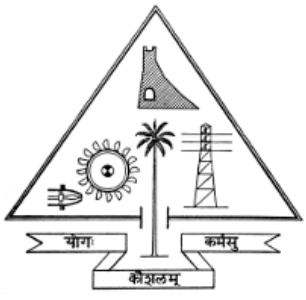
APJ Abdul Kalam Technological University

Machine Learning

Assignment 1

S7, CSE
Government Engineering
College, Thrissur

November 25, 2020



1. **Distinguish between classification and regression with an example. (Dec 2018) 4 marks**

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

1. a number in regression
2. a class code in the case of classification.

Classification: Identification of class to which a data belongs It is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y).

The output variables are often called labels or categories. The mapping function predicts the class or category for a given observation.

For example, an email of text can be classified as belonging to one of two classes: "spam" and "not spam".

- A classification problem requires that examples be classified into one of two or more classes.
- Classification can have real-valued or discrete input variables.
- Problem with two classes is often called a two-class or binary classification problem.
- Problem with more than two classes is often called a multi-class classification problem.
- Problem where an example is assigned multiple classes is called a multi-label classification problem.

It is common for classification models to predict a continuous value as the probability of a given example belonging to each output class. The probabilities can be interpreted as the likelihood or confidence of a given example belonging to each class. A predicted probability can be converted into a class value by selecting the class label that has the highest probability.

A common example of classification comes with detecting spam emails. To write a program to filter out spam emails, a computer programmer

can train a machine learning algorithm with a set of spam-like emails labelled as spam and regular emails labelled as not-spam. The idea is to make an algorithm that can learn characteristics of spam emails from this training set so that it can filter out spam emails when it encounters new emails. .

A specific email of text may be assigned the probabilities of 0.1 as being "spam" and 0.9 as being "not spam". We can convert these probabilities to a class label by selecting the "not spam" label as it has the highest predicted likelihood.

Regression is the task of approximating a mapping function (f) from input variables (X) to a continuous output variable (y).

A continuous output variable is a real-value, such as an integer or floating point value. These are often quantities, such as amounts and sizes.

For example, a house may be predicted to sell for a specific price , perhaps in the range of Rs 100,000 to Rs 200,000 by looking into a whole bunch of exogenous variables such as neighbourhood, location, bathrooms in the house, bedrooms, how far is it from the city .Thus here the input variables are neighbourhood, location, bathrooms in the house, bedrooms, how far is it from the city and the output variable is price of the house. Another example include predicting the number of death cases world wide due to COVID-19 for next 10 days using the help of previous COVID-19 death cases' time series table as training dataset which is used to train the regression model

- A regression problem requires the prediction of a quantity.
- A regression can have real valued or discrete input variables.
- A problem with multiple input variables is often called a multivariate regression problem.
- A regression problem where input variables are ordered by time is called a time series forecasting problem.

A classification algorithm may predict a continuous value, but the continuous value is in the form of a probability for a class label. A regression algorithm may predict a discrete value, but the discrete value in the form of an integer quantity.

2. **Briefly describe the concept of Expectation Maximization algorithm. (Dec 2018) 4 marks**

The expectation-maximization algorithm (EM algorithm) is an approach for performing maximum likelihood estimation in the presence of latent variables. It does this by first estimating the values for the latent variables, then optimizing the model, then repeating these two steps until convergence. It is an effective and general approach and is most commonly used for density estimation with missing data, such as clustering algorithms like the Gaussian Mixture Model.

The maximum likelihood estimation method (MLE) is a method for estimating the parameters of a statistical model, given observations. The method attempts to find the parameter values that maximize the likelihood function, or equivalently the log-likelihood function, given the observations. The expectation-maximisation algorithm is used to find maximum likelihood estimates of the parameters of a statistical model in cases where the equations cannot be solved directly. These models generally involve latent or unobserved variables in addition to unknown parameters and known data observations. For example, a Gaussian mixture model can be described by assuming that each observed data point has a corresponding unobserved data point, or latent variable, specifying the mixture component to which each data point belongs.

Algorithm:

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

In the case of Gaussian mixture problems, because of the nature of the function, finding a maximum likelihood estimate by taking the derivatives of the log-likelihood function with respect to all the parameters

and simultaneously solving the resulting equations is nearly impossible. So we apply the EM algorithm to solve the problem.

3. **A patient takes a lab test and the result comes back positive. It is known that the test returns a correct positive result in only 98% of the cases and a correct negative result in only 97% of the cases. Furthermore, only 0.008 of the entire population has this disease.**

   (a) What is the probability that this patient has cancer?

   (b) What is the probability that he does not have cancer?

   (c) What is the diagnosis? (May 2019) 4 marks

From the given data, we can determine the following probabilities

P(cancer) = 0.008
P(¬ cancer) = 1 - 0.008 = 0.992
P(+ve | cancer) = 0.98
P(-ve | cancer) = 1 - 0.98 = 0.02
P(-ve | ¬ cancer) = 0.97
P(+ve | ¬ cancer) = 1 - 0.97 = 0.03

According to Bayes theorem:

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

(a) Using Bayes formula:

$$P(cancer| + ve) = \frac{P(+ve|cancer) * P(cancer)}{P(+ve)}$$

$$= \frac{P(+ve|cancer) * P(cancer)}{P(+ve|cancer) * P(cancer) + P(+ve|\neg cancer) * P(\neg cancer)}$$

$$= \frac{0.98 * 0.008}{0.98 * 0.008 + 0.03 * 0.0992}$$

$$= 0.21$$

(b) Using Bayes formula:

$$P(\neg cancer \mid +ve) = \frac{P(+ve|\neg cancer) * P(\neg cancer)}{P(+ve)}$$

$$= \frac{P(+ve|\neg cancer) * P(\neg cancer)}{P(+ve|\neg cancer) * P(\neg cancer) + P(+ve|cancer) * P(cancer)}$$

$$= \frac{0.03 * 0.992}{0.98 * 0.008 + 0.992 * 0.03}$$

$$= 0.79$$

(c) From the given data, we can see that every 8 out of 1000 people has cancer. It is also clear that the number of false positive and false negative cases in cancer detection is very low. From (a) it is clear that the probability of having cancer in a positive test result is 21%. From (b) it is clear that the probability of not having cancer in a positive test result is 79%.

4. . **Explain any two model combination scheme to improve the accuracy of a classifier. (Sept 2020) 4 marks**

Ensemble is a machine learning concept in which multiple models are trained using the same learning algorithm. The principle behind the ensemble model is that a group of weak learners come together to form a strong learner, thus increasing the accuracy of the model.

Bagging and Boosting are the ensemble learning methods i.e., the methods for combining the predictions from different models.

Bagging: The term bagging is also known as bootstrap aggregation. In bagging methods, ensemble model tries to improve prediction accuracy and decrease model variance by combining predictions of individual models trained over randomly generated training samples. The final prediction of ensemble model will be given by calculating the average of all predictions from the individual estimators.

One of the best examples of bagging methods are random forests.

Boosting: In boosting method, the main principle of building ensemble model is to build it incrementally by training each base model estimator sequentially. As the name suggests, it basically combine several week base learners, trained sequentially over multiple iterations of training data, to build powerful ensemble. During the training of week base learners, higher weights are assigned to those learners which were misclassified earlier.

An example of boosting method is AdaBoost.

5. **Explain the procedure for the computation of the principal components of the data. (Sept 2020) 4 marks**

Firstly, lets see why we would need to compute the principal components of the data. We use certain procedures to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set. Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize and make analyzing data much easier and faster for machine learning algorithms without extraneous variables to process. We call the process as Principal Component Analysis or PCA. Thus using PCA, we reduce the number of variables of a data set, while preserving as much information as possible.

Lets look at the steps involved in PCA:

Step 1: Standardization

The aim of this step is to standardize the range of the continuous initial variables so that each one of them contributes equally to the analysis. More specifically, the reason why it is critical to perform standardization prior to PCA, is that the latter is quite sensitive regarding the variances of the initial variables. That is, if there are large differences between the ranges of initial variables, those variables with larger ranges will dominate over those with small ranges (For example, a variable that ranges between 0 and 100 will dominate over a variable that ranges between 0 and 1), which will lead to biased results. So, transforming the data to comparable scales can prevent this problem.

Mathematically, this can be done by subtracting the mean and dividing by the standard deviation for each value of each variable.

$$z = \frac{value - mean}{standard\ \ deviation}$$

Once the standardization is done, all the variables will be transformed to the same scale.

Step 2: Covariance Matrix computation

The aim of this step is to understand how the variables of the input data set are varying from the mean with respect to each other, or in other words, to see if there is any relationship between them. Because sometimes, variables are highly correlated in such a way that they contain redundant information. So, in order to identify these correlations, we compute the covariance matrix.

The covariance matrix is a p × p symmetric matrix (where p is the number of dimensions) that has as entries the covariances associated with all possible pairs of the initial variables. For example, for a 3-dimensional data set with 3 variables x, y, and z, the covariance matrix is a 3×3 matrix of this form:

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

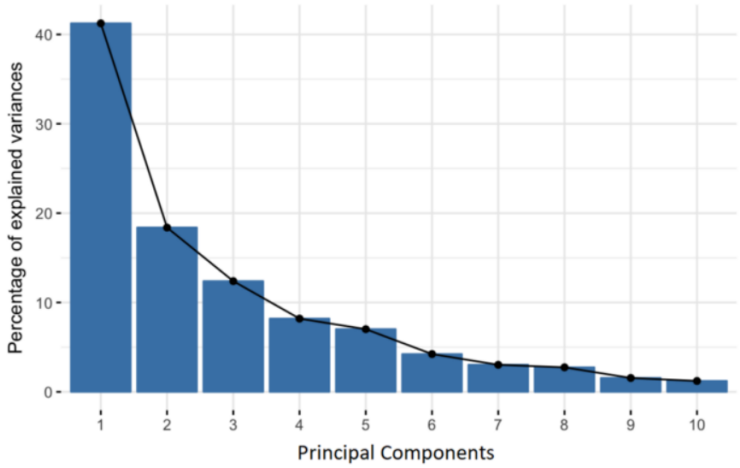
Since the covariance of a variable with itself is its variance, i.e., $(Cov(a,a) = Var(a))$, in the main diagonal (Top left to bottom right) we actually have the variances of each initial variable. And since the covariance is commutative, i.e., $(Cov(a,b) = Cov(b,a))$, the entries of the covariance matrix are symmetric with respect to the main diagonal, which means that the upper and the lower triangular portions are equal. The covarience table summarises the correlations between all the possible pairs of variables. If the sign of the covarience is:

- negative, the two variables increase or decrease together(correlated)
- positive, one increases when the other one decreases(inversely correlated)

Step 3: Compute the eigenvectors and eigenvalues of the covariance matrix to identify the principal components

Eigenvectors and eigenvalues are the linear algebra concepts that we need to compute from the covariance matrix in order to determine the principal components of the data. Before getting to the explanation of these concepts, let's first understand what do we mean by principal components.

Principal components are new variables that are constructed as linear combinations or mixtures of the initial variables. These combinations are done in such a way that the new variables (i.e., principal components) are uncorrelated and most of the information within the initial variables is squeezed or compressed into the first components. So, the idea is 10-dimensional data gives you 10 principal components, but PCA tries to put maximum possible information in the first component, then maximum remaining information in the second and so on, until having something like shown in the screen plot below.



Organizing information in principal components this way, will allow you to reduce dimensionality without losing much information, and this by discarding the components with low information and considering the remaining components as your new variables.

An important thing to realize here is that, the principal components are less interpretable and don't have any real meaning since they are constructed as linear combinations of the initial variables.

Geometrically speaking, principal components represent the directions of the data that explain a maximal amount of variance, that is to say, the lines that capture most information of the data. The relationship between variance and information here, is that, the larger the variance carried by a line, the larger the dispersion of the data points along it, and the larger the dispersion along a line, the more the information it has. To put all this simply, just think of principal components as new axes that provide the best angle to see and evaluate the data, so that the differences between the observations are better visible.

Step 4: Feature Vector

Computing the eigenvectors and ordering them by their eigenvalues in descending order allow us to find the principal components in order of significance. In this step, what we do is, to choose whether to keep all these components or discard those of lesser significance (of low eigenvalues), and form with the remaining ones a matrix of vectors that we call Feature vector.

So, the feature vector is simply a matrix that has as columns the eigenvectors of the components that we decide to keep. This makes it the first step towards dimensionality reduction, because if we choose to keep only p eigenvectors (components) out of n, the final data set will have only p dimensions. Thus, it's up to you to choose whether to keep all the components or discard the ones of lesser significance, depending on what you are looking for. Because if you just want to describe your data in terms of new variables (principal components) that are uncorrelated without seeking to reduce dimensionality, leaving out lesser significant components is not needed.

Step 5: Recast the Data Along the Principal Components Axes

In the previous steps, apart from standardization, you do not make any changes on the data, you just select the principal components and form the feature vector, but the input data set remains always in terms of the original axes (i.e, in terms of the initial variables).

