Machine Learning

Chapter 1 : Introduction to Machine Learning

Q. 1 Explain classic and adaptive machines.

Ans. :

The classical system receives some input values and produces output as a result of processing them.

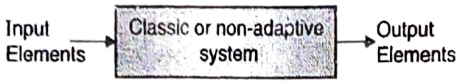


Fig. 1.1 : Environment of Classical System

Adaptive system has the ability to adapt its behavior to external signals like datasets or real time input to predict the future.

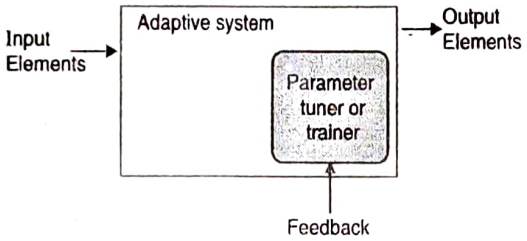


Fig. 1.2 : Adaptive System

Q. 2 Define Machine Learning and its different approaches.

Ans. :

As per the definition given by Giuseppe Bonaccorso, machine learning is an engineering approach that gives maximum importance to every technique that increases or improves the propensity for changing adaptively.

Therefore, the main goal of machine learning is to study, engineer, and improve mathematical models, which can be trained (once or continuously) with context-related data (provided by a generic environment), to infer the future and to make decisions without complete knowledge of all influencing elements (external factors).

Some of common approaches to machine learning are :

1. Supervised learning

It is a learning technique through previously given examples. In supervised learning both input and output variables are given which uses an algorithm to find some output(Y) from input(X) by deriving some mapping function like Y = f(X).

Supervised learning problems can be further divided into two parts :

a. **Classification:** In this case the output variable is a category or a group

For example "red" or "yellow" or "spam" and "not spam".

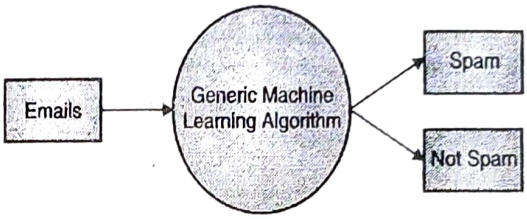


Fig. 1.3 : Example of Classification

b. **Regression:** In this type of problem, the output variable is a real value

For example "dollars" or "weight"

2. Unsupervised learning

It is a learning technique where the system discovers the patterns or structures directly from the example given.

In unsupervised learning, only input data (X) is given to find corresponding output variables.

Unsupervised learning problems can be further divided into parts :

a. **Association :** This type of problem discovers rules to describe large data.

For example, "If a person buys an item 'A' then also tends to buy item 'B'."

b. **Clustering :** This type of problem discovers groups of data based on similarities.

For example, "Customers' grouping based on their buying habit".

3. Reinforcement learning

It is a learning technique where dynamic environment is given and computer program interacts with it to perform a particular task. The program is also provided rewards and punishments as a feedback to navigate its problem space.

Here, the machine is trained to make particular decisions and it continuously trains using trial and error methods.

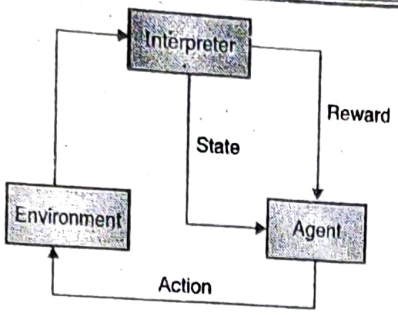


Fig. 1.4 : Example of Reinforcement Learning

Q. 3 State applications of deep learning

Ans. :

1. Deep learning is used in cancer detection to automatically detect cancer cells.
2. In driverless car to detect objects such as stop signs and traffic lights, but its development desires huge number of images and thousands of hours of videos.
3. In automated industry, it detects the unsafe distance between machines and human or objects.
4. It is used in speech translation like home appliances that respond to voice.

Q. 4 Differentiate between Deep Learning and Machine Learning.

Ans. :

Machine Learning	Deep Learning
A model is created by relevant features which are manually extracted from images to detect an object in the image.	Relevant features are automatically extracted from images. It is an end-to-end learning process.
When more examples or training data is given, then it is consistent at a certain level of performance.	As the size of data increases, it continues to improve.
Machine learning is less complex as compare to deep learning.	Deep learning is generally more complex.
Without a high-performance GPU and lots of labelled data, machine learning techniques can be used based on application.	To get reliable results in less time one should have a high-performance GPU and lots of labelled data.

Q. 5 Differentiate between Big Data and Machine Learning.

Ans. :

Sr. No.	Big Data	Machine Learning
1.	Big data analytics look for emerging patterns by extracting existing information which helps in the decision making process.	It teaches the machine by learning from existing data.
2.	Problem : Dealing with large volumes of data.	Problem Overfitting.
3.	It stores large volumes of data and finds out patterns from data.	It learns from trained data and predicts future results.
4.	It processes and transforms data to extract useful information.	Machine Learning uses data for predicting output.
5.	It deals with High-Performance Computing.	It is a part of Data Science.

Q. 6 Explain the important elements of Machine Learning.

Ans. : Important elements of machine learning

1. Data Formats

In supervised learning, a dataset is required, which is a finite set of real vector having n features.

Consider a data set X in which all samples are independent and identically distributed. This means all variables belong to the same distribution D, and considering an arbitrary subset of m values,

$$P\left(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m\right) = \prod_{i=1}^{m} P(\bar{x}_i)$$

If the corresponding output values are numerical continuous, then the process is called as Regression and if output is categorical then it is called as Classification.

Predictor : A predictor is a function f that maps an input x to an output y. In statistics, y is known as a response, and when x is a real vector it is known as the covariates.

Types of prediction tasks :

Binary classification (e.g., email ⇒ spam/not spam) :

$X \rightarrow f \rightarrow y \in \{+1, -1\}$

Regression (e.g., location, year ⇒ housing price) :

$X \rightarrow f \rightarrow y \in R$

Where f can be a regressor or classifier

Generic regressor : It is a vector-valued function, which gives continuous output.

Generic classifier : It is a vector-values function, which predicts output as categorical or discrete.

2. Parametric Learning

To simplify the learning process, assumptions or internal parameter vector can be considered, but this can limit the learning process. This approach is called parametric learning.

Parametric model : It summarizes data with a set of parameters of fixed size, so it doesn't matter how much data is being considered for parametric model.

Examples of parametric Machine Learning Algorithms :

(a) Logistic Regression

(b) Linear Discriminant Analysis

(c) Perceptron

(d) Naive Bayes

(e) Simple Neural Networks

Benefits of Parametric Machine Learning Algorithms :

(a) **Simpler :** Result analysis or interpretation is easy to understand.

(b) **Speed :** Learning using data is rapid.

(c) **Less Data :** Lesser amount of data is sufficient for training .

Limitations of Parametric Machine Learning Algorithms :

(a) **Constrained :** By choosing a functional form, these methods are highly constrained to the specified form.

(b) **Limited Complexity :** Only suitable for simple problems.

(c) **Poor Fit :** Not likely to match the primary mapping function.

3. Non-parametric Learning

These Types of Algorithms do not Make Strong Assumptions for Mapping Functions.

As there are no assumptions, any functional form can be learnt from the training data. This type of Non-Parametric learning is applicable when there is a lot of data and no prior knowledge is available.

A very common Non-Parametric family is called **Instance-Based Learning** and makes real-time predictions (without pre-computing parameter values) based on hypothesis determined only by the training samples (instance set).

Examples of Non-Parametric Machine Learning Algorithms :

(a) K-Nearest Neighbors

(b) Decision Trees like CART and C4.5

(c) Support Vector Machines

Benefits of Non-Parametric Machine Learning Algorithms :

(a) **Flexibility :** It is flexible to fit for a number of functional forms.

(b) **Power :** Assumptions about the primary function are not required.

(c) **Performance :** Models for prediction show high performance.

Limitations of Non-Parametric Machine Learning Algorithms :

(a) **More data :** Large training data are required to evaluate the mapping function.

(b) **Slower :** Training is slow as more parameters are required to train.

(c) **Overfitting :** Difficult to understand why specific predictions are made as there is a risk to overfit the training data.

4. Multiclass Strategies

In classification problem, when output classes is greater than one, there are two possibilities:

1. One-vs-all 2. One-vs-one

In both cases, the output will be the final value or class. e.g., classify a set of images of vegetables which may be onion, potato or tomato. Multiclass classification makes the assumption that each sample is either an onion or a potato, but not both at the same time.

1. One-vs-all

Most of algorithms of scikit-learn adopts this strategy.

n classifiers will be trained for n output classes to distinguish the actual class and remaining one.

Complexity is O(n) and at the most n-1 checks are required to get the correct class.

Interpretability is the advantage of this approach.

Since one and only one classifier represents each class, it is possible to gain knowledge about the class by inspecting its corresponding classifier.

2. One-vs-one

Train a model for each pair of classes.

The complexity is $O(n^2)$.

Majority voting is used to find the right class.

If two classes have-equal number of votes, it selects the class with the highest aggregate classification confidence by summing over the pair-wise classification confidence levels computed by the underlying binary classifiers.

More expensive and should be adopted only when a full dataset comparison is not preferable.

This method is usually slower than one-vs-all.

Q. 7 What is learnability.

Ans. :

A parametric model can be divided into two parts :

1. **Static structure of parameters :** It is determined by the choice of specific algorithm and mostly immutable unless model provides some re-modelling functionalities.

2. **Dynamic set of parameters :** It is the objective of optimization, which considers n unbounded parameters, which generates n-dimensional space.

The aim of a parametric learning process is :

(a) To find the best hypothesis

(b) Minimize the corresponding prediction error

(c) Avoid overfitting.

Chapter 2 : Feature Selection

Q. 1 Explain the concept of Feature selection in machine learning.

Ans. :

Feature selection is an important task that goes in hand in hand with feature engineering, where the job of a data scientist is to select the best possible subset of features and attribute that would help in building the right model. Feature engineering and selection is not a one time task, it needs to be carried out multiple times each time you build a model to get the best and optimal model for your problem. Data preprocessing and Feature Engineering is one of the toughest task in building a model for machine learning by data scientist.

Feature selection provides an effective way to solve a problem by removing redundant and irrelevant data. It is a process of isolating only those variables (or features) that are relevant to analysis. It keeps the best subset of predictors in the model. Feature selection is also called as variable selection or attributes selection. Selection of fewer attributes is advantageous as it reduces the complexity of the model and also reduces computational time.

Feature selection is different from dimensionality reduction (Feature extraction). In dimensionality reduction, the number of attributes are reduced by creating a new combination of attributes present in the data. (e.g. of dimensionality reduction is PCA (Principal Component Analysis).

Uses of Feature Selection

1. The complexity of a model is reduced; this makes it easier to interpret.
2. Enables the Machine Learning algorithm to train faster.
3. If a right subset is chosen, accuracy of a model is improved.
4. Over fitting is reduced.

Q. 2 With the help of an example explain the different types of data.

Ans. :

To understand data let us consider the following example of data. From the Fig. 2.1 we can see that data need not only be numeric representation, but it can also consists of names or symbols. Let us now bring the data into a table as shown in Fig. 2.2. Every data is given certain heading and the example data fits into heading (Name, age, Gender, score and experience). Some of the data may also require units as shown in the Fig. 2.2. In the Fig. 2.2, columns are variables. Rows are cases or observations (n).

Data can be classified into two types

1. Categorical Data

Responses that belong to groups or categories.

(a) From the data table, the first two columns are categorical data and the rest three are numerical data.

(b) E.g. Example of categorical data is Yes/No, Male/Female etc.

(c) Strongly agree to strongly disagree. Can be further classified as Nominal (no implied order), ordinal (order or rank).

2. Numerical Data

(a) A numerical value as a response.

(b) Discrete number or continuous.

(c) E.g. number of students in a class.

(d) Height of people in locality.

(e) Can be further classified as Interval (add/subtract), ratio (add, subtract, multiply and divide).

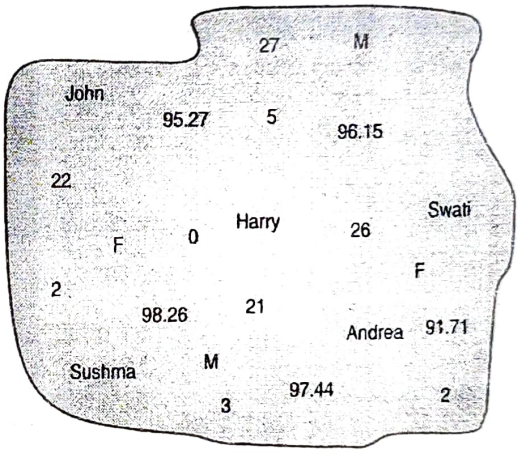


Fig. 2.1

Data table

Name	Gender	Age	Score	Experience
John	M	27	95.27	5
Sushma	F	25	96.15	2
Swati	F	26	91.71	3
Harry	M	21	97.44	0
Andrea	F	22	98.26	12
Nominal	nominal	ratio	ordinal	interval

Fig. 2.2 : Data Table

1. Qualitative Data

(a) No measurable meaning to the difference of numbers.

(b) E.g. Number in the shirt of a sports person (There is no significance to the meaning of numbers on their shirts e.g. a sports person having number 82 would not mean that he is a senior player when compared to a sports person wearing a shirt having number 45).

(c) Qualitative data can be further classified into Nominal and Ordinal.

2. Quantitative Data

(a) Meaning to the difference can be given.

(b) E.g. 80 marks and 60 marks (we can say one person has scored more than the other person).

Q. 3 Find out whether the following examples are cross sectional or time series data.

1. **Company has data on number of employees who are working on a Manufacturing project and the amount sanctioned to that project.**
2. **2000 people were asked whether a particular political party would win the election in the upcoming year.**
3. **The number of people who bought their grocery items from a supermarket for more than Rs.5000 for five days of the week.**
4. **100 customers give a movie review feedback, 50 ticked excellent, 30 ticked average and 20 ticked poor.**
5. **Number of cars parked in a parking lot for 7 days a week.**

Ans. :

1. Cross sectional : As it is taken at a certain point and not in different points for comparison.
2. Cross sectional.
3. Time series data : Data is measured according to some frequency.
4. Cross Sectional.
5. Time Series.

Q. 4 Explain the different methods of managing categorical data.

Ans. : Different methods of managing categorical data

I. Transforming Categorical Features

As we have seen that categorical variables can be of two types :

(a) Nominal

(b) Ordinal

Transforming Nominal Features

(a) Nominal attributes are categorical variables having a distinct discrete values.

(b) The format of nominal variables are in text or string, these values cannot be understood by machine learning algorithms.

(c) Due to which they need to be transformed into numeric format.

Transforming Ordinal Features

(a) Ordinal features are analogous to nominal variables except they have ordering among their values.

(b) They can be represented in text form, so a mapping can be used to represent them into numeric form.

II. Encoding Categorical Features

Need for Encoding categorical Features

(a) If the transformed numeric representations of categorical features have no ordinal relationships, transforming categorical features are not enough.

(b) If these transformed categorical features are fed into any algorithm, the model will treat them as raw numeric features and the notion of magnitude will be wrongly introduced in the system.

(c) If model were built using these features then it would result in sub optimal and incorrect model.

(d) To encode the categorical features techniques like one hot encoding, dummy encoding, effect encoding and feature hashing schemes can be used.

(e) For e.g. in our dataset we have a column for color and three different colors are present (Red, yellow and green).

(f) If we apply transformation then Red can have a numeric representation of 1, yellow = 2 and green = 3.

Color		Color
Red	⟶	1
Red		1
Yellow		2
Green		3
Yellow		2

Fig. 2.3 : Categorical Features having 3 labels

A. One Hot Encoding scheme

(i) Consider a numeric representation of any categorical features with m labels e.g. in the Fig. 2.3 (m = 3 labels).

(ii) In this technique it encodes or transforms the features into m (m=3) binary features containing a value of 1 or 0.

(iii) Each observation in the categorical feature is thus converted into a vector size of m (m=3) with only one of the value as 1, which indicates its active.

(iv) The binary variables are often called as dummy variables.

e.g.

Color	Red	Yellow	Green
Red	1	0	0
Red	1	0	0
Yellow ⟹	0	1	0
Green	0	0	1
Yellow	0	1	0

B. Dummy Coding scheme

(i) It's a technique similar to one hot encoding scheme, except we have m-1 binary features as compared to m features in one hot encoding scheme.

(ii) The categorical variable is converted into a vector of size m-1.

(iii) The extra feature is omitted, and if the category values ranges from {0, 1...., m-1} the 0^{th} or the $m\text{-}1^{th}$ feature is represented as a vector of all zeros(0).

e.g. (case i) 0^{th} feature is represented by a vector of all zeros(0)

Color	Yellow	Green
Red	0	0
Red	0	0
Yellow ⟹	1	0
Green	0	1
Yellow	1	0

(Case ii) $m\text{-}1^{th}$feature is removed and it is represented by a vector or all zeros (0)

Color	Red	Yellow
Red	1	0
Red	1	0
Yellow ⟹	0	1
Green	0	0
Yellow	0	1

C. Effect coding scheme

(i) In most aspects, Effect coding scheme is similar to dummy coding scheme.

(ii) In Effect coding scheme, those encoded features are replaced by -1 that are represented as all 0s in the dummy coding scheme.

e.g.

Color	Yellow	Green
Red	-1	-1
Red	-1	-1
Yellow ⟹	1	0
Green	0	1
Yellow	1	0

D. Bin counting scheme

(i) One hot encoding, dummy coding and effect coding are techniques suitable for small number of distinct categories.

