SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | Approved by AICTE

SCSA1601
MACHINE LEARNING

Dr. B.Bharathi, M.E., Ph.D.,
Dr. R.Sathya Bama Krishna, M.E., Ph.D.,
Department of Computer Science and Engineering
School of Computing

SCSA1601	MACHINE LEARNING	L	T	P	Credits	Total Marks
		3	*	0	3	100

COURSE OBJECTIVES

- To focus on the construction and study of algorithms that can learn from data.
- To emphasize on the logical, knowledge-based approach.
- To introduce students to the basic concepts and techniques of Machine Learning.
- To develop skills of using recent machine learning software for solving practical problems.
- To gain experience of doing independent study and research.

UNIT 1 INTRODUCTION TO MACHINE LEARNING **9 Hrs.**

Machine learning - examples of machine learning applications - Learning associations - Classification -Regression - Unsupervised learning - Supervised Learning - Learning class from examples - PAC learning -Noise, model selection and generalization - Dimension of supervised machine learning algorithm.

UNIT 2 DECISION THEORY **9 Hrs.**

Bayesian Decision Theory- Introduction- Classification - Discriminant function-Bayesian networks-Association rule - Parametric Methods - Introduction - Estimation –Multivariate methods-Data Parameter estimation–Dimensionality Reduction- PCA-Linear discriminant analysis.

UNIT 3 CLUSTERING AND REGRESSION **9 Hrs.**

Clustering - Mixture densities - k-means clustering - Supervised Learning after clustering - Hierarchical clustering - Nonparametric Methods - Density estimation - Generalization of multivariate data - Smoothing models -Decision Trees - Univariate trees - Multivariate trees - Learning rules from data - Linear Discrimination-Gradient Descent.

UNIT 4 MULTILAYER PERCEPTRONS **9 Hrs.**

Structure of brain - Neural networks as a parallel processing - Perceptron - Multilayer perceptron - Back propagation - Training procedures - Tuning the network size - Learning time.

UNIT 5 LOCAL MODELS **9 Hrs.**

Competitive learning - Adaptive resonance theory - Self organizing map -Radial Basis functions - Bagging- Boosting- Reinforcement Learning.

Max.45 Hrs.

COURSE OUTCOMES

On completion of the course, student will be able to

CO1 - Understand complexity of Machine Learning algorithms and their limitations.
CO2 - Understand modern notions in data analysis oriented computing.
CO3 - Be capable of confidently applying common Machine Learning algorithms in practice and implementing their own.
CO4 - Be capable of performing distributed computations.
CO5 - Can demonstrate working knowledge of reasoning in the presence of incomplete and/or uncertain information.
CO6 - Gain ability to apply knowledge representation, reasoning, and machine learning techniques to real-world problems.

TEXT / REFERENCE BOOKS

1. Ethem Alpaydin, "Introduction to Machine Learning", MIT Press, 2004.
2. Tom Mitchell, "Machine Learning", McGraw Hill, 1997.
3. Shai Shalev-Shwartz and Shai Ben-David, "Understanding Machine Learning: From Theory to Algorithms", Cambridge University Press, 2014.

UNIT 2
DECISION THEORY

Bayesian Decision Theory - Introduction - Classification - Discriminant function - Bayesian networks - Association rule - Parametric Methods - Introduction - Estimation - Multivariate Methods - Data Parameter estimation - Dimensionality Reduction - PCA - Linear discriminant analysis.

Introduction

- Supervised Learning
- Bayesian classifiers are the statistical classifiers based on Bayes' Theorem
- Bayesian classifiers can predict class membership probabilities i.e. the probability that a given tuple belongs to a particular class.
- It uses the given values to train a model and then it uses this model to classify new data

BAYESIAN DECISION THEORY

What is Bayes Theorem?

- Bayes' theorem, named after 18th-century British mathematician Thomas Bayes, is a mathematical formula for determining conditional probability .
- The theorem provides a way to revise existing predictions or theories given new or additional evidence.
- In finance, Bayes' theorem can be used to rate the risk of lending money to potential borrowers.

Where is it used?

- Trying to find the answer
- There are only two possible events possible for the given question:

  A: It is going to rain tomorrow

  B: It will not rain tomorrow.

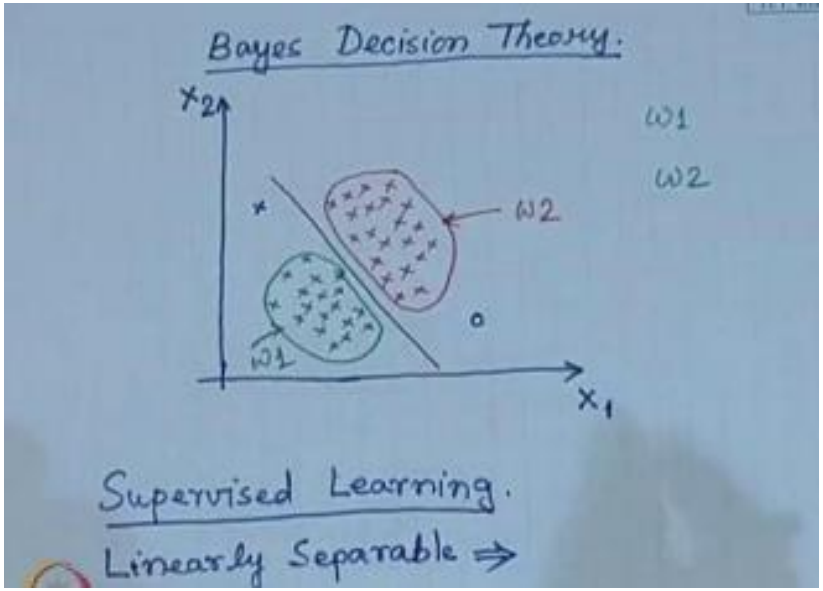
If you think intuitively

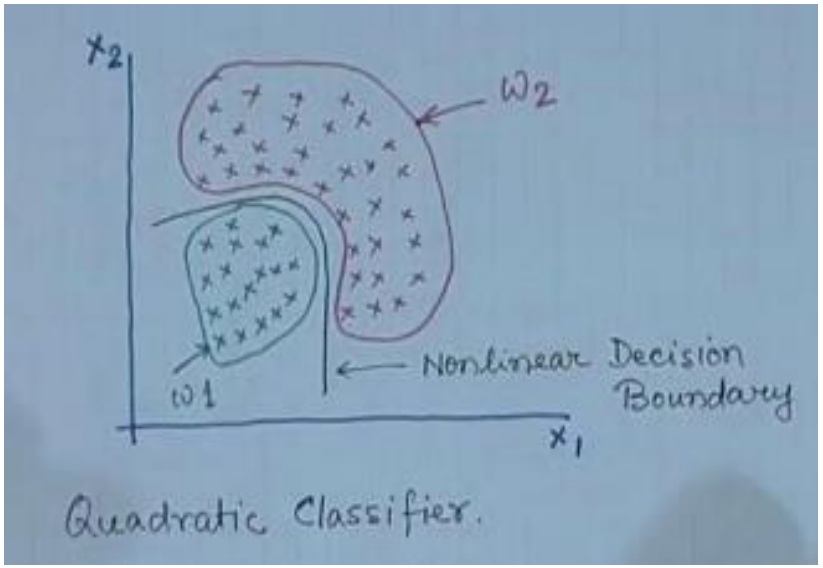
- It's either going to be raining today or it is NOt going to be raining today
- So technically there is 50% CHANCE OF RAIN tomorrow. Correct?

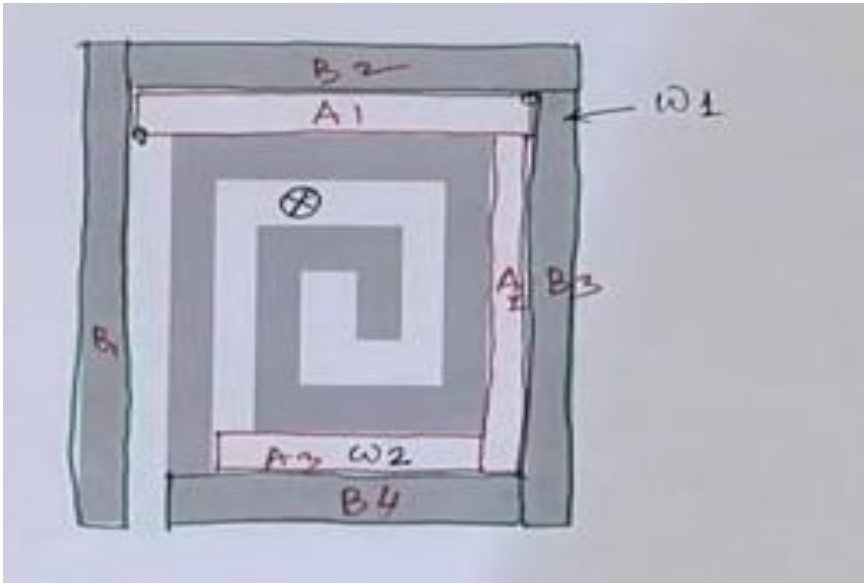
OVERVIEW

- Probability theory is the framework for making decisions under uncertainty.
- In classification, Bayes' rule is used to calculate the probabilities of the classes.
- Discuss how to make rational decisions among multiple actions to minimize expected risk.
- Classification appears in many disciplines for pattern recognition and detection methods.
- The Bayesian decision theory, which is based on the existence of prior distributions of the parameters is used.

OVERVIEW


Bayes Decision Theory.
x_2
ω_1
ω_2
ω_2
ω_1
x_1
Supervised Learning.
Linearly Separable ⇒


x_2
ω_2
Nonlinear Decision Boundary
ω_1
x_1
Quadratic Classifier.


B_2
A_1
ω_1
A_2
B_3
B_1
A_2 ω_2
B_4

INTRODUCTION

- Programming computers to make inference from data is a cross between statistics and computer science.
- Statisticians provide the mathematical framework of making inference from data and computer scientists work on the efficient implementation of the inference methods.
- Data comes from a process that is not completely known. This lack of knowledge is indicated by modeling the process as a random process.

EXAMPLE

- Tossing a coin is a random process because we cannot predict at any toss whether the outcome will be heads or tails—that is why we toss coins, or buy lottery tickets, or get insurance.

- We can only talk about the probability that the outcome of the next toss will be heads or tails.

EXAMPLE

- The extra pieces of knowledge that we do not have access to are named **unobservable variables**. *In the coin tossing example, the only **observe variables** is the outcome of the toss.*
- *Denoting the unobservable by **z** and the observable as **x**, in reality we have*

$$x = f(z)$$

- where *f (·) is the deterministic function that defines the outcome from* the unobservable pieces of knowledge.
- Because we cannot model the process this way, we define the outcome *X as a random variable drawn* from a probability distribution *P(X = x) that specifies the process.*

EXAMPLE

- The outcome of tossing a coin is heads or tails, and we define a random variable that takes one of two values.
- Let us say X = 1 denotes that the outcome of a toss is heads and X = 0 denotes tails.
- Such X are Bernoulli distributed where the parameter of the distribution p_o is the probability that the outcome is heads:

$$P(X = 1) = p_o \text{ and } P(X = 0) = 1 - P(X = 1) = 1 - p_o$$

- Assume that we are asked to predict the outcome of the next toss.
- If we know p_o, our prediction will be heads if $p_o > 0.5$ and tails otherwise.

EXAMPLE

- If P(X) is unknown and want to estimate it from a given sample, then we are in the realm of statistics.

- We have a sample, X, containing examples drawn from the probability distribution of the observables x^t , denoted as p(x).

- The aim is to build an approximator to it, ˆp(x), using the sample X.

EXAMPLE

In the coin tossing example, the sample contains the outcomes of the past N tosses. Then using X, we can estimate p_o, which is the parameter that uniquely specifies the distribution. Our estimate of p_o is

$$\hat{p}_o = \frac{\#\{\text{tosses with outcome heads}\}}{\#\{\text{tosses}\}}$$

Numerically using the random variables, x^t is 1 if the outcome of toss t is heads and 0 otherwise. Given the sample {heads, heads, heads, tails, heads, tails, tails, heads, heads}, we have $X = \{1,1,1,0,1,0,0,1,1\}$ and the estimate is

$$\hat{p}_o = \frac{\sum_{t=1}^{N} x^t}{N} = \frac{6}{9}$$

EXAMPLE

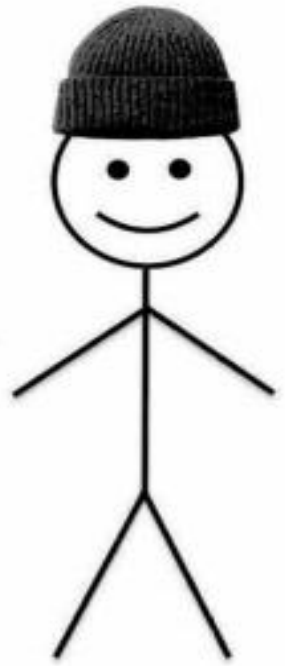
This is Bill.

Bill is 35 years old.

Bill earns 40000$/yr

Bill has a very fair credit raiting.

Will Bill buy a computer?



BAYES THEOREM

$$P(H|X) = P(X|H) * P(H) / P(X)$$

- H: Hypothesis that Bill will buy the computer
- X : Bill is 35 years old with fair credit rating and income of 40000$/year
- P(H|X) : The probability that Bill will buy the computer GIVEN that we know his age, income and credit rating [Posterior]
- P(H) : Probability that Bill will buy computer (REGARDLESS of knowing age, income and credit rating) [Prior]
- P(X|H) : Probability that someone is 35 years old, has fair credit rating, earns 40000$/yr AND has BOUGHT the computer. [Likelihood]
- P(X) : Probability that Bill is 35 years old, has fair credit rating, earns 40000$/yr [Evidence]

What are Bayesian Classifiers?