In this step, which is the last one, the aim is to use the feature vector formed using the eigenvectors of the covariance matrix, to reorient the data from the original axes to the ones represented by the principal components (hence the name Principal Components Analysis). This

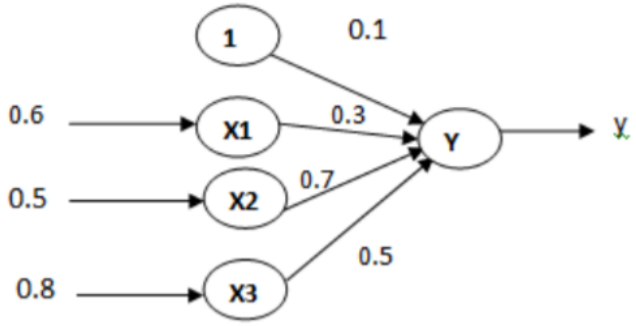
can be done by multiplying the transpose of the original data set by the transpose of the feature vector.

$$FinalDataSet = FeatureVector^T * StandardizedOriginalDataset^T$$

6. **Calculate output of the following neuron Y if the activation function is . (December 2018 Q6)**

   (a) Binary Sigmoid

   (b) Bipolar Sigmoid **4 marks**



Let input of activation function be x
x = 1 x w1 + X1 x w2 + X2 x w3 + X3 x w4
x = 1 x 0.1 + 0.6 x 0.3 + 0.5 x 0.7 + 0.8 x 0.5
x = 0.1 + 0.18 + 0.35 + 0.40
x = 1.03
Output of neuron is y

(a)

$$BinarySigmoid = \frac{1}{1+e^{-x}}$$

$$y = \frac{1}{1+e^{-1.03}} = 0.7369$$

(b)

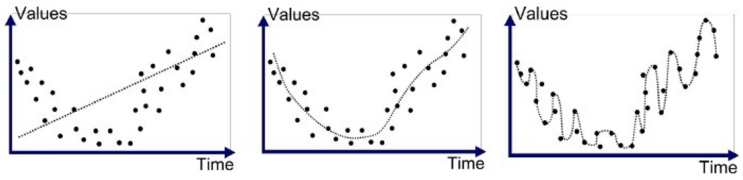
$$BipolarSigmoid = -1 + \frac{2}{1+e^{-x}}$$

$$y = -1 + \frac{2}{1 + e^{-1.03}} = -0.4738$$

7. . **Distinguish between overfitting and underfitting. How it can affect model generalization? (September 2020 - Q3)**
. **4 marks**

Overfitting	Underfitting
A statistical model is said to be overfitted, when we train it with a lot of data.	A statistical model or a machine learning algorithm is said to have underfitting when it cannot capture the underlying trend of the data.
When a model gets trained with lots of data it starts learning from the noise and inaccurate data entries in the data set. Then the model does not categorize the data correctly, because of too many details and noise.	Underfitting destroys the accuracy of the machine learning model. Its occurrence simply means that the model or the algorithm does not fit the data of too many details well enough.
The causes of overfitting are the non-parametric and non-linear methods because these types of Machine Learning algorithms have more freedom in building the model based on the dataset and hence they can really build unrealistic models.	It usually happens when we have less data to build an accurate model and also when we try to build a linear model with a non-linear data. In such cases, the rules of the machine learning model are too easy and flexible to be applied on such minimal data and therefore the model will probably make a lot of wrong predictions.
A solution to avoid overfitting is using a linear algorithm if we have linear data or using the parameters like the maximal depth if we are using decision trees.	Underfitting can be avoided by using more data and also reducing the features by feature selection

In terms of model generalization, in an overfitted model model gener-



alization will be less. The model will have a tendency to give correct output only for those inputs which are present in the training dataset and give incorrect results to inputs not present in the dataset. On the other hand, an underfitted model shows very high model generalization causing it to give incorrect results to even an input present in the training dataset itself.

8. **Describe any two techniques used for Ensemble learning. (December 2019 - Q7) 4 marks**

Ensemble methods are techniques that create multiple models and then combine them to produce improved results. Ensemble methods usually produces more accurate solutions than a single model would.
Bagging and **Stacking** are two of the ensemble methods.

Bagging- Bootstrap Aggregating

The idea behind bagging is combining the results of multiple models (for instance, all decision trees) to get a generalized result.

Bootstrapping is a sampling technique in which we create subsets of observations from the original dataset, with replacement. The size of the subsets is the same as the size of the original set.

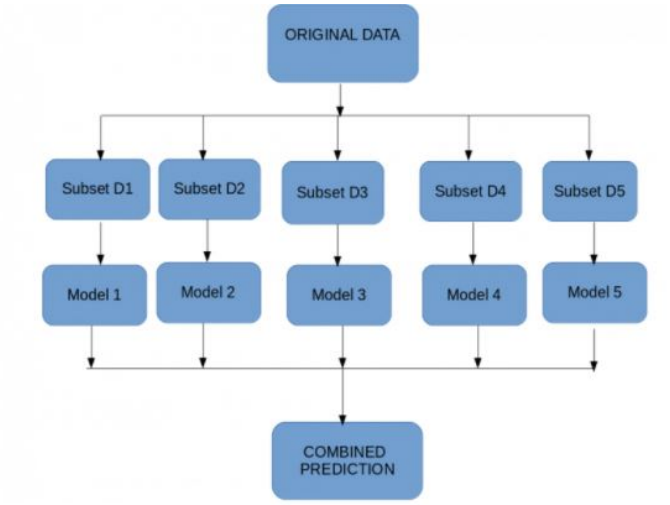
Bagging(or Bootstrap Aggregating) technique uses these subsets (bags) to get a fair idea of the distribution (complete set). The size of subsets created for bagging may be less than the original set.

(a) Multiple subsets are created from the original dataset, selecting observations with replacement.

(b) A base model (weak model) is created on each of these subsets.

(c) The models run in parallel and are independent of each other.

(d) The final predictions are determined by combining the predictions from all the models.



Stacking

Stacking is an ensemble learning technique that uses predictions from multiple models (for example decision tree, knn or svm) to build a new model. This model is used for making predictions on the test set. Below is a step-wise explanation for a simple stacked ensemble:

(a) The train set is split into 10 parts.

(b) A base model (suppose a decision tree) is fitted on 9 parts and predictions are made for the 10th part. This is done for each part of the train set.

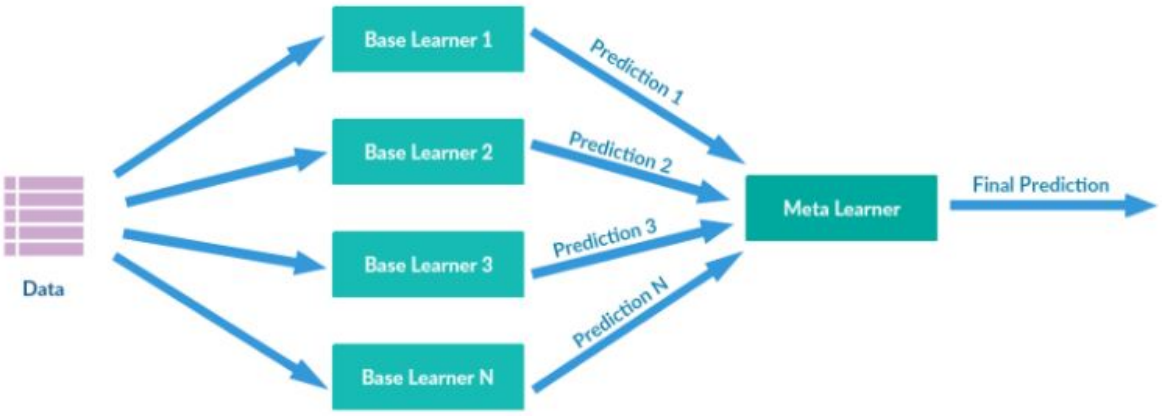
(c) The base model (in this case, decision tree) is then fitted on the whole train dataset.

(d) Using this model, predictions are made on the test set.

(e) Steps 2 to 4 are repeated for another base model (say knn) resulting in another set of predictions for the train set and test set.

(f) The predictions from the train set are used as features to build a new model.

(g) This model is used to make final predictions on the test prediction set.



9. . **Distinguish between supervised and reinforcement learning. illustrate with example. (September 2020 - Q11a) 4 marks**

Supervised Learning	Reinforcement Learning
In Supervised learning the decision is made on the initial input or the input given at the start	Reinforcement learning is all about making decisions sequentially. In simple words we can say that the output depends on the state of the current input and the next input depends on the output of the previous input
In supervised learning the decisions are independent of each other so labels are given to each decision.	In Reinforcement learning decision is dependent, So we give labels to sequences of dependent decisions.

Supervised learning Is the machine learning task of learning a function that maps an input to an output based on example input output pairs.	Reinforcement learning is the problem of getting an agent to act in the world so as to maximise its rewards.
In supervised learning, each example in the train set is a pair consisting of an input object and an output value.	In Reinforcement learning a learner is not told what actions to take as in most forms of machine learning, but instead must discover which actions yield the most reward by trying them.
Example: Object recognition	Example: Chess game

10. **Consider two data points in two dimensional A(5,8) and B(8,9).Calculate (a) City block distance (b) Chessboard distance between A & B. (December 2018 - Q10) 4 marks**

a.City block distance is given by,

$$\sum_{i=1}^{n} |x_{ik} - xjk|$$

$$d_{ba} = |5 - 8| + |8 - 9|$$

therefore

$$d_{ba} = 3 + 1 = 4$$

Therefore City block distance is 4

b.Chessboard distance between A and B is given by

$$\max(|x_2 - x_1|, |y_2 - y_1|)$$

$$\max(|5-8|, |8-9|)$$
$$\max(3, 1) = 3$$

Therefore Chessboard distance is 3

11. **Consider the training data in the following table where Play is a class attribute. In the table, the Humidity attribute has values "L" (for low) or "H" (for high), Sunny has values "Y" (for yes) or "N" (for no), Wind has values "S" (for strong) or "W" (for weak), and Play has values "Yes" or "No".**

Humidity	Sunny	Wind	Play
L	N	S	No
H	N	W	Yes
H	Y	S	Yes
H	N	W	Yes
L	Y	S	No

What is class label for the following day (Humidity=L, Sunny=N, Wind=W), according to naive Bayesian classification? (September 2020 - Q15b) 4 marks

We have given a table of instances and the corresponding target values for play class attribute.
We have to classify new instance Humidity=L, Sunny=N, Wind=W and find the target value, whether its YES or NO.
P(play = yes)= 3/5 = 0.6
P(play = no) = 2/5 = 0.4

$$v_{NB} = \underset{v_j \in \{yes, no\}}{\operatorname{argmax}} P(v_j) \prod_i P(a_i | v_j)$$

So for calculating the probability for the given instances,

P(v)P(humidity=low/v)P(sunny=no/v)P(wind=weak/v), where v ϵ yes/no.
P(yes)P(humidity=low/yes)P(sunny=no/yes)P(wind=weak/yes)
P(no)P(humidity=low/no)P(sunny=no/no)P(wind=weak/no)
P(humidity=low/yes) = 0/3 = 0
Here a condition occurs, model will assign a 0 (zero) probability and will be unable to make a prediction. It is called Zero frequency. To solve this issue, we can use smoothing technique. So we have to add smoothing parameter $\alpha = 1$ and k=2(number of values in class).
P(humidity=low/yes) = 0+1/3+2 = 1/5 = 0.2
P(sunny=no/yes) = 2/3 = 0.67
P(wind=weak/yes) = 2/3 = 0.67
P(yes) = 0.6
P(yes)P(humidity=low/yes)P(sunny=no/yes)P(wind=weak/yes) = 0.6*0.2*0.67*0.67 = 0.054

P(humidity=low/no) = 2/2 = 1
P(sunny=no/no) = 1/2 = 0.5
P(wind=weak/no) = 0+1/2+2 = 1/4 = 0.25
P(no) = 0.4
P(no)P(humidity=low/no)P(sunny=no/no)P(wind=weak/no) = 0.4*1*0.5*0.25 = 0.05

Since probability of play=yes is 0.054 and play=no is 0.05, we know that its sum should be equals 1. For this we have to done normalization.
P(play=yes) = 0.054/(0.054+0.05) = 0.52
P(play=no) = 0.05/(0.054+0.05) = 0.48
Since P(play=yes)+P(play=no) = 0.52+0.48 = 1
We can assume that according to naive Bayesian classification class label for the following day with attributes(Humidity = L, Sunny = N, Wind = W) is yes.

12. **Describe the significance of Kernal functions in SVM. List any two kernel functions. (Sept 2020) 4 marks**

Kernel Function is a method used to take data as input and transform into the required form of processing data.Kernel Function generally

transforms the training set of data so that a non-linear decision surface is able to transformed to a linear equation in a higher number of dimension spaces. Basically, It returns the inner product between two points in a standard feature dimension. SVM algorithms use a set of mathematical functions that are defined as the kernel. The function of kernel is to take data as input and transform it into the required form. Different SVM algorithms use different types of kernel functions. These functions can be different types. For example linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid. Introduce Kernel functions for sequence data, graphs, text, images, as well as vectors. The most used type of kernel function is RBF. Because it has localized and finite response along the entire x-axis.The kernel functions return the inner product between two points in a suit- able feature space. Thus by defining a notion of similarity, with little computational cost even in very high-dimensional spaces.
Gaussian Kernel: It is used to perform transformation, when there is no prior knowledge about data.
from sklearn.svm import SVC
classifier = SVC(kernel='rbf ', random state = 0)
training set in x, y axis
classifier.fit(x train, y train)

Polynomial Kernel: It represents the similarity of vectors in training set of data in a feature space over polynomials of the original variables used in kernel.
from sklearn.svm import SVC
classifier = SVC(kernel ='poly', degree = 4)
classifier.fit(x train, y train)

13. **List any four applications of Machine Learning.(May 2019) . 4 marks**

Machine learning is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that which makes it more similar to humans: The ability to learn. Machine learning is actively being used today, perhaps in many more places than one would expect. We probably use a learning

algorithm dozens of time without even knowing it. Applications of Machine Learning include:

(a) Web Search Engine:One of the reasons why search engines like google, bing etc work so well is because the system has learnt how to rank pages through a complex learning algorithm.