(ii) If the number of distinct categories becomes large, one hot encoding, dummy coding and effect coding techniques start causing a problem.

(iii) If the categorical feature has m distinct labels then m separate features are generated, this can lead to increase in the size of the feature set.

(iv) This can result in model training problems with regard to time, space and memory and also storage issues.

(v) Another major drawback that arises is the curse of dimensionality in which there are enormous number of features and not enough representative samples, model performance starts getting affected.

(vi) To deal with a large number of possible categories, Bin Counting scheme is useful.

(vii) In this technique instead of using the actual label values for encoding, probability based statistical information about the value and the actual target or response value, which is aimed for prediction by the model is used.

(viii) E.g. Historical data for IP address and the ones that were used for DDOS attacks, a probability can be built for DDOS attack been caused by any one of the IP addresses.

(ix) Using this information, input feature can be encoded which depicts whether the same IP address comes in the future, what is the probability of DDOS attack being caused.

E. Feature Hashing scheme

(i) This technique is another useful technique for handling large-scale categorical features.

(ii) Feature Hashing scheme uses a Hash function, a vector of predefined length is used which represent the number of encoded features pre-set.

(iii) The indices in this predefined vector are the hashed values of the features and they are updated accordingly

(iv) The technique can result in problem of collision as the hash functions maps a large number of values into a finite set of values.

(v) To overcome the problem of collisions, a signed hash function can be used. The sign of the value obtained after applying the hash function is used as the sign of the value, this sign is stored in the final feature vector at the appropriate index.

(vi) Hashing scheme is suitable on different data types like strings, numbers and structures like vectors.

(vii) A hashed output represented as a set of h bins, whenever hash is calculated on the same values, they are assigned the same bin out of h bins based on the hash value obtained.

(viii) The final size of the encoded feature vector for each categorical feature encoded using hashing scheme is h.

(ix) E.g. if there 100 distinct categories in a feature and h is set to 20, the output will have 20 features as compared to 100 if one Hot encoding scheme is used.

Q. 5 Explain data scaling and normalization

Ans. :

Data Scaling

When dealing with real time data having numeric features, some of the attributes may be unbounded in nature for e.g. number of times a video is viewed on YouTube or the number of hits on a web page. Using such raw values as input features may make the model bjased towards features having really high magnitude values. Models like Linear or Logistic Regression are highly sensitive to the magnitude or scale of features.

Tree based models can work without feature scaling. So it is better to normalize and scale down the features using feature-scaling methods, if different machine learning algorithms are to be explored. E.g. for example currency like 1 USD = 100 Yen, but if we don't scale the two currencies, then a difference of 1 USD will be considered as same as the difference in 1 Yen. By scaling, different variables can be compared on a common basis.

e.g. of Scaling

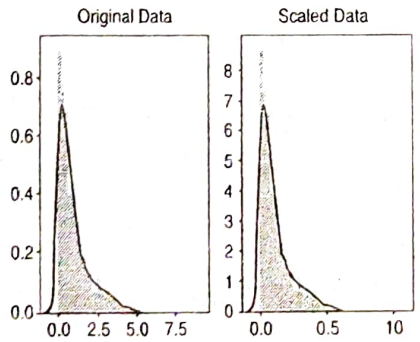


Fig. 2.4

A. Standardized Scaling

(i) In standardized scaling, the values in the feature column are standardized by removing mean and scaling the variance to be 1 from the values.

(ii) This technique is also known as centering and scaling.

(iii) It is popularly known as Z-score scaling.

(iv) It can be mathematically denoted as,

$$SS(X_i) = \frac{X_i - \mu_x}{\sigma_x}$$

(v) Where, X_i is the feature, μ_x is the mean, σ_x is the standard deviation

B. Min-Max Scaling

(i) In Min-Max scaling, the features are transformed and scaled such that the each value is in the range of [0,1].

(ii) The Min-Max scaler in scikit learn allows one to specify the upper and lower bound using the feature_range variable

$$MMS(X_i) = \frac{X_i - \min(X)}{\max(X) - \min(X)}$$

X is the feature

(iii) min(X) is the minimum value in the feature X

(iv) max(X) is the maximum value in the feature X

C. Robust Scaling

(i) The drawback of Min Max scaling is that the presence of outliers affect the scaled values.

(ii) Robust scaling uses statistical measure to scale the features without being affected by outliers.

$$RS(X_i) = \frac{X_i - \text{median}(X)}{IQR_{(1,3)}(X)}$$

Where,

X is the feature

Median(X) is the median value of X

$IQR_{(1,3)}(X)$ is the Inter quartile range difference between first quartile (25^{th}) and the third quartile (75^{th})

Data Normalization

The term scaling and normalization are sometimes used interchangeably, however there is a difference between the two terms. In both the cases we are transforming the data.

The difference is that in scaling the range of data is changed, while in normalization the shape of the distribution of data is changed. **e.g.**

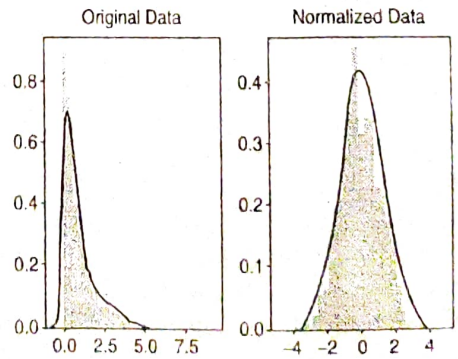


Fig. 2.5

Scikit learn provides a class for per-sample normalization, Normalizer. It has max, L1 and L2 normalization.

In Euclidean space the techniques may be given as,

$$\text{Max norm}: \|X\|_{max} = \frac{X}{|max_i\{X\}|}$$

$$\text{L1 norm}: \|X\|_{L1} = \frac{X}{\sum_i |x_i|}$$

$$\text{L2 norm}: \|X\|_{L2} = \frac{X}{\sqrt{\sum_i |x_i|^2}}$$

X is the feature value whose norm is to be calculated. x_i are the values in the features.

E.g.

X = [1.0, 2.0]

Max Norm = [0.5, 1.0]

L1 norm = [0.3333, 0.6666]

L2 norm = [0.4472, 0.8944]

Q. 6 Explain feature selection and filtering.

Ans. : Feature selection and filtering

1. Dealing with too many features results in "Curse of Dimensionality".
2. More features make the model more complex and difficult to interpret.
3. It can also lead to model overfitting on the training data.
4. This in turn will result in a model that may give high performance only on the data, which is used for training but will end up in poor performance on previously unseen data.
5. The objective is to select optimal number of features, these optimal features can be used for training and building models that generalize on data and prevent overfitting.
6. Feature selection techniques can be divided into three main areas based on strategy and techniques employed.

Q. 7 Explain Principal component analysis. Support your answer with a suitable example

Ans. : Principle Component Analysis (PCA)

Principal Component Analysis also known as PCA, is a statistical technique. It uses a process of Linear, orthogonal transformation to transform higher dimensional features (may be highly correlated) to a lower dimensional set of linearly uncorrelated features. These uncorrelated features generated by PCA are called as Principal Components or PCs. The total number of Principal components are always less than or equal to the initial number of features. The maximum variance of the original features is captured by the first component of PCA. The remaining components capture more variance. Such that these PCs are orthogonal to the preceding components. This technique is sensitive to feature scaling.

Pre-requisites for PCA (Mathematics Background)

This section deals with some mathematics background that is needed to understand the concept of PCA.

Mean

1. It is the average of the numbers.
2. In order to calculate mean, add up all the numbers and divide by how many numbers are there.
3. However mean does not tell us lot about the data, except its middle point.
4. e.g. {2, 5, 8}, mean = (2 + 5 + 8) / 3 = 5

Formula

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n}$$

Standard Deviation

1. It is a measure of how spread out the data is.
2. It is the average distance from the mean of the data set to a point.

Formula

$$s = \sqrt{\frac{\sum_{i=1}^{n} (X_i - \bar{X})^2}{(n-1)}}$$

$\bar{X}$: mean ;

n : Total number of elements

e.g.

X	$(X - \bar{X})$	$(X - \bar{X})^2$
0	– 10	100
8	– 2	4
12	2	4
20	10	100
Total		208
Divided by (n – 1)		69.333
Square Root		8.3266

Variance

It is another measure that tells about the spread of the data.

Formula

$$S^2 = \frac{\sum_{i=1}^{n} (X_i - \bar{X})^2}{(n-1)}$$

Covariance

1. The Standard deviation and Variance are 1D measures. However many datasets have more than 1 Dimensions.

2. The aim of the statistical analysis is to see if there are any relationships between the dimensions.
3. E.g. we could have dataset like the number of hours spent for studies and the marks obtained, statistical analysis could be performed to find out if the number of hours have any impact on the marks obtained.
4. Covariance is a measure that tells how much the dimensions vary from the mean with respect to each other.
5. It is always measured between two dimensions.
6. If one calculates covariance between one dimension and itself, variance is obtained.
7. cov(x, y) is equal to cov(y, x)

Formula

$$\text{cov}(X, Y) = \frac{\sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)}$$

Covariance Matrix

If the dataset has more than 2 dimensions, more than one covariance measure can be calculated.

E.g. for a three dimensional data set (dimensions x, y, z), cov(x, y), cov(y, z) and cov (x, z) may be calculated.

$$C = \begin{pmatrix} \text{cov}(x,x) & \text{cov}(x,y) & \text{cov}(x,z) \\ \text{cov}(y,x) & \text{cov}(y,y) & \text{cov}(y,z) \\ \text{cov}(z,x) & \text{cov}(z,y) & \text{cov}(z,z) \end{pmatrix}$$

e.g.

	Hours (H)	Mark (M)
Data	9	39
	15	56
	25	93
	14	61
	10	50
	18	75
	0	32
	16	85
	5	42
	19	70
	16	66
	20	80
Totals	167	749
Averages	13.92	62.42

Table 2.1 : 2-Dimensional data set and covariance calculation

H	M	$(H_i - \bar{H})$	$(M_i - \bar{M})$	$(H_i - \bar{H})(M_i - \bar{M})$
9	39	- 4.92	- 23.42	115.23
15	56	1.08	- 6.42	- 6.93
25	93	11.08	30.58	338.83
14	61	0.08	- 1.42	- 0.11
10	50	- 3.92	- 12.42	48.69
18	75	4.08	12.58	51.33
0	32	- 13.92	- 30.42	423.45
16	85	2.08	22.58	46.97
5	42	- 8.92	- 20.42	182.15
19	70	5.08	7.58	38.51
16	66	2.08	3.58	7.45
20	80	6.08	17.58	106.89
Total				**1149.89**
Average				**104.54**

Q. 8 Define and explain Eigen values and Eigen vectors.

Ans. :

Definition of Eigenvalue and Eigenvector

If A is an n × n matrix, then a nonzero vector x in R^n is called an eigenvector of A (or of the matrix operator T_A) if Ax is a scalar multiple of x ; that is,

$$Ax = \lambda x$$

For some scalar λ the scalar λ is called an eigenvalue of A (or of T_A), and x is said to be an eigenvector corresponding to λ.

When x is an Eigen vector of A, multiplication by A leaves the direction of the vector unchanged.

e.g. in R^2 or R^3multiplication by A maps each Eigen vector x of A (if any) along the same line through the origin as x.

Based on the sign and magnitude of eigen value of λ, eigen vector x, the operation Ax = λx either compresses or stretches x by a factor of λ, in the opposite direction where λ is negative as shown in the Fig. 2.6.

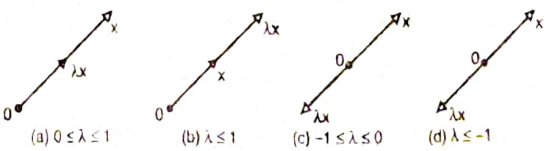


Fig. 2.6

e.g.

The vector $x = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ is an eigenvector of

$$A = \begin{bmatrix} 3 & 0 \\ 8 & -1 \end{bmatrix}$$

Corresponding to the eigenvalue $\lambda = 3$, since

$$Ax = \begin{bmatrix} 3 & 0 \\ 8 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \end{bmatrix} = 3x$$

Geometrically, multiplication by A has stretched the vector x by a factor of 3 (Fig. 2.7)

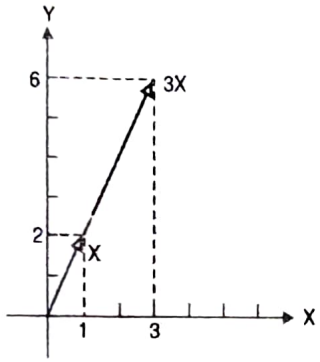


Fig. 2.7

Computing Eigen values and Eigen Vectors

A 2 × 2 matrix **A** has the following form :

$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ Where a, b, c and d are the elements of the matrix.

Calculating the trace and determinant

1. **Trace** : The trace of a matrix is defined as the sum of elements on the main diagonal (from upper left to lower right). It is also equal to the sum of the eigenvalues. In the case of our 2 × 2 matrix,

$$T = a + d$$

2. **Determinant** : The determinant of a matrix is useful in multiple further operations - for example while finding the inverse of a matrix. For a 2 × 2 matrix,

$$D = ad - bc$$

Determining the eigenvalues and Eigen vectors

Each 2 × 2 matrix A has two eigenvalues : λ_1 and λ_2. These are defined as real numbers that fulfill the following condition for a nonzero column vector $v = (v_1, v_2)$, called an eigenvector :

$$A * V = \lambda * V$$

You can also find another, equivalent version of the equation above :

$$(A - \lambda I)\, v = 0$$

where I is a 2 × 2 identity matrix.

Knowing the trace and determinant of the matrix A, you have to do is, input these values into the following equations :

$$\lambda_1 = \frac{T}{2} + \sqrt{\left(\frac{T^2}{4} - D\right)}$$

$$\lambda_2 = \frac{T}{2} - \sqrt{\left(\frac{T^2}{4} - D\right)}$$

Example : Finding Eigen Values

$$A = \begin{bmatrix} 3 & 0 \\ 8 & -1 \end{bmatrix}$$

Applying the above procedure we get,

Trace

$$T = a + d$$

$$T = 3 - 1$$

$$T = 2$$

Determinant D

$$D = ad - bc$$

$$D = (3 * -1) - (0 * 8)$$

$$D = -3$$

Finding Eigen Values

$$\lambda_1 = \frac{T}{2} + \sqrt{\left(\frac{T^2}{4} - D\right)}$$

$$\lambda_1 = \left(\frac{2}{2}\right) + \sqrt{\frac{2^2}{4} - (-3)}$$

$$\lambda_1 = 3$$

$$\lambda_2 = \frac{T}{2} - \sqrt{\left(\frac{T^2}{4} - D\right)}$$

$$\lambda_2 = \left(\frac{2}{2}\right) - \sqrt{\frac{2^2}{4} - (-3)}$$

$$\lambda_2 = -1$$

Now that we have found Eigen values we can find the Eigen vectors by the following equation

$$(A - \lambda I)\, v = 0$$

Let us use $\lambda_1 = 3$, and find the Eigen vector

Let $v = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

$$\begin{bmatrix} 3 & 0 \\ 8 & -1 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

$$\begin{bmatrix} 3-\lambda & 0 \\ 8 & -1-\lambda \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

$$\begin{bmatrix} 0 & 0 \\ 8 & -4 \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

$$8x_1 - 4x_2 = 0$$

$$2x_1 - x_2 = 0$$

$$2x_1 = x_2$$

$$x_1 = \frac{1}{2} x_2$$

Let $x_2 = t$

$$v = \begin{bmatrix} \frac{1}{2}t \\ t \end{bmatrix}$$

$$v = t\begin{bmatrix} \frac{1}{2} \\ 1 \end{bmatrix}$$

$$v = \begin{bmatrix} \frac{1}{2} \\ 1 \end{bmatrix}$$

Similarly, $\lambda_2 = -1$

$$v = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Q. 9 Enlist and explain the steps for computing PCA.

Ans. :

Step 1 : Given to you a dataset for e.g.

Data =

x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

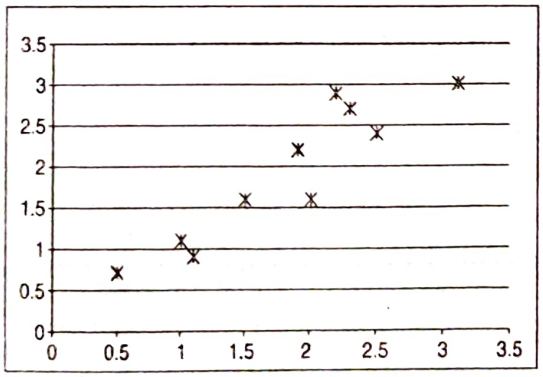


Fig. 2.8 : Data Plot

Step 2 : Calculate the Covariance Matrix.

