SKP Engineering College

Tiruvannamalai – 606611

A Course Material

on

Data analytics

By

K.Baskar & Raji V

Assistant Professor

Computer Science and Engineering Department

Quality Certificate

This is to Certify that the Electronic Study Material

Subject Code: IT 6006

Subject Name: Data Analytics

Year/Sem: IV / VII

Being prepared by me and it meets the knowledge requirement of the University curriculum.

Signature of the Authors

Name: Mr.K.Basker& Raji V

Designation: Assistant Professor

This is to certify that the course material being prepared by Mr K.Basker & Raji V are of the adequate quality. He has referred more than five books and one among them is from abroad author.

Signature of HDSignature of the Principal

Name:K.BaskarName: Dr.V.Subramania Bharathi

Seal:Seal:

.

Prerequisites

Are you intrigued by the world of Big Data innovations, yet discover it somewhat obscure and see the entire thing as a huge riddle. It is safe to say that you are looking to see how Big Data affect huge and little business and individuals like any of us? Do you feel numerous individuals discuss Big Data and Hadoop, and even don’t have the foggiest idea about the rudiments like history of Hadoop, significant players and sellers of Hadoop. At that point this blog is only for you. Prerequisites : CS6302Database Management Systems, IT6702Data Warehousing and Data Mining .

IT6006 Data Analytics L T P C 3 0 0 3

OBJECTIVES:

The Student should be made to:

- Be exposed to big data
- Learn the different ways of Data Analysis
- Be familiar with data streams
- Learn the mining and clustering
- Be familiar with the visualization

UNIT I INTRODUCTION TO BIG DATA 8

Introduction to Big Data Platform – Challenges of conventional systems – Web data – Evolution of Analytic scalability, analytic processes and tools, Analysis vs reporting – Modern data analytic tools, Stastical concepts: Sampling distributions, resampling, statistical inference, prediction error.

UNIT II DATA ANALYSIS 12

Regression modeling, Multivariate analysis, Bayesian modeling, inference and Bayesian networks, Support vector and kernel methods, Analysis of time series: linear systems analysis, nonlinear dynamics – Rule induction – Neural networks: learning and generalization, competitive learning, principal component analysis and neural networks; Fuzzy logic: extracting fuzzy models from data, fuzzy decision trees, Stochastic search methods.

UNIT III MINING DATA STREAMS 8

Introduction to Streams Concepts – Stream data model and architecture – Stream Computing, Sampling data in a stream – Filtering streams – Counting distinct elements in a stream – Estimating moments – Counting oneness in a window – Decaying window – Realtime Analytics Platform(RTAP) applications – case studies – real time sentiment analysis, stock market predictions.

UNIT IV FREQUENT ITEMSETS AND CLUSTERING 9

Mining Frequent itemsets – Market based model – Apriori Algorithm – Handling large data sets in Main memory – Limited Pass algorithm – Counting frequent itemsets in a stream – Clustering Techniques – Hierarchical – K- Means – Clustering high dimensional data – CLIQUE and PROCLUS – Frequent pattern based clustering methods – Clustering in non-euclidean space – Clustering for streams and Parallelism.

UNIT V FRAMEWORKS AND VISUALIZATION 8

MapReduce – Hadoop, Hive, MapR – Sharding – NoSQL Databases – S3 – Hadoop Distributed file systems – Visualizations – Visual data analysis techniques, interaction techniques; Systems and applications:

TOTAL: 45 PERIODS

TEXT BOOKS:

1. Michael Berthold, David J. Hand, Intelligent Data Analysis, Springer, 2007.
2. Anand Rajaraman and Jeffrey David Ullman, Mining of Massive Datasets, Cambridge University Press, 2012.

CONTENTS

Unit – I

Introduction

Part – A

1.Differentiate Big Data and Conventional Data [CO2-L2]

Big Data	Normal or Conventional Data
Huge data sets.	Data set size in control.
Unstructured data such as text, video, and audio.	Normally structured data such as numbers and categories, but it can take other forms as well.
Hard-to-perform queries and analysis.	Relatively easy-to-perform queries and analysis.
Needs a new methodology for analysis.	Data analysis can be achieved by using conventional methods.

2.Justify "Accuracy in big data is beneficial" [CO2-L2]

Accuracy in big data may lead to more confident decision making, and better decisions can result in greater operational efficiency, cost reduction and reduced risk.

When the dataset is free of noises and other missing values then the process performed on the data produces more relevant analysis and in turn provides better operational efficiency, so accuracy is highly beneficial

3.Show why Big data is considered as a Moving Target. [CO2-L1]

Big Data in simple terms can be said as a huge volume of data and reading the huge volume of data requires more disk space. Because of this reason it is tough to reread any data for a different process.

In addition to this Big Data is ever changing and can not be stored permanently, this property is called versatility. Because of these features the Big Data is considered a Moving Target.

4.What are the various dimensions of growth of Big Data? [CO2-L1]

The three dimensions of growth of big data are as follows

- **Volume** – amount of data gets added exponentially
- **Variety** – Various new categories of data are collected
- **Velocity** – The rate of collecting new data or updates

-

5.Is cost reduction relevant to big data analytics? Justify [CO2-L2]

Big data analytics is considered a huge cost cutter by providing predictive analysis. Predictive analytics uses many techniques from data mining, statistics, modeling, machine learning, and artificial intelligence to analyze current data to make predictions about future. Thus helping companies in understanding the market situation in the future.

6.Can Big Data Analytics be used as a Decision Making tool? Justify [CO2-L2]

Big Data Analytics can provide insight on very complex scenarios by simplifying the uncertainty to the maximum possible extent, if performed properly by taking all attributes into consideration then Big Data Analytics can support in decision making process.

7.Justify “Web Data is the most popular Big Data” [CO2-L2]

Web Data includes all the data collected across the internet specific to a corporate or to an individual, since the Internet has become more and more prevalent and the its versatile nature has made web data to be more useful to perform predictive analysis, because of these reasons the Web Data is considered the most popular big data.

8.How is semi-structured data different from Unstructured and Structured Data? [CO2-L2]

Semi-structured data is a form of structured data that does not conform to the formal structure of data models associated with relational databases or other forms of data tables, but however contains tags or other markers to separate semantic elements and enforce hierarchies of records and fields within the data. Therefore, it is also known as ***self-describing structure***.

9.What are the Advantages of using Semi-Structured Data format? [CO2-L1]

- Programmers continuing objects from their application to a database do not need to worry about object-relational impedance mismatch, but can often serialize objects via a light-weight library.
- Support for nested or hierarchical data often simplifies data models representing complex relationships between entities.
- Support for lists of objects simplifies data models by avoiding messy translations of lists into a relational data model.

10.What are the Disadvantages of using Semi-Structured Data format? [CO2-L1]

- The traditional relational data model has a popular and ready-made query language, SQL.
- Prone to ***"garbage in, garbage out"***; by removing restraints from the data model, there is less fore-thought that is necessary to operate a data application.
-

11.The challenge of big data analytics grows as the degree of granularity increases. Justify [CO2-L2]

Granularity is the level of depth represented by the data in a fact or dimension table in a data warehouse. High granularity means a minute, sometimes atomic grade of detail, often at the level of the transaction.

Therefore as more and more atomic attributes gets added to the analytic environment the time and computational power need grows drastically

12.Why domain expertise is required for any type of Data Analytics? [CO2-L1]

Domain expertise is important to properly interpret the output of the analysis with the help of suitable visualization tools. Making sure the people analyzing the data have a deep understanding of where the data comes from, what audience will be consuming the data and how that audience will interpret the information is important to improve the operational efficiency

13.What does the term Outliers mean in relation to statistics? [CO2-L1]

In statistics, an outlier is an observation point that is distant from other observations. An outlier may be due to variability in the measurement or it may indicate experimental error; the latter are sometimes excluded from the data set.

14.Why are outliers plotted as a separate chart? [CO2-L1]

Outliers typically represent about 1 to 5 percent of data, but when working with massive amounts of data, viewing 1 to 5 percent of the data becomes very difficult, so in order to read previously unseen and potentially valuable insight outliers are plotted as a separate chart.

15.What is a 360-degree view? [CO2-L1]

The 360 degree view of a particular item of interest is gaining knowledge about all the attributes and properties relevant to the item of interest. In practice 360 degree data is not easy to achieve and processing this data is much more difficult.

16.What are the typical behaviours that are captured from a customer? [CO2-L1]

Any action that a customer takes while interacting with an organization should be captured if it is possible to capture it. Some of them are,

- Purchases
- Requesting help
- Product views
- Forwarding a link
- Shopping basket additions
- Posting a comment
-

17.Define "Faceless Customer" [CO2-L1]

An arbitrary identification number that is not personally identifiable can be matched to each unique customer based on a logon, cookie, or similar piece of information. This creates what might be called a "faceless" customer record. It is used to provide privacy to the customer.

18.What is the use of analysing the customer's path? [CO2-L1]

The most important aspect of understanding the path taken by a customer inside a web site is to know which feature of the web site is more appealing to the user and this can be used to provide dynamic web pages based on the user`s choice.

19.What is an attrition model? [CO2-L1]

Attrition model is a type of predictive analysis that is used to indicate some kind of downfall or decrease in sales or loss of customer, companies may use attrition model to initiate proactive measures.

20.What is the reason for using logistic regression to create response models? [CO2-L1]

Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, because it could use both dependant and independent variables it is commonly used to create response models.

21.What would happen if the analysis is not scalable? [CO2-L1]

If the analysis is not scalable the result could not be used on the data collected in the future, since all the data collected are analysed, the size of the dataset grows constantly and is scalability is lost then the process of collecting data has no meaning

22.Analyse the similarity between the data mining and data analysis. [CO2-L2]

The following are the similarities between Data Mining and Data Analysis.

Data analyst create small chunks of data for analysis called datasets and the same is done in a data warehouse and this collection is data marts.
The Data Preparation steps are also similar, known as “Extract, Transform and Load”

23.What does In-Database Analysis mean? [CO2-L1]

In an enterprise data warehousing environment, most of the data sources have already been brought into one place. Move the analysis to the data instead of moving the data to the analysis. This is the concept of IN-DATABASE ANALYTICS.

24.What is the advantage of using a Massive Parallel Processing system? [CO2-L1]

Instead of using a single overloaded database server, an MPP database breaks the data into independent chunks with independent disk and CPU, this reduces the load of the system and increases the reliability of the processing.

25.What is called Scoring? [CO2-L1]

After a model has been created based on historical data, it can then be applied to new data in order to make predictions about unseen behavior.
The process of using a model to make predictions about behavior that has yet to happen is called "scoring." The output of the model (the prediction) is called a score.

26.Differentiate Private and Public Cloud [CO2-L1]

Private Cloud	Public Cloud
Virtualized resource is privately shared (Company owned Infrastructure)	Virtualized resource is publicly shared (3rd party owned Infrastructure)
Dedicated to a single purpose or single client	Shared by multiple clients or multipurpose
Suitable for highly confidential data processing	Suitable for cost effective data processing

27.Justify, “Grid computing is suitable to run large number of small to medium size tasks” [CO2-L1]

In Grid Computing instead of having one server managing its CPU and resources across jobs, jobs are parcelled out individually to the different machines to be processed in parallel. Each machine may only be able to handle a fraction of the work of the original server and can potentially handle only one job at a time and therefore grid computing is more suitable to run large number of small to medium size tasks.

28.What is MapReduce? [CO2-L1]

MapReduce is a parallel programming framework. It’s neither a database nor a direct competitor to databases.

MapReduce consists of two primary processes that a programmer builds: the “map” step and the “reduce” step.

29.What is an Analytic Sandbox? [CO2-L1]

An analytic sandbox provides a set of resources with which in-depth analysis can be done to answer critical business questions. An analytic sandbox is ideal for data exploration, development of analytical processes, proof of concepts, and prototyping.

30.Why is hybrid sandbox most preferred by data analysts? [CO2-L1]

The strengths of a hybrid sandbox environment are similar to the strengths of the internal and external options, plus having ultimate flexibility in the approach taken for an analysis. It is easy to avoid production impacts during early testing if work is done on the external sandbox.

31.Differentiate Development ADS and Production ADS [CO2-L1]

A development ADS is going to be the data set used to build an analytic process. It will have all the candidate variables that may be needed to solve a problem and will be very wide. A development ADS might have hundreds or even thousands of variables or metrics within it.

A production analytic data set, however, is what is needed for scoring and deployment. It’s going to contain only the specific metrics that were actually in the final solution.

32.Define Baye`s Rule. [CO2-L1]

Bayes Rule describes the probability of an event, based on conditions that might be related to the event.

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

33.Why Artificial Neural Networks are not commonly used for Data Mining tasks? [CO2-L1]

Neural networks have been successfully applied in a wide range of supervised and unsupervised learning applications. Neural network methods are not commonly used for data mining tasks, however, because they often produce incomprehensible models and require long training times.

34.What are the general ideas of hypothesis testing? [CO2-L1]

The general idea of hypothesis testing involves:

1.Making an initial assumption.

2.Collecting evidence (data).

3.Based on the available evidence (data), deciding whether to reject or not reject the initial assumption.

35.Define the term confidence interval in the context of Statistics. [CO2-L1]

In statistics, a confidence interval (CI) is a type of interval estimate of a population parameter. It is an observed interval (i.e., it is calculated from the observations), that includes the value of an unobservable parameter of interest if the experiment is repeated. How frequently the observed interval contains the parameter is determined by the ***confidence level*** or ***confidence coefficient***.\

Part – B

Introduction to Big Data Platform

1.Discuss in detail about the importance of the Big Data Platform in Modern Data Analytics [CO2-L1]

Big Data Platform

Big data is a term for data sets that are so large or complex that traditional data processing applications are inadequate. Challenges include

- Analysis,
- Capture,
- Data Curation,
- Search,
- Sharing,
- Storage,
- Transfer,
- Visualization,
- Querying,
- Updating

•Information Privacy.

The term often refers simply to the use of predictive analytics or certain other ***advanced methods*** to extract value from data, and seldom to a particular size of data set.

Accuracy in big data may lead to more confident decision making, and better decisions can result in greater operational efficiency, cost reduction and reduced risk.

Big data usually includes data sets with sizes beyond the ability of commonly used software tools to capture, curate, manage, and process data within a tolerable elapsed time. Big data "size" is a constantly moving target.

Big data requires a set of techniques and technologies with new forms of integration to reveal insights from datasets that are diverse, complex, and of a massive scale

Characteristics

Big data can be described by the following characteristics:

Volume

The quantity of generated and stored data. The size of the data determines the value and potential insight- and whether it can actually be considered big data or not.

Variety

The type and nature of the data. This helps people who **ANALYZE** it to effectively use the resulting insight.

Velocity

In this context, the speed at which the data is generated and processed to meet the demands and challenges that lie in the path of growth and development.

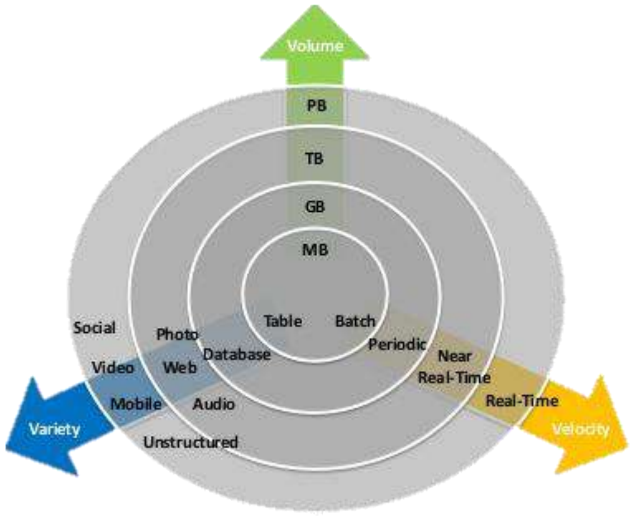
Variability

Inconsistency of the data set can hamper processes to handle and manage it.

Veracity

The quality of captured data can vary greatly, affecting accurate analysis.

Growth of Big Data



2.Compare Big Data with Conventional Data and indicate some of the importance of Big Data Analysis [CO2-L2]

Importance of Big Data Analytics

Big data analytics helps organizations harness their data and use it to identify new opportunities. That, in turn, leads to smarter business moves, more efficient operations, higher profits and happier customers. In his report Big Data in Big Companies, IIA Director of Research Tom Davenport interviewed more than 50 businesses to understand how they used big data. He found they got value in the following ways:

Cost reduction.

Big data technologies such as Hadoop and cloud-based analytics bring significant cost advantages when it comes to storing large amounts of data – plus they can identify more efficient ways of doing business.

Faster, better decision making.

With the speed of Hadoop and in-memory analytics, combined with the ability to analyze new sources of data, businesses are able to analyze information immediately – and make decisions based on what they've learned.

New products and services.

With the ability to gauge customer needs and satisfaction through analytics comes the power to give customers what they want. Davenport points out that with big data analytics, more companies are creating new products to meet customers' needs.

Big Data Analytics Vs. Conventional Data Analytics

Conventional analytical tools and techniques are inadequate to handle data that is unstructured (like text data), that is too large in size, or that is growing rapidly like social media data.

A cluster analysis on a 200MB file with 1 million customer records is manageable, but the same cluster analysis on 1000GB of Facebook customer profile information will take a considerable amount of time if conventional tools and techniques are used.

Facebook as well as entities like Google and Walmart generate data in petabytes every day. Distributed computing methodologies might need to be used in order to carry out such analysis.

Big Data	**Normal or Conventional Data**
Huge data sets.	Data set size in control.
Unstructured data such as text, video, and audio.	Normally structured data such as numbers and categories, but it can take other forms as well.
Hard-to-perform queries and analysis.	Relatively easy-to-perform queries and analysis.
Needs a new methodology for analysis.	Data analysis can be achieved by using conventional methods.
Need tools such as Hadoop, Hive, Hbase, Pig, Sqoop, and so on.	Tools such as SQL, SAS, R, and Excel alone may be sufficient.
Raw transactional data.	The aggregated or sampled or filtered data.
Used for reporting, basic analysis, and text mining. Advanced analytics is only in a starting stage in big data.	Used for reporting, advanced analysis, and predictive modeling .
Big data analysis needs both programming skills (such as Java) and analytical skills to perform analysis.	Analytical skills are sufficient for conventional data; advanced analysis tools don't require expert programing skills.

Petabytes/exabytes of data. Millions/billions of accounts. Billions/trillions of transactions.	Megabytes/gigabytes of data. Thousands/millions of accounts. Millions of transactions.
Generated by big financial institutions, Facebook, Google, Amazon, eBay, Walmart, and so on.	Generated by small enterprises and small banks.

3.List out and explain some of the challenges of Big Data [CO2-L1]

Challenges of Big Data

The following are the five most important challenges of the Big Data

Meeting the need for speed

In today’s hypercompetitive business environment, companies not only have to find and analyze the relevant data they need, they must find it quickly.

Visualization helps organizations perform analyses and make decisions much more rapidly, but the challenge is going through the sheer volumes of data and accessing the level of detail needed, all at a high speed.

The challenge only ***grows as the degree of granularity increases***. One possible solution is hardware. Some vendors are using increased memory and powerful parallel processing to crunch large volumes of data extremely quickly. Another method is putting data in-memory but using a grid computing approach, where many machines are used to solve a problem. Both approaches allow organizations to explore huge data volumes and gain business insights in near-real time.