- Statistical classifiers.
- Predict class membership probabilities, such as the probability that a given tuple belongs to a particular class.
- Based on Bayes' theorem
- Exhibits high accuracy and speed when applied to large databases.

BAYES DECISION THEORY

Problem Description

Suppose we have a classification problem statement where we have to classify among the object-1 and object-2 with the given set of features $\mathbf{X} = [x_1, x_2, \ldots, x_n]^T$.

Objective

object not yet seen i.e, not in training data.

w = w1 for object-1 and
w = w2 for object-2

BAYES DECISION THEORY

- P(w1)-Prob. accepted
- P(w2)-prob rejected

- Decision rule based on above prob.
  - P(w1)>p(w2)=> w1
  - P(w1)<p(w2)=> w2
- A Prior prob. – not logical decision

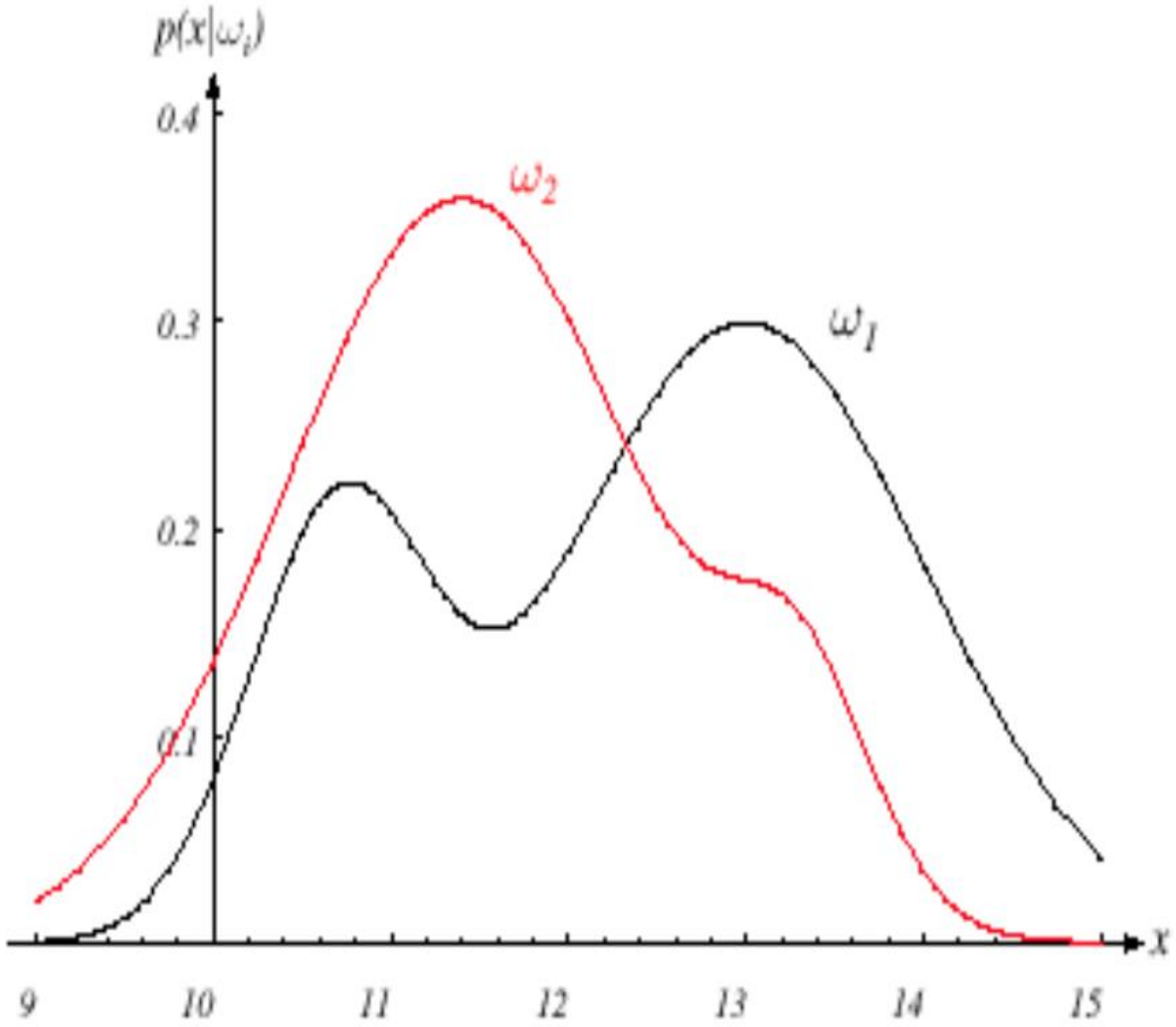
BAYES DECISION THEORY

- To make decision logical we combine some feature x to prob measure.
- Quality of polish of product, shape of the product (measurable or observation) to decide i.e., nothing but x.
- Take few samples from both class for which the class is already known and measure on x.
- Measure x for both P(x|w1) and p(x|w2) - Class Conditional Prob.density function

BAYES DECISION THEORY

- Decision problem-measure x for unknown object -to decide whether input belong to w1 or w2.
- Compute p(w1|x) or p(w2|x)- appears to be more logical.
- P(w1|x)> or < p(w2|x) decide.
- Conditional Probability – More Logical

BAYES DECISION THEORY



BAYES DECISION THEORY

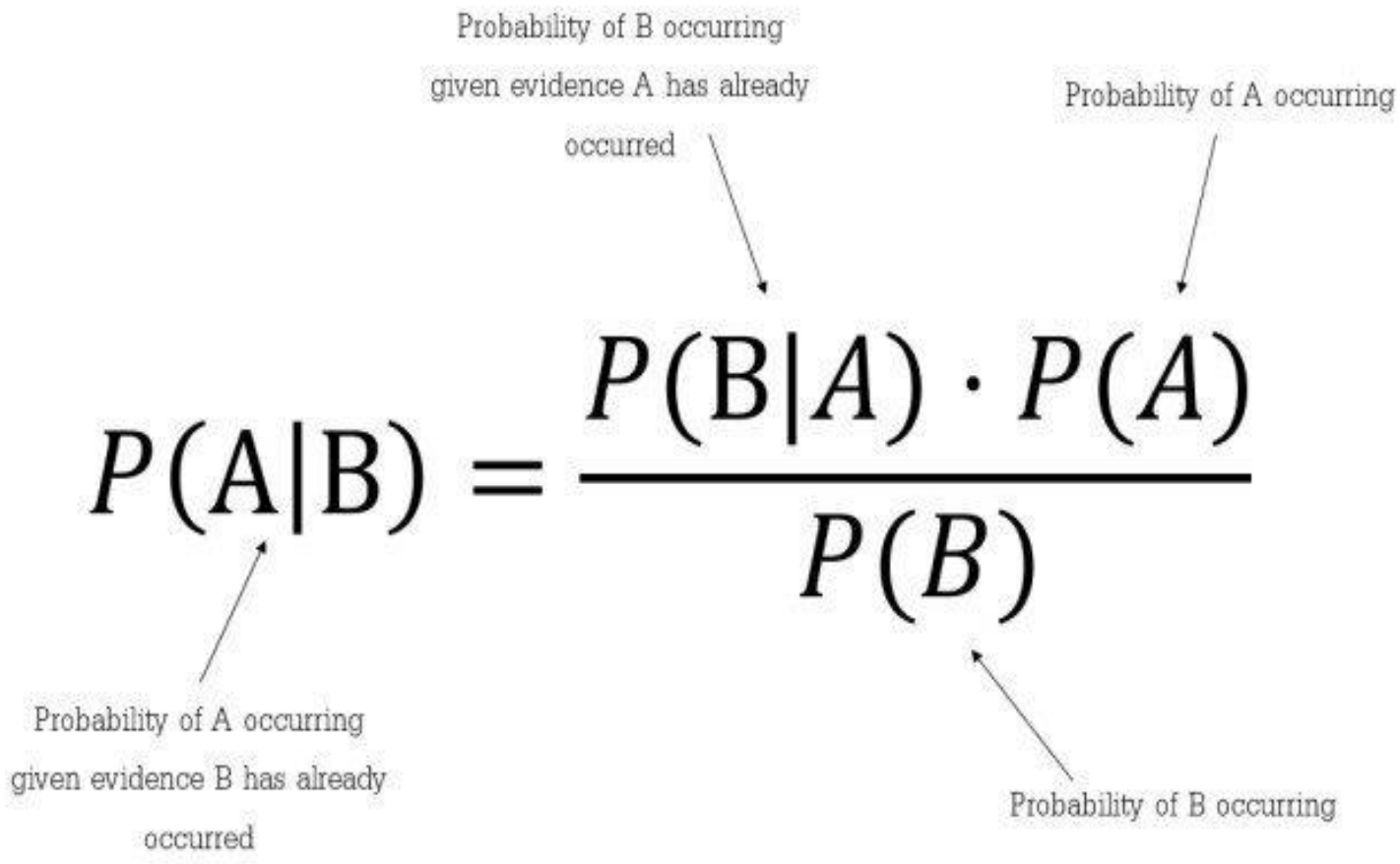
- Joint prob. Density function. Preliminary prob theory
- $P(wi,x) = P(wi|x).P(x)$

  $= P(x|wi).P(wi)$

  $P(wi|x) * P(x) = P(x|wi) * P(wi)$
- Unknown object = known object :

  $P(wi|x) = P(x|wi) * P(wi) / P(x)$

BAYES DECISION THEORY



The problem is to calculate P(C|**x). Using Bayes' rule,** it can be written as

Bayes' Classifier

$p(\boldsymbol{x}|C_i)$ is the probability of seeing $\boldsymbol{x}$ as the input when it is known to belong to class C_i. The posterior probability of class C_i can be calculated as

$$P(C_i|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_i)P(C_i)}{p(\boldsymbol{x})} = \frac{p(\boldsymbol{x}|C_i)P(C_i)}{\sum_{k=1}^{K} p(\boldsymbol{x}|C_k)P(C_k)}$$

and for minimum error, the *Bayes' classifier* chooses the class with the highest posterior probability; that is, we

$$\text{choose } C_i \text{ if } P(C_i|\boldsymbol{x}) = \max_k P(C_k|\boldsymbol{x})$$

Bayes Approach

- Bayesian theorem argues that the probability of an event taking place changes if there is information available about a related event.

  EX: This means that if you recall the previous weather conditions for the last week, and you remember that it has actually rained every single day, your answer will no longer be 50% .

- The Bayesian approach provides a way of explaining how you should change your existing beliefs in the light of new evidence.

Bayes' Classifier : Prerequisites

Random Variable

- A random variable is a function that maps a possible set of outcomes to some values like while tossing a coin and getting head H as 1 and Tail T as 0 where 0 and 1 are random variables.

Bayes' Classifier : Prerequisites

(a) Prior or State of Nature:

- Prior probabilities represent how likely is each Class is going to occur.
- Priors are known before the training process.
- The state of nature is a random variable $\mathbf{P(w_i)}$.
- If there are only two classes, then the sum of the priors is $\mathbf{P(w_1) + P(w_2)=1}$, if the classes are exhaustive.

Bayes' Classifier : Prerequisites

(b) Class Conditional Probabilities:

- It represents the probability of how likely a feature x occurs given that it belongs to the particular class. It is denoted by, **P(X|A)** where x is a particular feature
- It is the probability of how likely the feature x occurs given that it belongs to the class w_i.
- Sometimes, it is also known as the **Likelihood**.
- It is the quantity that we have to evaluate while training the data. During the training process, we have input(features) X labeled to corresponding class w and we figure out the likelihood of occurrence of that set of features given the class label.

Bayes' Classifier : Prerequisites

(c) Evidence:

- It is the probability of occurrence of a particular feature i.e. **P(X)**.
- It can be calculated using the chain rule as, $\mathbf{P(X) = \Sigma_{in}\ P(X \mid w_i)\ P(w_i)}$
- As we need the likelihood of class conditional probability is also figure out evidence values during training.

Bayes' Classifier : Prerequisites

(d) Posterior Probabilities:

- It is the probability of occurrence of Class A when certain Features are given
- It is what we aim at computing in the test phase in which we have testing input or features (the given entity) and have to find how likely trained model can predict features belonging to the particular class w_i.

Bayes Formula

- Bayes' theorem is also called Bayes' Rule or Bayes' Law

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

The problem is to calculate P(C|**x). Using Bayes' rule,** it can be written as

$$P(C|\boldsymbol{x}) = \frac{P(C)p(\boldsymbol{x}|C)}{p(\boldsymbol{x})}$$

QUIZ

In the formula for Bayes' theorem, which is the prior probability?

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

- ○ P(A)
- ○ P(B | A)
- ○ P(B)
- ○ P(A | B)

QUIZ

The estimated probabilities of an Event before any new data is collected are known as _____ probabilities

- Prior
- Posterior
- Conditional
- Simple

If $P(A) = 0.20$, $P(B|A)=0.60$ and $P(B|A')=0.25$, then $P(A|B) =$

- 0.3750
- 0.7059
- 0.6250
- 0.2941

QUIZ

Q. Review: A store sells shirts that are either small, medium, or large. Each size comes in either yellow, red, blue or green. If a person selects from these choices at random, what is the probability that they will select a large green shirt?

- 1/3
- 1/7
- 1/9
- 1/12

The updated probabilities of an Event in light of newly-collected data are known as _____ probabilities

- Prior
- Posterior
- Conditional
- Simple

QUIZ

The fire alarm rings, but there is no fire. This is a...

- True Positive
- False Positive
- True Negative
- False Negative

The fire alarm does not ring, and there is no fire. This is a...