The Covariance Matrix for a data set having 2 dimensions can be represented as

$$C = \begin{bmatrix} \text{cov}(x, x) & \text{cov}(x, y) \\ \text{cov}(y, x) & \text{cov}(y, y) \end{bmatrix}$$

x	x-xmean	xmean*xmean
2.5	0.69	0.4761
0.5	– 1.31	1.7161
2.2	0.39	0.1521
1.9	0.09	0.0081
3.1	1.29	1.6641
2.3	0.49	0.2401
2	0.19	0.0361
1	– 0.81	0.6561
1.5	– ..31	0.0961
1.1	– 0.71	0.5041
1.81	Sum	5.549
	Cov (x,x)	0.61655

x	y	x-xmean	y-ymean	xmean*ymean
2.5	2.4	0.69	0.49	0.3381
0.5	0.7	– 1.31	– 1.21	1.5851
2.2	2.9	0.39	0.99	0.3861
1.9	2.2	0.09	0.29	0.0261
3.1	3	1.29	1.09	1.4061
2.3	2.7	0.49	0.79	0.3871
2	1.6	0.19	– 0.31	– 0.0589
1	1.1	– 0.81	– 0.81	0.6561
1.5	1.6	– 0.31	– 0.31	0.0961
1.1	0.9	– 0.71	– 1.01	0.7171
Mean(1.81)	1.91			5.539
			cov(x,y) and cov(y,x)	0.61544

y	y-ymean	ymean*ymean
2.4	0.49	0.2401
0.7	– 1.21	1.4641
2.9	0.99	0.9801
2.2	0.29	0.0841
3	1.09	1.1881
2.7	0.79	0.6241
1.6	– 0.31	0.0961
1.1	– 0.81	0.6561
1.6	– 0.31	0.0961
0.9	– 1.01	1.0201
1.91	average	6.449
	cov (y, y)	0.71655

From the above tables the covariance matrix is calculated,

Therefore

$$C = \begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix}$$

Step 3 : Calculate the Eigen Vectors and Eigen Values from the covariance matrix

$$C = \begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix}$$

$$\det(C - \lambda I) = 0$$

$$\left| \begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \right| = 0$$

Determinant(D) = ad – bc

Trace(T) = (sum of diagonal elements)

Det = 0.6154*0.7165 – 0.6154*0.6154)

D = 0.4409341-0.3787

D = 0.0622

Trace = (0.6154+0.7165) =1.3319

Knowing the trace and determinant of the matrix C, finding the eigenvalues is to input these values into the following equations :

$$\lambda_1 = \frac{T}{2} + \sqrt{\frac{T^2}{4} - D}$$

$$\lambda_2 = \frac{T}{2} - \sqrt{\frac{T^2}{4} - D}$$

$\lambda_1 = 1.28$

$\lambda_2 = 0.0485$

To find the i^{th} Eigen vector

$$Ce_i = \lambda_1 e_i$$

$$\begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix} \begin{bmatrix} e_{11} \\ e_{12} \end{bmatrix} = 1.28 \begin{bmatrix} e_{11} \\ e_{12} \end{bmatrix}$$

$0.6165\, e_{11} + 0.6154\, e_{12} = 1.28\, e_{11}$ ...(1)

$0.6154 e_{11} + 0.7165\, e_{12} = 1.28\, e_{12}$ ...(2)

From (1) $1.28\, e_{11} - 0.6165\, e_{11} = 0.6154\, e_{12}$

$0.6635\, e_{11} = 0.6154\, e_{12}$

$e_{11} = \frac{0.6154}{0.6635} e_{12}$

$e_{11} = 0.9275\, e_{12}$

$$e_1 = \begin{bmatrix} 0.9275 \\ 1 \end{bmatrix}$$

Find the Euclidean length of above e_1 vector and divide the same to obtain the first Eigen vector (e_1)

$$\text{Euclidean length } e_1 = \sqrt{x_1^2 + x_2^2 \ldots + x_n^2}$$

Euclidean length = 1.363

Dividing e_1 elements with Euclidean length we get

Therefore e_1 = [0.6803 0.733]

Similarly applying above procedure we can obtain ,

e_2 = [– 0.734 0.677]

$$\text{Eigen vectors} = \begin{bmatrix} 0.6803 & 0.733 \\ -0.734 & 0.677 \end{bmatrix}$$

Step 4 : Choosing components and forming feature vector.

The Eigen vector with the highest Eigen value is the principle component of dataset.

Eigen vectors = [0.6803 0.733]

Reduce dimensionality and form feature vector. The eigenvector with the highest eigenvalue is the principal component of the data set.

Q. 10 Write short notes on Non negative Matrix factorization

Ans. :

Non-negative Matrix Factorization

1. NMF is a collection of algorithms in the field of linear algebra and multivariate data analysis.
2. Consider a matrix V. This matrix V can be factored into two matrices W and H such that

   $$WH = V$$

3. The dimensions of the matrix are V is m × n, W is m × k and H is k × n.
4. The matrix W and H are known as Factor matrices. These factor matrices are non negative, i.e. all the elements of factor matrices are equal or greater than zero.
5. One of the early work was done by Paatero and Tapper, 1994 (called positive matrix factorization).
6. The name Non Negative Matrix Factorization (NMF) was coined by Lee and Seung in 1999.
7. There is very little theoretical work on NMF : Donoho 2004 studied when factorization is unique, etc. Otherwise, only fixed point convergence has been shown for some algorithms.

NMF and low rank modelling

1. The statistical analysis of multivariate data using NMF is as follows :
2. Consider a multivariate n-dimensional data vectors, place these vectors in the columns of a n x m matrix V. m represents the number of examples in the data.
3. The matrix V can be factorized in to two matrices W and H. W having dimensions n x k and H having dimensions k x m
4. The size of W and H matrix have to smaller than the original matrix V, so k is usually chosen to be smaller than n or m.
5. The result of NMF is a compressed version of the original data matrix.
6. Low rank approximation $V_k = W_k H_k$ where all matrices are non-negative.

7. The factor matrices may have low rank than the matrix V or it can be approximated by V_k (rank k approximation of V).
8. A linear combination of column vectors in W and the coefficients given by elements of H are used to compute each column of V,

$$V_i = \sum_{j=1}^{N} H_{ji} W_j$$

9. The column vectors of W can be considered as basis vectors of vector space defined by V.
10. Columns of V are build from k columns of W

$$W_k H_{\cdot 1} = \begin{bmatrix} \vdots \\ w_1 \\ \vdots \end{bmatrix} h_{11} + \begin{bmatrix} \vdots \\ w_2 \\ \vdots \end{bmatrix} h_{21} + \ldots + \begin{bmatrix} \vdots \\ w_k \\ \vdots \end{bmatrix} h_{k1}$$

11. The representation is additive since W_k and H_k are non-negative.
12. A few basis vectors represent many data vectors, good approximation can be achieved only if these basis vectors discover structure latent in the data.

NMF factor interpretation

1. **Non-negative factors give benefits in interpretation**
   (a) Text processing requiring factorization of term by document matrix V_{mxn}, k can be considered the number of topics present in the document collection.
   (b) Element i, j of V tells how many times i^{th} word in vocabulary appears in document j.
   (c) W is a term by topic matrix whose columns are the basis vectors.
   (d) The nonzero elements of column 1 of W correspond to particular terms. Considering the highest weighted terms in this vector, one can assign label or topic to the basis vector.
2. **NMF interpretation**
   (a) Consider the field of image processing, the blocks in W and H matrix are sparse, where matrix H represents weight.
   (b) For example, in image database of face, each face image (obtained by stacking image into column vector) is a column in V.
   (c) W contains basis images and H their weights.
   (d) Basis images correspond to parts of a face such as eyes, nose, lips, hence **additive representation**.

NMF Applications

NMF can be used in a number of fields like

(i) Image Processing
(ii) Text processing and mining
(iii) Music transcription
(iv) Video analysis
(v) Bioinformatics
(vi) Chemistry

Q. 11 Write short notes on Sparse PCA

Ans. : Sparse PCA

1. Sparse PCA is a variant of Classical Principal Component Analysis technique.
2. The classical PCA dimensionality reduction tool makes use of near orthogonal vectors to find linear combination of a small number of features that maximizes the variance across data.
3. Need for Sparse PCA :
   (a) The features obtained using PCA are not sparse, thus they are difficult to interpret.
   (b) Sparse PCA improves the interpretability and relevance of the components; it also shows the structure of data.
   As the features in real time applications have a concrete physical meaning (e.g. sensors, people, genes).
   (c) Sparse components can be computed faster.
   (d) Sparse components provides better statistical regularization.

Sparse Principal Component Analysis using the Lasso (Elastic Net)

1. In this technique Standard PCA is formulated as a regression-type optimization problem.
2. The sparse loadings can be obtained by imposing the lasso (elastic net) constraint on the regression coefficients.

General SPCA Algorithm

1. Let α start at V[, 1:k] , the loading of first k ordinary principal components.
2. Given fixed α solve the following naïve elastic net problem for j = 1 ,2,...k

$$\beta_j = \arg\min_{\beta^*} \beta^{*T}(X^T X + \lambda)\beta^* - 2\alpha_j^T X^T X\beta^* + \lambda_{1,j}|\beta^*|_1$$

3. For each β do the SVD of $X^T X \beta = UDV^T$, then update $\alpha = UV^T$
4. Repeat steps 2-3 until β converges.
5. Normalization of

$$V_i = \frac{\beta_i}{|\beta_i|}, \; j = 1, 2, \ldots, k$$

Chapter 3 : Linear Regression

Q. 1 Explain the different regularization techniques used with Linear regression to handle multivariate data.

Ans. :

1. Regression analysis is one of the most widely used statistical techniques.
2. It estimates relationships among variables, a dependent (target, response) and independent variable (s)(predictor, explanatory)
3. It is a form of predictive modeling technique and also used for analyzing data.
4. Some of the regression analysis techniques include forecasting, time series modelling.
5. A curve or a line is fit to the available data, such that the differences between the distances of data points from the curve or line is minimum.

Uses of Regression Analysis

1. Modeling relationship between variables
2. Prediction of the target variable (forecasting)
3. Testing of hypothesis

Regression analysis can be categorized based on three metrics

1. Number and nature of Independent variable(s)
2. Number and nature of Dependent variable(s)
3. Shape of regression line

Q. 2 Explain different steps in conducting regression analysis.

Ans. : Steps in Conducting a Regression Analysis

Analyzing the correlation (strength and directionality of data), Fitting the regression or least squares line or best fit line. Evaluating the model's validity and its usefulness.

I) Analyzing the correlation (strength and directionality of data)

Definition of Correlation : Correlation, also known as correlation analysis, is a term used to denote the association or relationship between two (or more) quantitative variables. This analysis is based on the idea of a best fit line or a straight line, a linear relationship between two quantitative variables. The "strength" or the "extent" of an association between the variables and its direction is measured.

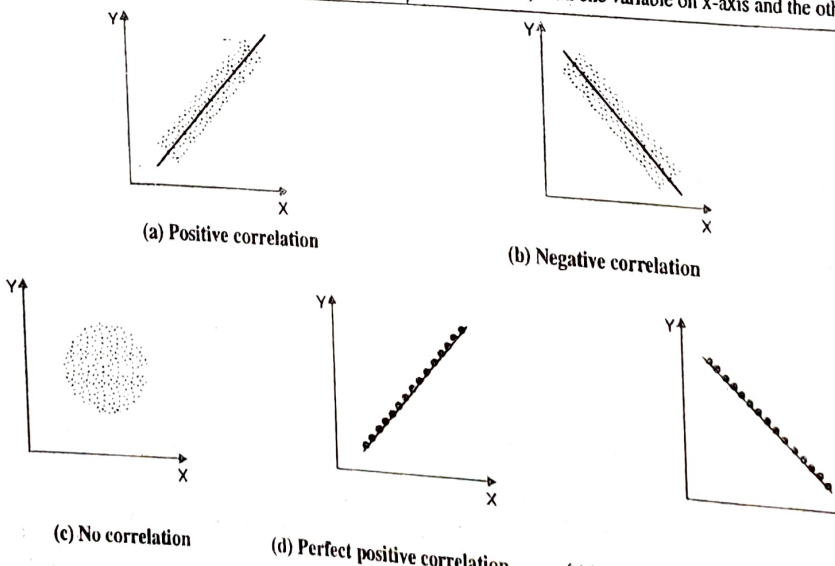
The correlation analysis results in a correlation coefficient whose values range from – 1 to + 1. A value of + 1 indicates that the two variables are related in a positive (linear) manner.

1. As X is increasing, Y is increasing.
2. As X is decreasing, Y is decreasing.

A value of – 1 indicates that the two variables are related in a negative (linear) manner.

1. As X is increasing, Y is decreasing.
2. As X is decreasing, Y is increasing.

Whereas a value of 0 (zero) indicates that there is no relationship between the two variables. A correlation analysis is shown in a scatter plot or scatter diagram (a graphical representation) with one variable on x-axis and the other on y-axis.



(a) Positive correlation (b) Negative correlation

(c) No correlation (d) Perfect positive correlation (e) Perfect negative correlation

Fig. 3.1

Fig. 3.1 Scatter plot between two variables (a) shows positive correlation (b) shows negative correlation (c) shows no correlation (d) Shows perfect positive correlation (e) shows perfect negative correlation

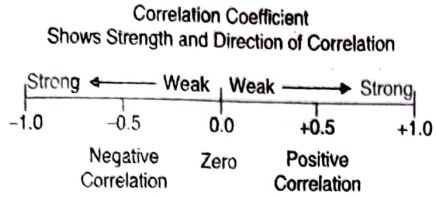


Fig. 3.2 : Spectrum of the correlation coefficient

II) Fitting the regression or least squares line

In Mathematics the equation of Line is given by

$$y = mx + c$$

Where m = slope or gradient of the line and c=intercept or where the line cuts the y axis.

In regression this equation is given by

$$y = b_0 + b_1 x$$

Where B_0 represents the intercept is that value of Y or the dependent variable, when the value of predictor variable is zero and B_1 represents the slope

III) Evaluating the validity and usefulness of the model

Validation techniques can be broadly classified into two types

(i) Numerical and (ii) Graphical

(i) In Numerical technique the value of R^2 (coefficient of Determination) is to be analyzed

R^2 (coefficient of Determination) predicts the extent of variability in the dependent variable. This variability can be explained by the independent variable

(ii) A graphical technique known as Graphical analysis of residuals can be used for validation, which uses graphs to visually inspect the data's robustness

Q. 3 Explain different methods for calculating correlation coefficient.

Ans. :

Calculating Correlation coefficients - Karl Pearson's Correlation Coefficient r and Spearman's Correlation Coefficient rho (ρ)

1. A correlation coefficient is a value that determines a relationship between the two variables.
2. Two methods can be used to calculate the correlation coefficient viz. The Karl Pearson's product moment correlation coefficient r and the Spearman's rank correlation coefficient rho ρ.
3. The Pearson's Correlation Coefficient determines the relationship between two variables based on three assumptions.
   1. Linear Relationship
   2. Independent variables
   3. Normal distribution of variables
4. When the assumptions of Pearson's coefficient are not met, Spearman's rho method may be used. This method is based on the ranks given to the observations and not on their actual values
5. Spearman's coefficient is a non-parametric equivalent of the Pearson's coefficient.
6. Spearman's coefficient is robust coefficient and can also be used when one of the variables is ordinal

Karl Pearson's Correlation Coefficient r

Methods can be broadly classified into two types

1. Direct Methods : Type I and Type II
2. Shortcut Method

1. Direct Method

(i) Type I

This method is used when the values are small in magnitude

Formula :
$$r = \frac{N\Sigma XY - \Sigma X \Sigma Y}{\sqrt{N\Sigma X^2 - (\Sigma X)^2}\sqrt{N\Sigma Y^2 - (\Sigma Y)^2}}$$

E.g. For the following data calculate the Karl Pearson's Correlation

Age (Years)	7	8	9	10	11
Weight (kg)	12	14	20	21	18

Age (X)	Weight (Y)	X^2	Y^2	X * Y
7	12	49	144	84
8	14	64	196	112
9	20	81	400	180
10	21	100	441	210
11	18	121	324	198
$\Sigma X = 45$	$\Sigma Y = 85$	$\Sigma X^2 = 415$	$\Sigma Y^2 = 1505$	$\Sigma X * Y = 784$

Substituting the values in above formula we get

$$r = 0.7756$$

(ii) Type II

Type II can be used when $\bar{X}$ and $\bar{Y}$ is not in fraction.

Formula is given by

$$r = \frac{\Sigma xy}{\sqrt{\Sigma x^2 \cdot \Sigma y^2}}$$

where x is the deviation of X from $\bar{X}$, and y is the deviation of Y from $\bar{Y}$. xy is the product of the two deviations, x^2 and y^2 is the square of the deviations

E.g.

Birth rate (X)	death rate (Y)	$x = X - \bar{X}$	$y = Y - \bar{Y}$	x^2	y^2	$x * y$
20	12	−8.5	−6	72.25	36	51
25	16	−3.5	−2	12.25	4	7
27	17	−1.5	−1	2.25	1	1.5
31	20	2.5	2	6.25	4	5
45	21	16.5	3	272.25	9	49.5
23	22	−5.5	4	30.25	16	−22
$\bar{X}$ = 28.5	$\bar{Y}$ = 18	$\Sigma x = 0$	$\Sigma y = 0$	Σx^2 = 395.5	Σy^2 = 70	Σxy = 92

Substituting in the above formula we get

$$r = 0.5529$$

2. Shortcut Method

(a) This method can be used when the mean is in fractions.

(b) Deviations are calculated from the assumed mean and the following formula is applied

$$r = \frac{N\Sigma dx\,dy - \Sigma dx\,\Sigma dy}{\sqrt{N\Sigma dx^2 - (\Sigma dx)^2}\ \sqrt{N\Sigma dy^2 - (\Sigma dy)^2}}$$

where Σdx = Sum of deviations of X series from its Assumed Mean i.e. $\Sigma (X - A_x)$

Σdy = Sum of deviations of Y series from its Assumed Mean i.e. $\Sigma (Y - A_y)$

Σdx^2 = Sum of squared deviations of X series from its Assumed Mean i.e. $\Sigma (X - A_x)^2$

Σdy^2 = Sum of squared deviations of Y series from its Assumed Mean i.e. $\Sigma (Y - A_y)^2$

$\Sigma dx\,dy$ = Sum of products of deviations of X and Y series from their respective assumed means.

