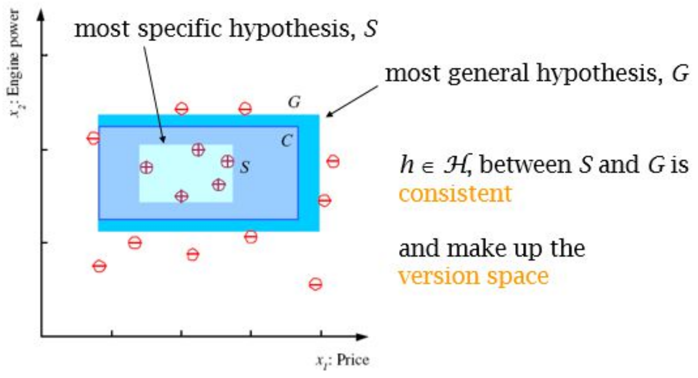
CS467 Machine Learning (Elective)

Question Answers

MODULE I

1. Relate AI, Machine learning and Data mining
   AI is engineering of making intelligent machines and programs. ML is not just a Database problem, it is also a part of AI. ML is a key enabler of AI.
   Machine learning is programming computers to optimize a performance criterion using example data or past experience.
   KDD/Data mining is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data"
   Application of Machine learning methods to large database is data mining.
2. What is hypothesis in machine learning? Using suitable example define the terms:
   i) Hypothesis class ii) Version space
   iii) Specific hypothesis iv) General hypothesis
   Hypothesis is a statement or a proposition declaring to explain a given set of facts or observations. It is the function generated by the learning algorithm for labelling the unseen samples.


most specific hypothesis, S
most general hypothesis, G
x_2: Engine power
G
C
S
$h \in \mathcal{H}$, between S and G is consistent
and make up the version space
x_1: Price

i). Hypothesis class is the set of possible classification functions/hypothesis you're considering/searching over; The learning algorithm picks a function from the hypothesis class. For a decision tree learner, the hypothesis class would just be the set of all possible decision trees.

ii) Version Space

Any h ∈ H between S and G is a valid hypothesis with no error and is said to be consistent with the training set, and such h make up the version space. The version space, V SH,D, with respect to hypothesis space H and training examples D, is the subset of hypotheses from H consistent with all training examples in D.

$$V\,S_{H,D} \equiv \{h \in H \mid Consistent(h, D)\}$$

iii) Specific Hypothesis

Tightest rectangles that includes all the possible examples and none of the negative examples.

Hypothesis h=S as the induced class, but actual class C may be larger than S

iv) General Hypothesis

The largest rectangle we can draw that includes all the positive examples and none of the negative examples.

3. Differentiate between supervised and unsupervised learning paradigms. Group the following problem on the basis of these categories.

a. Based on past information about spams, filtering out a new incoming email into Inbox (normal) or Junk folder (Spam).
b. Train your handwriting to OCR system and once trained, it will be able to convert your hand-writing images into text
c. Hotel booking company need to groups its housing listings into neighborhoods so that users can navigate listings more easily
d. Placement cell officer needs to group the students to two groups – Will be placed and will not be placed.

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. In supervised learning, each example in the training set is a pair consisting of an input object (typically a vector) and an output value. A supervised learning algorithm analyzes the training data and produces a function, which can be used for mapping new examples.

Unsupervised learning

Learning “what normally happens” i.e. finding regularities in input whose class label is unknown.Algorithm used to draw inferences from datasets consisting of input data without labeled responses. Clustering is the most common unsupervised learning method: which groups similar instances.

Groups a, b, d belong to supervised category and group c belong to unsupervised category

4. Compare and contrast Classification and regression. Identify two applications of classification and regression.

   Both regression and classification are supervised learning problems where there is an input X, output Y and the task is to learn mapping from input to output.

   Output Y is

   - a number in regression
   - a class code in the case of classification.

   Classification: Identification of class to which a data belongs.

   Regression/Prediction: predict a new value

   Classification is the problem of identifying to which of a set of categories a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known.

   Classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data.

   Example: Credit scoring, Medical diagnosis, predict the category of a car

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

   The value of the output variable may be a number, such as an integer or a floating point value.

   Example: Predicting the Price of a used car

5. Give examples for different types of Learning

   i) Association Analysis: ML method for discovering interesting relations, called “association rules”, between variables in large databases using some measures of “interestingness”

   There are several algorithms for generating association rules. Some of the well-known algorithms are listed below:

   Apriori algorithm, Eclat algorithm, FP-Growth Algorithm.

   ii) Supervised Learning :

   Training data includes desired outputs.

   a) Classification: Identification of class to which a data belongs.

   After training with the past data, a classification rule is learned.Once we have a rule that fits the past data, if the future is similar to the past, then we can make correct predictions for novel instances.There are several machine learning algorithms for classification. The following are some of the well-known algorithms.

Naive Bayes algorithm, k-NN algorithm, Decision tree algorithm

b) Regression/Prediction: predict a new value

These are often quantities, such as amounts and sizes. The input variables may be discrete or real-valued.

iii) Unsupervised Learning :

Learning “what normally happens” i.e. finding regularities in input whose class label is unknown.Algorithm used to draw inferences from datasets consisting of input data without labeled responses. That is training data does not include desired outputs. This method create an internal representation of the input e.g. form clusters; extract features.

Clustering (common unsupervised learning method): Grouping similar instances

Other applications: Summarization, Association Analysis

iv) Reinforcement Learning :

Reinforcement learning is the problem of getting an agent to act in the world so as to maximize its rewards. Agent acting in an environment and figure out what actions the agent must take at every step and action based on rewards or penalties agent gets in different states.A learner (program) is not told what actions to take, but instead must discover which actions yield most reward by trying them. In most interesting and challenging cases, actions may affect not only immediate reward but also next situations and, through that, all subsequent rewards.

Output of the system is a sequence of actions:

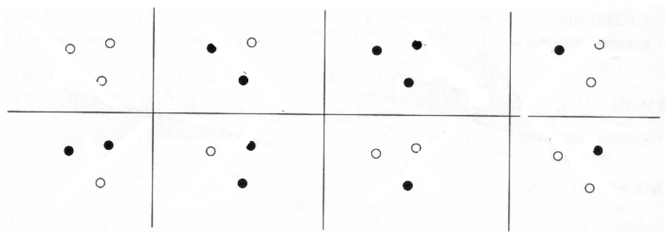
- Policies: Sequence of correct action to reach the goal; what actions should an agent take in a particular situation
- Utility estimation: how good is a state (• used by policy); ML program should be able to assess goodness of policies

No supervised output but delayed reward (Rewards from sequence of actions)

Applications:

- Game playing: a sequence of right moves
- Robot navigation; search of goal location

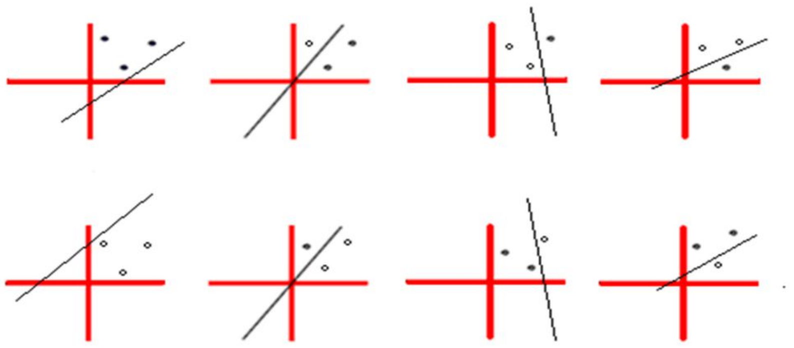
6. Explain VC dimension on the basis of the following data points. The data points can take positive and negative values. The dark dots denote the positive value and the light one denote the negative values.



H shatters N if there exists h Є H consistent for any of these problems. Maximum no. of points that can be shattered by H is called Vapnik-Chervonenkis dimension, VC dimension VC(H).The V C(H), of hypothesis space H defined over instance space X is the size of the largest finite subset of X shattered by H.

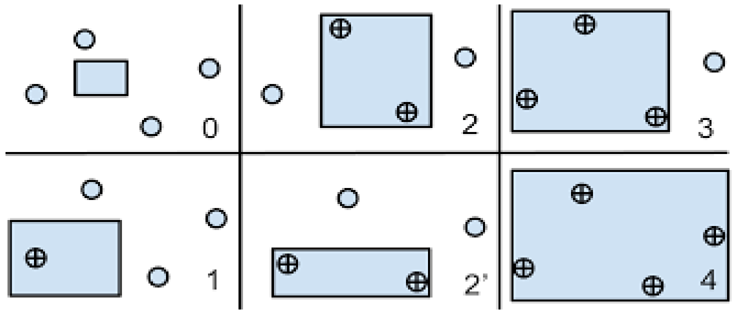
For a given decision function (classification function), the VC dimension indicates the maximum number of training sets that can be classified without any error.

If there are three points in the same two-dimensional space, it can be shattered with line function machine into eight (2^3) combination without error during training in the following manner.



In a two dimensional space, the maximum number of training points that can be shattered by the line function is 3.Hence for line function VC dimension h =3.