(b) Photo tagging Applications: Be it Facebook or any other photo tagging application, the ability to tag friends makes it even more happening. It is all possible because of a face recognition algorithm that runs behind the application.

(c) Spam Detector:Our mail agent like Gmail or Hotmail does a lot of hard work for us in classifying the mails and moving the spam mails to spam folder. This is again achieved by a spam classifier running in the back end of mail application.

(d) Traffic Alerts(Maps): Everyone using maps is providing their location, average speed, the route in which they are traveling which in turn helps Google collect massive Data about the traffic, which makes them predict the upcoming traffic and adjust your route according to it.

14. **Define Hidden Markov Model. (May 2019) 4 marks**

In Hidden Markov Model, (HMM), the states are not observable, but when we visit a state, an observation is recorded.Hidden Markov models (HMMs) are a formal foundation for making probabilistic models of linear sequence 'labeling' problems 1 , 2 . They provide a conceptual toolkit for building complex models just by drawing an intuitive picture. They are at the heart of a diverse range of programs, including gene finding, profile searches, multiple sequence alignment and regulatory site identification. HMMs are the Legos of computational sequence analysis.
Because of their flexibility and computational efficiency, hidden Markov models have found a wide application in many different fields. They are known for their use in temporal pattern recognition and generation such as speech recognition, handwriting recognition, and speech synthesis.
Hidden Markov models evolve according to two rules:

(a) The first rule is that the model moves from the current state to the next state, which may be the same state, according to some

probability distribution that depends only on the current state. This is known as the Markov property. Intuitively, this rule states that the system evolves without regard to past states of the system and only depends on the current state.

(b) The second rule is that after each transition, the model emits an observation whose distribution depends only on the current state. Since the model only emits the observations and the states generating the observations are unknown to the observer, the states generating those observations are called hidden states.

15. **The following data set contains factors that determine whether tennis is played or not. Using Naive Bayes classifier, find the play prediction for the day <Sunny, Cool, High, Strong> (May 2019)** **9 marks**

Day	Outlook	Temperature	Humidity	Wind	Play
Day 1	Sunny	Hot	High	Weak	NO
Day 2	Sunny	Hot	High	Strong	NO
Day 3	Overcast	Hot	High	Weak	YES
Day 4	Rain	Mild	High	Weak	YES
Day 5	Rain	Cool	Normal	Weak	YES
Day 6	Rain	Cool	Normal	Strong	NO
Day 7	Overcast	Cool	Normal	Strong	YES
Day 8	Sunny	Mild	High	Weak	NO
Day 9	Sunny	Cool	Normal	Weak	YES
Day 10	Rain	Mild	Normal	Weak	YES
Day 11	Sunny	Mild	Normal	Strong	YES
Day 12	Overcast	Mild	High	Strong	YES
Day 13	Overcast	Hot	Normal	Weak	YES
Day 14	Rain	Mild	High	Strong	NO

Given a Hypothesis (C) and evidence (X), Bayes' Theorem states that the relationship between the probability of the hypothesis before getting the evidence, $P(C)$, and the probability of the hypothesis after getting the evidence, $P(C|X)$, is :

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)}$$

For this reason, $P(H)$ is called the prior probability, while $P(H|E)$ is called the posterior probability. The factor that relates the two, $P(H|E)/P(E)$, is called the likelihood ratio.

Using these terms, Bayes' theorem can be rephrased as:

"The posterior probability equals the prior probability times the likelihood ratio."

ie,

$$Posterior = \frac{Likelihood \times Prior}{Evidence}$$

Here we have our data, which comprises 4 attributes: the day, outlook, humidity, and wind conditions. The final column is 'Play,' i.e., can we play outside, which we have to predict.

Now, we will create "look-up tables" for each of these attributes, and write in the probability that a game of tennis will be played based on this attribute.

In these tables we have to note that there are 5 cases of not being able to play a game, and 9 cases of being able to play a game.

Outlook	**Play = Yes**	**Play = No**	**Total**
Sunny	2/9	3/5	5/14
Overcast	4/9	0/5	4/14
Rain	3/9	2/5	5/14

Temperature	Play = Yes	Play = No	Total
Hot	2/9	2/5	4/14
Mild	4/9	2/5	6/14
Cool	3/9	1/5	4/14

Humidity	Play = Yes	Play = No	Total
High	3/9	4/5	7/14
Normal	6/9	1/5	7/14

Wind	Play = Yes	Play = No	Total
Strong	3/9	3/5	6/14
Weak	6/9	2/5	8/14

We also must note the following probabilities for $P(C)$:

$$P(Play = Yes) = 9/14$$
$$P(Play = No) = 5/14$$

Testing

Now, we are in the testing phase. For this, say we were given a new instance, and we want to know if we can play a game or not, then we need to lookup the results from the tables above. So, this new instance is:

$$X = (Outlook = Sunny, Temperature = Cool, Humidity = High, Wind = Strong)$$

Firstly we look at the probability that we can play the game, so we use the lookup tables to get:

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

Next we consider the fact that we cannot play a game:

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

Then, using those results, you have to multiple the whole lot together. So you multiple all the probabilities for Play=Yes such as:

$$P(X|Play = Yes)P(Play = Yes) = (2/9)*(3/9)*(3/9)*(3/9)*(9/14) = 0.0053$$

And this gives us a value that represents $P(X|C)P(C)$, or in this case $P(X|Play = Yes)P(Play = Yes)$.
We also have to do the same thing for $Play = No$:

$$P(X|Play = No)P(Play = No) = (3/5)*(1/5)*(4/5)*(3/5)*(5/14) = 0.0206$$

Finally, we have to divide both results by the evidence, or $P(X)$. The evidence for both equations is the same, and we can find the values we need within the 'Total' columns of the look-up tables.
Therefore:

$$P(X) = P(Outlook = Sunny)*P(Temperature = Cool)*P(Humidity = High)*P(Wind =$$

$$P(X) = (5/14) * (4/14) * (7/14) * (6/14)$$

$$P(X) = 0.02186$$

Then, dividing the results by this value:

$$P(Play = Yes|X) = 0.0053/0.02186 = 0.2424$$

$$P(Play = No|X) = 0.0206/0.02186 = 0.9421$$

So, given the probabilities, can we play a game or not?
To do this, we look at both probabilities and see which once has the highest value, and that is our answer.
Therefore:

$$P(Play = Yes|X) = 0.2424$$

$$P(Play = No|X) = 0.9421$$

Since 0.9421 is greater than 0.2424 then the answer is 'no', we cannot play a game of tennis today.

16. **Compare Unsupervised Learning and Reinforcement learning with examples. (Dec 2018)** **5 marks**

UNSUPERVISED LEARNING	REINFORCEMENT LEARNING
Unsupervised Learning deals with clustering and associative rule mining problems.	Reinforcement Learning deals with exploitation or exploration, Markov's decision processes, Policy Learning, Deep Learning and value learning.
In Unsupervised Learning the input data is unlabelled.	The data is not predefined in Reinforcement Learning.
Unsupervised Learning explores patterns and predicts the output.	Reinforcement Learning follows a trial and error method.
Algorithms used in Unsupervised Learning are K – Means, C – Means, Apriori.	In Reinforcement Learning Q – Learning, SARSA algorithms are used.
Example : Recommendation System, Anomaly Detection.	Example : Self Driving Cars, Gaming, Healthcare.

17. **What are the benefits of pruning in decision tree induction? Explain different approaches to tree pruning. (Sept 2020(S))** **5 marks**

Lack of data points makes it difficult to predict correctly the class of 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.Decision trees can suffer from repetition and replication, making them overwhelming to interpret.

Repetition occurs when an attribute is repeatedly tested along a given branch of the tree

In replication, duplicate subtrees exist within the tree.

These situations can impede the accuracy and comprehensibility of a decision tree.
Pruned trees:

(a) These tend to be smaller and less complex and, thus, easier to comprehend.

(b) They are usually faster and better at correctly classifying independent test data than unpruned trees.

(c) Pruned trees tend to be more compact than their unpruned counterparts.

There are two approaches:

(a) Pre-pruning
In the pre-pruning approach, a tree is "pruned" by halting its construction early (e.g. by deciding not to further split or partition the subset of training tuples at a given node).

(b) Post-pruning
In post-pruning approach,a subtree at a given node is pruned by removing its branches and replacing it with a leaf.

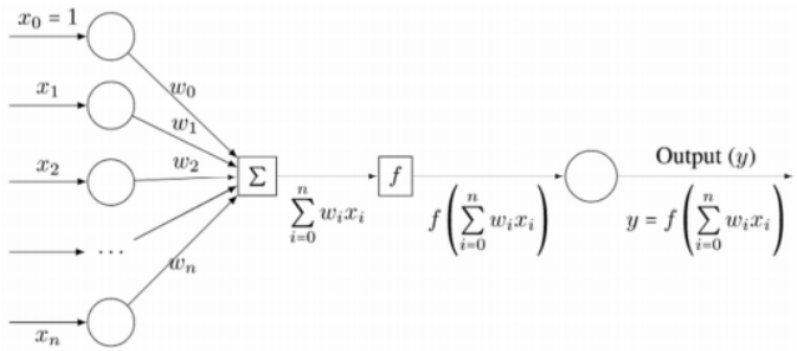
18. **Calculate the output y of a three input neuron with bias. The input feature vector is (x1, x2,x3)= (0.8,0.6,0.4) and weight values are [w1,w2,w3, b] [0.2, 0.1, -0.3, 0.35]. Use binary Sigmoid function as activation function. (Sept 2020) 4 marks**

An artificial neuron is a mathematical function conceived as a model of biological neurons. Artificial neurons are elementary units in an

artificial neural network. The artificial neuron receives one or more inputs and sums them to produce an output. Each input is separately weighted, and the sum is passed through a function known as an activation function or transfer function.

The output y of a neural network is given by f($\sum_{i=1}^{n} w_i x_i$+b) . Where f is the activation function. For binary Sigmoid function as activation function,

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



Answer

First we calculate sum $\Sigma_{n=1}^{3} w_i x_i$

$\Sigma_{n=1}^{3} w_i x_i + b = 0.8 * 0.2 + 0.6 * 0.1 + 0.4 * (-0.3) + 0.35 = 0.45$

$Outputy = f(sum_{i=1}^{3} w_i x_i + b)$

Where f is binary Sigmoid function

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

$y = f(0.45) = 0.61$

19. **Use K Means clustering to cluster the following data into two groups. Assume cluster centroids are m1=2 and m2=4. The distance function used is Euclidean distance. {2, 4, 10, 12, 3, 20, 30, 11, 25} (May 2019)** **6 marks**

n = 9

Data: {2, 4, 10, 12, 3, 20, 30, 11, 25}

m1 = 2, m2 = 4
$distance1 = \|xi - m1\|$, $distance2 = \|xi - m2\|$

Iteration 1:

xi	m1	m2	distance1	distance2	Nearest cluster
2	2	4	0	2	1
3	2	4	1	1	1
4	2	4	2	0	2
10	2	4	8	6	2
11	2	4	9	7	2
12	2	4	10	8	2
20	2	4	18	16	2
25	2	4	23	21	2
30	2	4	28	26	2

m1 = 2.5, m2 = 16

Iteration 2:

xi	m1	m2	distance1	distance2	Nearest cluster
2	2.5	16	0.5	14	1
3	2.5	16	1.5	13	1
4	2.5	16	2.5	12	1
10	2.5	16	7.5	6	2
11	2.5	16	8.5	5	2
12	2.5	16	9.5	4	2
20	2.5	16	17.5	4	2
25	2.5	16	22.5	9	2
30	2.5	16	27.5	14	2

m1 = 3, m2 = 18

Iteration 3:

xi	m1	m2	distance1	distance2	Nearest cluster
2	3	18	1	16	1
3	3	18	0	15	1
4	3	18	1	14	1
10	3	18	7	8	2
11	3	18	8	7	2
12	3	18	9	6	2
20	3	18	17	2	2
25	3	18	22	7	2
30	3	18	27	12	2

m1 = 4.75, m2 = 19.6

Iteration 4:

xi	m1	m2	distance1	distance2	Nearest cluster
2	4.75	19.6	2.75	17.6	1
3	4.75	19.6	1.75	16.6	1
4	4.75	19.6	0.75	15.6	1
10	4.75	19.6	5.25	9.6	1
11	4.75	19.6	6.25	8.6	1
12	4.75	19.6	7.25	7.6	1
20	4.75	19.6	15.25	0.4	2
25	4.75	19.6	20.25	5.4	2
30	4.75	19.6	25.25	10.4	2

m1 = 7, m2 = 25

Iteration 5:

xi	m1	m2	distance1	distance2	Nearest cluster
2	7	25	5	23	1
3	7	25	4	22	1
4	7	25	3	21	2
10	7	25	3	15	2
11	7	25	4	14	2
12	7	25	5	13	2
20	7	25	13	5	2
25	7	25	18	0	2
30	7	25	23	5	2

m1 = 7, m2 = 25

There is no change between iteration 4 and 5. So the clusters are: k1 = {2, 3, 4, 10, 11, 12} and k2 = {20, 25, 30}, centroids are: m1 = 7 and m2 = 25.

20. **What are the different methods for measuring classifier performance? (May 2019)** **4 marks**

For classification, especially for two-class problems, a variety of measures has been proposed

	Predicted Class		
True class	Positive	Negative	Total
Positive	tp: true positive	fn: false negative	p
Negative	fp: false positive	tn: true negative	n
Total	p'	n'	N

After we classify a data set based on a chosen learning method using positive and negative examples, we can compare to see how many examples from test are classified correctly. There are four possible cases, as shown in table.

