NEHRU COLLEGE OF ENGINEERING AND RESEARCH CENTRE
(NAAC Accredited)
(Approved by AICTE, Affiliated to APJ Abdul Kalam Technological University, Kerala)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE MATERIALS

CS 467 MACHINE LEARNING

VISION OF THE INSTITUTION

To mould true citizens who are millennium leaders and catalysts of change through excellence in education.

MISSION OF THE INSTITUTION

NCERC is committed to transform itself into a center of excellence in Learning and Research in Engineering and Frontier Technology and to impart quality education to mould technically competent citizens with moral integrity, social commitment and ethical values.

We intend to facilitate our students to assimilate the latest technological know-how and to imbibe discipline, culture and spiritually, and to mould them in to technological giants, dedicated research scientists and intellectual leaders of the country who can spread the beams of light and happiness among the poor and the underprivileged.

ABOUT DEPARTMENT

- Established in: 2002
- Course offered : B.Tech in Computer Science and Engineering
  M.Tech in Computer Science and Engineering
  M.Tech in Cyber Security
- Approved by AICTE New Delhi and Accredited by NAAC
- Affiliated to the University of A P J Abdul Kalam Technological University.

DEPARTMENT VISION

Producing Highly Competent, Innovative and Ethical Computer Science and Engineering Professionals to facilitate continuous technological advancement.

DEPARTMENT MISSION

1. To Impart Quality Education by creative Teaching Learning Process
2. To Promote cutting-edge Research and Development Process to solve real world problems with emerging technologies.
3. To Inculcate Entrepreneurship Skills among Students.
4. To cultivate Moral and Ethical Values in their Profession.

PROGRAMME EDUCATIONAL OBJECTIVES

PEO1: Graduates will be able to Work and Contribute in the domains of Computer Science and Engineering through lifelong learning.

PEO2: Graduates will be able to Analyse, design and development of novel Software Packages, Web Services, System Tools and Components as per needs and specifications.

PEO3: Graduates will be able to demonstrate their ability to adapt to a rapidly changing environment by learning and applying new technologies.

PEO4: Graduates will be able to adopt ethical attitudes, exhibit effective communication skills, Teamworkand leadership qualities.

PROGRAM OUTCOMES (POS)

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSO)

PSO1: Ability to Formulate and Simulate Innovative Ideas to provide software solutions for Real-time Problems and to investigate for its future scope.

PSO2: Ability to learn and apply various methodologies for facilitating development of high quality System Software Tools and Efficient Web Design Models with a focus on performance optimization.

PSO3: Ability to inculcate the Knowledge for developing Codes and integrating hardware/software products in the domains of Big Data Analytics, Web Applications and Mobile Apps to create innovative career path and for the socially relevant issues.

COURSE OUTCOMES

CO1	To understand various learning approaches and to learn the concepts of supervised learning
CO2	To acquire knowledge about various dimensionality reduction techniques.
CO3	To learn about various performance measures and to apply various techniques like Bayesian classification used in machine learning.
CO4	To apply theoretical concepts of decision trees to find best split and to understand the concepts of artificial neural networks
CO5	To Enumerate the concepts of classifier models like SVM and HMM
CO6	To understand different clustering algorithms and applying it in real world problems.

MAPPING OF COURSE OUTCOMES WITH PROGRAM OUTCOMES

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1	**3**		**3**	**3**	**3**							
CO2	**3**	**3**	**3**	**3**	**2**							
CO3	**3**	**2**	**3**	**3**	**3**							
CO4	**3**	**2**	**3**	**3**	**3**							
CO5	**3**		**3**	**3**	**3**							
CO6	**3**	**2**	**3**	**3**	**3**							

Note: H-Highly correlated=3, M-Medium correlated=2, L-Less correlated=1

MAPPING OF COURSE OUTCOMES WITH PROGRAM SPECIFIC OUTCOMES

	PSO1	PSO2	PSO3
CO1	3	3	3
CO2		3	3
CO3	2	3	3
CO4	3	3	3
CO5	2	2	
CO6	3	3	3

SYLLABUS

Course code	Course Name	L-T-P Credits	Year of Introduction
CS467	MACHINE LEARNING	3-0-0-3	2016

Course Objectives:

- To introduce the prominent methods for machine learning
- To study the basics of supervised and unsupervised learning
- To study the basics of connectionist and other architectures

Syllabus:
Introduction to Machine Learning, Learning in Artificial Neural Networks, Decision trees, HMM, SVM, and other Supervised and Unsupervised learning methods.

Expected Outcome:
The Students will be able to :

i. differentiate various learning approaches, and to interpret the concepts of supervised learning
ii. compare the different dimensionality reduction techniques
iii. apply theoretical foundations of decision trees to identify best split and Bayesian classifier to label data points
iv. illustrate the working of classifier models like SVM, Neural Networks and identify classifier model for typical machine learning applications
v. identify the state sequence and evaluate a sequence emission probability from a given HMM
vi. illustrate and apply clustering algorithms and identify its applicability in real life problems

References:

1. Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
2. Ethem Alpaydın, *Introduction to Machine Learning* (Adaptive Computation and Machine Learning), MIT Press, 2004.
3. Margaret H. Dunham. Data Mining: introductory and Advanced Topics, Pearson, 2006
4. Mitchell. T, *Machine Learning,* McGraw Hill.
5. Ryszard S. Michalski, Jaime G. Carbonell, and Tom M. Mitchell, *Machine Learning : An Artificial Intelligence Approach*, Tioga Publishing Company.

Course Plan

Module	Contents	Hours	End Sem. Exam Marks %
I	Introduction to Machine Learning, Examples of Machine Learning applications - Learning associations, Classification, Regression, Unsupervised Learning, Reinforcement Learning. Supervised learning- Input representation, Hypothesis class, Version space, Vapnik-Chervonenkis (VC) Dimension	6	15
II	Probably Approximately Learning (PAC), Noise, Learning Multiple classes, Model Selection and Generalization, Dimensionality reduction- Subset selection, Principle Component Analysis	8	15
FIRST INTERNAL EXAM			
III	Classification- Cross validation and re-sampling methods- K-fold cross validation, Boot strapping, Measuring classifier performance- Precision, recall, ROC curves. Bayes Theorem, Bayesian classifier, Maximum Likelihood estimation, Density functions, Regression	8	20
IV	Decision Trees- Entropy, Information Gain, Tree construction, ID3, Issues in Decision Tree learning- Avoiding Over-fitting, Reduced Error Pruning, The problem of Missing Attributes, Gain Ratio, Classification by Regression (CART), Neural Networks- The Perceptron, Activation Functions, Training Feed Forward Network by Back Propagation.	6	15
SECOND INTERNAL EXAM			
V	Kernel Machines- Support Vector Machine- Optimal Separating hyper plane, Soft-margin hyperplane, Kernel trick, Kernel functions. Discrete Markov Processes, Hidden Markov models, Three basic problems of HMMs- Evaluation problem, finding state sequence, Learning model parameters. Combining multiple learners, Ways to achieve diversity, Model combination schemes, Voting, Bagging, Booting	8	20
VI	Unsupervised Learning - Clustering Methods - K-means, Expectation-Maximization Algorithm, Hierarchical Clustering Methods , Density based clustering	6	15
END SEMESTER EXAM			

Question Paper Pattern

1. There will be ***FOUR*** parts in the question paper – **A, B, C, D**
2. **Part A**
   a. **Total marks : 40**
   b. ***TEN*** questions, each have **4 marks**, covering **all the SIX modules** (***THREE*** questions from **modules I & II**; ***THREE*** questions from **modules III & IV**; ***FOUR*** questions from **modules V & VI**).
   All the TEN questions have to be answered.
3. **Part B**
   a. **Total marks : 18**
   b. ***THREE*** questions, each having **9 marks**. One question is from **module I**; one question is from **module II**; one question ***uniformly*** covers **modules I & II**.
   c. ***Any TWO*** questions have to be answered.
   d. Each question can have ***maximum THREE*** subparts.
4. **Part C**
   a. **Total marks : 18**
   b. ***THREE*** questions, each having **9 marks**. One question is from **module III**; one question is from **module IV**; one question ***uniformly*** covers **modules III & IV**.
   c. ***Any TWO*** questions have to be answered.
   d. Each question can have ***maximum THREE*** subparts.
5. **Part D**
   a. **Total marks : 24**
   b. ***THREE*** questions, each having **12 marks**. One question is from **module V**; one question is from **module VI**; one question ***uniformly*** covers **modules V & VI**.
   c. ***Any TWO*** questions have to be answered.
   d. Each question can have ***maximum THREE*** subparts.
6. There will be ***AT LEAST* 60%** analytical/numerical questions in all possible combinations of question choices.

QUESTION BANK

MODULE I			
Q:NO:	**QUESTIONS**	**CO**	**KL**
1	List out the various applications of machine learning	CO1	K2
2	Discuss about classification and regression	CO1	K2
3	Discuss about reinforcement learning	CO1	K2
4	Define machine learning and list out its main components	CO1	K2
5	Differentiate between Supervised learning & unsupervised learning	CO1	K4
6	Explain the concept of VC dimension	CO1	K2
7	Illustrate with a diagram the concept of supervised learning	CO1	K4
8	Explain about hypothesis space & version space	CO1	K2
9	Write a note on association techniques used in learning	CO1	K2
10	Define feature and input representation.	CO1	K2
11	List out different types of data.	CO1	K2
12	An open interval in R is defined as (a, b) = {x € R, a < x < b}. a and b are two parameters. Show that the set of all open intervals has a VC dimension of 2.	CO1	K6
MODULE II			
1	Define Noise.	CO2	K2
2	What are the reasons for noise and its effects on data?	CO2	K4
3	Write a note on dimensionality reduction technique	CO2	K2
4	List out advantages of using simple model.	CO2	K2
5	Describe the backward selection algorithm in detail	CO2	K2
6	Describe the forward selection algorithm in detail	CO2	K2
7	Write a note on a) True error b) Size of concept c) MSE	CO2	K2
8	Elaborate about PCA in detail	CO2	K4
9	Write a note on generalization in detail	CO2	K2
10	Write a note on PAC learning technique	CO2	K2
11	How multiple classes are classified in Machine learning	CO2	K2
12	Describe about subset selection method	CO2	K2

13	Write a note on model selection	CO2	K2
	MODULE III		
1	Discuss about k-fold cross validation method	CO3	K2
2	How bootstrapping can be used in machine learning	CO3	K3
3	Discuss about performance metrics in classifiers	CO3	K2
4	Define the term precision, recall and specificity	CO3	K2
5	Explain the use of ROC curve in machine learning	CO3	K2
6	Discuss about different regression models	CO3	K2
7	Discuss about ROC in detail	CO3	K2
8	State Bayes theorem	CO3	K2
9	How maximum likelihood is estimated in machine learning	CO3	K3
10	Discuss about different density functions	CO3	K2
	MODULE IV		
1	Define Gini index, Gini split index and gain ratio	CO4	K2
2	Define entropy with the help of a example	CO4	K2
3	Elaborate about the ID3 Decision tree algorithm	CO4	K4
4	Define CART algorithm and terms	CO4	K2
5	Describe about issues in decision learning? Specify the steps to avoid it	CO4	K2
6	Discuss about three different activation functions	CO4	K2
7	Illustrate with diagram, the Perceptron concept in neural networks	CO4	K5
8	Represent x1 AND x2 using perceptron	CO4	K6
9	Discuss about different activation functions	CO4	K2
10	Analyze the backpropagation concept used in neural networks	CO4	K4

MODULE V			
1	Write a note on support vector machine	CO5	K2
2	Discuss about soft margin and optimal separating hyperplane	CO5	K2
3	Describe about the hidden Markov model	CO5	K2
4	Elaborate about 3 problems in HMM	CO5	K4
5	Write a note on kernel functions	CO5	K2
6	Differentiate between bagging and boosting	CO5	K4
7	Discuss about voting method.	CO5	K2
8	List out the various kernel functions used.	CO5	K2
9	What is the use of stack variable in soft margin hyperplane?	CO5	K2
10	Define Margin and support vector	CO5	K2
MODULE VI			
1	Write a note on K-means clustering algorithm.	CO6	K2
2	Discuss about expectation maximization algorithm.	CO6	K2
3	Differentiate between hierarchical and density-based clustering	CO6	K4
4	Point out various distance measures used in clustering	CO6	K3
5	Elaborate about divisive clustering	CO6	K4
6	Write a note on agglomerative clustering	CO6	K2
7	Illustrate the concept of dendrogram construction using complete linkage method.	CO6	K5
8	Illustrate the concept of dendrogram construction using single linkage method.	CO6	K5
9	Discuss about DBSCAN algorithm	CO6	K2
10	Write a note on DIANA algorithm	CO6	K2

APPENDIX 1	
CONTENT BEYOND THE SYLLABUS	
S:NO;	**TOPIC**
1	Ensemble Learning
2	Expert Systems

MODULE NOTES

.

.

SDN

Introduction to Machine Learning

Machine learning can be defined as learning process where by using different techniques the machine learns from the training data.

Definition of Machine Learning

The term Machine Learning was coined by Arther Samuel in 1959. He defined machine learning as the field of study that gives computers the ability to learn without being explicitly programmed. It can be defined in other ways also:

It can be also defined as:

a) Machine learning is programming computers to optimize a performance criterion using example data or past experience. A model is defined up on some parameters, and learning is the execution of a computer program to optimize the parameters of the model using the training data or past experience.

b) The field of study known as machine learning is concerned with the question of how to construct computer programs that automatically improve with experience.

Model

Model is defined as:

- some mathematical expression or equation
- mathematical structures such as graphs and trees
- a division of sets into disjoint subsets
- a set of logical "if- then-else rules

Definition of Learning

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks T, as measured by P, improves with experience E.

Scanned by CamScanner

Examples.

1) Handwriting Recognition Problem.

Task T: Recognising and classifying handwritten words.

Performance P:- Percent of words correctly classified.

E:- A data set of handwritten words with given classification.

2) A robot driving learning problem

3) A chess learning problem.

Basic Components of learning process

1) Data storage:- facilities for storing large amount of data

Eg: hard disks, flashmemory etc.

2) Abstraction

process of extracting knowledge about stored data.

3) Generalisation

process of converting knowledge into a form that can be used for future use.

4) Evaluation

Process of giving feedback to the user

Applications of Machine Learning

1) In retail business, used to study consumer behaviour
2) In finance, banks analyze models and used in credit applications, fraud detection and stock market.
3) In manufacturing, used for optimization, control & troubleshooting.
4) In medicine, used for medical diagnosis.
5) In telecommunication, used for network optimization and maximising quality of service.
6) In science, large amount of data can be analysed. Used in world wide web, for giving relevant information.
7) In artificial intelligence.
8) used in computer vision, speech recognition and robotics
9) used in computer controlled vehicles eg. google car
10) in playing games such as chess.

feature

It is a recorded property or a characteristi of examples.

Different forms of data are used

1) Numeric data
2) Categorical or nominal
3) ordinal data
:- with categories filling in a ordered list.

Example :- instance of the unit or observatⁿ of which properties have been recorded

Learning Associations

Association rules are used

Association rule learning is a machine learning method for discovering interesting relations, called as association rules.

Example

In a super market, The manager of supermarket thinks

If a customer buys onion and potatoes together, then he/she is likely to buy hamburgers also.

Association is represented in the form of a rule as:

$$\{onion, potato\} \Rightarrow \{burger\}$$

Measured by Conditional probability

Association rules

Rules are of the form

$$X \Rightarrow Y$$

Measure by conditional probability

$$P(Y/X)$$

Support and confidence is used

Algorithms used.

1) Apriori Algorithm
2) Frequency pattern Growth algorithm

if score1 ≥ 20 & score2 ≥ 40

if score1 + score2 ≥ M

if score1 ≥ m_1 & score2 ≥ m_2

Classification

In machine learning, classification is the problem of identifying to which of a set of categories a new observation belongs.

It is based on the training set whose category or class is known.

Real life Examples

1) optical character recognition
2) face recognition
3) speech recognition
4) Medical diagnosis
5) knowledge Extraction
6) Compression

Discriminant

Defined as a rule or a function that is used to assign class labels to the observation

Algorithms used in classification

1) Logistic regression
2) Naive Bayes Algorithm
3) Decision tree Algorithm
4) Support vector Machines

Can have real valued or discrete input variables classification is divided into two:

1) Binary classification

A problem categorized with two classes is called two class or binary classification

2) Multi class classification

A problem with more than two classes is called multi class classification

Regression

In machine learning, a regression is the problem of predicting the value of a numeric variable based on observed values of variable

output variable can be a number such as integer or floating point.

can be a quantity such as amounts and sizes.

input - discrete or real valued.

- Approach used

Let

x - denotes set of input variables

y - output variable.

General approach is a model, which is a mathematical relation b/w x, y and some parameters θ.

$$y = f(x, \theta)$$

The function $f(x, \theta)$ is called regression function.

Different Regression models

Differ based on 1) number and type of independent variable

2) type of dependent variable

3) shape of regression line.

A. Simple linear Regression

There is only one continuous independent variable x. Relation b/w x and dependent variable y is.

$$y = a + bx$$

B. Multivariate linear Regression

More than one independent variable $x_1, x_2 \ldots x_n$

Relation b/w dependent and independent variable

$$y = a_0 + a_1 x_1 + a_2 x_2 \ldots + a_n x_n$$

C. Polynomial Regression

only one continuous independent variable x

Relation is

$$y = a_0 + a_1 x + \ldots + a_n x^n$$

D. Logistic Regression

Dependent Variable is binary.

Variable takes Values 0 and 1.

Different types of Learning.

Learning algorithms are classified into three types:

1) Supervised learning.

It is defined as the machine learning task of learning a function that maps an input to an output based on the basis of training examples

In this, each example in the training set is a pair consisting of an input object and an output value.

Algorithm analyzes training data and produce a function which is used for mapping new variables.

2) Unsupervised learning

It is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labelled responses.

In this, classification is not included. There is no output values and no estimation of function.

Most Common unsupervised learning method is cluster analysis used for finding hidden patterns or grouping the data.

3) Reinforcement Learning.

It is the problem of getting an agent to act in the world so as to maximize its rewards.

There is no predefined actions, instead, we will discover actions, so as to maximize the reward or output.

Examples

used to chess game, scheduling jobs, Controlling a robot limb etc.

Reinforcement learning is different from Supervised learning method.

Input Representation

General classification problem deals with assigning a class label to an unknown instance.

object or Entity to have a large number of features. All these features are not important or relevant.

only those are significant and relevant is to be considered and taken. These features are called Input features.