- True Positive
- False Positive
- True Negative
- False Negative

QUIZ

A doctor is to visit a patient. From the past experience, it is known that the probabilities that he will come by train, bus, and scooter or by other means of transport are respectively 0.3, 0.2, 0.1 and 0.4. The probabilities that he will be late are 1/4 , 1/3 and 1/12, if he comes by train, bus and scooter respectively, but if he comes by other means of transport, then he will not be late. When he arrives, he is late. The probability that he comes by bus is:

- A. 1/6
- B. 4/9
- C. 5/18
- D. 1/18

QUIZ

Probability that A speaks truth is 5/9 . A coin is tossed and reports that a head appears. The probability that actually there was head is:

- **A.** 5/9
- **B.** 5/18
- **C.** 2/9
- **D.** 4/9

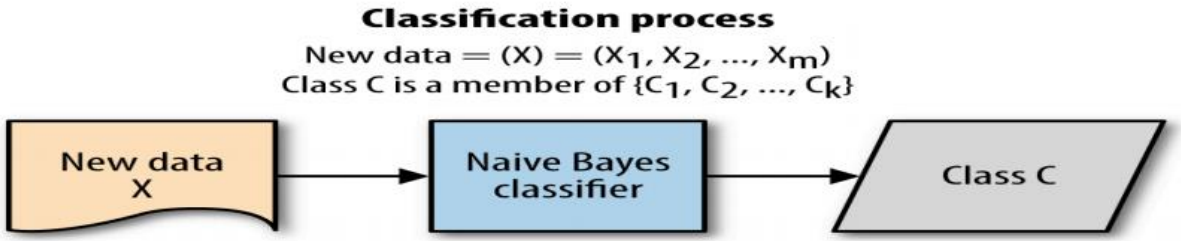
A man is known to speak truth 3 out of 4 times. He throws a die and reports that it is a six. Find the probability that it is actually a six.

- **A.** 7/12
- **B.** 1/4
- **C.** 1/2
- **D.** 3/8

NAIVE BAYES CLASSIFICATION

NAIVE BAYES CLASSIFICATION

- Before explaining the mathematical representations, let us see the basic principle of Bayesian classification :
- Predict the most probable class for each instance. How ?
- Find out the probability of the previously unseen instance belonging to each class, and then select the most probable class.



NAIVE BAYES CLASSIFICATION

- A Naive Bayes Classifier is a program which predicts a class value given a set of attributes. For each known class value,
- Calculate probabilities for each attribute, conditional on the class value.
- Use the product rule to obtain a joint conditional probability for the attributes.
- Use Bayes rule to derive conditional probabilities for the class variable. Once this has been done for all class values, output the class with the highest probability.

Example: Problem

Given database:

Class-Labeled Training Tuples from the *AllElectronics* Customer Database

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

NAIVE BAYES CLASSIFICATION

The dataset is divided into two parts, namely, **feature matrix** and the **response vector**.

- Feature matrix contains all the vectors(rows) of dataset in which each vector consists of the value of **dependent features**. In above dataset, features are ‘income’, ‘Age’, ‘Credit Rating’ and ‘Student’.
- Response vector contains the value of **class variable**(prediction or output) for each row of feature matrix. In above dataset, the class variable name is ‘Buys_Computer’.

NAIVE BAYES CLASSIFICATION

The fundamental Naive Bayes assumption is that each feature makes an:

- independent
- equal

contribution to the outcome.

With relation to our dataset, this concept can be understood as:

- First, We assume that no pair of features are dependent
- Secondly, each feature is given the same weight (or importance).

Example

Given database:

Class-Labeled Training Tuples from the *AllElectronics* Customer Database

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

$X = (age = youth, income = medium, st$

We need to maximize $P(X|C_i)P(C_i)$, for i
class, can be computed based on the training

$P(buys\_computer = yes) = 9/14 = 0.643$
$P(buys\_computer = no) = 5/14 = 0.357$

To compute $P(X|C_i)$, for $i = 1, 2$, we compute the following conditional probabilities:

$$P(age = youth \mid buys\_computer = yes) = 2/9 = 0.222$$
$$P(age = youth \mid buys\_computer = no) = 3/5 = 0.600$$
$$P(income = medium \mid buys\_computer = yes) = 4/9 = 0.444$$
$$P(income = medium \mid buys\_computer = no) = 2/5 = 0.400$$
$$P(student = yes \mid buys\_computer = yes) = 6/9 = 0.667$$

$$P(student = yes \mid buys\_computer = no) = 1/5 = 0.200$$
$$P(credit\_rating = fair \mid buys\_computer = yes) = 6/9 = 0.667$$
$$P(credit\_rating = fair \mid buys\_computer = no) = 2/5 = 0.400$$

Example

$$P(age = youth \mid buys\_computer = yes) = 2/9 = 0.222$$
$$P(age = youth \mid buys\_computer = no) = 3/5 = 0.600$$
$$P(income = medium \mid buys\_computer = yes) = 4/9 = 0.444$$
$$P(income = medium \mid buys\_computer = no) = 2/5 = 0.400$$
$$P(student = yes \mid buys\_computer = yes) = 6/9 = 0.667$$
$$P(student = yes \mid buys\_computer = no) = 1/5 = 0.200$$
$$P(credit\_rating = fair \mid buys\_computer = yes) = 6/9 = 0.667$$
$$P(credit\_rating = fair \mid buys\_computer = no) = 2/5 = 0.400$$

Using these probabilities, we obtain

$$\begin{aligned} P(X \mid buys\_computer = yes) &= P(age = youth \mid buys\_computer = yes) \\ &\times P(income = medium \mid buys\_computer = yes) \\ &\times P(student = yes \mid buys\_computer = yes) \\ &\times P(credit\_rating = fair \mid buys\_computer = yes) \\ &= 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044. \end{aligned}$$

Similarly,

$$P(X \mid buys\_computer = no) = 0.600 \times 0.400 \times 0.200 \times 0.400 = 0.019.$$

To find the class, C_i, that maximizes $P(X \mid C_i)P(C_i)$, we compute

$$P(X \mid buys\_computer = yes)P(buys\_computer = yes) = 0.044 \times 0.643 = 0.028$$
$$P(X \mid buys\_computer = no)P(buys\_computer = no) = 0.019 \times 0.357 = 0.007$$

Therefore, the naïve Bayesian classifier predicts $buys\_computer = yes$ for tuple X.

Example

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Naïve Bayes Classification :

Assumption:

The fundamental Naive Bayes assumption is that each feature makes an:

- Independent
- Equal contribution to the outcome.

Naïve Bayes Classification : Learning Phase

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Temperature	Play=*Yes*	Play=*No*
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Wind	Play=*Yes*	Play=*No*
Strong	3/9	3/5
Weak	6/9	2/5

Humidity	Play=*Yes*	Play=No
High	3/9	4/5
Normal	6/9	1/5

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

P (Play=*Yes*) = 9/14

P (Play=*No*) = 5/14

Naïve Bayes Classification : Testing Phase

Given a new instance,

–**x**'= (Outlook=*Sunny,* Temperature=*Cool,*
Humidity=*High,* Wind=*Strong*)

Look up tables

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

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

Naïve Bayes Classification : Testing Phase

MAP rule :

- P(*Yes*|**x**') = [P(*Sunny*|Yes) P(*Cool*|*Yes*) P(*High*|Yes) P(*Strong*|*Yes*)]

  * P(Play=*Yes*)

- P(*No*|**x**') = [P(*Sunny*|No) P(*Cool*|No) P(*High*|*No*) P(*Strong*|*No*)]

  * P(Play=*No*)

Output : Given the factP(*Yes*|**x**') < P(*No*|**x**'), then we label **x**' to be "*No*".

ALGORITHM: Naïve Bayes Classification

STEP 1 : Let D be a training data set of tuples and that associated class labels. Each tuple is represented by n dimensional attribute vector, where $x=\{x_1,x_2,x_3.....,x_n\}$ and their attributes are represented as $A_1,A_2.....,A_n$.

STEP 2 : Suppose there are m class labels namely $c_1,c_2....,c_m$. Given a tuple x, the classifier would predict that x belongs to a class having higher posterior probability on 'x'. Naive Bayes predicts the tuple 'x' belongs to a class c_i if and only if it satisfies the below condition.

$P(c_i/x)> P(c_j>x)$ for $i=1<=j<=m$, where $P(c_i/x)=(P(x/c_i)* P(c_i)/ P(x)$

Thus the maximize P(x/ci) there by maximizing P(ci/x)

i.e., Posterior hypothesis.

ALGORITHM: Naïve Bayes Classification

STEP 3 : P(x) is a constant for all given classes hence only P(x/ci).P(ci) needs to be maximized.

STEP 4 : Given datasets with many attributes the computational cost is more for P(x/ci) in order to reduce the computational cost. Assume that all given attributes are conditionally independent i.e., the values of attributes are independent of one another. Therefore :

$$P\left(\frac{x}{Ci}\right) = \prod_{k=1}^{n} P\left(\frac{xk}{Ci}\right)$$

where k represents number of attributes and C_i=class labels.

STEP 5 : Inorder to predict the class labels of 'x' .the P(x/ci) is evaluated for all given attributes of 'x' and then the product of P(x/ci) and P(ci) is evaluated for all the classes and then the maximum product will provide us the predicted class label.

Advantages of Naive Bayes Algorithm

- They are extremely fast for both training and prediction
- They are often very easily interpretable
- They have very few (if any) tunable parameters
- If the independence assumption holds then it works more efficiently than other algorithms.
- It requires less training data and is highly scalable.
- It can make probabilistic predictions.
- Can handle both continuous and discrete data.
- Insensitive towards irrelevant features.
- Best suited for text classification problems.

Disadvantages of Naive Bayes Algorithm

- The strong assumption about the features to be independent which is hardly true in real life applications.
- Data scarcity and Chances of loss of accuracy.
- Zero Frequency i.e. if the category of any categorical variable is not seen in training data set then model assigns a zero probability to that category and then a prediction cannot be made.

NAIVE BAYES CLASSIFICATION - APPLICATIONS

- Real time Prediction
- Multi class Prediction
- Text Classification/ Spam Filtering/ Sentiment Analysis
- Recommendation System

BAYESIAN NETWORKS

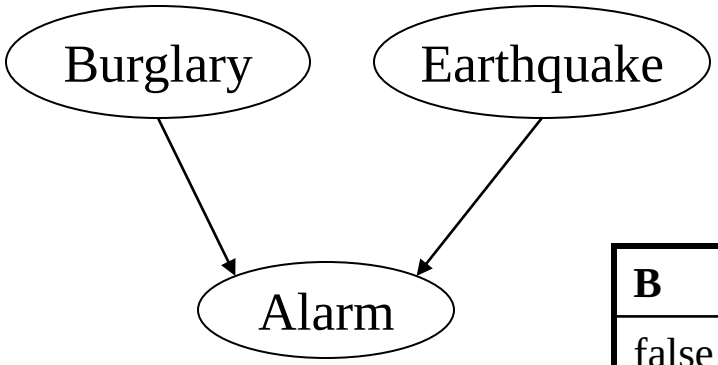
Bayesian Networks

- Bayesian networks have been the most important contribution to the field of AI in the last 10 years
- Provide a way to represent knowledge in an uncertain domain and a way to reason about this knowledge
- Many applications: medicine, factories, help desks, spam filtering, etc.

A Bayesian Network

B	P(B)
false	0.999
true	0.001

E	P(E)
false	0.998
true	0.002

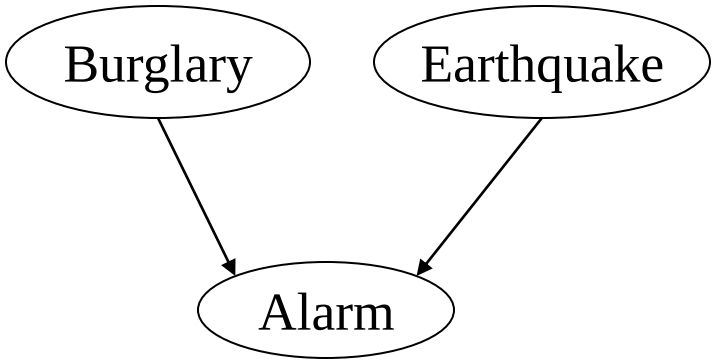


A Bayesian network is made up of two parts:

1. A directed acyclic graph
2. A set of parameters

B	E	A	P(A|B,E)
false	false	false	0.999
false	false	true	0.001
false	true	false	0.71
false	true	true	0.29
true	false	false	0.06
true	false	true	0.94
true	true	false	0.05
true	true	true	0.95

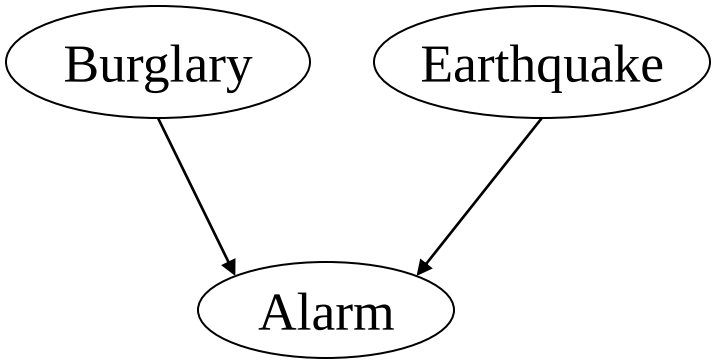
A Directed Acyclic Graph



1. A directed acyclic graph:

- The nodes are random variables (which can be discrete or continuous)
- Arrows connect pairs of nodes (X is a parent of Y if there is an arrow from node X to node Y).