Understanding the data

It takes a lot of understanding to get data in the **RIGHT SHAPE** so that you can use visualization as part of data analysis.

For example, if the data comes from social media content, you need to know who the user is in a general sense – such as a customer using a particular set of products – and understand what it is you’re trying to visualize out of the data. Without some sort of context, visualization tools are likely to be of less value to the user.

One solution to this challenge is to have the ***proper domain expertise*** in place. Make sure the people analyzing the data have a deep understanding of where the data comes from, what audience will be consuming the data and how that audience will interpret the information.

Addressing data quality

Even if you can find and analyze data quickly and put it in the proper context for the audience that will be consuming the information, the value of data for DECISION-MAKING PURPOSES will be jeopardized if the data is not accurate or timely.

This is a challenge with any data analysis, but when considering the volumes of information involved in big data projects, it becomes even more pronounced. Again, data visualization will only prove to be a ***valuable tool*** if the data quality is assured.

To address this issue, companies need to have a data governance or information management process in place to ensure the data is clean. It’s always best to have a ***pro-active method*** to address data quality issues so problems won’t arise later.

Displaying meaningful results

Plotting points on a graph for analysis becomes difficult when dealing with extremely large amounts of information or a variety of categories of information.

For example, imagine you have 10 billion rows of retail SKU data that you’re trying to compare. The user trying to view 10 billion plots on the screen will have a hard time seeing so many data points. One way to resolve this is to cluster data into a higher-level view where smaller groups of data become visible. By grouping the data together, or “binning,” you can more effectively visualize the data.

Dealing with outliers

The graphical representations of data made possible by visualization can communicate trends and outliers much faster than tables containing numbers and text.

Users can easily spot issues that need attention simply by glancing at a chart. Outliers typically represent about 1 to 5 percent of data, but when you’re working with massive amounts of data, viewing 1 to 5 percent of the data is rather difficult. How do you represent those points without getting into plotting issues? Possible solutions are to remove the outliers from the data (and therefore from the chart) or to create a separate chart for the outliers.

We can also bin the results to both view the distribution of data and see the outliers. While outliers may not be representative of the data, they may also reveal previously unseen and potentially valuable insights.

As more and more businesses are discovering, data visualization is be-coming an increasingly important component of analytics in the age of big data. The availability of new in-memory technology and high-performance analytics that use data visualization is providing a better way to analyze data more quickly than ever. Visual analytics enables organizations to take raw data and present it in a meaningful way that generates the most value. However, when used with big data, visualization is bound to lead to some challenges.

Web Data

4.Justify the Statement “Web Data is the Most Popular Big Data” with reference to data analytic professional [CO2-L1]

Web Data

Web data is one of the most popular type of big data, no other big data source is as widely used today as web data. Organizations across a number of industries have integrated detailed, customer-level behavioral data sourced from a web site into their enterprise analytics environments. Integrating detailed clickstream data with other data instead to keeping it isolated by itself is also a part of web data.

Web Data Overview

Organizations have talked about a 360-degree view of their customers for years. what is really meant is that the organization has as full a view of its customers as possible considering the technology and data available at that point in time.

Cutting-edge companies started to have basic recency, frequency, and monetary value (RFM) metrics attached to customers. Such metrics look at when a customer last purchased (recency), how often they have purchased (frequency), and how much they spent (monetary value).

These RFM summaries might be tallied for the past year and possibly over a customer’s lifetime.

Today, organizations collect newly evolving big data sources related to their customers from a variety of extended and newly emerging touch points such as web browsers, mobile applications, kiosks, social media sites, and more.

With today’s data storage and processing capabilities, it is absolutely possible to achieve success, and many forward-thinking companies have already proven it.

Missed Data

For a web site, 95 percent of browsing sessions do not result in a basket being created. Of that 5 percent, only about half, or 2.5 percent, actually begin the check-out process. And, of that 2.5 percent only two-thirds, or 1.7 percent, actually complete a purchase.

This means is that information is missing on more than 98 percent of web sessions if only transactions are tracked. Traditional web analytics focus on aggregated behavior, summarized in an environment where only web data was included.

The goal needs to be moving beyond reporting of summary statistics, even if they can be viewed in some detail, to actually combining customer level web behavior data with other cross-channel customer data.

Possibilities of Improvement

Knowing everything customers do as they go through the process of doing business with your organization. Not just what they buy, but what they are thinking about buying along with what key decision criteria they use. Such knowledge enables a new level of understanding about your customers and a new level of interaction with your customers. It allows you to meet their needs more quickly and keep them satisfied.

New Source of Information

This big data source isn’t a simple extension of existing data sources. In the case of detailed web behaviour, there is no existing analog to most of the data that can be collected. It is a fundamentally new source of information.

One of the most exciting aspects of web data is that it provides factual information on customer preferences, future intentions, and motivations that are virtually impossible to get from other sources outside of a direct conversation or survey.

Once customers’ intentions, preferences, and motivations are known, there are completely new ways of communicating with them, driving further business, and increasing their loyalty.

What Data should be Collected?

Any action that a customer takes while interacting with an organization should be captured if it is possible to capture it. That means detailed event history from any customer touch point. Common touch points today include web sites, kiosks, mobile apps, and social media.

Behaviours That Can Be Captured

- Purchases
- Requesting help
- Product views
- Forwarding a link
- Shopping basket additions
- Posting a comment
- Watching a video
- Registering for a webinar
- Accessing a download
- Executing a search
- Reading / writing a review

Privacy

An arbitrary identification number that is not personally identifiable can be matched to each unique customer based on a logon, cookie, or similar piece of information. This creates what might be called a “faceless” customer record.

While all of the data associated with one of these identifiers is from one person, the people doing the analysis have no ability to tie the ID back to the actual customer.

With today’s database technologies, it is possible to enable analytic professionals to do analysis without having any ability to identify the individuals involved. This can remove many privacy concerns.

5.Relate Web Data with modern data collection and processing from the point of view of a data analyst [CO2-L2]

Web Data – Area of Interest

There are a number of specific areas where web data can help organizations understand their customers better than is possible without web data.

Shopping Behaviors

A good starting point to understanding shopping behavior is identifying how customers come to a site to begin shopping.

- What search engine do they use?
- What specific search terms are entered?
- Do they use a bookmark they created previously?

Analytic professionals can take this information and look for patterns in terms of which search terms, search engines, and referring sites are associated with higher sales rates.

Note that analysts will be able to look into higher sales rates not just within a given web session, but also for the same customer over time.

One very interesting capability enabled by web data is to identify product bundles that are of interest to a customer before they make a purchase.

Customer Purchase Paths and Preferences

Once the aspects of a site that appeal to customers on an individual basis are known, they can be targeted with messages that meet their needs much more effectively.

Research Behaviours

Once customers' patterns are known, it is possible to alter what they see when they visit a site in order to make it easier for them to find their favourite options quickly.

Another way to use web data to understand customers' research patterns is to identify which of the pieces of information offered on a site are valued by the customer base overall and the best customers specifically.

Web site feature the organization was considering removing is a big favourite among a critical segment of customers. In that case, the feature might be kept.

Identifying which site features are important to each customer and how each customer leverages the site for research can help better tailor a site to the individual. For customers who always drill to detailed product specifications, perhaps those specifications come up as soon as a product is viewed. For those who always want to see photos, perhaps photos are featured in full size instead of as thumbnails.

Feedback Behaviours

The best information customers can provide is detailed feedback on products and services. The fact that customers are willing to take the time to do so indicates that they are engaged with a brand. Text mining to understand the tone, intent, and topic of a customer’s feedback

Web Data In Action

It is possible that the information missing paints a totally different picture than expected.

It is possible to make suboptimal, if not totally wrong, decisions.

Organizations should strive to collect and analyse as much data as possible.

How organizations can apply web data to enhance existing analytics, enable new analytics, and improve their business.

The Next Best Offer

A very common marketing analysis is to predict what the next best offer is for each customer. The web provides direct clues as to what is of interest to customers and if they are still engaged. Consider the case of a catalog retailer that also has many store locations. The cataloger collects the following for each customer, among other data:

- Last products browsed
- Last products reviewed
- Historical purchases
- Marketing campaign and response history

The effort leads to major changes in the promotional efforts versus the cataloger’s traditional approach, providing the following results:

- A decrease in total mailings
- A reduction in total catalog promotions pages
- A materially significant increase in total revenues

Web data can help completely overhaul activities for the better.

Attrition Modeling

In the telecommunications industry, companies have invested massive amounts of time and effort to create, enhance, and perfect “churn” models. Churn models flag those customers most at risk of cancelling their accounts so that action can be taken proactively to prevent them from doing so. Churn is a major issue for the industry and there are huge amounts of money at stake. The models have a major impact on the bottom line.

Response Modeling

Many models are created to help predict the choice a customer will make when presented with a request for action. Models typically try to predict which customers will make a purchase, or accept an offer, or click on an e-mail link. For such models, a technique called logistic regression is often used.

The main difference is that in an attrition model, the goal is predicting a negative behavior (churn) rather than a positive behavior (purchase or response).

In theory, every customer has a unique score. In practice, since only a small number of variables define most models, many customers end up with identical or nearly identical scores. This is particularly true among customers who are not very frequent or high-spending. In such cases, many customers can end up in big groups with very similar, very low scores.

Web data can help greatly increase differentiation among customers. This is especially true among low-value or infrequent customers where customers can have a large uplift in score based on the web data.

Customer Segmentation

Web data also enables a variety of completely new analytics. One of those is to segment customers based solely upon their typical browsing patterns. Such segmentation will provide a completely different view of customers than traditional demographic or sales-based segmentation schemas.

Assessing Advertising Results

Traditional web analytics provide high-level summaries such as total clicks, number of searches, cost per click or impression, keywords leading to the most clicks, and page position statistics. However, these metrics are at an aggregate level and are rolled up only from the individual browsing session level. The context is also traditionally limited solely to the web channel. Once a customer leaves the web site and his web session ends, the scope of the analysis is complete.

Evolution of Analytic scalability

6.Illustrate the evolution of Analytic Scalability from conventional Database Management System [CO2-L1]

Evolution of Analytic Scalability

The world of big data requires new levels of scalability. As the amount of data organizations process continues to increase, the same old methods for handling data just won’t work anymore. Organizations that don’t update their technologies to provide a higher level of scalability will quite simply choke on big data.

The convergence of the analytic and data environments, massively parallel processing (MPP) architectures, the cloud, grid computing, and MapReduce.

A History Of Scalability

Until well into the 1900s, doing analytics was very, very difficult. To do a deep analysis, such as a predictive model, it required manually computing all of the statistics.

Scalability of any sort was virtually nonexistent.

The amount of data has grown at least as fast as the computing power of the machines that process it.

As new big data sources become available the boundaries are being pushed further.

The Convergence Of The Analytic And Data Environments

It used to be that analytic professionals had to pull all their data together into a separate analytics environment to do analysis.

None of the data that was required was together in one place, and the tools that did analysis didn’t have a way to run where the data sat.

The only option was to pull the data together in a separate analytics environment and then start performing analysis. Much of the work analytic professionals do falls into the realm of advanced analytics, which encompasses data mining, predictive modeling, and other advanced techniques.

There is an interesting parallel between what analysts did in the early days and what data warehousing is all about.

There’s not much difference between a data set as analysts define it and a “table” in a database.

Analysts have done “merges” of their data sets for years. That is the exact same thing as a “join” of tables in a database.

In both a merge and a join, two or more data sets or tables are combined together.

Analysts do what is called “data preparation.” In the data warehousing world this process is called “extract, transform, and load (ETL).”

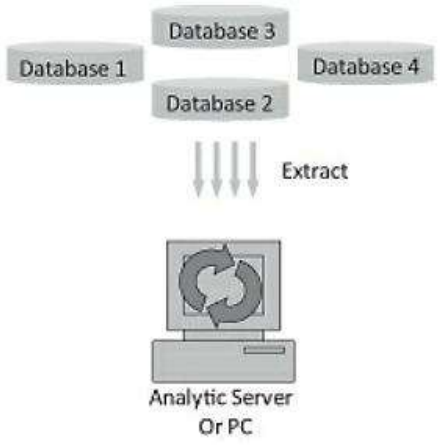
Basically, analysts were building custom data marts and mini–data warehouses before the terms data mart or data warehouse were invented.

The Relational Database Management System (RDBMS) started to not only become popular, but to become more scalable and more widely adopted.

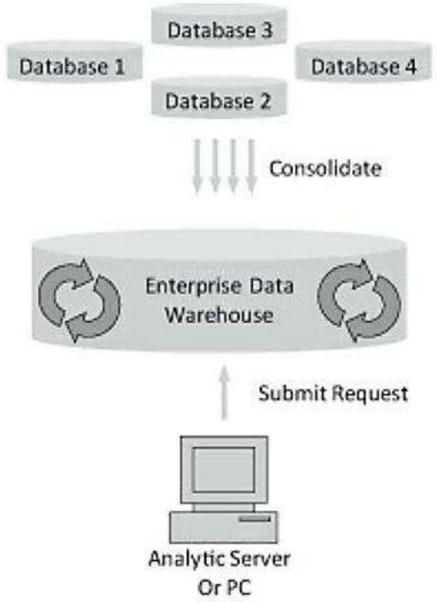
Initially, databases were built for each specific purpose or team, and relational databases were spread all over an organization. Such single purpose databases are often called “data marts.”

Combining the various database systems into one big system called an Enterprise Data Warehouse (EDW).

Traditional Analytic Architecture



Modern In-Database Architecture



In an enterprise data warehousing environment, most of the data sources have already been brought into one place. Move the analysis to the data instead of moving the data to the analysis. This is the concept of IN-DATABASE ANALYTICS.

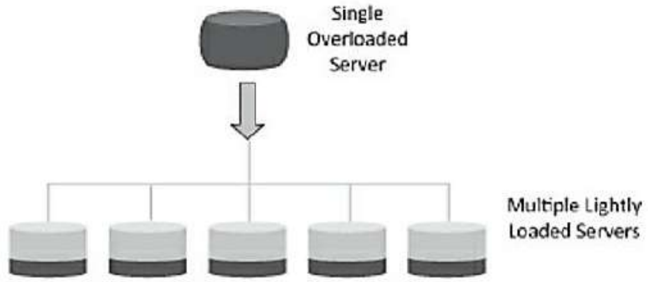
7.Comment on the statement “Massive Parallel Processing Systems are more faster in processing big data” also show how Data Scoring is performed [U}

Massively Parallel Processing Systems

Individual vendor architectures may vary, MPP is the most mature, proven, and widely deployed mechanism for storing and analyzing large amounts of data.

An MPP database spreads data out into independent pieces managed by independent storage and central processing unit (CPU) resources.

Massively Parallel Processing System Data Storage

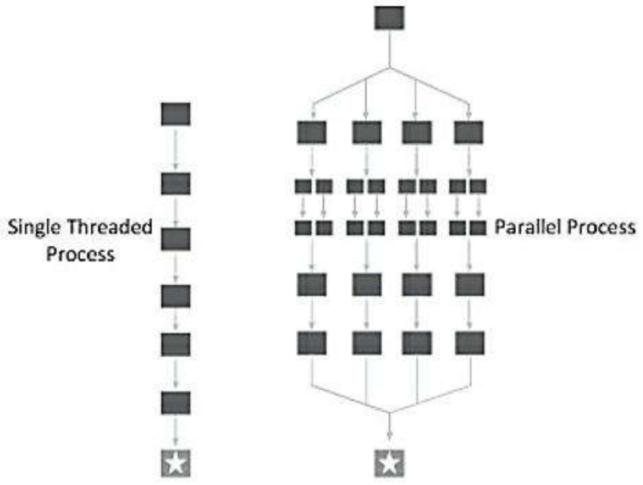


Instead of a single overloaded database, an MPP database breaks the data into independent chunks with independent disk and CPU.

In non-MPP database architectures, there are at least a few points during processing where the number of lanes is reduced to one, if not for the entire drive. This allows much faster query execution, since many independent smaller queries are running simultaneously instead of just one big query.

It gets a little more complicated in cases where data must be moved from one processing unit to another processing unit as part of the requirements of the query, but MPP systems are built to handle that in a very, very fast way as well.

Traditional Query versus an MPP Query



Instead of a single threaded process to work through the data, an MPP system breaks the job into pieces and allows the different sets of CPU and disk to run the process concurrently.

MPP systems build in redundancy so that data is stored in more than one location to make recovery easy in cases where there’s equipment failure.

They also have resource management tools to manage the CPU and disk space, query optimizers to make sure queries are being optimally executed, and other items that make it easier and more efficient for users to leverage the systems.

Using MPP Systems for Data Preparation and Scoring

The reason MPP can be a huge benefit to advanced analytics is because most of the processing pain in advanced analytics comes during the data preparation stages. Data preparation is made up of joins, aggregations, derivations, and transformations. This is the process of combining various data sources to pull together all the information needed for an analysis. Aggregations involve combining information from multiple records into one.

The logic required for most data preparation tasks is actually relatively simple. Today's SQL can handle most, if not all, data preparation tasks required for the majority of analysis. The use of SQL is where in-database processing started on MPP systems. Analysts simply pushed tasks into the database by writing SQL instead of writing in an analytics language that requires pulling the data off the database.

Before SQL was upgraded with more robust functionality, it led to the continued need to pull data off in order to do necessary analytical processing.

As SQL has greatly evolved most of the operations involved in data preparation can now be done via SQL right in the database. A lot of analytic professionals still code in SQL instead of an analytic tool today

As the concept of in-database processing has evolved, analytical tool vendors have started to enable their applications to push processing directly into the database system. Code is written in the native language of such tools, but the software now recognizes when an MPP database engine is being accessed, and the software will pass through instructions to the database and let the database handle the heavy lifting instead of performing a large extract.

There are at least four primary ways for data preparation and scoring to be pushed into a database today:

1.SQL Push Down,
2.User-Defined Functions (UDFS),
3.Embedded Processes,
4.Predictive Modelling Markup Language (PMML) Scoring.

SQL Push Down

SQL is the native language of an MPP system and it’s efficient for a wide range of requirements.

Many core data preparation tasks can be either translated into SQL by the user, or an analytic tool can generate SQL on the user’s behalf and “push it down” to the database.

SQL is also easy to generate for many common analytical algorithms that have fairly simple scoring logic. Linear regression, logistic regression, and decision trees are examples

Analytic tools will often translate the logic from a model into SQL for the user. Or, sometimes, users will code an SQL script themselves once the model is complete.

The data preparation or scoring processes end up being executed purely with SQL.

User-Defined Functions

User-defined functions (UDFs) are a relatively new feature of relational databases. The capabilities of a UDF go beyond those of SQL. It extend SQL functionality by allowing a user to define logic that can be executed in the same manner as a native SQL function.

The UDF will apply whatever logic is required to the data, and this logic can be more complex than is possible in pure SQL.

User-defined functions are coded in languages like C++ or Java. Some procedural language capabilities can be embedded within them.

The downside of UDFs is that a lot of analytic professionals don’t know how to program in the languages required. That’s where analytic tools that automatically generate the functions are a very good thing. Such tools take care of generating appropriate user-defined functions for the analyst and loading it into the database so it is ready to use.

Embedded Processes

A very new option for pushing processing into a database is the concept of an embedded process. An embedded process is an even higher level of integration than the user-defined functions. A user-defined function involves compiling code into new database functions that can be called from an SQL query the analytical tool has had to translate what it needs into the language of the database.