Representation of these input feature is Called input representation.

Example

Problem is to assign or classify the label "family Car" or "not a family Car" to a set of cars.

There are different features to the entity Car.

important features for family Car:

1) price
2) Engine power
3) seating Capacity.

These are Called input features.

Hypothesis Space

Deals with binary classification problem, ie, there is only two-classes.

Class labels can be

1) Either 0 or 1
2) True or false
3) Yes or No
4) Pass or Fail

Positive Examples

Examples with class label 1, true, yes or pass

Negative Examples

examples with class label 0, false, No or fail.

Definition of Hypothesis

In a binary classification problem, a hypothesis is a statement or a proposition which explains a given set of facts or observation.

Hypothesis space

Hypothesis space for a binary classification problem is a set of hypothesis for the problem Denoted by 'H'.

Consistency and satisfying

Let

x - example.

$c(x)$ - class label assigned to x

$x(c(x))$ = Either 0 or 1.

D - set of training examples

h - hypothesis for the problem

$h(x)$ - class label assigned to x by h.

Consistency

Hypothesis h is said to be Consistent with set of training example D, if $h(x) = c(x)$, for all $x \in D$.

Satisfy

Example x Satisfies the hypothesis h if $h(x) = 1$.

ordering of Hypothesis is also important in a binary classification problem.

Hypothesis Space - set of all hypothesis, H is represented as

$H = \{h_m : m \text{ is a real number}\}$.

Version Space

In a binary classification problem,

Let

D - set of training examples

H - set of hypothesis (Hypothesis Space)

Version Space for the problem is defined as with respect to D and H is the set of hypothesis from H consistent with D. i.e,

version space is denoted as $VS_{D,H}$

$VS_{D,H} = \{h \in H : h(x) = c(x), \text{ for all } x \in D\}$

Consistency of hypothesis is checked.

VC Dimension

VC dimension is an mathematical theory of learnability. VC dimension is the Vapnik - chervonenkis dimension named after its inventors Vladimir Vapnik and Alexey chervonenkis in 1971.

VC dimension is defined as

Let

H - Hypothesis space for a given problem.

The Vapnik - chervonenkis dimension of H, also called VC dimension of H, denoted by VC(H)

VC dimension is a measure of the complexity or capacity or flexibility of the space, H.

Shattering of a set

Let

D - dataset

N - no of examples for a binary classification problem

class labels - Either 0 or 1.

H - Hypothesis space for the problem

Each hypothesis h in H partitions D into two disjoint subset as

$\{x \in D \mid h(x) = 0\}$ and $\{h(x) = 1$ for all $x \in D\}$

Such a partition is called as dichotomy of D. There are possible 2^N dichotomies.

For each dichotomy, H can be assigned either class label 0 or 1.

If S is a subset of D; then S defines a unique hypothesis h as

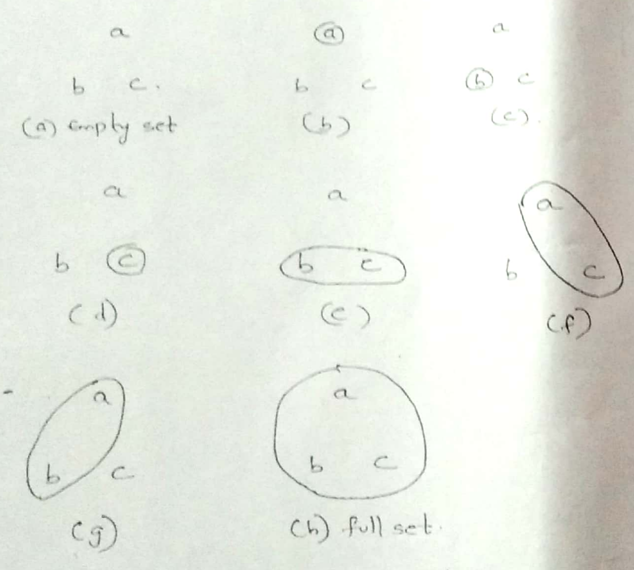
$$h(x) = \begin{cases} 1 & \text{if } x \in S \\ 0 & \text{otherwise.} \end{cases}$$

__Example__

for $D = \{a, b, c\}$

for all $\{x \in S,\ h(x) = 1\}$

Different forms are



Circle and ellipse represent the sets.

__Definition__

A set of examples D is said to be shattered by a hypothesis space H if and only if,

for every dichotomy of D, there exists some hypothesis in H consistent with the dichotomy of D.

__Definition of VC__

Vapnik - chervonenkis dimension (VC dimension) of a hypothesis space H defined over an instance space (set of all possible examples) X, denoted by $VC(H)$ is the set of all size of the ~~largest~~ largest finite subset of X shattered by H.

for large subsets of X, shattered by H, then

$$VC(H) = \infty.$$

(Appr...

CS467 Machine Learning
Module – I

Question Bank

1. List out the various applications of machine learning
2. Discuss about classification and regression
3. Discuss about reinforcement learning
4. Define machine learning and list out its main components
5. Differentiate between Supervised learning & unsupervised learning
6. Explain the concept of VC dimension
7. Illustrate with a diagram the concept of supervised learning
8. Explain about hypothesis space & version space
9. Write a note on association techniques used in learning
10. Define feature and input representation.
11. List out different types of data.
12. An open interval in R is defined as $(a, b) = \{x \in R, a < x < b\}$.a and b are two parameters. Show that the set of all open intervals has a VC dimension of 2.

CS467

Machine Learning

Module – II

Learning Multiple Classes

The Classification can be divided into two. They are:

- Binary Classification
- Multiclass Classification

Binary Classification

We have only two classes.

For eg: True or false

Multi Class Classification

Here there will be more than two classes. In this different methods are used to classify the data. The two methods used are:

- One-against-all
- One- against-one

One – against- all method (OAO)

There are k classes denoted by $C_1, C_2, C_3 \ldots\ldots C_K$. Each input instance belongs to exactly one of them.

The K-class classification problem is taken as K two-class problems. In the i-th two-class problem, the training examples belonging to C_i are taken as the positive examples and the examples of all other classes are taken as the negative examples. So, we have to find K hypotheses $h_1, h_2, \ldots h_K$.
Each h_i is defined as:

$$h_i(x) = \begin{cases} 1 & \text{if } x \text{ is in class } C_i \\ 0 & \text{otherwise} \end{cases}$$

for a given x, only one $h_i(x)$ will be 1. T!
C_i will be assigned for that class.
If more than one $h_i(x)$ is 1. we can choose a class. In such cases, classifiers re such choices.

Backword selection

In this, we start with set containing all features and each step remove the one feature that cause least error.

Procedure

Notations same as forward selection.

Algorithm

1. set $F_0 = \{x_1, x_2 \ldots x_n\}$ and $E(F_0) = \infty$.
2. for $l = 0, 1 \ldots$ Repeat the following steps until $E(F_{l+1}) \geq E(F_l)$

   (a) for all possible values of x, train the model with input $F_l - \{x\}$ and Calculate $E(F_l - \{x\})$ on validation set.

   (b) Choose input Variable x_m such that it causes least error $E(F_l - \{x\})$

   $$m = \arg\min_j E(F_l - \{x_j\})$$

   (c) set $F_{l+1} = F_l - \{x\}$.
3. set F_l is given as output.

Principal Component Analysis

It is a statistical procedure uses an orthogonal transformation to convert a set of possibly correlated Variables into a set of linearly uncorrelated Variables.

no of principal components $\leq$ no of original observation

PCA Algorithm

Step 1

Data

we have a dataset of n features denoted by $X_1, X_2 \ldots X_n$. Let there be N examp

Features	Ex 1	Ex 2	Ex3
X_1	X_{11}	X_{12}	X_{13} ..
X_2	X_{21}	X_{22}	X_{23}
X_3	X_{31}	X_{32}	X_{33} ..
⋮	⋮		
X_n	X_{n1}	X_{n2}	X_{n3}

Step 2

Compute the mean of the Variables

mean $\bar{X}_i$

$$\bar{X}_i = \frac{1}{N}(X_{i1} + X_{i2} + X_{i3} + \cdots + X_{iN})$$

A. Feature Selection

In this, we are finding k out of total n features that gives us most information. Remaining (n-k) features are discarded.

Eg. Subset selection method.

B. Feature Extraction

In this, we are finding a new set of k features, that is a combination of original n features.

Can be supervised or unsupervised methods

Eg. 1) Principal Component Analysis (PCA)
2) Linear discriminant Analysis (LDA)

Measures of Error

In regression problems, we use mean squared Error (MSE) or Root mean squared error (RMSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

y_i = observed values.

$\hat{y}_i$ = predicted values

Misclassification rate

It is also a measure of error.

This starts with no variables, and at each step, a feature is added one by one.

Procedure

Notation used

n - no of input variables.

$x_1, x_2 \dots x_n$ - denoting input variables

F_i - Subset of input variables

$E(F_i)$ - error occured in validation when only inputs in F_i is taken

Algorithm

1. Set $F_0 = \phi$ and $E(F_0) = \infty$.
2. for $i = 0, 1, \dots$ repeat the following until $E(F_{i+1}) \geq E(F_i)$
   (a) for all possible values x, train the model with $F_i \cup \{x\}$ and calculate $E(F_i \cup \{x\})$
   (b) Choose input variable x_m that causes least error $E(F_i \cup \{x\})$.
   $m = \arg\min_j E(F_i \cup \{x\})$.
   (c) Set $F_{i+1} = F_i \cup \{x_m\}$.
3. Set F_i's outputted as best output

Length or Dimension

It is defined as the no of data elements present in instance space

Commonly used instance spaces are

1) if X = n-dimensional euclidean space
   length = n.
2) if X = Conjunction of n Boolean literals
   length = n

Size of an Concept

Denoted by Size (C)

Size of a Concept C is defined as Size of smallest representation of C using some finite alphabet Σ.

Dimensionality Reduction

The complexity of any classifier depends on the number of inputs. The dataset may contain large no of variables.

for eg: In situations like image processing, internet search engines, time series analysis etc

Dimensionality reduction is defined as process of reducing the number of variables under consideration by obtaining a smaller set of principal variables.

misclassification rate = $\frac{}{\text{Total no of examples}}$

Advantages

1) Reducing the complexity of classifier
2) cost of extraction can be reduced
3) more Robust and less variance
4) Easy to explain with few features
5) Can be represented in a few dimension

Subset selection

In machine learning, subset selection or variable selection or feature selection is the process of selecting a subset of relevant features

Advantages

1) Simplification of model
2) Shorter training time
3) To avoid problems of dimensionality
4) Enhanced generalisation by avoiding overfitting

Mainly 2 methods are used

a) forward selection
b) backward selection

In 1984 Leslie Valiant proposed this framework. PAC is a computational learning theory, which is a subfield of artificial intelligence.

In this framework, the learner receive samples and must select a hypotheses from a certain class of hypothesis. The main aim is that high probability hypothesis will cause low generalisation error.

Notations used

a) X - set of instances or instance space
It can be finite or infinite

for eg: Set of all points in a plane

b) Concept class C:
It is defined on X as.

$c: X \rightarrow \{0, 1\}$. A member of C is called as Concept.

If C is a subset of X, then there is a unique function μ_c

$\mu_c : x \rightarrow \{0, 1\}$.

$$\mu_c(x) = \begin{cases} 1 & \text{if } x \in C \\ 0 & \text{otherwise} \end{cases}$$

(c) A hypothesis h is also a function

$h: \{0,1\}$

H - set of hypothesis

c) Training examples are drawn randomly from based on F.

Definition

Let X be an instance space, C is a concept class for X, h is hypothesis in C and F is an fixed probability distribution over X. The concept class C is said to be PAC-learnable if there is an algorithm A, which draws samples using F and any concept $c \in C$, with high probability produce a hypothesis $h \in H$, whose error is small.

Additional notations

True error

True error of a hypothesis h with respect to target concept c denoted by $error_F(h)$. Defined by.

$$error_F(h) = P_{x \in F}(h(x) \neq c(x))$$

$P_{x \in F}$ - probability is taken for x drawn using F.

This error denotes the probability of h misclassifying the instance x from X.

It Can be also defined as process of choosing one approach from different approaches.

This Can also be process of choosing an appropriate algorithm from a selection of possible algorithms.

one of the technique used is inductive bias.

Inductive Bias

In learning problem, Some times data itself is not sufficient to find the solution. So extra assumptions have to be made

The set of assumptions taken to find the Solution is Called as inductive bias of the algorithm.

Eg. In regression, assuming a linear function is an inductive bias.

Advantages of simple model

1) Easy to use
2) easy to train
3) fewer parameters
4) Easy to explain
5) Easy to Generalize (principle used is Occam's Razor)

The model should not be too simple.

Generalization

Generalization is defined as how well a model trained on the training set predicts the right output for new instance.

Goal of good machine learning algorithm is to generalise well from the training data.

Two problems in Generalization
1) under fitting
2) over fitting.

Underfitting

It is defined as production of a machine learning model that is not complex enough to accurately capture relationships between dataset features and target variable

overfitting

overfitting is the production of an analysis which corresponds too closely or exactly to a particular set of data

for testing generalisation, validation methods are used.

Mainly used methods are Cross Validation etc

It is also called as one versus one (ovo) strategy. In this a classifier is constructed for each pair of classes.

If there are K different class labels, then a total of $\frac{K(K-1)}{2}$ classifiers are constructed. The class getting most votes are assigned.

for eg: If there are 3 class labels A, B, C

$$\text{No of classifiers} = \frac{K(K-1)}{2}$$

$$= \frac{3 \times 2}{2} = 3$$

If any x to be classified. The classifier classify it as

A, B, B.

B gets majority votes

So x is assigned to B.

Noise

Noise is any unwanted anamoly in the data. Noise is created due to several factors:

a) Due to the problems in recording the input attributes.

b) Errors in labelling or classification

c) Additional attributes are not taken into account.

Effect of noise

a) Noise distorts the data and cause lot of problems.

b) Due to noise, accurate results cannot be produced

c) Simple hypothesis is not sufficient, complex one is used

d) Additional computing and wastage of resources.

Model Selection

Model is defined as a mathematical equation or expression or mathematical structures, like graph, tree, sets etc.

Model selection is defined as the process of choosing a particular model from a set of models.

consider variables x_l and x_j (l and j may not be different). The covariance pair (x_l, x_j).

$$\operatorname{Cov}(x_l, x_j) = \frac{1}{N-1} \sum_{k=1}^{N} (x_{lk} - \bar{x}_l)(x_{jk} - \bar{x}_j)$$

Construct a $n \times N$ S called as Covariance matrix.

$$S = \begin{bmatrix} \operatorname{Cov}(x_1, x_1) & \operatorname{Cov}(x_1, x_2) & \cdots & \operatorname{Cov}(x_1, x_n) \\ \operatorname{Cov}(x_2, x_1) & \operatorname{Cov}(x_2, x_2) & \cdots & \operatorname{Cov}(x_2, x_n) \\ \vdots & & & \\ \operatorname{Cov}(x_n, x_1) & \operatorname{Cov}(x_n, x_2) & & \operatorname{Cov}(x_n, x_n) \end{bmatrix}$$

4) Calculate the eigen values and eigen vectors

Let S be covariance matrix

iv) We form the following $n \times N$ matrix:

$$X = \begin{bmatrix} X_{11} - \bar{X}_1 & X_{12} - \bar{X}_1 & \cdots & X_{1N} - \bar{X}_1 \\ X_{21} - \bar{X}_2 & X_{22} - \bar{X}_2 & \cdots & X_{2N} - \bar{X}_2 \\ \vdots & & & \\ X_{n1} - \bar{X}_n & X_{n2} - \bar{X}_n & \cdots & X_{nN} - \bar{X}_n \end{bmatrix}$$

v) Next compute the matrix:

$$X_{new} = FX.$$

Note that this is a $p \times N$ matrix. This gives us a dataset of N samples having p features.

Step 6. New dataset

The matrix X_{new} is the new dataset. Each row of this matrix represents the values of a feature. Since there are only p rows, the new dataset has only features.

Step 7. Conclusion

- This is how the principal component analysis helps us in dimensional reduction of the dataset. Note that it is not possible to get back the original n-dimensional dataset from the new dataset.

4.4.3 Illustrative example

We illustrate the ideas of principal component analysis by considering a toy example. In the discussions below, all the details of the computations are given. This is to give the reader an idea of the complexity of computations and also to help the reader do a "worked example" by hand computations without recourse to software packages.

Problem

Given the data in Table 4.2, use PCA to reduce the dimension from 2 to 1.

Feature	Example 1	Example 2	Example 3	Example 4
X_1	4	8	13	7
X_2	11	4	5	14

Table 4.2: Data for illustrating PCA

Solution

1. Scatter plot of data

We have

$$\bar{X}_1 = \tfrac{1}{4}(4 + 8 + 13 + 7) = 8,$$
$$\bar{X}_2 = \tfrac{1}{4}(11 + 4 + 5 + 14) = 8.5.$$

Figure 4.2 shows the scatter plot of the data together with the point $(\bar{X}_1, \bar{X}_2)$.