(a) For a positive example, if the prediction is also positive, this is a **true positive (tp)**

(b) If our prediction is negative for a positive example, this is a **false negative (fn)**

(c) For a negative example, if the prediction is also negative, we have a **true negative(tn)**

(d) If we predict a negative example as positive it is a **false positive (fp)**.

If we predict a negative example as positive it a false positive (fp). Example consider an authentication model, users log on to their accounts by voice. A false positive is wrongly logging on an impostor and a false negative is refusing a valid user. It is clear that the two type of errors are not equally bad; the former is much worse

True positive rate, **tp-rate**, also known as **hit rate**, measures what proportion of valid users we authenticate and false positive rate, **fp-rate**, also known as **false alarm rate**, is the proportion of impostors we wrongly accept.

Name	Formula
error	$(\mathrm{f}_p + f_n)/N$
accuracy	$(\mathrm{t}_p + t_n)/N = 1 - error$
$\mathrm{t}_p rate$	t_p/p
$\mathrm{f}_p rate$	f_p/n
precision	t_p/p'
recall	$\mathrm{t}_p/p = t_p rate$
sensitivity	$\mathrm{t}_p/p = t_p rate$
specificity	$\mathrm{t}_n/n = 1 - f_p rate$

Precision, Recall, Sensitivity, Specificity and Class Confusion Matrix

We consider an example from information retrieval ,

Consider there is a database of records; we make a query, for example, by using some keywords, and a system (basically a two-class classifier) returns a number of records.

In the database, there are relevant records and for a query, the system may retrieve some of them (true positives) but probably not all (false negatives); it may also wrongly retrieve records that are not relevant (false positives).

The set of relevant(L) and retrieved records(R) can be visualized using

a Venn diagram, as shown in figure a.

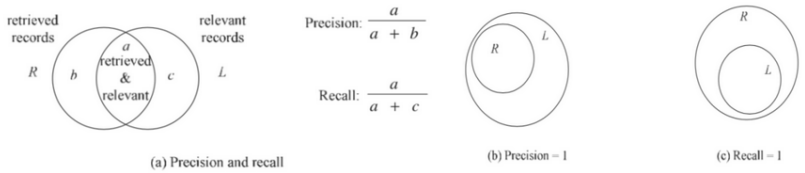


Fig (a) Definition of precision and recall using Venn diagrams. Fig b (b) Precision is 1; ie all the retrieved records are relevant but there may be relevant ones not retrieved. Fig (c) Recall is 1; ie all the relevant records are retrieved but there may also be irrelevant records that are retrieved. L is the set of relevant records and R is the set of Retrieved records.

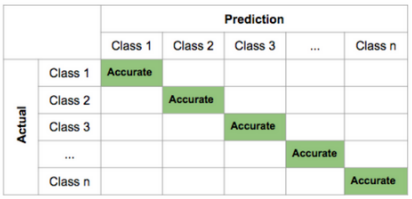
Precision is the number of retrieved and relevant records divided by the total number of retrieved records; if precision is 1, all the retrieved records may be relevant but there may still be records that are relevant but not retrieved.

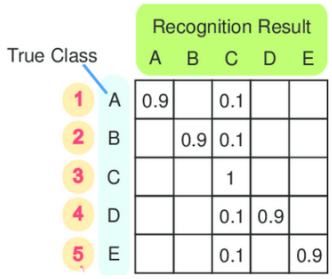
Recall is the number of retrieved relevant records divided by the total number of relevant records; even if recall is 1, all the relevant records may be retrieved but there may also be irrelevant records that are retrieved, as shown in figure c.

Sensitivity and specificity From another perspective but with the same aim, there are the two measures of *sensitivity* and *specificity*. Sensitivity is the same as tp-rate and recall. Specificity is how well we detect the negatives, which is the number of true negatives divided by the total number of negatives; this is equal to 1 minus the false alarm rate. One can also draw a sensitivity vs. specificity curve using different thresholds.

Class confusion matrix: In the case of no of classes K ¿ 2 classes, if we are using 0/1 error, the class confusion matrix is a K $\times$ K matrix whose

entry (i, j) contains the number of instances that belong to C_i but are assigned to C_j . Ideally, all off-diagonals should be 0, for no misclassification. The class confusion matrix allows us to pinpoint what types of misclassification occur, namely, if there are two classes that are frequently confused. Or, one can define K separate two-class problems, each one separating one class from the other K-1.





Consider the example – there are five classes (A, B, C, D, E), the diagonal shows percentage of classes that are rightly classified. Index(1, 3) shows a miss classification i.e – .1 examples from class A is classified into Class C. Similarly ((2, 3), (4, 3),(5, 3)) also shows unclassified percentage.

21. **What is meant by k-fold cross validation. Given a data set with 1200 instances, How k- fold cross validation is done with k=1200.? (Dec 2018) 4 marks**

(a) In K-fold cross-validation, the dataset X is divided randomly into K equal- sized parts, Xi , i = 1,...,K. To generate each pair, we keep one of the K parts out as the validation set and combine the remaining K - 1 parts to form the training set. Doing this K times, each time leaving out another one of the K parts out, hence by we get K pairs each

(b) When K = 1200

$V_1 = X_1 \quad T_1 = X_2\ U\ X_3\ U\ X_4 \ldots\ldots\ldots U\ X_{1200}$

$V_2 = X_2 \quad T_1 = X_1\ U\ X_3\ U\ X_4 \ldots\ldots\ldots U\ X_{1200}$

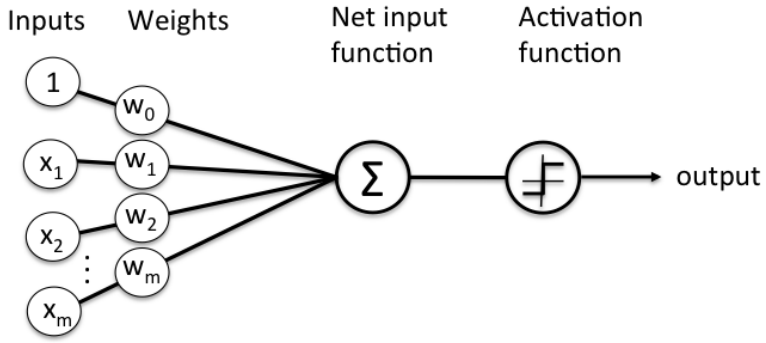
$V_3 = X_3 \quad T_1 = X_1\ U\ X_2\ U\ X_4 \ldots\ldots\ldots U\ X_{1200}$

.....
.....
.....
.....
$V_{1200} = X_{1200} \quad T_1 = X_1 \, U \, X_2 \, U \, X_3 ..... U \, X_{1199}$

where
X={..... is the subset of all the datasets}
T={ ..... is the subset of all the K division Combination of the dataset}
V={..... is the subset of Test values of the datasets}

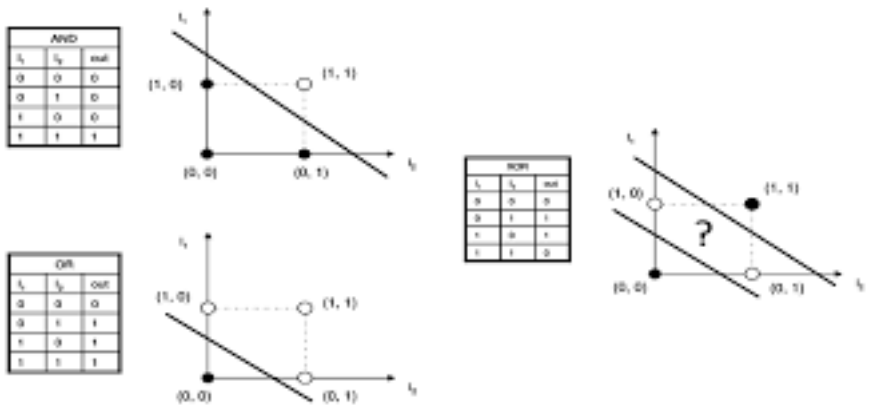
22. **Why does a single perceptron cannot simulate simple XOR function? Explain how this limitation is overcome? (Dec 2019) 4 marks**

Perceptron
The perceptron can be represented by the schematic shown in the figure below.



As we can see, it calculates a weighted sum of its inputs and thresholds it with a step function. Geometrically, this means the perceptron can separate its input space with a hyperplane. That's where the notion

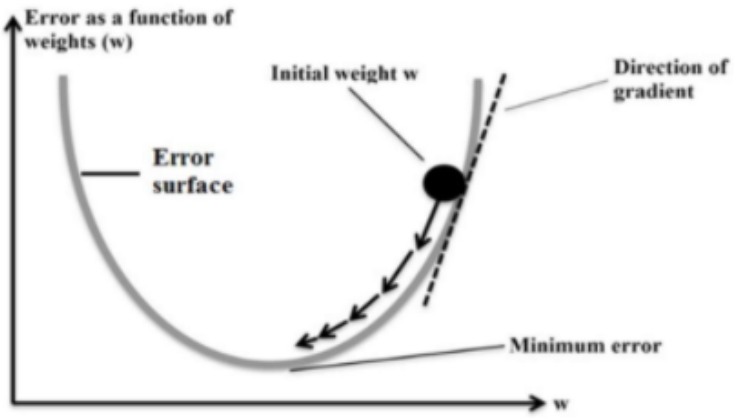
that a perceptron can only separate linearly separable problems came from. Since the XOR function is not linearly separable, it really is impossible for a single hyperplane to separate it.



So, a single perceptron cannot simulate simple XOR function.
The delta rule is used when the training data set is not linearly separable.

Gradient Descent and Delta Rule

The technique used to determine how much a weight should be changed is known as gradient descent method. At every stage of the computation, the error is a function of the weights. If we plot the error against the weights, we get a higher dimensional analog of something like a curve or surface. At any point on this surface, the gradient suggests how steeply the error will be reduced or increased for a change in the weight. The algorithm will attempt to change the weights that result in the greatest reduction in error.
A simplified model of the error surface showing the direction of gradient.



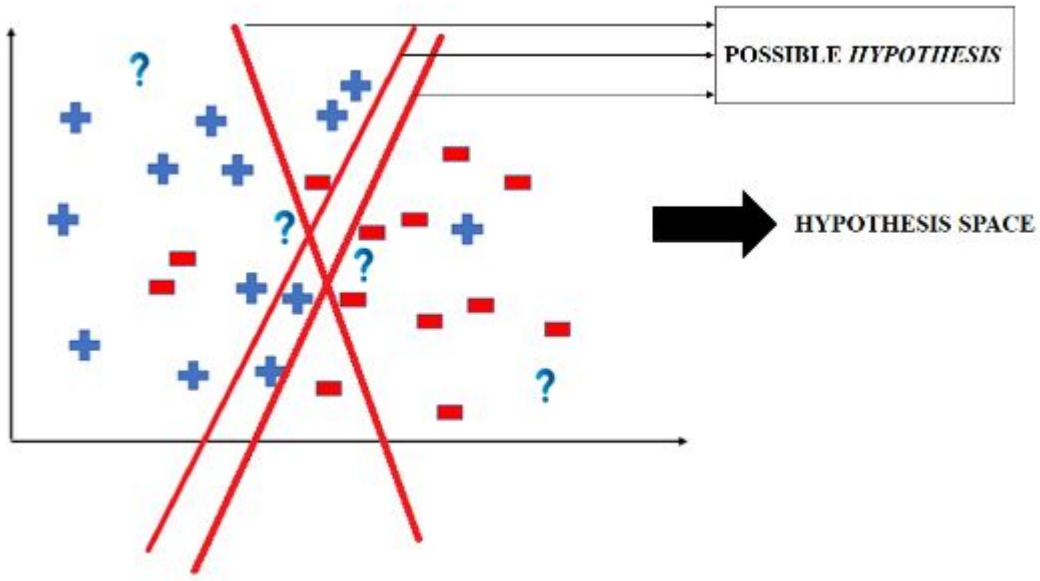
The method used to calculate the adjusted weights is known as the delta rule.
The rule for computing the adjusted weights can be succinctly stated as follows. Let w be the weight and w^+ its adjusted weight. Let E be the total sum of squares of errors . Then w^+ is computed by

$$w^+ = w - \eta \frac{\partial E}{\partial w}$$

Here $\frac{\partial E}{\partial w}$ is the gradient of E with respect to w; that is, the rate at which E is changing with respect to (The set of all such gradients specifies the direction in which E is decreasing the most rapidly, that is, the direction of quickest descent.)

23. **Define the terms Hypothesis space and Version space. Illustrate with an example. (May 2019) 4 marks**

Hypothesis space:
For a binary classification problem , There may be more than one hypothesis present, such a set of hypothesis is called a hypothesis space.



Consistency:

Let 'x' be an example of classification problem and C(x) denote the class label to ,'x', i.e, C(x) = 1 or 0.

Let D be the training set; 'h' be the hypothesis for the problem, h(x) be the class label assigned to 'x' by h.

h is consistent, if h(x) = C(x) , $\forall$ D.