An embedded process, however, is a version of an analytic tool’s engine actually running on the database itself. An embedded process is therefore capable of running programs directly inside the database. Actual analytic tool code that has been published to the database engine.

There is no translation required. This method requires the least amount of change to underlying code but is also a much harder solution for tools to implement.

Predictive Modeling Markup Language

Predictive modeling markup language (PMML) is a way to pass model results from one tool to another.

It contains the minimum information needed to produce an accurate piece of scoring code. The type of information included in a PMML feed includes a model type, the variable names and formats, and the parameter values.

One drawback of PMML isn't immediately apparent. The exact same variables in the exact same format must be available in the system where the PMML is being deployed

8.Illustrate how Cloud Computing and Big Data Analytics work together and what are the resources provided by the Cloud Architecture [CO2-L1]

Cloud Computing

Three criteria for a cloud environment are,

1.Enterprises incur no infrastructure or capital costs, only operational costs. Those operational costs will be incurred on a pay per-use basis with no contractual obligations.
2.Capacity can be scaled up or down dynamically, and immediately. This differentiates clouds from traditional hosting service providers where there may have been limits placed on scaling.
3.The underlying hardware can be anywhere geographically. The architectural specifics are abstracted from the user. In addition, the hardware will run in multi-tenancy mode where multiple users from multiple organizations can be accessing the exact same infrastructure simultaneously.

Five essential characteristics of a cloud environment.

1.On-demand self-service
2.Broad network access
3.Resource pooling
4.Rapid elasticity

5.Measured service

The two primary types of cloud environments:

1.Public clouds
2.Private clouds.

Public Clouds

With a public cloud users are basically loading their data onto a host system and they are then allocated resources as they need them to use that data. They will get charged according to their usage.

Advantages of Public Cloud

1.The bandwidth is as-needed and users only pay for what they use.
2.It isn't necessary to buy a system sized to handle the maximum capacity ever required and then risk having half of the capacity sitting idle much of the time.
3.If there are short bursts where a lot of processing is needed then it is possible to get it with no hassle. Simply pay for the extra resources.
4.There's typically very fast ramp-up. Once granted access to the cloud environment, users load their data and start analyzing.
5.It is easy to share data with others regardless of their location since a public cloud by definition is outside of a corporate firewall. Anyone can be given permission to log on to the environment created.

Disadvantages of Public Cloud

1.Few performance guarantees
2.High variability in performance
3.Concerns around the security of the data
4.It can get expensive if a cloud isn't used wisely since users will be charged for everything that they do.
5.If an audited trail of data and where it sits is required, it is not possible to have that in a public cloud.

The best use for a public cloud is pure research and development work, where performance variability isn't something to worry about.

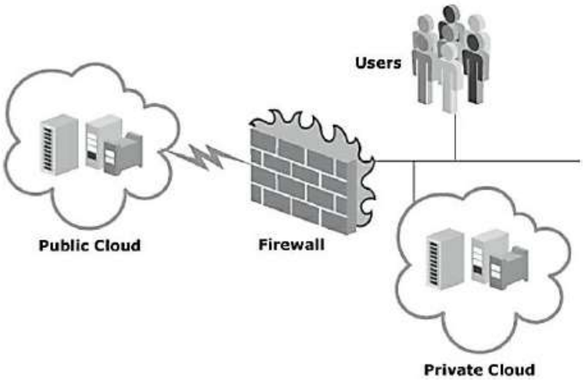
For non-mission-critical analytical processes, the cloud is a potential long-term host even for deployed processes.

A public cloud can be problematic if data security is a big concern. It's necessary to apply good security protocols and tools to a public cloud and keep your environment highly secure.

Private Clouds

A private cloud has the same features of a public cloud, but it's owned exclusively by one organization and typically housed behind a corporate firewall. A private cloud is going to serve the exact same function as a public cloud, but just for the people or teams within a given organization.

Private Cloud Vs. Public Cloud



One huge advantage of an onsite private cloud is that the organization will have complete control over the data and system security. Data is never leaving the corporate firewall so there's absolutely no concern about where it's going. The data is at no more risk than it is on any other internal system.

One downside of an onsite private cloud is that it is necessary to purchase and own the entire cloud infrastructure before allocating it out to users, which could in the short term negate some of the cost savings.

9.Discuss about the suitability of Grid Computing environment to process large tasks or processing more number of medium sized tasks [CO2-L2]

Grid Computing

There are some computations and algorithms that aren't cleanly converted to SQL or embedded in a user-defined function within a database. In these cases, it's necessary to pull data out into a more traditional analytics environment and run analytic tools against that data in the traditional way.

A grid configuration can help both cost and performance. It falls into the classification of “high-performance computing.” Instead of having a single high-end server (or maybe a few of them), a large number of lower-cost machines are put in place. As opposed to having one server managing its CPU and resources across jobs, jobs are parceled out individually to the different machines to be processed in parallel. Each machine may only be able to handle a fraction of the work of the original server and can potentially handle only one job at a time.

Using such a grid enables analytic professionals to scale an environment relatively cheaply and quickly.

If a large organization has many processes being run and most of them are small to medium in size, a grid can be a huge boost.

MapReduce

MapReduce is a parallel programming framework. It’s neither a database nor a direct competitor to databases.

MapReduce consists of two primary processes that a programmer builds: the “map” step and the “reduce” step.

These steps get passed to the MapReduce framework, which then runs the programs in parallel on a set of worker nodes.

In the case of MapReduce, there is a lot of commodity hardware to which data is being passed as needed to run a process. Each MapReduce worker runs the same code against its portion of the data.

The workers do not interact or even have knowledge of each other.

MapReduce is a programming framework popularized by Google and used to simplify data processing across massive data sets. Hadoop is a popular open-source version of MapReduce supplied by the Apache organization. Hadoop is the bestknown implementation of the MapReduce framework.

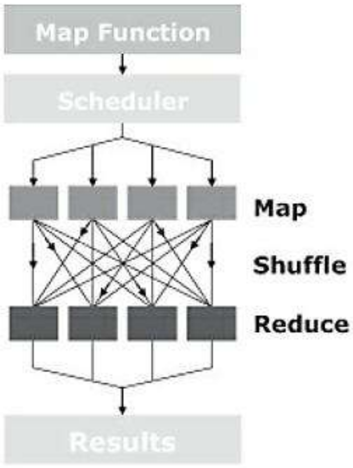
A big distinction of a MapReduce environment is the specific ability to handle unstructured text. In a relational database, everything is already in tables and rows and columns. The data already has well-defined relationships. This is not always true with raw data streams. That’s where MapReduce can really be powerful. Loading big chunks of text into a “blob” field in a database is possible, but it really isn’t the best use of the database or the best way to handle such data.

Working of MapReduce

Let’s assume there are 20 terabytes of data and 20 MapReduce server nodes for a project. The first step is to distribute a terabyte to each of the 20 nodes using a simple file copy process. Note that this data has to be distributed prior to the MapReduce process being started. Also note that the data is in a file of some format determined by the user. There is no standard format like in a relational database.

Next, the programmer submits two programs to the scheduler. One is a map program; the other is the reduce program. In this two-step processing, the map program finds the data on disk and executes the logic it contains. This occurs independently on each of the 20 servers in our example. The results of the map step are then passed to the reduce process to summarize and aggregate the final answers.

MapReduce Process



MapReduce Strengths and Weaknesses

MapReduce can run on commodity hardware. As a result, it can be very cheap to get up and running. It can also be very cheap to expand. It is easy to expand the capacity because all that is required is to buy more servers and bolt them on to the platform.

MapReduce is at its best when there is a large set of input data where much of the data isn’t required for analysis. If only a small piece of the data is really going to be important, but it isn’t clear up-front which pieces will be important, MapReduce can help.

MapReduce can be a terrific way to sort through the masses of data and pull out the important parts.

MapReduce is not a database, so it has no built-in security, no indexing, no query or process optimizer, no historical perspective in terms of other jobs that have been run, and no knowledge of other data that exists. While it provides the ultimate flexibility to process different kinds of data, it also comes with the responsibility to define exactly what the data is in every process created.

10.Analyse the role of Analytic Sandbox and its benefits in the Analytic Process and differentiate how the views of IT professional is different from Analytic Professional. [CO2-L2]

Analytic Processes

As analytic professionals begin constantly using a database platform for their work through a sandbox, they will be doing some tasks repeatedly.

Enterprise analytic data sets are key tools to help drive consistency and productivity, and lower risk into an organization's advanced analytics processes.

Enterprise analytic data sets are key tools to help drive consistency and productivity, and lower risk into an organization's advanced analytics processes.

Analytic Sandbox

One of the uses of such a database system is to facilitate the building and deployment of advanced analytic processes. In order for analytic professionals to utilize an enterprise data warehouse or data mart more effectively, however, they need the correct permissions and access to do so. An analytic sandbox is the mechanism for achieving this. If used appropriately, an analytic sandbox can be one of the primary drivers of value in the world of big data. Other terms used for the sandbox concept include an agile analytics cloud and a data lab, among others.

Definition and Scope

An analytic sandbox provides a set of resources with which in-depth analysis can be done to answer critical business questions. An analytic sandbox is ideal for data exploration, development of analytical processes, proof of concepts, and prototyping.

A sandbox is going to be leveraged by a fairly small set of users. There will be data created within the sandbox that is segregated from the production database. Sandbox users will also be allowed to load data of their own for brief time periods as part of a project, even if that data is not part of the official enterprise data model.

Data in a sandbox will have a limited shelf life. The idea isn't to build up a bunch of permanent data. During a project, build the data needed for the project. When that project is done, delete the data. If used appropriately, a sandbox has the capability to be a major driver of analytic value for an organization.

Analytic Sandbox Benefits

Benefits from the view of an analytic professional:

1)Independence.

Analytic professionals will be able to work independently on the database system without needing to continually go back and ask for permissions for specific projects.

2)Flexibility.

Analytic professionals will have the flexibility to use whatever business intelligence, statistical analysis, or visualization tools that they need to use.

3)Efficiency.

Analytic professionals will be able to leverage the existing enterprise data warehouse or data mart, without having to move or migrate data.

4)Freedom.

Analytic professionals can reduce focus on the administration of systems and babysitting of production processes by shifting those maintenance tasks to IT.

5)Speed.

Massive speed improvement will be realized with the move to parallel processing. This also enables rapid iteration and the ability to "fail fast" and take more risks to innovate.

Benefits from the view of IT:

1)Centralization.

IT will be able to centrally manage a sandbox environment just as every other database environment on the system is managed.

2)Streamlining.

A sandbox will greatly simplify the promotion of analytic processes into production since there will be a consistent platform for both development and deployment.

3)Simplicity.

There will be no more processes built during development that have to be totally rewritten to run in the production environment.

4)Control.

IT will be able to control the sandbox environment, balancing sandbox needs and the needs of other users. The production environment is safe from an experiment gone wrong in the sandbox.

5)Costs

Big cost savings can be realized by consolidating many analytic data marts into one central system.

11.Illustrate the need for different types of Analytic Sandboxes and what are their benefits? [CO2-L2]

Analytical Sandbox

An analytic sandbox provides a set of resources with which in-depth analysis can be done to answer critical business questions. An analytic sandbox is ideal for data exploration, development of analytical processes, proof of concepts, and prototyping.

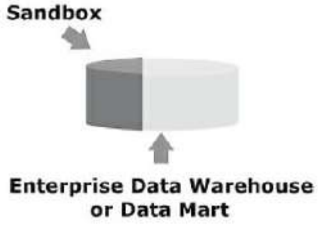
A sandbox is going to be leveraged by a fairly small set of users. There will be data created within the sandbox that is segregated from the production database. Sandbox users will also be allowed to load data of their own for brief time periods as part of a project, even if that data is not part of the official enterprise data model.

An Internal Sandbox

For an internal sandbox, a portion of an enterprise data warehouse or data mart is carved out to serve as the analytic sandbox.

If the sandbox is physically located on the production system. However, the sandbox database itself is not a part of the production database. The sandbox is a separate database container within the system.

An Internal Sandbox



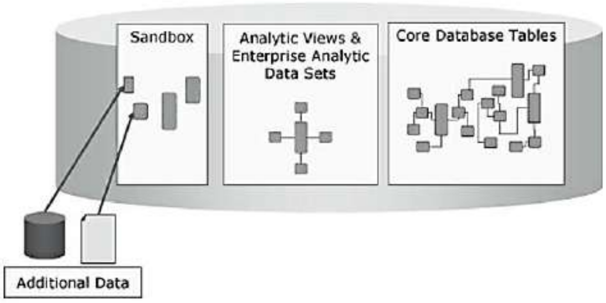
It is better to add a MapReduce environment into the mix. This would typically be installed alongside the database platform unless you're using a system that can combine the two environments together.

The MapReduce environment will require access to the internal sandbox. Data can be shared between the two environments as required.

One strength of an internal sandbox is that it will leverage existing hardware resources and infrastructure already in place. This makes it very easy to set up. From an administration perspective, there's no difference in setting up a sandbox than in setting up any other database container on the system.

The biggest strength of an internal sandbox is the ability to directly join production data with sandbox data. Since all of the production data and all of the sandbox data are within the production system, it's very easy to link those sources to one another and work with all the data together.

Detailed Internal Sandbox View



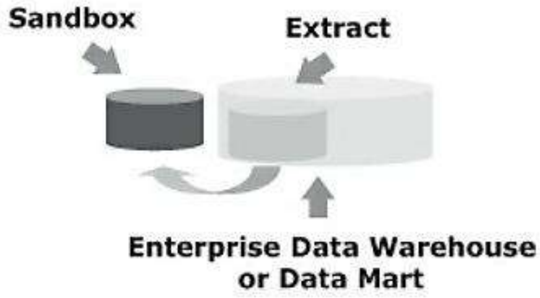
An internal sandbox is very cost-effective since no new hardware is needed. The production system is already in place. It is just being used in a new way. The elimination of any and all cross-platform data movement also lowers costs. The one exception is any data movement required between the database and the MapReduce environment.

There are a few weaknesses of an internal sandbox. One such weakness is that there will be an additional load on the existing enterprise data warehouse or data mart. The sandbox will use both space and CPU resources (potentially a lot of resources). Another weakness is that an internal sandbox can be constrained by production policies and procedures.

An External Sandbox

A physically separate analytic sandbox is created for testing and development of analytic processes. It’s relatively rare to have an environment that’s purely external. Internal or hybrid sandboxes, which we’ll talk about next, are more common. It is important to understand what the external sandbox is, however, as it is a component of a hybrid sandbox environment.

An External Sandbox



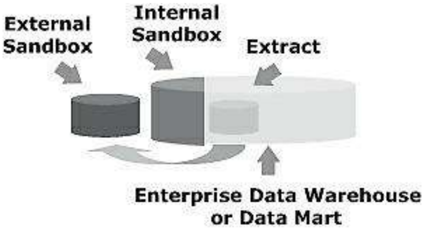
The biggest strength of an external sandbox is its simplicity. The sandbox is a stand-alone environment, dedicated to advanced analytics development. It will have no impact on other processes, which allows for flexibility in design and usage.

Another strength of an external sandbox is reduced workload management. When only analytic professionals are using the system, it isn’t necessary to worry much about tuning and balancing. There will be predictable, stable performance in both the sandbox and production environments.

A Hybrid Sandbox

A hybrid sandbox environment is the combination of an internal sandbox and an external sandbox. It allows analytic professionals the flexibility to use the power of the production system when needed, but also the flexibility of the external system for deep exploration or tasks that aren’t as friendly to the database.

A Hybrid Sandbox



The strengths of a hybrid sandbox environment are similar to the strengths of the internal and external options, plus having ultimate flexibility in the approach taken for an analysis. It is easy to avoid production impacts during early testing if work is done on the external sandbox.

Another advantage is if an analytic process has been built and it has to be run in a “pseudo-production” mode temporarily while the full production system process is being deployed. Such processes can be run out of the internal sandbox easily.

The weaknesses of a hybrid environment are similar to the weaknesses of the other two options, but with a few additions. One weakness is the need to maintain both an internal and external sandbox environment. Not only will it be necessary to keep the external sandbox consistent with the production environment in this case, but the external sandbox will also need to be kept consistent with the internal sandbox.

It will also be necessary to establish some guidelines on when each sandbox option is used.

Workload Management and Capacity Planning

As analytic professionals start to use a sandbox, there are a lot of built-in components of database systems that will enable it to work smoothly. Sandbox users can be assigned to a group that has permissions that make sense for the purpose of developing new advanced analytics processes.

For example, it is possible to limit how much of the CPU a given sandbox user can absorb at one time.

One of the important things to do is to limit disk space usage through data retention policies. When a data set is in a sandbox and it hasn’t been touched in a couple of months, the default should be that it is deleted. A sandbox should not just continuously build up data sets, as often happens in traditional environments.

Especially with an internal sandbox, as more analytics are implemented, it will change the mix and level of resource usage in both the sandbox environment and the production environment.

12.Explain briefly about Analytic Data Set and its types, with suitable diagrammatic representations indicating the differences. [CO2-L1]

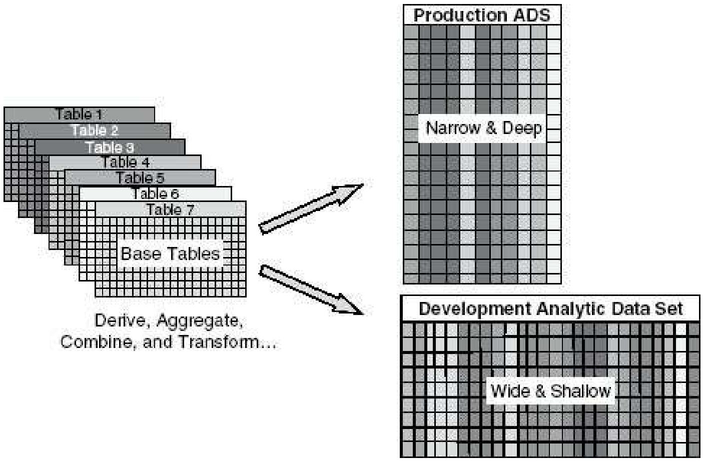
Analytic Data Set

An analytic data set (ADS) is the data that is pulled together in order to create an analysis or model. It is data in the format required for the specific analysis at hand. An ADS is generated by transforming, aggregating, and combining data. It is going to mimic a denormalized, or flat file, structure. What this means is that there will be one record per customer, location, product, or whatever type of entity is being analyzed. The analytic data set helps to bridge the gap between efficient storage and ease of use.

Development versus Production Analytic Data Sets

A development ADS is going to be the data set used to build an analytic process. It will have all the candidate variables that may be needed to solve a problem and will be very wide. A development ADS might have hundreds or even thousands of variables or metrics within it.

Development versus Production Analytic Data Sets



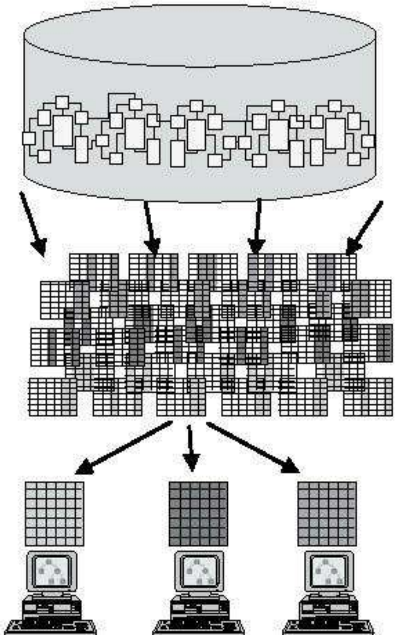
A production analytic data set, however, is what is needed for scoring and deployment. It’s going to contain only the specific metrics that were actually in the final solution. Typically, most processes only need a small fraction of the metrics explored during development. A big difference here is that the scores need to be applied to every entity, not just a sample.