$\Sigma dx\,dy = \Sigma (X - A_x)(Y - A_y)$

N = Number of pairs

E.g. Given X and Y calculate the Karl Pearsons Correlation Coefficient

x	32	36	22	21	24	23	18	19
y	23	22	15	16	14	15	15	21

The assumed mean for X is $A_x = 15$ and $A_y = 14$

x	y	$x - A_x = dx$	$y - A_y = dy$	dx^2	dy^2	dxdy
32	23	17	9	289	81	153
36	22	21	8	441	64	168
22	15	7	1	49	1	7
21	16	6	2	36	4	12
24	14	9	0	81	0	0
23	15	8	1	64	1	8
18	15	3	1	9	1	3
19	21	4	7	16	49	28
		$\Sigma dx = 75$	$\Sigma dy = 29$	$\Sigma dx^2 = 985$	$\Sigma dy^2 = 201$	$dx\,\Sigma dy = 379$

Substituting in the above formula we get

$$r = \frac{N\Sigma dx\,dy - \Sigma dx\,\Sigma dy}{\sqrt{N\Sigma dx^2 - (\Sigma dx)^2}\ \sqrt{N\Sigma dy^2 - (\Sigma dy)^2}}$$

$$r = \frac{(8 * 379) - (75 * 29)}{\sqrt{8 * 985 - (75 * 75)} * \sqrt{(8 * 201) - (29 * 29)}}$$

$$r = \frac{857}{1315.13}$$

$$r = 0.6516$$

Spearman's Correlation Coefficient rho (ρ)

1. In this method variables of both the series are provided ranks.
2. These ranks are used for calculating the coefficient of correlation.
3. Ranks of X series are denoted by R_1 and Ranks for Y series are denoted as R_2.
4. The difference between R_1 and R_2 is calculated and denoted as D.
5. Differences are squared up and denoted as D^2.
6. The total of D^2 is obtained and expressed as ΣD^2.

$$r_k = 1 - \frac{6\Sigma D^2}{N^3 - N}$$

Here ΣD^2 = the total of the squares of differences of corresponding ranks

N = Number of pairs of observations

r_k = coefficient of correlation.

Q. 4 List the limitations of linear regression

Ans. :

Limitations of linear regression

1. Linear regression is limited to Linear relationships :

   Linear regression looks at linear relationship between the dependent variable and independent variable. Sometimes this may not be true.

2. Linear regression looks at the mean of the dependent variable.

3. Linear regression is sensitive to outliers :

   Outliers can have a effect on the regression

4. Linear regression assumes the data are independent.

Q. 5 Explain need for regularization techniques in Generalized Linear Models (GLM).

Ans. :

1. No particular distribution is assumed for the dependent variable. The dependent variable may follow distributions like normal, binomial, Poisson.
2. The variance –bias tradeoff is addressed, it generally lowers the variance.
3. The technique is more robust to handle multicollinearity.
4. Sparse data is handled better.
5. Natural feature selection.
6. Overfitting on the trained data is minimized which results in more accurate predictions.

Q. 6 Write short notes on : Ridge regression.

Ans. :

1. Ridge regression is an extension to Linear Regression.
2. This technique shrinks the regression coefficients, which results in variables with minor contribution resulting in their coefficients close to zero.
3. The shrinkage is achieved by a term called as L2-Norm which is used in penalizing the regression model, which is the sum of squared coefficients.
4. The amount of penalty can be fine tuned with lambda (λ) a constant.
5. Selection of a good value for lambda is important.
6. When $\lambda = 0$, the term penalty will have no effect, and ridge regression will be equivalent to ordinary least squares coefficients.
7. As λ increases to a large infinite value the shrinkage penalty grows and the ridge regression coefficients will get close to zero.
8. As compared to Ordinary Least squares regression, Ridge regression is highly sensitive to the scale of predictors.
9. Hence it is better to standardize the scale of the predictors before applying the Ridge regression so that all the predictors are on the same scale.
10. This technique shrinks the coefficients close to zero but it will not set the coefficients exactly to zero.
11. This can be overcome using Lasso regression.
12. The least squares criterion or the cost function is given as

$$\sum_{i=1}^{M} \left(y_i - \hat{y}_i\right)^2$$

This can be rewritten as

$$\sum_{i=1}^{M} \left(y_i - \hat{y}_i\right)^2 = \sum_{i=1}^{M} \left(y_i - \sum_{j=0}^{p} \omega_j \times x_{ij}\right)^2$$

M- represents the instances

p-number of features

1. In Ridge regression , the above cost function is altered by adding a penalty term which is square of the magnitude of the coefficients

$$\sum_{i=1}^{M} \left(y_i - \hat{y}_i\right)^2 = \sum_{i=1}^{M} \left(y_i - \sum_{j=0}^{p} \omega_j \times x_{ij}\right)^2 + \lambda \sum_{j=0}^{M} \omega_j^2$$

2. The constraint on Ridge regression coefficients is given below

$$\text{For some } c > 0, \sum_{j=0}^{p} \omega_j^2 < c$$

3. Ridge regression places a constraint on the coefficients (ω).
4. Ridge regression shrinks the coefficients and it helps to reduce the model complexity and multicollinearity.
5. This technique performs better when the outcome is a function of many predictors, where all the coefficients are of roughly equal size.

Q. 7 Write short notes on : Lasso Regression

Ans. :

Lasso Regression

1. Lasso stands for Least Absolute Shrinkage and Selection Operator.
2. This technique shrinks the regression coefficients towards zero by penalizing the model with a penalty term called as L1-norm.
3. The penalty term is the sum of absolute coefficients.
4. The coefficients with a minor contribution to the model are made exactly to zero by the penalty term.
5. Lasso can be seen as a subset or feature selection method, which can reduce the complexity of the model.
6. Selection of a good value for lambda λ is important.
7. Lasso when compared to Ridge regression model produces more simple and interpretable models, that uses only a reduced set of predictors.
8. Lasso regression performs better in situations where some of the predictors have large coefficients and the remaining have small coefficients.
9. To make a choice between the two techniques (Ridge and Lasso regression), cross validation methods may be used for identifying which of the technique performs better on a particular data set.
10. The cost function for Lasso Regression can be given as follows :

$$\sum_{i=1}^{M} \left(y_i - \hat{y}_i\right)^2 = \sum_{i=1}^{M} \left(y_i - \sum_{j=0}^{p} \omega_j \times x_{ij}\right)^2 + \lambda \sum_{j=0}^{p} |\omega_j|$$

11. The constraint given by the technique is for some

$$t > 0, \quad \sum_{j=0}^{p} |w_j| < t$$

12. This technique leads to zero coefficients, meaning some of the coefficients are completely neglected for evaluating the output.

13. The amount of penalty can be fine tuned with lambda (λ) a constant.

14. So Lasso regression helps in reducing the over fitting and also helps in feature selection.

Q. 8 Write short notes on : ElasticNet regression

Ans. : ElasticNet Regression

The Lasso regression technique sometimes does not perform well with highly correlated data and often performs worse than ridge in prediction. To overcome this drawback, a penalty that combines l1 norm and l2 norm is developed. The result of this is to effectively shrink the coefficients (like in ridge regression) and to set come coefficients to zero like lasso regression.

The penalty for Elastic net is given as

$$\lambda \left[\frac{1}{2}(1-\alpha) \sum_{j=1}^{p} \beta_j^2 + \alpha \sum_{j=1}^{p} |\beta_j| \right]$$

Here $\alpha \in [0,1]$ is called the mixing parameter and λ has the same interpretation. For $\alpha = 0$ we have the Ridge Regression and for $\alpha = 1$ we have the Lasso regression.

Q. 9 Write short notes on :Robust regression with random sample consensus.

Ans. :

Robust Regression with Random Sample Consensus

1. Robust regression with Random Sample Consensus (RANSAC) is a general algorithm which can be used with other parametric estimation methods to obtain robust models.
2. RANSAC is aimed to determine function parameters when the data has gross erroneous samples that can mislead the parameter estimation.
3. The assumption RANSAC makes is that the training data consists of inliers that can be explained with the model and outliers that are gross erroneous samples that do not fit the model at all.
4. If this assumption does not hold , RANSAC will not harm the parameter estimation as in this condition it will consider the whole dataset as inliers.
5. It trains the model using only the inliers while ignoring the outliers.
6. To reject the outliers, RANSAC uses small set of samples to train a model instead of using the whole of the data. It then enlarges the set with appropriate samples.
7. Steps followed in RANSAC(General version)
   (a) A subset of data is sampled uniformly at random (i.e. the minimum number of points needed to estimate the model).
   (b) Using the sampled subset, estimate the parameters for the model of choice.
   (c) Error is calculated for all the remaining samples using an error function.
   (d) The number of inliers are calculated (i.e. all samples below a threshold error).
   (e) Recompute the Model using all inliers and hypothesis if the number of inliers are above a given threshold.

   Repeat the above to find the best model.

Q. 10 Explain advantages and disadvantages of robust regression with random sample consensus.

Ans. :

Advantages

1. The method is simple and general.
2. It can be applied to many different problems.
3. It often works well in practice.

Disadvantages

1. Tuning of parameters is required.
2. Too many iterations are needed in some cases.
3. The method can fail if there is very low inlier ratio.

Q. 11 Write short notes on : Polynomial regression

Ans. : Polynomial Regression

1. **Polynomial regression** is a form of linear regression.
2. An n^{th} order Polynomial is used to model the relationship between the independent variable x and the dependent variable y.
3. It fits a non-linear relationship between the value of x and conditional mean of y represented as E(y|x).
4. It is considered to be a special case of multiple linear regression.
5. We know that the Linear regression model is represented as

$$Y = \beta_0 + \beta_1 x + \varepsilon$$

6. This model can be used for fitting any relationship that is linear in the unknown parameters β_1.
7. In many cases such conditions may not hold, the relationship may be curvilinear between x and y. In such a cases an important class of Polynomial regression models may be used.

E.g. The second order polynomial in one variable

$$Y = \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon$$

8. In two variables the second order polynomial can be written as

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 + \varepsilon$$

9. Polynomial regression models are used when the response is curvilinear.

Q. 12 Define Bias and variance.

Ans. :

1. Bias

(a) The error due to the simplistic assumptions made by the model in fitting the data is called Bias.

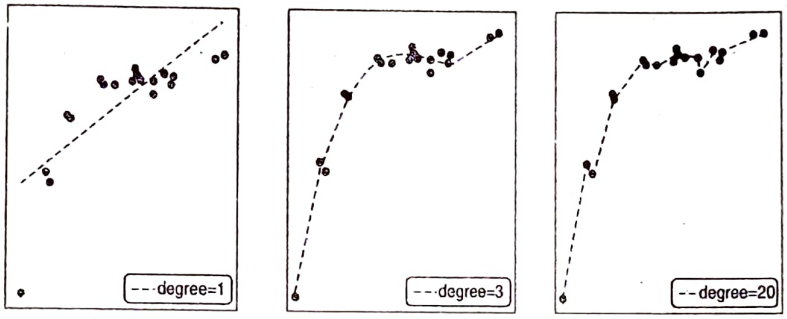
(b) A high value of bias indicates that the model is unable to capture the pattern in the data, this is known as under fitting.

2. Variance

(a) A complex model trying to fit the data would result in an error, this is known as variance.

(b) A high value of variance would mean the model passes through most of the data points, but it results in overfitting.

(c) A graphical representation of the same is shown in Fig. 3.2



Under fit
High Bias
Low Variance

Correct fit
Low Bias
Low Variance

Over fil
Low Bias
High Variance

Fig. 3.2

Q. 13 Write short notes on : Isotonic regression.

Ans. :

Isotonic Regression

1. Isotonic Regression is a regression technique that belongs to the family of regression algorithms
2. There may be datasets, which have non decreasing points, presenting low level oscillation (noise).
3. In such a situation, a linear regression model cannot capture the internal dynamics of the data.
4. In such situations we can make use of Isotonic Regression which produces piecewise interpolating function minimizing the functional

$$f(x) = \sum_i w_i \left(y_i - \hat{y}_i\right)^2$$

where $y_0 \le y_1 \le y_2 \le \ldots\ldots \le y_n$

E.g.

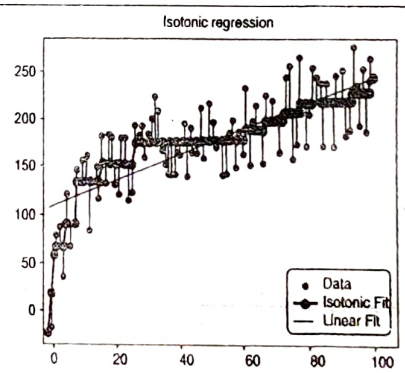


Fig. 3.3

Q. 14 Write short note on : Linear classification.

Ans. :

Linear classification is a classification algorithm that makes its classification based on a Linear predictor function, combining a set of weights with feature vector.

$$Y = f\left(\sum_j w_i x_j\right)$$

In this classifier the decision boundary is flat (e.g. Line, plane). Let us consider the task of Binary Classification. The goal is to predict a binary value target.

Example includes :

1. A medical diagnosis system to predict whether a patient is suffering from a given disease.
2. Whether a given email is spam or not spam.
3. Whether a given transaction is fraudulent or not fraudulent.
4. A binary classifier computes a linear function of inputs, and determine whether or not the value is larger than some threshold th.
5. A linear function of the input can be written as

$$w_1 x_1 + ...+w_D x_D + b = w^T x + b$$

where w is a weight vector and b is a scalar – valued bias

6. The prediction y can be computed as

$$z = w^T x + b$$

$$y = \begin{cases} 1 & \text{if } z \geq th \\ 0 & \text{if } z < th \end{cases}$$

Example of Linear Classifier

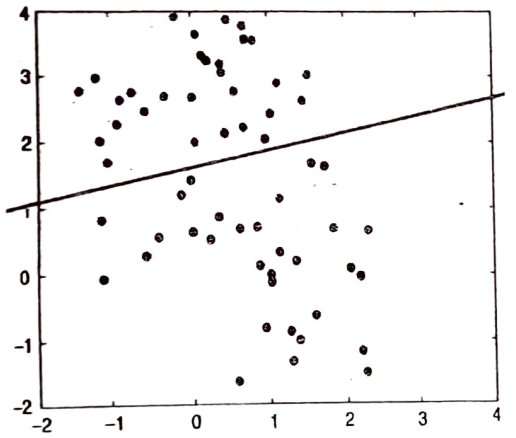


Fig. 3.4

Q. 15 Explain Logistic Regression.

Ans. :

Logistic regression

1. **Logistic Regression** is a statistical technique for analyzing a dataset that predicts the probability of an outcome that can only have two values (i.e. a dichotomy)
2. The goal is to find the best fitting model to describe relationship between the dichotomous characteristics of interest (dependent variable) and a set of independent variable(predictor or explanatory variable)
3. Logistic regression seeks to
   (a) Model the probability of an event occurring depending on the values of the independent variables , which can be categorical or numerical.
   (b) Estimate the probability that an event occurs for a randomly selected observation versus the probability that the event does not occur.
   (c) Predict the effect of a series of variables on a binary response variable.
   (d) Classify observations by estimating the probability that an observation is in particular category (such as approved or not approved)
4. Linear Regression method has some major problem,
   (a) Binary data does not have a normal distribution, which is a condition needed for most other types of regression.
   (b) Predicted values of the dependent variable can be beyond 0 and 1, which violates the definition of probability.
5. In logistic regression the predicted value has to be between 0 and 1.

Probability Review for Logistic Regression

I. Probability

The probability is given as

$$p = \frac{\text{Outcomes of Interest}}{\text{all possible outcomes}}$$

E.g. Probability of getting a head from a Fair coin Flip is given as

$$p(\text{heads}) = \frac{1}{2} = 0.5$$

E.g. Probability of pulling out a diamond from a Deck of playing cards

$$p(\text{diamond}) = \frac{13}{52} = \frac{1}{4} = 0.25$$

II. Odds

Odds is the probability of something occurring divided by the probability of not occurring and is given as

$$\text{odds} = \frac{P(\text{occuring})}{P(\text{not occuring})}$$

$$\text{odds} = \frac{p}{1-p}$$

E.g. The odds of getting a head for flipping a fair coin

$$\text{odds(heads)} = \frac{0.5}{0.5} = 1$$

E.g. The odds of pulling out a diamond from a Deck of playing cards

$$\text{odds(heads)} = \frac{0.25}{0.75} = 0.333$$

Odds Ratio

Odds ratio is ratio of two odds and is given as

$$\text{odds(heads)} = \frac{\text{odds}_1}{\text{odds}_0}$$

The dependent variable is binary its either 0 or 1. We need to link the probability between 0 and 1 back to our independent variables. The dependent variable in Logistic Regression follows the Bernoulli distribution having an unknown probability p. Bernoulli distribution is a special case of the Binomial distribution

Where, n = 1 (just one triai) ;

Success is 1 and failure is 0

So the probability of success is p and failure is q = 1 – p.