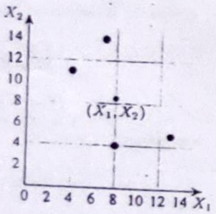


Figure 4.2: Scatter plot of data in Table 4.2

2. Calculation of the covariance matrix

The covariances are calculated as follows:

$$\begin{aligned}
\mathrm{Cov}(X_1, X_2) &= \frac{1}{N-1}\sum_{k=1}^{N}(X_{1k} - \bar{X}_1)^2 \\
&= \tfrac{1}{3}\left((4-8)^2 + (8-8)^2 + (13-8)^2 + (7-8)^2\right) \\
&= 14
\end{aligned}$$

$$\begin{aligned}
\mathrm{Cov}(X_1, X_2) &= \frac{1}{N-1}\sum_{k=1}^{N}(X_{1k} - \bar{X}_1)(X_{2k} - \bar{X}_2) \\
&= \tfrac{1}{3}((4-8)(11-8.5) + (8-8)(4-8.5) \\
&\quad + (13-8)(5-8.5) + (7-8)(14-8.5) \\
&= -11
\end{aligned}$$

$$\begin{aligned}
\mathrm{Cov}(X_2, X_1) &= \mathrm{Cov}(X_1, X_2) \\
&= -11
\end{aligned}$$

$$\begin{aligned}
\mathrm{Cov}(X_2, X_2) &= \frac{1}{N-1}\sum_{k=1}^{N}(X_{2k} - \bar{X}_2)^2 \\
&= \tfrac{1}{3}\left((11-8.5)^2 + (4-8.5)^2 + (5-8.5)^2 + (14-8.5)^2\right) \\
&= 23
\end{aligned}$$

The covariance matrix is

$$\begin{aligned}
S &= \begin{bmatrix} \mathrm{Cov}(X_1, X_1) & \mathrm{Cov}(X_1, X_2) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Cov}(X_2, X_2) \end{bmatrix} \\
&= \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}
\end{aligned}$$

3. Eigenvalues of the covariance matrix

The characteristic equation of the covariance matrix is

$$\begin{aligned}
0 &= \det(S - \lambda I) \\
&= \begin{vmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{vmatrix} \\
&= (14-\lambda)(23-\lambda) - (-11)\times(-11) \\
&= \lambda^2 - 37\lambda + 201
\end{aligned}$$

Solving the characteristic equation we get

$$\begin{aligned}
\lambda &= \tfrac{1}{2}(37 \pm \sqrt{565}) \\
&= 30.3849,\ 6.6151 \\
&= \lambda_1,\ \lambda_2 \quad \text{(say)}
\end{aligned}$$

4. Computation of the eigenvectors

To find the first principal components, we need only compute the eigenvector corresponding to the largest eigenvalue. In the present example, the largest eigenvalue is λ_1 and so we compute the eigenvector corresponding to λ_1.

The eigenvector corresponding to $\lambda = \lambda_1$ is a vector $U = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$ satisfying the following equation:

$$\begin{aligned}
\begin{bmatrix} 0 \\ 0 \end{bmatrix} &= (S - \lambda_1 I)X \\
&= \begin{bmatrix} 14-\lambda_1 & -11 \\ -11 & 23-\lambda_1 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \\
&= \begin{bmatrix} (14-\lambda_1)u_1 - 11u_2 \\ -11u_1 + (23-\lambda_1)u_2 \end{bmatrix}
\end{aligned}$$

This is equivalent to the following two equations:

$$\begin{aligned}
(14-\lambda_1)u_1 - 11u_2 &= 0 \\
-11u_1 + (23-\lambda_1)u_2 &= 0
\end{aligned}$$

Using the theory of systems of linear equations, we note that these equations are not independent and solutions are given by

$$\frac{u_1}{11} = \frac{u_2}{14-\lambda_1} = t,$$

that is

$$u_1 = 11t, \quad u_2 = (14-\lambda_1)t,$$

where t is any real number. Taking $t = 1$, we get an eigenvector corresponding to λ_1 as

$$U_1 = \begin{bmatrix} 11 \\ 14-\lambda_1 \end{bmatrix}.$$

To find a unit eigenvector, we compute the length of X_1 which is given by

$$\begin{aligned}
\|U_1\| &= \sqrt{11^2 + (14-\lambda_1)^2} \\
&= \sqrt{11^2 + (14 - 30.3849)^2} \\
&= 19.7348
\end{aligned}$$

Therefore, a unit eigenvector corresponding to $lambda_1$ is

$$\begin{aligned}
e_1 &= \begin{bmatrix} 11/\|U_1\| \\ (14-\lambda_1)/\|U_1\| \end{bmatrix} \\
&= \begin{bmatrix} 11/19.7348 \\ (14 - 30.3849)/19.7348 \end{bmatrix} \\
&= \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix}
\end{aligned}$$

By carrying out similar computations, the unit eigenvector e_2 corresponding to the eigenvalue $\lambda = \lambda_2$ can be shown to be

$$e_2 = \begin{bmatrix} 0.8303 \\ 0.5574 \end{bmatrix}.$$

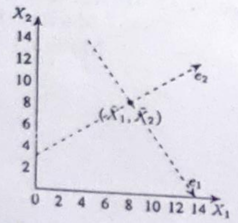


Figure 4.3: Coordinate system for principal components

5. Computation of first principal components

Let $\begin{bmatrix} X_{1k} \\ X_{2k} \end{bmatrix}$ be the k-th sample in Table 4.2. The first principal component of this example is given by (here "T" denotes the transpose of the matrix)

$$e_1^T \begin{bmatrix} X_{1k} - \bar{X}_1 \\ X_{2k} - \bar{X}_2 \end{bmatrix} = [0.5574 \quad -0.8303] \begin{bmatrix} X_{1k} - \bar{X}_1 \\ X_{2k} - \bar{X}_2 \end{bmatrix}$$
$$= 0.5574(X_{1k} - \bar{X}_1) - 0.8303(X_{2k} - \bar{X}_2).$$

For example, the first principal component corresponding to the first example $\begin{bmatrix} X_{11} \\ X_{21} \end{bmatrix} = \begin{bmatrix} 4 \\ 11 \end{bmatrix}$ is calculated as follows:

$$[0.5574 \quad -0.8303] \begin{bmatrix} X_{11} - \bar{X}_1 \\ X_{21} - \bar{X}_2 \end{bmatrix} = 0.5574(X_{11} - \bar{X}_1) - 0.8303(X_{21} - \bar{X}_2)$$
$$= 0.5574(4 - 8) - 0.8303(11 - 8.5)$$
$$= -4.30535$$

The results of calculations are summarised in Table 4.3.

X_1	4	8	13	7
X_2	11	4	5	14
First principal components	-4.3052	3.7361	5.6928	-5.1238

Table 4.3: First principal components for data in Table 4.2

6. Geometrical meaning of first principal components

As we have seen in Figure 4.1, we introduce new coordinate axes. First we shift the origin to the "center" $(\bar{X}_1, \bar{X}_2)$ and then change the directions of coordinate axes to the directions of the eigenvectors e_1 and e_2 (see Figure 4.3).

Next, we drop perpendiculars from the given data points to the e_1-axis (see Figure 4.4). The first principal components are the e_1-coordinates of the feet of perpendiculars, that is, the projections on the e_1-axis. The projections of the data points on e_1-axis may be taken as approximations of the given data points hence we may replace the given data set with these points. Now, each of these

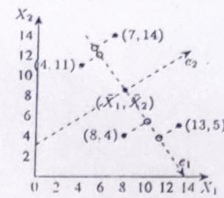


Figure 4.4: Projections of data points on the axis of the first principal component

PC1 components	-4.305187	3.736129	5.692828	-5.123769

Table 4.4: One-dimensional approximation to the data in Table 4.2

approximations can be unambiguously specified by a single number, namely, the e_1-coordinate of approximation. Thus the two-dimensional data set given in Table 4.2 can be represented approximately by the following one-dimensional data set (see Figure 4.5).

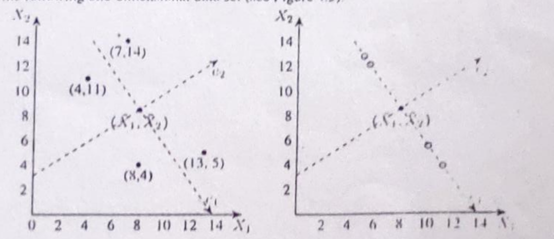


Figure 4.5: Geometrical representation of one-dimensional approximation to the data in Table 4.2

4.5 Sample questions

(a) Short answer questions

1. What is dimensionality reduction? How is it implemented?
2. Explain why dimensionality reduction is useful in machine learning.
3. What are the commonly used dimensionality reduction techniques in machine learning?
4. How is the subset selection method used for dimensionality reduction?
5. Explain the method of principal component analysis in machine learning.
6. What are the first principal components of a data?
7. Is subset selection problem an unsupervised learning problem? Why?

Machine 3

TV

Module - III

Classification

classification is the process of assigning different class labels to the data items based on certain criteria.

Also defined as the process of categorizing the given data into different class labels.

Algorithm which is used for classification is called classifier.

efficiency of an algorithm depends on:

a) error rate
b) training time
c) testing time
d) easy programmability

Performance of classification algorithms is measured using different Techniques.

one of such technique is Validation method

There are different types of Validation methods one of them is cross Validation.

Cross Validation

To test the performance, we need two sets of data. They are:-

a) Testing pair

b) Validation pair

These pairs are formed from dataset X.

If the dataset X is large, then testing pairs are formed randomly.

Dataset X.

- Training set
- Validation/testing set.

Generally X is not large enough, so we use cross validation method.

Cross Validation

Defined as a technique to evaluate the performance by partitioning original sample into:

a) Training set:-
used to train the model.

b) Test set:-
used for evaluation.

K-fold cross Validation

In K-fold cross validation, dataset X is randomly divided into k equal sized parts denoted by X_i, $i = 1, 2 \dots k$.

To generate a pair of:

one of k-parts is taken as validation set (V_i).

Remaining k-1 parts as testing set (T_i).

This process is repeated k-times.

Represented as:-

(Test) (Train)

$V_1 = X_1 \quad T_1 = X_2 \cup X_3 \cup \dots \cup X_k$

$V_2 = X_2 \quad T_2 = X_1 \cup X_3 \cup \dots \cup X_k$

$\vdots$

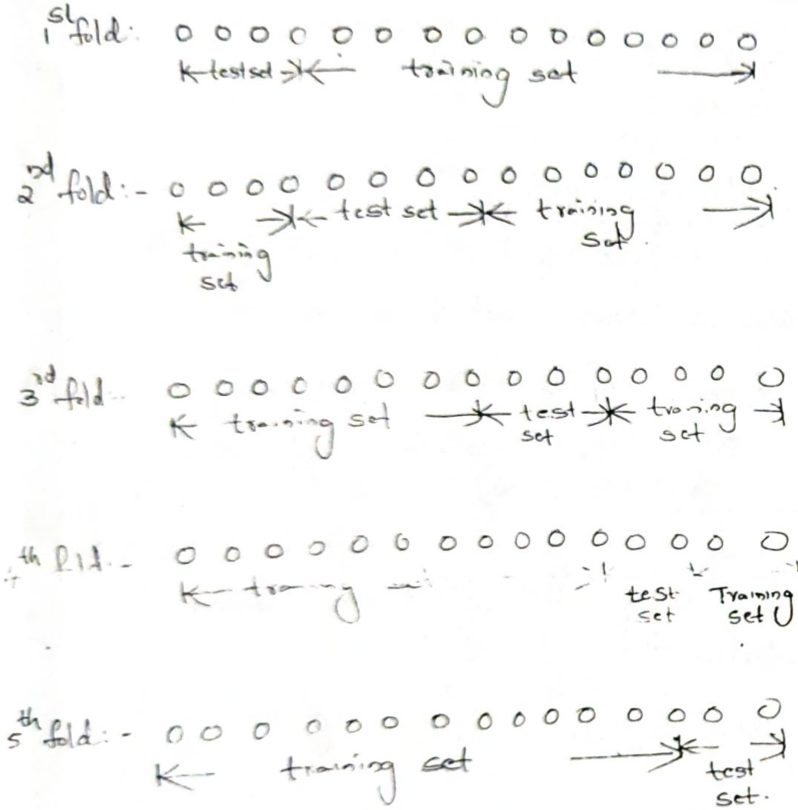
$V_k = X_k \quad T_k = X_1 \cup X_2 \cup \dots \cup X_{k-1}$

Basic concepts

1) Small validation sets are used, so that ~~testing~~ training set is large.

2) k is typically have values: 10 or 30.

Diagrammatic Representation of a 5 fold cross Validation is shown below:-



Another set of cross Validation is 5×3 Cross Validation

N
Valid 1
Train N-1

Leave one-out cross Validation

Extreme case of K- fold cross Validation.

The dataset consists of N instances

one instance - Validation set.

Remaining N-1 instance - Training set.

So V_e = one instance

T_e = N-1 instance

Application

Mainly used in applications such as medical diagnosis etc

Boot strapping

In statistics, the term bootstrap or bootstrap Sampling, refers to the process of random Sampling with replacement.

Example

Let there are five balls, A, B, C, D and E in an basket. we have to select two different balls from the basket.

In bootstrap Sampling, steps used are:-

a) we select two balls from the basket. Let it be balls A and E.

b) put the two balls back to the basket.

c) we select two balls from basket. Let balls be B and E.

d) put the balls back into the basket.

This process is repeated. So here Samples are obtained by replacement. So Bootstrap process is used.

Bootstrapping in Machine Learning.

In machine learning, bootstrapping is the process of computing performance measure using several randomly selected training and testing set. The samples are taken with replacement.

Performance measures.

Measuring Error.

We are using a binary classification model with a two class data set. Let the class labels be.

C and ¬C.

x be the test instance.

1. True positive (TP)

Let the true class label of x be C. If the model predicts the class label as C. Then we say that classification of x is True positive.

2. False Negative (FN)

True class label of x be C.

Predicted class label of x is ¬C.

Classification of x is False negative.

3. True Negative (TN).

Let the

True class label of x be ¬C.

Predicted class label of x be ¬C.

Classification of x is True Negative.

4. False Positive (FP)

Let.

True class label of x be ¬C.

Predicted class label of x be C.

Classification of x is False positive.

Diagrammatic Representation :-

	Actual class label is C	Actual class label is ¬C
predicted class label is C.	True Positive	False positive
predicted class label is ¬C	False Negative	True negative

Confusion Matrix

Confusion matrix is used to describe the performance of a classification model on a data set.

confusion table is also a table that categorizes prediction according to whether they match actual value

Confusion matrix depends on the classification type

a) Two - class datasets.

Here, there are only two classes. Confusion matrix is a table with two rows & two columns that uses TP, TN, FP & FN.

confusion matrix is

	Actual value is True	Actual value is False
Predicted value is true	TP	FP
predicted value is false	FN	TN.

Multi class Datasets.

Here, there are more than 2 class labels.

Confusion matrix can be created for multiclass datasets. All the class labels will be represented in confusion matrix.

Example

A classification system has to be trained for distinguishing cats, dogs & rabbits. Confusion matrix has to be created. There a sample of 27 animals.

out of which

no of cats = 8

" dogs = 6.

" rabbits = 13.

S/m predicted that.

out of 8 actual cats, 3 were dogs.

out of 6 dogs, one was rabbit and two were cats.

13 rabit → 2 dog.

Confusion matrix

	Actual "cat"	Actual "Dog"	Actual "Rabbit"
Predicted cat	5	2	0
Predicted dog	3	3	2
Predicted Rabbit	0	1	11

Precision and Recall

In machine learning, Precision & recall are two measures used to asses the quality of results produced by binary classifier. They are defined as:-

Let there be a binary classifier which classify a collection of data. Let -

TP - no of True Positive

TN - no of True Negative

FP - " " False positive

FN - " " False Negative.

Precision

Denoted by P.

$$P = \frac{TP}{TP + FP}$$

Recall / Sensitivity / TPR

Denoted by R.

$$R = \frac{TP}{TP + FN}$$

other measures of performance

There are several measures of performance They are:-

1) Accuracy.

Accuracy determines the correctness of the algorithm.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

2) Error rate

Defined as a measure of error.

Error rate = 1 - Accuracy

3) Sensitivity / Recall.

Defined as.

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

4) Specificity

Defined as:

$$\text{Specificity} = \frac{TN}{TN + FP}$$

5) F-measure

Defined as

$$\text{F-measure} = \frac{2 \times TP}{2 \times TP + FP + FN}$$

Receiver operating characteristic (ROC).

The acronym ROC stands for receiver operating characteristic.

It is a term which is used in Signal detection theory

ROC curve was first developed and used by electrical engineers & radar engineers during world war - II.

Used for detecting enemy objects in war fields.

ROC is also used in machine learning and data mining research

Binary classifier is used to classify the data.

TPR and FPR

TP = no of True Positive

TN = no of True Negative

FP = no of False positive

FN = no of False Negative

a) TPR (Recall)

Represents true positive rate.

Defined as.

$$TPR = \frac{TP}{TP + FN}$$

= fraction of positive examples correctly classified

= Sensitivity

FPR

Represents False Positive Rate.

$$FPR = \frac{FP}{FP+TN}$$

= Fraction of negative examples incorrectly classified

= 1 − specificity.

ROC Space

we plot the values of FPR along the X-axis (horizontal axis) and values of TPR along y-axis (vertical axis) in a plane.

for each classifier, there is a unique point in the plane represented by

• (FPR, TPR)

ROC space give the indication of performance of a classifier.

Special points in ROC space

1. Left bottom corner point (0, 0)

Always Negative prediction.

A classifier which produces this point in the ROC space never classify an example as positive.

Here

TP = 0 and FP = 0.

2. Right Top Corner Point (1, 1)

Always positive prediction.

A classifier which produces this point in the ROC space never classify an example as negative.

Here

FN = 0 and TN = 0.

3. Left Top Corner Point (0, 1)

Perfect prediction.

A classifier in this point is taken as perfect prediction.

Here FP = 0 and FN = 0.

points along diagonal

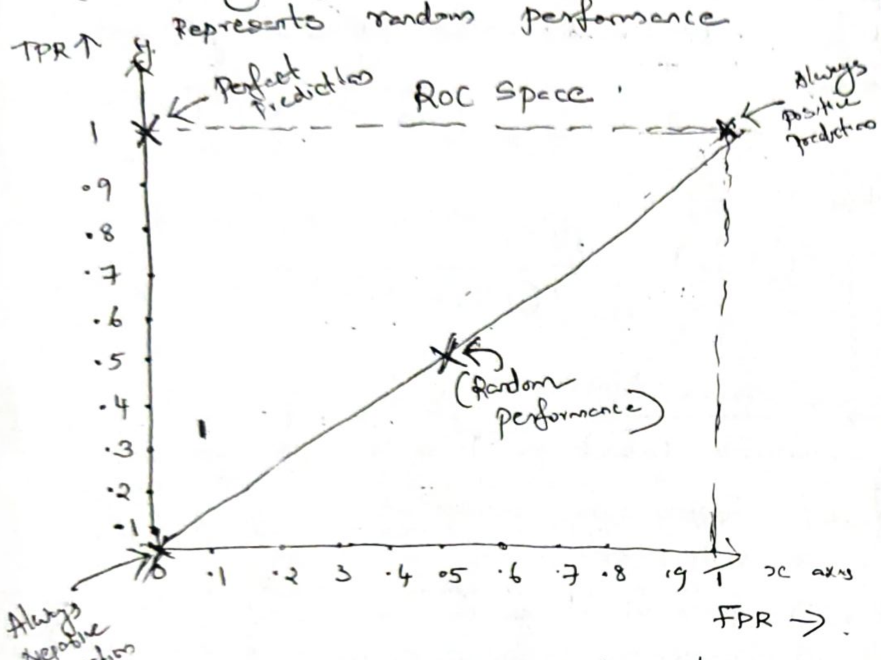


fig:- ROC space & special points.