Traditional Analytic Data Sets

In a traditional environment, all analytic data sets are created outside of the database Each analytic professional creates his or her own analytic data sets independently. This is done by every analytic professional, which means that there are possibly hundreds of people generating their own independent views of corporate data.

Traditional Analytic Data Set Process

Traditional Analytic Data Set Process:
A dedicated ADS is generated outside the database for every project.



A huge issue with the traditional approach to analytic data set generation is the repetitious work. If analytic professionals are creating very similar data sets again and again, it’s not just the space and system resources they are using, but it’s their time. They have to set up the ADS processes, they have to run them, and they have to babysit them and make sure they are complete.

Enterprise Analytic Data Sets

An EADS is a shared and reusable set of centralized, standardized analytic data sets for use in analytics.

Data Analytic Tools

13.Summarize the most commonly used data analytical tools used by Big Data Analyst. [CO2-L1]

Modern Data Analytic Tools

Data analysis tools and software are typically used to sort through enterprise data in order to identify patterns and establish relationships. Similar to "data mining," data analysis techniques can help virtually any business gain greater insight into organizational, industry, and customer trends.

basic statistical concepts

covering such things as what 'probability means', the notions of sampling and estimates based on samples, elements of inference, as well as more recently developed tools of intelligent data analysis such as cross-validation and bootstrapping

statistical model structures

Most intelligent data analysis involves issues of how variables are related, such as multivariate models, the wide range of generalised linear models, are a key tool in the data analyst's tool set.

Until recently, almost all statistical practice was carried out in the 'frequentist' tradition. This is based on an objective interpretation of probability, considering it as a real property of events.

Bayesian approach

A classical approach to supervised classification methods was to combine and transform the raw measured variables to produce 'features', defining a new data space in which the classes were linearly separable.

This basic principle has been developed very significantly in the view of support vector machines, which uses mathematics to permit the use of an effectively infinite number of features.

Early experience suggests that methods based on these ideas produce highly effective classification algorithms.

Statistics and machine learning

Statistics and machine learning are the two legs on which modern intelligent data analysis stands, but they have differences in emphasis.

One of these differences is the importance given to the interpretability of a model. For example, in both domains, recursive partitioning or tree methods have been developed.

Neural Network

In the context of data analytics an artificial neural network is a structure of simple processors with parameterised interconnections. By choosing the values of the parameters appropriately, one can use the network as a very flexible function estimator.

This makes such networks powerful tools – essentially providing a very good fit to a data set and then being shrunk to avoid overfitting problems. Their flexibility means that they are less subject to the bias problems essential to methods which assume a particular model form to start with (e.g. the linear form of classical regression).

Artificial neural networks are important because of their power as models, but they may turn out to be just as important because of the motivation they are giving to enhanced understanding of inference and the nature of induction.

Fuzzy Reasoning And Fuzzy Logic

Probability, and the theories of inferential statistics built on it, are the most widely accepted and used tool for handling uncertainty.

However, uncertainty comes in many shapes and forms. There is, for example, stochastic uncertainty arising from the basic mechanism leading to the data, but there is also uncertainty about the values of measurements or the meaning of terms.

While many (especially Bayesians) feel that this second kind can also be handled by probabilistic arguments, not everyone agrees, and other approaches have been developed.

Genetic Algorithms

One of the most exciting developments which has resulted from the growth of computer power has been the probabilistic solution of previously intractable methods by means of stochastic search and optimisation methods, such as simulated annealing and genetic algorithms

14.Explain with appropriate examples how various statistical concepts are applied by Data Scientists to understand data [CO2-L2]

Statistical Concepts

Statistics is the study of the collection, analysis, interpretation, presentation, and organization of data. In applying statistics to, e.g., a scientific, industrial, or social problem, it is conventional to begin with a statistical population or a statistical model process to be studied.

Populations and Parameters

A population is any large collection of objects or individuals, such as Americans, students, or trees about which information is desired.

A parameter is any summary number, like an average or percentage that describes the entire population.

The population mean μ and the population proportion p are two different population parameters. For example:

- We might be interested in learning about **μ**, the average weight of all middle-aged female Americans. The population consists of all middle-aged female Americans, and the parameter is **μ**.
- Or, we might be interested in learning about ***p***, the proportion of likely American voters approving of the president's job performance. The population comprises all likely American voters, and the parameter is ***p***.

The problem is that 99.999999999999... % of the time, we don't — or can't — know the real value of a population parameter.

Samples and statistics

A sample is a representative group drawn from the population.

A statistic is any summary number, like an average or percentage that describes the sample.

The sample mean, $\bar{\boldsymbol{x}}$, and the sample proportion $\hat{\boldsymbol{p}}$ are two different sample statistics. For example:

•We might use $\bar{\boldsymbol{x}}$ the average weight of a random sample of 100 middle-aged female Americans, to estimate **μ**, the average weight of all middle-aged female Americans.
•Or, we might use $\hat{\boldsymbol{p}}$, the proportion in a random sample of 1000 likely American voters who approve of the president's job performance, to estimate $\boldsymbol{p}$, the proportion of all likely American voters who approve of the president's job performance.

Because samples are manageable in size, we can determine the actual value of any statistic. We use the known value of the sample statistic to learn about the unknown value of the population parameter.

Confidence Intervals

Suppose we want to estimate an actual population mean μ. As you know, we can only obtain $\bar{\boldsymbol{x}}$, the mean of a sample randomly selected from the population of interest. We can use $\bar{\boldsymbol{x}}$ to find a range of values:

$$\boldsymbol{Lower\ value} < population\ mean\ \mu < Upper\ value$$

that we can be really confident contains the population mean μ. The range of values is called a ***"confidence interval"***

General form of most confidence intervals

The previous example illustrates the general form of most confidence intervals, namely:

$$\boldsymbol{Sample\ estimate} \pm \boldsymbol{margin\ of\ error}$$

That is:

$$\boldsymbol{the\ lower\ limit\ L\ of\ the\ interval = estimate - margin\ of\ error}$$

and:

$$\boldsymbol{the\ upper\ limit\ U\ of\ the\ interval = estimate + margin\ of\ error}$$

Once we've obtained the interval, we can claim that we are really confident that the value of the population parameter is somewhere between the value of L and the value of U.

So far, we've been very general in our discussion of the calculation and interpretation of confidence intervals. To be more specific about their use, let's consider a specific interval, namely the ***"t-interval for a population mean µ."***

Hypothesis Testing

The general idea of hypothesis testing involves:

1.Making an initial assumption.
2.Collecting evidence (data).
3.Based on the available evidence (data), deciding whether to reject or not reject the initial assumption.

Every hypothesis test, regardless of the population parameter involved, requires the above three steps.

Errors in hypothesis testing

Type I error: The null hypothesis is rejected when it is true.

Type II error: The null hypothesis is not rejected when it is false.

Test of Proportion

Let us consider the parameter p of population proportion. For instance, we might want to know the proportion of males within a total population of adults when we conduct a survey. A test of proportion will assess whether or not a sample from a population represents the true proportion from the entire population.

Unit – II

Introduction

Part – A

1.What is data analysis? [CO2-L1]

Data analysis is a process that aims to describe, summarise and evaluate data. This process looks at massive amounts of raw data with the purpose of drawing conclusions from it.

2.What are the sub processes of Intelligent Data Analysis? [CO2-L1]

The IDA process demands a combination of processes like extraction, analysis, conversion, classification, organization, and reasoning. It is challenging to choose appropriate methods to resolve the difficulties encountered in the process.

3.Classify the stages in the process of IDA. [CO2-L1]

The process of IDA typically consists of the following three stages:

1.Data preparation.
2.Data mining and rule finding.
3.Result validation and interpretation.

4.How is the quality of the information evaluated in IDA? [CO2-L1]

Rule finding performs data mining on the created data set and extracting the rules that summarise the information contained in the data set. It uses a set of standard algorithms. Result validation requires generating some guarantees of the quality of information. Interpretation of the rules so generated, results in a set of decisions that would be useful to an enterprise.

5.Justify the statement "Data Analysis is not a decision-making system, but a decision-supporting system" [CO2-L2]

Merely analysing data is not sufficient from the point of view of making a decision. How does one interpret from the analyzed data is more important. Thus, data analysis is not a decision-making system, but a decision-supporting system.

6.Data analysis is particularly relevant in this age of "big data". Justify. [CO2-L2]

Companies of all sizes are beginning to utilise big data and its associated analytics to gain information to better support their company and serve their customers.

Convergence of social, mobile, cloud and big data technologies presents new requirements - getting the right information to the consumer just in time, ensuring the reliability of external data, validating the relationships among data elements, etc. Thus, with the advent of big data, it is even more vital to find a way to analyze the ever (faster) growing disparate data

7.Can Data Analysis help in optimizing device performance? Support your claim. [CO2-L1]

Yes. Data analytics can be utilised to collect energy usage data from smart meters, analyze usage patterns and effectively provide smart grids that can optimise energy usage, etc.

Looking into usage patterns of machines at large manufacturing plants, analytics can effectively predict machine down time and that helps to perform preventive maintenance resulting in saving of huge amount of resources.

8.Analyse how Big Data Analysis could be beneficial to Financial Trading. [CO2-L2]

High-frequency trading is an area where big data finds a lot of use today. Data analytics technologies have advanced sufficiently to provide millisecond latency on large data sets. Here, big data analysis algorithms are used to make trading decisions. Today, the majority of equity trading now takes place via data algorithms that take into account signals from social media networks and news websites, weather predictions, etc. to make, buy and sell decisions in split seconds.

Regression Modelling

9.Suggest a method to calculate the degree of linear association between two variables in a Regression Model. [CO2-L1]

10.

Correlation analysis measures the degree of linear association between two variables. It is a statistical measure to indicate the degree of dependence of one variable over the other. The following formula is normally used to estimate the correlation factor between two variables X and Y:

$$\boldsymbol{r = \frac{n\sum XY - \sum X \sum Y}{\sqrt{(n\sum X^2 - (\sum X)^2)(n\sum Y^2 - (\sum Y)^2)}}}$$

Here, $\boldsymbol{X}$ is the independent variable and $\boldsymbol{Y}$ is the dependent variable.

11.What is the simplest way of fitting a regression line? [A]

The regression line could be straight or curved depending on the type of-the relationship between the two variables. The following linear function is used for linear regression:

$$\boldsymbol{Predicted\, Y \;=\; b_0 + b_1 X}$$

- The parameter $\boldsymbol{b_0}$ is called the intercept
- The parameter $\boldsymbol{b_1}$is called the slope of the regression line.

12.In what situation do data analysts opt to use Multiple Linear Regression? [CO2-L1]

Multiple regression is a statistical technique that is used when examining the relationship between three or more continuous variables and in which one of the variables can be thought of as predicted by the other variables. Similar to simple linear regression, in which there is only one independent or predictor variable, in multiple regression there are two or more independent variables.

13.Represent how the regression line is drawn. [CO2-L1]

Linear regression establishes a relationship between dependent variable $(\boldsymbol{Y})$ and one or more independent variables $(\boldsymbol{X})$ using a best fit straight line. This line is known as regression line.

The Regression line aims at minimizing the sum of the squared distance between the data points and the line.

14.Use the correlation coefficient to determine if the regression is positive or negative. [CO2-L1]

The correlation coefficient $\boldsymbol{r}$ indicates whether a relationship exists between the two variables and if so what is the degree of that relationship.

The value of $\boldsymbol{r}$ lies between -1 and +1,

- $-\mathbf{1}$ indicating a perfect negative linear relationship
- $+\mathbf{1}$ indicating a perfect positive linear relationship.
- A value around zero indicates no linear relationship between the two variables

15.Contradict the statement "Low correlation coefficient value means no relationship" [CO2-L2]

If there is a low correlation coefficient value does not necessarily mean that there is no dependency between the two variables. All it says is that there is no linear relationship between the variables but there may be a strong relationship of a ***non-linear*** kind.

Multivariate Analysis

16."The regression line is not always straight", Justify. [CO2-L2]

Using the principle of least squares regression, a regression line is fitted to the data in such a way so that the sum of the squares of the distances between the data points and the line is minimised.

The resulting regression line could be straight or curved depending on the type of-the relationship between the two variables. Therefore the regression line is not always straight.

17.Modify the simple linear regression function to fit in 3 predictors. [A]

The general form of a multiple regression model with k predictors is as follows:

$$\boldsymbol{Predicted\,Y} = \boldsymbol{b_0} + \boldsymbol{b_1 X_1} + \boldsymbol{b_2 X_2} + \boldsymbol{b_3 X_3}$$

18.List the stages in the linear regression process. [CO2-L1]

Linear regression consists of the following steps:

- Collect sample data of the dependent and independent values X and Y, respectively.
- Compute values of $\boldsymbol{b_0}$ and $\boldsymbol{b_1}$.
- Use these values to formulate the linear regression equation.
- Given new values for X predict value of Y.

19.What is Multiple Linear Regression? [CO2-L1]

Multiple regression is a statistical technique that is used when examining the relationship between three or more continuous variables and in which one of the variables can be thought of as predicted by the other variables.

20.When is MLR used and how is it modelled? [CO2-L1]

MLR is used when there are two or more independent variables where the model using population information is given by

$$\boldsymbol{\gamma_i} = \boldsymbol{\beta_0} + \boldsymbol{\beta_1 x_{i1}} + \boldsymbol{\beta_2 x_{i2}} + \cdots + \boldsymbol{\beta_p x_{ip}}$$

21.Suggest a simple method to identify curvilinear relationship between variables. [CO2-L1]

A visual analysis by plotting a graph of the dependent variable versus the independent variable will help identify any non-linear patterns that correlation analysis has failed to spot. If the results of data analysis show that there is a non-linear (also known as curvilinear) association between the two variables, then we need to develop a non-linear regression model to explain the data

Bayesian Modelling

22.What are the methods that can be use to handle non-linear data? [CO2-L1]

We can handle non-linear data in two ways: by using a polynomial rather than linear regression model or by transforming the data and then using a linear regression model.

23.How do we fit a polynomial function to model non-linear data? [CO2-L1]

Polynomials are equations that involve powers of the independent variable. The following two functions represent second-degree (quadratic) and third-degree (cubic) polynomial functions:

$$\boldsymbol{Predicted}(\boldsymbol{Y}) = \boldsymbol{b_0} + \boldsymbol{b_1 X} + \boldsymbol{b_2 X^2}$$
$$\boldsymbol{Predicted}(\boldsymbol{Y}) = \boldsymbol{b_0} + \boldsymbol{b_1 X} + \boldsymbol{b_2 X^2} + \boldsymbol{b_3 X^3}$$

24.What is coefficient of determination? [CO2-L1]

The coefficient of determination, denoted R^2 or r^2, is a number that indicates how well the data fits the developed model - a line or a curve. An $R^2$of 1 indicates that the regression model perfectly fits the data while an R^2 of 0 indicates that the model does not fit the data at all.

25.How is coefficient of determination calculated for MLR? [CO2-L1]

R^2 is the square of the coefficient of multiple correlations. The coefficient of determination ranges from 0 to 1:

Sum of Squares Regression (SSR):$\sum\left(\boldsymbol{\widehat{Y}_i - \overline{Y}_i}\right)^{\mathbf{2}}$

Sum of Squared Error (SSE):$\sum\left(\boldsymbol{\widehat{Y}_i - Y_i}\right)^{\mathbf{2}}$

Sum of Squared Total (SST):$\sum(\boldsymbol{Y_i - \overline{Y}_i})^{\mathbf{2}}$

Sum of Squared Total (SST) = SSR + SSE

$$\boldsymbol{R^2 = 1 - \frac{SSR}{SST}}$$

26.List out some of the most commonly used transformations. [CO2-L1]

The three most popular transformations are

•The square root $\left(\sqrt{\boldsymbol{X}}\right)$,

•The logarithm ($\mathbf{log}\,\boldsymbol{X}$)

•The negative reciprocal ($\boldsymbol{-1/X}$).

27.How is logistic regression different from linear regression? [CO2-L1]

Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution. Logistic regression extends the ideas of MLR to the situation where the dependent variable, $\boldsymbol{y}$, is binary

28.What is Multivariate Analysis? [CO2-L1]

MVA comprises a set of techniques dedicated to the analysis of data sets with more than one dependent variable. The MVA is essentially the statistical process of simultaneously analysing multiple independent (or predictor) variables with multiple dependent (outcome or criterion) variables using matrix algebra (most multivariate analyses are correlational).

29.List out the stages in the Bayesian Data Analysis. [CO2-L1]

Bayesian data analysis typically consists of the following three steps:

1.Setting up the prior distribution: Using domain expertise or prior knowledge to develop a joint probability distribution for all parameters of the data under consideration and also the output data (which needs to be predicted). This is termed as the prior distribution,

2.Setting up the posterior distribution: After taking into account the observed'data (given data set) calculating and interpreting the appropriate posterior distribution. This is estimating the conditional probability distribution of the data parameters, given the observed data.

3.Evaluating the fit of the model: How well does the developed model fit the data? Are the conclusions reasonable? How sensitive are the results to the modelling assumptions in step 1? In response, one can alter or expand the model and repeat the three steps.

30.Define the process of Bayesian Inference. [CO2-L1]

Bayesian inference derives the posterior probability of any event as a consequence of two factors, a prior probability and a ‘likelihood function derived from a statistical model for the observed data.

31.Derive the Bayesian Theorem. [CO2-L3]

Consider $P(H_i, D_{oba}|I)$ using the product rule:

$P(H_i, D_{oba}|I) = P(H_\mathrm{i}|I)P(D_{oba}|H_i, I)$

$= P(D_{oba}|I)P(H_\mathrm{i}|D_{oba}, I)$

Solve for the posterior probability:

$$P(H_i, D_{oba}|I) = P(H_i|I)\frac{P(D_{oba}|H_i, I)}{P(D_{oba}|I)}$$

Theorem holds for any propositions, but for hypotheses & data the factors have names:

$$Posterior \propto prior\ x\ likelihood$$

$$norm.const.P(D_{oba}\mid I) = prior\ predictive$$

Support Vector And Kernel Methods

32.Identify how different hypotheses can be compared. [CO2-L2]

When we need to compare different hypotheses, only the factors in the numerator are important, that is, P(H) and P(E | H). Thus, we have

1.*(more precisely)* The posterior probability of a hypothesis is determined by a combination of the inherent likeliness of a hypothesis (the prior) and the compatibility of the observed evidence with the hypothesis (the likelihood).

2.*(more concisely)* Posterior is proportional to likelihood times prior.

Bayes’ rule can also be written as follows:

$$P(H|E) = (P(H) * P(E|H)) * \alpha$$

33.Why the hypothesis that is most probable is picked in NBC? [CO2-L2]

Class conditional independence is giving by

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i)$$

The value $P(x_1|C_i), \dots, P(x_n, C_i)$ can be estimated from the training sample. The NBC combines this model with a decision rule. One common rule is to pick the hypothesis that is most probable; this is known as the maximum a posteriori or MAP decision rule.