In logistic regression we are estimating an unknown p for any given linear combination of the independent variables. So we need to link the independent variables to Bernoulli distribution, that link is called as Logit. In Logistic Regression we do not know the value of p. So the Goal of Logistic Regression is to estimate p for a linear combination of the independent variables. Estimate of p is $\hat{p}$ (p-hat). The natural log of the odds ratio, the Logit is that Link function that will map the Linear combination of variables to Bernoulli distribution that could result in a value between 0 and 1. This is given by

$$ln(\text{odds}) = ln\left(\frac{p}{1-p}\right) \text{ is the logit (p) OR}$$

$$\ln(p) - ln(1-p) = \text{logit}(p)$$

This can be graphically represented as shown in Fig. 3.5

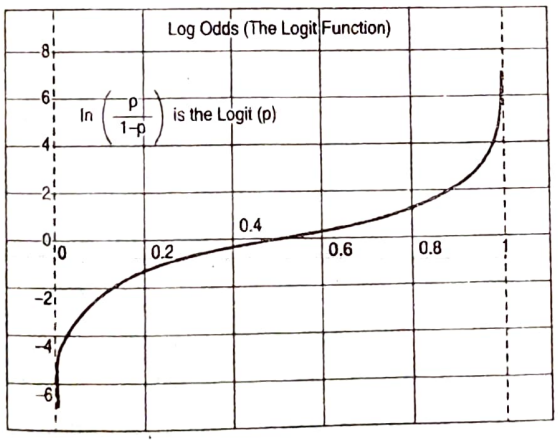


Fig. 3.5

When p = 0, ln(0) is undefined

When p = 1, is again going to undefined

But when p = 0.5, ln(1) = 0, that is what we can see graphically this shows a sigmoid curve.

Inverse Logit

In the above graphical representation our probabilities are shown on x-axis , we want them on y-axis . we can get this by taking the inverse of the logit function

We know that

$$\text{logit}(p) = ln\left(\frac{p}{1-p}\right)$$

where p is between 0 and 1

when we take the inverse of logit(p) we get

$$\text{logit}^{-1}(p) = \frac{1}{1+e^{-\alpha}} = \frac{e^{\alpha}}{1+e^{\alpha}}$$

α = some number

We can represent this graphically as shown in Fig. 3.6

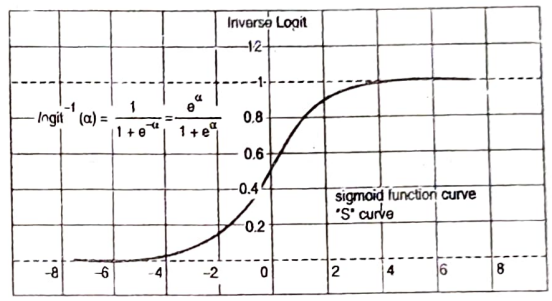


Fig. 3.6

The Estimated Regression equation

The antilog of the logit function allows us to find the estimated regression equation.

We know that

$$\text{logit}(p) = ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1$$

antilog will give us

$$\frac{p}{1-p} = e^{\beta_0+\beta_1 x_1}$$

$$p = e^{\beta_0+\beta_1 x_1}(1-p)$$

$$= e^{\beta_0+\beta_1 x_1} - e^{\beta_0+\beta_1 x_1} * p$$

$$p + e^{\beta_0+\beta_1 x_1} * p = e^{\beta_0+\beta_1 x_1}$$

$$p(1+e^{\beta_0+\beta_1 x_1}) = e^{\beta_0+\beta_1 x_1}$$

$$\hat{p} = \frac{e^{\beta_0+\beta_1 x_1}}{(1+e^{\beta_0+\beta_1 x_1})}$$

This is the estimated regression equation.

Q. 16 Explain Stochastic Gradient descent technique.

Ans. :

Stochastic gradient descent technique

1. Optimization is the task of minimizing / maximizing an objective function f(x) parameterized by x

2. In machine learning it refers to the task of minimizing the cost/loss function, parameterized by model's parameters.
3. The goal of optimization algorithms is to have one of the following

(a) Finding the global minimum of the convex objective function

(b) Finding the lowest possible value of the non convex objective function within its neighborhood

4. Gradient descent is the most common optimization algorithm in machine learning
5. Machine Learning models have parameters (e.g. weights and biases) and a cost function to evaluate the goodness of particular set of parameters.
6. During training the goal is to find the predicted values close to the target values so that cost J(W) is minimum.
7. Gradient descent is an iterative method used to minimize the cost function J(W) parameterized by a model parameters W.
8. The gradient or derivative tells the slope/cost of the function, In order to reduce it, an opposite direction is chosen.
9. Let us consider for e.g. a logistic regression model having two parameters weight w and bias b

Step 1 : Initialize the weight w and bias b to random values.

Step 2 : Pick a value for the learning rate α.

10. Learning Rate determines the step size for each iteration :

(a) If α is small, it would take longer to converge and also it would be computationally expensive.

(b) If α is large, it may fail to converge.

Step 3 :

On each iteration take the partial derivate of the cost function J(W) with respect to each parameter e.g. in this case weight and bias

$$\frac{\partial}{\partial w} J(w) = \nabla_w J$$

And $$\frac{\partial}{\partial b} J(w) = \nabla_b J$$

Step 4 : Update the Equations

$$w = w - \alpha \nabla_w J$$

$$b = b - \alpha \nabla_b J$$

Let us ignore say bias, if the slope of the current value of $w > 0$, this means we are to the right of the optimal w^*. Therefore update will be negative and start going close to optimal value. However if the value of $w < 0$ then the update will be positive and this will increase the value of w and converge to the optimal value of w^*. This is shown in the Fig. 3.7.

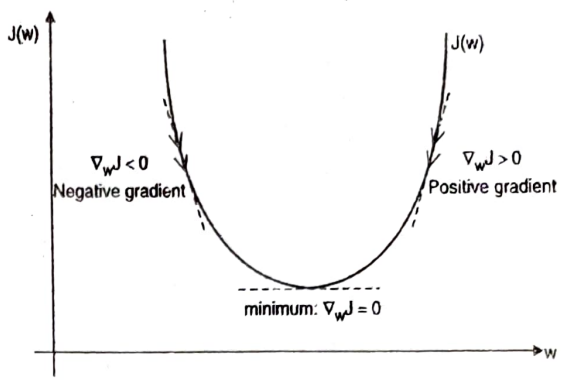


Fig. 3.7

Step 5 : Continue the process until convergence

Example of Linear Regression with Gradient descent

$$h_\theta(x) = \sum_{j=0}^{n} \theta_j x_j$$

$$J_{train}(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Repeat {

$$\theta_j = \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

(for every j = 0,....,n)

where $h(\theta)$ is the hypothesis function.

J_{train} is the cost function and

Repeat { } block is to update the θ parameter.

Stochastic gradient descent

1. When there is a large training data set, Gradient descent becomes computationally expensive.
2. In such cases a modification to gradient descent algorithm called as stochastic gradient descent can be used.
3. In Stochastic gradient descent unlike gradient descent, all of training data is not used, only a single example is used.
4. Example of Linear Regression with Stochastic Gradient descent is given below :

$$cost(\theta, x^{(i)}, y^{(i)}) = \frac{1}{2}(h_\theta(x^{(i)}) - y^{(i)})^2$$

$$J_{train}(\theta) = \frac{1}{m} \sum_{i=1}^{m} cost(\theta, (x^{(i)}, y^{(i)}))$$

Steps in Stochastic Gradient descent

1. Randomly shuffle (reorder) training examples
2. Repeat {

for i : = 1,....,m {

$$\theta_j := \theta_j - \alpha (h_\theta(x)^{(i)} - y^{(i)}) x_j^{(i)}$$

(for every j = 0,....,n)

}

}

Q. 17 Explain different classification metrics.

Ans. :

Classification metric

1. Evaluating machine learning model is essential.
2. Below are different performance metrics used to evaluate classification algorithms.

Confusion Matrix

1. It is used to find the correctness and accuracy of the model.
2. It is used in classification problem where the output is two or more types of classes.
3. It is a table with two dimensions ("Actual" and "Predicted") and sets of classes in both dimensions.
4. The Actual dimension is presented along the rows and Predicted dimension along the columns.
5. E.g. Consider a Binary classification problem

	Predicted class C_1 (YES)	Predicted class $\neg C_1$ (NO)
Actual class C_1 (YES)	True Positives (TP)	False Negatives (FN)
Actual class $\neg C_1$ (NO)	False Positives (FP)	True Negatives (TN)

The terms in the confusion matrix are :

1. **True Positives (TP)** : The cases in which the prediction was YES and the actual output was also YES.
2. **True Negatives (TN)** : The cases in which the prediction was NO and the actual output was NO.
3. **False Positives (FP)** : The cases in which the prediction was YES and the actual output was NO.
4. **False Negatives (FN)** : The cases in which the prediction was NO and the actual output was YES.

Accuracy

Accuracy is the percentage of test tuples that are correctly classified.

It is calculated as follows :

$$\text{Accuracy} = \frac{\text{True Positive + True Negative}}{\text{Total number of Samples}}$$

Error Rate

The Error rate may be calculated as follows :

$$\text{Error Rate} = 1 - \text{Accuracy}$$

Or

$$\text{Error rate} = \frac{\text{False Positive + False Negative}}{\text{Total number of Samples}}$$

Sensitivity

True positive recognition rate

$$\text{Sensitivity} = \frac{\text{True Positive}}{\text{Total Number of Positive samples}}$$

Specificity

True positive recognition rate

$$\text{Specificity} = \frac{\text{True Negative}}{\text{Total Number of Negative samples}}$$

Precision

It is the exactness that the classification algorithm gives. It is the percentage of tuples that the classifier labeled as positive are actually positive. The formula to calculate Precision is given below :

$$\text{Precision} = \frac{\text{True Negative}}{\text{True Positive + False positive}}$$

Recall

Recall is also the completeness. It is the percentage of positive tuples did the classifier labeled as positive. The formula to calculate Recall is given below :

$$\text{Recall} = \frac{\text{True Negative}}{\text{True Positive + False positive}}$$

F-measure (F_1 or F-Score)

It is the harmonic mean of precision and recall

$$F = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision + recall}}$$

Example for Classification Metric

Actual Class\Predicted class	cancer = yes	Cancer = no	Total	Recognition (%)
cancer = yes	90	210	300	30.00 (*sensitivity*)
cancer = no	140	9560	9700	98.56 (*specificity*)
Total	230	9770	10000	96.40 (*accuracy*)

$$\text{Precision} = \frac{90}{230} = 39.13\% \; ; \quad \text{Recall} = \frac{90}{300} = 30.00\%$$

Q. 18 Write short note on ROC Curve.

Ans. :

ROC Curve

1. ROC curve stands for Receiver Operating Characteristics Curve.
2. Using ROC curve one can visually compare classification models.
3. ROC curve has its originating root from signal detection theory.
4. A trade-off between the true positive rate and the false positive rate is shown on ROC curve
5. The accuracy of the model can be measured by the area under the ROC curve.
6. Vertical axis represents the true positive rate and horizontal axis represents the false positive rate.
7. The model with perfect accuracy will have an area of 1.0.

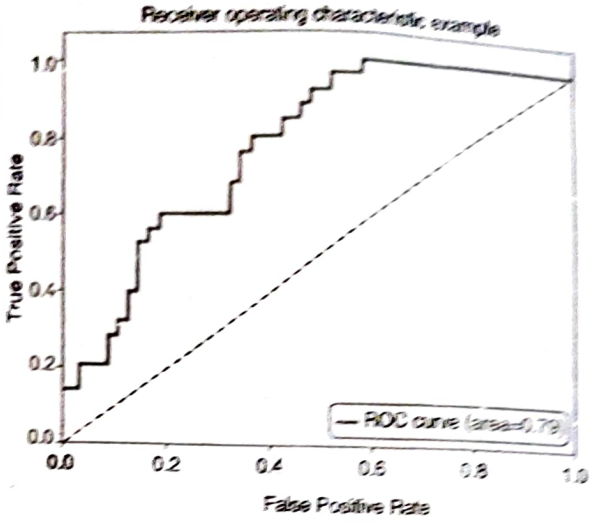


Fig. 3.8

Chapter 4 : Naïve Bayes and Support Vector Machine

Q. 1 Explain Bayes Theorem.

Ans. : Bayes Theorem

1. Consider $X=\{x_1,x_2,x_3,\ldots x_n\}$ as a sample, whose components represent values made on a set of n attributes.
2. In Bayes Theorem X is considered evidence. Let H be a Hypothesis, such that data X belongs to a class C.
3. In classification, the goal is to determine P(H|X), the probability that the hypothesis H holds given the evidence (observed data sample X).
4. In other words, the probability that sample X belongs to class C.
5. P(H|X) is called as a Posteriori probability of H conditioned on X, and is given as :

$$P(H|X) = \frac{P(X|H)\,P(H)}{P(X)}$$

Where P(H) is the a priori Probability which is independent of X.

P(X) is the a priori probability of X.

P(X|H) is the a posteriori probability of X conditioned on H.

6. Let us understand this concept with the help of an example.
7. Suppose we have a data set of customers. In this dataset age and income are the attributes and buys_computer as a class with values as Yes and No.
8. Let us consider a sample X{ age : 45 years, income : Rs. 50,000}, we have to classify this unseen tuple as buys_computer = yes or buys_computer = No.
9. P(H|X) is the probability that the customer X will buy a computer given his age and income, in our case age = 45 and income = Rs. 50,000.
10. P(H) is the probability any customer will buy a computer regardless of his age and income, it is independent of X.
11. P(X) is the probability from the data set that the customer is 45 years old and earns Rs. 50,000.
12. P(X|H) is the probability that a customer , is 45 years old and he earns Rs. 50,000, given that he will buy a computer.

Maximum A Posteriori (MAP) Hypothesis

Based on Bayes Rule, we can compute the Maximum a Posteriori Hypothesis for the data :

$$h_{map} = \arg\max_{h \in H} P(h|D)$$

$$= \arg\max_{h \in H} \frac{P(D|h)\,P(h)}{P(D)}$$

$$= \arg\max_{h \in H} P(D|h)\,P(h)$$

H : set of all Hypothesis

P(D) can be dropped as the probability of the data is constant(and independent of hypothesis)

Maximum Likelihood

1. Now assume that all hypotheses are equally probable a priori, i.e., $P(h_i) = P(h_j)$ for all $h_i, h_j \in H$.
2. This is called assuming a uniform prior. It simplifies computing the posterior.

$$h_{ML} = \arg\max_{h \in H} P(D|h)$$

3. This hypothesis is called maximum likelihood hypothesis.

Q. 2 Give properties of Bayes Classifiers.

Ans. :

1. Instrumentality

The prior and likelihood can be updated dynamically with each training example.

Flexible and robust to errors.

2. Combines prior knowledge and observed data

Given the training data, the prior probability of a hypothesis is multiplied with probability of the hypothesis.

3. Probabilistic hypotheses

The output of Bayes classifier includes classification as well as probability distribution over all classes.

4. Meta-classification

The output of several classifiers can be combined together.

For e.g. the probabilities of all classifiers predicted for a given class can be multiplied together.

Q. 3 Explain the different variants of Naïve Bayes in scikit learn library.

Ans. :

1. Based on the same number of different probabilistic distributions, scikit – learn provides implementation of three variants of Naïve bayes : Bernoulli Naïve Bayes, Multinomial Naïve bayes and Gaussian Naïve Bayes.
2. Bernoulli naïve bayes is a binary distribution and this type is useful when a feature is present or absent.
3. Multinomial Naïve Bayes is a discrete distribution and is useful when a feature needs to represented using a whole number.
4. The Gaussian distribution is a continuous distribution and is characterised by mean and variance.

1. Bernoulli naïve bayes

(a) Consider X is a random variable having Bernoulli distribution.

(b) It can assume only two values (say for e.g. 0 and 1) and their probability is given as follows :

$$P(X) = \begin{cases} p & \text{if } X = 1 \\ q & \text{if } X = 0 \end{cases}$$

Where $q = 1 - p$ and $0 < p < 1$

E.g. : Implementation of Bernoulli Naïve Bayes in Scikit-Learn. Generating a dummy data set :

```
from sklearn.datasets import make_classification
nb_samples = 300
X, Y = make_classification(n_samples=nb_samples,
n_features=2, n_informative=2, n_redundant=0)
```

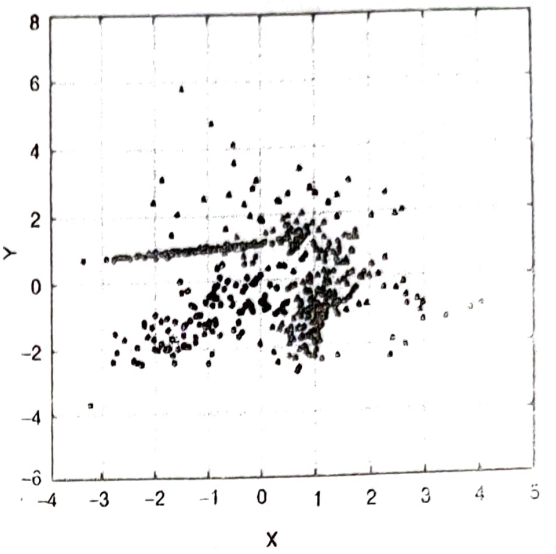


Fig. 4.1 . Dummy data generated

1. Bernoulli naive Bayes needs binary feature vectors.
2. The binarize parameter in BernoulliNB class allows using a threshold that can be used internally to transform the features to binary
3. Let us use 0.0 as the binary threshold.

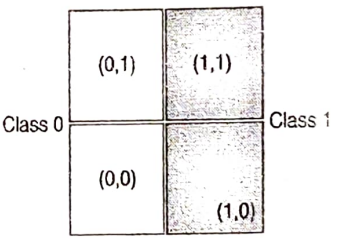


Fig. 4.2 : Internally binarized data

```
from sklearn.naive_bayes import BernoulliNB
from sklearn.model_selection
import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.25)
bnb= BernoulliNB(binarize=0.0)
bnb.fit(X_train, Y_train)
print(bnb.score(X_test, Y_test))
```

Output

```
0.85333333333333339
```

Checking for Prediction

```
data = np.array([[0, 0], [0, 1], [1, 0], [1, 1]])
print(bnb.predict(data))
```

Output

```
array([0, 0, 1, 1])
```

2. Multinomial naïve bayes

(a) Multinomial Naïve Bayes is a model used when the value represents the number of occurrences of a term or its relative frequency.