Example:

$$\left.\begin{array}{l} \text{h1 If x} > 17 \text{ then 1 else 0} \\ \text{h2 If x} \geq 18 \text{ then 1 else 0} \\ \text{h3 If x} \geq 20 \text{ then 1 else 0} \\ \text{h4 If x} \geq 23 \text{ then 1 else 0} \end{array}\right\} \text{Hypothesis space}$$

Version space:

Consider a binary classification problem , Let D be the data set ,H be the Hypothesis space,Then version space of the problem with respect to D and the space H consistent with D,i.e,

$VS_{D,H}$ = { h $\in$ H ,h(x)=C(x) }

Feature	Class label
27	1
15	0
23	1
30	1
25	1
17	0
12	0
31	1
6	0

Example:

x1	x2	Class label
6	8	+1
12	14	-1
3	2	+1
18	17	-1

h1 if x1 $<$ 10 and X2 $<$ 10 then +1 else -1
h2 if x1+x2 $<$ 15 then +1 else -1
h3 if x1 $>$ 10 And x2 $>$ 10 then -1 else +1

Collection of consistent hypothesis is called version space.

24. **Illustrate the two approaches used in subset selection. (May 2019) 6 marks**

There are two approaches to Subset Selection

i. Forward Selection
ii. Backward selection

Let us denote by F, a feature set of input dimensions, xi, i = 1,...,d.

E(F) denotes the error incurred on the validation sample when only the inputs in F are used. Depending on the application, the error is either the mean square error or misclassification error.

FORWARD SELECTION: we start with no input variables and add them one by one, at each step adding the one that decreases the error the most, until any further addition does not decrease the error (or decreases it only slightly) Checking of the error is done on a validation set distinct from the training set because we want to test the generalization accuracy. With more features, generally we have lower training error, but not necessarily lower validation error.
In sequential forward selection, the steps are

i. We start with no features: F = ∅.

ii. At each step, for all possible xi, we train our model on the training set and calculate E(F ∪ xi) on the validation set.

iii. Then, we choose that input xj that causes the least error j =argminE(F ∪xi) i and we add xj to F if E(F∪xj) < E(F)

iv. We stop when

[U+25CF] if adding any feature does not decrease E.

[U+25CF] We may even decide to stop earlier if the decrease in error is too small, where there is a user-defined threshold that depends on the application constraints. Adding a new feature introduces the cost of observing the feature, as well as making the classifier/regressor more complex.

Complexity analysis of forward search

This process may be costly because to decrease the dimensions from d to k, we need to train and test the system d+(d−1)+(d−2)+···+(d−k) times, which is O(d2).

Disadvantage with Forward Selection - Forward selection is a local search procedure and does not guarantee finding the optimal subset, namely, the minimal subset causing the smallest error. For example, xi and xj by themselves may not be good but together may decrease the error a lot, but because this algorithm is greedy and adds attributes one by one, it may not be able to detect this. It is possible to generalize and add multiple features at a time, instead of a single one, at the

expense of more computation.

Solution -

We can backtrack and check which previously added feature can be removed after a current addition, thereby increasing the search space, but this increases complexity.

floating search methods -

in which the number of added features and removed features can also change at each step.

BACKWARD SELECTION - we start with all variables and remove them one by one, at each step removing the one that decreases the error the most (or increases it only slightly), until any further removal increases the error significantly.

In sequential backward selection, steps are

i. we start with F containing all features

ii. At each step, for all possible xi, we train our model on the training set by removing one attribute at a time and calculate E(F - xi) on the validation set and, and we remove the one that causes the least error

iii. Then, we choose that input xj that causes the least error j =argmin E(F −xi) i and we remove xj from F if E(F−xj) ¡ E(F)

iv. We stop [U+25CF] if removing a feature does not decrease the error. To decrease complexity, we may decide to remove a feature if its removal causes only a slight increase in error. All the variants possible for forward search are also possible for backward search.

The complexity of backward search has the same order of complexity as forward search, except that training a system with more features is more costly than training a system with fewer features, and forward search may be preferable especially if we expect many useless features. Subset selection is supervised method because outputs are used by the regressor or classifier to calculate the error

In an application like face recognition, feature selection is not a good

method for dimensionality reduction because individual pixels by themselves do not carry much discriminative information; it is the combination of values of several pixels together that carry information about the face identity. This is done by feature extraction method like Principal Component Analysis.

25. **Compare cross validation with bootstrapping techniques (Sept 2020)**

. 4 marks

Both cross validation and bootstrapping are resampling methods.

Cross validation

Cross validation resamples without replacement and thus produces surrogate data sets that are smaller than the original.

To test the performance of a classifier, we need to have a number of training/validation set pairs from a dataset X. Cross- validation is a technique to evaluate predictive models by partitioning the original sample 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. Then the function is used to predict the output values for the data in the testing set (it has never 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.

Cross validation is a procedure for validating a model's performance, and it is done by splitting the training data into k parts. We assume that the k-1 parts is the training set and use the other part is our test set. We can repeat that k times differently holding out a different part of the data every time. Finally, we take the average of the k scores as our performance estimation. Cross validation can suffer from bias or variance. Increasing the number of splits, the variance will increase too and the bias will decrease. On the other hand, if we decrease the number of splits, the bias will increase and the variance will decrease.

Bootstrapping in machine learning

The term bootstrap sampling refers to the process of "random sampling with replacement". In Machine Learning, bootstrapping is the

process of computing performance measures using several 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.

26. **Explain the various method to perform cross validation (May 2019) 4 marks**

Cross-validation is a statistical technique which involves partitioning the data into subsets, training the data on a subset and use the other subset to evaluate the model's performance.

Types of cross validation

- LOOCV (Leave one out cross-validation)
- K Fold
- Stratified cross-validation
- Time series cross-validation

1. Leave one out cross validation — LOOCV

In LOOCV we divide the data set into two parts. In one part we have a single observation, which is our test data and in the other part, we have all the other observations from the dataset forming our training data. If we have a data set with n observations then training data contains n-1 observation and test data contains 1 observation. This process is iterated for each data point as shown below. Repeating this process n times generates n times Mean Square Error(MSE).

2. K fold cross validation

This technique involves randomly dividing the dataset into k groups or folds of approximately equal size. The first fold is kept for testing and the model is trained on k-1 folds. The process is repeated K times and each time different fold or a different group of data points are used for validation. As we repeat the process k times, we get k times Mean

Square Error(MSE).So k-Fold CV error is computed by taking average of the MSE over K folds.

3. Stratified cross-validation

Stratification is a technique where we rearrange the data in a way that each fold has a good representation of the whole dataset. It forces each fold to have at least m instances of each class. This approach ensures that one class of data is not overrepresented especially when the target variable is unbalanced. For example in a binary classification problem where we want to predict if a passenger on Titanic survived or not. we have two classes here Passenger either survived or did not survive. We ensure that each fold has a percentage of passengers that survived and a percentage of passengers that did not survive.

4. Time series cross-validation

Splitting time series data randomly does not help as the time-related data will be messed up. If we are working on predicting stock prices and if we randomly split the data then it will not help. Hence we need a different approach for performing cross-validation. For time series cross-validation we use forward chaining also referred as rolling-origin. Origin at which the forecast is based rolls forward in time. In time series cross-validation each day is a test data and we consider the previous day's data is the training set. D1, D2, D3 etc. are each day's data and days highlighted in blue are used for training and days highlighted in yellow are used for test. we start training the model with a minimum number of observations and use the next day's data to test the model and we keep moving through the data set. This ensures that we consider the time series aspect of the data for prediction.

27. **Discuss the issues involved in decision tree learning. (May 2019)** **5 marks**

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 t outliers. It creates poor accuracy for unseen samples.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 over[U+FB01]t the training examples

i) Overfitting due to noise in the training data.

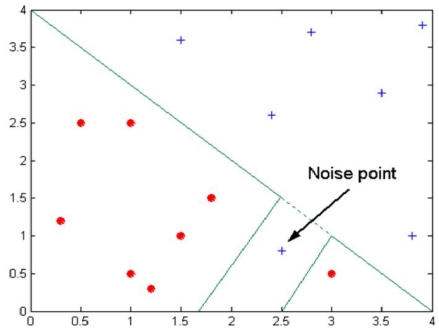
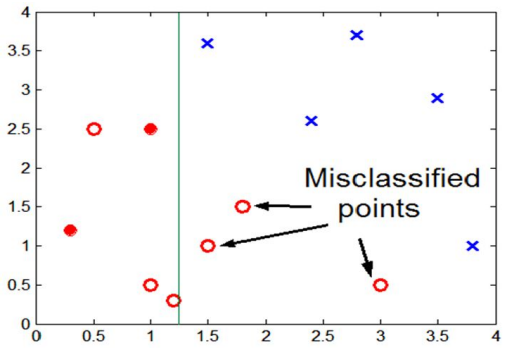


Figure shows that decision boundary is distorted by noise point.
ii) Overfitting due to Insufficient Examples (sample is too small) Lack of data points makes it difficult to predict correctly the class labels of that region



The main approach to avoid over[U+FB01]tting 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 [U+FB01]nal classi[U+FB01]er, and hence improves predictive accuracy by the reduction of over[U+FB01]tting.
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

28. **Explain (a) Hypothesis space (b) Version space (c) Most Generalhypothesis(d) Most specific hypothesis in the context of a classifica-tion problem. (Dec 2019) 4 marks**

(a) Hypothesis space
Hypothesis space is the set of all the possible legal hypothesis. This is the set from which the machine learning algorithm would determine the best possible (only one) which would best describe the target function or the outputs.

(b) Version space
Any h element of 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 subset of hypotheses from H consistent with all training examples in D.

(c) Most Generalhypothesis
General Hypothesis is the largest rectangle we can draw that includes all the positive examples and none of the negative examples.

(d) Specific hypothesis
Specific Hypothesis is the tightest rectangles that includes all the possible examples and none of the negative examples.
Hypothesis h=S as the induced cl

29. **Define the following terms (a) sensitivity (b) Specificity (c) Preci-sion (d) Accuracy for a classification problem. (Dec 2018) 4 marks**

(a) Sensitivity
Sensitivity is a measure of the proportion of actual positive cases that got predicted as positive or true positive. Sensitivity = (True Positive)/(True Positive + False Negative)

(b) Specificity
It is defined as the proportion of actual negatives, which got predicted as the negative or true negative. Specificity = (True Negative)/(True Negative + False Positive)

(c) Precision
Precision is the number of correct positive results divided by the number of positive results predicted by the classifier.

(d) Accuracy
Accuracy is the ratio of number of correct predictions to the total number of input samples.

30. **Explain the basic elements of a Hidden Markov Model (HMM).Listany two applications of HMM. (Sep 2020) 4 marks**

An HMM can be completely described by the following elements:-
(a) N is the number of the hidden states in the model. Individual states are denoted as S = S1, S2, . . . , SN , and the state at time t as qt.
(b) M is the number of distinct observation symbols for each state and is called the alphabet size and correspond to the physical output of the

system which is modelled.
(c) The state transition probability distribution A = aij, where aij = P[qt = Sj —qt−1 = Si], 1 ≤ i, j ≤ N , aij ≥ 0
(d) The observation symbol probability distribution in state j, B = bj (k),where bj (k) = P[vk at t—qt = Sj], 1 ≤ j ≤ N and 1 ≤ k ≤ M
(e) The initial state distribution, describing the probability of beginning the state sequence in a certain initial state. i = P[q1 = Si], 1 ≤ i ≤ N
(f) The observation sequence is denoted as O = O1O2 · · · OT . represents the complete parameter set of a model, where = (A, B,).
HMM can be applied in Cryptanalysis, Speech Recognition, etc...

31. **Differentiate between bagging, boosting and voting. (May 2019) 4 marks**

Bagging
Building multiple models, typically of the same type from different subsamples of the training dataset.

Boosting
Building multiple models, typically of the same type each of which learns to fix the prediction errors of a prior model in the chain.

Voting
Building multiple models, typically of differing types and simple statistics like calculating the mean are used to combine predictions.

32. **State Occam's razor principle. Illustrate its necessity in learning hypothesis. (Dec 2018) 4 marks**

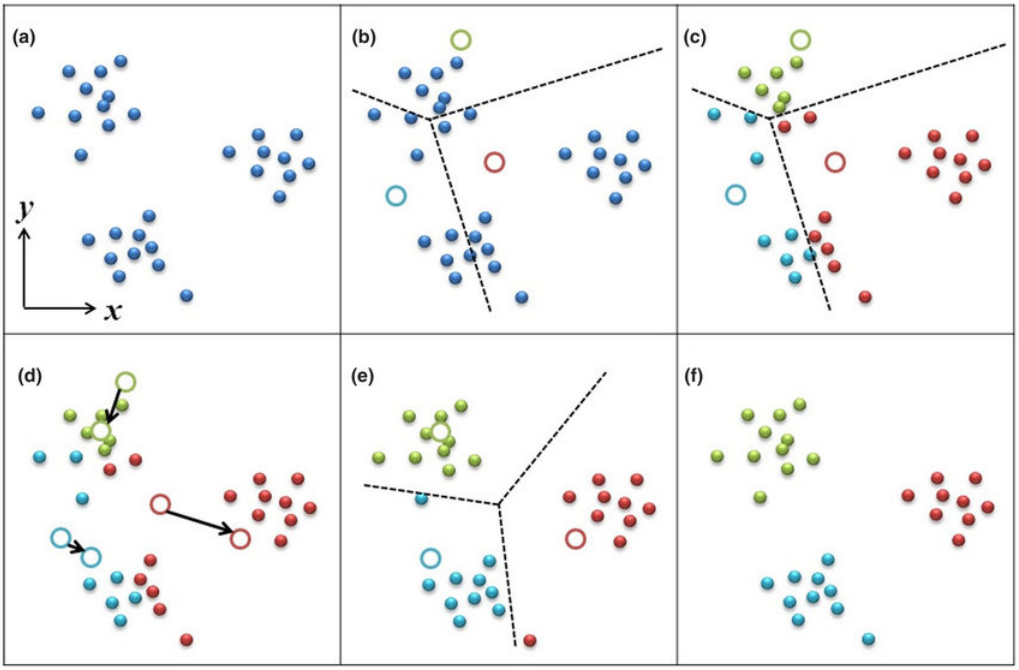
A simple model would generalize better than a complex model. This principle is known as Occam's razor, which states that simpler explanations are more plausible and any unnecessary complexity should be shaved off. A common preference relation on the whole hypothesis space is to prefer in the spirit of Occam's razor simple hypotheses over complicated ones. When choosing a boundary between positive and negative training examples, a hyperplane is e.g. preferred over a non-differentiable surface. Especially, in the presence of noise (i.e. when the labels of the training data may be wrong with some probability) Occam's razor is often used to avoid the danger of overfitting the training data, that is, to choose a hypothesis that perfectly fits the training

data but is very complex and hence often does not generalize well. There has been some discussion on the validity of Occam's razor (and also of the more or less synonymous overfitting avoidance) also in the machine learning community. While Occam's razor often remains a rather vague principle, there are some theoretical results and attempts to clarify what Occam's razor in machine learning exactly is. Thus, it has been argued [Domingos, 1998] that the term "Occam's razor" is actually used for two different principles in the machine learning literature. POSTULATE 7 Occam's first razor. Given two models with the same error on the whole instance space X, choose the simpler one. POSTULATE 8 Occam's second razor. Given two models with the same error on the training sample, choose the simpler one.