34.What is the purpose of the BBN model? [CO2-L1]

The purpose of the BBN model is that we can determine the posterior probability distribution for a set of query variables given a set of observed events. We achieve the posterior probability distribution by using inference within the network.

35.List out any two situations that explain the learning of Bayesian networks. [CO2-L1]

The following two situations explain the learning of Bayesian networks:

1.The structure of the network is known: only the CPTs must be learned.
2.The structure of the network is unknown and must be learned.

If the network structure is known and given to us by a domain expert then it is a very easy to learn the probabilities. Use the data set giving values for all the attributes, simply compute the entries for the CPTs by counting, just as for NBC.

36.What is possibility of clustering unlabelled data when supervised learning is not possible? [CO2-L2]

In the case of unlabelled data, when supervised learning is not possible, we need an unsupervised learning method that can find natural clusters of data points. This clustering mechanism should be able to map new data points to any of the clusters formed. SVMs can be extended to what is called support vector clustering (SVC) and this is commonly used in industrial applications either when data is not labelled or when only some data is labelled as a pre-processing for a classification pass

37.Select the hyperplane for linear classification [CO2-L2]

There are many hyperplanes that might classify the data. One reasonable choice as the best hyperplane is the one that represents the largest separation, or margin, between the two classes. Hence, we choose that hyperplane such that the distance from it to the nearest data point on either side is maximized. If such a hyperplane exists, it is known as the maximum-margin hyperplane and the linear classifier it defines is known as a ***maximum-margin classifier***

Analysis Of Time Series

38.What is Taylor Expansion? [CO2-L1]

Taylor Expansion

Value of a multi-dimensional function $F(x)$ at point x can be approximated as

$$\boldsymbol{F(\vec{x}) = F(\vec{x}_0) + \nabla F(\vec{x}_0)(\vec{x}-\vec{x}_0) + (\vec{x}-\vec{x}_0)^T \nabla^2 F(\vec{x}_0)(\vec{x}-\vec{x}_0) + O}(\|\vec{x}-\vec{x}_0\|^3)$$

Therefore, $\boldsymbol{F(x)}$ can be considered as a linear combination of complex attributes derived from the original ones:

$$\boldsymbol{F(\vec{x}) = F(\vec{x}_0) + \sum_{i=1}^{m} a_i x_i + \sum_{i,j=1}^{m} a_{ij} x_i x_j + \sum_{i,j,k=1}^{m} a_{ijk} x_i x_j x_k + O}(\|\vec{x}-\vec{x}_0\|^3)$$

39.In SVM, if the training sets are linearly separable, where does the maximum margin hyperplane lie? [CO2-L2]

If the training data are linearly separable, we can select two parallel hyperplanes that separate the two classes of data, so that the distance between them is as large as possible. The region bounded by these two hyperplanes is called the "margin", and the maximum-margin hyperplane is the hyperplane that lies halfway between them. These hyperplanes can be described by the equations

$$\boldsymbol{\vec{w} \cdot \vec{x} + b = 1}$$

And

$$\boldsymbol{\vec{w} \cdot \vec{x} + b = -1}$$

40.Why is hinge loss function introduced in SVM? [CO2-L1]

To extend SVM to cases in which the data are not linearly separable, we introduce the hinge loss function,

$$\boldsymbol{\max(0, 1 - y_i(\vec{w} \cdot \vec{x} + b))}$$

This function is zero if the constraint is satisfied, in other words, if $\boldsymbol{\vec{x}_i}$lies on the correct side of the margin. For data on the wrong side of the margin, the function's value is proportional to the distance from the margin.

41.List out some of the potential drawbacks of the SVM. [CO2-L1]

Potential drawbacks of the SVM are as follows:

- Requires full labelling of input data
- Uncalibrated class membership probabilities
- The SVM is only directly applicable for two-class tasks. Therefore, algorithms that reduce the multi-class task to several binary problems have to be applied; see the multi-class SVM section.
- Parameters of a solved model are difficult to interpret.

Rule induction

42.What is Rule Induction? [CO2-L1]

Rule induction is an area of machine learning in which formal rules are extracted from a set of observations. The rules extracted may represent a full scientific model of the data, or merely represent local patterns in the data.

43.Relate confidence and support in association rule. [A]

Confidence is an indication of how often the rule has been found to be true.

The *confidence* value of a rule, $X \Longrightarrow Y$ with respect to a set of transactions T is the proportion of the transactions that contains X which also contains Y

Confidence is defined as:

$$conf(X \Longrightarrow Y) = supp(X \cup Y)/Supp(X)$$

44.What is the purpose of linearizing the decision tree? [CO2-L1]

The decision tree can be linearized into **decision rules**, where the outcome is the contents of the leaf node, and the conditions along the path form a conjunction in the if clause. In general, the rules have the form:
if condition1 *and* condition2 *and* condition3 *then* outcome.

Decision rules can also be generated by constructing association rules with the target variable on the right.

Neural networks

45.When the problem of Overfitting does occurs? [CO2-L1]

One of the problems that occur during neural network training is called overfitting. The error on the training set is driven to a very small value, but when new data is presented to the network the error is large. The network has memorized the training examples, but it has not learned to generalize to new situations.

46.Suggest a way to improve generalization in ANN when caused by noisy data. [A]

A simple way to improve generalization, especially when caused by noisy data or a small dataset, is to train multiple neural networks and average their outputs.

For instance, here 10 neural networks are trained on a small problem and their mean squared errors compared to the means squared error of their average.

First, the dataset is loaded and divided into a design and test set.

47.How can the performance function be improved to increase the efficiency of ANN? [CO2-L2]

The typical performance function used for training feed forward neural networks is the mean sum of squares of the network errors.

It is possible to improve generalization if you modify the performance function by adding

Part – B

Data Analysis

Write a short note on Data Analysis and its Importance. [CO2-L1]

Data Analysis

Data analysis is a process that aims to describe, summarise and evaluate data. This process looks at massive amounts of raw data with the purpose of drawing conclusions from it. It is used in many industries, and allows the companies and organization to make better business decisions. In the field of research it is used to verify or disprove existing models or theories. In a more detailed definition, data analysis is a process of inspecting, cleaning, transforming and **MODELLING DATA WITH THE GOAL OF DISCOVERING USEFUL INFORMATION, SUGGESTING CONCLUSIONS AND SUPPORTING DECISION-MAKING.**

Intelligent data analysis (IDA) uses concepts from **ARTIFICIAL INTELLIGENCE, INFORMATION RETRIEVAL, MACHINE LEARNING, PATTERN RECOGNITION, VISUALIZATION DISTRIBUTED PROGRAMMING** and a host of other computer science concepts to automate the task of extracting implicit, previously unknown and potentially valuable information or knowledge from large amounts of data. It helps to extract desirable knowledge or interesting patterns from existing large data stores.

IDA is also a kind of **decision support process**, as it extracts useful knowledge and interesting models from a lot of online data in order to help decision makers make the right choices. Thus, IDA is an interdisciplinary study concerned with the effective data analysis.

The IDA process demands a combination of processes like ***extraction, analysis, conversion, classification, organization, and reasoning***. It is challenging to choose appropriate methods to resolve the difficulties encountered in the process. IDA methods and tools, as well as the authenticity of obtained results, pose us continued challenges.

The process of IDA typically consists of the following three stages:

1.Data preparation.
2.Data mining and rule finding.
3.Result validation and interpretation.

DATA PREPARATION involves selecting the required data from several data sources and integrating them into a data set, which can then be used for data mining. RULE FINDING means performing data mining on the created data set and extracting the rules that summarise the information contained in the data set. It uses a set of standard algorithms. RESULT VALIDATION requires generating some guarantees of the quality of information. INTERPRETATION of the rules so generated, results in a set of decisions that would be useful to an enterprise.

Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, in different business, science and social science domains.

Data mining is a particular data analysis technique that focuses on modelling and knowledge discovery for predictive rather than purely descriptive purposes. Business intelligence covers data analysis that relies heavily on aggregation, focusing on business information. In statistical applications, some people divide data analysis into descriptive statistics, **EXPLORATORY DATA ANALYSIS** (EDA) and **CONFIRMATORY DATA ANALYSIS** (CDA). EDA focuses on discovering new features in the data and CDA on confirming or falsifying existing hypotheses. Predictive analytics focuses on the application of statistical models for predictive forecasting or classification while text analytics applies statistical, linguistic and structural techniques to extract and classify information from textual sources, a species of unstructured data. All the above are varieties of data analysis. Data integration is a precursor to data analysis, and data analysis is closely linked to data visualization and data dissemination.

Various IDA techniques includes Regression ***Multivariate Analysis (MVA), Bayesian Modelling, Bayesian Networks, Support Vector Matching and Kernel Methods, Time Series Methods and Rule Induction***

Importance Of Data Analysis

That data analysis is important to businesses will be an understatement in today's world. In fact, we can clearly and emphatically state that no business can survive today without effectively analysing the available data and taking effective just in time decisions. For example, VISUALISE the following situations:

1.A drug company is performing trials on a number of patients to test its new drug to fight against diabetes. The number of patients under the trial is well over 50,000.
2.A company wants to launch a new variant of its existing line of sports shoes. It wants to carry out the survey analysis and arrive at some meaningful conclusion. Sales director of a company knows that there is something wrong with one of its successful

products, however, has not yet carried out any market research on data analysis. How and what does he conclude?

The above situations are indicative enough to conclude that data analysis is the lifeline of any business. Whether one wants to arrive at some marketing decisions or fine-tune new product launch strategy, data analysis is the key to all the problems.

Merely analysing data is not sufficient from the point of view of making a decision. How does one interpret from the analyzed data is more important. Thus, ***data analysis is not a decision-making system, but a decision-supporting system***. Data analysis offers the following benefits:

1.Structuring the findings from survey research or other means of data collection.
2.Provides a picture of data at several levels of granularity from a macro picture into a micro one.
3.Acquiring meaningful insights from the data set which can be effectively exploited to take some critical decisions to improve productivity.
4.Helps to remove human bias in decision making, through proper statistical treatment.

Data analysis is particularly relevant in this age of “big data”. In fact, companies of all sizes are beginning to utilise big data and its associated analytics to gain information to better support their company and serve their customers.

Further **convergence of social, mobile, cloud and big data technologies** presents new requirements - getting the right information to the consumer just in time, ensuring the reliability of external data, validating the relationships among data elements, etc. Thus, with the advent of big data, it is even more vital to find a way to analyze the ever (faster) growing disparate data coursing through their environments and give it meaning.

Point out some of the applications of Data Analysis and its impact on various fields. [CO2-L1]

Data Analytics Applications

Big data analytics will very soon impact every aspect of modern life. Most of these applications can be categorised into the following 10 areas:

Understanding and targeting customers:

Data analytics is extremely useful to understand and predict customer behaviour. The trend is towards getting a 360 degree view of each customer which includes data from traditional customer purchase data as well as the non-traditional unstructured data sets like social media, web logs, customer clicks on e-retail sites, etc. This picture helps businesses predict which customers may move to a rival (called customer churn), predict what products will sell, predict living patterns that can help insurance companies charge a differential premium and so on. The list is endless with potential business benefits.

Understanding and optimizing business processes:

Information is collected from social media resulting in sentiment analysis. This along with company profiles data are analyzed using data analytics tools to effectively predict demand for products and thus help to retain stock in warehouses to an optimal level. For example, Apple excessively uses sentiment analysis information from social media to gauge the potential sales of their new iPhone 6s offering. By providing a holistic view of assets and business processes, enterprises are now able to gain unparalleled insight into optimizing those assets and processes. For example, a Fortune 50 company was able to optimise accounts receivable collections without increasing collector headcount. This was possible because data analytics tools crawled and mined historical data, identified the factors that affect late payments, provided insights in the collection system, and provided ongoing recommendations that helped improved Accounts Receivable collections by 65% over the prior year.

Personal quantification and performance optimization:

Personal quantification is an emerging trend in big data science. Self-quantification of personal health and wellness data are contributing heavily towards more self-managed health care. Advances in network and wearable sensor technologies easily help to capture and share significant health-related information on a daily basis. New functionalities in wearable devices and the associated apps enable individuals to measure vital signs, access analytical tools and quantify data about themselves faster and more ubiquitously than ever before. For example, keeping diaries of food intake, converting these collected data into numbers, analysing them and using them to make better decisions regarding personal health are part of personal quantification.

Improving healthcare and public health:

The healthcare industry historically has massive amounts of data in its archives. This may be due to record keeping, compliance and regulatory requirements, continual patient care, etc. This voluminous data totally renders itself to data analytics applications. This data includes clinical data from hospitals and clinical decision support systems (physicians written notes and prescriptions, medical imaging, laboratory test data, pharmacy prescriptions and sales, insurance data, patient data in electronic patient records (EPRs)) machine generated/sensor data, such as from monitoring vital signs; social media data, news feeds, and articles in medical journals, etc. This data can be analyzed to effectively provide customised medical care to patients, detect epidemics, curtail infections and several such applications. Interesting current application include holistic cancer treatment, genomics, identifying and stopping hospital fraud and the like.

Improving sports performance:

Any sports gets affected by advances in the capture, storage and analysis of data. Data analytics allows athletes to train better and more effectively and it allows teams to alter their in-game decision-making based on what they are seeing. Like other businesses, sports teams strive to make better decisions faster. Coaching staff, and players are leveraging analytics to better understand the performance of their own teams as well as that of the opposition.

It is also changing the way broadcasters produce sports entertainment and the way teams and broadcasters engage with fans. Big data is also playing an increasingly important role not just in broadcast production, but in broadcast and digital distribution. Multi-platform distribution of content and the use of social media and consumer-generated content in broadcasts uses metrics to determine popular content and are able to offer different pricing models.

Improving science and research:

The emergent field of data analytics is rapidly changing the direction and speed of scientific research by letting people fine-tune their inquiries by tapping into giant data sets. In the past, certain fields of science relied heavily on big data sets, such as high-energy particle physics or research on nuclear fusion. But as information becomes available from more sources, collecting and analysing large amounts of data is becoming common in other fields of research too. One such recent example is the research being conducted at CERN, the Swiss nuclear physics laboratory with its Large Hadron Collider. The CERN data centre has 65,000 processors to analyze its 30 petabytes of data. It uses the computing powers of thousands of computers distributed across 150 data centres worldwide to analyze the data.

Optimizing machine and device performance:

Big data analytics helps machines and devices become smarter and more autonomous. Data analytics can be utilised to collect energy usage data from smart meters, analyze usage patterns and effectively provide smart grids that can optimise energy usage. Looking into usage patterns of machines at large manufacturing plants, analytics can effectively predict machine down time and that helps to perform preventive maintenance resulting in saving of huge amount of resources.

Improving security and law enforcement:

By taking advantage of big data, crime analysts identify trends and make recommendations based on then observations. Through analysis and computer mapping, crime analysts play a crucial role in helping law enforcement agencies quantify, evaluate, and respond to the changing landscape of criminal activity in their jurisdictions. Typical applications use big data techniques to detect and prevent cyberattacks. Police forces use big data tools to catch criminals and even predict criminal activity and credit card companies use big data to detect fraudulent transactions.

Improving and optimizing cities and countries:

Today s towns and cities generate about 5 terabytes per day per square kilometre of urbanized land area. This includes location data collected by smart phones to data generated by GPS instruments, payment cards, smart ID cards, loyalty and store cards, bank cards, toll payments, etc. Further sources of data include data created by traffic management systems, from traffic lights to the sensors on our roads; from the provision of utilities such as gas, electricity and drinking water, etc. All this data can be analyzed to improve many aspects of our cities and citizens daily life. For example, we can have intelligent route planning systems that are based on real-time traffic information as well as social media and weather data. Smart cities are planned by integrating and analysing all subsystems in a city like energy, traffic, police, etc.

Financial trading:

High-frequency trading is an area where big data finds a lot of use today. Data analytics technologies have advanced sufficiently to provide millisecond latency on large data sets. Here, big data analysis algorithms are used to make trading decisions. Today, the majority of equity trading now takes place via data algorithms that take into account signals from social media networks and news websites, weather predictions, etc. to make, buy and sell decisions in split seconds.

1.Explain the purpose of using Regression Modelling in Data Analysis and what kind of inferences it provides [CO2-L1]

Regression Modelling Techniques

One fundamental task in data analysis is to attempt to find how different variables are related to each other, and one of the central tools in statistics for learning about such relationships is regression. The basic idea behind regression is ***"Use the existing historical data to identify potential relationships among variables"*** and then ***"use these relationships to make predictions about the future***".

Regression analysis is a statistical process for ***estimating the relationships among variables***. It helps to model and analyze several variables when the focus is on the relationship between a dependent variable and one or more independent variables (or "predictors"). For example, the effect of a price increase in petrol upon demand for petrol-run cars or say the effect of global fall in oil prices to the inflation rate, etc.

It also helps to study how the changing behaviour of a **SET OF PREDICTORS** affects the behaviour of the dependent variable and allows us to use numerical values to model these effects. In simple terms, regression analysis allows us to model the dependent variable as a function of its predictors. Regression techniques assume the existence of a large volume of data on the underlying variables of interest and use this data to estimate the **QUANTITATIVE EFFECT** of the causal variables upon the variable that they influence. As a by-product, regression also attempts to assess the ***"statistical significance"*** of the estimated model which gives an estimate of the precision of the model.

Regression analysis is widely used for **prediction and forecasting** and thus is a useful data analysis technique. It can also be used for causal inference and descriptive modelling. There are various kinds of regression techniques available. These techniques are mostly driven by three metrics: number of independent variables, type of dependent variables and **SHAPE OF REGRESSION LINE**. In the following sections, we shall discuss some commonly used regression models.

1.Linear regression establishes a relationship between dependent variable (T) and one or more independent variables (X) using a best fit straight line (also known as regression line). This includes simple linear regression analysis, in which there is a single independent variable and the equation for predicting a dependent variable Y is a linear function of a given independent variable X. The **multiple linear regression**

(MLR) model, finds the relationship of a variable Y to a setoff quantitative explanatory variables but still in a linear fashion.

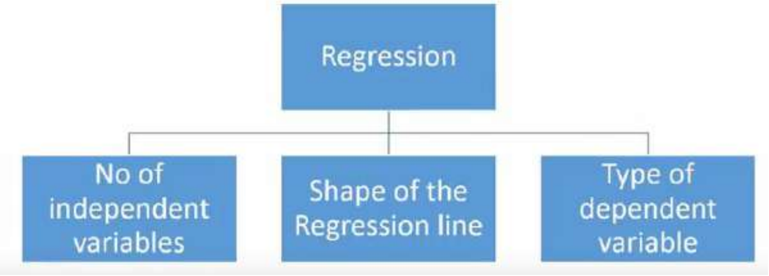
2.Logistic regression is the standard way to model **BINARY OUTCOMES**. It is used to find the probability of event = success and event = failure.

3.If the relationship between the variables being analyzed is not linear in parameters, a number of ***non-linear regression techniques*** may be used to obtain a more accurate regression. This is also sometimes called polynomial regression.

4.There may be situations where the number of dependent variables may be more than one. We thus need models which jointly determine the values of two or more dependent variables using two or more equations. Such models are called multivariate regression models because they attempt to EXPLAIN MULTIPLE DEPENDENT VARIABLES.

Types Of Regression Techniques

There are various kinds of regression techniques available to make predictions. These techniques are mostly driven by three metrics (number of independent variables, type of dependent variables and shape of regression line).