(b) If a feature vector has n elements and each of them have k different values with probability p_k then

$$P(X_1 = x_1 \cap X_2 = x_2 \cap .... \cap X_k = x_k)$$

$$= \frac{n!}{\prod_i x_i!} \prod_i p_i^{x_i}$$

(c) To prevent the model from setting null probabilities when the frequency is zero , an alpha parameter (called Laplace smoothing factor) is used whose default value is 1.0

(d) E.g in scikit-learn. There are two records a first record is of city and the second one of countryside.

(e) The output class for city is 1 and countryside is 0.

```
from sklearn.feature_extraction import DictVectorizer
data = [{'house': 100, 'street': 50, 'shop': 25, 'car': 100, 'tree': 20}, {'house': 5, 'street': 5, 'shop': 0, 'car': 10, 'tree': 500, 'river': 1}]
dv = DictVectorizer(sparse=False)
X = dv.fit_transform(data)
Y = np.array([1, 0])
>>>X
#OUTPUT
array([[ 100., 100., 0., 25., 50., 20.], [ 10., 5., 1., 0., 5., 500.]])
```

The term river is missing from the first set, so alpha is set to 1.0. Training the Multinomial Naïve Bayes

```
from sklearn.naive_bayes import MultinomialNB
mnb = MultinomialNB()
mnb.fit(X, Y) MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
```

Testing the model

```
test_data = data = [ {'house': 80, 'street': 20, 'shop': 15, 'car': 70, 'tree': 10, 'river': 1},
{'house': 10, 'street': 5, 'shop': 1, 'car': 8, 'tree': 300, 'river': 0}
]
mnb.predict(dv.fit_transform(test_data))
#Output
array([1, 0])
```

3. Gaussian Naïve Bayes

(a) Gaussian Naïve Bayes is used when the values are continuous whose probabilities can be modeled using a Gaussian Distribution.

(b) Conditional Probabilities are also Gaussian Distributed, mean and variance needs to be estimated of each of them using maximum likelihood approach.

(c) Using the Gaussian property we get,

$$L(\mu; \sigma^2; x_i | y) = \log \prod_k P(x_i^{(k)} | y) = \sum_k \log P(x_i^{(k)} | y)$$

k- refers to the sample in the data set

An example, comparing Gaussian Naive Bayes with logistic regression using the ROC curves. The dataset has 300 samples with two features. Each sample belongs to a single class :

```
from sklearn.datasets import make_classification
nb_samples = 300
X, Y = make_classification(n_samples= nb_samples, n_features=2, n_informative=2, n_redundant=0)
```

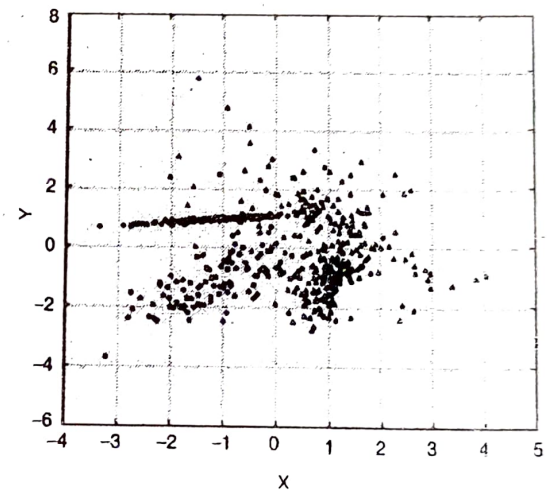


Fig. 4.3 : A plot of dataset

Training both the models and generating ROC curves

```
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import roc_curve, auc
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.25) >>>gnb = GaussianNB()
gnb.fit(X_train, Y_train)
Y_gnb_score = gnb.predict_proba(X_test)
lr = LogisticRegression()
lr.fit(X_train, Y_train)
Y_lr_score = lr.decision_function(X_test)
fpr_gnb, tpr_gnb, thresholds_gnb = roc_curve(Y_test, Y_gnb_score[:, 1])
fpr_lr, tpr_lr, thresholds_lr = roc_curve(Y_test, Y_lr_score)
```

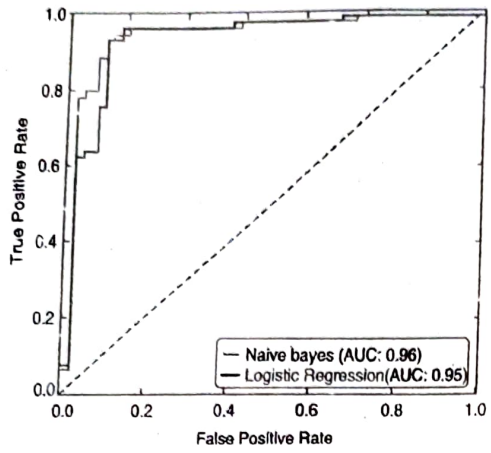


Fig. 4.4

As can be seen from the Fig. 4.4, Naïve bayes performs slightly better than Logistic regression, both the classifiers have similar accuracy and AUC (Area Under Curve).

Q. 4 State of Application of Naïve Bayes.

Ans. : Application of Naïve Bayes

1. Real-time Prediction

Due to high speed of Naïve bayes algorithm, it can be used for making predictions in real time.

2. Multi-class Prediction

The posterior probability of multiple classes of the target variable can be predicted by the algorithm.

3. Text classification/ Spam Filtering/ Sentiment Analysis

(a) Due to their better results in multiclass problems and independence rule, Naïve Bayes classifier is suitable for text classification.

(b) It has a high success rate compared to other algorithms.

(c) Due to its success rates they are widely used in spam filtering and sentiment analysis applications.

4. Recommendation System

Naïve Bayes can be combined together with collaborative filtering to make a recommendation system in machine learning and data mining to filter the unseen information and predict the users likes and dislikes.

Q. 5 State of Algorithm Advantages of Naïve Bayes

Ans. : Algorithm advantages

1. The algorithm performs well for multiclass prediction.
2. The prediction of class on test data is faster and it is also easy to apply.
3. A Naïve Bayes classifier performs better when compared to other classifiers like e.g. Logistic regression when the assumption of independence holds, it also requires less amount of training data.
4. The algorithm performs better for categorical input variable(s) when compared to numerical variable(s). The numerical variable assumes a normal distribution.

Q. 6 State of Algorithm disadvantages of Naïve Bayes.

Ans. :

Algorithm disadvantages

1. The model will be unable to make a prediction if the categorical variable has a category in test data, which was not available in training data.
2. In such a case the model assigns a zero (0) probability. This is called as "Zero Frequency". This problem can be overcome with a smoothing technique. One of the simplest method used for smoothing is called Laplace estimation.
3. The assumption of independent predictors in Naïve Bayes algorithm is a limitation as in real world it is almost impossible to get a set of predictors which are independent.

Q. 7 Explain Support vector Machines.

Ans. :

Support Vector Machine (SVM)

1. Support vector Machine is another method used for classification.
2. It can classify both Linear as well as Non Linear Data.
3. The objective of SVM is to find a hyperplane (A decision boundary separating the tuples of one class from another) in an N - dimensional space (where N represents the number of features) that distinctly classify the data points.

E.g.

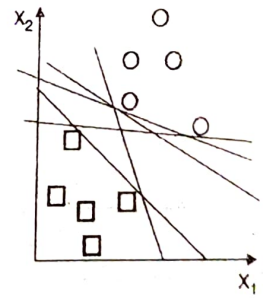


Fig. 4.5 : All possible Hyperplanes to classify the data points

4. As we can see in the above diagram, there are many possible hyperplanes that could be chosen, but the main objective is to find a plane that has maximum margin i.e. maximum distance between data points of both classes as shown in the Fig. 4.6.
5. Maximizing provides reinforcement so that future data points can be classified with more confidence.

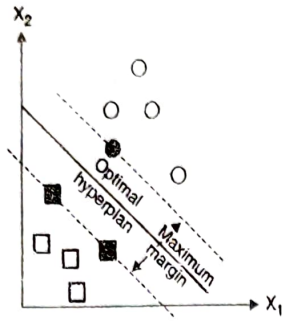


Fig. 4.6 : Optimal hyperplane having maximum margin

Working of support vector machine

1. SVM uses a Non Linear mapping to transform the original training data into a higher dimension.
2. In this dimension it searches for a linear optimal separating hyperplane.
3. It finds this hyperplane using support vectors (essentially training tuples) and margins (defined by support vectors).
4. In SVM, if the output of the Linear function is greater than 1, it is identified as one class and if the output is -1 it is identified with another class.

Support vectors

1. Support vectors are data points that are closer to hyperplane.
2. These support vectors are used to maximize the margin of the classifier.
3. Deletion of these support vectors will change the position of the Hyperplane.

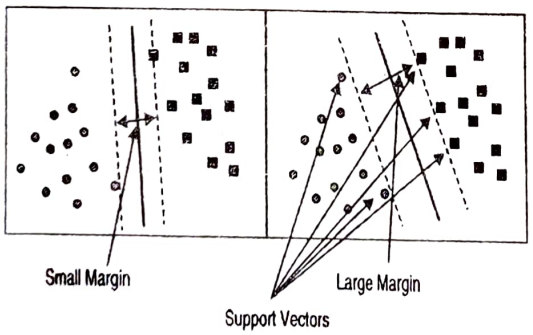


Fig. 4.7 : Support vectors and Margin separating the data points

Q. 8 State Advantages and disadvantages and applications of SVM

Ans. :

Advantages

1. SVM is a highly accurate method due to the ability of modeling complex non-linear decision boundaries.
2. When compared to other methods, SVM is less prone to overfitting.
3. The support vectors of the model provide a description of the learned model.
4. SVM can be used for prediction (numeric) and also as a classifier.

Disadvantages

1. The training time of even the fastest SVMs can be extremely slow on large data sets.
2. Less effective on noisier datasets with overlapping classes.

Applications

1. Handwritten digit recognition.
2. Object recognition.
3. Speaker identification.
4. Benchmark time-series prediction tests.

Q. 9 Explain kernel based classification in Support vector machines.

Ans. : Kernel based classification

1. Some problems cannot be solved using Linear hyper plane for e.g. as shown below.
2. In this type of cases, SVM uses a kernel trick to transform the input space to a higher dimensional as shown in the Fig. 4.8(b).
3. The data points are plotted on x-axis and z-axis, now these points can be easily segregated.

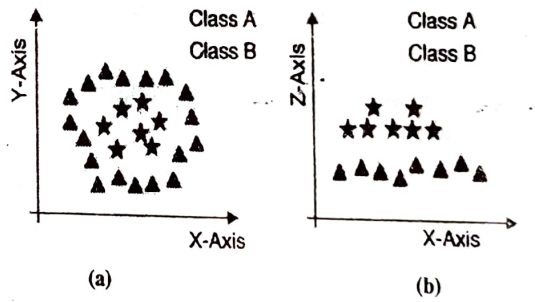


Fig. 4.8

4. The kernel takes a low dimensional space and transforms it into higher dimensional space.
5. It converts a non-separable problem into a separable problem by adding more dimensions to it.
6. This is most useful in non-linear separation problem.

Following are the different kernels used in SVM :

Linear Kernel

1. A Linear kernel is a dot product of any two given observations.
2. The product between two vectors is the sum of the multiplication of each pair of input values.

Polynomial Kernel

1. A more generalized form of Linear kernel is Polynomial kernel.
2. The polynomial kernel can differentiate between curved or nonlinear input space.

Radial Basis Function Kernel

1. The input space can be mapped in infinite dimensional space by RBF kernel.
2. It is a local kernel and can create complex regions within the feature space like e.g. closed polygons in 2D space.

Sigmoid Kernel

1. The sigmoid kernel has its origin from Neural networks.
2. Neural network approach as we know is used for classification of input data.

Kernel Functions

$$K(X_i, X_j) = \begin{cases} X_i \cdot X_j & \text{Linear} \\ (\gamma X_i \cdot X_j + C)^d & \text{Polynomial} \\ \exp(-\gamma |X_j - X_j|^2) & \text{RBF} \\ \tanh(\gamma X_i \cdot X_j + C) & \text{Sigmoid} \end{cases}$$

where $K(X_i, X_j) = \phi(X_i) \cdot \phi(X_j)$

Q. 10 Explain Support vector regression.

Ans. : Support vector regression

1. The method of Support Vector machines can be extended to solve regression problems. This is called as Support Vector Regression.
2. As discussed above the model produced by Support Vector machine for classification uses only a subset of training data, as the cost function for building the model does not consider the data that lie beyond the margin.
3. Similarly the model produced by Support vector regression also depends on a subset of training data as the cost function for building the model ignores any data close to the model prediction.
4. In Scikit-learn, there are three different implementations of Support vector regression SVR, NuSVR and LinearSVR.
5. LinearSVR is faster than SVR but only considers Linear Kernels.
6. NuSVR implements slightly different from SVR and LinearSVR.

E.g. of SVR in Scikit-learn

```
from sklearn import svm
>>>X = [[0, 0], [2, 2]]
>>>y= [0.5, 2.5]
>>>clf=svm.SVR()
>>>clf.fit(X, y)
SVR(C=1.0, cache_size=200, coef0=0.0, degree=3,
epsilon=0.1,gamma='auto_deprecated', kernel='rbf',
max_iter=-1, shrinking=True,tol=0.001, verbose=False)
>>>clf.predict([[1, 1]]) array([1.5])
```

Chapter 5 : Decision Trees and Ensemble Learning

Q. 1 Explain different impurity measures.

Ans. : Impurity measures

1. Gini impurity index

(a) Suppose all attributes are continuous-valued.

(b) Assume that each value of an attribute has many possible splits.

(c) It can be adapted for categorical attributes.

(d) Gini is used in CART (Classification and Regression Trees), IBM's IntelligentMiner system, SPRINT (Scalable PaRallelizable INduction of decision Trees).

(e) If a data set T contains examples from n classes, Gini index, Gini (T) is defined as

$$\text{gini}(T) = 1 - \sum_{j=1}^{n} p_j^2$$

(f) Where, p_j is the relative frequency of class j in T.

(g) gini(T) is minimized if the classes in T are skewed.

(h) Gini index of split is defined below after split T1 and T2 of sizes N1 and N2 respectively.

$$\text{gini}_{split}(T) = \frac{N_1}{N} \text{gini}(T_1) + \frac{N_2}{N} \text{gini}(T_2)$$

(i) For every attribute, each of the possible binary splits is considered. For a discrete-valued attribute, the attribute providing smallest gini_{split} (T) is picked out to split the node. For continuous-valued properties, each possible split-point must be believed.

(j) The Gini impurity measures the probability of a misclassification if a label is randomly selected using the probability distribution of the branch. If all the samples belong to a single category, then index reaches its minimum (0.0).

2 Cross-entropy impurity index

(a) **Cross-entropy impurity index** is based on Information theory.

(b) When all the samples belong to one class are represented in a split, then null values are assumed. But if there is uniform distribution, then it is maximum.

(c) It permits to select the split which actually minimizes uncertainty about classification.

$$\text{Entropy} = -\sum_{j} p_j \log_2 p_j$$

(d) Information Gain (IG), determine which attribute in a given set of training feature vectors is most useful. We will use it to decide the ranking of attributes in the nodes of a decision tree.

(e) The Information Gain (IG) can be defined as follows:

(f) All attributes are considered to be categorical.

(g) It can be adapted for continuous-valued attributes.

(h) The attribute which has the highest information gain is selected for split.

(i) Assume there are two classes, P and N.

(j) Consider S samples, out of these p samples belongs to class P and n samples belongs to class N.

(k) The amount of information, needed to decide if random example in S belongs to P or N is defined as

$$I(p, n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

(l) Assume that using attribute A, a set S will be partitioned into sets $\{S_1, S_2, \ldots, S_v\}$.

(m) If S_i contains p_i examples of P and n_i examples of N, the entropy, or the expected information needed to classify objects in all sub trees S_i is

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n} I(p_i, n_i)$$

Entropy (E)

The expected amount of information (in bits) needed to assign a class to a randomly drawn object in S under the optimal, shortest-length code.

Calculate information gain i.e. gain (A) : Measures reduction in entropy, achieved because of the split. Take the split that achieves most reductions (maximizes GAIN)

$$\text{Gain}(A) = I(p, n) - E(A)$$

3. Misclassification impurity index

The misclassification impurity is the simplest index, defined as :

$$I_E = 1 - \max\{p(i|t)\},$$

It is a useful criterion for pruning but not recommended for growing a decision tree since it is less sensitive to changes in the class probabilities of the nodes.

Q. 2 What is Ensemble Learning?

Ans. : Ensemble learning

Ensemble learning combines various set of learners (individual models) together which actually improvise on the stability and predictive power of the model.

1. Combining classifier is an ensemble method which increases the accuracy.
2. To get new improved model M*, combine a series of n learned models, $M_1, M_2, \ldots, M_n$

Popular ensemble methods are :

1. Bagged (or Bootstrap) trees

(a) Base learners are generated in parallel, so it is a parallel ensemble method (e.g. Random Forest).

(b) The basic motivation of parallel methods is to exploit independence between the base learners since the error can be reduced dramatically by averaging.

(c) For example, we can train M different trees on different subsets of the data (chosen randomly with replacement) and compute the ensemble :

$$f(x) = 1/M \sum_{m=1}^{M} f_m(x)$$

(d) In generalized bagging, different learners can be used in different population to reduce the variance error.

