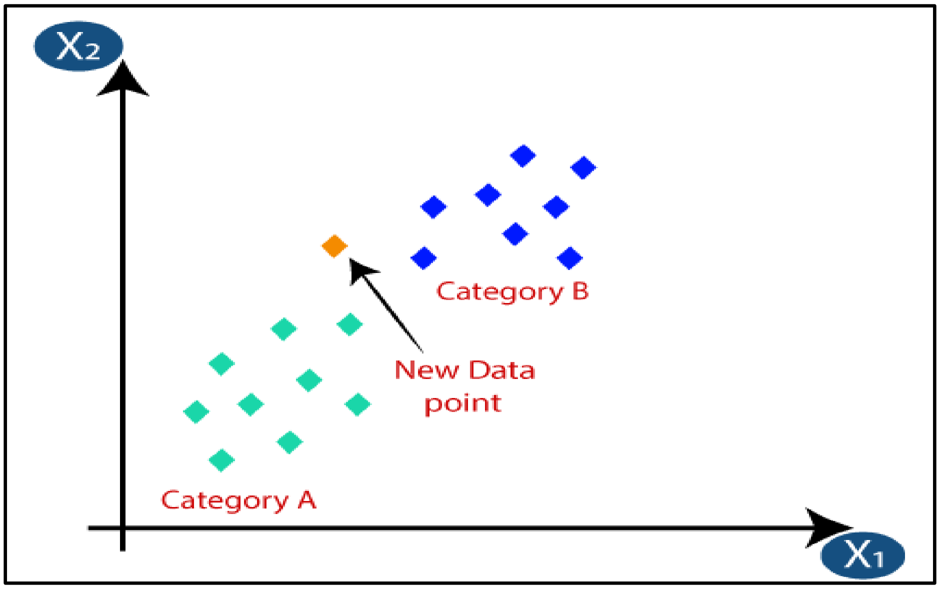
Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points in each category.

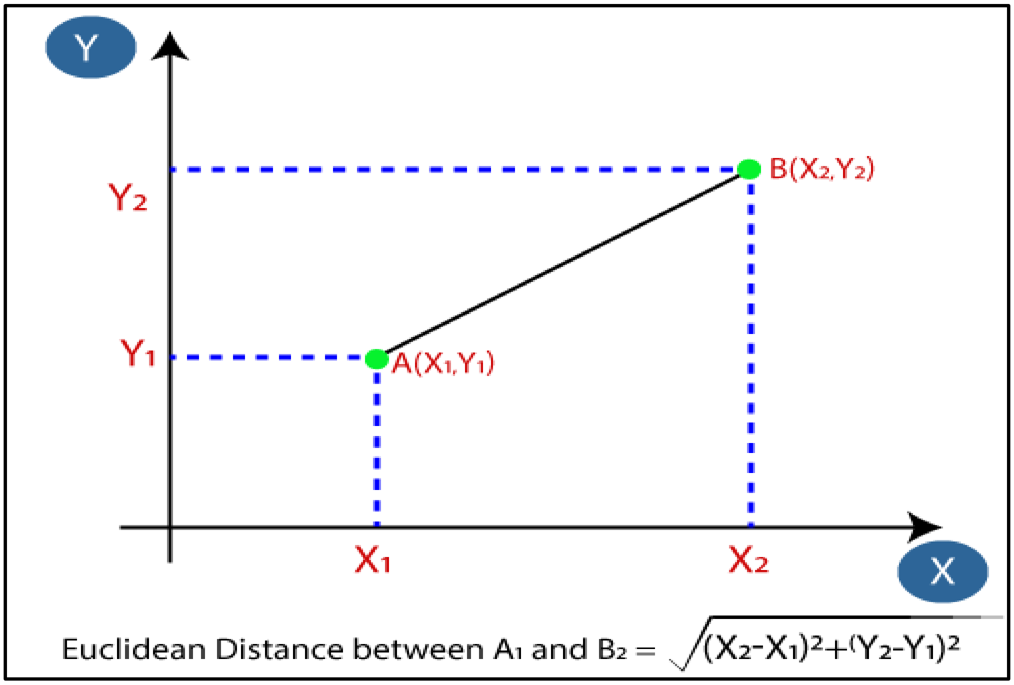
Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

Step-6: Our model is ready.