7. Show that axis aligned rectangle can shatter 4 points in 2 dimension space.



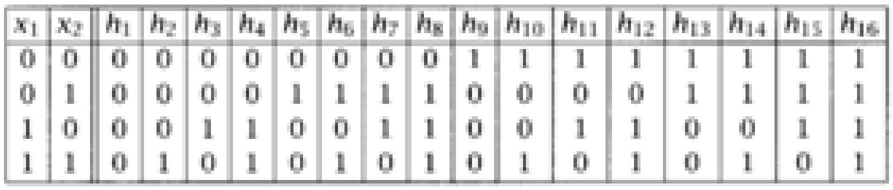
The VC dimension of rectangles is 4 because **there exists** a set of 4 points that can be shattered by a rectangle and any set of 5 points can not be shattered by a rectangle. So, while it's true that a rectangle cannot shatter a set of four collinear points with alternate positive and negative, the VC-dimension is still 4 because there exists one configuration of 4 points which can be shattered.

8. An open interval in R is defined as(a,b) = {x€R | a<x<b}. It has two parametersa and b. Show that the set of all open intervals has a VC dimension of 2

MODULE II

9. In a data set the attributes are denoted by X_1 and X_2. It can take only the values 0 and 1. The function *hi* denotes the different hypothesis class. The conditions are as follows. Find which all hypothesizes will be selected based on the conditions.

X_1	X_2	Result
0	0	1
0	1	0
1	0	0
1	1	1



x_1	x_2	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8	h_9	h_{10}	h_{11}	h_{12}	h_{13}	h_{14}	h_{15}	h_{16}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Step-1 : case 1 selects h9-h16 and eliminates h1-h8

Step-2: case 2 selects h9-h12 and eliminates h13-h16

Step-3: case 3: selects h9 and h10

Final case selects h10

10. Explain Probably Approximately Correct Learning (PAC).

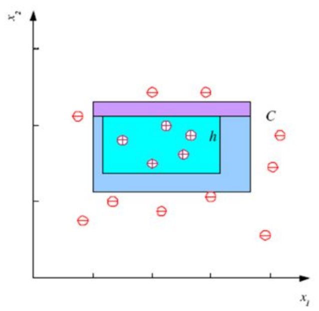
Probably approximately correct learning (PAC learning) is a frame work for mathematical analysis of machine learning algorithms.In this framework, the learner (that is, the algorithm) receives samples and must select a hypothesis from a certain class of hypotheses. The goal is that, with high probability (the "probably" part), the selected hypothesis will have low generalization error (the "approximately correct" part).

How many training examples N should we have, such that with probability at least **1 – δ**, hypothesis ***h*** has error at most **ε**? Where $\delta \leq 1/2$ and $\varepsilon > 0$

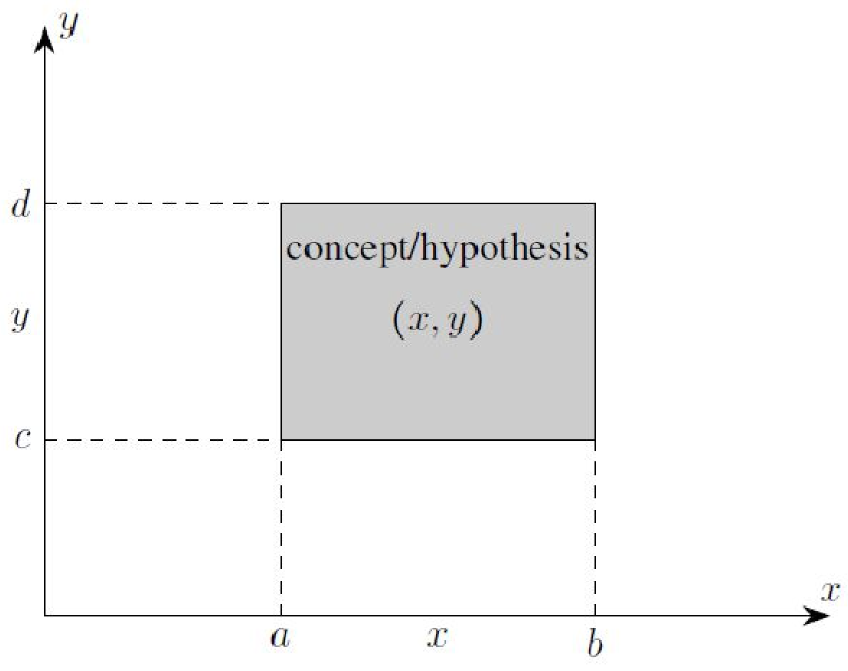
Definition of PAC learnability contains two approximation parameter; accuracy parameter **ε,** determines how far the output classifier can be from optimal one (approximately), and confidence parameter **δ** indicating how likely the classifier is to meet that accuracy requirement (probably).In PAC given a class C and examples drawn from some unknown but fixed probability distribution, to find the number of examples, N with confidence probability at least 1-δ .

S is the tightest rectangle, error region between ***C*** and ***h***=***S*** is the sum of four rectangular strips. Each strip is at most **ε/4**, Probability that we miss a strip is **1–ε/4**. Probability that N instances miss 4 strips **4(1 – ε/4)N**

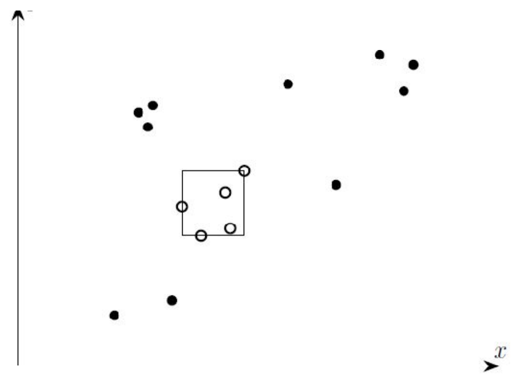
Take at least **(4/ε)log(4/δ)** independent examples from ***C*** and use the tightest rectangle as our hypothesis a given point will be misclassified with error probability at most **ε.**



11. Let X=R and C be the set of all possible rectangles in two dimensional plane which are axis aligned (not rotated). Show that this concept class is PAC learnable.



- Let the instance space be the set X of all points in the Euclidean plane. Each point is representedby its coordinates (x; y). So, the dimension or length of the instances is 2.
- Let the concept class C be the set of all "axis-aligned rectangles" in the plane; that is, the setof all rectangles whose sides are parallel to the coordinate axes in the plane (see Figure above).
- Since an axis-aligned rectangle can be defined by a set of inequalities of the following formhaving four parametersa <=x <=b; c <=y<=dthe size of a concept is 4.
- We take the set H of all hypotheses to be equal to the set C of concepts, H = C.
- Given a set of sample points labeled positive or negative, let L be the algorithm which outputsthe hypothesis defined by the axis-aligned rectangle which gives the tightest fit to the positiveexamples (that is, that rectangle with the smallest area that includes all of the positiveexamples and none of the negative examples)



Axis-aligned rectangle which gives the tightest fit to the positive examples

It can be shown that, in the notations introduced above, the concept class C is PAC-learnable bythe algorithm L using the hypothesis space H of all axis-aligned rectangles.

12. What are the factors influence generalization performancein a model? Describe how we can achieve best generalization in the prepared model.

Generalization indicates how well a model performs on new data. For best generalization we should match the complexity of the hypothesis class with complexity of function underlying the data.

We can choose the generalization performance by controlling the model complexity.For good generalization performance, high model complexity is not required, can be low enough to fit the data.

Two factors influence the generalization performance:

Overfitting

A complex model with insufficient data results overfitting. H more complex than C or f.

Example: Fitting a sixth order polynomial to noisy data sampled from a third order polynomial

A hypothesis h is said to overfit the training data if there is another hypothesis, h', such that h has smaller error than h' on the training data but h has larger error on the test data than h'.

Underfitting

Underfitting is the production of a machine learning model that is not complex enough to accurately capture relationships between a dataset features and a target variable. A simple model with large training set will result underfitting

Hypothesis H less complex than C or f. When model is too simple, both training and test errors are large.

Example:

Trying to fit a line to data sampled from a third order polynomial.

13. Describe the two dimensionality reduction techniques.

i) Featureselection

In feature selection, we are interested in finding k of the total of n features that give us the mostinformationanddiscardtheother(n−k)dimensions. Two popular feature selection approaches.

a) Forward Selection

Inforwardselection,startwithnovariablesandaddthemonebyone,ateachstepaddingtheone thatdecreases the errorthemost, untilany further additiondoesnot decrease the error (ordecreases it only slightly).

b) Backward Selection

Insequentialbackwardselection,startwiththesetcontainingallfeaturesandateachstepremove the one feature that causes the least error.