ROC Curve

It is a curve obtained by plotting the points in the ROC space (FPR, TPR).

Area under ROC Curve.

Measure of area under the ROC curve is denoted by acronym AUC

AUC is measure of performance of classifier

For perfect classifier AUC = 1·0.

for a perfect classifier

Bayesian classifier

It is one of the type of classification algorithm used.

Conditional Probability.

Probability of occurrence of an event A given that an event B has already occurred is called conditional probability of A given B and is denoted by $P(A|B)$ we have

$$P(A|B) = \frac{P(A \cap B)}{P(B)}, \quad P(B) \neq 0.$$

Independent Events.

Two events A and B are said to be independent if

$$P(A \cap B) = P(A) \cdot P(B)$$

Pair wise Independence

Three events A, B and C are said to be pairwise independent if

$$P(B \cap C) = P(B) \cdot P(C)$$
$$P(C \cap A) = P(C) \cdot P(A)$$
$$P(A \cap B) = P(A) \cdot P(B)$$

Mutual Independence

Three events A, B and C are said to be mutually independent if

$$P(B \cap C) = P(B) \cdot P(C)$$
$$P(C \cap A) = P(C) \cdot P(A)$$
$$P(A \cap B) = P(A) \cdot P(B)$$
$$P(A \cap B \cap C) = P(A) \cdot P(B) \cdot P(C).$$

Bayes Theorem

Let A and B any two events in a random experiment. If $P(A) \neq 0$, then

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

Terms used:-

1) A is called proposition
2) B is called evidence
3) P(A) - Prior probability of Proposition.
4) P(B) - Prior Probability of evidence
5) P(A/B) = Posterior probability of A given B.
7) P(B/A) = likelihood of B given A

Generalized form

The sample space is divided into disjoint events: $B_1, B_2 \dots B_n$ and A be any event

Then

$$P(B_k/A) = \frac{P(A/B_k) \cdot P(B_k)}{\sum_{l=1}^{n} P(A/B_l) \cdot P(B_l)}$$

Naive Bayes Algorithm.

works based on following assumptions:-

1) All the features are independent and unrelated to each other.

ie, Presence or absence of a feature does not influence the presence or absence of any other feature.

2) Data has class Conditional Independence

ie, the events that are independent remains independent as long as they are Conditioned on the same class value.

Basic concepts

we have a Training set consisting of N examples having n features.

Features are denoted by ($F_1, F_2 \ldots F_n$).

feature Vector is of the form $(f_1, f_2, \ldots f_n)$.

set of class labels :- $c_1, c_2 \ldots c_p$

Test instance X

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

The main aim of the algorithm is to determine the most appropriate class label for X.

for this, we have to compute conditional probabilities

$$P(c_1/X), \quad P(c_2/X) \quad \ldots \quad P(c_p/X).$$

and choose the maximum among them.

Let maximum probability $= P(c_k/X)$.

Then c_k will be class label of X.

Computation

using Bayes Theorem.

$$P(c_k/X) = \frac{P(X/c_k) \cdot P(c_k)}{P(X)} \quad -(1)$$

By our assumption, data has class conditional independence, so the events $x_1/c_k, x_2/c_k \ldots x_n/c_k$ are independent.

Hence we have

$$P(X/c_k) = P(x_1, x_2 \ldots x_n)/c_k)$$
$$= P(x_1/c_k) \cdot P(x_2/c_k) \ldots P(x_n/c_k)$$

Substituting in eq (1) we get.

$$P(c_k/X) = \frac{P(x_1/c_k) \cdot P(x_2/c_k) \cdots P(x_n/c_k) \cdot P(c_k)}{P(X)}.$$

we have to find maximum of.

$$P(x_1/c_k) \cdot P(x_2/c_k) \cdots P(x_n/c_k) \cdot P(c_k)$$

$$P(c_k) = \frac{\text{No of examples with class label } c_k}{\text{Total no of examples}}$$

$$P(x_j/c_k) = \frac{\text{No of examples with } j^{th} \text{ feature equal to } x_j \text{ having class label } c_k}{\text{No of examples with class label } c_k.}$$

Algorithm

The training set has n features.

Denoted by $F_1, F_2 \dots F_n$

f_1 - denote arbitrary value of F_1. f_2 of F_2 and so on.

set of class labels: $C_1, C_2 \dots C_p$.

Test instance X.

$$X = (x_1, x_2 \dots x_n).$$

learning

Step 1:- Compute the probabilities $P(C_k)$ for $k = 1, 2 \dots p$.

step 2:- form a table showing conditional probabilities

$$P(f_1/c_k),\ P(f_2/c_k) \dots P(f_n/c_k).$$

for all values of $f_1, f_2, \dots f_n$ and $k = 1, 2 \dots p$.

testing

Step 3:- Compute the products-

$$q_k = P(x_1/C_k) \cdot P(x_2/C_k) \cdots P(x_n/c_k) \cdot P(C_k)$$

for $k = 1, 2 \dots p$.

Step 4:- find a j such that $q_j = \max\{q_1, q_2 \dots q_p\}$

step 5:- Assign the class label C_j to the test instance X.

In the above algorithm,

Step 1 and step 2:- learning phase of Algorithm

Remaining steps:- Testing phase of Algorithm.

Naive Bayes Algorithm can be applied to a dataset having features are categorical.

If a feature is numeric, it has to be discretized before applying the algorithm

Binning

If the feature is numeric, naive Bayes algorithm cannot be applied, we will convert it into categorical values.

Numerical values are putted into categories and they are called as 'bins'.

The process is called binning.

Binning can be done in two ways:-

1) Using Natural cut points.
2) By using manual split points.

Maximum likelihood estimation

Also called as ML estimation.

In the Bayesian classification method, we need to classify or compute the probabilities

$P(x/c_k)$ for all class labels $c_1, c_2 \ldots c_p$.

- This one is called likelihood.

Maximum likelihood Estimation (MLE) is a method of estimating the parameters of a statistical model.

MLE finds a parameter that maximizes the likelihood function

General MLE method

we have random samples,

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

It makes a probability distribution having probability density function $p(x/\theta)$.

where x denotes value of random variable

θ denotes set of parameters.

Likelihood of X is a function of parameter θ

Given by

$l(\theta) = p(x_1/\theta) \cdot p(x_2/\theta) \ldots p(x_n/\theta)$.

Maximum likelihood estimation, we find the value of θ that makes likelihood function maximum.

$L(\theta) = \log(l(\theta))$:

$= \log(p(x_1/\theta) \cdot p(x_2/\theta) \ldots p(x_n/\theta))$.

maximum likelihood estimate of θ is denoted by $\hat{\theta}$.

Density Functions

1) Bernoulli Density

In Bernoulli distribution, there are two outcomes:-

1) An event occurs or
2) An event not occurs.

The event occurs and the random variable X takes values 1 with probability P.

Non occurence of an event takes value 0 with probability $(1-P)$.

probability density function of x is given by

$$f(x/p) = p^{x} \cdot (1-p)^{1-x}, \quad x = 0, 1.$$

p is the only parameter.

MLE of p is given by $\hat{p}$.

$$\hat{p} = \frac{1}{n}(x_1 + x_2 + \cdots + x_n)$$

2. Multinomial Density

If the outcome of a random event is out of K classes, with ~~each~~ has a probability P_i.

Then $P_i = P_1 + P_2 + \cdots P_K$

$= 1.$

MLE is.

$$\hat{P}_K = \frac{1}{n}(x_{K1} + x_{K2} + \cdots + x_{Kn})$$

3) Gaussian Density Function

Also called as Normal density function.

A continuous random variable X has a gaussian or normal distribution. The density function is given by

$$f(x/\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(x-\mu)}{2\sigma^2}\right) \quad -\infty < x < \infty$$

Here μ, σ are parameters.

Maximum likelihood estimate of μ and σ is given by.

$$\hat{\mu} = \frac{1}{n}(x_1 + x_2 + \cdots + x_n)$$

$$\hat{\sigma}^2 = 1/n\left((x_1 - \mu)^2 + \cdots + (x_n - \hat{\mu})^2\right)$$

Regression

Regression is used to predict the output variable based on input variables. It is applied on continuous values.

It is a relation b/w x and y and a set of parameters θ

$$y = g(x, \theta)$$

where $g(x, \theta)$ is called regression function.

four types:

1) Simple linear Regression
2) Multivariate Regression
3) polynomial Regression
4) logistical Regression.

Regression Problem

It is the problem of determining a relation b/w one or more Independent Variables and o/p Variable

1. Simple Regression

There is only one independent Variable x. The relation b/w x and y is given by.

$$y = a + bx.$$

General form:-

$$y = \alpha + \beta x.$$

for finding optimal Values of α and β, we are using ordinary least square (OLS) method.

Let $y = a + bx$

we have to find 'a' and 'b'.

Mean of x and y is given by.

$$\hat{x} = 1/n \sum x_i$$

$$\hat{y} = 1/n \sum y_i$$

Variance of x is given by

$$Var(x) = \frac{1}{n-1} \sum (x_i - \bar{x}_i)^2$$

Covariance of x and y is denoted as.

$Cov(x, y)$

Defined as.

$$Cov(x, y) = \frac{1}{n-1} \sum (x_i - \bar{x})(y_i - \bar{y})..$$

Then

$$\boxed{\begin{aligned} b &= \frac{Cov(x, y)}{Var(x)}. \\ a &= \bar{y} - b\bar{x} \end{aligned}}$$

2. Polynomial Regression

Let there be only one Variable x and the relation b/w x and y is defined as :-

$$y = a_0 + a_1 x + a_2 x^2 + \dots a_n x^n$$

for some positive integer $n > 1$.

General form

$$y = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \dots + \alpha_n x^k$$

Values of α_i is to found out is given by :-

a) $\Sigma y_i = \alpha_0 n + \alpha_1 (\Sigma x_i) + \dots + \alpha_k (\Sigma x_i^k)$

b) $\Sigma y_i \cdot x_i = \alpha_0 \Sigma x_i + \alpha_1 \Sigma x_i^2 + \dots + \alpha_k (\Sigma x_i^{k+1})$

c) $\Sigma y_i x^2 = \alpha_0 \Sigma x_i^2 + \alpha_1 \Sigma x_i^3 + \dots + \alpha_k (\Sigma x_i^{k+2})$

$\vdots$

$\Sigma y_i x_i^k = \alpha_0 \Sigma x_i^k + \alpha_1 (\Sigma x_i^{k+1}) + \dots + \alpha_k (\Sigma x_i^{2k})$

Solving these system of linear equations, we will get Values of α_i

It can also be found by matrix representation.

Let

$$D = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^k \\ 1 & x_2 & x_2^2 & \dots & x_2^k \\ 1 & x_3 & x_3^2 & \dots & x_3^k \\ \vdots & & & & \\ 1 & x_n & x_n^2 & \dots & x_n^k \end{bmatrix}$$

and

$$\vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \text{ and } \vec{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_k \end{bmatrix}$$

$$\vec{a} = (D^T \cdot D)^{-1} \cdot D^T \cdot \vec{y}$$

where T denotes Transpose

3. Multiple linear Regression

Let there are N independent Variables say $x_1, x_2 \ldots x_N$. The relation b/w x and y is given by.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_N$$

The N-independent Variables denoted by.

$x_1, x_2 \ldots x_N$.

y - dependent Variable

Data form

Variable (features).	Values (Examples).			
	Eg 1	Eg2	......	Example N
x_1	x_{11}	x_{12}		x_{1n}
x_2	x_{21}	x_{22}		x_{2n}
⋮				
x_N	x_{N1}	x_{N2}		x_{Nn}
y (outcomes)	y_1	y_2	$y_.$	y_n

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots \beta_n X_N$$

So here

$$X = \begin{bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{N1} \\ 1 & x_{12} & x_{22} & \cdots & x_{N2} \\ & & \vdots & & \\ 1 & x_{1n} & x_{2n} & \cdots & x_{Nn} \end{bmatrix}$$

and

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \text{and} \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{bmatrix}.$$

Regression ~~coeff~~ coefficients are given by

$$\beta = (X^T \cdot X)^{-1} \cdot X^T \cdot y.$$

T denoted transpose.

Module - 3

Question Bank

1) Disscuss about Cross Validation
2) write a note on k- fold cross Validation - Illustrate a 5- fold Cross Validation
3) Explain the Concept of Bootstrapping.
4) Define precision, Recall, Sensitivity and Specificity.
5) Problems on precision, recall, sensitivity and Specifity.
6) Explain the Concept of ROC with necessary diagrams.
7) write a note on Bayes Theorem.
8) Discuss about MLE estimation.
9) Problems on linear regression and polynomial regression.
10) Explain about different density functions.
11) Define the terms TPR, FPR, f-measures, Accuracy
12) Illustrate the Concept of Naive - Bayes algorithm
13) Problems based on Naive Bayes Algorithms
14) Concepts of simple and Polynomial regression.

Module - IV

Decision trees.

Decision tree learning is a method for approximating discrete Valued target functions. The learned function is represented by a decision tree.

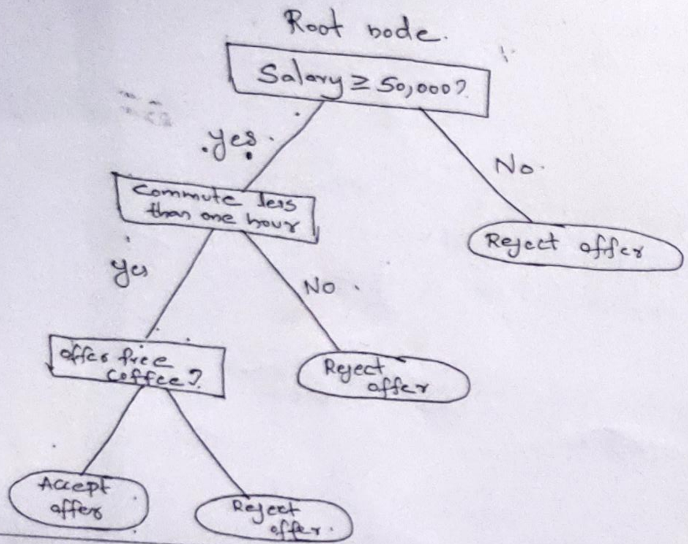
one of the most widely and practically used method for inductive reference.

Example:-

The given scenario is:-

Somebody is hunting for a job. At the beginning, he decides that he will consider only those jobs for which monthly salary is atleast Rs. 50,000. Job hunter does not like to travel much. He is comfortable only if commuting time is less than one hour. Also he expects Company to arrange a free coffee every morning.

The decision to accept or reject can be represented in a tree format. This tree is called as decision tree



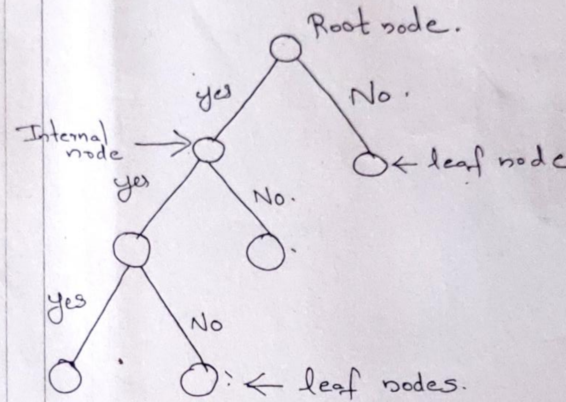
Tree refers to the concept of a tree in graph Theory.

Tree is defined as an undirected graph in which any two vertices are connected by exactly one path.

Nodes or vertices shown in ellipse shape is called as '<u>leaf nodes</u>'.

Starting node is called as '<u>root node</u>'

All other nodes except root node is called "<u>internal nodes</u>"



There are two types of decision trees:-

a) <u>Classification trees:</u>

Here the target variable can take a discrete set of values. In this, leaves represent class labels and branch represents conjunction of features.

b) <u>Regression trees</u>

Decision trees where the target variable can take continous real values.

Eg:- Price of a house, no of days patients stayed in a hospital, etc

Classification trees.

Based on the given data set, the Different Classification tree can be constructed. rules or criteria is used in construction.

The various elements in the classification tree is:-

1) Nodes in classification tree are identified by feature name of the given data

2) Branches in the tree are identified by values of features.

3) The leaf nodes are identified by class labels.

Classification tree depends on the order of selecting the features. Different feature selection measures are used.

Stopping Criteria

There will be large number of features in the dataset. Each feature has different several possible values.

The construction of classification tree is complex and time consuming, The commonly used stopping criteria's are:

a) All or nearly all of the examples have the same class.

b) There are no remaining features to distinguish

c) The tree has a predefined size limit.

Feature selection measures.

If the data set consists of n features, then ~~deciding~~ deciding the root node is a complex task. To make the process easier, we are using some methods which is termed as feature selection measures.

Populer feature selection measures are

a) Information Gain

b) Gini index

Information gain also depends on the 'entropy' of the data set.

Entropy

Entropy is a measure of impurity in the dataset.

Purity

The degree to which a subset of examples contains only a single class.

subset composed of only a single class is called a pure class.

Sets with high Entropy - diverse data

Entropy is measured in bits. If there are only two possible classes, entropy values can range from 0 to 1.

- for n classes, entropy ranges from 0 to $\log_2(n)$.
- minimum value of Entropy - Data is homogeneous.
- maximum value of Entropy - Data is diverse.

Definition

Consider a segment S of a data set having 'c' number of classes.

Let P_i - proportion of examples in S having ith class label.

Entropy of S is defined ---.

$$Entropy(S) = \sum_{i=1}^{c} -P_i \log_2(P_i).$$

The value of $0 \times \log_2(0)$ is taken as 'zero'.

Special case

Let the data segment S has only two class labels, say "yes" or "no".

P - proportion of examples having class label "yes"

1-P - proportion of examples having class label "no".

Entropy of S is defined as:

$$Entropy(S) = -P\log_2(P) - (1-P)\log_2(1-P)$$

Example 1:-

Consider a data set of animal's. The class labels are "amphi", "bird", "mammal", "reptile", fish.

No of examples with class label "amphi" = 3.

No of examples with class label "brd" = 2

" " " " " "fish" = 2

No of class label = 5

No of examples with class label "reptile" = 1.