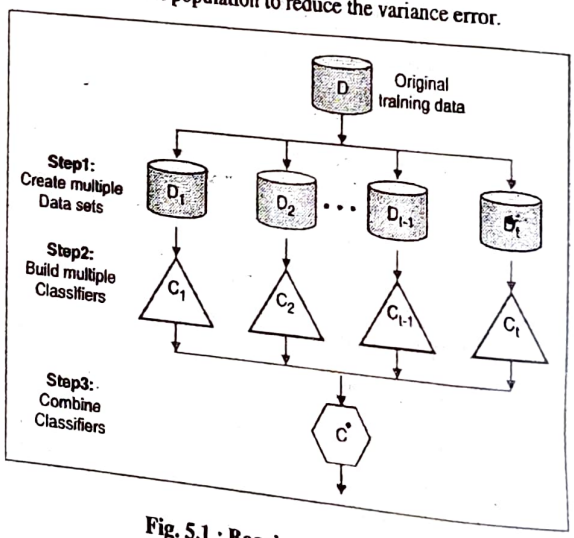


Fig. 5.1 : Bagging or bootstrap

(e) It averages the prediction from the accumulation of various classifiers used.

(f) A bootstrap method is used, for data set D of n tuples, for each iteration n tuple are sampled with replacement from D.

(g) In every iteration , a classifier model M is learned from training data set

(h) For unknown sample Y, each classifier gives class prediction.

(i) The bagged classifier M* uses the voting method i.e. the sample tuple Y is assigned the class with the most votes to tuple Y.

(j) For continuous values, it can be used for prediction by taking the average of all predictions for a given sample.

2. Boosted trees

(a) It is a family of algorithms that are able to convert weak learners to strong learners.

(b) It is a sequential ensemble method where the base learners are generated sequentially (e.g. AdaBoost).

(c) The basic motivation of sequential methods is to exploit the dependence between the base learners The overall performance can be boosted by weighing previously mislabelled examples with higher weight.

(d) In boosting, each training tuple has weight.

(e) n number of classifiers are learned iteratively.

(f) After learning of M_i classifier, every time the weights are updated for next classifier learning i.e. M_{i+1}. So if the tuples which were misclassified by M_i will get higher weight for next classifier.

(g) Use voting method, where check the votes of each classifier to get the final M* which helps to get the accuracy.

(h) The extended boosting algorithm works for the prediction of continuous values.

(i) Boosting tends to accomplish greater accuracy as compared to bagging, there is a risk of overfitting the model.

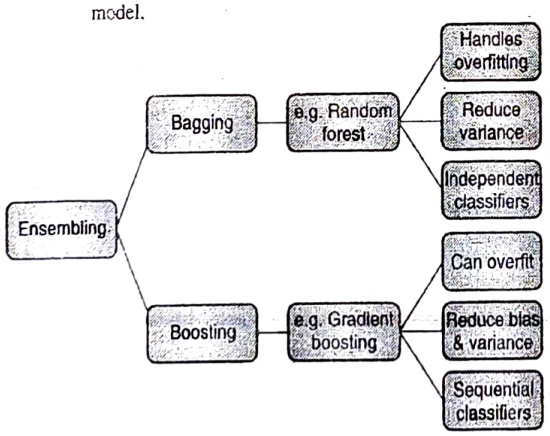


Fig. 5.2 : Ensemble Learning

Q. 3 Explain Random Forest algorithms in detail

Ans. :

Random forest

1. It is a supervised classification algorithm.
2. It creates the forest with a number of trees as the name suggests.
3. If the number of trees in the forest are more, the accuracy result is high.
4. Random forest handles the missing values.
5. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features.
6. The algorithm selects random subset of features to split the node.

Random Forest pseudocode

1. Select "k" features randomly from total "m" features where k << m.
2. Using the bet split point, calculate the node d from selected k features.
3. Using the best split, split the node d into child nodes.
4. Repeat 1 to 3 steps until "l" number of nodes has been reached.
5. Build forest by repeating steps 1 to 4 for "n" number times to create "n" number of trees.

Random forest prediction pseudo code

1. Predict and store the outcome using the rules of each randomly created decision tree.
2. Count on the votes for each predicted target.
3. The last prediction is selected by taking the high voted predicted outcome.

With two trees, you can see how a random forest would look like

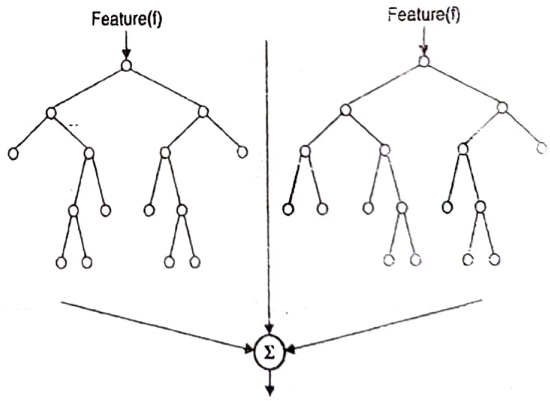


Fig. 5.3 : Random forest with 2 trees

Advantages

1. It can be used for both regression and classification tasks.
2. It is a very handy and easy to use algorithm as it's default hyper parameters often produce a good prediction result.
3. The classifier won't overfit the model if there are enough trees in the forest.

Disadvantages

1. Many trees are generated which makes algorithm slow and not suited for real time prediction.
2. It's not a descriptive tool, but only predictive model.
3. If dataset is noisy , then random forest may overfit.
4. For data, including categorical variables with different number of levels, random forests are biased in favour of those attributes with more levels. Therefore, the variable importance scores from random forest are not reliable for this type of data.

Q. 4 Explain Ada Boost algorithms in detail

Ans. : Ada boost

1. It combines weak classifier algorithm to form strong classifier, then by selecting, training set at every iteration multiple classifiers are combined which gives good accuracy score.
2. Correct amount of weight can be assigned in final voting, which improves accuracy.

Algorithm

Step 1 : Choose the training set and train the algorithm.

Step 2 : Retrains the algorithm iteratively by selecting another set of training data based on the accuracy of previous training.

Step 3 : The weight-age depends on the accuracy achieved for each trained classifier.

Step 4 : It assigns weight to each training item.

Step 5 : Higher weights are assigned to misclassified item so that those items can appear in the training subset of next classifier with higher probability.

Step 6 : After training weights is assigned to each classifier also based on accuracy.

Step 7 : Higher weights are assigned to more accurate classifier to get more impact on the final outcome.

Step 8 : The mathematical formula for Adaboost is given below.

Where,

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

T : Number of classifiers

$h_t(x)$: Output of weak classifier t for input x

α_t : Weight assigned to the classifier.

α_t is calculated as follows:

$\alpha_t = 0.5 * \ln((1-E)/E)$ where E is error rate.

Step 9 : Initially, all the input training examples has equal weightage. The mathematical formula to update the weight of each training example is given below.

$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

Where,

D_t : weight at previous level.

Z_t : sum of all weights used to normalize the weights.

y_i is y par of training example (xi, yi) y coordinate for simplicity.

Q. 5 Explain K-Means algorithms in detail.

Ans. :

K-means

1. In 1967, J. MacQueen and then in 1975 J. A. Hartigan and M. A. Wong developed K-means clustering algorithm.
2. In k-means, k is the number of clusters given by user and objects are classified into k clusters based on their attributes.
3. K-means is one of the simplest unsupervised learning algorithms.
4. Define K centroids for K clusters which are generally far away from each other.
5. Group the objects into clusters based on the distance with respect to centroid
6. After this first step, again calculate the new centroid for each cluster based on the elements of that cluster.
7. Follow the same method and group the elements based on new centroid.
8. At every step, the centroid changes and elements move from one cluster to another.
9. Do the same process till no element is moving from one cluster to another i.e. till two consecutive steps with the same centroid and the same elements are obtained.
10. Finally, this algorithm aims at minimizing an *objective function*, in this case a squared error function. The objective function is given below,

$$J = \sum_{j=1}^{k}\sum_{i=1}^{n} \| x_i^{(j)} - c_j \|^2$$

Where, $x_i^{(j)}$ = A data point

c_j = The cluster centre

n = Number of data points

k = Number of clusters

$\| x_i^{(j)} - c_j \|^2$ = Distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j

K-means Algorithm

k : number of clusters

n : sample feature vectors $x_1, x_2, ..., x_n$

m_i : the mean of the vectors in cluster i

1. Assume k < n.
2. Make initial guesses for the means $m_1, m_2, ..., m_k$.
3. Until there are no changes in any mean.
4. Use the estimated means to classify the samples into clusters.

for i = 1 to k

Replace m_i with the mean of all of the samples for cluster i end_for

end_until

Following the three steps are repeated until convergence:

Iterate till no object moves to a different group :

Step 1 : Find the centroid coordinate. **Step 2 :** Find the distance of each object to the centroids.

Step 3 : Based on minimum distance group the objects.

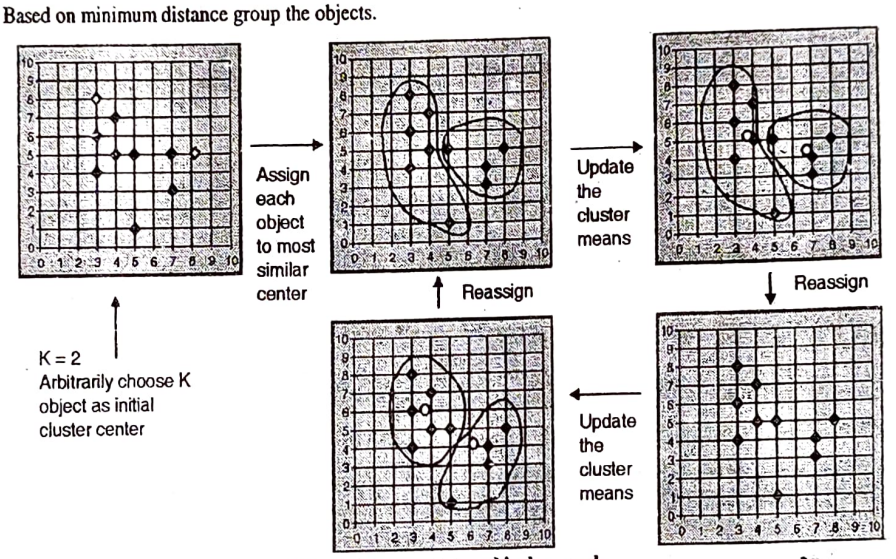


Fig. 5.3 : K-means graphical example

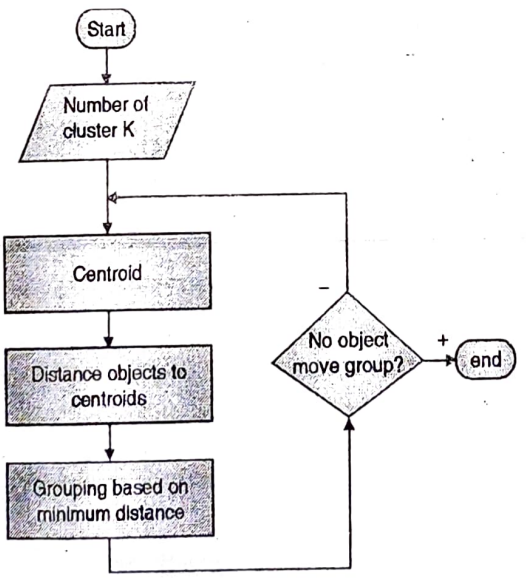


Fig. 5.4 : Basic steps for K-means clustering

Given a cluster $K_i = \{t_{i1}, t_{i2}, ..., t_{im}\}$, the cluster mean is $m_i = (1/m)(t_{i1} + ... + t_{im})$

Q. 6 What are different methods to determine number of clusters? Explain.

Ans. : Finding optimal number of clusters

The disadvantage of k-means is to find the optimal number of clusters. If the number of clusters are less, then large element grouping can be formed with heterogeneous elements. With the more number of clusters, it will be difficult to get the dissimilarities among clusters.

Therefore, different methods to determine the number of clusters are

1. Optimizing the inertia
2. Silhouette score
3. Calinski-Harabasz index
4. Cluster instability

1. Optimizing the inertia

(a) The assumption that an appropriate number of clusters must produce a small inertia.

(b) When number of clusters is same as number of data elements, this value reaches its minimum (0.0).

(c) So don't look for the minimum, but for a value which is a trade-off between the inertia and the number of clusters.

(d) This method calculates the inertias for different number of clusters.

2. Silhouette score

(a) **Silhouette score is** a way to measure how the elements within a cluster are close to the points in its neighbouring clusters.

(b) It is based on the principal of "maximum internal cohesion and maximum cluster separation" means how similarly an object is in its own cluster (cohesion) compared to other clusters (separation).

(c) It finds the optimal value of k (number of clusters) during clustering.

(d) Define a distance metric and compute the average intra-cluster distance for each element :

$$a(\bar{x}_i) = E_{\bar{x}\, j \in C}[d\, \bar{x}_i, \bar{x}_j] \; \forall \; \bar{x}_i \in C$$

(e) Calculate the average nearest-cluster distance which is the lowest inter-cluster distance.

$$b(\bar{x}_i) = E_{\bar{x}\, j \in D}[d\, \bar{x}_i, \bar{x}_j] \; \forall \; \bar{x}_i \in C \text{ where } D$$

$$= \text{argmin}\{d(C, D)\}$$

(f) The silhouette score for an element x_i is defined as :

$$S(\bar{x}_i) = \frac{b(\bar{x}_i) - a(\bar{x}_i)}{\max\{a(\bar{x}_i), b(\bar{x}_i)\}}$$

(g) This value of silhouette score is bounded between -1 and 1.

(h) Score closer to 1 means assigned to the cluster correctly and score closer to – 1 is assigned to a wrong cluster. A score close to 0 means the point lies between almost at the boundary of both the clusters.

(i) scikit-learn allows computing the average silhouette score to have an immediate overview for different numbers of clusters.

3. Calinski-Harabasz index

(a) It is a concept of dense and well-separated clusters.

(b) It is required to define the inter cluster dispersion initially.

(c) The inter-cluster dispersion (BCD) is defined as :

$$BCD(k) = Tr(B_k)$$

$$\text{where} \quad B_k = \sum_t n_t (\mu - \mu_j)^T (\mu - \mu_j)$$

(d) Where k is the cluster with their relative centroids μ_i and the global centroid μ, n_t is the number of elements belonging to the cluster k.

(e) The intra-cluster dispersion (WCD) is defined as :

$$WCD \quad (k) = Tr(X_k)$$

$$\text{where} \quad X_k = \sum_t \sum_{x \in C_k} (x - \mu_t)^T (x - \mu_t)$$

(f) The Calinski-Harabasz index is defined as the ratio between *BCD(k)* and *WCD(k)* :

$$CH(k) = \frac{N-k}{k-1} \cdot \frac{BCD(k)}{WCD(k)}$$

(g) We have to find the number of clusters which gives a maximum index Calinski-Harabasz index.

4. Cluster instability

(a) It is based on the concept of cluster instability defined in Von Luxburg U.

(b) If we have a dataset X, we can define a set of m perturbed (or noisy) versions :

$$X_n = \{X_n^0, X_n^1, \ldots, X_n^m\}$$

(c) Considering a distance metric $d(C(X_1), C(X_2))$ between two clusterings with the same number (k) of clusters, the instability is defined as the average distance between couples of clusterings of noisy versions :

$$I(C) = E_{n^{i,j} \in Xn}[d(C(X_n^i), C(X_n^j))]$$

(d) We need to find the value of k that minimizes I(C) (and therefore maximizes the stability).

Q. 7 Explain DBSCAN algorithms in detail.

Ans. : DBSCAN

DBSCAN : Density Based Spatial Clustering of Applications with Noise.

The algorithm DBSCAN, based on the formal notion of density-reachability for *k*-dimensional points, is designed to discover clusters of arbitrary shape. The runtime of the algorithm is of the order O(n log n) if region queries are efficiently supported by spatial index structures, i.e. at least in moderately dimensional spaces.

Explanation of DBSCAN Steps

1. Epsilon (Eps) and Minimum points (MinPts) are the two parameters needed by DBSCAN. An unvisited point is chosen as the starting point. Eps between the starting point and its neighbors is calculated and the points within it are considered.
2. If the number of data points in the neighbourhood is greater than or equal to MinPts then a cluster is formed. The starting point chosen is marked as visited.

3. The above steps are then repeated for all the remaining neighbours.
4. If the data points found in the neighbourhood are less than MinPts then they are marked as noise.
5. If all the points within reach in a cluster are visited then the algorithm proceeds by choosing other remaining invested points in the dataset.

Basic concept

1. For any cluster, we have :
2. A central point (p) i.e. core point.
3. A distance from the core point (Eps).
4. Minimum number of points within the specified distance (MinPts).

Major features

(i) Discover clusters of arbitrary shape.
(ii) Handles noise.
(iii) One scan.
(iv) Need density parameters as termination condition.

DBSCAN method

(a) Clusters of arbitrary shape and size are grown which are dense.
(b) The algorithm is as follows :
  (1) Core objects and density reachable objects are found out, merge these density reachable core objects and their clusters are discovered.
  (2) When no new points can be added to any of the clusters the execution may be stopped.
(c) Clusters are dense regions of objects separated by regions of low density (noise).
(d) Outliers will not affect the creation of cluster.

Input

(1) MinPts : Minimum number of points in any cluster.
(2) ϵ : For each point in the cluster there must be another point in its less than this distance away.

ϵ-neighborhood : Points within ϵ distance of a point.

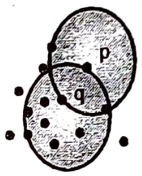
$N\epsilon(p)$: {q belongs to D | dist(p,q) <= ϵ }

Core point : ϵ - neighborhood dense enough (MinPts)

Directly density-reachable : A point p is directly density-reachable from a point q if the distance is small (ϵ) and q is a core point.