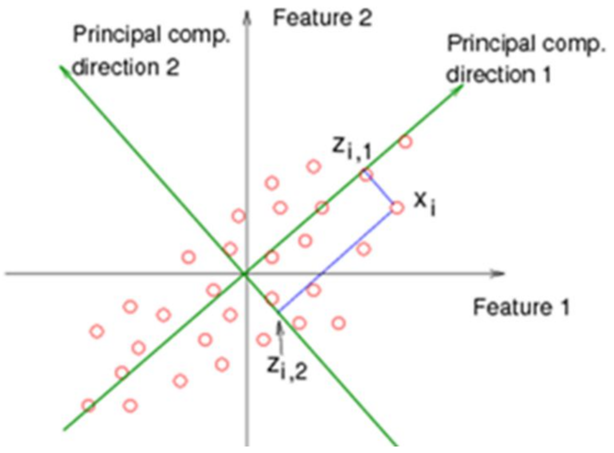
ii) Featureextraction

Infeatureextraction,weareinterestedinfindinganewsetof k featuresthatarethecombination of the original n features. These methods may be supervised or unsupervised depending on whether or not they use the output information. The best known and most widely used feature extraction methods are Principal Components Analysis (PCA) and Linear Discriminant Analysis (LDA), which are both linear projection methods, unsupervised and supervised respectively.

14. Explain Principal Components Analysis (PCA).

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the smaller of the number of original variables or the number of observations. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible),and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components.

Reconstruction error should be minimum



Choose directions such that a total variance of data will be maximum.

- Maximize Total Variance
- Choose directions that are orthogonal
- Minimize correlation

- Find a low-dimensional space such that when x is projected there, information loss is minimized.
- The projection of x on the direction of w is: $z = w^T x$

Find w such that Var(z) is maximized.

PCA algorithm

Step1. Data

Consider a dataset having n features or variables denoted by X1,X2,...,Xn. Let there be N examples.

Step2. Compute the means of the variables

Step3. Calculate the covariance matrix

Step4. Calculate the eigen values and eigen vectors of the covariance matrix

Step5. Derive new dataset

Order the eigenvalues from highest to lowest.The unit eigenvector corresponding to the largest eigenvalue is the first principal component.The unit eigenvector corresponding to the next highest eigenvalue is the second principal component, and so on.Choose the eigenvectors corresponding to the eigenvalues λ1, λ2, ...,λp and form ap×n matrix (we write the eigenvectors as row vectors):. This gives us a dataset of N samples having pfeatures Then compute matrix X_{new}which 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.

15. Describe the commonly used methods for generating multiple training-validation sets from a given dataset.

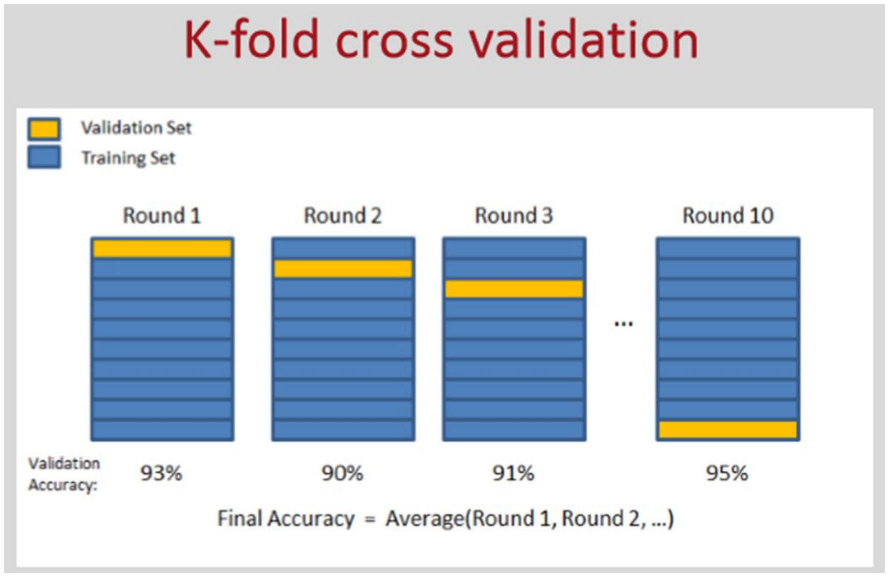
i) Cross validation

To test the performance of a classifier, we need to have a number of training/validation set pairs from a dataset X. Cross-validationisatechniquetoevaluatepredictivemodelsbypartitioningtheoriginalsample into a training set to train the model, and a test set to evaluate it.

The holdout method is the simplest kind of cross validation. The data set is separated into two sets, called the training set and the testing set. The algorithm fits a function using the training set only. Thenthefunctionisusedtopredicttheoutputvaluesforthedatainthetestingset(ithasne ver seen these output values before). The errors it makes are used to evaluate the model. This method is mainly used when the data set D is large.

ii) K-fold cross-validation

In K-fold cross-validation, the dataset X is divided randomly into K equal-sized parts, Xi, i= 1,...,K. Togenerateeachpair,wekeeponeoftheK partsoutasthevalidationsetVi,andcombine the remaining K−1 parts to form the training set Ti. Doing this K times, each time leaving out another one of the K parts out, we get K pairs(Vi,Ti): Then compute the average test set score of the k rounds.



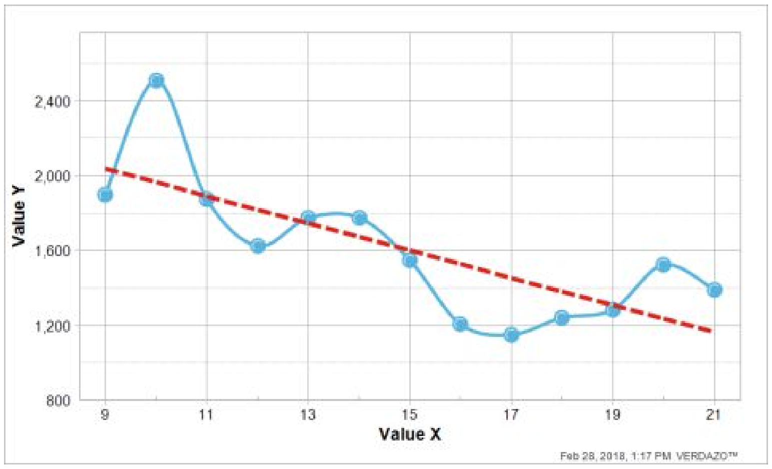
iii) Leave-one-outcross-validation

An extreme case of K-fold cross-validation is leave-one-out where given a dataset of N instances, only one instance is left out as the validation set and training uses the remaining N−1 instances. We then get N separate pairs by leaving out a different instance at each iteration. This is typically used in applications such as medical diagnosis, where labeled data is hard to find.

iv) Bootstrapping in machine learning

The term bootstrap sampling refers to process of "random sampling with replacement".Inmachinelearning,bootstrappingistheprocessofcomputingperform ancemeasuresusingseveral randomly selected training and test datasets which are selected through a process of sampling with replacement, that is, through bootstrapping. Sample datasets are selected multiple times. The bootstrap procedure will create one or more new training datasets some of which are repeated. The corresponding test datasets are then constructed from the set of examples that were not selected for the respective training datasets.

16. What is noise in data? Explain how it effects hypothesizes on the basis of the following graph.



Noise is any unwanted anomaly in the data. Noise distorts data. When there is noise in data, learning problems may not produce accurate result.

Due to noise the class may be more difficult to learn.Zero error may be infeasible with a simple hypothesis class. There may be imprecision in recording input attributes, which may shift data points in the input space.

There may be error in labelling data points, called teacher noise.

For example, in a binary classification problem with two variables, when there is noise, there may not be a simple boundary between the positive and negative instances and to separate them. So, when there is noise, we may make a complex model which makes a perfect fit to the data and attain zero error; or, we may use a simple model and allow some error.

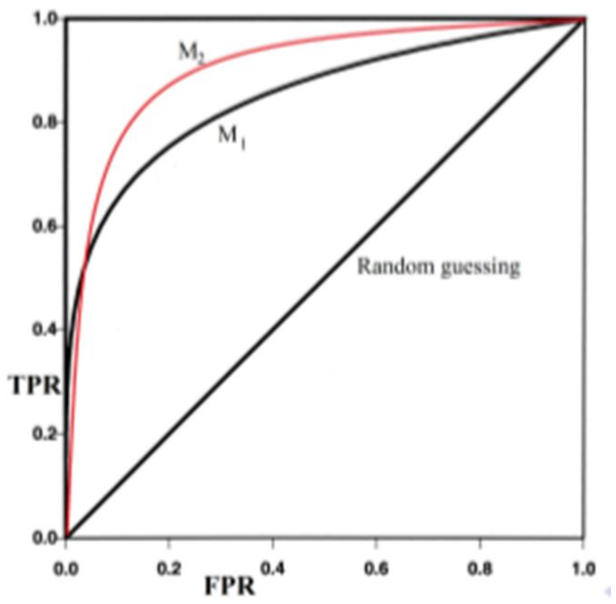
MODULE III

17. Explain the significance of ROC with a diagram.

It is a plot of the true positive rate (TPR) against the false positive rate (FPR).

TPR is the sensitivity also called recall: Fraction of positive examples correctly classified. FPR is the fraction of negative examples incorrectly classified. Each prediction result or instance of a confusionmatrix represents one point in the ROC space. The position of the point (FPR,TPR)in the ROC space gives an indication of the performance of the classifier. The closer the ROC curve is to the top left corner (0,1) of the ROC space, the better the accuracy of the classifier.