Total no of examples = 3+2+2+2+1

= 10

$$Entropy(S) = \sum_{i=0}^{5} -P_i \log_2(P_i)$$

$$= -3/10 \log_2(3/10) + -2/10 \log_2(2/10) + -2/10 \log_2(2/10) + -2/10 \log_2(2/10) + -1/10 \log_2(1/10)$$

= 2.2464

Information Gain

Denoted by IG

Definition

Let S be set of examples

A - feature or attribute

S_v - subset of S with A = v.

Values (A) :- possible values for attribute A.

Information Gain of an attribute A relative to set S.

Denoted as Gain (S, A). Defined as:-

$$Gain(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \times Entropy(S_v)$$

where

$|S_v|$ - denotes no of elements in S_v.

$|S|$ - denotes no of elements in S.

Gini Indices

It is another type of selection measure to find suitable features.

a) Gini Index

S - a data set.

r - class labels count.

denoted by $c_1, c_2 \ldots c_r$.

P_i - proportion of examples having class label c_i

Gini index of the data set S

Denoted by Gini (S). Defined

by:-

$$Gini(S) = 1 - \sum_{i=1} p_i^2.$$

Consider the data set given in Example 1:-
The Gini index of the data set is:-

$$Gini(S) = 1 - \sum_{i=1}^{5} p_i^2.$$

$$= 1 - \left[\left(\frac{3}{10}\right)^2 + \left(\frac{2}{10}\right)^2 + \left(\frac{2}{10}\right)^2 + \left(\frac{2}{10}\right)^2 + \left(\frac{1}{10}\right)^2 \right]$$

$$= \underline{0.78}.$$

Gini split index

feature selection measure used in the construction of classification trees.
used in CART Algorithm.

S - set of examples in data set
A - feature or attribute
S_v - Subset of S with $A = v$.
$Values(A)$ - possible values for A.

Gini split index is denoted by
$Gini_{split}(S, A)$
Defined as:-

$$Gini_{split}(S, A) = \sum_{v \in values(A)} \frac{[illegible]}{|S|} \cdots v).$$

$|S_v|$ - denotes no of elements in S_v
$|S|$ - denotes no of elements in S.

Gain Ratio

Third feature selection measure in the construction of classification trees.

Let,
S - set of examples.
A - a feature having c different values
~~Values (A) - set of Values of A~~

Then

$$Gain(S, A) = Entropy(S) - \sum_{v \in values(A)} \frac{|S_v|}{|S|} \times Entropy(S_v)$$

Split Information

Denoted by $SplitInformation(S, A)$.
Defined as:-

$$SplitInformation(S, A) = -\sum_{i=1}^{c} \frac{|S_i|}{|S|} . \log_2 \frac{|S_i|}{|S|}.$$

$S_1, S_2 \ldots S_c$ - subsets of S

Gain ratio is defined by

$$\text{Gain ratio}(S,A) = \frac{\text{Gain}(S,A)}{\text{split Information}(S,A)}$$

Example

Consider a data set S having an attribute A.

$|S| = 10.$

Entropy $(S) = 2.2464.$

Gain $(S,A) = 0.5709.$

No. of examples with class label "yes" $= 4$

No of examples with class label "no" $= 6$

we have

$$\text{Split Information}(S,A) = -\frac{|S_{yes}|}{|S|}\cdot\log_2\frac{|S_{yes}|}{|S|} - \frac{|S_{no}|}{|S|}\cdot\log_2\frac{|S_{no}|}{|S|}.$$

$$= -4/10 \cdot \log_2(4/10) - 6/10 \cdot \log_2(6/10)$$

$$= \underline{\underline{0.9710}}$$

$$\text{Gain Ratio} = \frac{\text{Gain}(S,A)}{\text{split Information}(S,A)}$$

$$= \frac{0.5709}{0.9710} = \underline{\underline{0.5880}}$$

Decision tree algorithm

Helps in creating decision trees based on the information in the dataset.

Basic Concepts

a) place the best feature or attribute of the data set at root of the tree.

b) Split the training set into subsets. Subset should be in a such a way that each subset contains data with same value for a feature.

c) Repeat step 1 and step 2 on each subset until we find leaf nodes in all branches.

Well known Decision tree Algorithms.

1) ID3 (Iterative Dichotomiser 3) :- Developed by Ross quinlan

2) C 4.5 :. Developed by Ross quinlan

3) C 5.0 - Developed by Ross quinlan

4) CART (classification & Regression tree).

5) 1R (one Rule) :- Developed by Robert Holte in 1993.

5) RIPPER (Repeated incremental pruning to Produce error reduction):-

- Developed by william c Cohen, in 1995.

ID3 Algorithm

Iterative dichotomiser 3 developed by Ross quinlan in 1975.

Assumptions:-

a) The algorithm uses information gain to select the most useful attribute for classification

b) we assume that there are only two class labels named as "+" and "-".

Examples with class labels "+" - positive eg

Examples with class labels "-" - Negative eg.

Algorithm

Basic Notations

following Notations are used in the algorithm:-

S - set of examples

C - set of class labels.

F - set of features

A - Artibrary feature.

Values (A) - Set of all values of feature A.

ϑ - An arbitrary Value of A.

S_ϑ - Set of examples with $A = \vartheta$.

Root - Root node of a tree.

Algorithm

ID3 (S, F, C)

1. Create a root node for the tree
2. If (all examples in S are positive) then
3. return single node tree Root with label '+'.
4. end if.
5. If (all examples in S are negative) then
6. return single node tree Root with label '-'.
7. end if
8. If (number of feature is 0) then
9. return single node tree Root with label equal to most common class label.
10. else
11. Let A be the feature in F with highest IG

12. Assign ...

13. for all (Values v of A) do

14. Add a new ~~branch~~ branch to the tree below root corresponding to v.

15. if (S_v is empty) then

16. Below this branch add a leaf node with most common class label in set S.

17. else

18. Below this branch add the subtree formed by applying the same ID3 algorithm with Values ID3(S_v, C, F - {A}).

19. end if

20. end for

21. end if.

Regression Tree.

Regression problem is the problem of determining a relation between one or more independent Variables and an output Variables.

output Variable is a real continous Variable.

Relation is used to predict the Value of the dependent Variable.

Regression problems are ... predict numerical Value of Variables. Trees can be used for such predictions.

A tree used for making prediction of numerical Variables is called a "Prediction tree" or "Regression tree".

Example

Based on the given data, construct a tree to predict the Values of y.

x_1	1	3	4	6	10	15	2	7	16	0
x_2	12	23	21	10	27	23	35	12	27	17
y	10.1	15.3	11.5	13.9	17.8	23.1	12.7	43	17.6	14.9

Solution:-

we have to construct a tree based on the given data

Step 1:- Arbitrarily split the Values of x_1 into two sets.

$x_1 < 6$ and $x_1 \geq 6$.

The new data table is:-

x_1	1	3	4	2	0
x_2	12	23	21	35	17
y	10.1	15.3	11.5	12.7	14.9

Table (1): $x_1 < 6$.

x_1	6	10	15	7	16
x_2	10	27	23	12	27
y	13.9	17.8	23.1	43.0	17.6

Table (2): $x_1 \geq 6$.

Step 2: (a) split the table (1) based on value of x_2.

$x_2 < 21$ and $x_2 \geq 21$

x_1	1	0
x_2	12	17
y	10.1	14.9

(a) Table (3).
$x_2 < 21$.

x_1	3	4	2
x_2	23	21	35
y	15.3	11.5	12.7

Table (4):
$x_2 \geq 21$.

(b) [illegible]

$x_2 < 23$ and $x_2 \geq 23$.

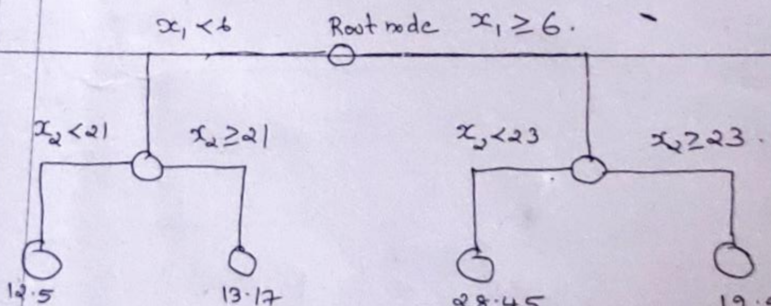
x_1	6	7
x_2	10	12
y	13.9	43.0

Table (5):-
$x_2 < 23$.

x_1	10	15	16
x_2	27	23	27
y	17.8	23.1	17.6

Table (6):-
$x_2 \geq 23$.

It can be diagrammatically represented as:-



Leaf node value is the average value of y in table (3), (4), (5) and (6).

Algorithm

Mainly three elements are considered:-

1. A way to select split to divide the data
2. A rule for determining the terminal nodes
3. A rule for assigning a value to the terminal node.

Notations used:-

$x_1, x_2 \dots x_n$:- input Variables

N :- number of Samples in the data set

$y_1, y_2 \dots y_N$:- The Values of o/p Variable

T :- A tree

c - leaf of T.

n_c :- no of data elements in leaf c.

C :- set of indices of data elements in the leaf node c.

m_c :- mean of Values of y which are in leaf c

S_T :- Sum of squares of errors in T.

Then

$$m_c = \frac{1}{n_c} \sum_{i \in C} y_i$$

$$S_T = \sum_{c \in leaves(T)} \sum_{i \in C}$$

Algorithm

step 1:- Start with a single node containing all data points. Calculate m_c and S_T.

step 2:- If all the points have same Value for all the independent Variables. stop.

Step 3:- otherwise, Search over all binary splits of all Variables for the one which will reduce S_T.

Step 3(a):- If S_T < threshold δ, one of the resulting nodes contain less than q points, stop. Assign m_c to the node.

3(b):- otherwise, take that split, create two new nodes.

Step 4:- In each node, do step 1 to 3.

CART Algorithm

CART (Classification and Regression tree) was introduced in 1984.

Developed by Leo Breiman, Jerome fredman, Richard olshen and charles Stone

Two types of trees in CART:-

Classification Tree

The target Variable is Categorical and the tree is used to identify the "class".

Regression Tree

The target Variable is continous and tree is used to predict the Value.

Main elements are:-

(a) Rules for splitting the data at a node based on Value of one Variable

(b) Stopping rule for deciding terminal node.

(c) prediction of target Variable at terminal node.

Other Decision Tree algorithms.

1. **C 4.5 algorithm**

Developed by Ross quinlan as Improvement of ID3.

Important improvements in ID3:-

(a) Handling both continous & discrete attributes.

(b) Handling training data with missing attribute Values

(c) Handling ......

(d) pruning trees after creation.

2. **C 5.0 algorithm**

An improvement of the C 4.5 algorithm.

Developed by Ross quinlan.

Main features are:-

(a) speed:- C 5.0 is faster than C 4.5

(b) Memory usage:- C 5.0 is more memory efficient than C 4.5.

Smaller decision trees are formed.

Issues in decision tree learning

1. **Avoid overfitting of data**

Important concept to be considered while constructing a decision tree.

Definition

A hypothesis overfits the training examples if some other hypothesis that fits the training example less, but performs better over the entire distribution of instances.

Impact of overfitting.

1) Accuracy of the tree is based on overfitting.

Accuracy decreases, then the predicted output will be getting wrong.

Approaches for avoiding overfitting

Main approach to avoid overfitting is:-

Pruning

Pruning is a technique that reduces the size of decision trees by removing section of the tree that is not important.

Pruning reduces the complexity of the classifier. Increases the accuracy of prediction.

Two types of pruning:-

(a) Pre pruning:-

we apply pruning earlier, ie, before the classification of data.

(b) post pruning:-

Allow the tree to overfit the data, then applying the pruning process.

Reduced error pruning.

In reduced error Pruning, each of the decision tree is taken as candidate for pruning

Pruning process consists of removing the subtree at that node and making it a leaf node with most common classification value.

2. Problem of missing attributes.

The missing attribute values in the dataset are indicated by '?'.

Methods for avoiding problem of missing attributes:-

(a) Deleting cases with missing attribute values.

(b) Replacing the missing attribute value by most common value.

(c) Assigning all possible values to the missing values.

(d) Replacing the missing value by the mean of numerical attributes.

(e) Assigning the missing value with value taken from nearest 't' cases

(f) replacing the missing attribute value by a new value.

Neural networks.

Artificial Neural network (ANN) models the relationship between a set of input signals and an output signal using a model.

Artificial Neurons.

Artificial neuron is a mathematical function which models the concept of biological neurons.

Elementary units in ANN are called artificial neurons.

Each input signals are seperately weighted, and the sum is passed through a function.

The function is called as 'activation function' or 'transfer function'

The diagrammatic representation of a artificial neuron is shown below:-

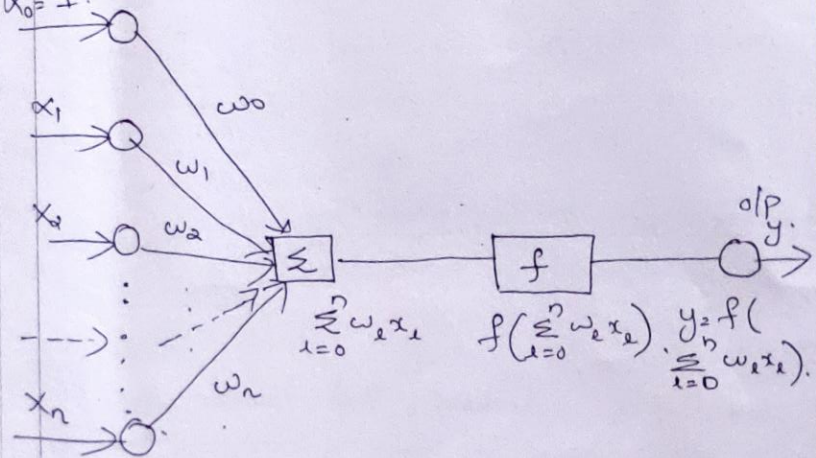


fig:- Artificial Neuron.

Basic Notations

$x_1, x_2 \ldots x_n$:- input signals.

$w_1, w_2 \ldots w_n$:- weights associated with each input signals.

x_0 :- Input signal with constant value 1.

w_0 :- weight associated with x_0. Called as "bias" or "threshold"

Σ :- Indicates summation

f :- function which produces the o/p.

y :- o/p signal

output signal $y =$

$$y = f\left(\sum_{i=0}^{n} w_i x_i\right).$$

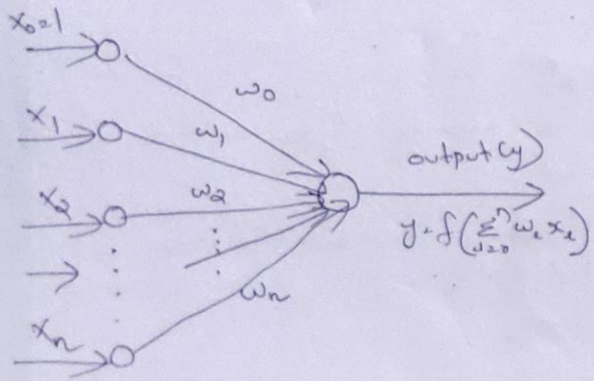
Small circles in the neuron are called "nodes" of the neuron.

Circle on left side which receives $x_0, x_1, x_2 \dots x_n$ are called "input nodes"

Circle on right side which outputs the value of y is called "output node"

Square represents the process taking place in the neuron.

It can be simply represented as:-



Activation function

Definition:-

In ANN, the function which takes the incoming signals as input and produces the output signal is known as activation function.

$f\left(\sum_{i=0}^{n} w_i x_i\right)$ represents a activation function.

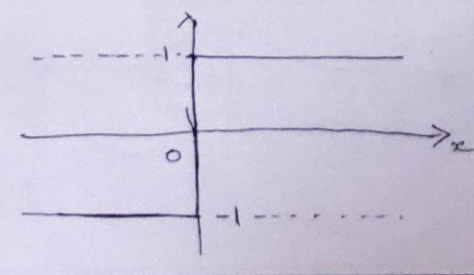
Simple Activation function

1. Threshold activation function

The threshold activation function is defined by:-

$$f(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x \le 0. \end{cases}$$

The graph of the function can be represented as:-

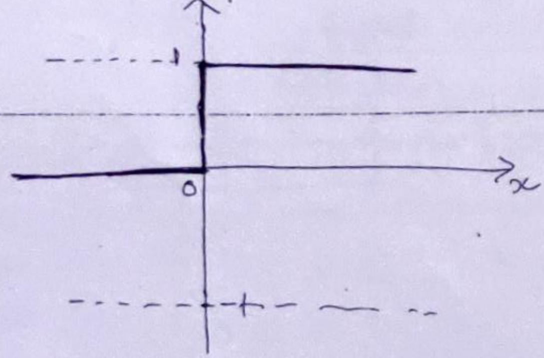


2. Unit step function

The activation function can be a unit step function. Defined as:-

$$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

The graph of the function is:-



3. Sigmoidal Activation function

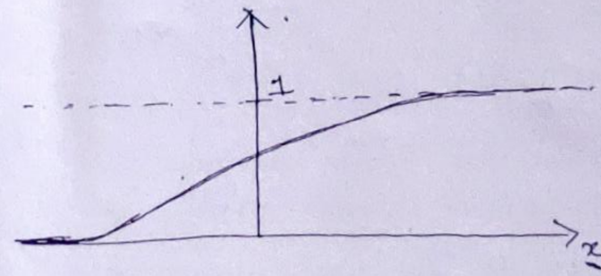
It is also called as logistic function.

One of the most commonly used activation function.

Defined as:-

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

Graph of the function is:-



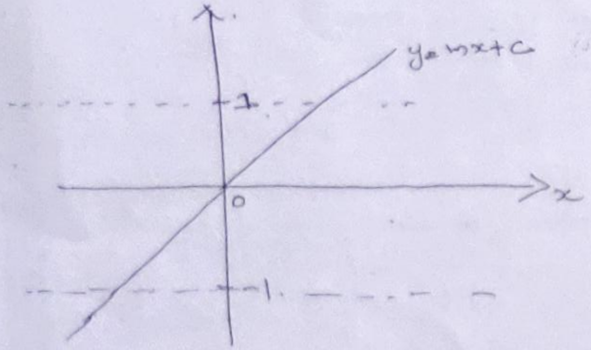
4. Linear activation function

Linear activation function is defined by:-

$$f(x) = mx + c$$

Defines a straight line in the xy plane.