Regression
No of independent variables
Shape of the Regression line
Type of dependent variable

Use of Regression Analysis

As mentioned above, regression analysis estimates the relationship between two or more variables. Let’s understand this with an easy example:

Let’s say, you want to estimate growth in sales of a company based on current economic conditions. You have the recent company data which indicates that the growth in sales is around two and a half times the growth in the economy. Using this insight, we can predict future sales of the company based on current & past information.

There are multiple benefits of using regression analysis. They are as follows:

1. It indicates the **significant relationships** between dependent variable and independent variable.

2. It indicates the **strength of impact** of multiple independent variables on a dependent variable.

Regression analysis also allows us to compare the effects of variables measured on different scales, such as the effect of price changes and the number of promotional activities. These benefits help market researchers / data analysts / data scientists to eliminate and evaluate the best set of variables to be used for building predictive models.

Explain the concept of Linear Regression and Calculate the Correlation Coefficient between the annual sales and number of agents working for the insurance company using the given data, also describe the type of relationship [E]

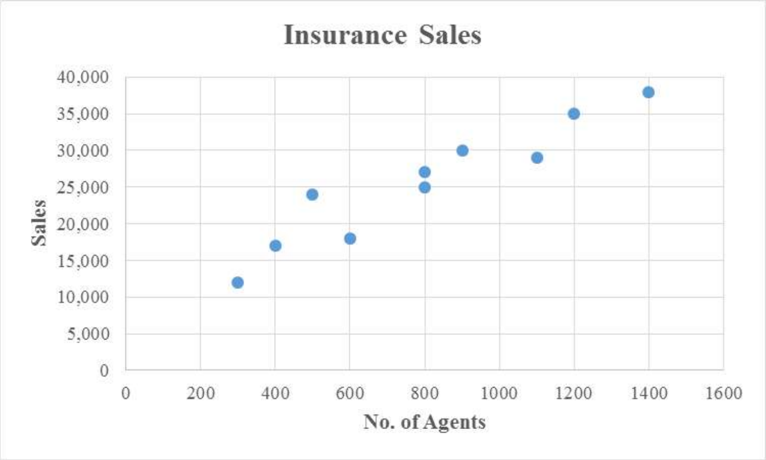
Company	Sales (in 1000s)	No. of Agents (in 100s)
A	25	8
B	35	12
C	29	11
D	24	5
E	38	14
F	12	3
G	18	6
H	27	8
I	17	4
J	30	9

Linear Regression

The regression model to be developed will relate the volume of sales to the number of agents. As it is logical to expect the volume of sales to depend on the number of agents and not the other way round, we denote sales as the dependent variable and agent number to be the independent variables

We must first verify that the sales is indeed affected by the number of agents before we set out to develop the regression model. This ensures that the regression model developed can be utilized effectively to produce accurate forecasts and thus make a significant contribution to decision making. To test the dependence between any two variables, two common techniques are normally employed. These include a graphical tool like a scatter diagram or a statistics-based tool correlation analysis. The results from the analysis of data will tell us whether to use regression analysis as the forecasting tool and what type of regression model to develop.

The scatter diagram shows the volume of sales against the number of agents for the insurance data



The relationship between two variables can be tested graphically using the scatter diagram. We can clearly see that as the number of agents increases so does the sales of insurance policies and vice versa. This indicates that a relationship exists between the two variables. As this increase is linear (i.e. the value of $\boldsymbol{Y}$ increases with the value of $\boldsymbol{X}$ a linear way), the relationship between the two variables is a linear relationship.

Correlation Coefficient Calculation

Correlation analysis measures the degree of linear association between two variables. It is a statistical measure to indicate the degree of dependence of one variable over the other. The following formula is normally used to estimate the correlation factor between two variables X and Y:

$$\boldsymbol{r} = \frac{\boldsymbol{n}\sum \boldsymbol{XY} - \sum \boldsymbol{X} \sum \boldsymbol{Y}}{\sqrt{(\boldsymbol{n}\sum \boldsymbol{X^2} - (\sum \boldsymbol{X})^2)(\boldsymbol{n}\sum \boldsymbol{Y^2} - (\sum \boldsymbol{Y})^2)}}$$

Here, $\boldsymbol{X}$ is the independent variable and $\boldsymbol{Y}$ is the dependent variable.

$\sum \boldsymbol{X}$ is the sum of all the values of the independent variable and $\sum \boldsymbol{X^2}$ is the sum of all the squared values of the independent variable.

$\boldsymbol{n}$ is the number of observations (the number of data points in the sample).

The value computed, $\boldsymbol{r}$ is known as the ***correlation coefficient*** and its value determines the both, the strength and direction of the linear association between the two variables under examination.

The value of $\boldsymbol{r}$ indicates whether a relationship exists between the two variables and if so what is the degree of that relationship.

Further, in the case of a relationship existing between ***X*** and ***Y***, the value of ***r*** will also indicate whether the value of the dependent variable increases or decreases as the value of the independent variable goes up.

The value of $\boldsymbol{r}$ lies between -1 and +1,

➥$\mathbf{-1}$ indicating a perfect negative linear relationship
➥$\mathbf{+1}$ indicating a perfect positive linear relationship.
➥A value around zero indicates no linear relationship between the two variables.

Thus, correlation coefficient of values near -1 or +1 indicate a strong correlation between the two variables, whereas the values of the correlation coefficient near zero indicate no correlation between the two variables

Y Sales (1000s)	**X No. of Agents (100s)**	$\boldsymbol{Y^2}$	$\boldsymbol{X^2}$	$\boldsymbol{XY}$
25	8	625	64	200
35	12	1225	144	420
29	11	841	121	319
24	3	576	25	120
38	14	1444	196	532
12	3	144	9	36
18	6	324	36	108
27	8	729	64	216
17	4	289	16	68
30	9	900	81	270
255	80	7097	756	2289

$\boldsymbol{n = 10}$; $\sum \boldsymbol{X = 80}$; $\sum \boldsymbol{Y = 255}$; $\sum \boldsymbol{XY = 2289}$; $\sum \boldsymbol{X^2 = 756}$; $\sum \boldsymbol{Y^2 = 7097}$; $(\sum \boldsymbol{X})^2 = (\mathbf{80})^2 = \mathbf{6400}$; $(\sum \boldsymbol{Y})^2 = (\mathbf{255})^2 = \mathbf{65025}$

Substituting the above results into the formula, we have the correlation coefficient:

$$\boldsymbol{r} = \frac{\mathbf{10 \times 2289 - (80 \times 255)}}{\sqrt{(\mathbf{10 \times 756 - (80)^2}) \times (\mathbf{10 \times 7097 - (255)^2})}} = \mathbf{0.9482 \approx 0.95}$$

Since the Correlation coefficient is close to +1, the relationship between number of agents and the total sales is positive linear regression.

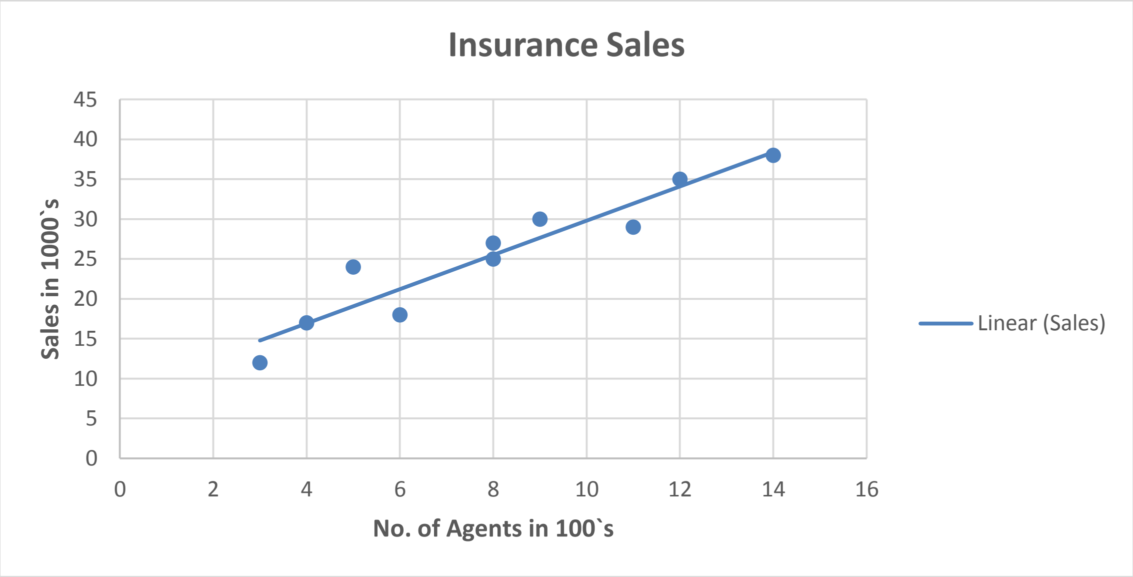
Develop a linear regression model using the data give below [C]

Company	**Sales (in 1000s)**	**No. of Agents (in 100s)**
A	25	8
B	35	12
C	29	11
D	24	5
E	38	14

F	12	3
G	18	6
H	27	8
I	17	4
J	30	9

Developing a Linear Regression Model

For the data give, we can see that both the scatter diagram and the value of the correlation coefficient (0.93) indicates that a strong linear relationship exists between agent numbers and policy sales. We shall, therefore, try to develop a linear model for the given data.



Regression line for the data.

The graph shows the volume of sales against agent numbers fitted with a straight line. This line is called regression line and it is the result of using regression analysis. The process that produced this line is as follows

Fitting the Regression Line

This regression line has been fitted in such a way that the sum of the distances between the data points and the line (i.e. the gaps between the data points and the line) is minimised. Because of this, the regression line is also known as the line of best fit.

One way to fit such a line is to visually draw a straight line through the data points of the scatter diagram in such a way that the line passes through all the points as closely as possible. The problem with this approach, however, is that, no matter how good our fit is, one could come up with a better fit. We, therefore, fit a regression line using a more statistical approach, which is known as the least squares regression method.

Using the principle of least squares regression, a regression line is fitted to the data in such a way so that the sum of the squares of the distances between the data points and the line is minimised.

The resulting regression line could be straight or curved depending on the type of-the relationship between the two variables. The following linear function is used for linear regression:

$$\boldsymbol{Predicted\,Y\ =\ b_0 + b_1X}$$

- The parameter $\boldsymbol{b_0}$ is called the intercept
- The parameter $\boldsymbol{b_1}$is called the slope of the regression line.

The value of the intercept determines the point where the regression line meets the T-axis of the graph. The value of the slope represents the amount of change in Y when X increases by one unit.

The literature frequently also uses the term predictor to refer to the independent variable. A simple regression model uses only one predictor. A model which uses more than one predictor is called a multiple regression model. The general form of a multiple regression model with k predictors is as follows:

$$\boldsymbol{Predicted\,Y = \ b_0 + b_1X_1 + b_2X_2 + \cdots + b_kX_k}$$

In order to develop a linear regression model of the form $\boldsymbol{Predicted\,Y = \ b_0 + b_1X}$, we need to calculate the values of $\boldsymbol{b_0}$ and $\boldsymbol{b_1}$. These values are given by the following relations:

$$\boldsymbol{b_1 = \frac{n\sum XY - \sum X \sum Y}{\sum X^2 - (\sum X)^2}}$$

$$\boldsymbol{b_0 = \frac{\sum Y}{n} - b_1 \frac{\sum X}{n}}$$

Application of the above formulae to the insurance example data will produce the following results:

$$\boldsymbol{b_1 = \frac{10 \times 2289 - (80 \times 255)}{(10 \times 756 - (80)^2)} = 2.1466}$$

$$\boldsymbol{b_0 = \frac{255}{10} - 2.1466\frac{80}{10} = 8.3272}$$

The regression equation will thus be

$$\boldsymbol{Predicted\ (Y)\ =\ 8.3272\ + 2.1466X}$$

The above equation can thus be used to predict the volume of sales for an insurance company given its agent numbers. Thus, if a company has 1000 agents (10 hundreds) the predicted value of sales will be around 30,000.

In summary, linear regression consists of the following steps:

- Collect sample data of the dependent and independent values X and Y, respectively.
- Compute values of $\boldsymbol{b_0}$ and $\boldsymbol{b_1}$.
- Use these values to formulate the linear regression equation.
- Given new values for X predict value of Y.

The regression model is learned from existing sample data and then used to predict future values. Larger and better the sample of data, more accurate will be the regression model and thus this will lead to more accurate forecasts.

2.Suggest a way of handling a curved regression line and what to do when there are more than one independent variables. [CO2-L2]

Handling Non-Straight Regression Lines

The resulting regression line could be straight or curved depending on the type of-the relationship between the two variables. The following linear function is used for linear regression:

$$\boldsymbol{Predicted\ Y\ =\ b_0 + b_1X}$$

- The parameter $\boldsymbol{b_0}$ is called the intercept
- The parameter $\boldsymbol{b_1}$is called the slope of the regression line.

The value of the intercept determines the point where the regression line meets the Y-axis of the graph. The value of the slope represents the amount of change in Y when X increases by one unit.

A model which uses more than one predictor is called a multiple regression model. The general form of a multiple regression model with k predictors is as follows:

$$\boldsymbol{Predicted\,Y = b_0 + b_1X_1 + b_2X_2 + \cdots + b_kX_k}$$

In order to develop a linear regression model of the form $\boldsymbol{Predicted\,Y = b_0 + b_1X}$, we need to calculate the values of $\boldsymbol{b_0}$ and $\boldsymbol{b_1}$. These values are given by the following relations:

$$\boldsymbol{b_1 = \frac{n\sum XY - \sum X \sum Y}{\sum X^2 - (\sum X)^2}}$$

$$\boldsymbol{b_0 = \frac{\sum Y}{n} - b_1\frac{\sum X}{n}}$$

Multiple Linear Regression (MLR)

Multiple regression is a statistical technique that is used when examining the relationship between three or more continuous variables and in which one of the variables can be thought of as predicted by the other variables. Similar to simple linear regression, in which there is **only one independent or predictor variable, in multiple regression there are two or more independent variables**.

MLR is the model of choice in various data analytics applications like predicting customer activity on credit cards from customer locations and asset profiles, predicting the time to failure of equipment based on utilization and environment conditions, predicting expenditures on vacation travel based on historical frequent flier data, predicting staffing requirements at help desks based on historical data and product and sales information, predicting sales from cross-selling of products from historical information and predicting the impact of discounts on sales in retail outlets.

MLR is an extension of simple linear regression. As discussed above, it is reasonable to assume that most real-life scenarios are too complex to be modelled using just one independent variable. MLR is used when there are two or more independent variables where the model using population information is given by

$$\boldsymbol{\gamma_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}}$$

The sample data consist of $\boldsymbol{n}$ rows of observations also called cases, which give us values $\boldsymbol{\gamma_i, x_{i1}, x_{i2}, \dots, x_{ip}, i = 1, 2, \dots, n}$. The estimates for the $\boldsymbol{\beta}$ coefficients are computed so as to minimise the sum of squares of differences between the fitted (predicted) values and the observed values in the data. The sum of squared differences is given by

$$\sum_{i=1}^{n}\left(\gamma_i - \beta_0 - \beta_1 x_{i1} - \beta_2 x_{i2} - \cdots - \beta_p x_{ip}\right)^2$$

The data results in n equations where we need to solve for values of $ the coefficients. Any numerical method or business computer software such as Excel, SAS, etc. will solve these equations and return the coefficient values. This helps to build the model. On the arrival of new data for the independent variables $x_{i1}, x_{i2}, \ldots, x_{ip}$, we can predict the outcome γ_i. NEURAL NETWORKS OR GENETIC ALGORITHMS can also be effectively used to find the coefficient values.

3.What regression technique could be used when the dependent variable is not linearly dependent on the independent variable? Explain using an example. [CO2-L1]

Non-Linear Regression

Both in the case of linear and multiple linear regression, the dependent variable is linearly dependent on the independent variable(s). But, in several situations, the situation is not so simple where the two variables might be related in a non-linear way. This may be the case where the results from correlation analysis show no linear relationship but these variables might still be closely related. A visual analysis by plotting a graph of the dependent variable versus the independent variable will help identify any non-linear patterns that correlation analysis has failed to spot. If the results of data analysis show that there is a non-linear (also known as curvilinear) association between the two variables, then we need to develop a non-linear regression model to explain the data,

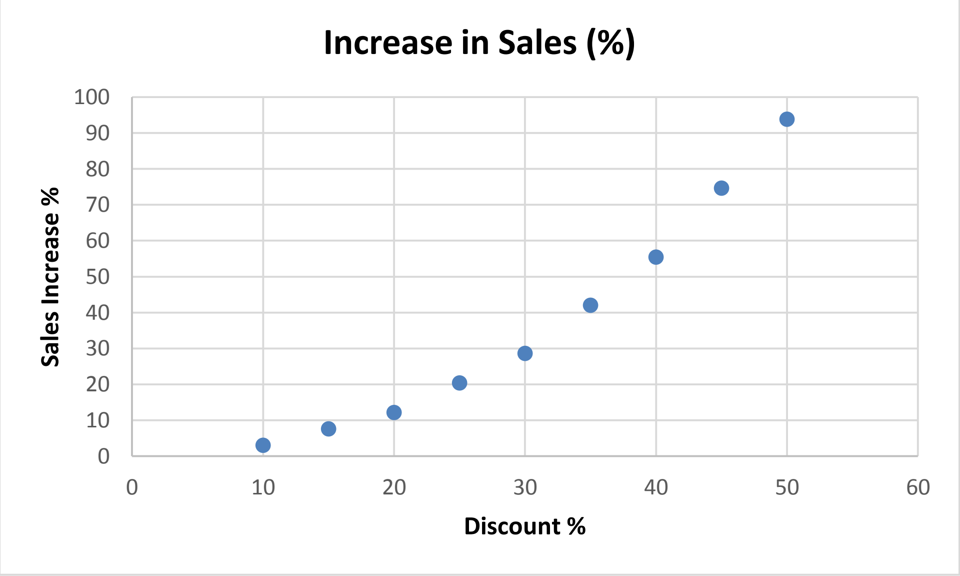
We can handle non-linear data in two ways: by using a polynomial rather than linear regression model or by transforming the data and then using a linear regression model. The two methods are covered in the rest of this section.

We shall illustrate the process by using an example. Table below shows the percentage increase in sales at a retail store based on discount percentages.

Retail store example data based on discount percentages.

Discount (%)	**Increase in Sales (%)**

10	3.05
15	7.62
20	12.19
25	20.42
30	28.65
35	42.06
40	55.47
45	74.68
50	93.88



The scatter diagram of sales increase for various discount percentages.