The measure of the area under the ROC curve is denoted by the acronym AUC. The value of AUC is a measure of the performance of a classifier. For the perfect classifier, AUC=1.0.



Special points in ROC space

1. The left bottom corner point (0,0): Always negative prediction

   For this point TP =0 and FP =0. It always makes negative predictions. All positive instances are wrongly predicted and all negative instances are correctly predicted.

2. The right top corner point (1,1): Always positive prediction

   For this point FN=0 and TN=0. All positive instances are correctly predicted and all negative instances are wrongly predicted.

3. The left top corner point (0,1): Perfect prediction

   It produces no false positives and no false negatives.

4. Points along the diagonal: Random performance

   Class labels are randomly guessed points in the ROC space will be lying very near the diagonal line joining the points(0,0)and(1,1).

15. a) State Bayes theorem.

Goal: To determine the most probable hypothesis, given the data D plus any initial knowledge about the prior probabilities of the various hypotheses in H.

Bayes' theorem is named after Thomas Bayes.P(H|X), probability that hypothesis H holds given observed data tuple X.

Posterior probability of a hypothesis H on X, P(H|X) follows the Baye's theorem

$$P(H \mid X) = \frac{P(X \mid H)P(H)}{P(X)}$$

P(X|H) is likelihood of X conditioned on H. P(H) is prior probability of hypothesis H and P(X) is the prior probability of X (evidence).

b) Consider a set of patients coming for treatment in a certain clinic. Let A denote the event that a "Patient has liver disease" and B the event that a "Patient is an alcoholic." It is known from experience that 10% of the patients entering the clinic have liver disease and 5% of the patients are alcoholics. Also, among those patients diagnosed with liver disease, 7% are alcoholics. Given that a patient is alcoholic, what is the probability that he will have liver disease? Use Bayes theorem.

P(A)=10%=0.10

P(B)=5%=0.05

P (B|A)=7%=0.07

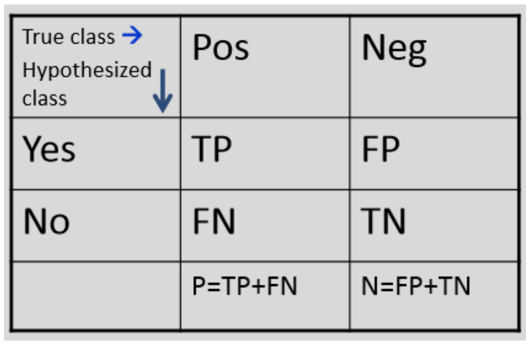
Applying Bayes Theorem

P (A|B)= {P(B|A)P(A)}/ P(B)

= (0.07×0.10)/ 0.05= 0.14

16. Describe confusion Matrix

A confusion matrix is used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. A confusion matrix is a table that categorizes predictions according to whether they match the actual value.Foratwo-classdataset,aconfusionmatrixisatablewithtworowsandtwocolumnsthatreportsthe number of false positives, false negatives, true positives, and true negatives. Table shows the confusion matrix of two class dataset which records the four parameters TP, FP, TN, FN measured from the classifier which can be used to calculate the various classifier performance measures.



True class → Hypothesized class ↓	Pos	Neg
Yes	TP	FP
No	FN	TN
	P=TP+FN	N=FP+TN

TP- True positive: It is number of correctly classified positive samples.

TN- True negative :It is number of correctly classified negative samples

FP- False positive: It is number of negative samplesincorrectly classified as positive samples

FN: False negative: It is number of positive samples incorrectly classified negative samples

17 a) Explain the various classifier performance measures (accuracy, precision, recall, sensitivity, specificity, ROC curve).

1. Accuracy = (TP +TN)/(TP +TN +FP +FN)

2. Sensitivity = recall = TP /(TP +FN)

3. Precision= TP/(TP +FP)

4. Specificity = TN /(TN +FP)

Precision p is the number of correctly classified positive examples (TP) divided by the total number of examples that are classified as positive.

precision = TP/(TP + FP)

Recall r is the number of correctly classified positive examples divided by the total number of actual positive examples in the test set.

recall = TP/(TP + FN)

5. ROC curve (see question no. 1)

b) Suppose a computer program for recognizing dogs in photographs identifies eight dogs in a picture containing 12 dogs and some cats. Of the eight dogs identified, five actually are dogs while the rest are cats. Compute the precision and recall of the computer program.

Total pictures = 12 dogs + some cat (nos. unknown)

No. of pictures identified as dog = 8

No. of dogs identified as dog (TP) = 5

No. of cats identified as dog = FP = 8-5 = 3

No. of dogs identified as cats = FN = 12-5 = 7

Precision P = TP/ (TP+FP) = 5/(5+3) = 0.625

Recall R = TP /(TP+FN) = 5/(5+7) = 0.4165

18. Explain naive Bayes algorithm (Bayesian classifier)

Bayesian classifiers are statistical classifiers. They can predict class membership probabilities, such as the probability that a given tuple belongs to a particular class. Bayesian classification is based on "Bayes" theorem. The neural network classifiers are suitable for larger data bases only and take long time to train.Bayesian classifiers are more accurate and much faster to train and suitable for low, medium and large sized data base. But they are slower when applied to new instances.

Naïve Bayesian algorithm steps

1. Let D be a training set of tuples and their associated class labels, and each tuple is represented by n-dimensional attribute vector X = (x1, x2, …, xn)
2. Suppose there are m classes C1, C2, …, Cm.

The classifier will predict that X belongs to class having highest posterior probability, conditioned on X.

Classification is to derive the maximum posteriori, i.e., the maximal P(Ci|X)

This can be derived from Bayes' theorem $P(C_i|\mathbf{X}) = \frac{P(\mathbf{X}|C_i)P(C_i)}{P(\mathbf{X})}$

3. Since P(X) is constant for all classes, only

$$P(C_i|\mathbf{X}) = P(\mathbf{X}|C_i)P(C_i)$$

needs to be maximized.

P(Ci)=|Ci,D|/|D|, where |Ci,D| is number of training tuples of class Ci in D.

4. A simplified assumption: attributes are conditionally independent (i.e., no dependence relation between attributes):

$$P(\mathbf{X}|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \ldots \times P(x_n|C_i)$$

This greatly reduces the computation cost: Only counts the class distribution

a) If A_k is categorical, $P(x_k|Ci)$ is the # of tuples in Ci having value x_k for A_k divided by |Ci, D| (# of tuples of Ci in D)

b) If A_k is continous-valued, $P(x_k|Ci)$ is usually computed based on Gaussian distribution with a mean μ and standard deviation σ

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

and P(x_k|Ci) is $P(\mathbf{X}|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$

5. To predict the class label of X, P(X|Ci)P(Ci) is evaluated for each class Ci.

The classifier predicts that the class label of tuple X is the class Ci if and only if P(X|Ci)P(Ci) > P(X|Cj)P(Cj) for $1 \le j \le m$, $j \ne i$.

In other words, the predicted class label is the class Ci for which P(X|Ci)P(Ci) is the maximum.

19. Explain the general MLE method for estimating the parameters of a probability distribution.

Maximum likelihood estimation (MLE) is particular method to estimate the parameters of a probability distribution.To develop a Bayesian classifier, we need to estimate the probabilities P(x|ck) for the class labels c1,...,ck from the given data.

If computed probabilities are good approximations of true probabilities, then there would be an underlying probability distribution.Suppose the underlying distribution has a particular form, say binomial, Poisson or normal and are defined by probability density functions.There are parameters which define these functions, and these parameters are to be estimated to test whether a given data follow some particular distribution

MLE attempts to find the parameter values that maximize the likelihood function, given the observations. The resulting estimate is called a maximum likelihood estimate, which is also abbreviated as MLE.

Suppose we have a random sample X = {x1,...,xn} taken from a probability distribution having the probability density function p(x|θ), where x denotes a value of the random variable and θ denotes the set of parameters that appear in the function.

The likelihood of sample X is a function of the parameter θ and is defined as

$l(\theta)=p(x_1|\theta)p(x_2|\theta)...p(x_n|\theta)$

In ML estimation, find the value of θ that makes the value of the likelihood function maximum.

For computation convenience, define the log likelihood function as the logarithm of the likelihood function:

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

$= \log p(x_1|\theta)+\log p(x_2|\theta)+\cdots+\log p(x_n|\theta)$

Hence, in maximum likelihood estimation, we find θ that maximizes the log likelihood function.

20. Describe the various density functions.

1. Bernoulli density

In a Bernoulli distribution there are two outcomes: An event occurs or it does not,

For example, an instance is a positive example of the class, or it is not.

The event occurs and the Bernoulli random variable X takes the value 1 with probability p,

The nonoccurrence of the event has probability 1−p and this is denoted by X taking the value 0. The probability function of X is given by

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

Consider a random sample X={x_1,...,xn} taken from a Bernoulli distribution with probability function f(x|p).