A Directed Acyclic Graph



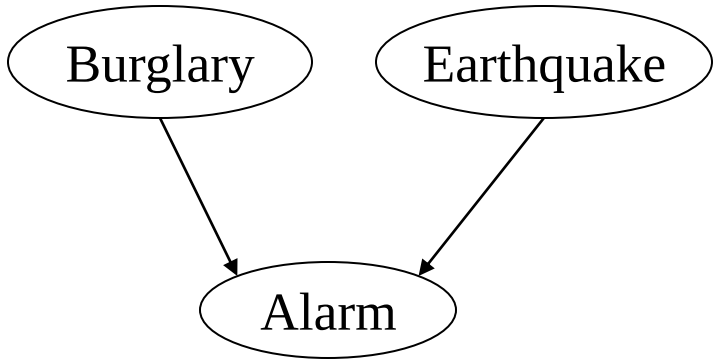
- Intuitively, an arrow from node X to node Y means X has a direct influence on Y (we can say X has a casual effect on Y)
- Easy for a domain expert to determine these relationships
- The absence/presence of arrows will be made more precise later on

A Set of Parameters

B	P(B)
false	0.999
true	0.001

E	P(E)
false	0.998
true	0.002

B	E	A	P(A|B,E)
false	false	false	0.999
false	false	true	0.001
false	true	false	0.71
false	true	true	0.29
true	false	false	0.06
true	false	true	0.94
true	true	false	0.05
true	true	true	0.95



Each node X_i has a conditional probability distribution $P(X_i \mid Parents(X_i))$ that quantifies the effect of the parents on the node

The parameters are the probabilities in these conditional probability distributions

Because we have discrete random variables, we have conditional probability tables (CPTs)

A Set of Parameters

Conditional Probability Distribution for Alarm

B	E	A	P(A|B,E)
false	false	false	0.999
false	false	true	0.001
false	true	false	0.71
false	true	true	0.29
true	false	false	0.06
true	false	true	0.94
true	true	false	0.05
true	true	true	0.95

Stores the probability distribution for Alarm given the values of Burglary and Earthquake

For a given combination of values of the parents (B and E in this example), the entries for P(A=true|B,E) and P(A=false|B,E) must add up to 1 eg. P(A=true|B=false,E=false) + P(A=false|B=false,E=false)=1

If you have a Boolean variable with k Boolean parents, how big is the conditional probability table?

How many entries are independently specifiable?

BAYESIAN BELIEF NETWORK

"A Bayesian network is a probabilistic graphical model which represents a set of variables and their conditional dependencies using a directed acyclic graph."

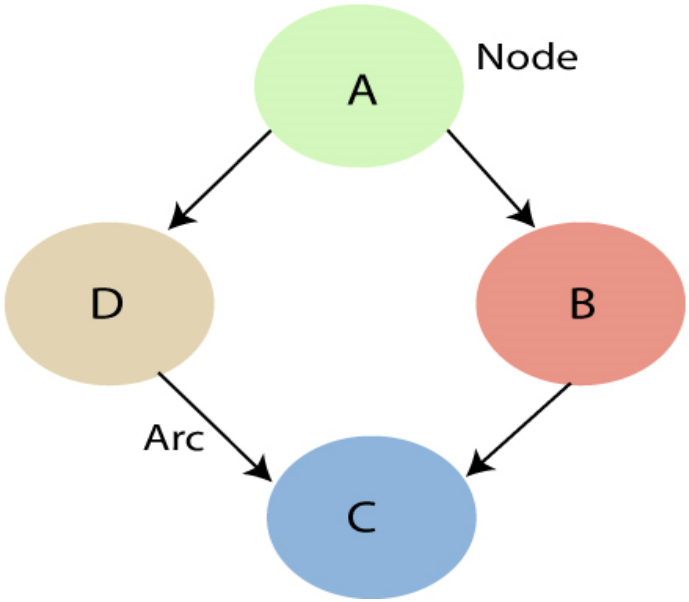
- It is also called a **Bayes network, belief network, decision network**, or **Bayesian model**.

- Bayesian networks are probabilistic, because these networks are built from a **probability distribution**, and also use probability theory for prediction and anomaly detection.

BAYESIAN BELIEF NETWORK

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.**



Conditional Probability

- Given A and B are dependent events, the conditional probability is calculated as :

$$P(A|B) = P(A \text{ and } B) / P(B)$$

- If A and B are independent events, then the expression for conditional probability is given by :

$$P(A|B) = P(A)$$

BAYESIAN NETWORKS : 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.

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

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

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

BAYESIAN NETWORKS : JOINT PROBABILITY DISTRIBUTION

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

List of all events occurring in this network:

- Burglary (B)
- Earthquake(E)
- Alarm(A)
- David Calls(D)
- John calls(J)

BAYESIAN NETWORKS : JOINT PROBABILITY DISTRIBUTION

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

$$P[D, J, A, B, E] = P[D \mid J, A, B, E].\ P[J, A, B, E]$$

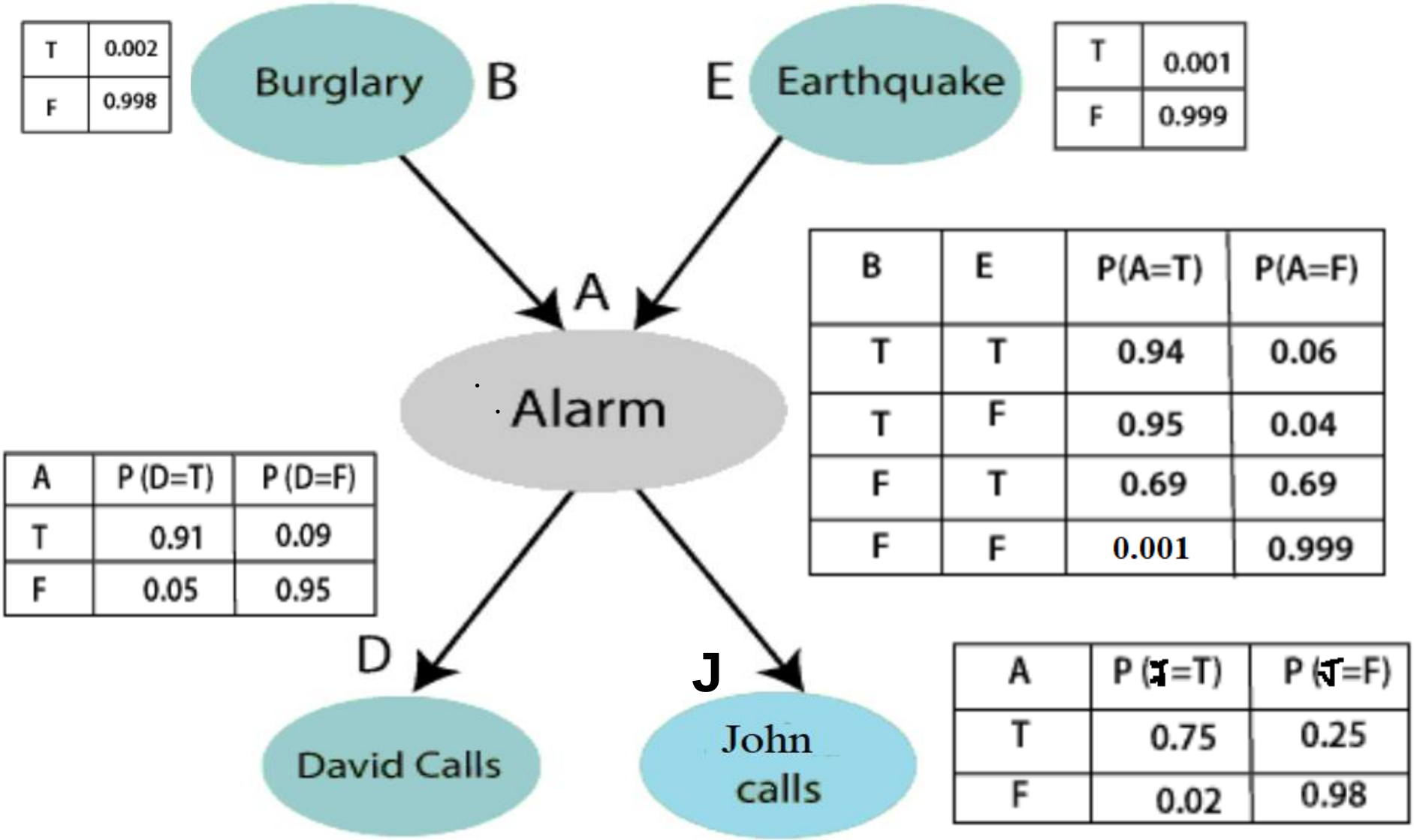
$$= P[D \mid J, A, B, E].\ P[J \mid A, B, E].\ P[A, B, E]$$

$$= P[D \mid A].\ P[J \mid A, B, E].\ P[A, B, E]$$

$$= P[D \mid A].\ P[J \mid A].\ P[A \mid B, E].\ P[B, E]$$

$$= P[D \mid A].\ P[J \mid A].\ P[A \mid B, E].\ P[B \mid E].\ P[E]$$

BAYESIAN NETWORKS :
JOINT PROBABILITY DISTRIBUTION



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	0.001	0.999

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

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

BAYESIAN NETWORKS :
JOINT PROBABILITY DISTRIBUTION

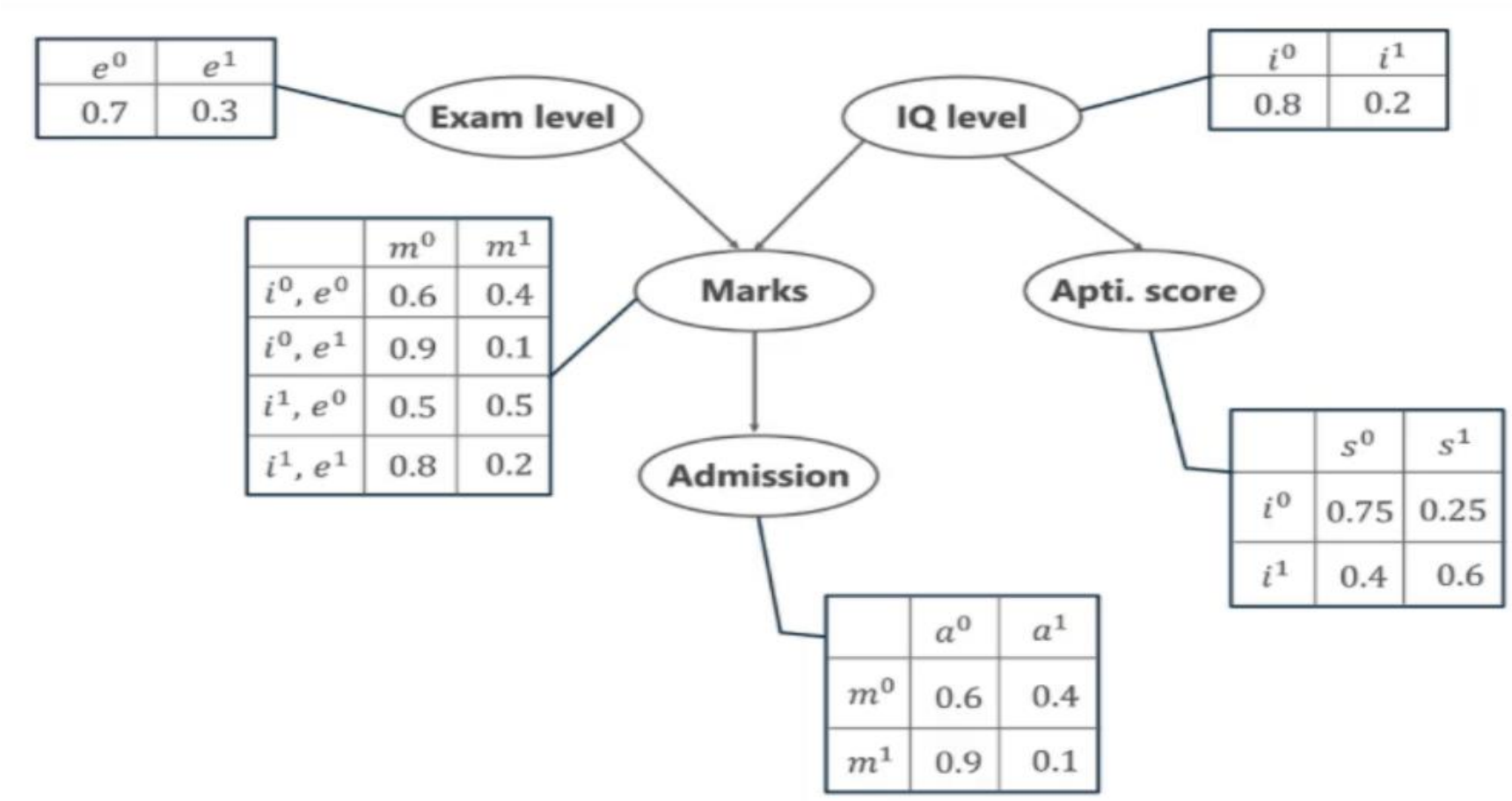
From the formula of joint distribution, we can write the problem statement in the form of probability distribution:

P(J, D, A, ¬B, ¬E) = P (J|A) *P (D|A)*P (A|¬B ^ ¬E) *P (¬B) *P (¬E).

= 0.75* 0.91* 0.001* 0.998*0.999

= 0.00068045.