33. **Write down the major differences between K-means clustering and hierarchical clustering. (May 2019) 4 marks**

K-Means Clustering

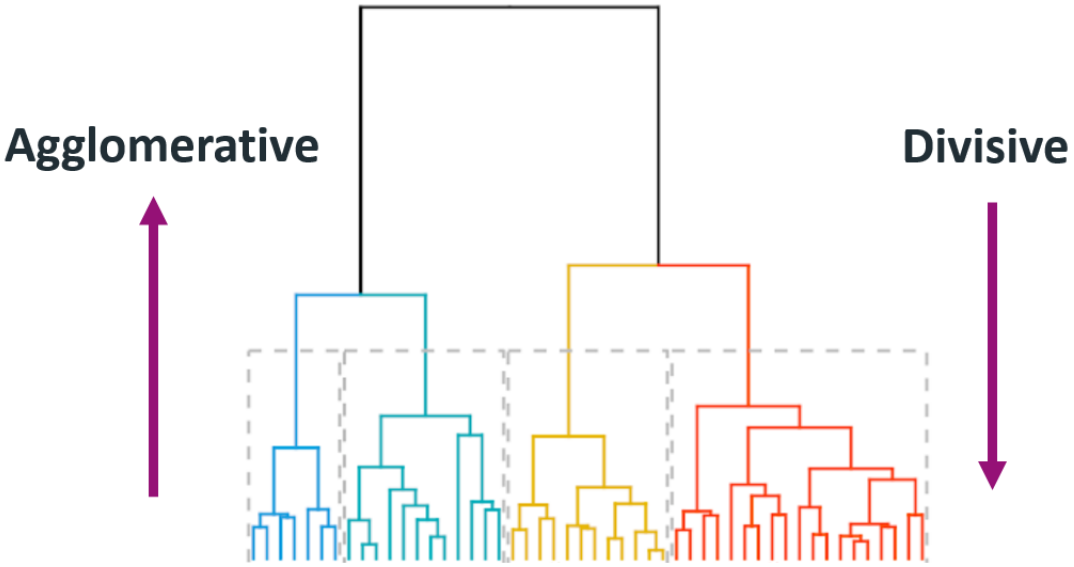
K-Means clustering intends to partition n objects into k clusters in which each object belongs to the cluster with the nearest mean. This method produces exactly k different clusters of greatest possible distinction. The best number of clusters k leading to the greatest separation (distance) is not known as a priori and must be computed from the data. The main objective of the K-Means algorithm is to minimize the sum of distances between the points and their respective cluster centroid.



Hierarchical Clustering

Hierarchical clustering, also known as hierarchical cluster analysis, is an unsupervised clustering algorithm which involves creating clusters that have predominant ordering from top to bottom. The algorithm groups similar objects into groups called clusters. The endpoint is a set of clusters, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other. This clustering technique is divided into two types:

- Agglomerative Hierarchical clustering Technique
- Divisive Hierarchical clustering Technique



Difference between K-Means and Hierarchical clustering

- Hierarchical clustering can't handle big data well but K-Means clustering can. This is because the time complexity of K-Means is linear i.e. O(n) while that of hierarchical clustering is quadratic i.e. $O(n^2)$.
- In K-Means clustering, since we start with random choice of clusters, the results produced by running the algorithm multiple times might differ. While results are reproducible in Hierarchical clustering.
- K-Means clustering requires prior knowledge of K i.e. no. of clusters you want to divide your data into. But you can stop at whatever number of clusters you find appropriate in hierarchical clustering by interpreting the dendrogram.
- K-means clustering produces a single partitioning. Hierarchical clustering can give different partitioning depending on the level of resolution we are looking at.
- The result of K-means is unstructured, but that of hierarchal is more interpretable and informative.
- K-means clustering a simply a division of the set of data objects

into non-overlapping subsets(clusters) such that each data object is in exactly one subset. A hierarchical clustering is a set of nested clusters that are arranged as a tree.

- K-Means clustering is found to work well when the structure of the clusters is hyper spherical (like circle in 2D, sphere in 3D). Hierarchical clustering doesn't work well when the shape of the clusters is hyper spherical.

34. . **Calculate the dissimilarity between two data points A(2,3,4) and B(4,3,5) using**
(a) Euclidian distance
(b) Manhattan Distance (Dec 2019) **4 marks**

(a)Euclidean Distance is given by,

$$= \sqrt[2]{(x2-x1)^2+(y2-y1)^2+(z2-z1)^2}$$

where points are (x1, y1, z1) and (x2, y2, z2).
Given A(2,3,4) and B(4,3,5), Euclidean Distance between them is

$$= \sqrt[2]{(4-2)^2+(3-3)^2+(5-4)^2} = \sqrt[2]{4+0+1} = \mathbf{2.2360679775}$$

(b) Manhattan Distance is given by,

$$= |x1-x2|+|y1-y2|+|z1-z2|$$

where points are (x1, y1, z1) and (x2, y2, z2).
Given A(2,3,4) and B(4,3,5), Manhattan Distance between them is

$$= |2-4|+|3-3|+|4-5| = |-2|+|0|+|-1| = 2+0+1 = \mathbf{5}$$

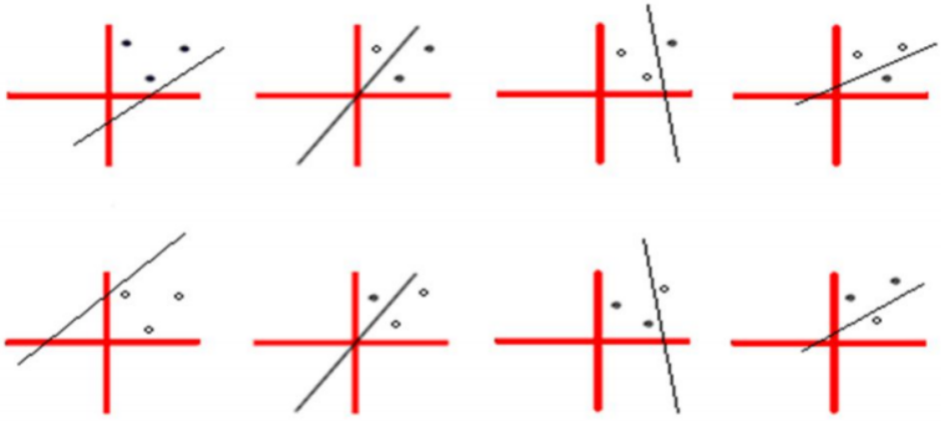
35. **Define VC dimension. Show that VC dimension of a line hypothesis is three. (Dec 2019)** **4 marks**

The **Vapnik-Chervonenkis dimension**, VC(H), of hypothesis space H defined over instance space X is the size of the largest finite subset of X shattered by H. If arbitrarily large finite sets of X can be shattered by H , then

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

For a given decision function (classification function), the VC dimension indicates the maximum number of training sets that can be classified

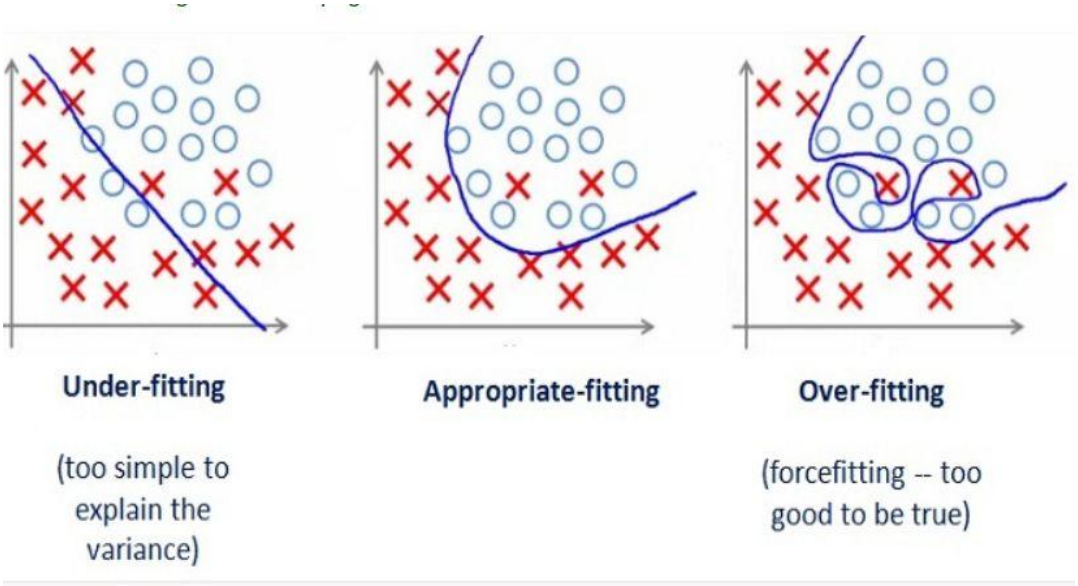
without any error. If there are three points in the same two-dimensional space, it can be shattered with line function machine into eight (2^3) combination without error during training in the following manner.



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.

36. **Explain the concept of Overfitting and Underfitting model with suitable diagrams. (Dec 2019) 4 marks**

When we are working on a data set for predicting or classifying, we calculate accuracy for our model. If the accuracy is low or average, we tends to increase the accuracy by increasing or decreasing data features in our model. Sometimes, this creates our model to perform poorly. It is because of either the model is too simple to describe the data **(underfitting)**, or too complex in describing the data **(overfitting)**.



- When we look at the left graph, it does not cover all points in the graph. Such model gives poor results due to underfitting of the data, which is also called High Bias.
- When we look at the right graph, it covers each and every point in the graph. We might think that it is a very good model, but it is not because it covers all noises and outliers in the data provided. Such model also gives poor results due to overfitting of the data, which is also called High Variance.
- Now look at the middle graph. It covers all important points and leaves out noises and outliers. Hence it gives the best results.
- Bias means how much we ignore the data, so high bias means we ignore too much data. Variance means how much we are dependent on the data, so high variance means we are too much dependent/reliant on the data. In any model, there will be trade off between bias and variance. But we try to achieve the best balance.
- Though underfitting and overfitting leads to poor performance, common problem of these two is overfitting. To limit overfitting, two techniques are used:
  - Validation

– K-fold Cross-Validation

37. **Suppose 10000 patients get tested for flu; out of them, 9000 are actually healthy and 1000 are actually sick. For the sick people, a test was positive for 620 and negative for 380. For the healthy people, the same test was positive for 180 and negative for 8820. Construct a confusion matrix for the data and compute the precision and recall for the data. (Sept 2020) 4 marks**

The basic definitions in this scenario will be like :

- **True positive (TP):** Positive test result that matches reality, i.e, the person is actually sick and tested positive.
- **False positive (FP):** Positive test result that doesn't match reality, i.e, the test is positive but the person is not actually sick.
- **True negative (TN):** Negative test result that matches reality, i.e, the person is not sick and tested negative.
- **False negative (FN):** Negative test result that doesn't match reality, i.e, the test is negative but the person is actually sick.

Using these definitions, we can form the confusion matrix:

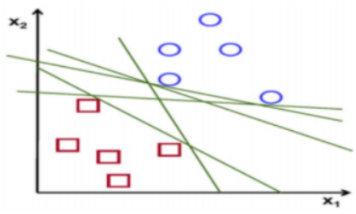
	Actual sick (TRUE)	**Actual sick (FALSE)**
Predicted sick (TRUE)	TP= 620	FP= 180
Predicted sick (FALSE)	FN= 380	TN= 8820

- **Precision (P)**
  = **TP/(TP+FP)**
  = 620/(620+180)
  = 620/800
  = **0.775**
- **Recall (R)**
  = **TP/(TP+FN)**
  = 620/(620+380)

$= 620/1000$
$= \mathbf{0.62}$

38. . **Explain how SVM can be used for classification of linearly separable data. (May 2019, Qn 17(a)) (Mark 6)**

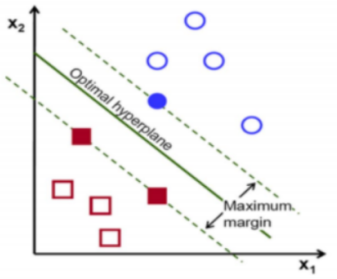
SVM, also known as Support vector machines can be used for both classification and regression problems. But it is mostly used in classification problems. The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. Consider the data points given below, As



you can see in the figure, the two data points can be separated in n number of ways by using a straight line. But our aim is to find out an optimal hyperplane that efficiently classifies the two data points. The main terminologies associated with SVM are:
a) Optimal Hyperplane
b) Marginal Planes
c) Marginal Distance
d) Support vectors

Once the hyperplane is obtained, we obtain two planes that are parallel to the hyperplane and which pass through the data points which is nearest to the hyperplane. Here in the second figure, we can see two parallel planes(Marginal planes) which pass connecting the data points which are nearest to the hyperplane. The distance between the two marginal planes is calculated by $\frac{2}{|w|}$ (where w is a unit vector) and is called Marginal distance.