The log likelihood function is

$L(p)=\log f(x_1|p)+\cdots+\log f(x_n|p)$

$=\log {}^{px1}(1-p)^{1-x1}+\cdots+\log p^{xn}(1-p)^{1-xn}$

$=[x_1 \log p+(1-x_1)\log(1-p)]+\cdots+[x_n \log p+(1-x_n)\log(1-p)]$

To find the value of p that maximizes L(p)we set up the equation

$$dL/ dp=0,$$

that is,

$$\left[\frac{x_1}{p}-\frac{1-x_1}{1-p}\right]+\cdots+\left[\frac{x_n}{p}-\frac{1-x_n}{1-p}\right]=0.$$

Solving this equation, we have maximum likelihood estimate of p as

$$\hat{}\ p=1/n\,(x1+\cdots+xn).$$

2. Multi nominal Density

Suppose that the outcome of a random event is one of K classes, each of which has a probability of occurring pi with $p_1+\cdots+p_K=1$.

We represent each outcome by an ordered K-tuple $x=(x_1,...,x_K)$ where exactly one of $x_1,...,x_K$ is 1 and all others are 0. xi=1 if the outcome in the i-th class occurs. The probability function can be expressed as

$$f(x|p, ...,p_K)=p_1^{x1} ...p_K^{xK} .$$

Here, $p_1,...,p_K$ are the parameters.

We choose n random samples. The i-th sample may be represented by

$$xi=(x_{1i},...,x_{Ki}).$$

The values of the parameters that maximizes the likelihood function can be shown to be

$$\hat{p}_k = \frac{1}{n}(x_{k1} + x_{k2} + \cdots + x_{kn})$$

3. Gaussian (normal) Density

A continuous random variable X has the Gaussian or normal distribution if its density function is

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

Here μ and σ are the parameters.

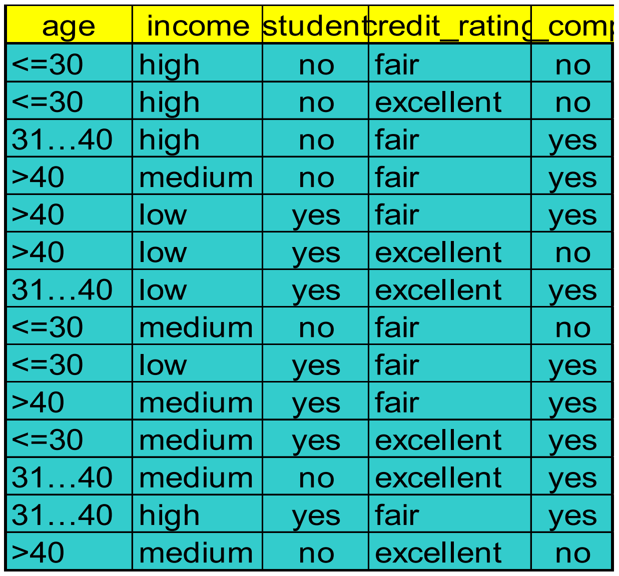
Given a sample $x_1,x_2,...,xn$ from the distribution. the log likelihood function is

$$L(\mu, \sigma) = -\frac{n}{2}\log(2\pi) - n\log\sigma - \frac{1}{2\sigma^2}\left[(x_1-\mu)^2 + \cdots + (x_n-\mu)^2\right]$$

Setting up the equations dL/dμ=0, dL/dσ=0and solving for μ and σ we get the maximum likelihood estimates of μ and σ as

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

21. Apply Naive Bayes classification in the given dataset and determine the class of the given sample. Given a new instance, X = (age <=30,Income = medium,Student = yes, Credit_rating = Fair)



age	income	student	credit_rating	comp
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

P(Ci): P(buys_computer = "yes") = 9/14 = 0.643

P(buys_computer = "no") = 5/14= 0.357

Compute P(X|Ci) for each class

P(age = "<=30" | buys_computer = "yes") = 2/9 = 0.222

P(age = "<= 30" | buys_computer = "no") = 3/5 = 0.6

P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444

P(income = "medium" | buys_computer = "no") = 2/5 = 0.4

P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667

P(student = "yes" | buys_computer = "no") = 1/5 = 0.2

P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667

P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4

X = (age <= 30 , income = medium, student = yes, credit_rating = fair)

P(X|Ci) : P(X|buys_computer = "yes") = 0.222 x 0.444 x 0.667 x 0.667 = 0.044

P(X|buys_computer = "no") = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

P(X|Ci)*P(Ci) : P(X|buys_computer = "yes") * P(buys_computer = "yes") = 0.028

P(X|buys_computer = "no") * P(buys_computer = "no") = 0.007

Therefore, X belongs to class ("buys_computer = yes")

22. What are the different types of regression.? Explain each

A regression problem is the problem of determining a relation between one or more independent variables and an output variable which is a real continuous variable, given a set of observed values of the set of independent variables and the corresponding values of the output variable.

Example

Consider the navigation of a mobile robot, say an autonomous car. The output is the angle by which the steering wheel should be turned at each time, to advance without hitting obstacles and deviating from the route. Inputs are provided by sensors on the car like a video camera, GPS, and so forth.

Differentregressionmodels

Thedifferentregressionmodelsaredefinedbasedontypeoffunctionsusedtorepresentherelation between the dependent variable y and the independent variables

1. Simplelinearregression

   Assume that there is only one independent variable x. If the relation between x and y is modeled by the relation $y=a+bx$

   then we have a simple linear regression.

2. Multipleregression

   Let there be more than one independent variable, say x1, x2, ..., xn, and let the relation between y and the independent variables be modeled as

   $$y = \alpha_0+\alpha_1 x_1+\cdots+\alpha_n x_n$$

   then it is case of multiple linear regression or multiple regression.

3. Polynomialregression

   Let there be only one variable x and let the relation between x, y be modeled as

   $$y=a_0+a_1x+a_2x^2+\cdots+a_nx^n$$

   for some positive integer n>1, then we have a polynomial regression.

4. Logisticregression

   Logistic regression is used when the dependent variable is binary (0/1, True/False, Yes/No) in nature. Even though the output is a binary variable, what is being sought is a probability function which may take any value from 0 to 1.

MODULE IV

1. What is entropy? What is its role in decision tree creation?

   Entropy is a measure of uncertainty/purity associated with a random variable. The entropy value ranges from 0-1. The entropy is 0 if the outcome is ``certain''. The entropy is maximum if we have no knowledge of the system (or any outcome is equally possible).

   Consider: S is a sample of training examples

   p+ is the proportion of positive examples in S

   p- is the proportion of negative examples in S

   Entropy of S: average optimal number of bits to encode information about certainty/uncertainty about S

   Entropy(S) = p+(-log2p+) + p-(-log2p-) = -p+log2p+- p-log2p-

   That is expected information (entropy) needed to classify a tuple in a dataset D:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

2. Identify the best splitting attribute for the following data using information gain.

Attributes				Classes
Gender	Car ownership	Travel cost	Income	Transportation mode
Male	0	cheap	Low	Bus
Male	1	cheap	Medium	Bus
Female	1	cheap	Medium	Train
Female	0	cheap	Low	Bus
Male	1	cheap	Medium	Bus
Male	0	standard	Medium	Train
Female	1	standard	Medium	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car

Entropy (D) = -(4/10)*log2(4/10) -(3/10)*log2(3/10) - (3/10)*log2(3/10)

= 1.3

Entropy (travel_cost) =5/10[-(4/5)*log2(4/5)-(1/5)log2(1/5)] +

3/10[-(3/3)*log2(3/3) + 2/10[-(2/2)*log2(2/2)

=0.36

Gain (travel_cost) = 1.3 - 0.36 = 0.94

Entropy (gender) =5/10[-(1/5)*log2(1/5)- (2/5)*log2(2/5)- (2/5)*log2(2/5)] +
5/10[-(3/5)*log2(3/5)- (1/5)*log2(1/5)- (1/5)*log2(1/5)]
= 1.4

Gain (gender) = 1.3-1.4 = -0.1

Entropy (ownership) =3/10[-(2/3)*log2(2/3)- (1/3)*log2(1/3)] +
5/10[-(2/5)*log2(2/5)- (1/5)*log2(1/5)- (2/5)*log2(2/5)]
+ 2/10[(2/2)*log2(2/2)]
= 1.035

Gain (ownership) = 1.3-1.035 = 0.275

Entropy (income) = (6/10) * 1.459 = 0.8754

Gain (income) = 1.3-0.8754 = 0.4246

Gain of Attribute travel cost is higher. Hence Best splitting attribute is Travel cost

For the complete tree construction refer the ppt given.

3. Examine the various attribute selection measures used in popular decision tree induction algorithms.

For selecting the splitting criterion that, best separates a given data partition D of class labelled training tuples into individual classes.Attribute selection Measures provides ranking for each attribute describing the given training tuples.The attribute having best score for the measure is chosen as splitting attribute for the given tuples.The tree node created for partition D is labelled with splitting criterion, branches are grown for each outcome and tuples are partitioned accordingly.

Information gain, gain ratio and Gini Index are the popular attribute selection measures.

a. Information Gain:

Select the attribute with the highest information gain. The attribute with the highest information gain is chosen as the splitting attribute for node N. This attribute minimizes the information needed to classify the tuples in the resulting partitions and reflects the least randomness or “impurity” in these partitions. Such an approach minimizes the expected number of tests needed to classify a given tuple and guarantees that a simple (but not necessarily the simplest) tree is found.