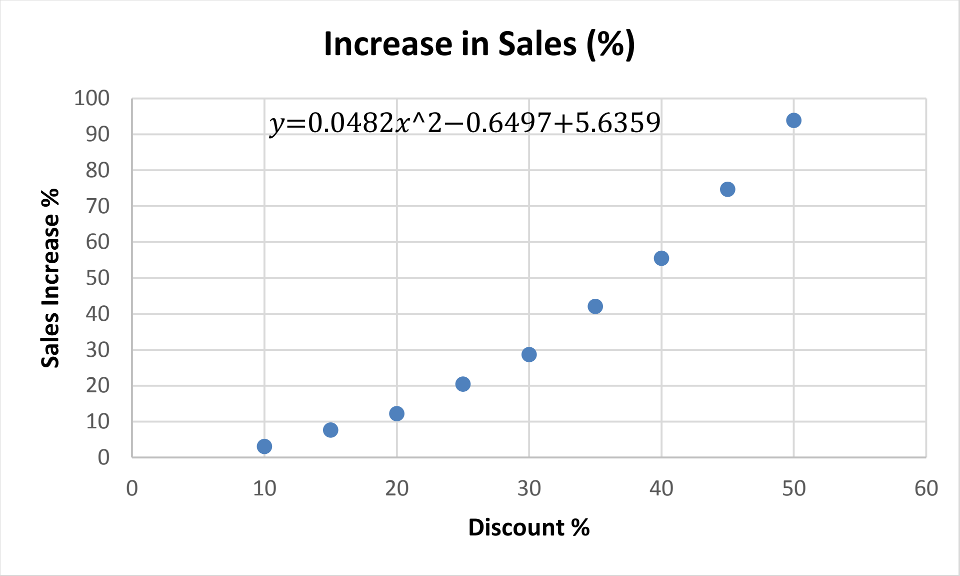
At first glance, the graph seems to be indicative of the two variables being related in a linear way. In fact, this is confirmed by correlation analysis (the value of r in this case has been calculated to be 0.97, indicating a very strong, almost perfect, positive correlation). However, on closer inspection, we will notice that the data values appear to form a slight curve. This indicates that non-linear regression could be a better option.

One option in non-linear regression is to fit a polynomial function to model the data. Polynomials are equations that involve powers of the independent variable. The following two functions represent second-degree (quadratic) and third-degree (cubic) polynomial functions:

$$\boldsymbol{Predicted(Y) = b_0 + b_1X + b_2X^2}$$

$$\boldsymbol{Predicted(Y) = b_0 + b_1X + b_2X^2 + b_3X^3}$$

The parameter $\boldsymbol{b_0}$ is the intercept of the regression model and the parameters $\boldsymbol{b_1}, \boldsymbol{b_2}$and $\boldsymbol{b_3}$ are the coefficients of the predictor. These are still simple regression models, as they both use only one predictor. Tools exist in software such as Excel, SAS, etc. to fit a curve in a non-linear fashion given the data. These estimate the values of the coefficients $\boldsymbol{b_0}, \boldsymbol{b_1}$ etc., thus allowing one to predict the increase in sales given a discount %. Figure below shows the same .graph with a curve fitted on the data. That curve has been produced by a quadratic non-linear regression model based on the equations described earlier.



Regression line for retail example second-order regression.

Calculate the coefficient of determination and transform the data to make a linear regression line fit the given data. [E]

Discount (%)	Increase in Sales (%)
10	3.05
15	7.62
20	12.19
25	20.42
30	28.65
35	42.06
40	55.47
45	74.68

50	93.88

Coefficient of determination

Polynomials are equations that involve powers of the independent variable. The following two functions represent second-degree (quadratic) and third-degree (cubic) polynomial functions:

$$\boldsymbol{Predicted(Y) = b_0 + b_1X + b_2X^2}$$

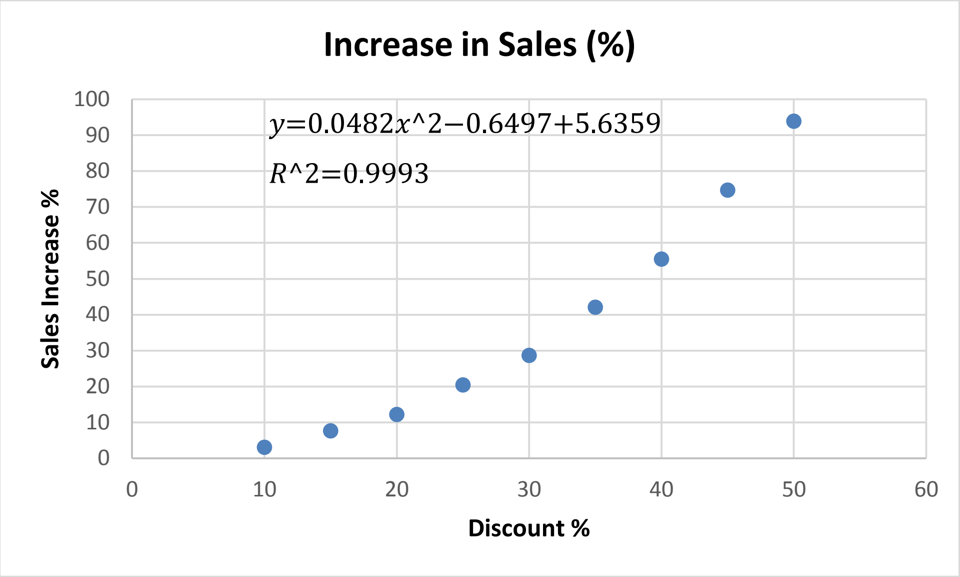
$$\boldsymbol{Predicted(Y) = b_0 + b_1X + b_2X^2 + b_3X^3}$$

The parameter $\boldsymbol{b_0}$ is the intercept of the regression model and the parameters $\boldsymbol{b_1, b_2}$and $\boldsymbol{b_3}$ are the coefficients of the predictor. These are still simple regression models, as they both use only one predictor. Tools exist in software such as Excel, SAS, etc. to fit a curve in a non-linear fashion given the data. These estimate the values of the coefficients $\boldsymbol{b_0, b_1}$ etc., thus allowing one to predict the increase in sales given a discount %.

The above equation can be used for predicting the value of Y. Another useful statistic that is computed is R^2 whose value is found to be 0.9993.

$\boldsymbol{R^2}$ is a useful statistic, known as the coefficient of determination that will be discussed in the next section. Basically, the nearer the value of R^2 to 1, the better the fit produced by the regression line.

In statistics, the coefficient of determination, denoted R^2 or r^2 and pronounced R-squared, is a number that indicates how well the data fits the developed model - a line or a curve. An $R^2$of 1 indicates that the regression model perfectly fits the data while an R^2 of 0 indicates that the model does not fit the data at all. This, therefore, indicates that the above quadratic regression model has produced an excellent fit.



In simple linear regression, r^2 is used instead of R^2. In this case, r^2 is simply the square of the sample correlation coefficient (i.e. *r*) between the outcomes and their predicted values. If it is MLR, R^2 is the square of the coefficient of multiple correlations. In both cases, the coefficient of determination ranges from 0 to 1:

Sum of Squares Regression (SSR):$\sum\left(\hat{\boldsymbol{Y}}_{\boldsymbol{i}} - \bar{\boldsymbol{Y}}_{\boldsymbol{i}}\right)^{\mathbf{2}}$

Sum of Squared Error (SSE):$\sum\left(\hat{\boldsymbol{Y}}_{\boldsymbol{i}} - \boldsymbol{Y}_{\boldsymbol{i}}\right)^{\mathbf{2}}$

Sum of Squared Total (SST):$\sum(\boldsymbol{Y}_{\boldsymbol{i}} - \bar{\boldsymbol{Y}}_{\boldsymbol{i}})^{\mathbf{2}}$

Sum of Squared Total (SST) = SSR + SSE

$$\boldsymbol{R}^{\mathbf{2}} = \mathbf{1} - \frac{\boldsymbol{SSR}}{\boldsymbol{SST}}$$

For the general case, we can use the following equations where $\bar{\boldsymbol{Y}}$ is the mean of the actual values of $\boldsymbol{Y}$. $\hat{\boldsymbol{Y}}_{\boldsymbol{i}}$ is predicted values of $\boldsymbol{Y}_{\boldsymbol{i}}$**.** In the above example, a value of 0.99 for $\boldsymbol{R}^{\mathbf{2}}$ indicates that a quadratic model is a good fit for the data.

Transforming Data

Another preferable way to perform non-linear regression is to try to transform the data in order to make the relationship between the two variables more linear and then use a linear regression model rather than a polynomial one. Transformations aim to make a non-linear relationship between two variables more linear so that it can be described by a linear (rather than non-linear) regression model.

The three most popular transformations are the square root $(\sqrt{\boldsymbol{X}})$, the logarithm $(\log \boldsymbol{X})$ and the negative reciprocal $(-\mathbf{1}/\boldsymbol{X})$. We shall illustrate these transformations using our retail data example. Table 5 shows how the square root transformation has been applied on $\boldsymbol{Y}$. The result of the above transformation can be seen on the graph in Fig. 8. The square root transformation has increased the value of the correlation coefficient from 0.97 to 0.99.

Table 5 Square root transformation of retail data

Discount (%)	**Square Root Sales Increase (%)**
10	1.75
15	2.76
20	3.49
25	4.52
30	5.35
35	6.49
40	7.45
45	8.64
50	9.69

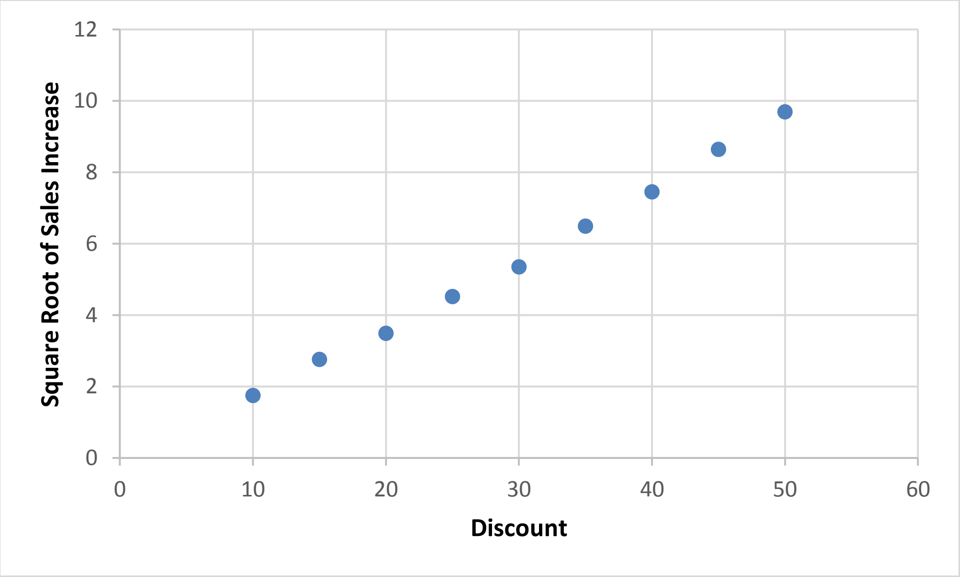


Figure 8 Square root transformation.

In a similar way, logarithmic and negative reciprocal transformations can be performed.

A group of 20 students spent between 0 and 6 hours studying for an examination. How the number of hours does spent studying affect the probability that the student will pass the examination? Use Logistic Regression. [E]

Hours	0.50	0.75	1.00	1.25	1.50	1.75	1.75	2.00	2.25	2.50	2.75	3.00
Pass/ Fail	0	0	0	0	0	0	1	0	1	o	1	o

Hours	3.25	3.50	4.00	4.25	4.50	4.75	5.00	5.50
Pass/ Fail	1	0	1	1	1	1	1	1

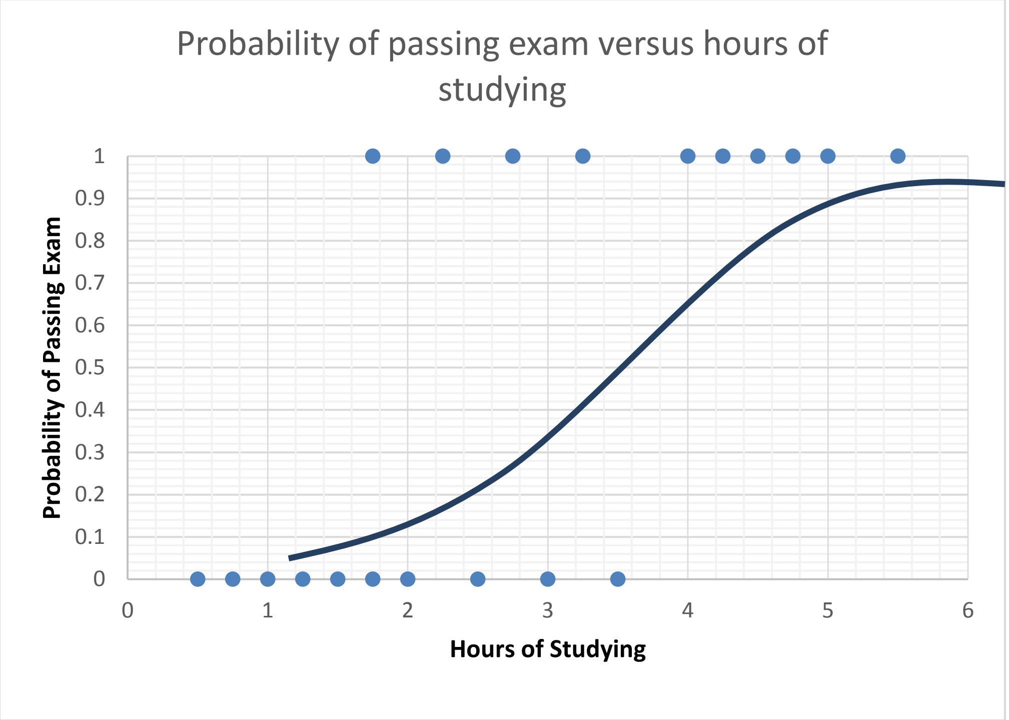
Pass/fail statistics based on study hours.

Logistic Regression

Logistic regression, or logit regression, is a regression model where the dependent variable is categorical. Most common is the case where the dependent variable is binary; where it can take only two values, such as pass/fail, win/lose, alive/dead or healthy/diseased. Cases with more than two categories are referred to as multinomial logistic regression, or, if the multiple categories are ordered, as ordinal logistic regression.

Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution. Logistic regression extends the ideas of MLR to the situation where the dependent variable, $\boldsymbol{y}$, is binary (for convenience, we often code these values as 0 and 1). As with MLR, the independent variables $\boldsymbol{x_1, x_2, \ldots, x_k}$ may be categorical or continuous variables or a mixture of these two types. The model of logistic regression, however, is based on quite different assumptions from those of linear regression. Since the predicted values are probabilities and are therefore restricted to (0, 1), a logistic regression model only predicts the probability of particular outcomes given the values of the existing data.

The graph below shows the probability of passing the examination versus the number of hours of studying, with the logistic regression curve fitted to the data.



Logistic regression curve.

Thus, in logistic regression, we have a binary output variable $\boldsymbol{T}$, and we want to model the conditional probability $\boldsymbol{Pr}(\boldsymbol{Y} = \mathbf{1} \mid \boldsymbol{X} = \boldsymbol{x})$ as a function of $\boldsymbol{x}$; any unknown parameters in the function are to be estimated by maximum likelihood. The following techniques are commonly used:

[1]The simplest idea is to let $\boldsymbol{p}(\boldsymbol{x})$ be a linear function of $\boldsymbol{x}$. Every increment of a component of x. would add or subtract to the probability. The conceptual problem here is that p must be between 0 and 1, and linear functions are unbounded. Moreover, in many situations, we may have the case that changing p by the same amount requires a bigger change in $\boldsymbol{x}$ when $\boldsymbol{p}$ is already large (or small) than when $\boldsymbol{p}$ is close to 1/2. Linear models cannot handle this situation.
[2]The next idea is to let $\boldsymbol{log}\, \boldsymbol{p}(\boldsymbol{x})$ be a linear function of $\boldsymbol{x}$, so that changing an input variable multiplies the probability by a fixed amount. The problem is that logarithms are unbounded in only one direction, and linear functions are not.
[3]Finally, the easiest modification of $\boldsymbol{log}\, \boldsymbol{p}$ which has an unbounded range is the logistic (or logit) transformation, $log(\, p/1 - \, p)$. This can be treated as a linear function of x without any problem. This alternative is a logistic regression.

We have

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + x\beta$$

Solving for p gives us

$$p(x;\beta) = \frac{e^{\beta_0+x\beta}}{1+e^{\beta_0+x\beta}} = \frac{1}{1+e^{-(\beta_0+x\beta)}}$$

Solving the equations using a software tool we have coefficients as $\boldsymbol{\beta_0} = \mathbf{-4.0777}$ and $\boldsymbol{\beta} = \mathbf{1.5046}$. These coefficients are entered in the logistic regression equation to estimate the probability of passing the examination:

[1]Probability of passing the examination $= \mathbf{1/[1 + \boldsymbol{exp}\{-(-4.0777 + 1.5046} \times \boldsymbol{hours})\}]$. For example, for a student who studies 2 h, entering the value hours = 2 in the equation gives the estimated probability of passing the examination of $p = 0.26$.
[2]Probability of passing the examination $= \mathbf{1/[1 + \boldsymbol{exp}\{-(-4.0777 + 1.5046 * 2)\}] = 0.26}$. Similarly, for a student who studies 4 h, the estimated probability of passing the exam is $\boldsymbol{p} = \mathbf{0.87}$.
[3]Probability of passing the examination $= \mathbf{1/[1 + \boldsymbol{exp}\{-(-4.0777 + 1.5046 * 4)\}\} = 0.87}$.

The Table below shows the probability of passing the examination for several hours of studying.

Logistic regression prediction for examination data.

Hours of Study	**Probability of Passing the Examination**
1	0.07
2	0.26
3	0.61
4	0.87
5	0.97

The regression coefficients can also be estimated using maximum likelihood estimation. Unlike linear regression with normally distributed residuals, it is not possible to find a closed-form expression for the coefficient values that maximise the likelihood function, so that an iterative process must be used instead; for example, Newtons method.

5.Explain in detail about the various Multivariate Analysis Techniques with examples. [CO2-L1]

Classical Multivariate Analysis

As the name indicates, MVA comprises a set of techniques dedicated to the analysis of data sets with more than one dependent variable. The MVA is essentially the statistical process of simultaneously analysing multiple independent (or predictor) variables with multiple dependent (outcome or criterion) variables using matrix algebra (most multivariate analyses are correlational).

There exist several applications in data analysis which need MVA. Some of them are listed below:

1.Petrochemical and refining operations, including early fault detection and gasoline blending and optimization.
2.Food and beverage applications, particularly for consumer segmentation and new product development.
3.Agricultural analysis, including real-time analysis of protein and moisture in wheat, barley and other crops.
4.Business intelligence and marketing for predicting changes in dynamic markets or better product placement.

The multivariate data is normally defined in the form of a data matrix - which consists of rows representing each subject and columns representing each variable; however, the matrix can also be a correlation matrix, a variance/covariance matrix, or a sum-of-squares/cross-product matrix.

The matrix is then solved simultaneously through matrix (linear) algebra and yields linear ***composite scores*** which are linear combinations of the variables upon which the final solution is based.

The importance of each variable is determined by its weight (degree of contribution) to its set of variables and is usually indicated by a numerical coefficient.

Some examples for MVA include the following:

Example 1:
A researcher has collected data on three psychological variables, four academic variables (standardised test scores) and the type of educational programme the student is in for 600 high school students. She is interested in how the set of psychological variables is related to the academic variables and the type of programme the student is in.

Example 2:
A doctor has collected data on cholesterol, blood pressure and weight. She also collected data on the eating habits of the subjects (e.g. how many ounces of red meat, fish, dairy products and chocolate consumed per week). She wants to investigate the relationship between the three measures of health and eating habits:

Example 3:
A researcher is interested in determining what factors influence the health of sunflower plants. She collects data on the average leaf diameter, the mass of the root ball and the average diameter of the blooms, as well as how long the plant has been in its current container. For predictor variables, she measures several elements in the soil as well as the amount of light and water each plant receives.