Support vectors are nothing but the nearest data points to the hyperplane. It can also be defined as the points which pass through the marginal plane. Our aim is to always obtain a generalised model. Therefore we must select the optimal hyperplane as the plane which



is having the maximum marginal distance. The greater the marginal distance more will be the efficiency in classifying data points into the corresponding classes. This is how SVM can be used for the classification of linearly separable data.

39. **Discuss the issues involved in decision tree learning.(May 2019, Qn 16 (b)) (Mark 5)**

(a) Overfitting the data: In decision-tree algorithms, it grows each branch of the tree just deeply enough to perfectly classify the training examples but it can lead to difficulties when there is noise in the data, or when the number of training examples is too small to produce a representative sample of the true target function.

(b) Handling continuous valued attributes: There are following restrictions to attributes for decision tree: - The target attribute whose value is predicted by the learned tree must be discrete valued. - The attributes tested in the decision nodes of the tree must also be discrete valued.

(c) Attributes with many values: There is a natural bias in the information gain measure that favors attributes with many values over those with few values. For example, In case of attribute date,it has so many possible values that it is bound to separate the training examples into very small subsets. Because of this, it will have a very high information gain relative to the training examples.

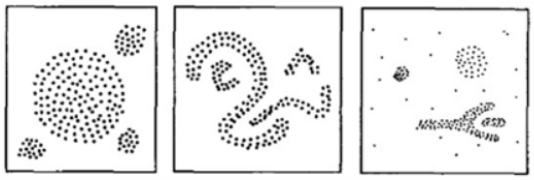
(d) Handling missing attribute values: In certain cases, the available data may be missing values for some attributes. For example, in a medical domain in which we wish to predict the patient outcome

based on various laboratory tests, it may be that the Blood-Test-Result is available only for a subset of the patients.

(e) Handling attributes with differing costs: In some learning tasks, the instance attributes may have associated costs. For example, in learning to classify medical diseases we might describe patients in terms of attributes such as Temperature, Pulse, etc. These attributes vary significantly in their costs, both in terms of monetary cost and cost to patient comfort.

40. **Describe the concept of Density-based clustering and steps involved in DBSCAN Algorithm.(May 2019, Qn 18 (b)) (Mark 6)**

In density-based clustering, clusters are defined as areas of higher density than the remainder of the data set. Objects in these sparse areas - that are required to separate clusters - are usually considered to be noise and border points. The most popular density-based clustering method is DBSCAN (Density-Based Spatial Clustering of Applications with Noise). K means clustering will fail to cluster based on density as it would cluster based on the distance to the nearest centroid. It would obtain different clusters in comparison to density based , Its fails to capture the complex density pattern in the data sets. Fig shows examples of cases where Density based clustering can be applied to capture the complex patterns.



: Clusters of points and noise points not belonging to any of those clusters

DBSCAN algorithm requires two parameters –

(a) eps :It defines the neighborhood around a data point i.e. if the distance between two points is lower or equal to 'eps' then they are considered as neighbors. If the eps value is chosen too small

then large part of the data will be considered as outliers. If it is chosen very large then the clusters will merge and majority of the data points will be in the same clusters. One way to find the eps value is based on the k-distance graph.

(b) MinPts: Minimum number of neighbors (data points) within eps radius. Larger the dataset, the larger value of MinPts must be chosen. As a general rule, the minimum MinPts can be derived from the number of dimensions D in the dataset as, MinPts ¿= D+1. The minimum value of MinPts must be chosen at least 3.

In this algorithm, we have 3 types of data points:(i)Core Point: A point is a core point if it has more than MinPts points within eps. (ii)Border Point: A point which has fewer than MinPts within eps but it is in the neighbourhood of a core point. (iii)Noise or outlier: A point which is not a core point or border point.

Steps involved in DBSCAN Algorithm:

(a) Find all the neighbor points within eps and identify the core points or visited with more than MinPts neighbors.

(b) For each core point if it is not already assigned to a cluster, create a new cluster.

(c) Find recursively all its density connected points and assign them to the same cluster as the core point.A point a and b are said to be density connected if there exist a point c which has a sufficient number of points in its neighbors and both the points a and b are within the eps distance. This is a chaining process. So, if b is neighbor of c, c is neighbor of d, d is neighbor of e, which in turn is neighbor of a implies that b is neighbor of a.

(d) Iterate through the remaining unvisited points in the dataset. Those points that do not belong to any cluster are noise.

41. **What is the significance of optimal separating hyperplanes in SVM? (May 2019, Q7) (Mark 4)**

In a binary classification problem, given a linearly separable data set, the optimal separating hyperplane is the one that correctly classifies all the data while being farthest away from the data points. In this respect, it is said to be the hyperplane that maximizes the margin, defined

as the distance from the hyperplane to the closest data point. The optimal separating hyperplane should not be confused with the optimal classifier is known as the Bayes classifier: the Bayes classifier is the best classifier for a given problem, independently of the available data but unattainable in practice, whereas the optimal separating hyperplane is only the best linear classifier one can produce given a particular data set. The optimal separating hyperplane is one of the core ideas behind the support vector machines. In particular, it gives rise to the so-called support vectors which are the data points lying on the margin boundary of the hyperplane. These points support the hyperplane in the sense that they contain all the required information to compute the hyperplane: removing other points does not change the optimal separating hyperplane. Elaborating on this fact, one can actually add points to the data set without influencing the hyperplane, as long as these points lie outside of the margin.

42. **With suitable equations, explain any two types of activation functions used in neural networks. (May 2019, Qn 6) (Mark 6)**

ReLu:- The Rectified Linear Unit (ReLU) function is one of the most popular Activation Functions in Deep Learning models. It is a fast-learning AF that promises to deliver state-of-the-art performance. Compared to other Activation Functions like the sigmoid and tanh functions, the ReLU function offers much better performance and generalization in deep learning. The function is a nearly linear function that retains the properties of linear models, which makes them easy to optimize with gradient-descent methods. The ReLU function performs a threshold operation on each input element where all values less than zero are set to zero. Thus, the ReLU is represented as:

$$f(x) = max(0, x) = \begin{cases} x_i, & if\ x_i \geq 0 \\ 0, & if\ x_i < 0 \end{cases}$$

Softmax: - The softmax function is a type of Activation Function used in neural networks to compute probability distribution from a vector of

real numbers. This function generates an output that ranges between values 0 and 1 and with the sum of the probabilities being equal to 1. The softmax function is represented as follows:

$$f(x_i) = \frac{exp\ (x_i)}{\sum_j exp\,(x_j)}$$

43. **Explain regression with an example. (May 2019 Q 11a) (Marks 5)**

Regression is a statistical method used in finance, investing, and other disciplines that attempts to determine the strength and character of the relationship between one dependent variable (usually denoted by Y) and a series of other variables (known as independent variables).
Regression helps investment and financial managers to value assets and understand the relationships between variables, such as commodity prices and stocks of business dealing in those commodities.
The two basic types of regression are simple linear regression and multiple linear regression, although there are non-linear regression methods for more complicated data and analysis. Simple linear regression uses one independent variable to explain or predict the outcome of the dependent variable Y, while multiple linear regression uses two or more independent variables to predict the outcome.Regression can help finance and investment professionals as well as professionals in other businesses. Regression can also help predict sales for a company based on weather, previous sales, GDP growth, or other types of conditions. The capital asset pricing model (CAPM) is an often-used regression model in finance for pricing assets and discovering costs of capital.

The general form of each type of regression is:
Simple linear regression: Y = a + bX + u
Multiple linear regression: Y = a + b1X1 + b2X2 + b3X3 + ... + btXt + u
Where:
Y = the variable that you are trying to predict (dependent variable).

X = the variable that you are using to predict Y (independent variable).
a = the intercept.
b = the slope.
u = the regression residual.

Regression takes a group of random variables, thought to be predicting Y, and tries to find a mathematical relationship between them. This relationship is typically in the form of a straight line (linear regression) that best approximates all the individual data points. In multiple regression, the separate variables are differentiated by using subscripts.

A Real World Example of How Regression Analysis Is Used

Regression is often used to determine how many specific factors such as the price of a commodity, interest rates, particular industries, or sectors influence the price movement of an asset. The aforementioned CAPM is based on regression, and it is utilized to project the expected returns for stocks and to generate costs of capital. A stock's returns are regressed against the returns of a broader index, such as the SP 500, to generate a beta for the particular stock.

Beta is the stock's risk in relation to the market or index and is reflected as the slope in the CAPM model. The return for the stock in question would be the dependent variable Y, while the independent variable X would be the market risk premium.

Additional variables such as the market capitalization of a stock, valuation ratios, and recent returns can be added to the CAPM model to get better estimates for returns. These additional factors are known as the Fama-French factors, named after the professors who developed the multiple linear regression model to better explain asset returns.

44. **Is regression a supervised learning technique? Justify your answer. Compare regression with classification with examples.**

(Dec 2019, Q11 a) **(Marks 5)**

Part 1:

Linear Regression is Supervised because the data you have include both the input and the output. So, for instance, if you have a dataset for, say, car sales at a dealership. You have, for each car, the make, model, price, color, discount etc. but you also have the number of sales for each car. If this task was unsupervised, you would have a dataset that included, maybe, just the make, model, price, color etc. (not the actual number of sales) and the best you could do is cluster the data. Linear regression requires a quality check of the predicted output by comparing it with the actual desired output.

Part 2:

Regression is the process of finding a model or function for distinguishing the data into continuous real values instead of using classes or discrete values. It can also identify the distribution movement depending on the historical data. Because a regression predictive model predicts a quantity, therefore, the skill of the model must be reported as an error in those predictions.

For Example: Predicting the amount of rainfall that we'll get on a particular day based on historical data.

Classification is the process of finding or discovering a model or function which helps in separating the data into multiple categorical classes i.e. discrete values. In classification, data is categorized under different labels according to some parameters given in input and then the labels are predicted for the data. The derived mapping function could be demonstrated in the form of "IF-THEN" rules. The classification process deals with the problems where the data can be divided into binary or multiple discrete labels.

For Example: Classifying whether a given image is a fruit or a vegetable.

45. **Differentiate between supervised and unsupervised training. Explain with suitable examples. (May 2019, Q11 b)(Mark 5)**

Supervised Training:

Supervised training is a machine learning technique in which we have prior knowledge of what the output values for our training data set should be.
This means that we have a full set of labelled data while training. Fully labeled means that each example in the training data set is tagged with the answer the algorithm should come up with on its own. So, a labeled data set of fruit images would tell the model which photos were of apples, bananas, oranges and so on. When the model encounters a new image, it compares it to the training examples to predict the correct label.

Therefore the goal of supervised training is to train a machine learning model which given a sample of data and desired outputs, best approximates the relationship between input and output observed in the data.

For Example: for a supervised classification model that is used to predict a fruits name given an image, first we train the model on a data set of fruit images with names, then we test it on images containing fruits and check the accuracy of the trained model.

Unsupervised Training:

Unsupervised training, on the other hand, does not have labeled outputs, so its goal is to infer the natural structure present within a set of data points.
The machine learning model is given a data set without explicit instructions on what to do with it. The training data set is a collection of examples without a specific desired output or correct answer. The model then attempts to automatically find structure in the data by extracting useful features and analyzing its structure. But since no labels are provided, there is no specific way to compare model performance in most unsupervised training methods.

For Example: for an unsupervised clustering model that separates the pictures of dogs according to their species according to the differences in fur colour, height, etc. We can use a trained unsupervised model to roughly group unlabeled images into similar or dissimilar groups.

46. **Explain basic problems associated with hidden markov model. (December 2018, Q 17 a)) (Mark 6)**

The Hidden Markov model is associated with three basic problems - evaluation, decoding, and learning to find the most likelihood classification. Evaluation problem can be used for isolated(word) recognition. Decoding problem is related to the continuous recognition as well as to the segmentation. Learning problem must be solved if we want to train an HMM for the subsequent use of recognition tasks.

The Evaluation problem:

By evaluation, we mean the probability that a particular sequence of symbols is produced by a particular model. Our first problem is to compute the likelihood of a particular observation sequence or in other words which model gives the highest likelihood to the data.

Computing Likelihood: Given an HMM λ= (A, B) and an observation sequence O, determine the likelihood P(O|λ).

For an HMM with N hidden states and an observation sequence of T observations, there are N^T possible hidden sequences. For real tasks, where N and T are both large, N^T is a very large number, and so one can not compute the total observation likelihood by computing a separate observation likelihood for each hidden state sequence and then summing them up. Instead of using such an extremely exponential algorithm, we use an efficient algorithm called the forward algorithm. The forward algorithm computes the observation probability by summing over the probabilities of all possible hidden-state paths that could generate the observation sequence.

The Decoding problem:

Decoding means determining the most likely sequence of states that produced the sequence. Thus given an observation sequence O and an HMM λ= (A, B), discover the best-hidden state sequence Q. For this problem, we use the Viterbi algorithm.