Let pi be the probability that an arbitrary tuple in D belongs to class Ci, estimated by |Ci, D|/|D|

Expected information (entropy) needed to classify a tuple in D:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

Information needed (after using A to split D into v partitions) to classify D:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times I(D_j)$$

Information gained by branching on attribute A

$$Gain(A) = Info(D) \quad Info_A(D)$$

Information gain is defined as the difference between the original information requirement (i.e.,based on just the proportion of classes) and the new requirement (i.e., obtained after partitioning on A). Gain(A) tells us how much would be gained by branching on A.

b. Gain Ratio

It is a kind of normalization to information gain using a “split information” value defined analogously with Info(D). Normalization is applied to overcome the problem of multivalued attribute.

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

GainRatio(A) = Gain(A)/SplitInfo(A)

c. Gini Index

Gini index measures the impurity of D, a data partition or set of training tuples, as

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2,$$

where pi is the probability that a tuple in D belongs to class Ci and is estimated by |Ci,D|/|D|. The sum is computed over m classes.

If a data set D is split on A into two subsets D1 and D2, the gini index gini(D) is defined as

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

Reduction in Impurity:

$$\Delta gini(A) = gini(D) - gini_A(D)$$

The attribute provides the smallest ginisplit(D) (or the largest reduction in impurity) is chosen to split the node (need to enumerate all the possible splitting points for each attribute)

It considers binary split for each attribute.

4. Outline the basic algorithm/ID3 for inducing a decision tree from training tuples.

```
ID3(D, Attributes, Attribute_selection_method)
    1. Create a node N;
    2. If all tuples in D are of same class C, then
       return N as a leaf node labelled with class C;
    3. If Attributes is empty then
       return N as a leaf node labelled with the majority class in D;.
    4. Apply attribute_selection_method(D, attribute_list) to find the attribute
       that best classifies tuples in D.
    5. Label node N with splitting_criterion;
    6. If splitting_attribute is discrete-valued and multiway splits allowed
       then
       Attribute_list = attribute_list-splitting_attribute
    7. For each outcome j of splitting_criterion
    8. Let Dj be the et of tuples in D that satisfying the outcome j;
    9. If Dj is empty then
       attach a leaf node with label of the majority class in D
    10. Else   add the subtree ID3(Dj, Attribute_list)
    11. Return N
```

5. Analyze the practical issues of decision tree learning

Learning a tree that classifies the training data perfectly may not lead to the tree with the best generalization performance. Training error no longer provides a good estimate of how well the tree will perform on previously unseen records

The following are the practical issues of decision tree learning:

i) Overfitting: H more complex than C or f

Overfitting results in decision trees that are more complex than necessary.Too many branches, some may reflect anomalies due to noise or outliers. It creates poor accuracy for unseen samples.

A hypothesis h is said to overfit the training data if there is another hypothesis, h’, such that h has smaller error than h’ on the training data but h has larger error on the test data than h’.

Two cases of overfitting:

When there is noise in the data or when the number of training examples are too small. In these cases the algorithm can produce trees that overfit the training examples.

i) Overfitting due to noise in the training data.

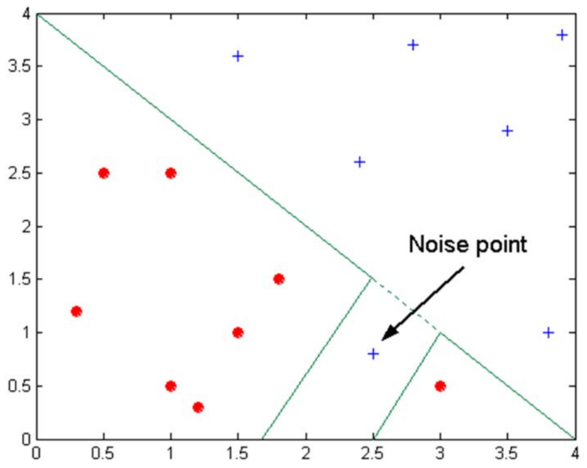
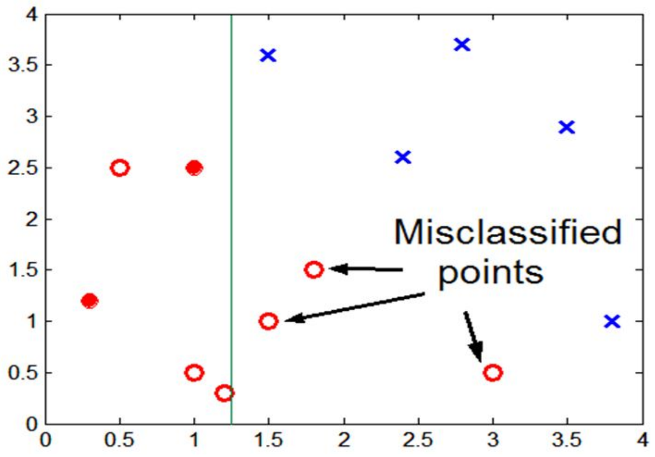


Figure shows that decision boundary is distorted by noise point.

ii) Overfitting due to Insufficient Examples (sample is too small)

Lack of data points makes it difficult to predict correctly the class labels of that region



The main approach to avoid overfitting is pruning.

Pruning is a technique that reduces the size of decision trees by removing sections of the tree that provide little power to classify instances. Pruning reduces the complexity of the final classifier, and hence improves predictive accuracy by the reduction of overfitting.

ii) Underfitting: H less complex than C or f

When model is too simple, both training and test errors are large.

iii) Missing Values

iv) Costs of Classification

6. Describe the approaches for handling the overfitting of decision trees.

Two approaches to avoid overfitting are:

i) Prepruning: Halt tree construction early—

We may apply pruning earlier, that is, before it reaches the point where it perfectly classifies the training data. That is do not split a node if this would result in the goodness measure falling below a threshold. Difficulty is to choose an appropriate threshold.

a) Evaluate splits before installing them:
- Don't install splits that don't look worthwhile
- When no worthwhile splits to install, done

b) Typical stopping conditions for a node:
- Stop if all instances belong to the same class
- Stop if all the attribute values are the same

c) More restrictive conditions:
- Stop if number of instances is less than some user-specified threshold
- Stop if class distribution of instances are independent of the available features.
- Stop if expanding the current node does not improve impurity measures (e.g., Gini or information gain).

ii) Postpruning: Remove branches from a "fully grown" tree—

We may allow the tree to overfit data, and then post-prune the tree.One commonly used criterion used to determine the correct final tree size is get a sequence of progressively pruned trees. Use a set of data different from the training data to decide which is the "best pruned tree".

Post-pruning: A cross validation approach

- Partition training data into “grow” set and “validation” set.
- Build a complete tree for the “grow” data
- Until accuracy on validation set decreases, do:

  For each non-leaf node in the tree

  Temporarily prune the tree below; replace it by majority vote

  Test the accuracy of the hypothesis on the validation set

  Permanently prune the node with the greatest increase

in accuracy on the validation test.

7. CART Algorithm: Regression Trees for Prediction

Regression trees are used with continuous outcome variable. Procedure similar to classification tree.Many splits attempted, choose the one that minimizes impurity.

The CART, or Classification And Regression Tree is a binary decision tree that is constructed by splitting a node into two child nodes repeatedly, beginning with the root node that contains the whole learning sample. That is data is split into two partitions. Partitions can also be split into sub-partitions. Hence procedure is recursive. Each Split based on only one variable.

- CART tree is generated by repeated partitioning of data set
- Parent gets two children
- Each child produces two grandchildren
- Four grandchildren produce 8 great grandchildren

The main elements of CART are:

- Rules for splitting data at a node based on the value of one variable
- Stopping rules for deciding when a branch is terminal and can be split no more
- A prediction for the target variable in each terminal node

Key CART features

- o Automated field selection
    - ▪ Handles any number of fields
        - Automatically selects relevant fields
- No data preprocessing needed
    - Does not require any kind of variable transforms

        - Impervious to outliers
    - Missing value tolerant
        - Moderate loss of accuracy due to missing values

CART Algorithm Steps

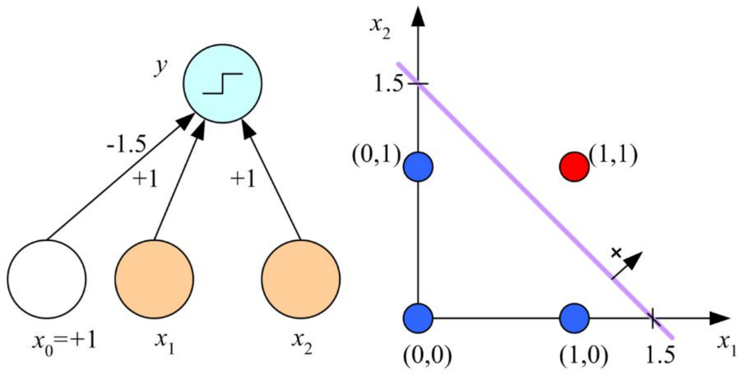
Decision Tree building algorithm involves a few simple steps and these are:

i. Take Labelled Input data - with a Target Variable and a list of Independent Variables
ii. Best Split: Find Best Split for each of the independent variables
iii. Best Variable: Select the Best Variable for the split
iv. Split the input data into Left and Right Nodes
v. Continue step ii-iv on each of the nodes until meet stopping criteria
vi. Decision Tree Pruning : Steps to prune Decision Tree built