BAYESIAN NETWORKS :
JOINT PROBABILITY DISTRIBUTION



e^0	e^1
0.7	0.3

	i^0	i^1
	0.8	0.2

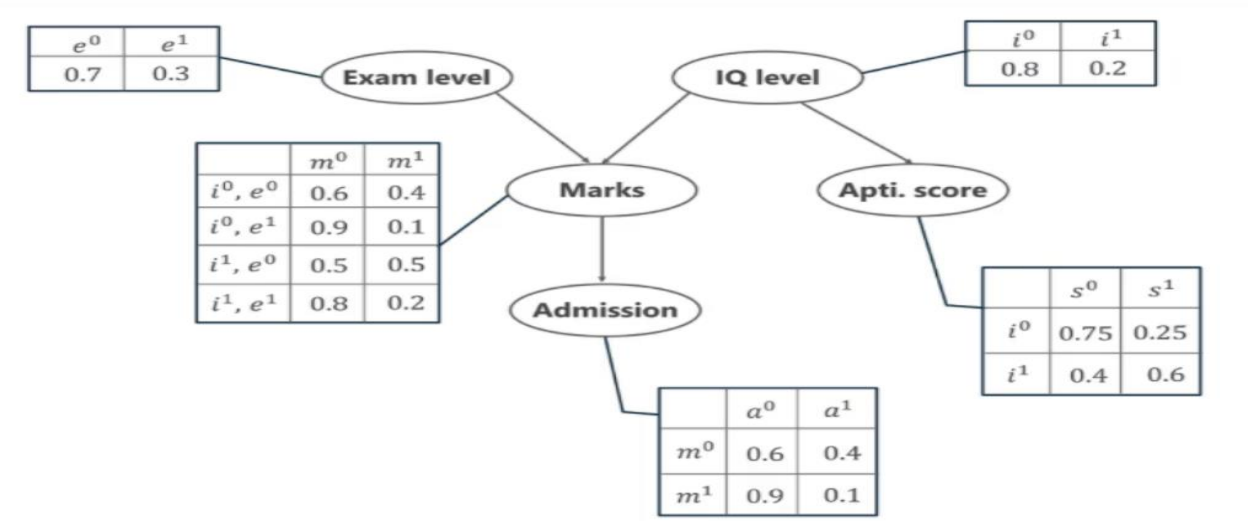
	m^0	m^1
i^0, e^0	0.6	0.4
i^0, e^1	0.9	0.1
i^1, e^0	0.5	0.5
i^1, e^1	0.8	0.2

	s^0	s^1
i^0	0.75	0.25
i^1	0.4	0.6

	a^0	a^1
m^0	0.6	0.4
m^1	0.9	0.1

P[A, M, I, E, S] = P(A | M) . P(M | I, E) . P(I) . P(E) . P(S | I)

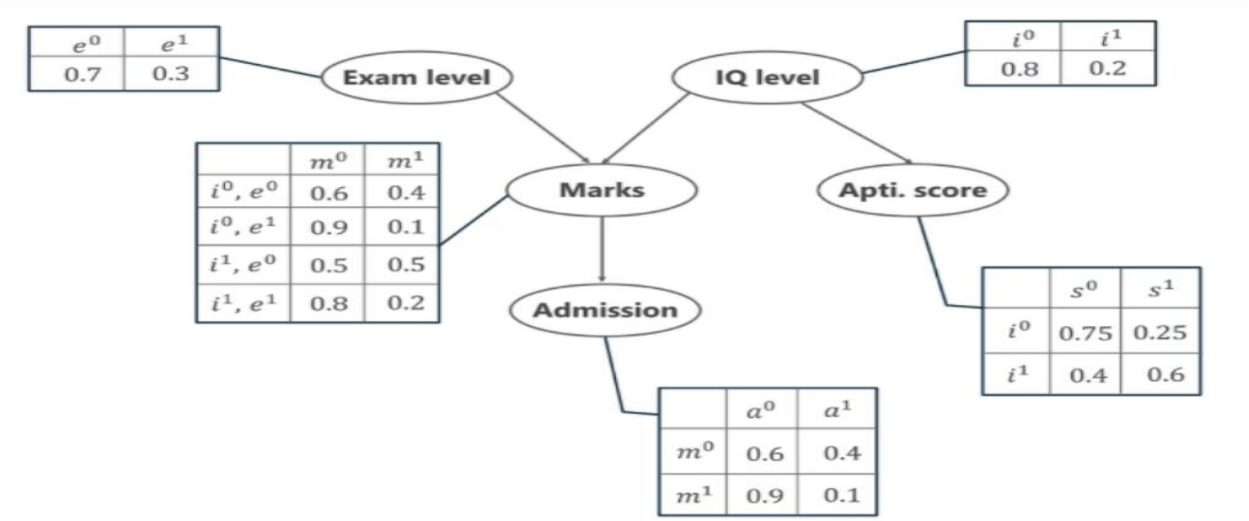
BAYESIAN NETWORKS :
JOINT PROBABILITY DISTRIBUTION



Case 1: Calculate the probability that in spite of the exam level being difficult, the student having a low IQ level and a low Aptitude Score, manages to pass the exam and secure admission.

P[A=1, M=1, I=0, E=1, S=0]

BAYESIAN NETWORKS :
JOINT PROBABILITY DISTRIBUTION

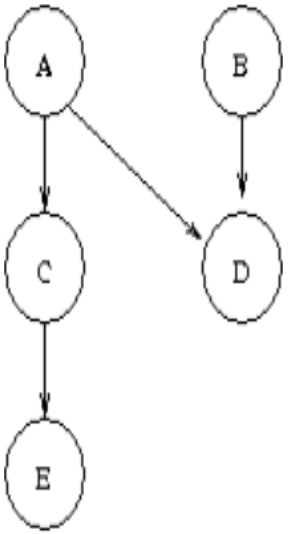


Case 2: In another case, calculate the probability that the student has a High IQ level and Aptitude Score, the exam being easy yet fails to pass and does not secure admission.

$$P[A=0, M=0, I=1, E=0, S=1]$$

BAYESIAN NETWORKS :
JOINT PROBABILITY DISTRIBUTION

Consider the following Bayesian network:



Thus, the independence expressed in this Bayesian net are that
A and B are (absolutely) independent.
C is independent of B given A.
D is independent of C given A and B.
E is independent of A, B, and D given C.

Suppose that the net further records the following probabilities:

Prob(A=T) = 0.3
Prob(B=T) = 0.6
Prob(C=T|A=T) = 0.8
Prob(C=T|A=F) = 0.4
Prob(D=T|A=T,B=T) = 0.7
Prob(D=T|A=T,B=F) = 0.8
Prob(D=T|A=F,B=T) = 0.1
Prob(D=T|A=F,B=F) = 0.2
Prob(E=T|C=T) = 0.7
Prob(E=T|C=F) = 0.2

Prob(D=T):

P(D=T) =

P(D=T,A=T,B=T) + P(D=T,A=T,B=F) + P(D=T,A=F,B=T) + P(D=T,A=F,B=F)

BAYESIAN NETWORKS :
JOINT PROBABILITY DISTRIBUTION

Prob(D=T):

P(D=T) =

P(D=T,A=T,B=T) + P(D=T,A=T,B=F) + P(D=T,A=F,B=T) + P(D=T,A=F,B=F)

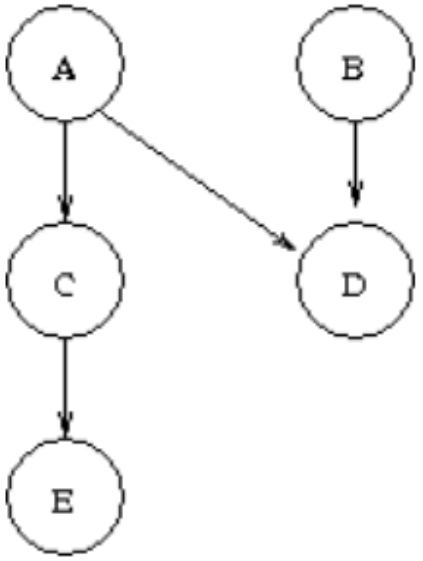
$$P(D=T|A=T,B=T)\,P(A=T,B=T) + P(D=T|A=T,B=F)\,P(A=T,B=F) +$$
$$P(D=T|A=F,B=T)\,P(A=F,B=T) + P(D=T|A=F,B=F)\,P(A=F,B=F) =$$

(since A and B are independent absolutely)

$$P(D=T|A=T,B=T)\,P(A=T)\,P(B=T) + P(D=T|A=T,B=F)\,P(A=T)\,P(B=F) +$$
$$P(D=T|A=F,B=T)\,P(A=F)\,P(B=T) + P(D=T|A=F,B=F)\,P(A=F)\,P(B=F) =$$

= 0.7*0.3*0.6 + 0.8*0.3*0.4 + 0.1*0.7*0.6 + 0.2*0.7*0.4

= 0.32



Suppose that the net further records the following probabilities:

Prob(A=T) = 0.3
Prob(B=T) = 0.6
Prob(C=T|A=T) = 0.8
Prob(C=T|A=F) = 0.4
Prob(D=T|A=T,B=T) = 0.7
Prob(D=T|A=T,B=F) = 0.8
Prob(D=T|A=F,B=T) = 0.1
Prob(D=T|A=F,B=F) = 0.2
Prob(E=T|C=T) = 0.7
Prob(E=T|C=F) = 0.2

BAYESIAN NETWORKS :
JOINT PROBABILITY DISTRIBUTION

Prob(A=T|C=T):

$$P(A=T|C=T) = P(C=T|A=T)P(A=T) / P(C=T)$$

$$P(C=T) = P(C=T,A=T) + P(C=T,A=F) =$$

$$P(C=T|A=T)P(A=T) + P(C=T|A=F)P(A=F) =$$

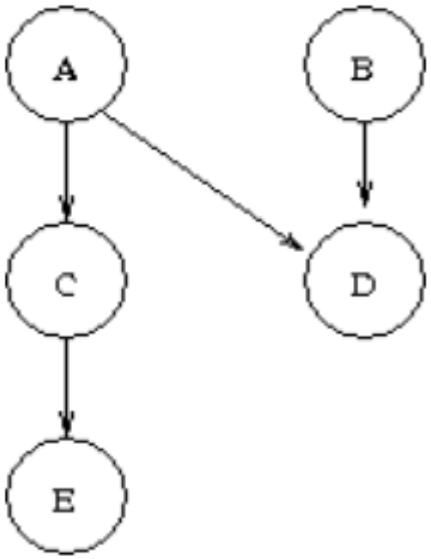
$$= 0.8* 0.3 + 0.4* 0.7$$

$$= 0.52$$

$$P(C=T|A=T)P(A=T) / P(C=T) =$$

$$= 0.8* 0.3 / 0.52$$

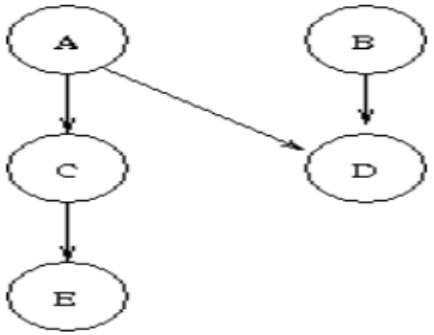
$$= 0.46$$



Suppose that the net further records the following probabilities:

Prob(A=T) = 0.3
Prob(B=T) = 0.6
Prob(C=T|A=T) = 0.8
Prob(C=T|A=F) = 0.4
Prob(D=T|A=T,B=T) = 0.7
Prob(D=T|A=T,B=F) = 0.8
Prob(D=T|A=F,B=T) = 0.1
Prob(D=T|A=F,B=F) = 0.2
Prob(E=T|C=T) = 0.7
Prob(E=T|C=F) = 0.2

BAYESIAN NETWORKS :



Prob(D=F,C=T)

P(D=F,C=T) = P(D=F,C=T,A=T,B=T) + P(D=F,C=T,A=T,B=F) +
P(D=F,C=T,A=F,B=T) + P(D=F,C=T,A=F,B=F)
=
P(D=F,C=T|A=T,B=T) P(A=T,B=T) + P(D=F,C=T|A=T,B=F) P(A=T,B=F) +
P(D=F,C=T|A=F,B=T) P(A=F,B=T) + P(D=F,C=T|A=F,B=F) P(A=F,B=F) =
(since C and D are independent given A and B)

P(D=F|A=T,B=T) P(C=T|A=T,B=T) P(A=T,B=T) + P(D=F|A=T,B=F) P(C=T|A=T,B=F)
P(A=T,B=F) +
P(D=F|A=F,B=T) P(C=T|A=F,B=T) P(A=F,B=T) + P(D=F|A=F,B=F) P(C=T|A=F,B=F)
P(A=F,B=F) =
(since C is independent of B given A and A and B are independent absolutely)

P(D=F|A=T,B=T) P(C=T|A=T) P(A=T) P(B=T) + P(D=F|A=T,B=F) P(C=T|A=T) P(A=T)
P(B=F) + P(D=F|A=F,B=T) P(C=T|A=F) P(A=F) P(B=T) + P(D=F|A=F,B=F) P(C=T|A=F)
P(A=F) P(B=F)

= 0.3*0.8*0.3*0.6 + 0.2*0.8*0.3*0.4 + 0.9*0.4*0.7*0.6 + 0.8*0.4*0.7*0.4
= 0.3032

Given two random variables x and y,

$P(X=0) = 0.2$

$P(X=1) = 0.8$

$P(Y=0) = 0.6$

$P(Y=1) = 0.4$

Conditional

$P(X/Y)$

	X=0	X=1
Y=0	0.25	0.75
Y=1	0.45	0.55

from Baye's Theorem,

$$P(C_i/x) = \frac{P(x/C_i)P(C_i)}{P(x)} = \frac{P(X=0/Y=1)P(Y=1)}{P(X=0)}$$

$$= \frac{(0.45)(0.4)}{0.2}$$

$$= 0.9$$

Algorithm :

LOSSES AND RISKS

Bayes' Classifier

Bayes Rule :

$$P(C_i|\mathbf{x}) = \frac{p(\mathbf{x}|C_i)P(C_i)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|C_i)P(C_i)}{\sum_{k=1}^{K} p(\mathbf{x}|C_k)P(C_k)}$$

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

For Minimum Error, Bayes Classifier :

$$P(C_i|\mathbf{x}) = \max_k P(C_k|\mathbf{x})$$

Losses and Risks

Example: A financial institution when making a decision for a loan applicant should take into account the potential gain and loss as well.