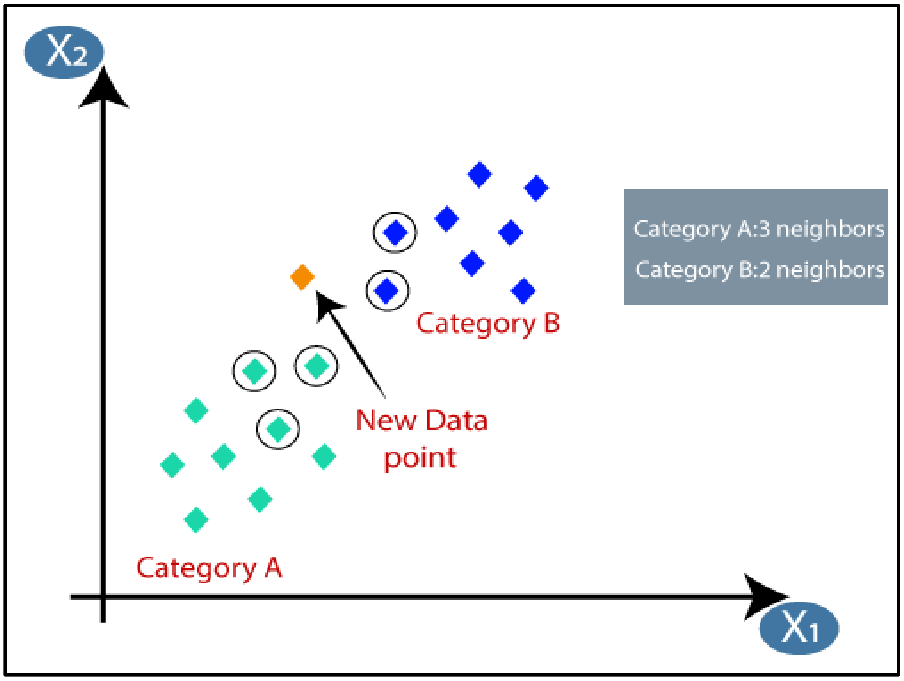
Suppose we have a new data point and we need to put it in the required category. Consider the below image:



- Firstly, we will choose the number of neighbors, so we will choose the k=5.
- Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



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



- As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

How to select the value of K in the K-NN Algorithm?
Below are some points to remember while selecting the value of K in the K-NN algorithm:

- There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
- A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
- Large values for K are good, but it may find some difficulties.

Advantages of KNN Algorithm:

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

Disadvantages of KNN Algorithm:

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples

5.9.3 Problem: Consider the following training data set of 10 data instances shown in Table 4.12 which describes the award performance of individual students based on GPA and No. of projects done. The target variable is 'Award' which is a discrete valued variable that takes 2 values 'Yes' or 'No'.

Table : Training Dataset

S.No.	GPA	No. of Projects done	Award
1.	9.5	5	Yes
2.	8.0	4	Yes
3.	7.2	1	No
4.	6.5	5	Yes
5.	9.5	4	Yes
6.	3.2	1	No
7.	6.6	1	No
8.	5.4	1	No
9.	8.9	3	Yes
10.	7.2	4	Yes

Given a test instance (GPA -7.8, No. of projects done - 4), use the training set to classify the test instance. Choose k=3 by using k-Nearest Neighbor classifier

Solution:

Step 1: Calculate the Euclidean distance between the test instance (GPA -7.8, No. of projects done - 4) and each of the training instances as shown in Table 1.

Euclidean Distance $d = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$

x_1=7.8, y_1=4

Table : Euclidean Distance

S.No.	GPA (x_1)	No. of Projects done (y_1)	Award	Euclidean Distance
1.	9.5	5	Yes	$D1 = \sqrt{(9.5-7.8)^2+(5-4)^2}$ $= 1.972308292$
2.	8.0	4	Yes	$D2 = \sqrt{(8.0-7.8)^2+(4-4)^2} = 0.2$
3.	7.2	1	No	$D3 = \sqrt{(7.2-7.8)^2+(1-4)^2}$ $= 3.059411708$
4.	6.5	5	Yes	$D4 = \sqrt{(6.2-7.8)^2+(5-4)^2}$ $= 1.640121947$
5.	9.5	4	Yes	$D5 = \sqrt{(9.5-7.8)^2+(4-4)^2} = 1.7$
6.	3.2	1	No	$D6 = \sqrt{(3.2-7.8)^2+(1-4)^2}$ $= 5.491812087$
7.	6.6	1	No	$D7 = \sqrt{(6.6-7.8)^2+(1-4)^2}$ $= 3.231098884$
8.	5.4	1	No	$D8 = \sqrt{(5.4-7.8)^2+(1-4)^2}$ $= 3.841874542$
9.	8.9	3	Yes	$D9 = \sqrt{(8.9-7.8)^2+(3-4)^2}$ $= 1.486606875$
10.	7.2	4	Yes	$D10 = \sqrt{(7.2-7.8)^2+(4-4)^2} = 0.6$

Step 2: Sort the distances in the ascending order and select the first 3 nearest training data instances to the test instance. The selected nearest neighbors are shown in Table 2. k=3

Table 2 Nearest Neighbors

Instance	Euclidean distance	Class
2	D2=0.2	Yes
10	D10=0.6	Yes
9	D9=1.487	Yes

Step 3: Predict the class of the test instance by majority voting.

The class for the test instance is predicted as "Yes".

.