The Graph of the function is represented as:-



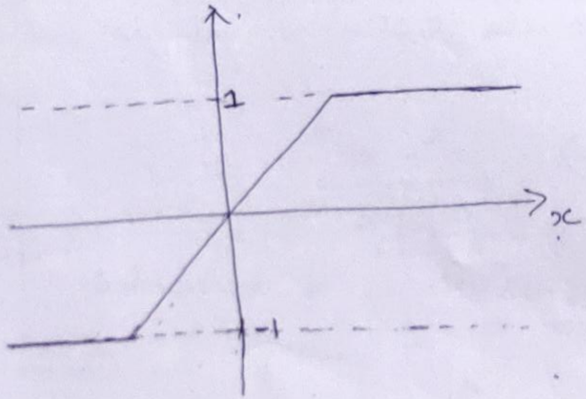
5. Saturated Linear activation function

Also called as piecewise linear activation function.

Defined by:-

$$f(x) = \begin{cases} 0 & \text{if } x < x_{min} \\ mx+c & \text{if } x_{min} \leq x \leq x_{max} \\ 0 & \text{if } x > x_{max} \end{cases}$$

The graph of the function is represented as:-

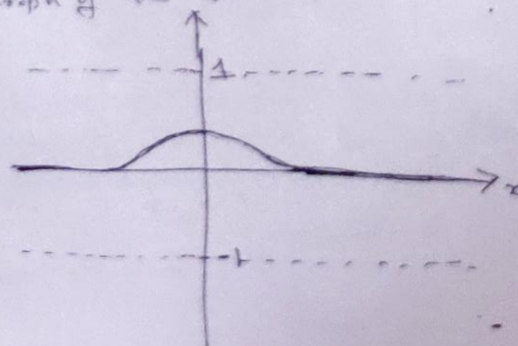


6. Gaussian activation function

Defined by:-

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Graph of the function is :-

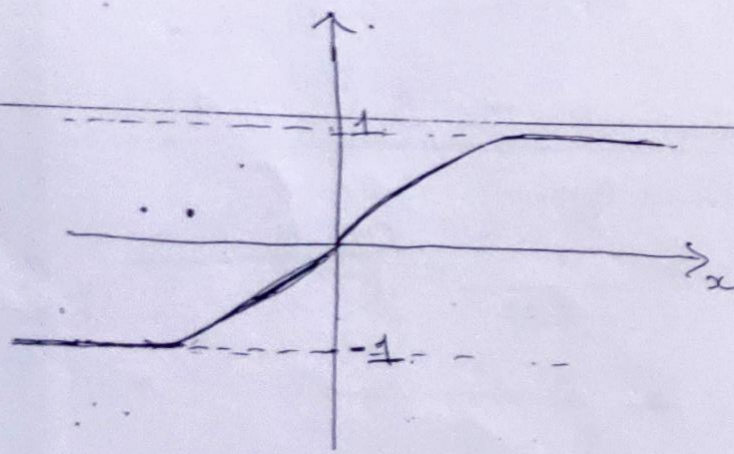


7. Hyperbolic tangential activation function

The activation function is defined by:-

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

Graph of the function is represented as:-



Perceptron

Perceptron is a special type of artificial neuron in which the activation function has a special form.

Definition:-

Perceptron is an artificial neuron in which the activation function is:-

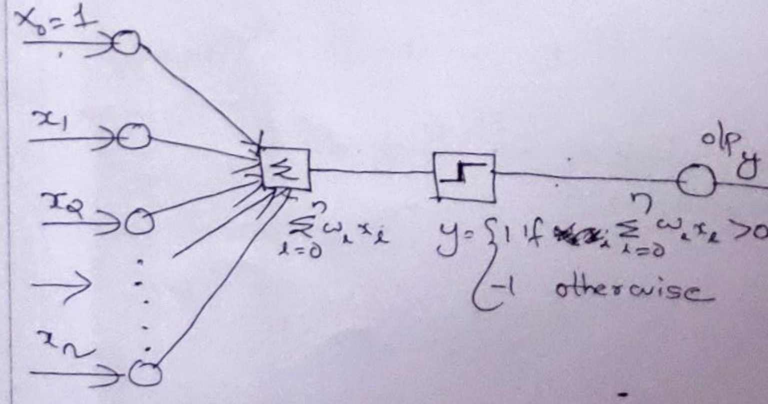
$$o(x_1, x_2 \cdots x_n) = \begin{cases} 1 & \text{if } w_0 + w_1 x_1 + w_2 x_2 + w_n x_n > 0 \\ -1 & \text{if } w_0 + w_1 x_1 + \cdots + w_n x_n \leq 0 \end{cases}$$

$x_1, x_2 \cdots x_n$ is the input signals.

$w_1, w_2 \cdots w_n$ - weight associated with input signals.

w_0 - constant, called as bias.

The schematic diagram of a perceptron is:-



Representation of boolean function by perceptron

The different boolean function like 'AND', 'OR', 'NAND', 'NOR' etc can be represented with the help of a perceptron. The Values are taken as -1 and 1.

$-1 \Rightarrow$ represents "false"

$1 \Rightarrow$ represents "true".

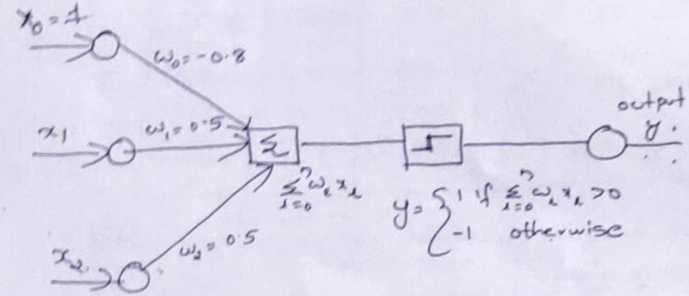
Representation of x_1 AND x_2.

Let x_1, x_2 be two boolean Variables.

The boolean function x_1 AND x_2 can be represented as.

x_1	x_2	x_1 AND x_2
-1	-1	-1
1	-1	-1
-1	1	-1
1	1	1

Diagrammatic representation of [illegible] perceptron is:-



The output y is given by:-

$$y = \begin{cases} 1 & \text{if } \sum_{i=0}^{n} w_i x_i > 0 \\ -1 & \text{otherwise} \end{cases}$$

or

$$y = \begin{cases} 1 & \text{if } -0.8 + 0.5 x_1 + 0.5 x_2 > 0 \\ -1 & \text{otherwise} \end{cases}$$

The boolean functions x_1 OR x_2, x_1 NAND x_2, x_1 NOR x_2 Can be represented using perceptron.

The weights assigned for these boolean function is:-

Boolean function	w_0	w_1	w_2
x_1 AND x_2	-0.8	0.5	0.5
x_1 OR x_2	0.3	0.5	0.5
x_1 NAND x_2	0.8	-0.5	-0.5
x_1 NOR x_2	-0.3	-0.5	-0.5

All boolean functions cannot be represented by perceptron.

x_1 NOR x_2 cannot be represented using ~~Perceptron~~ because, the values of w_0, w_1, w_2 cannot be find out such that it produces correct output.

Learning a perceptron

Also called as perceptron learning rule.

Learning a perceptron means the process of assigning values to the weights and threshold such that perceptron produces correct output.

Perceptron learning algorithm

The following notations are used in the algorithm:-

n :- number of input variable

$y = f(z)$:- output from the perceptron for an input vector z.

$D = (x_1, d_1), (x_2, d_2) \ldots (x_s, d_s)$:- training set of s examples.

$x_j = (x_{j0}, x_{j1}, \ldots x_{jn})$:- n dimensional input vector.

d_j :- Desired o/p value of the perceptron for input x_j.

x_{jk} :- Value of i^{th} feature of j^{th} training vector.

x_{j0} :- Constant has a value 1.

w_i :- weight of the i^{th} input variable

$w_i(t)$:- weight i at the t^{th} iteration.

Algorithm

Step 1:- Initialise weights and threshold; weights are initialized to 0 or to a small random value

Step 2:- for each example, j in the training set D, perform the following steps over x_j and d_j

(a) Calculate the actual output:-

$$y_j(t) = f\left[w_0(t)\cdot x_{j0} + w_1(t)\cdot x_{j1} + \ldots + w_n(t)\cdot x_{jn}\right]$$

(b) update the weights

$$w_i(t+1) = w_i(t) + (d_j - y_j(t))\cdot x_{ji}$$

for all features, $0 \le i \le n$.

Step 3:- step 2 is repeated until

(i) iteration Error $= \frac{1}{n} \sum_{j=1}^{n} |d_j - y_j(t)|$ is less than a threshold value γ

or

(ii) a predefined number of iterations have been completed

Above algorithm [illegible]
training examples are linearly seperable.

Characteristics of ANN

The artificial neural network can be defined and implemented in different ways. The main charactoristics of ANN are:-

1. Activation function

This function explain how the combined input signals are transformed into a single output signal.

There are different types of activation function used in ANN, which is explained above in the topic activation function.

2. Network Topology

It defines the patterns and structures which is used in the interconnected nodes.

Topology determines the complexity of the task to be done.

Network Topology is based on:-

a) Number of layers :-

In ANN, there can be different types of layers.

Input nodes:- Those nodes which receive unprocessed signals directly from input data

output nodes:- Those node which produce the final predicted values. They can be one or more than one.

Hidden node:- Node that process the signals from the input nodes (or other nodes) prior to reaching the output nodes.

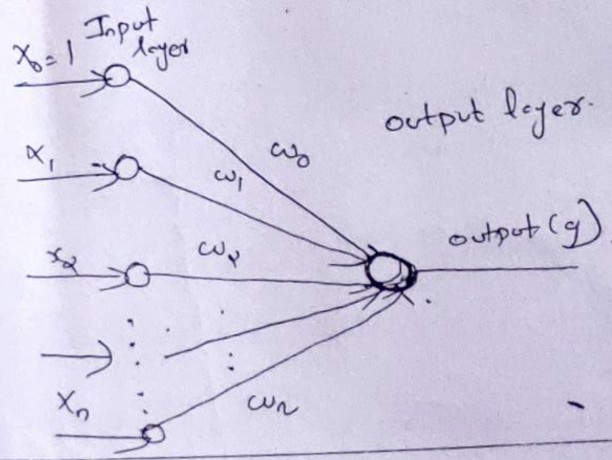
These nodes are arranged in layers.

set of nodes which receive the unprocessed signals from the data form the "first layer."

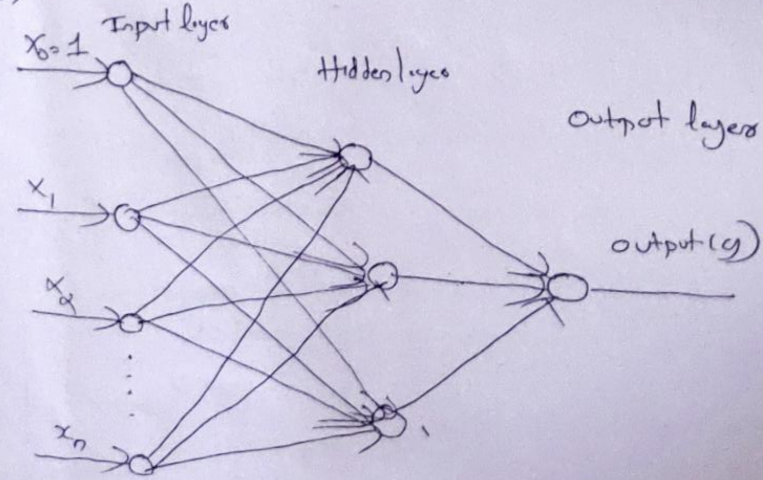
The set of nodes which receive the signals from input nodes, ie, hidden nodes constitute "second layer"

Eg:-

(1).A ANN with only one layer



(2) An ANN with two layers



(b) Direction of information travel

Networks in which the input signals are fed continously in one direction from connection to connection until it reaches the output node is called "feed forward networks"

Network which allow signals to travel in both directions using loop is called "recurrent networks" or "feedback networks"

Commonly used one is the feed forward network.

The multilayer feed forward network is called as "Multilayer perceptron" (MLP).

(c) No of nodes in each layer

The number of input nodes is determined by the number of features in the data

no of input nodes. → no of features in the data

The number of output nodes is determined by the :

(a) no of out comes.

or

(b) no of class levels.

The number of hidden nodes can be decided by the user depending on the problem.

3. Training Algorithm

There are mainly two methods used to train a perceptron. they are:-

(a) perceptron rule:-

Applied when the given training data set is linearly seperable.

(b) delta rule:-

Applied when the traning data set is not linearly seperable.

The most commonly used algorithm now is the back propagation algorithm.

4. Cost function

Cost function is a function that measures how well the algorithm maps the target function.

Cost function is also called as "loss function" or "objective function" or "scoring function" or "error function".

Let y be the output Variable.

$y_1, y_2 \cdots y_n$:- Actual values of y.

$\hat{y}_1, \hat{y}_2 \cdots \hat{y}_n$ be the predicted values of y

Then we use two methods.

(a) Sum of Square Error (SSE)

Defined by

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

(b) Mean Square Error (MSE)

Defined by

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Back propagation

The backpropagation algorithm was developed in 1985-86.

The Basic idea of the algorithm is:-

1. Initialize the weights which are assigned a random value.
2. Algorithm Iterates through many cycles which consists of two processes until a stopping criterion is reached.

Each cycle is called as "epoch".

Each epoch includes:-

(a) forward phase:-

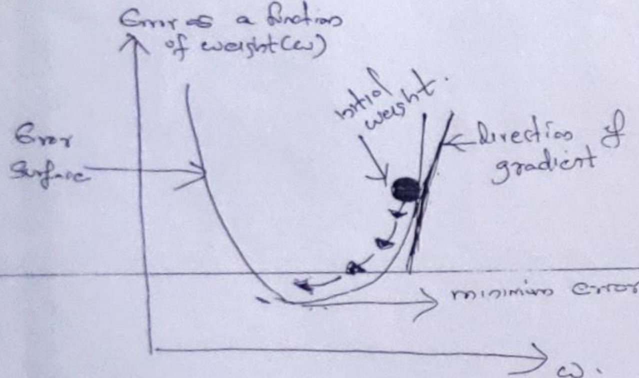
The neurons are activated in sequence from input layer and after applying the activation function, final layer is reached producing the output signal.

(b) Backward phase:-

Network's output signal from the forward phase is compared with the true actual value in the training data

The difference between the output signal and true output is sent as a feedback to the network. The weights and threshold value has to be updated.

The Technique used to change how much a weight is to be changed is determined by the "gradient descent method."



Question bank

1. Explain about
(a) Entropy (b) Information Gain (c) Gain ratio
(d) Gini index
2. write a note on decision tree used in machine learning
3. Explain the ID3 algorithm in detail.
4. write a note on CART algorithm..
5. Discuss about issues in decision tree learning.
6. Problems based on feature selection measures (Entropy, IG, Gini index etc).
7. Problems based on ID3 algorithm
8. write a note on perceptron used in ANN.
9. Define activation function.
10. with the help of diagrams, explain different type of activation function.
11. write a note on artificial neural networks
12. Discuss about the backpropagation technique used for training.

10. Write a short note on

(a) Gini split index (b) Bias

14:- Discuss about the characteristics of ANN.

15. Elaborate the concept of Regression trees.

16. Problems for finding root node based on the given data set.

17. Write a note on various decision tree algorithms.

Each epoch includes:

a) forward phase :-

neurons are activated in sequence from input layer to output layer. Applying weight and activation function. o/p signal is produced.

b) Backward phase

Network output signal resulting from forward phase is compared with true target value in training set.

Technique used to determine how weight can be changed is called gradient descent method.

Back propagation algorithm is used.

Machine 5

Module - 5

Support Vector Machines

Important concept in machine learning.

used for taking a particular decision or predicting a particular value

Data set can be of two types.

a) Two class data set

Here the variable can have only two values or labels.

for eg: yes or no.

when there are only two class labels Then data set is called Two class data set.

Scatter plot

Graphical representation of the data points. we plot the features or parameters.

Since, there is two class data set, one parameter is x- axis and other in y-axis.

Seperating Line

we can draw a straight line seperating the two types of points such a straight line is called seperating line

It should have following property:

1) $ax + by - c < 0$ and
2) $ax + by - c > 0$.

Linear seperable Data

If there is a seperating line exists for dividing the data, Then it is called linearly seperable.

There can be several seperating lines to divide the data into sections.

Margin of a seperating line

To choose the best seperating line we use 'margin' Concept.

perpendicular distance of data points from seperating line.

Double of shortest perpendicular distance is called margin of seperating line.

Maximum margin seperating line

Best seperating line is one with the maximum margin.

Seperating line with maximum margin is called maximum margin line or optimal seperating line. → SVM.

This line is also called as support vector machine.

Support Vectors

Datapoints which are closest to maximum margin line are called support Vectors.

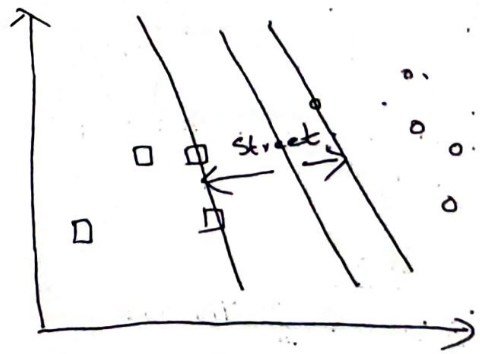
Different criterions are used:

Street of maximum width

we draw a line through support vector (1 and 2) on one side of seperating line parallel to maximum margin line.

Another line through support vector on other side of seperating line. parallel to maximum margin line.

Region between these two parallel lines are called street of maximum width



Any line can be of the form

$$ax+by-c=0.$$

Seperates the plain into two halves.

one half - $ax+by-c>0$ and

other half - $ax+by-c\leq 0$.

Finite Dimensional Vector Spaces

Definition

Let n be a positive integer.

n-dimensional vector - is an ordered n-tuple of real numbers of form $(x_1, x_2 \ldots x_n)$.

Denote vectors as $\vec{x}, \vec{y}$.

Component

$$\vec{x} = (x_1, x_2 \ldots x_n)$$

The numbers $x_1, x_2 \ldots x_n$ are called co-ordinates or components of $\vec{x}$.

The real numbers are also called as scalars.

n-dimensional vector space is represented by R^n

operations

Addition of Vectors

Let $\vec{x} = (x_1, x_2, \ldots x_n)$ and $\vec{y} = (y_1, y_2 \ldots y_n)$.