- An accepted low-risk applicant increases profit, while a rejected high-risk applicant decreases loss.
- The loss for a high-risk applicant erroneously accepted may be different from the potential gain for an erroneously rejected low-risk applicant.

Losses and Risks

- Actions: α_i –as the decision to assign the input to class C_i
- Loss of α_i : λ_{ik} –loss incurred for taking action α_i when the input actually belongs to C_k
- Expected risk for taking action α_i is :

$$R(\alpha_i|\boldsymbol{x}) = \sum_{k=1}^{K} \lambda_{ik} P(C_k|\boldsymbol{x})$$

$$\text{choose } \alpha_i \text{ if } R(\alpha_i|\boldsymbol{x}) = \min_k R(\alpha_k|\boldsymbol{x})$$

Losses and Risks: 0/1 Loss

Let us define K actions α_i ,i=1,2.....,K, where α_i is the action of assigning x to C_i .In the special case of the 0/1 loss case where,

$$\lambda_{ik} = \begin{cases} 0 \text{ if } i = k \\ 1 \text{ if } i \neq k \end{cases}$$

all correct decisions have no loss and all errors are equally costly. The risk of taking action α_i *is:*

$$\begin{aligned} R(\alpha_i|\boldsymbol{x}) &= \sum_{k=1}^{K} \lambda_{ik} P(C_k|\boldsymbol{x}) \\ &= \sum_{k \neq i} P(C_k|\boldsymbol{x}) \\ &= 1 - P(C_i|\boldsymbol{x}) \end{aligned}$$

Hence for minimum risk, choose the most probable class

Losses and Risks: Reject

- In some applications, wrong decisions—namely, misclassifications—may have very high cost;
  - In few cases Manual Decision is made if the automatic system has low certainty of its decision.
- In such a case, we define an additional action of reject or doubt, α_{K+1}, with α_i, i = 1,...,K, being the usual actions of deciding on classes Ci, i = 1,...,K.

Losses and Risks: Reject

- A possible loss function is

$$\lambda_{ik} = \begin{cases} 0 & \text{if } i = k \\ \lambda & \text{if } i = K+1, \quad 0 < \lambda < 1 \\ 1 & \text{otherwise} \end{cases}$$

where $0 < \lambda < 1$ is the loss incurred for choosing the $(K+1)^{st}$ action of reject. Th

$$R(\alpha_{K+1}|\boldsymbol{x}) = \sum_{k=1}^{K} \lambda P(C_k|\boldsymbol{x}) = \lambda$$

and the risk of choosing class C_i is

$$R(\alpha_i|\boldsymbol{x}) = \sum_{k \neq i} P(C_k|\boldsymbol{x}) = 1 - P(C_i|\boldsymbol{x})$$

Losses and Risks: Reject

- The optimal decision rule is to

$$\text{choose } C_i \quad \text{if } R(\alpha_i|\boldsymbol{x}) < R(\alpha_k|\boldsymbol{x}) \text{ for all } k \neq i \text{ and } R(\alpha_i|\boldsymbol{x}) < R(\alpha_{K+1}|\boldsymbol{x})$$

and

$$\text{reject} \quad \text{if } R(\alpha_{K+1}|\boldsymbol{x}) < R(\alpha_i|\boldsymbol{x}), i = 1, \ldots, K$$

- This could be rewritten as,

$$\text{choose } C_i \quad \text{if } P(C_i|\boldsymbol{x}) > P(C_k|\boldsymbol{x}) \text{ for all } k \neq i \text{ and } P(C_i|\boldsymbol{x}) > 1 - \lambda$$

and

$$\text{reject} \quad \text{otherwise}$$

Discriminant Functions

Classification can be seen as implementing a set of discriminant functions, $g_i(\mathbf{x}), i = 1,\ldots,K$,such that we,

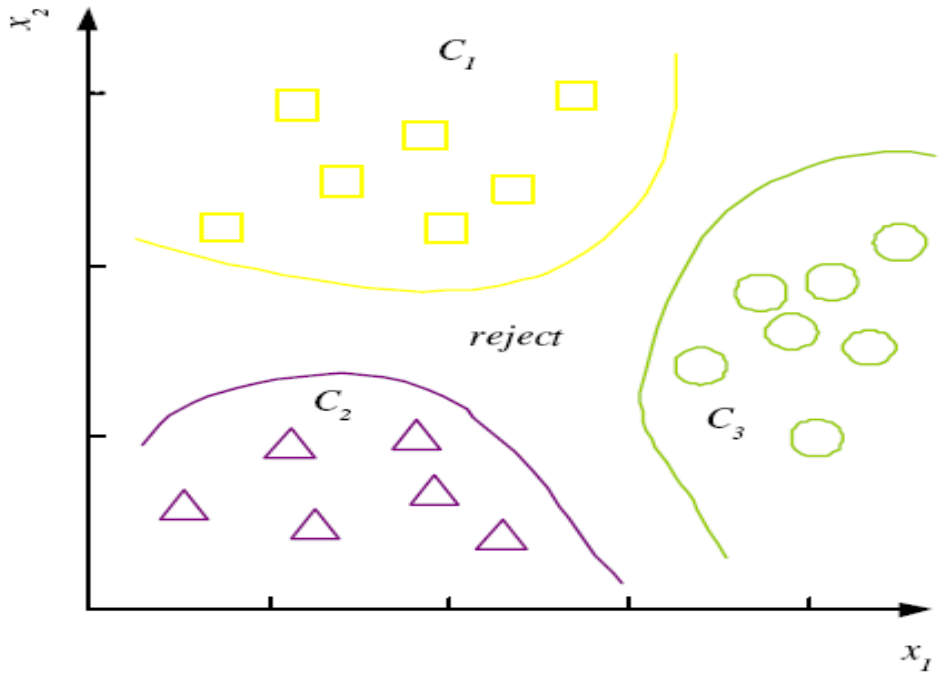
$$\text{choose } C_i \text{ if } g_i(\boldsymbol{x}) = \max_k g_k(\boldsymbol{x})$$

$$g_i(\mathbf{x}) = \begin{cases} -R(\alpha_i \mid \mathbf{x}) & \text{Max .corresponds to min. conditional risk} \\ P(C_i \mid \mathbf{x}) & \text{Using 0/1 loss function} \\ p(\mathbf{x} \mid C_i)P(C_i) & \text{Ignoring p(x)} \end{cases}$$

This divides the feature space into *K* decision regions $\mathcal{R}_1,\ldots,\mathcal{R}_K$, where

$$\mathcal{R}_i = \{\boldsymbol{x} \mid g_i(\boldsymbol{x}) = \max_k g_k(\boldsymbol{x})\}$$

Discriminant Functions



Example of decision regions and decision boundaries.

$$g(\boldsymbol{x}) = g_1(\boldsymbol{x}) - g_2(\boldsymbol{x})$$

$$\text{choose} \begin{cases} C_1 & \text{if } g(\boldsymbol{x}) > 0 \\ C_2 & \text{otherwise} \end{cases}$$

Discriminant Functions

$$g(\boldsymbol{x}) = g_1(\boldsymbol{x}) - g_2(\boldsymbol{x}) \quad \text{choose} \begin{cases} C_1 & \text{if } g(\boldsymbol{x}) > 0 \\ C_2 & \text{otherwise} \end{cases}$$

- An example is a two-class learning problem where the positive examples can be taken as C1 and the negative examples as C2.
- When *K = 2,* the classification system is a ***DICHOTOMIZER*** *and for K ≥ 3, it is a* ***POLYCHOTOMIZER****.*

PARAMETRIC METHODS

Parametric Methods

- A statistic is any value that is calculated from a given sample.
- In statistical inference, we make a decision using the information provided by a sample.
- Our first approach is parametric where we assume that the sample is drawn from some distribution that obeys a known model, for example, Gaussian.
- The advantage of the parametric approach is that the model is defined up to a small number of parameters—for example, mean, variance—the *sufficient statistics of the distribution.*

Parametric Methods

- *Once those parameters* are estimated from the sample, the whole distribution is known.
- We estimate the parameters of the distribution from the given sample, plug in these estimates to the assumed model, and get an estimated distribution, which we then use to make a decision.
- The method we use to estimate the parameters of a distribution is maximum likelihood estimation.

Parametric Estimation

- **Density estimation**, which is the general case of estimating p(x). We use this **for classification** where the estimated densities are the class densities, p(x|Ci), and priors, P(Ci), to be able to calculate the posteriors, P(Ci|x), and make our decision.
- **For Regression** where the estimated density is p(y|x). Here x is one-dimensional and thus the densities are univariate.

Parametric Estimation

- Let us say we have an independent and identically distributed (iid) sample

$$X = \{x^t\}_{t=1}^N$$

- Assume that x^t are instances drawn from some known probability density family, $p(x|\theta)$, defined up to parameters, θ: $x^t \sim p(x|\theta)$
- We want to find θ that makes sampling x^t from $p(x|\theta)$ as likely as possible.

Maximum Likelihood Estimation

- As x^t are independent, the likelihood of parameter θ given sample X is the product of the likelihoods of the individual points

- **Likelihood** of θ given the sample X

$$l(\theta|\mathcal{X}) \equiv p(\mathcal{X}|\theta) = \prod_{t=1}^{N} p(x^t|\theta)$$

Maximum Likelihood Estimation

- In **maximum likelihood estimation**, finding θ that makes X the most likely to be drawn.
- We thus search for θ that maximizes the likelihood, which we denote by **l(θ|X).** We can maximize the log of the likelihood without changing the value where it takes its maximum.
- log(·) converts the product into a sum and leads to further computational simplification when certain densities are assumed, for example, containing exponents. The **log likelihood** is defined as

$$\mathcal{L}(\theta|\mathcal{X}) \equiv \log l(\theta|\mathcal{X}) = \sum_{t=1}^{N} \log p(x^t|\theta)$$

Maximum Likelihood Estimation

- Maximum likelihood estimator (MLE)

$$\theta^* = \text{argmax}_\theta L(\theta|X)$$

- If we have a two-class problem, the distribution we use is **Bernoulli.**

- When there are K > 2 classes, its generalization is the **Multinomial.**

- **Gaussian (normal) density** is the one most frequently used for modeling class-conditional input densities with numeric input.

For these three distributions,

we discuss the maximum likelihood estimators (MLE) of their parameters.

BERNOULLI DENSITY

- The event occurs and the Bernoulli random variable *X takes* the value 1 with probability *p, and the nonoccurrence of the event has* probability 1 – *p and this is denoted by X taking the value 0. This is* written as

$$P(x) = p^x(1-p)^{1-x}, x \in \{0,1\}$$

- The expected value and variance can be calculated as

$$E[X] = \sum_x xp(x) = 1 \cdot p + 0 \cdot (1-p) = p$$

$$\mathrm{Var}(X) = \sum_x (x - E[X])^2 p(x) = p(1-p)$$

BERNOULLI DENSITY

p is the only parameter and given an iid sample $\mathcal{X} = \{x^t\}_{t=1}^N$, where $x^t \in \{0, 1\}$, we want to calculate its estimator, $\hat{p}$. The log likelihood is

$$\begin{aligned}\mathcal{L}(p|\mathcal{X}) &= \log \prod_{t=1}^{N} p^{(x^t)}(1-p)^{(1-x^t)} \\ &= \sum_t x^t \log p + \left(N - \sum_t x^t\right) \log(1-p)\end{aligned}$$

$\hat{p}$ that maximizes the log likelihood can be found by solving for $d\mathcal{L}/dp = 0$. The hat (circumflex) denotes that it is an estimate.

$$\hat{p} = \frac{\sum_t x^t}{N}$$

MULTINOMIAL DENSITY

Consider the generalization of Bernoulli where instead of two states, the outcome of a random event is one of K mutually exclusive and exhaustive states, for example, classes, each of which has a probability of occurring p_i with $\sum_{i=1}^{K} p_i = 1$. Let $x_1, x_2, \ldots, x_K$ are the indicator variables where x_i is 1 if the outcome is state i and 0 otherwise.

$$P(x_1, x_2, \ldots, x_K) = \prod_{i=1}^{K} p_i^{x_i}$$

Let us say we do N such independent experiments with outcomes $\mathcal{X} = \{\boldsymbol{x}^t\}_{t=1}^{N}$ where

$$x_i^t = \begin{cases} 1 & \text{if experiment } t \text{ chooses state } i \\ 0 & \text{otherwise} \end{cases}$$

with $\sum_i x_i^t = 1$. The MLE of p_i is

$$\hat{p}_i = \frac{\sum_t x_i^t}{N}$$

GAUSSIAN (NORMAL) DISTRIBUTION

X is Gaussian (normal) distributed with mean $E[X] \equiv \mu$ and variance $\text{Var}(X) \equiv \sigma^2$, denoted as $\mathcal{N}(\mu, \sigma^2)$, if its density function is

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right], -\infty < x < \infty$$

Given a sample $\mathcal{X} = \{x^t\}_{t=1}^N$ with $x^t \sim \mathcal{N}(\mu, \sigma^2)$, the log likelihood is

$$\mathcal{L}(\mu, \sigma | \mathcal{X}) = -\frac{N}{2}\log(2\pi) - N\log\sigma - \frac{\sum_t (x^t - \mu)^2}{2\sigma^2}$$

The MLE that we find by taking the partial derivatives of the log likelihood and setting them equal to 0 are

$$m = \frac{\sum_t x^t}{N}$$

$$s^2 = \frac{\sum_t (x^t - m)^2}{N}$$

MULTIVARIATE METHODS

Linear Discriminant Analysis(LDA)

- Logistic regression is a classification algorithm traditionally limited to only two-class classification problems.