In formal words, given as input an HMM λ= (A,B) and a sequence of observations O = o1,o2, ...,oT, find the most probable sequence of states Q = q1q2q3 . . .qT .

We can run the forward algorithm and compute the likelihood of the observation sequence for the given hidden state sequence. Then choose

the hidden state sequence with the max observation likelihood. But this can not be done since there is an exponentially large number of state sequences. Instead, the most common decoding algorithm for HMMs is the Viterbi algorithm.

Viterbi algorithm is identical to the forward algorithm exceptions. It takes the max over the previous path probabilities whereas the forward algorithm takes the sum. It contains a backpointers component that the forward algorithm lacks. This is because while the forward algorithm needs to produce observation likelihood, the Viterbi algorithm must produce a probability and also the most likely state sequence. This best state sequence is computed by keeping track of the path of hidden states that led to each state.

The Learning problem:

Generally, the learning problem is the adjustment of the HMM parameters, so that the given set of observations or the training set is represented by the model in the best way for the intended application.

In other words, Given a model λ and a sequence of observations O = o1,o2, ...,oT, how should we adjust the model parameters {A, B, π} in order to maximize P(O|λ).

The quantity' which is to be optimized during the learning process can be different from application to application. There may be several optimization criteria for learning, from which a suitable one is selected depending on the application.

So in order to obtain the desired model that best fits the data, the following 3 algorithms are applied:

- MLE (maximum likelihood estimation)
- Viterbi training(different from Viterbi decoding)
- Baum Welch - forward-backward algorithm

47. **Explain the concept of association rule analysis with its application. (Dec 2019, Q 13 (b))) (Mark 3)**

Association Rule analysis, as the name suggests, association rules are simple. If/Then statements that help discover relationships between seemingly independent relational databases or other data repositories.

Most machine learning algorithms work with numeric datasets and hence tend to be mathematical. However, association rule analysis is suitable for non-numeric, categorical data and requires just a little bit more than simple counting.

Association rule analysis is a procedure which aims to observe frequently occurring patterns, correlations, or associations from datasets found in various kinds of databases such as relational databases, transactional databases, and other forms of repositories.

An association rule has 2 parts:

an antecedent (if) and a consequent (then) An antecedent is something that's found in data, and a consequent is an item that is found in combination with the antecedent. Have a look at this rule for instance:

"If a customer buys bread, he's 70% likely of buying milk."

In the above association rule, bread is the antecedent and milk is the consequent. Simply put, it can be understood as a retail store's association rule to target their customers better. If the above rule is a result of a thorough analysis of some data sets, it can be used to not only improve customer service but also improve the company's revenue.Association rules are created by thoroughly analyzing data and looking for frequent if/then patterns. Then, depending on the following two parameters, the important relationships are observed: Support: Support indicates how frequently the if/then relationship appears in the database. Confidence: Confidence tells about the number of times these relationships have been found to be true. So, in a given transaction with multiple items, Association Rule analysis primarily tries to find the rules that govern how or why such products/items are often bought together. For example, peanut butter and jelly are frequently purchased together because a lot of people like to make PBJ sandwiches.

Application:

Market Basket Analysis: This is the most typical example of association analysis. Data is collected using barcode scanners in most supermarkets. This database, known as the "market basket" database, consists of a large number of records on past transactions. A single record lists all the items bought by a customer in one sale. Knowing which groups are inclined towards which set of items gives these shops the freedom to adjust the store layout and the store catalog to place the optimally

concerning one another. This is the most typical example of association analysis. Data is collected using barcode scanners in most supermarkets. This database, known as the "market basket" database, consists of a large number of records on past transactions. A single record lists all the items bought by a customer in one sale. Knowing which groups are inclined towards which set of items gives these shops the freedom to adjust the store layout and the store catalog to place the optimally concerning one another.

Association Rule analysis has helped data scientists find out patterns they never knew existed.

48. **Explain the concept of Probably Approximately correct learning(May 2019) 4 marks**

In Probably Approximately Correct (PAC) learning, given a class, C, and examples drawn from some unknown but fixed probability distribution, p(x), we want to find the number of examples, N, such that with probability at least
1- δ, the hypothesis h has error at most ϵ, for arbitrary $\delta \leq 1/2$ and $\epsilon > 0$. In short, the goal of a PAC learner is to build a hypothesis with high probability (denoted 1- δ) that would be approximately correct (ie: error rate less than ϵ).
P$\{C\Delta h \leq \epsilon\} \geq$ 1- δ.
Where CΔh is the region of difference between C and h. We would like to make sure that the probability of a positive example falling in here (and causing an error) is at most ϵ, with a confidence 1- δ.
Assuming S as the tightest possible rectangle, the error region between C and h is the sum of four rectangular strips .
For any of these strips, if we can guarantee that the probability is upper bounded by $\epsilon/4$, the error is at most $[4(\epsilon/4)=\epsilon]$.
We count the overlaps in the corners twice, and the total actual error in this case is less than $4(\epsilon/4)$.
The probability that a randomly drawn example misses this strip is 1- $\epsilon/4$.
The probability that all N independent draws miss the strip is (1 - $\epsilon/4$)N , and the probability that all N independent draws miss any of the four strips is at most $4(1 - \epsilon/4)$N , which we would like to be at most δ. We have the inequality $(1 - x) \leq \exp[-x]$

So if we choose N and such that we have $4\exp[-\epsilon/4] \leq \delta$ we can also write 4 (1- $\epsilon/4$)N $\leq\delta$. Dividing both sides by 4, taking (natural) log and rearranging terms, we have N $\geq (4/\epsilon)\ \log(4/\delta)$ Therefore, provided that we take at least $(4/\epsilon)\log(4/\delta)$ independent examples from C and use the tightest rectangle as our hypothesis h, with confidence probability at least $1 - \delta$, a given point will be misclassified with error probability at most ϵ. We can have arbitrary large confidence by decreasing δ and arbitrary small error by decreasing . The number of examples is a slowly growing function of $1/\epsilon$ and $1/\delta$, linear and logarithmic, respectively.

49. **Compare k means clustering with hierarchical clustering techniques? (September 2020, Qn 10) 4 marks**

k-means Clustering

k-means, using a pre-specified number of clusters, the method assigns records to each cluster to find the mutually exclusive cluster of spherical shape based on distance. It needs advance knowledge of K i.e. no. of clusters one want to divide your data. We can use the median or mean as a cluster centre to represent each cluster. In K Means clustering, since one starts with a random choice of clusters, the results produced by running the algorithm many times may differ. It also found to work well when the structure of the clusters is hyperspherical (like a circle in 2D, the sphere in 3D). The main advantage of K means clustering is that Convergence is guaranteed. In k-means Clustering, K-Value is difficult to predict. Also, it won't work well with a global cluster.

Hierarchical Clustering

In hierarchical clustering we can stop at any number of clusters, we can find appropriate by interpreting the dendrogram. Hierarchical methods can be either divisive or agglomerative. Agglomerative methods begin with 'n' clusters and sequentially combine similar clusters until only one cluster is obtained. Divisive methods work in the opposite direction, beginning with one cluster that includes all the records and Hierarchical methods are especially useful when the target is to arrange the clusters into a natural hierarchy. Results are reproducible in Hierarchical clustering. Hierarchical clustering is a set of nested clusters that are arranged as a tree.Hierarchical clustering doesn't work as well as, k means when the shape of the clusters is hyperspherical. Main advantages of hierarchial clustering is the Ease of handling of any forms of similarity or distance. Disadvantage is that Hierarchical clustering

requires the computation and storage of an n×n distance matrix. For very large datasets, this can be expensive and slow 1.

50. **Calculate the dissimilarity between two data points x1(2,3,4) and x2(4,3,5) using**
(a) Euclidian Distance (b) Manhattan Distance(Drc 2019) 4 marks

Definitions

A number of Machine Learning Algorithms - Supervised or Unsupervised, use Distance Metrics to know the input data pattern in order to make any Data Based decision. A good distance metric helps in improving the performance of Classification, Clustering and Information Retrieval process significantly.Euclidian distance and manhattan distance are two distance metrics used in machine learning algorithms for classification.

Formulas

distance calculation between two points A(p1,p2) and B(q1,q2)

i)Euclidian distance

$$D_e = (\sum_{i=1}^{n}(p_i - q_i)^2)^{1/2}$$

ii)Manhattan distance

$$D_m = \sum_{i=1}^{n} \mid p_i - q_i \mid$$

where n is the no. of dimensions

Answer

By question n=3 and x1(2,3,4) x2(4,3,5)
(a)Euclidian distance

$$D_e = (\sum_{i=1}^{3}(p_i - q_i)^2)^{1/2}$$

$$= ((2-4)^2 + (3-3)^2 + (4-5)^2)^{1/2}$$
$$= (4+0+1)^{1/2}$$
$$= 5^{1/2}$$
$$= 2.23$$

(b)Manhattan Distance

$$D_m = \sum_{i=1}^{3} \mid p_i - q_i \mid = \mid (2-4) \mid + \mid (3-3) \mid + \mid (4-5) \mid$$
$$= 2+0+1$$
$$= 3$$

51. **Discuss any four Examples of machine learning models?(September 2020: Qno 11b)** **4 marks**

(a) Supervised Learning
Supervised learningis the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labelled training data consisting of a set of training examples In supervised learning, each example is a pair consisting of an input object (typically a vector) and the desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances.

(b) Decision tree
A decision tree is a flowchart like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.In decision analysis, a decision tree

and the closely related influence diagram are used as a visual and analytical decision support tool, where the expected values (or expected utility) of competing alternatives are calculated.

A decision tree consists of three types of nodes:

- Decision nodes – typically represented by squares
- Chance nodes – typically represented by circles
- End nodes – typically represented by triangles

(c) Random forests or random decision forests

classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees.Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

(d) Neural Networks

A neural network is a series of algorithms that endeavours to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature. Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria. The concept of neural networks, which has its roots in artificial intelligence, is swiftly gaining popularity in the development of trading systems.

52. . **Explain the steps involved in Expectation-Maximization algorithm?(May 2019)** **4 marks**

Algorithm:

- Given a set of incomplete data, consider a set of starting parameters.
- Expectation step (E – step): Using the observed available data of the dataset, estimate (guess) the values of the missing data.

- Maximization step (M – step): Complete data generated after the expectation (E) step is used in order to update the parameters.
- Repeat step 2 and step 3 until convergence.

Initially, a set of initial values of the parameters are considered. A set of incomplete observed data is given to the system with the assumption that the observed data comes from a specific model.
The next step is known as "Expectation" – step or E-step. In this step, we use the observed data in order to estimate or guess the values of the missing or incomplete data. It is basically used to update the variables.
The next step is known as "Maximization"-step or M-step. In this step, we use the complete data generated in the preceding "Expectation" – step in order to update the values of the parameters. It is basically used to update the hypothesis.
Now, in the fourth step, it is checked whether the values are converging or not, if yes, then stop otherwise repeat step-2 and step-3 i.e. "Expectation" – step and "Maximization" – step until the convergence occurs.

53. . **Explain DBSCAN algorithm for density based clustering. List out its advantages compared to K-means?(Dec 2018) 10 marks**

Clustering analysis is an unsupervised learning method that separates the data points into several specific bunches or groups, such that the data points in the same groups have similar properties and data points in different groups have different properties in some sense. Density-Based Clustering refers to unsupervised learning methods that identify distinctive groups/clusters in the data, based on the idea that a cluster in data space is a contiguous region of high point density, separated from other such clusters by contiguous regions of low point density
Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a base algorithm for density-based clustering. It can discover clusters of different shapes and sizes from a large amount of data, which is containing noise and outliers.
The DBSCAN algorithm uses two parameters:

- minPts: The minimum number of points (a threshold) clustered together for a region to be considered dense.

- eps (): A distance measure that will be used to locate the points in the neighborhood of any point.

These parameters can be understood if we explore two concepts called Density Reachability and Density Connectivity. Reachability in terms of density establishes a point to be reachable from another if it lies within a particular distance (eps) from it. Connectivity, on the other hand, involves a transitivity based chaining-approach to determine whether points are located in a particular cluster. For example, p and q points could be connected if p-¿r-¿s-¿t-¿q, where a-¿b means b is in the neighborhood of a.

Algorithmic steps for DBSCAN clustering

- The algorithm proceeds by arbitrarily picking up a point in the dataset (until all points have been visited).
- If there are at least 'minPoint' points within a radius of '' to the point then we consider all these points to be part of the same cluster
- The clusters are then expanded by recursively repeating the neighborhood calculation for each neighboring point

The Complexity analysis of DBSCAN clustering algorithm

- Best Case: If an indexing system is used to store the dataset such that neighborhood queries are executed in logarithmic time, we get O(nlogn) average runtime complexity
- Average Case: Same as best/worst case depending on data and implementation of the algorithm.

Advantages of DBSCAN over K-means clustering

K-Means clustering may cluster loosely related observations together. Every observation becomes a part of some cluster eventually, even if the observations are scattered far away in the vector space. Since clusters depend on the mean value of cluster elements, each data point plays a role in forming the clusters. A slight change in data points might affect the clustering outcome. This problem is greatly reduced in DBSCAN due to the way clusters are formed. This is usually not a big problem unless we come across some odd

shape data. Another challenge with k-means is that you need to specify the number of clusters ("k") in order to use it. Much of the time, we won't know what a reasonable k value is a priori. What's great about DBSCAN is that you don't have to specify the number of clusters to use it. All you need is a function to calculate the distance between values and some guidance for what amount of distance is considered "close". DBSCAN also produces more reasonable results than k-means across a variety of different distributions.