Sum of $\vec{x}$ and $\vec{y}$, denoted by $\vec{x}+\vec{y}$.

$$\vec{x}+\vec{y} = (x_1+y_1, x_2+y_2, \ldots x_n+y_n)$$

2. Mutiplication by Scalar

Let α be a scalar and $\vec{x} = (x_1, x_2 \ldots x_n)$

Product of $\vec{x}$ by α.

Denoted as $\alpha\vec{x}$.

$$\alpha\vec{x} = (\alpha x_1, \alpha x_2 \ldots \alpha x_n).$$

3. Zero Vector

It is represented as $(0, 0 \ldots 0)$.

All components has a value equal to 0.

denoted by 0.

4. Negative of a vector

Let $\vec{x} = (x_1, x_2 \ldots x_n)$.

Negative of a vector is denoted by $-\vec{x}$, defined by

$$-\vec{x} = (-x_1, -x_2, \ldots -x_n)$$

$$\vec{x} + (-\vec{y}) = \vec{x} - \vec{y}.$$

Properties

1) closure under addition: $\vec{x}+\vec{y}$ is also a n-dimensional vector
2) Commutativity $\vec{x}+\vec{y} = \vec{y}+\vec{x}$.
3) Associativity. $\vec{x}+(\vec{y}+\vec{z}) = (\vec{x}+\vec{y})+\vec{z}$
4) Existence of identity. $\vec{x}+0=\vec{x}$.

5) Existence of inverse

$$\vec{x} + (-\vec{x}) = 0.$$

Norm and Inner product

Norm

The norm of the n-dimensional vector $\vec{x} = (x_1, x_2, \dots x_n)$ denoted by

$$\|\vec{x}\|.$$

Defined by

$$\|\vec{x}\| = \sqrt{x_1^2 + x_2^2 + \dots x_n^2}.$$

Inner product

if 0 then ⊥r

The inner product of $\vec{x}$ and $\vec{y}$

$$\vec{x} = (x_1, x_2 \dots x_n)$$

$$\vec{y} = (y_1, y_2 \dots y_n).$$

Denoted as $\vec{x} \cdot \vec{y}$

Defined by.

$$\vec{x} \cdot \vec{y} = x_1 y_1 + x_2 y_2 + \dots x_n y_n.$$

Angle b/w vectors

$$\cos\theta = \frac{\vec{x} \cdot \vec{y}}{\|x\| \, \|y\|}$$

Perpendicularity

Two vectors $\vec{x} = (x_1, x_2 \dots x_n)$ and $\vec{y} = (y_1, y_2, \dots y_n)$ are said to be perpendicular if.

$$\vec{x} \cdot \vec{y} = 0.$$

Hyperplanes

✓ Hyperplanes are subsets of finite dimensional vector spaces.

Definition

Consider n-dimensional vector space R^n. The set of all vectors.

$$\vec{x} = (x_1, x_2 \dots x_n). \text{ In } R^n$$

satisfies the equation of form.

$$\alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \dots \alpha_n x_n = 0.$$

where $\alpha_0, \alpha_1, \dots \alpha_n$ - are called scala

Then it is called a Hyperplane.

Hyperplane divides the space R^n into two halfs.

one half

$$- \alpha_0 + \alpha_1 x_1 + \dots \alpha_n x_n > 0.$$

other half

$$- \alpha_0 + \alpha_1 x_1 + \dots \alpha_n x_n < 0.$$

Distance of hyperplane from a point

Perpendicular distance is computed

= we have a line $\alpha_0 + \alpha_1 x_1 + \cdots \alpha_2 x_2 = 0$.

point $P(x_1', y_1')$.

Distance

$$= \frac{|\alpha_0 + \alpha_1 x_1' + \alpha_2 x_2'|}{\sqrt{\alpha_1^2 + \alpha_2^2}}$$

Two class Data sets

In machine learning problem, variable being predicted is called output variable or target variable.

It is also called as dependent variable or response.

Two class data set is the target variable takes only two values.

If the the target variable takes more than two possible value, then it is called as multiclass data set.

Linearly seperable data

Consider a two class data set having n features and two class labels +1 and -1.

$\vec{x} = (x_1, x_2, \ldots x_n)$

Dataset is linearly seperable if the hyperplane has following properties.

$$\alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \cdots \alpha_n x_n = 0.$$

1) for each instance $\vec{x}$ with class label -1,

$$\alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \cdots \alpha_n x_n < 0.$$

2) for each instance $\vec{x}$ with class label +1

$$\alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n > 0.$$

Hyperplane having above properties are called seperating hyperplane for set.

Maximal Margin Hyperplanes

A linearly seperable data set with two class labels +1 and -1.

Calculate the perpendicular distance between seperating line and points.

Double of this smallest distance is called margin of seperating hyperplane H.

Hyperplane for which margin is largest is called maximal margin hyperplane or optimal seperating hyperplane.

Maximal margin seperating hyperplane is called support vector machine.

Data points that lie close to the maximal margin hyper plane are called support vectors.

Algorithm for SVM Classifier

Given a two class linearly seperable dataset of N points of the form

$$(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots (\vec{x}_N, y_N).$$

where y_i's can be either +1 or -1.

1) find $\vec{\alpha} = (\alpha_1, \alpha_2 \dots \alpha_n)$ which maximises

$$\phi(\vec{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1} \alpha_i \alpha_j y_i y_j (\vec{x}_i \cdot \vec{x}_j)$$

Subject to.

$$\sum_{i=1}^{N} \alpha_i y_i = 0.$$

$\alpha_i > 0$ for $i = 1, 2 \dots N$.

2) Compute $\vec{\omega} = (\sum_{i=1}^{N} \alpha_i y_i \vec{x}_i$

3) Compute $b = \frac{1}{2}(\min_{i, y_i = +1} (\omega \cdot \vec{x}_i) + \max_{i, y_i = -1} (\vec{\omega} \cdot \vec{x})$)

4) SVM classifier function is given by.

$$f(\vec{x}) = \vec{\omega} \cdot \vec{x} - b$$

α_i - non zero

v - Support vector

Soft Margin Hyperplanes

The algorithms for finding svm classifier works only when two class data set is linearly seperable.

But for not linearly seperable dataset, we uses soft margin hyperplane. Additional variable is used. It is called as slack variable ξ_i.

Slack variable stores deviations from margin. Two type of deviations are there:

a) may lie on wrong side of hyperplane. misclassified

b) may lie in the margin.

If $\xi_i = 0$, then $\vec{x}$ is correctly classified and no problem.

If $0 < \xi_i < 1$, then $\vec{x}$ is correctly classified, but it lies in the margin

If $\xi_i > 1$, - data is misclassified

The sum $\sum_{i=1}^{N} \xi_i$ is called soft error

Kernel functions

A kernel function is a function of the form $K(\vec{x}, \vec{y})$, where

$\vec{x}$ and $\vec{y}$ - n dimensional vectors having a special property.

These functions are used to classify not linearly seperable data.

Definition

Let $\vec{x}$ and $\vec{y}$ be arbitrary vectors in the n-dimensional vector space R^n. Let ϕ be a mapping from R^n to some vector space.

A function $K(\vec{x}, \vec{y})$ is called a kernel function. If there is a function ϕ such that.

$$K(\vec{x}, \vec{y}) = \phi(\vec{x}) \cdot \phi(\vec{y}).$$

Eg:

we have

$$K(\vec{x}, \vec{y}) = (\vec{x} \cdot \vec{y})^2$$

$$= (x_1 y_1 + x_2 y_2)^2.$$

$$= x_1^2 y_1^2 + 2x_1 x_2 y_1 y_2 + x_2^2 y_2^2.$$

Now

$$\phi(\vec{x}) = (x_1^2, \sqrt{2}x_1 x_2, x_2^2) \in R^3$$

$$\phi(\vec{y}) = (y_1^2, \sqrt{2}y_1 y_2, y_2^2) \in R^3.$$

$\phi(\vec{x}) \cdot \phi(\vec{y})$ ..

$$x_1^2 y_1^2 + 2x_1 x_2 y_1 y_2 + x_2^2 y_2^2$$

$$= K(\vec{x} \cdot \vec{y}).$$

So $K(\vec{x}, \vec{y})$ is a kernel function.

Important kernel functions

we have.

$\vec{x} = (x_1, x_2 \cdots x_n)$ and

$\vec{y} = (y_1, y_2 \cdots y_n)$ ..

1. Homogeneous polynomial kernel

$$K(\vec{x} \cdot \vec{y}) = (\vec{x} \cdot \vec{y})^d$$

where d is some positive integer.

2. Non-homogeneous polynomial kernel

$$K(\vec{x}, \vec{y}) = (\vec{x} \cdot \vec{y} + \theta)^d.$$

d - positive integer, θ - real constant.

3. Radial basis function (RBF) kernel

$$K(\vec{x}, \vec{y}) = e^{-\|x-y\|^2 / 2\sigma^2}.$$

4. Laplacian kernel function

$$K(\vec{x}, \vec{y}) = e^{-\|\vec{x}-\vec{y}\|/\sigma}$$

5. Hyperbolic tangent kernel function

Also called as Sigmoid kernel function.

$$K(\vec{x}, \vec{y}) = \tanh(\alpha(\vec{x}\cdot\vec{y}) + c).$$

$$K(\vec{x}, \vec{y}) = (\vec{x}\cdot\vec{y} + \theta)^2$$

Kernel Trick

$\phi(\vec{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2\theta}x_1, \sqrt{2\theta}x_2, \sqrt{\theta}) \in R^6$

Also called as kernel method.

Basic Idea

a) choose an appropriate kernel function $K(\vec{x}, \vec{y})$.

b) formulate and solve optimize problem obtained by replacing each inner product $\vec{x}\cdot\vec{y}$ by $K(\vec{x}, \vec{y})$.

c) formulation of classifier function for svm problem by using inner products of unclassified data $\vec{z}$ and input vector x.

Replace each inner product $\vec{z}\cdot\vec{x}_i$ by $K(\vec{z}, \vec{x}_i)$.

New classifier function is obtained

TNeFP - TN

TN / TN+FP

Algorithm

Given a two class linearly seperable data set of N points of form.

$(\vec{x}_1, y_1), (\vec{x}_2, y_2) \ldots (\vec{x}_N, y_N)$.

where y_i can be either +1 or -1.

There is a kernel function $K(\vec{x}, \vec{y})$.

1. find $\vec{\alpha} = (\alpha_1, \alpha_2 \ldots \alpha_N)$ which maximises

$$\sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(\vec{x}_i, \vec{x}_j)$$

Subject to. $\sum_{i=1}^{N} \alpha_i y_i = 0$.

for $\alpha_i > 0$, $i = 1, 2 \ldots N$.

2. Compute $\vec{\omega} = \sum_{i=1}^{N} \alpha_i y_i \vec{x}_i$.

3. Compute $b = \frac{1}{2}\left(\min_{i:y_i=+1} K(\vec{\omega}, \vec{x}_i) + \max_{i:y_i=-1} K(\ldots\right.$

4. Svm classifier function is given by

$$f(\vec{z}) = \sum_{i=1}^{N} \alpha_i y_i K(\vec{x}_i, \vec{z}) + b.$$

$(x_1y_1 + x_2y_2 + \theta)^2$

Hidden Markov models

one of the most important concept in machine learning.

- used in Speech and language processing.

Discrete Markov Process

Main concepts used in a markov process are:

1. System and states

System represents the main dataset or where operations are going to be done

states represents different values for the system.

for eg:

Let system be stock market.

States are:

S_1 : Bull market trend

S_2 : Bear market trend

S_3 : Stagnant market trend.

2. Transition Probability

The system can change from one state to another. The probabilities associated with these transitions are called transition probabilities.

3. Markov property

property states that state in week t+1 depends only on state in week t, regardless of previous weeks.

Representation of Transition probabilities

The different transition probabilities are given below. using state transition diagram

for eg. System - stock market.

States - 3 state

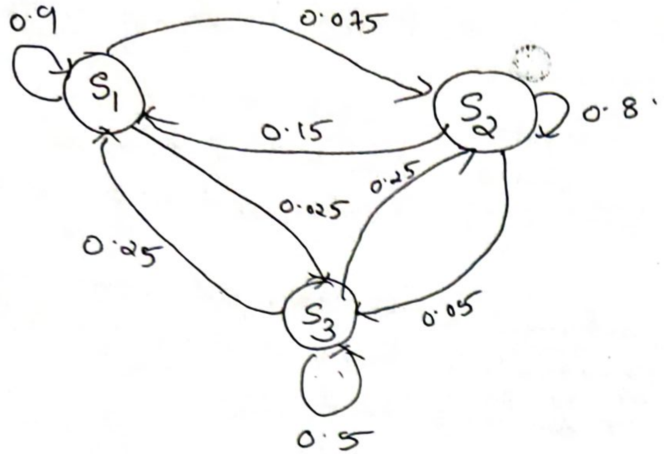
S_1 - Bull market

S_2 - Bear market

S_3 stagnant market

Transition probabilities

	S_1	S_2	S_3
S_1	0.9	0.075	0.025
S_2	0.15	0.8	0.05
S_3	0.25	0.25	0.5



State Transition probabilities can be represented in matrix form.

Matrix is called state transition matrix.

$$P = \begin{bmatrix} 0.90 & 0.075 & 0.025 \\ 0.15 & 0.80 & 0.05 \\ 0.25 & 0.50 & 0.25 \end{bmatrix}$$

Initial probabilities

The initial probababilities are the probabilities that the system is in initially.

Denoted by π_i

Represented as vector

$$\pi = \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{bmatrix}$$

Discrete Markov process

There will be a system and states with markov property and transition probabilities given by matrix P and initial probabilities by vector π constitutes a discrete markov process.

Probabilities for future states :

$$= \pi^T P.$$ Transpose

Discrete Markov Process: General Case

Markov process is a random process indexed by time t, by the property that future is independent of the past.

A markov process with a system and states $S_1, S_2 \dots S_N$ satisfies the markov property, it is called discrete markov process.

Hidden Markov model

Coin tossing example

Consider a room which is divided into two halfs by a curtain through one cannot see other half and what happening there.

Person A is sitting on one half and person B is sitting in the other half.

Person B is tossing the coin, but B will not tell anything about what he is doing. Person B only will announce the result.

Let typical sequence of announcements be:

$$O = o_1 \; o_2 \; o_3 \; \cdots \; o_T$$
$$= H\,H\,T\,H\,H\,T\,T\,T \,.. \; H$$

where

H - denotes head

T - denotes Tail.

Person A wants to create a mathematical model. Person A notes that B is announcing results based on some discrete Markov Process.

Then the markov process is called Hidden markov process. By curtain, rest of the world cannot see what is happening.

1) Let B has 2 biased coins and flipping it in some order.

System - state of coin

States - Each coin is a state for two states S_1 and S_2.

2) outcomes of the flip of coin are the observations. Represented by symbols H and T for heads and tails.

3) After flipping coin, one of the coin should be flipped next. Procedure for this is a random process. Transition from one state to another associated with transition probabilities.

Probability matrix A is used.

4) Since the coins are biased, there should be some probability for getting H or T. Called as observation probability.

5) There should be some steps for selecting first coin. specified by initial probability.

π.

Another example is Urn and Ball model, which works like as above.

Hidden markov model (HMM) is defined by:

$\lambda = (A, B, \pi)$

where π is the initial probability.

Three basic problems in HMM

Given General model of HMM, there are 3 basic problems, that must be solved for real-time applications.

1. Evaluation problem F/w B/w algo

Given the observation sequence

$O = o_1 \, o_2 \, \cdots \, o_T$

and a HMM model

$\lambda = (A, B, \pi)$

we want to compute $P(O/\lambda)$.

probability of observation sequence O given λ.

2. Finding state sequence problem Viterbi algo & posterior decoding

Given the observation sequence

$O = o_1 \, o_2 \, \cdots \, o_T$

and HMM model

$\lambda = (A, B, \pi)$

Here we find state sequence

$Q = q_1 \, q_2 \cdots q_T$

which has highest probability of generating O.

ie, To find Q such that it maximises probability $P(Q, |O, \lambda)$.

3. Learning model parameters Baum welch algo

Given a training set X

Here model is defined by

$\lambda = (A, B, \pi)$

Here we find λ that maximises the Probability of generating X.

ie, we find λ that maximises the Probability $P(X/\lambda)$.

Solutions for these problems are:

Problem 1 is solved by Forward-Backwards algorithms.

Problem 2 is solved by Viterbi algorithms and posterior decoding.

Problem 3 is solved by Baum-welch algorithms.

HMM can be used to recognize isolated words.

Combining Multiple learners

There are several algorithms for learning a task. But different algorithm produce different results.

Need for Combining many learners

1) Each learning algorithm carries a set of assumptions. This leads to error, if assumption not hold.

2) Learning is an ill-posed problem.

3) performance of learner can be tuned to higher accuracy.

4) Most accurate o/p can be produced.

Ways to acheive diversity

when many algorithms are combined, the individual algorithms in the collection are called base-learners.

Different ways for selecting base learners:-

1) use different learning algorithms
2) use same algorithm with different hyperparameters
3) use different representation of input object.
4) use different training sets to train.
5) multiexpert combination methods.
6) multistage combination methods.

Model Combination schemes

Different schemes are used:-

a) Voting
b) Bagging
c) Boosting

Voting

Simplest procedure for combining outcomes of several learning algorithms.

1. Binary classification problem

There are two class labels +1 and -1. Let there are L Base learners and a test instance x. Each learners will assign a label to x.

If class label is +ve 1, we say it votes for +1 and label +1 gets a vote.

no of votes is counted.

label which gets maximum votes is assigned to x.

2. Multi-class classification problem

Let there be labels $C_1, C_2 \ldots C_n$.
Let x be a test instance.
L- Base learners.

In this class label, which gets maximum no of votes is assigned to x.

3. Regression

There are L Base learners for predicting variable y.

$$\hat{y}_e = \omega_1 \hat{y}_1 + \omega_2 \hat{y}_2 + \ldots + \omega_L \hat{y}_L$$

weighted voting scheme

$$\omega_e = \frac{1}{L} \quad \text{for } j = 1, 2 \ldots L.$$

Bagging

It is a Voting method wh... by base learners are made different by training them over different traning sets.

Unstable algorithm - learning algorithm, if small changes in training set causes a large difference in output.

Algorithm such as decision tree and multilayer perceptrons are unstable.

Boosting

In this, we try to generate complementary base learners by training next learner on mistakes of previous learners.

makes weaker algorithms stronger.

Module - 6.

Unsupervised learning

one of the important learning methods is unsupervised learning. Here there is no training set.

Different methods are used. one of them is clustering.