8. Explain the concept of perceptron by implementing Binary ANDand OR logic.

Y = s(x1+x2-1.5)

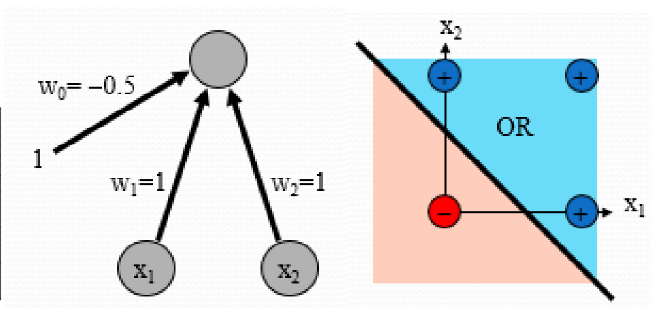
x_1	x_2	r
0	0	0
0	1	0
1	0	0
1	1	1



Boolean OR logic

Y = s(x1+x2-0.5)

input x1	input x2	ouput
0	0	0
0	1	1
1	0	1
1	1	1



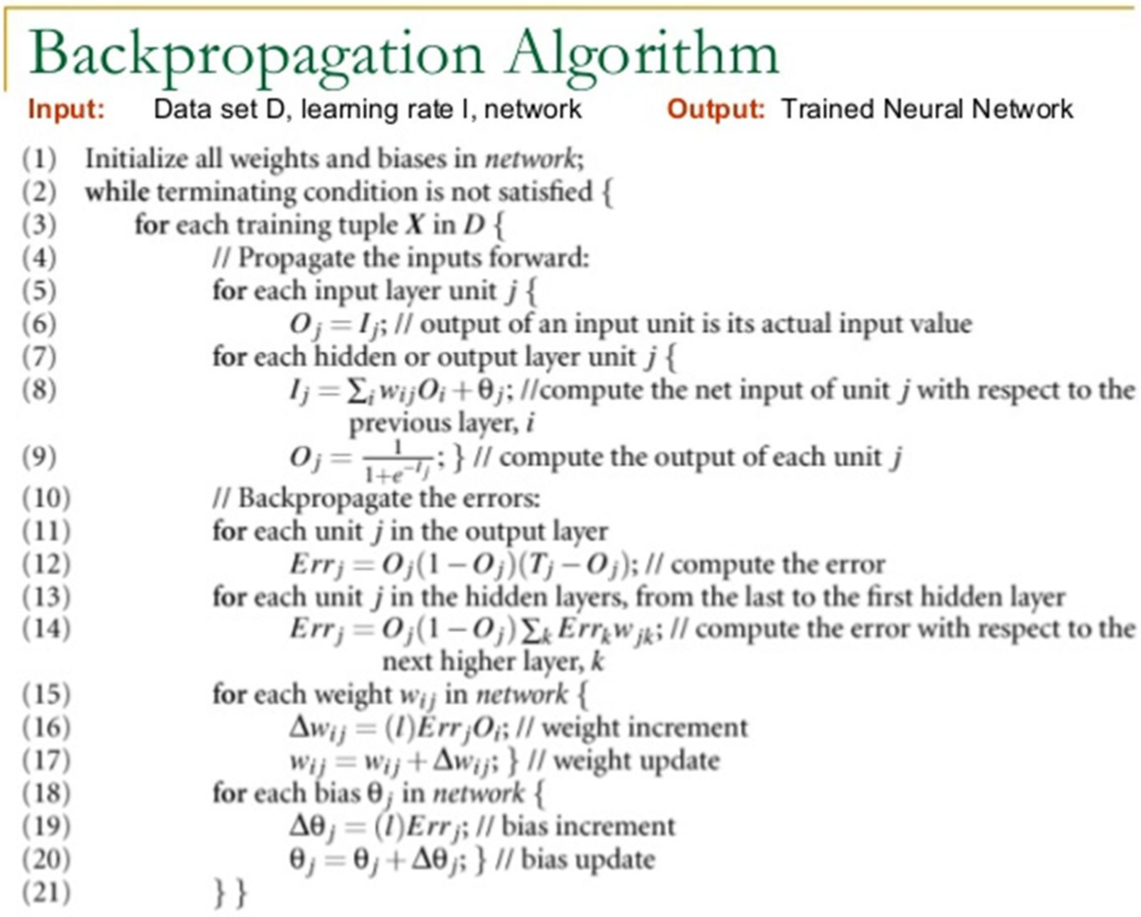
Boolean functions AND and OR are linearly separable.

9. Explain back propagation algorithm and write down back propagation algorithm in Neural networks.

Back propagation algorithm iteratively process a set of training tuples & compare the network's prediction with the actual known target value. For each training tuple, the weights are modified to minimize the mean squared error between the network's prediction and the actual target value. Modifications are made in the “backwards” direction: from the output layer, through each hidden layer down to the first hidden layer, hence the name “backpropagation”

Steps Involved are:

- Initialize weights to small random #s (-1.0 to 1.0) and biases (threshold θj) in the network
- Propagate the inputs forward (by applying activation function)
- Backpropagate the error (by updating weights and biases)
- Terminating condition (when error is very small, etc.)



Backpropagation Algorithm

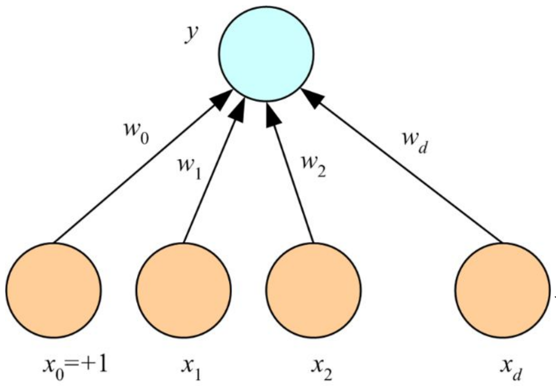
Input: Data set D, learning rate l, network **Output:** Trained Neural Network

```
(1)  Initialize all weights and biases in network;
(2)  while terminating condition is not satisfied {
(3)      for each training tuple X in D {
(4)          // Propagate the inputs forward:
(5)          for each input layer unit j {
(6)              $O_j = I_j$; // output of an input unit is its actual input value
(7)          for each hidden or output layer unit j {
(8)              $I_j = \sum_i w_{ij}O_i + \theta_j$; //compute the net input of unit j with respect to the
                     previous layer, i
(9)              $O_j = \frac{1}{1+e^{-I_j}}$; } // compute the output of each unit j
(10)         // Backpropagate the errors:
(11)         for each unit j in the output layer
(12)             $Err_j = O_j(1-O_j)(T_j - O_j)$; // compute the error
(13)         for each unit j in the hidden layers, from the last to the first hidden layer
(14)             $Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$; // compute the error with respect to the
                     next higher layer, k
(15)         for each weight $w_{ij}$ in network {
(16)             $\Delta w_{ij} = (l)Err_j O_i$; // weight increment
(17)             $w_{ij} = w_{ij} + \Delta w_{ij}$; } // weight update
(18)         for each bias $\theta_j$ in network {
(19)             $\Delta\theta_j = (l)Err_j$; // bias increment
(20)             $\theta_j = \theta_j + \Delta\theta_j$; } // bias update
(21)         } }
```

10. What is perceptron?

It is the basic processing element called neuron; Figure shows the representation of a perceptron.Input come from environment or other perceptrons. Input is associated with connection weight called synaptic weight. Output is the weighted sum of inputs.w0 weight coming from an extra bias unit, x0 =1 to make the model more general.

The n-dimensional input vector x is mapped into variable **y** by means of the scalar product and a nonlinear function mapping



$$y = \sum_{j=1}^{d} w_j x_j + w_0 = \mathbf{w}^T \mathbf{x}$$

$$\mathbf{w} = [w_0, w_1, \ldots, w_d]^T$$

$$\mathbf{x} = [1, x_1, \ldots, x_d]^T$$

11. What is activation function?

A function that could convert any given value to a value between 0 to 1. Logistic and sigmoid are the commonly used activation function. This function is also referred to as a squashing function.

To fit a binary decision a sigmoid function prevents extreme points from moving the zero crossing point (the inflection point).With a sigmoid function, adding or removing extreme values won't cause the fitted intercept change very much.

It has an exponential in its function formula, it makes it easy to compute the derivative of the function (derivatives are computed to find the gradient descent).

The logistic function is nonlinear and differentiable, allowing the backpropagation algorithm to model classification problems that are linearly inseparable.

12. Describe Neural network training process

The ultimate objective of training is to obtain a set of weights that makes almost all the tuples in the training data classified correctly.