- If you have more than two classes then Linear Discriminant Analysis is the preferred linear classification technique.

Limitations of Logistic Regression

- Logistic regression is a simple and powerful linear classification algorithm.
- **Two-Class Problems**. Logistic regression is intended for two-class or binary classification problems. It can be extended for multi-class classification, but is rarely used for this purpose.
- **Unstable With Well Separated Classes**. Logistic regression can become unstable when the classes are well separated.
- **Unstable With Few Examples**. Logistic regression can become unstable when there are few examples from which to estimate the parameters.

LDA

- Linear Discriminant Analysis does address each of these points and is the go-to linear method for multi-class classification problems.
- Even with binary-classification problems, it is a good idea to try both logistic regression and linear discriminant analysis.
- Linear Discriminant Analysis or LDA is a dimensionality reduction technique.
- It is used as a pre-processing step in Machine Learning and applications of pattern classification.
- The goal of LDA is to project the features in higher dimensional space onto a lower-dimensional space in order to avoid the curse of dimensionality and also reduce resources and dimensional costs.

LDA

- The original technique was developed in the year 1936 by Ronald A. Fisher and was named Linear Discriminant or Fisher's Discriminant Analysis.

- The original Linear Discriminant was described as a two-class technique. The multi-class version was later generalized by C.R Rao as Multiple Discriminant Analysis. They are all simply referred to as the Linear Discriminant Analysis.

- LDA is a supervised classification technique that is considered a part of crafting competitive machine learning models. This category of dimensionality reduction is used in areas like image recognition and predictive analysis in marketing.

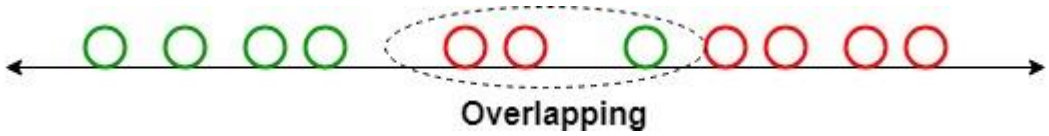
What is Dimensionality Reduction?

- The techniques of dimensionality reduction are important in applications of Machine Learning, Data Mining, Bioinformatics, and Information Retrieval.
- The main agenda is to remove the redundant and dependent features by changing the dataset onto a lower-dimensional space.
- In simple terms, they reduce the dimensions (i.e. variables) in a particular dataset while retaining most of the data.
- Multi-dimensional data comprises multiple features having a correlation with one another. You can plot multi-dimensional data in just 2 or 3 dimensions with dimensionality reduction.
- It allows the data to be presented in an explicit manner which can be easily understood by a layman.

Representation of LDA Models

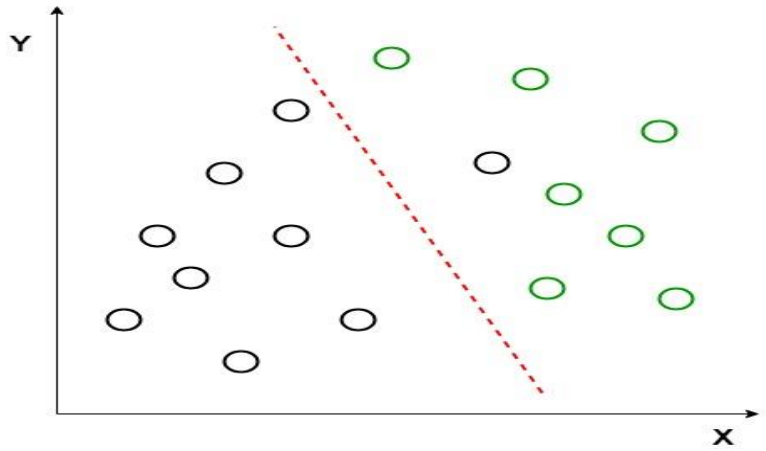
- The representation of LDA is straight forward.
- It consists of statistical properties of your data, calculated for each class.
- For a single input variable (x) this is the mean and the variance of the variable for each class.
- For multiple variables, this is the same properties calculated over the multivariate Gaussian, namely the means and the covariance matrix.
- These statistical properties are estimated from the data and plug into the LDA equation to make predictions.

- **Linear Discriminant Analysis** or **Normal Discriminant Analysis** or **Discriminant Function Analysis** is a dimensionality reduction technique which is commonly used for the supervised classification problems.
- It is used for modeling differences in groups i.e. separating two or more classes. It is used to project the features in higher dimension space into a lower dimension space.
- For example, we have two classes and we need to separate them efficiently. Classes can have multiple features. Using only a single feature to classify them may result in some overlapping as shown in the below figure. So, we will keep on increasing the number of features for proper classification

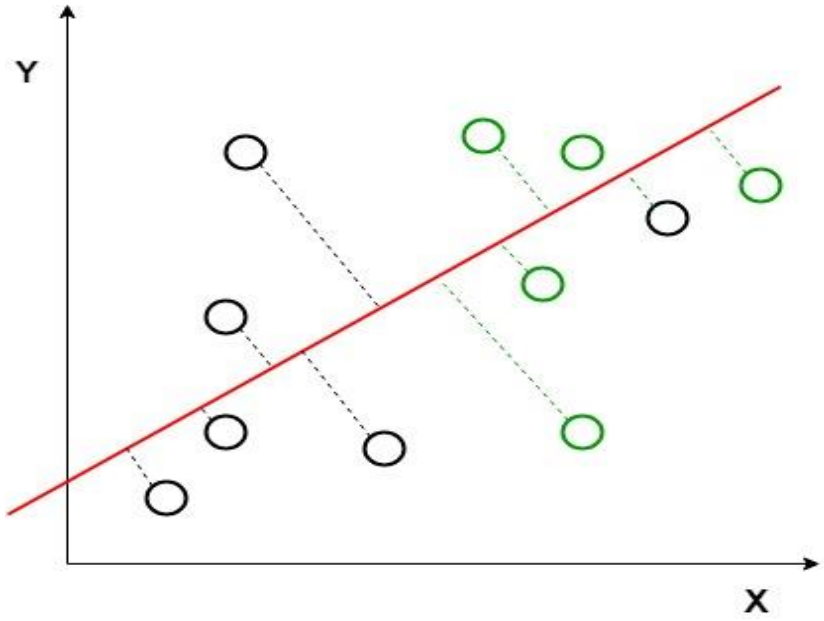


Example:

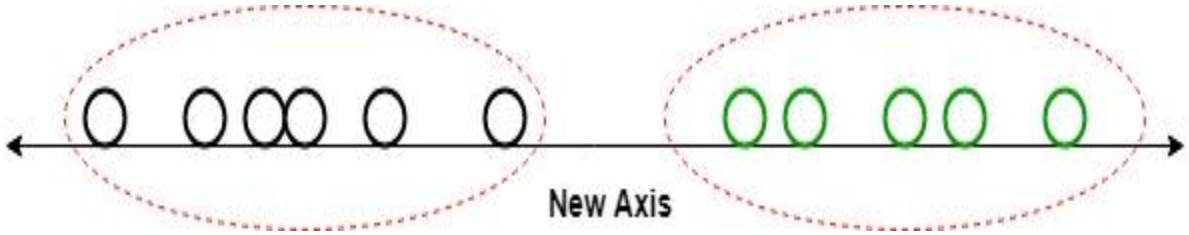
- Suppose we have two sets of data points belonging to two different classes that we want to classify.
- As shown in the given 2D graph, when the data points are plotted on the 2D plane, there's no straight line that can separate the two classes of the data points completely.
- Hence, in this case, LDA (Linear Discriminant Analysis) is used which reduces the 2D graph into a 1D graph in order to maximize the separability between the two classes.



- Here, Linear Discriminant Analysis uses both the axes (X and Y) to create a new axis and projects data onto a new axis in a way to maximize the separation of the two categories and hence, reducing the 2D graph into a 1D graph.
- Two criteria are used by LDA to create a new axis:
  - Maximize the distance between means of the two classes.
  - Minimize the variation within each class.



- In simple terms, this newly generated axis increases the separation between the data points of the two classes.
- After generating this new axis using the above-mentioned criteria, all the data points of the classes are plotted on this new axis and are shown in the figure given below.



- But Linear Discriminant Analysis fails when the mean of the distributions are shared, as it becomes impossible for LDA to find a new axis that makes both the classes linearly separable. In such cases, we use non-linear discriminant analysis.

Extensions to LDA:

- **Quadratic Discriminant Analysis (QDA):** Each class uses its own estimate of variance (or covariance when there are multiple input variables).
- **Flexible Discriminant Analysis (FDA):** Where non-linear combinations of inputs is used such as splines.
- **Regularized Discriminant Analysis (RDA):** Introduces regularization into the estimate of the variance (actually covariance), moderating the influence of different variables on LDA.

Applications:

- **Face Recognition:**
  - In the field of Computer Vision, face recognition is a very popular application in which each face is represented by a very large number of pixel values.
  - Linear discriminant analysis (LDA) is used here to reduce the number of features to a more manageable number before the process of classification.
  - Each of the new dimensions generated is a linear combination of pixel values, which form a template.
  - The linear combinations obtained using Fisher's linear discriminant are called Fisher faces.

Applications:

- **Medical:**
  - Linear discriminant analysis (LDA) is used to classify the patient disease state as mild, moderate or severe based upon the patient various parameters and the medical treatment he is going through.
  - This helps the doctors to intensify or reduce the pace of their treatment.

- **Customer Identification:**
  - Linear discriminant analysis helps to identify and select the features which can describe the characteristics of the group of customers that are most likely to buy that particular product in the shopping mall.

- **How does LDA work?**
- LDA focuses primarily on projecting the features in higher dimension space to lower dimensions. You can achieve this in three steps:
- Firstly, you need to calculate the separability between classes which is the distance between the mean of different classes. This is called the *between-class variance*.

$$S_b = \sum_{i=1}^{g} N_i (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T$$

- Secondly, calculate the distance between the mean and sample of each class. It is also called the within-class variance.

$$S_w = \sum_{i=1}^{g} (N_i - 1) S_i = \sum_{i=1}^{g} \sum_{j=1}^{N_i} (x_{i,j} - \bar{x}_i)(x_{i,j} - \bar{x}_i)^T$$

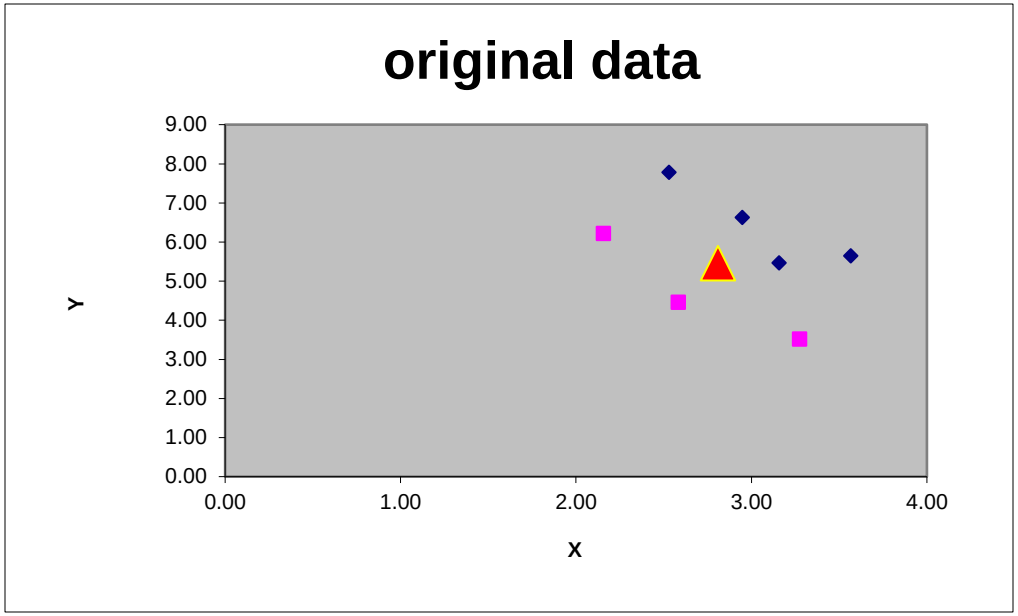
- Finally, construct the lower-dimensional space which maximizes the between-class variance and minimizes the within-class variance. P is considered as the lower-dimensional space projection, also called Fisher's criterion.

$$P_{lda} = \arg\max_{P} \frac{\left| P^T S_b P \right|}{\left| P^T S_w P \right|}$$

Problem

1. Factory “ABC” produces very expensive and high quality chips rings that their qualities are measured in terms of curvature and diameter. Results of quality control experts is given below:

Curvature	diameter	QCR
2.95	6.63	Passed
2.53	7.79	Passed
3.57	5.65	Passed
3.16	5.47	Passed
2.58	4.46	Not Passed
2.16	6.22	Not Passed
3.27	3.52	Not Passed



x = features (or independent variables) of all data. Each row (denoted by k) represents one object; each column stands for one feature.

y = group of the object (or dependent variable) of all data. Each row represents one object and it has only one column.