Clustering

clustering or cluster analysis is the task of grouping objects based on a particular feature.

The group is called a cluster

Applications of clustering

Mainly used in:

1) exploratory data mining
2) machine learning
3) pattern recognition
4) image analysis
5) information retreival
6) bio informatics
7) data compression etc
8) Computer Graphics.

Examples

In many applications, clustering is used

a) used in optical character recognition
b) used in speech recognition.

optical character recognition means the digits and letters can be written in 2 ways
(i) american style (ii) European style
clustering is based on these styles.

k-means clustering

k-means clustering is one of the simplest unsupervised learning algorithm for solving the clustering problem.

The data can be classified into different clusters, say k clusters.

k points are arbitrarily choosen and called 'centre' of the clusters. Associate each point with this nearest centre.

Repeat this process until centre converges to a fixed point.

Algorithm

Notations

Each data point is a n-dimensional vector.

$\vec{x} = (x_1, x_2 \cdots x_n)$.

Distance b/w two data points is:-

$\vec{x} = (x_1, x_2, x_3 \cdots x_n)$.

and $\vec{y} = (y_1, y_2, y_3 \cdots y_n)$

Distance =

$$\|\vec{x} - \vec{y}\| = \sqrt{(x_1-y_1)^2 + \cdots + (x_n-y_n)^2}$$

Set of data points

$X = (\vec{x}_1, \vec{x}_2 \cdots \vec{x}_N)$

$V = (\vec{v}_1, \vec{v}_2 \cdots \vec{v}_k)$

Set V is called set of centres.

C_e - $e = 1, 2 \cdots k$, be no of data points in e^{th} cluster.

Basic idea:-

when algorithm aims to achieve partition X into k different clusters $S = (S_1, S_2 \cdots S_k)$ and set of points V, which minimises

$$\sum_{l=1}^{K} \sum_{x \in S_l} \| \vec{x} - \vec{v}_l \|^2$$

Algorithm

1) Randomly select K cluster centres $\vec{v}_1, \vec{v}_2 \dots \vec{v}_k$.

2) Calculate the distance b/w each data point $\vec{x_i}$ and each cluster center $\vec{v_j}$.

3) for each $j = 1, 2 \dots N$ assign data point x_j to cluster centre $\vec{v_l}$ for which the distance $\|x_j - v_l\|$ is minimum.

Let $\vec{x}_{l1}, \vec{x}_{l2}, x_{l3} \dots \vec{x}_{lk}$ be data points assigned to v_l.

4) Recalculate the cluster center using:-

$$\vec{V}_l = \frac{1}{C_l}\left(x_{l1} + x_{l2} + \dots + \vec{x}_{lC_l}\right)$$

for $l = 1, 2 \dots k$.

5) Recalculate distance b/w each data point and newly obtained centres.

6) If no data point was reassigned then stop. else go to step 3.

Methods for selecting K

1) Randomly take some K data points

2) Calculate the mean of all data and select K points.

3) Calculate principal component, divide the range into k equal intervals.

Disadvantages & Advantages

Main advantages are

1) fast
2) Robust
3) easy to understand

Disadvantages are:-

1) Requires apriori specification of cluster centre's.
2) depends on initial v_l
3) different results on different input data
4) May not produce desired output.
5) Algorithm cannot be applied to categorical data

Applications

a) image segmentation
b) Data compression.

Expectation - Maximisation Algorithm

Maximum likelihood estimation method (MLE) is a method for estimating the parameters of a statistical model.

The method attempts to find maximum likelihood function and its parameters. ie, log likelihood estimation.

Expectation - maximisation algorithm (EM algorithm) is used to find maximum likelihood estimates of parameters. This is used when equations cannot be solved directly.

Involves latent or unobserved values. EM algorithm is a general procedure to create algorithm for specific MLE problems.

Outline of Algorithm

1) Initialise the parameters θ to be estimated

2) Expectation step (E-step)

Take the expected value of complete data given the observation and current parameter estimate $\hat{\theta}_j$. This is a function of θ and $\hat{\theta}_j$

$= Q(\theta, \hat{\theta}_j)$

3. Maximization step (M-step)

find the values θ that maximizes the function $Q(\theta, \hat{\theta}_j)$.

4. Repeat step 1 and 2 until all parameters values or likelihood function converges.

Heirarchial clustering

Heirarchial clustering is also called as heirarchial cluster analysis or HCA is a method of cluster analysis which seeks to build a heirarchy.

The heirarchial clustering produces clusters in each level.

Decision of merging clusters is based on measure of disimmilarity. Mainly distance is taken as measure.

Dendograms

Heirarchial clustering is represented by a rooted binary tree. Nodes of tree represents group of clusters.

Root node represents entire dataset. Terminal nodes represent one of individual observations.

Each non terminal node has two daughter nodes.

Height of each node is directly proportional to value of distance b/w two daughter nodes.

Dendogram is a tree diagram used to illustrate arrangement of clusters.

Mainly used in Computational biology for clustering of genes or samples.

Methods for heirarchial clustering

Mainly two methods are used for heirarchial clustering

a) Agglomerative method (bottom-up method)

b) Divisive method (top-down method)

eg:. for a data (a, b, c, d, e).

Dendogram is.

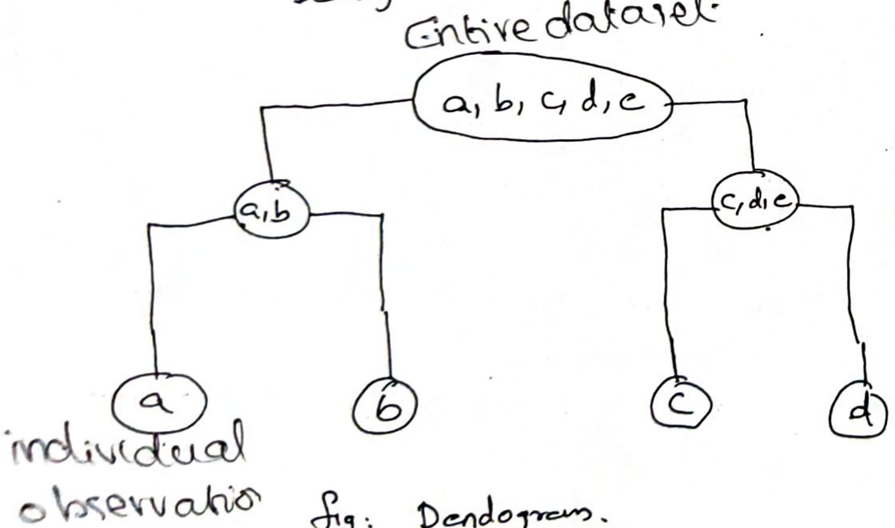
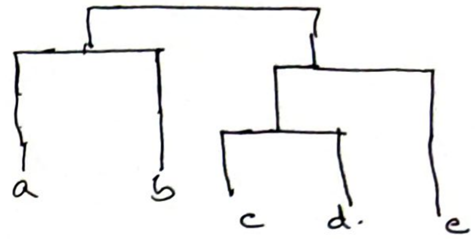


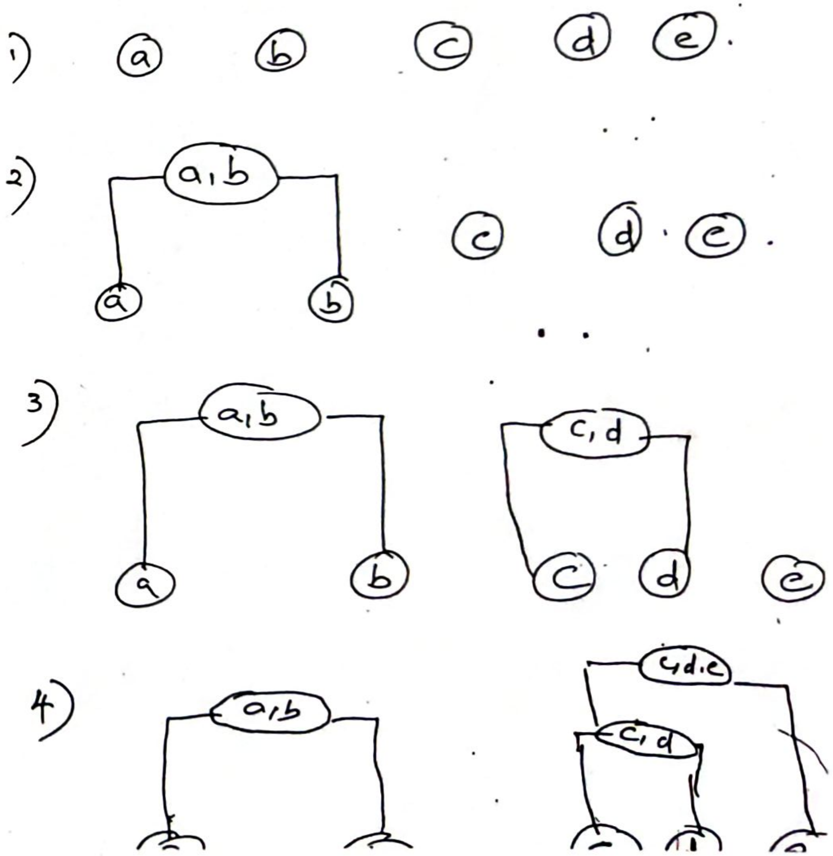
fig: Dendogram.

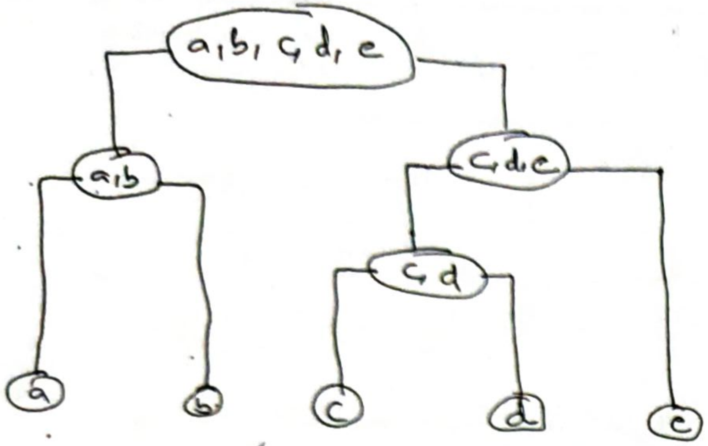
Different ways of drawing dendograms:-



Agglomerative method

In this, we start at bottom and merge a selected pair of data points into a cluster.



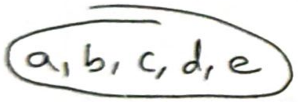


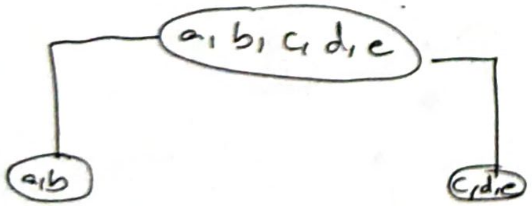
Divisive method

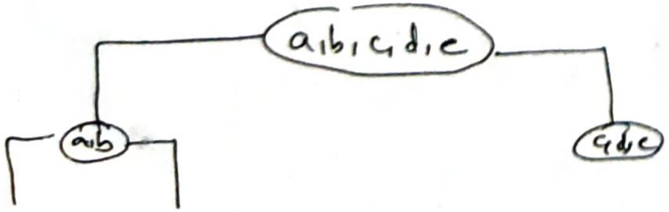
Top - down approach.

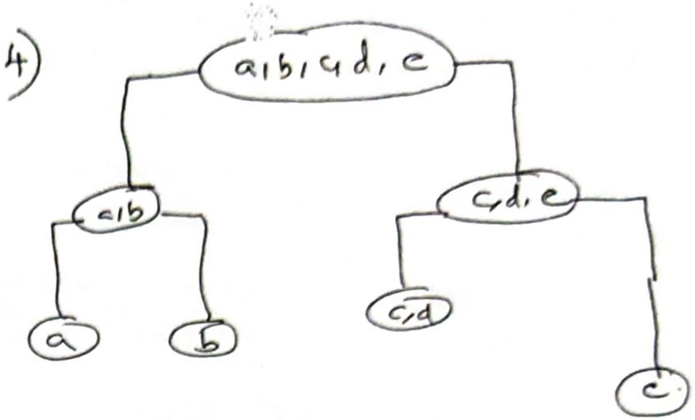
Starts at top end and split one of the existing clusters into new clusters.

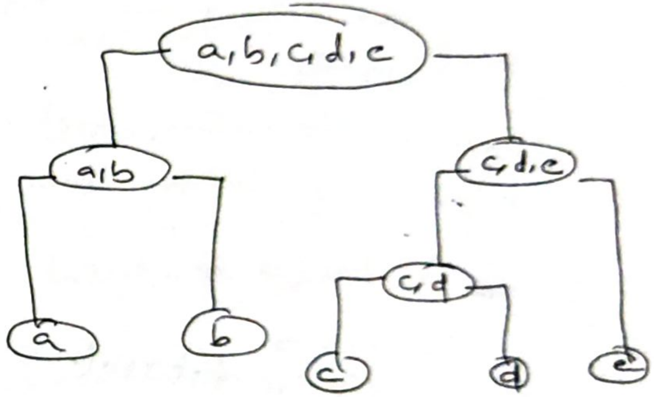
DIANA Algorithm is used.

1) 

2) 



4) 

5) 

Density Based clustering

In this, clusters are defined as areas of higher density.

Noise and Border points are considered.

Most popular density based algorithm is DBSCAN algorithm.

CONTENT BEYOND SYLLABUS

Ensemble Learning

Definition

Ensemble learning is a machine learning paradigm where multiple learners are trained to solve the same problem. In contrast to ordinary machine learning approaches which try to learn *one* hypothesis from training data, ensemble methods try to construct a *set* of hypotheses and combine them to use. An ensemble contains a number of learners which are usually called *base learners*. The **generalization** ability of an ensemble is usually much stronger than that of base learners. Actually, ensemble learning is appealing because that it is able to boost *weak learners* which are slightly better than random guess to *strong learners* which can make very accurate predictions. So, “base learners” are also referred as “weak learners”. Base learners are usually generated from training data by a *base learning algorithm* which can be decision tree, neural network or other kinds of machine learning algorithms. Most ensemble methods use a single base learning algorithm to produce *homogeneous* base learners, but there are also some methods which use multiple learning algorithms to produce *heterogeneous* learners. In the latter case there is no single base learning algorithm and thus, some people prefer calling the learners *individual learners* or *component learners* to “base learners”, while the names “individual learners” and “component learners” can also be used for homogeneous base learners.

Constructing Ensembles

Typically, an ensemble is constructed in two steps. First, a number of base learners are produced, which can be generated in a *parallel* style or in a *sequential* style where the generation of a base learner has influence on the generation of subsequent learners. Then, the base learners are combined to use, where among the most popular combination schemes are *majority voting* for classification and *weighted averaging* for regression. The **bias-variance decomposition** is often used in studying the performance of ensemble methods

Applications

Ensemble learning has already been used in diverse applications such as optical character recognition, text categorization, face recognition, computer-aided medical diagnosis, gene expression analysis, etc. Actually, ensemble learning can be used wherever machine learning techniques can be used.

Expert Systems

In artificial intelligence, an **expert system** is a computer system that emulates the decision-making ability of a human expert. Expert systems are designed to solve complex problems by reasoning through bodies of knowledge, represented mainly as if–then rules rather than through conventional procedural code. The first expert systems were created in the 1970s and then proliferated in the 1980s. Expert systems were among the first truly successful forms of artificial intelligence (AI) software. However, some experts point out that expert systems were not part of true artificial intelligence since they lack the ability to learn autonomously from external data.

An expert system is divided into two subsystems: the inference engine and the knowledge base. The knowledge base represents facts and rules. The inference engine applies the rules to the known facts to deduce new facts. Inference engines can also include explanation and debugging abilities.

Software architecture

An expert system is an example of a knowledge-based system. Expert systems were the first commercial systems to use a knowledge-based architecture. A knowledge-based system is essentially composed of two sub-systems: the knowledge base and the inference engine.

The knowledge base represents facts about the world. In early expert systems such as Mycin and Dendral, these facts were represented mainly as flat assertions about variables. In later expert systems developed with commercial shells, the knowledge base took on more structure and used concepts from object-oriented programming. The world was represented as classes, subclasses, and instances and assertions were replaced by values of object instances. The rules worked by querying and asserting values of the objects.

The inference engine is an automated reasoning system that evaluates the current state of the knowledge-base, applies relevant rules, and then asserts new knowledge into the knowledge base. The inference engine may also include abilities for explanation, so that it can explain to a user the chain of reasoning used to arrive at a particular conclusion by tracing back over the firing of rules that resulted in the assertion.

There are mainly two modes for an inference engine: forward chaining and backward chaining. The different approaches are dictated by whether the inference engine is being driven by the antecedent (left hand side) or the consequent (right hand side) of the rule. In forward chaining an antecedent fires and asserts the consequent. For example, consider the following rule:

A simple example of forward chaining would be to assert Man(Socrates) to the system and then trigger the inference engine. It would match R1 and assert Mortal(Socrates) into the knowledge base.

Backward chaining is a bit less straight forward. In backward chaining the system looks at possible conclusions and works backward to see if they might be true. So if the system was trying to determine if Mortal(Socrates) is true it would find R1 and query the knowledge base to see if Man(Socrates) is true. One of the early innovations of expert systems shells was to integrate inference engines with a user interface. This could be especially powerful with backward chaining. If the system needs to know a particular fact but doesn't, then it can simply generate an input screen and ask the user if the information is known. So in this example, it could use R1 to ask the user if Socrates was a Man and then use that new information accordingly.

Advantages

The goal of knowledge-based systems is to make the critical information required for the system to work explicit rather than implicit. In a traditional computer program the logic is embedded in code that can typically only be reviewed by an IT specialist. With an expert system the goal was to specify the rules in a format that was intuitive and easily understood, reviewed, and even edited by domain experts rather than IT experts. The benefits of this explicit knowledge representation were rapid development and ease of maintenance.

Ease of maintenance is the most obvious benefit. This was achieved in two ways. First, by removing the need to write conventional code, many of the normal problems that can be caused by even small changes to a system could be avoided with expert systems. Essentially, the logical flow of the program (at least at the highest level) was simply a given for the system, simply invoke the inference engine. This also was a reason for the second benefit: rapid prototyping. With an expert system shell it was possible to enter a few rules and have a prototype developed in days rather than the months or year typically associated with complex IT projects.