Steps

- Initialize weights with random values
- Feed the input tuples into the network one by one
- For each unit
  - Compute the net input to the unit as a linear combination of all the inputs to the unit
  - Compute the output value using the activation function
  - Compute the error
  - Update the weights and the bias

i) Initialize the weights: The weights in the network are initialized to small random numbers (e.g., ranging from−1.0 to 1.0, or−0.5 to 0.5). Each unit has a bias associated with it. The biases are similarly initialized to small random numbers. Each training tuple, X, is processed by the following steps.

ii) Propagate the inputs forward: First, the training tuple is fed to the network's input layer. The inputs pass through the input units, unchanged. That is, for an input unit, j its output, Oj, is equal to its input value, Ij.

iii) Next, the net input and output of each unit in the hidden and output layers are computed. The net input to a unit in the hidden or output layers is the outputs of the units connected to it in the previous layer. Each connection has a weight.

iv) To compute the net input to the unit, each input connected to the unit is multiplied by its corresponding weight, and this is summed. Given a unit, j in a hidden or output layer, the net input, Ij, to unit j is

$$I_j = \sum_i w_{ij} O_i + \theta_j$$

wherewij is the weight of the connection from unit i in the previous layer to unit j; Oi is the output of unit i from the previous layer; and θj is the bias of the unit.

v) Each unit in the hidden and output layers takes its net input and then applies an activation function to it. The logistic, or sigmoid, function is used. Given the net input Ij to unit j, then Oj, the output of unit j, is computed as

It maps a large input domain onto smallerrange of 0 to 1.

$$O_j = \frac{1}{1+e^{-I_j}}$$

vi) Backpropagate the error: The error is propagated backward by updating the weights and biases to reflect the error of the network's prediction. For a unit j in the output layer, the error Errj is computed by

$$Err_j = O_j(1-O_j)(T_j - O_j)$$

whereOj is the actual output of unit j, and Tj is the known target value of the given training tuple. Oj(1−Oj) is the derivative of the logistic function.

vii) To compute the error of a hidden layer unit j, the weighted sum of the errors of the units connected to unit j in the next layer are considered. The error of a hidden layer unit j is

$$Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$$

wherewjk is the weight of the connection from unit j to a unit k in the next higher layer, and Errk is the error of unit k.

viii) The weights and biases are updated to reflect the propagated errors. Weights are updated by the following equations, where Δwij is the change in weight wij:

$$\boxed{w_{ij} = w_{ij} + (l)Err_j O_i} \quad \Delta wij = (l)ErrjOi$$

The variable l is the learning rate, a constant typically having a value between 0.0 and 1.0. Backpropagation learns using a gradient descent method to search for a set of weights that fits the training data so as to minimize the mean squared distance between the network's class prediction and the known target value of the tuples

ix) Biases are updated by the following equations, where Δθj is the change in bias θj:

Δθj=(l)Errj. θj =θj+Δθj.

one iteration through the training set is an epoch.

x) Terminating condition: Training stops when

All Δwij in the previous epoch are so small as to be below some specified threshold, or The percentage of tuples misclassified in the previous epoch is below some threshold,or A prespecified number of epochs has expired.

13. Sample calculations for learning by the backpropagation algorithm. (refer class notes for details)

Figure 6.18 shows amultilayerfeed-forwardneuralnetwork.Letthelearningratebe0.9.Theinitialweight and bias values of the network are given in Table 6.3, along with the first training tuple, X =(1,0,1), whose class label is 1. This example shows the calculations for backpropagation, given the first training tuple, X. The tuple is fed into the network, and the net input and output of each unit are computed. These values are shown in Table 6.4. The error of each unit is computedand propagated backward. The error values are shown in Table 6.5. The weight and bias updates are shown in Table 6.6.

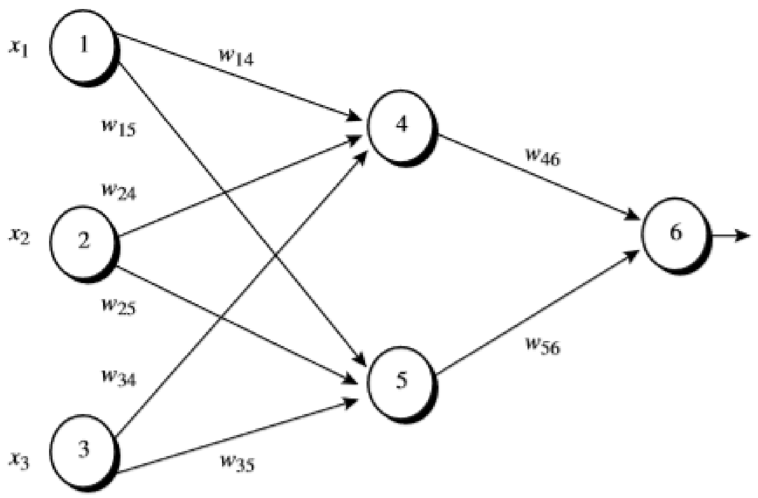


Figure 6.18 An example of a multilayer feed-forward neural network.

Table 6.3 Initial input, weight, and bias values.

x_1	x_2	x_3	w_{14}	w_{15}	w_{24}	w_{25}	w_{34}	w_{35}	w_{46}	w_{56}	θ_4	θ_5	θ_6
1	0	1	0.2	−0.3	0.4	0.1	−0.5	0.2	−0.3	−0.2	−0.4	0.2	0.1

Table 6.4 The net input and output calculations.

Unit j	*Net input, I_j*	*Output, O_j*
4	$0.2+0-0.5-0.4=-0.7$	$1/(1+e^{0.7})=0.332$
5	$-0.3+0+0.2+0.2=0.1$	$1/(1+e^{-0.1})=0.525$
6	$(-0.3)(0.332)-(0.2)(0.525)+0.1=-0.105$	$1/(1+e^{0.105})=0.474$

Table 6.5 Calculation of the error at each node.

Unit j	*Err_j*
6	$(0.474)(1-0.474)(1-0.474)=0.1311$
5	$(0.525)(1-0.525)(0.1311)(-0.2)=-0.0065$
4	$(0.332)(1-0.332)(0.1311)(-0.3)=-0.0087$

Table 6.6 Calculations for weight and bias updating.

Weight or bias	*New value*
w_{46}	$-0.3+(0.9)(0.1311)(0.332)=-0.261$
w_{56}	$-0.2+(0.9)(0.1311)(0.525)=-0.138$
w_{14}	$0.2+(0.9)(-0.0087)(1)=0.192$
w_{15}	$-0.3+(0.9)(-0.0065)(1)=-0.306$
w_{24}	$0.4+(0.9)(-0.0087)(0)=0.4$
w_{25}	$0.1+(0.9)(-0.0065)(0)=0.1$
w_{34}	$-0.5+(0.9)(-0.0087)(1)=-0.508$
w_{35}	$0.2+(0.9)(-0.0065)(1)=0.194$
θ_6	$0.1+(0.9)(0.1311)=0.218$
θ_5	$0.2+(0.9)(-0.0065)=0.194$
θ_4	$-0.4+(0.9)(-0.0087)=-0.408$

try it now

A KTU STUDENTS PLATFORM

SYLLABUS

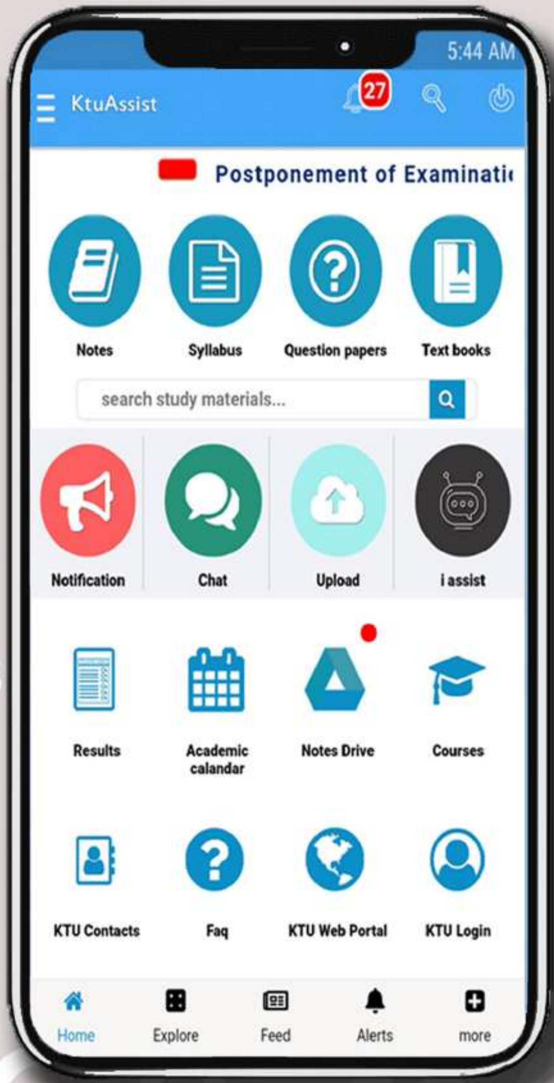
NOTES

TEXT BOOKS

QUESTION PAPERS

KTU NOTIFICATIONS

DOWNLOAD IT FROM GOOGLE PLAY



CHAT

CALENDAR

FAQ

LOGIN

MUCH MORE

DOWNLOAD APP

ktuassist.in

instagram.com/ktu_assist

facebook.com/ktuassist