(1) p belongs to $N\epsilon(q)$
(2) core point condition :

$|N\epsilon(q)| >$ = MinPts



MinPts = 5

Eps = 1cm

Issues

1. One of the limitations is that, the user has to set the MinPts and Eps threshold. This needs a good knowledge of the dataset. Sometimes in high dimensional data set, it is difficult to decide.
2. Some of the dataset distribution may be globally inconsistent. E.g. some of the area in the dataset may be too dense compared to the other areas, some of the sections may not have clusters or noise may be present.
3. The process is extremely sensitive to noise which leads to very different clusters.

Chapter 6 : Clustering Techniques

Q. 1 Explain Hierarchical Clustering basic algorithm.

Ans. :

The set of data objects is decomposed hierarchically using certain criteria. In hierarchical clustering, algorithms either follow the top down or bottom up approach.

Hierarchical clustering technique (Basic algorithm)

1. Calculate the distance and find a distance matrix or proximity matrix
2. Consider each data point or object as a cluster.
3. Repeat.
4. Combine the two closest clusters.
5. Update the proximity matrix.
6. Until only a single cluster remains.

Q. 2 Explain difference between Agglomerative and Divisive Hierarchical Clustering

Ans. :

Difference between Agglomerative and Divisive Hierarchical Clustering

Agglomerative	Divisive
Start by considering each individual element as a cluster.	Start with only one cluster by combining all elements.
At every step, clusters are merged based on closest pair until a single or k clusters are left	At every step, the cluster is divided or split until each cluster contains a single element or k clusters are left.

Q. 3 Explain Advantages and disadvantages Hierarchical Clustering basic algorithm.

Ans. : Advantages

1. This algorithm is simple and gives an output as a hierarchy
2. The structure obtained is easy to understand and more informative.
3. There is no need to pre-specify the number of clusters

Disadvantages

1. Merging and splitting is critical once the clusters are formed, as undo is not possible. Every time it performs on newly generated clusters.
2. If decision to merge and split is wrong, then low quality clusters are formed
3. Scikit-learn implements only the agglomerative clustering as the complexity of divisive is higher than Agglomerative and has similar performance of Divisive approach.

Q. 4 Explain Expectation Maximization Clustering algorithms in detail.

Ans. : Expectation maximization clustering

(a) The Expectation Maximization (EM) algorithm can be applied to bring forth the best theory for the distributional parameters of some multi-modal data.

(b) The best hypothesis for the distributional parameters is the maximum likelihood hypothesis – the one that maximizes the probability that this data we are looking at comes from K distributions, each with a mean m_k and variance $sigma_k^2$

(c) The Expectation Maximization (EM) Algorithm deal with missing labels by alternating between two steps :

1. **Expectation (E)** : Fix model and estimate missing labels.
2. **Maximization (M)** : Fix missing labels (or a distribution over the missing labels) and find the model that maximizes the expected log-likelihood of the data.

General EM Algorithm

The alternate steps until model parameters don't change much :

1. **E step :** Estimate distribution over labels given a certain fixed model.
2. **M step :** Choose new parameters for model to maximize expected log-likelihood of observed data and hidden variables.

Formal Setup for General EM Algorithm

Let $D = \{x^{(1)},....x^{(n)}\}$ be n observed data vectors.

Let $Z = \{z^{(1)},....z^{(n)}\}$ be n values of hidden variables (i.e. the cluster labels)

Log-likelihood of observed data given model :

$$L(\theta) = \log p(D|\theta) = \log \sum_{Z} p(D,Z|\theta)$$

Where theta is a vector of unknown parameters.

In clustering, K (number of clusters) are not known initially. We can produce two results for the clustering :

Hard clustering : In this method , data object or observation i belongs to only one cluster, so clusters can be produced like { C1, C2, ...Ck}.

Soft clustering : The method says that data object or observation is more likely to belong to one of the K cluster by producing a probability distribution :

$$\mathbb{P}[X_i \in C_k] = \gamma_{k,i}$$

$$\text{with } \sum_{k=1}^{K} \gamma_{k,i} = 1$$

From Fig. 6.1, we can observe that hard clustering is possible on the left side figure. But right side figure is not clear, so only based on probability samples can be classified to one of the class. This is the purpose of soft clustering. Algorithms like hierarchical clustering or K means produce some Hard clustering. The purpose of the EM clustering is to propose a Soft clustering method.

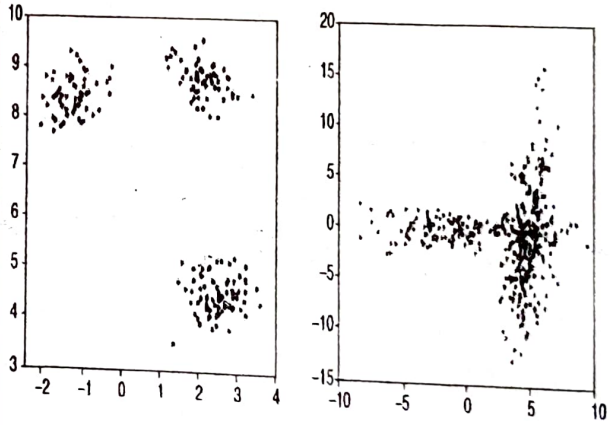


Fig. 6.1 : Visual point of view hard clustering and soft clustering

The basic steps for the algorithm are :

1. An initial guess is made for the model's parameters and a probability distribution is made. This is sometimes called the **"E-Step"** for the **"Expected" distribution"**.
2. Newly observed data is fed into the model.
3. The chance distribution of the E-step is tweaked to let in the fresh information. This is sometimes called the "M-step".
4. Steps 2 through 4 are repeated until stability (i.e. a distribution that doesn't change from the E-step to the M-step) is reached.

Q. 5 What are various strategy to Aggregate Different Clusters.

Ans. :

Strategy to Aggregate Different Clusters

Different approaches to defining the distance between clusters distinguish the different algorithms, but scikit learn supports the three most common ones i.e. Complete Linkage, Average Linkage and Ward's linkage.

1. **Single-linkage**

Single Linkage clustering is also called as minimum method, the minimum distance from any object of one cluster to any object of another cluster is considered. In the single linkage method, **D(A,B)** is computed as

D(A,B) = Min {d(i,j)

Where object i is in cluster A and object j is in cluster B}

This measure of inter-group distance is illustrated in the Fig. 6.2.

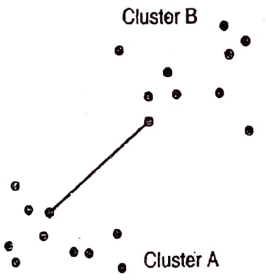


Fig. 6.2 : Single-linkage clustering

2. **Complete linkage**

(a) **Complete linkage** also called as maximum method, the maximum distance between any object of one cluster to any object of another cluster is considered.

(b) In the complete linkage method, **D(A,B)** is computed as

D(A,B) = Max { d(i, j) :

Where object i is in cluster A and object j is in cluster B}

(c) The measure is illustrated in the Fig. 6.3.

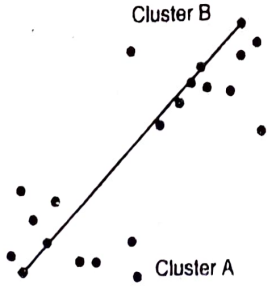


Fig. 6.3 : Complete-linkage clustering

3. **Average-linkage**

In average-linkage clustering, we consider the distance between any two clusters A and B is taken to be the average of all distances between pairs of objects "i" in A and "j" in B, that is, the mean distance between elements of each cluster.

In the average linkage method, **D(A,B)** is computed as :

D(A,B) = mean{d(i,j) :

Where object i is in cluster A and object j is in cluster B}

The Fig. 6.4 illustrates average linkage clustering.

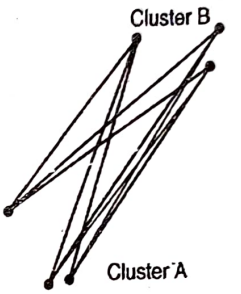
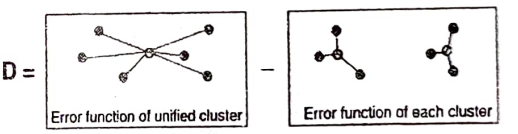


Fig. 6.4 : Average-linkage clustering

4. **Ward's linkage**

In this method, the error function is defined for every cluster. This error function is the average (RMS) distance of each data-point in a cluster to the centre of gravity in the cluster.



The distance (D) between two clusters is defined as the error function of the unified cluster minus the error functions of the individual clusters.

The ward's linkage method computes the distance using formula,

$$\forall C_i, C_j \; L_{ij} = \sum_{x_a \in C_i} \sum_{x_b \in C_j} \|x_a - x_b\|^2$$

The Ward's linkage supports only the Euclidean distance.

5. **Centroid clustering**

In centroid method, the distance between two clusters is calculated by finding the distance between two centroids (i.e. mean value of cluster) of the clusters. At every step, two clusters are combined that have minimum centroid distance.

In the centroid clustering method, D(A,B) is computed as

$$D(A,B) = d(A_{centroid}, B_{centroid})$$

Where $A_{centroid}$ is the mean value or centroid of cluster A and $B_{centroid}$ is the mean value or centroid of cluster B. The Fig. 6.5 illustrates centroid clustering.

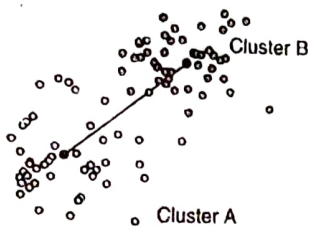


Fig. 6.5 : Centroid clustering

Q. 6 What is Dendrogram?

Ans. :

Dendrograms

An agglomerative clustering is typically visualized as a *dendrogram* as shown in Fig. 6.6 where each merge is represented by a horizontal line. Dendrogram is the aggregation of clusters from bottom to top. Initially leaf nodes are single clusters, which are merged till a single root node or cluster generated.

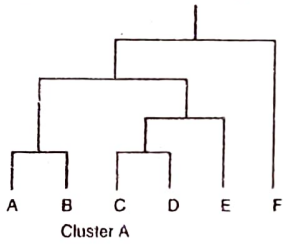


Fig. 6.6 : Dendrogram

A tree data structure which illustrates hierarchical clustering techniques. Each level shows clusters for that level.

1. Leaf – individual clusters
2. Root – one cluster

A cluster at level i is the union of its children clusters at level i + 1. A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component form a cluster. Unfortunately, scikit-learn doesn't support the dendrogram but SciPy provides some useful built-in functions.

The program in Python is given below to create dendrogram.

```
# program to visualize dendrogram
from sklearn.datasets import make_blobs
nb_samples = 25
X, Y = make_blobs(n_samples=nb_samples, n_features=2,
centers=3,cluster_std=1.5)
# computing a distance matrix chosen a Euclidean metric
from scipy.spatial.distance import pdist
Xdist = pdist(X, metric='euclidean')
# linkage used is Ward
from scipy.cluster.hierarchy import linkage
Xl = linkage(Xdist, method='ward')
# create and visualize a dendrogram
from scipy.cluster.hierarchy import dendrogram
Xd = dendrogram(Xl)
```

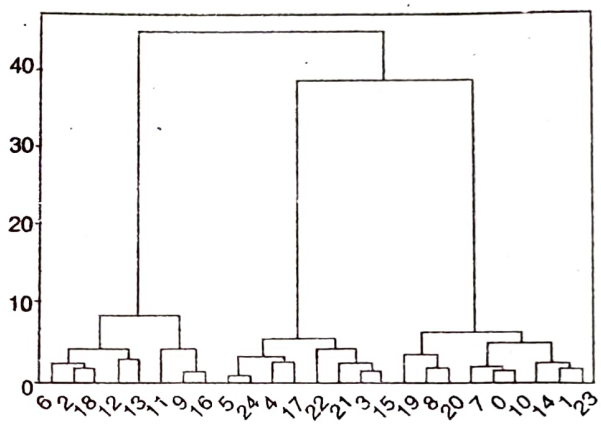


Fig. 6.7 : Snapshot of resulting plot of dendrogram

Q. 7 Write short note on Connectivity Constraints

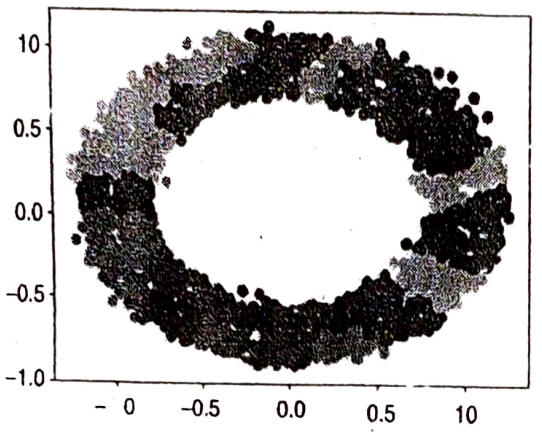
Ans. :

Connectivity Constraints

An interesting aspect of Agglomerative Clustering is that connectivity constraints which can be added to this algorithm where only adjacent clusters can be merged together and those clusters which are distant i.e. non-adjacent are skipped. Connectivity matrix can be used as a constraint by Scikit-learn and find the clusters to merge. These constraints are useful to impose a certain local structure, but they also make the algorithm faster, especially when the number of the samples is high.

K-Nearest Neighbour graph function is a common method used. Consider circular dummy dataset in the example given below.

```
from sklearn.datasets import make_circles
import matplotlib.pyplot as plt
from sklearn.cluster import AgglomerativeClustering
nb_samples = 3000
X, _ = make_circles(n_samples=nb_samples, noise=0.05)
ac = AgglomerativeClustering(n_clusters=20,
linkage='average')
ac.fit(X)
plt.scatter(X[:,0],X[:,1], c=ac.labels_, cmap='rainbow')
```




```
# Now we can try to impose a constraint with different values for k:
from sklearn.neighbors import kneighbors_graph
acc = []
k = [50, 100, 200, 500]
for i in range(4):
    kng = kneighbors_graph(X, k[i])
    ac1 = AgglomerativeClustering(n_clusters=20, connectivity=kng,linkage='average')
    ac1.fit(X)
    plt.scatter(X[:,0],X[:,1], c=ac1.labels_, cmap='rainbow')
    acc.append(ac1)
```

Imposing a constraint based on K-Ncarest Neighbours, allows controlling how the agglomeration creates new clusters and can be a powerful tool for tuning the models, or for avoiding elements whose distance is large in the original space could be taken into account during the merging phase. It is helpful in clustering the images.

Q. 8 Write short note on Recommendation Systems

Ans. :

Recommendation Systems

Recommendation system is used to predict future preferences of a set of items and also recommend the top n items to users. Many e-commerce websites uses recommendation system to increase their sales by using the power of data. This system finds the user's interest to recommend items. Traditionally, there are two methods to construct a recommendation system :

1. User or Content-based recommendation
2. Collaborative Filtering

1. **Content-based recommendation system:** It makes recommendations using a user's item and profile features. To find the user's future interest, the system needs the information about user's past interest in the items. So only similar items can be grouped together based on their features.
2. **Collaborative Filtering systems:** It filters the items in which we are interested. This system is based on the assumption that if user1 likes item X and user2 likes the same item X plus another item Y, then user1 may be interested in item Y. So system can use historical data to predict new interactions.

Q. 9 Explain in detail Content Based Systems

Ans. :

Content Based Systems

Recommendations are based on information on the content of items rather than on other users' opinions. Uses machine learning algorithms to induce a profile of the user's preferences from examples based on content features.

1. No need for data about other users.
2. No cold-start or sparsity problems.
3. Able to recommend to users with unique tastes.
4. No first-rater problem.

Cold Start : enough users in the system to find a match.

Sparsity : The user/ratings matrix is sparse, and it is hard to find users that have rated the same items.

First Rater : Not for an item that has not been previously rated. This is one of the simplest methods which is based on products and modelled as feature vectors :

$$I = \left\{ \bar{l}_1, \bar{l}_2, \ldots, \bar{l}_n \right\} \text{ where } \bar{l}_n \in \mathbb{R}^n$$

The features can also be categorized like users. For example, the type of a book or a movie, and they can be used together with numerical values (like price, length, number of positive reviews, and so on) after encoding them.

Q. 10 Explain in detail Alternating Least Squares

Ans. :

Alternating least squares

Alternating Least Squares (ALS) is a model use to fit our data and find similarities. Least square optimization problem is defined as a loss function to find the latent factor as given below.

$$L = \sum_{(i,j)} \left(r_{ij} - \bar{p}_i \cdot \bar{q}_j^T \right)^2 + \alpha \left(\|\bar{p}_i\|^2 + \|\bar{q}_j\|^2 \right)$$

L can be identified by using known samples (user, item). The second term of the above equation is a regularization factor. Any optimization method can be used to solve the whole problem. Two main iterating steps of algorithm are :

1. p_i is fixed and q_j is optimized
2. q_j is fixed and p_i is optimized

Q. 11 What is Deep Learning? Explain.

Ans. :

Defining Deep Learning

'Deep Learning' means using a neural network with several layers of nodes between input and output. It is the series of layers between input and output for feature identification and processing in a series of stages, just as our brains seem to.

Uses of Deep Learning

Annually designed features are incomplete and takes long time, but learned features are adaptable and fast to learn. Deep learning provides a very flexible, (almost?) Universal, a learnable framework for representing the world, visual and linguistic information. Can learn both unsupervised and supervised. Effective end-to-end joint system learning utilize large amounts of training data.

❑❑❑