In our example, $\mathbf{x} = \begin{bmatrix} 2.95 & 6.63 \\ 2.53 & 7.79 \\ 3.57 & 5.65 \\ 3.16 & 5.47 \\ 2.58 & 4.46 \\ 2.16 & 6.22 \\ 3.27 & 3.52 \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \end{bmatrix}$

$\mathbf{x}_k$ = data of row k . For example, $\mathbf{x}_3 = [3.57 \quad 5.65]$, $\mathbf{x}_7 = [3.27 \quad 3.52]$

g = number of groups in $\mathbf{y}$. In our example, $g = 2$

$\mathbf{x}_i$ = features data for group i . Each row represents one object; each column stands for one feature. We separate $\mathbf{x}$ into several groups based on the number of category in $\mathbf{y}$.

$$\mathbf{x}_1 = \begin{bmatrix} 2.95 & 6.63 \\ 2.53 & 7.79 \\ 3.57 & 5.65 \\ 3.16 & 5.47 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} 2.58 & 4.46 \\ 2.16 & 6.22 \\ 3.27 & 3.52 \end{bmatrix}$$

$\boldsymbol{\mu}_i$ = mean of features in group i , which is average of $\mathbf{x}_i$

$\boldsymbol{\mu}_1 = [3.05 \quad 6.38]$, $\boldsymbol{\mu}_2 = [2.67 \quad 4.73]$

$\boldsymbol{\mu}$ = global mean vector, that is mean of the whole data set.

In this example, $\boldsymbol{\mu} = [2.88 \quad 5.676]$

$\mathbf{x}_i^o$ = mean corrected data, that is the features data for group i , $\mathbf{x}_i$, minus the global mean vector $\boldsymbol{\mu}$

$$\mathbf{x}_1^o = \begin{bmatrix} 0.060 & 0.951 \\ -0.357 & 2.109 \\ 0.679 & -0.025 \\ 0.269 & -0.209 \end{bmatrix}, \quad \mathbf{x}_2^o = \begin{bmatrix} -0.305 & -1.218 \\ -0.732 & 0.547 \\ 0.386 & -2.155 \end{bmatrix}$$

$$\mathbf{c}_i = \frac{\left(\mathbf{x}_i^o\right)^T \mathbf{x}_i^o}{n_i}$$ = covariance matrix of group i

$$\mathbf{c}_1 = \begin{bmatrix} 0.166 & -0.192 \\ -0.192 & 1.349 \end{bmatrix}, \quad \mathbf{c}_2 = \begin{bmatrix} 0.259 & -0.286 \\ -0.286 & 2.142 \end{bmatrix}$$

$C(r,s) = \frac{1}{n}\sum_{i=1}^{g} n_i \cdot c_i(r,s)$ = pooled within group covariance matrix. It is calculated for each entry (r,s) in the matrix. In our example, $\frac{4}{7}\cdot 0.166 + \frac{3}{7}\cdot 0.259 = 0.206$, $\frac{4}{7}(-0.192) + \frac{3}{7}(-0.286) = -0.233$ and $\frac{4}{7}\cdot 1.349 + \frac{3}{7}\cdot 2.142 = 1.689$, therefore

$$\mathbf{C} = \begin{bmatrix} 0.206 & -0.233 \\ -0.233 & 1.689 \end{bmatrix}$$

The inverse of the pooled covariance matrix is $\mathbf{C}^{-1} = \begin{bmatrix} 5.745 & 0.791 \\ 0.791 & 0.701 \end{bmatrix}$

$\mathbf{P}$ = prior probability vector (each row represent prior probability of group i). If we do not know the prior probability, we just assume it is equal to total sample of each group divided by the total samples, that is $p_i = \frac{n_i}{N}$

$$\mathbf{P} = \begin{bmatrix} 0.571 \\ 0.429 \end{bmatrix} = \begin{bmatrix} 4/7 \\ 3/7 \end{bmatrix}$$

Discriminant function

$$f_i = \boldsymbol{\mu}_i \mathbf{C}^{-1} \mathbf{x}_k^T - \tfrac{1}{2} \boldsymbol{\mu}_i \mathbf{C}^{-1} \boldsymbol{\mu}_i^T + \ln(p_i)$$

We should assign object k to group i that has maximum f_i

Discriminant function

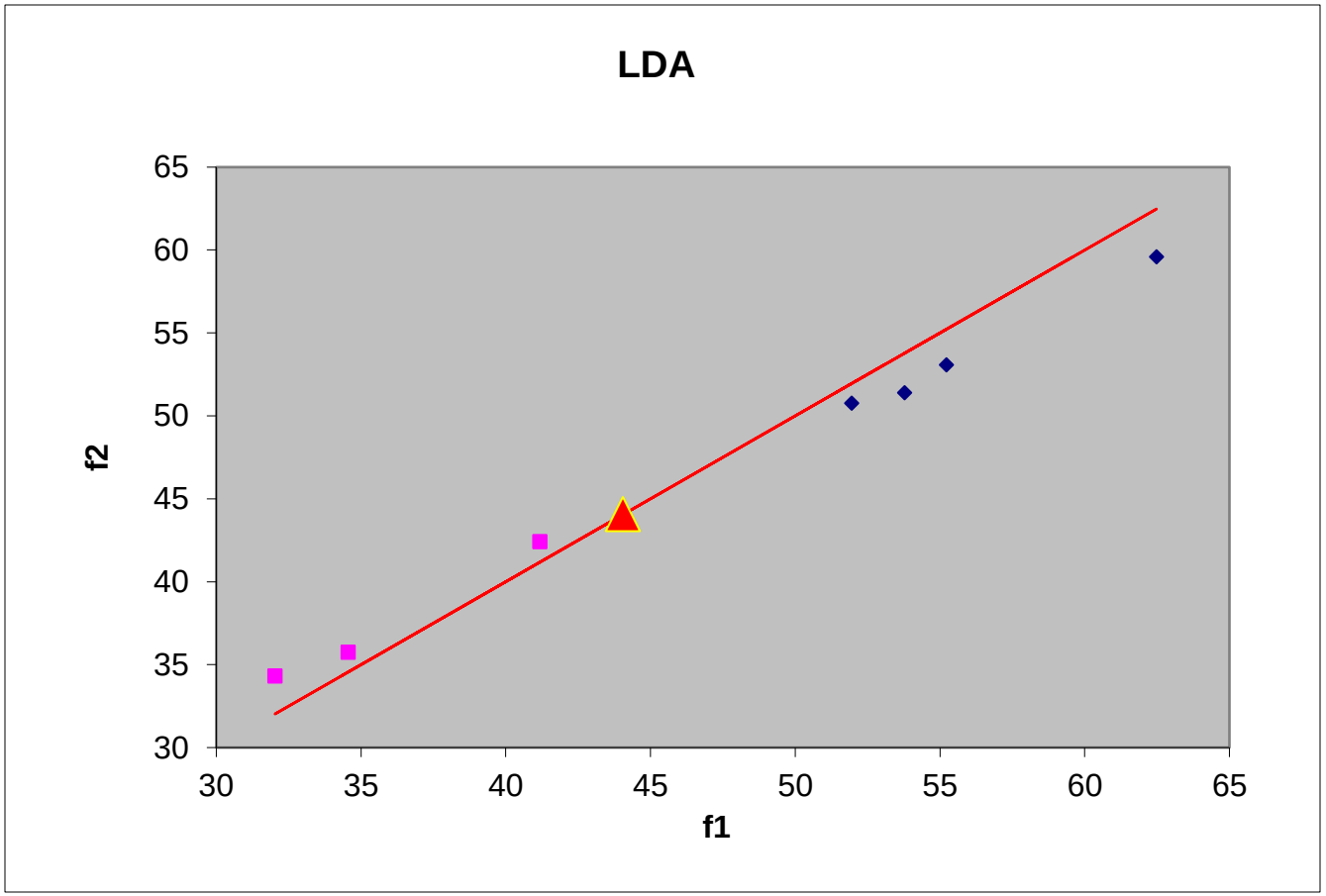
$$f_i = \boldsymbol{\mu}_i \mathbf{C}^{-1} \mathbf{x}_k^T - \tfrac{1}{2} \boldsymbol{\mu}_i \mathbf{C}^{-1} \boldsymbol{\mu}_i^T + \ln(p_i)$$

X1	X2
2.95	6.63
2.53	7.79
3.57	5.65
3.16	5.47
2.58	4.46
2.16	6.22
3.27	3.52
2.81	**5.46**

Discriminant function	
f1	f2
55.220	53.071
53.774	51.394
62.476	59.589
51.953	50.764
32.028	34.313
34.554	35.757
41.174	42.414
44.049	**44.085**

Results
Classification
1
1
1
1
2
2
2
2

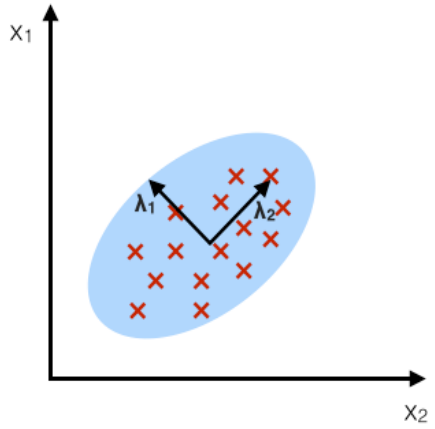
The discriminant function is our classification rules to assign the object into separate group. If we input the new chip rings that have curvature 2.81 and diameter 5.46, reveal that it does not pass the quality control.


LDA
65
60
55
50
45
40
35
30
f2
30
35
40
45
50
55
60
65
f1

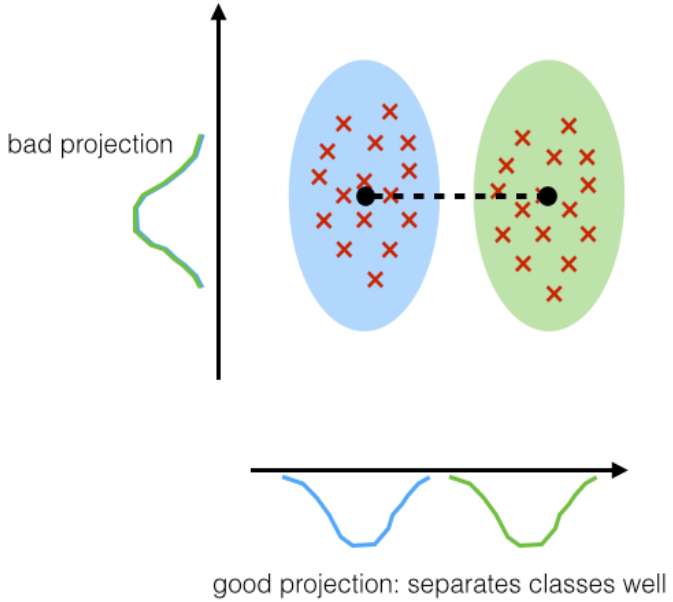
Linear Discriminant Analysis vs PCA

- PCA ignores class labels and focuses on finding the principal components that maximizes the variance in a given data. Thus it is an unsupervised algorithm.
- On the other hand, LDA is a supervised algorithm that intends to find the linear discriminants that represents those axes which maximize separation between different classes.
- LDA performs better multi-class classification tasks than PCA. However, PCA performs better when the sample size is comparatively small. An example would be comparisons between classification accuracies that are used in image classification.
- Both LDA and PCA are used in case of dimensionality reduction. PCA is first followed by LDA.

PCA:
component axes that maximize the variance



LDA:
maximizing the component axes for class-separation



The most popularly used dimensionality reduction algorithm is Principal Component Analysis (PCA). Which of the following is/are true about PCA?

1. PCA is an unsupervised method
2. It searches for the directions that data have the largest variance
3. Maximum number of principal components <= number of features
4. All principal components are orthogonal to each other

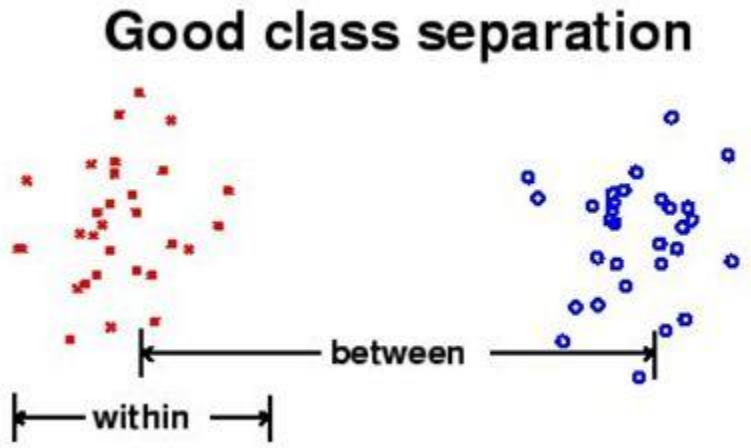
Which of the following is true about LDA?

A. LDA aims to maximize the distance between class and minimize the within class distance

B. LDA aims to minimize both distance between class and distance within class

C. LDA aims to minimize the d
maximize the distance with

D. LDA aims to maximize both
distance within class



Which of the following comparison(s) are true about PCA and LDA?

1. Both LDA and PCA are linear transformation techniques
2. LDA is supervised whereas PCA is unsupervised
3. PCA maximize the variance of the data, whereas LDA maximize the separation between different classes,

What will happen when eigenvalues are roughly equal?

A. PCA will perform outstandingly
B. PCA will perform badly
C. Can’t Say
D. None of above

Imagine, you have 1000 input features and 1 target feature in a machine learning problem. You have to select 100 most important features based on the relationship between input features and the target features.

Do you think, this is an example of dimensionality reduction?

A. Yes

B. No

It is not necessary to have a target variable for applying dimensionality reduction algorithms.

State True or False ?

Dimensionality reduction algorithms are one of the possible ways to reduce the computation time required to build a model.

A. True

B. False

What is the number of parameters needed to represent a Naive Bayes classifier with n Boolean variables and a Boolean label ?

Options:

(a) $2n + 1$

(b) $n + 1$

(c) $2n$

(d) n

Which of the following statement is TRUE about the Bayes classifier?

- Bayes classifier works on the Bayes theorem of probability.
- Bayes classifier is also known as maximum apriori classifier.
- Bayes classifier is an unsupervised learning algorithm.
- It assumes the independence between the independent variables or features.