Described below are a set of techniques that can be used to perform multivariate data analysis. Only a brief overview of each technique is presented here.

Multiple regression analysis:

Multiple regression is the most popularly utilised multivariate technique. It examines the relationship between a single dependent variable and two or more metric independent variables. The technique determines the linear relationship with the lowest sum of squared variances. The beta coefficients (weights) are the marginal impacts of each variable, and the size of the weight can be interpreted directly. Multiple regression is often used for prediction.

Logistic regression analysis:

This technique is a variation of multiple regression that allows for the prediction of an event. Here, the dependent variables are binary and the objective is to arrive at a probabilistic assessment of these variables. The independent variables can be either discrete or continuous. This method helps to predict the outcome when two possible choices are present. For example a bank decision system has to select Yes or No to a customer requesting a loan.

Discriminant analysis:

The purpose of discriminant analysis is to correctly classify given data into homogeneous groups. It builds a linear discriminant function, which can then be used to classify the observations. The overall fit is assessed by looking at the degree to which the group means differ and how well the model classifies. This model is useful to categorise people like buyers and non-buyers based on several factors in their profile.

Multivariate analysis of variance (MANOVA):

This technique examines the relationship between several categorical independent variables and two or more metric dependent variables. MANOVA examines the dependence relationship between a set of dependent measures across a set of groups. This analysis is usually used in experimental design, and usually a hypothesised relationship between dependent measures is used. This technique is slightly different in that the independent variables are categorical and the dependent variable is metric.

Factor analysis:

When there are many variables in a data set, it is often helpful to reduce the variables to a smaller set of factors. This is an independent technique, in which there is no dependent variable. In the ideal case, the independent variables are normal and continuous, with at least 3-5 variables loading onto a factor. The sample size should be over 50 observations, with over five observations per variable. There are two main factor analysis methods: common factor analysis, which extracts factors based on the variance shared by the factors, and principal component analysis, which extracts factors based on the total variance of the factors. Common factor analysis is used to look for the hidden underlying factors, whereas principal component analysis is used to find the fewest number of independent variables that explain the most variance.

Cluster analysis:

The purpose of cluster analysis is to reduce a large data set to meaningful subgroups of individuals or objects. The division is accomplished on the basis of similarity of the objects across a set of specified characteristics. Outliers are a problem with this technique, often caused by too many irrelevant variables. The sample should be representative of the population, and it is desirable to have uncorrelated factors. There are three main clustering methods: hierarchical, partitional and a combination of both. There are four main rules for developing clusters: the clusters should be different, they should be reachable, they should be measurable and the clusters should be profitable (big enough to matter) . This is a great tool for market segmentation.

Multidimensional scaling (MDS):

The purpose of MDS is to transform consumer judgements of similarity into distances represented in multidimensional space. This is a decompositional approach that uses perceptual mapping to present the dimensions. As an exploratory technique, it is useful in examining unrecognised dimensions about products and in uncovering comparative evaluations of products when the basis for comparison is unknown.

Correspondence analysis:

This technique provides for dimensional reduction of object ratings on a set of attributes, resulting in a perceptual map of the ratings. However, unlike MDS, both independent variables and dependent variables are examined at the same time. This technique is more similar in nature to factor analysis. It is a compositional technique and is useful when there. are many attributes and many companies. It is most often used in assessing the effectiveness of advertising campaigns. It is also used when the attributes are too similar for factor analysis to be meaningful. The main structural approach is the development of a contingency (crosstab) table. This means that the form of the variables should be non-metric. The model can be assessed by examining the Chi-square value for the model. Correspondence analysis is difficult to interpret, as the dimensions are a combination of independent and dependent variables.

Conjoint analysis:

Conjoint analysis is often referred to as “trade-off analysis”, since it allows for the evaluation of objects and the various levels of the attributes to be examined. It is both a compositional technique and a dependence technique. A utility value is calculated for each level of each attribute, and combinations of attributes at specific levels are summed to develop the overall preference for the attribute at each level. Models can be built that identify the ideal levels and combinations of attributes for products and services.

Canonical correlation:

The most flexible of the multivariate techniques, canonical correlation simultaneously correlates several independent variables and several dependent variables. This powerful technique utilises metric independent variables, unlike MANOVA, such as sales, satisfaction levels and usage levels. It can also utilise non-metric categorical variables.

Structural equation modelling:

Structural equation modelling examines multiple relationships between sets of variables simultaneously. This represents a family of techniques. For example, to compute the placement worthiness of a student we need to infer this value from direct measurement of variables such as test scores, the level of education, grade point average and other related measures. These tools are often used to evaluate many scaled attributes.

Each of the multivariate techniques described above has a specific type of application for which it is best suited. Each technique also has certain strengths and weaknesses that should be clearly understood by the analyst before attempting to interpret the results of the technique. Current statistical packages (SAS, SPSS, S-Plus and others) make it increasingly easy to run a procedure, but expertise is required to analyze and interpret the results to take better decisions.

Bayesian Modelling

6.Write a short note on Bayesian Data Analysis and Explain Bayesian Inference process in detail. [CO2-L1]

Bayesian Modelling, Inference and Bayesian Networks

Bayesian data analysis deals with a set of practical methods for making inferences from the available data. These methods use probability models to model the given data and also predict future values. Thus, the essential feature in Bayesian methods is their explicit use of probability for quantifying uncertainty in inferences based on statistical data analysis.

Bayesian data analysis typically consists of the following three steps:

1.Setting up the prior distribution: Using domain expertise or prior knowledge to develop a joint probability distribution for all parameters of the data under consideration and also the output data (which needs to be predicted). This is termed as the prior distribution,
2.Setting up the posterior distribution: After taking into account the observed'data (given data set) calculating and interpreting the appropriate posterior distribution. This is estimating the conditional probability distribution of the data parameters, given the observed data.
3.Evaluating the fit of the model: How well does the developed model fit the data? Are the conclusions reasonable? How sensitive are the results to the modelling assumptions in step 1? In response, one can alter or expand the model and repeat the three steps.

This section deals with the different aspects of Bayesian data analysis. We first introduce the concept of Bayesian inference, and then give a brief overview of a few Bayesian models for prediction. This section concludes with a discussion of Bayesian networks and how they can be used for inference and prediction.

Bayesian Inference

Bayesian inference is a method of statistical inference in which Bayes' theorem is used to update the probability for a hypothesis as more evidence or information becomes available. Bayesian inference is the process of fitting a probability model to a set of data which results in a probability distribution on the parameters of the model. This model is then extended to new unobserved data and thus makes predictions for new observations.

More formally, we can define this process as follows:

"Bayesian inference derives the posterior probability of any event as a consequence of two factors, a prior probability and a 'likelihood function derived from a statistical model for the observed data." This inference essentially attempts to assess hypotheses by calculating their probabilities $\boldsymbol{P(Hi|)}$ conditional on known and/or presumed information using the rules of probability theory.

The following axioms of probability are particularly useful for Bayesian inference:

'OR' (Sum Rule): $\boldsymbol{P(H_1 \vee H_2|I) = P(H_1|I) + P(H_2|I) - P(H_1, H_2|I)}$

'AND' (Product Rule): $\boldsymbol{P(H_1, D|I) = P(H_1|I)P(D|H_1, I)}$

$= \boldsymbol{P(D|I)P(H_1|D, I)}$

'NOT' $\boldsymbol{P(\overline{H}_1|I) = 1 - P(H_1|I)}$

Further, all Bayesian methods are based on three important theorems in probability:

1.Bayes' theorem (BT).
2.Law of total probability (LTP).
3.Normalization.

Bayes Theorem (BT)

Consider $P(H_i, D_{oba}|I)$ using the product rule:

$P(H_i, D_{oba}|I) = P(H_\mathrm{i}|I)P(D_{oba}|H_i, I)$

$= P(D_{oba}|I)P(H_\mathrm{i}|D_{oba}, I)$

Solve for the posterior probability:

$$P(H_i, D_{oba}|I) = P(H_i|I)\frac{P(D_{oba}|H_i, I)}{P(D_{oba}|I)}$$

Theorem holds for any propositions, but for hypotheses & data the factors have names:

$$Posterior \propto prior\ x\ likelihood$$

$$norm.const. P(D_{oba} \mid I) = prior\ predictive$$

Law of Total Probability (LTP)

Consider exclusive, exhaustive $\{B_i\}$ (I asserts one of them must be true).

$$\sum_i P(A, B_i|I) = \sum_i P(B_i|A, I)P(A|I) = P(A|I)$$

$$= \sum_i P(B_i|I)P(A|B_i, I)$$

If we do not see how to get $P(A \mid I)$ directly, we can find a set $\{B_i\}$ and use it as a *“basis”*—extend the conversation:

$$P(A|I) = \sum_i P(B_i|I)P(A|B_i, I)$$

If our problem already has B_i in it, we can use LTP to get $P(A|I)$ from the joint probabilities—marginalization:

$$P(A|I) = \sum_i P(A, B_i|I)$$

Normalization

For exclusive, exhaustive

$$H_i, \sum_i P(\,H_i|\dots) = 1$$

Bayesian inference computes the posterior probability according to Bayes’ theorem:

$$P(H|E) = \frac{P(H) \times P(E|H)}{P(E)}$$

Where

1.$\boldsymbol{H}$ stands for any hypothesis whose probability may be affected by data (called 'evidence below). Often there are competing hypotheses from which one must choose the most probable.
2.The evidence $\boldsymbol{E}$ corresponds to new unseen data which was not used in computing the prior probability.
3.$\boldsymbol{P(H)}$, the prior probability, is the probability of $\boldsymbol{H}$ before $\boldsymbol{E}$ is observed. This indicates a previous estimate of the probability that a hypothesis is true, before seeing the current evidence.
4.$\boldsymbol{P(H \mid E)}$, the posterior probability, is the probability of $\boldsymbol{H}$ given $\boldsymbol{E}$, that is, after $\boldsymbol{E}$ is observed.
5.$\boldsymbol{P(E \mid H)}$ is the probability of observing E given H. As a function of H with E fixed, this is the likelihood. It indicates the compatibility of the evidence with the given hypothesis.
6.$\boldsymbol{P(E)}$ is sometimes termed the marginal likelihood or “model evidence”. This factor is the same for all possible hypotheses being considered. (This can be seen by the fact that the hypothesis H does not appear anywhere in the symbol, unlike for all the other factors.) This means that this factor does not enter into determining the relative probabilities of different hypotheses.

When we need to compare different hypotheses, only the factors in the numerator are important, that is, P(H) and P(E | H). Thus, we have

3.*(more precisely)* The posterior probability of a hypothesis is determined by a combination of the inherent likeliness of a hypothesis (the prior) and the compatibility of the observed evidence with the hypothesis (the likelihood).
4.*(more concisely)* Posterior is proportional to likelihood times prior.

Note that Bayes’ rule can also be written as follows:

$$P(H|E) = (P(H) * P(E|H)) * \alpha$$

where $\boldsymbol{\alpha = 1/P(E)}$ is a normalizing constant ensuring that $\boldsymbol{P(H\,|E)}$ sums to 1 for each state of $\boldsymbol{E}$.

7.Explain Naïve Bayes Classifier. Use the data give below to indicate the probability of a player who enjoys playing a sport [CO2-L3]

Outlook	Temperature	Humidity	Windy	Enjoys playing

Sunny	Hot	High	False	N
Sunny	Hot	High	True	N
Overcast	Hot	High	False	P
Rain	Mild	High	False	P
Rain	Cool	Normal	False	P
Rain	Cool	Normal	True	N
Overcast	Cool	Normal	True	P
Sunny	Mild	High	False	N
Sunny	Cool	Normal	False	P
Rain	Mild	Normal	False	P
Sunny	Mild	Normal	True	P
Overcast	Mild	High	True	P
Overcast	Hot	Normal	False	P
Rain	Mild	High	True	N

Bayesian Model - Naive Bayes Classifier (NBC)

Naive Bayes classifiers (NBC) are a family of simple probabilistic classifiers based on applying Bayes theorem with strong (naive) independence assumptions between the features. Naive Bayes has been extensively used for many data analytics applications. It was first used in the context of text categorization and in spite of being a simple model with naive assumptions, it is a popular (baseline) method for text categorization, the problem of judging documents as belonging to one category or the other (such as spam or legitimate, sports or politics, etc.) with word frequencies as the features. It is also a popular model in the field of automatic medical diagnosis.

NBCs are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of models.

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all NBCs assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, consider the job of computing the probability of a final year undergraduate landing a job (indicate Yes/No with probability). It is reasonable to assume that some features of the student will include grades, soft skills, aptitude, project skills, etc. An NBC considers each of these features to contribute independently to the probability that the student will get a job, regardless of any possible correlations between the grades, soft skills, etc. This assumption is known as class conditional independence and since this is unlikely to be true always, this model is termed Naive.

An advantage of Naive Bayes is that it only requires a small amount of training data to estimate the parameters necessary for classification.

The Model

Consider each data instance to be an ***n-dimensional*** vector of attribute values (i.e. features):

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

Given $\boldsymbol{m}$ classes $C_1, C_2, \dots \dots, C_m$ a data instance $\boldsymbol{X}$ is assigned to the class for which it has the greatest posterior probability, conditioned on $\boldsymbol{X}$, that is, $\boldsymbol{X}$ is assigned to $\boldsymbol{C_i}$ if and only if

$\boldsymbol{P(C_i|X)} > P\left(\boldsymbol{C_j}\middle|\boldsymbol{X}\right)\forall \boldsymbol{j}$ such that $\boldsymbol{1 \leq j \leq m, j \neq i}$

According to Bayes theorem

$$P(C_i|X) = \frac{P(C_i) \times P(X|C_i)}{P(X)}$$

Since $\boldsymbol{P(X)}$ is constant for all classes, only the numerator $\boldsymbol{P(X|C_i)P(C_i)}$ needs to be maximized. If the class probabilities $\boldsymbol{P(C_i)}$ are not known, they can be assumed to be equal, so that we need only maximize $\boldsymbol{P(X|C_i)}$. But with the existence of a data set, we can estimate $\boldsymbol{P(C_i)}$ from the proportions in some training sample of the given data set.

In general, the computation for evaluating $\boldsymbol{P(X|C_i)}$ is intractable and very expensive. If each component X_k can have one of c values, there are C_n possible values of $\boldsymbol{X}$ to consider.

Chain rule for conditional independence.

$$P(C_k, x_1, \dots, x_n) = P(C_k)P(x_1, \dots, x_n|C_k)$$

$$= P(C_k)P(x_1|C_k)P(x_2, \dots, x_n|C_k, x_1)$$

$$= P(C_k)P(x_1|C_k)P(x_2|C_k, x_1)P(x_3, \dots, x_n|C_k, x_1, x_2)$$

$$= P(C_k)P(x_1|C_k)P(x_2|C_k, x_1) \dots P(x_n, \dots, x_n|C_k, x_1, x_2, x_3, \dots, x_{n-1})$$

Consequently, the assumption of class conditional independence is often made, giving

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i)$$

The value $P(x_1|C_i), \dots, P(x_n, C_i)$ can be estimated from the training sample. The NBC combines this model with a decision rule. One common rule is to pick the hypothesis that is most probable; this is known as the maximum a posteriori or MAP decision rule. The corresponding classifier, a Bayes classifier, is the function that assigns a class label C_k for some k as follows:

$$ArgMax(k \in 1,2,\dots,k) = P(C_k) \prod_{k=1}^{n} P(x_k|C_i)$$

1.If i^{th} attribute is categorical: $\boldsymbol{P(x_i|C)}$ is estimated as the relative frequency of samples having value X_i; as i^{th} attribute in class C.
2.If i^{th} attribute is continuous: $\boldsymbol{P(x_i|C)}$ is estimated through a Gaussian density function.

We shall illustrate the process using an example.

Table below indicates the computation of the various conditional probabilities. We have

$$\boldsymbol{P(P) = 9/14}$$

$$\boldsymbol{P(N) = 5/14}$$

Here, P stands for Yes and N stands for No.

Computation of various probabilities

Outlook	
$P(Sunny)P) = 2/9$	$P(Sunny|N) = 3/5$
$P(Overcast|P) = 4/9$	$P(Overcast|N) = 0$
$P(Rain|P) = 3/9$	$P(Rain|N) = 2/5$
Temperature	
$P(Hot|P) = 2/9$	$P(Hot|N) = 2/5$
$P(Mild|P) = 4/9$	$P(Mild|N) = 2/5$
$P(Cool|P) = 3/9$	$P(Cool|N) = 1/5$
Humidity	
$P(High|P) = 3/9$	$P(High|N) = 4/5$
$P(Normal|P) = 6/9$	$P(Normal|N) = 2/5$
Windy	

$P(True\|P) = 3/9$	$P(True\|N) = 3/5$
$P(False\|P) = 6/9$	$P(False\|N) = 2/5$

Now, let us say we want to predict “Enjoy Sport” on a day with the following conditions:

< Outlook = sunny; temparature = cool;

Humidity = high; windy = strong >

This means, we want to compute $(P(v|o = sunny, t = cool, h = high, w = strong))$ using the training data. Now, we have

$$\frac{\#\ days\ of\ enjoying\ sports\ with\ strong\ wind}{\#\ days\ of\ enjoying\ sport}$$

$$p(P)p(sun \mid P)\ p(cool \mid P)\ p\ (high \mid P)p(strong \mid P) = .005$$

$$p(N)p(sun \mid N)\ p(cool \mid N)\ p(high \mid N)p(strong \mid N) = .021$$

Since we have probability of No to be larger according to the MAP hypothesis, we can predict “Enjoy Sport” to be No on that day.

In summary, we have Naive Bayes model with the following advantages:

1.Training is very easy and fast; it just requires considering each attribute in each class separately.
2.Test is straightforward; just looking up tables or calculating conditional probabilities with normal distributions.
3.Performance is competitive with most of the state-of-the-art classifiers even in the presence of violating independence assumption.
4.Many successful applications, for example, spam mail filtering, sentiment analysis, news article categorization, etc.

The major disadvantage is that the NBC makes the often erroneous assumption that all features are independent to each other. This condition is addressed in the next section, where we look at a Bayes network model.

8.Illustrate the purpose of using Bayesian Belief Network in Data Analysis. [CO2-L1]

Bayesian Networks

Naive Bayes assumption of conditional independence is too restrictive. However, computing all joint probabilities without some flexibility is intractable. A Bayesian belief network (BBN) (Bayesian net) describes conditional independence among subsets of variables (attributes). It combines prior knowledge about dependencies among variables using the given training data.

Thus, BBNs is a compromise model which has the following features:

1.Specifies which conditional independence assumptions are valid.
2.Provides sets of conditional probabilities to specify the joint probability distributions wherever dependencies exist.

BBNs achieve these objectives by

1.Specifying the conditional independence assumptions in a directed acyclic graph.
•Each node denotes a variable.
•Each arc indicates a dependency between the nodes at its start and finish.
•Consequently, a variable X is conditionally independent of variable Y, given the immediate predecessors of X, if no path exists from X to Y.
2.A conditional probability table (CPT) is provided for each node.
•This specifies the probability distribution of the associated variable given the values of the nodes immediate predecessors.
3.More specifically, belief networks represent dependence between variables, and they give a concise specification of the joint probability distribution - by making conditional independence relationships explicit. A belief network $\boldsymbol{B} = (\boldsymbol{V}, \boldsymbol{E})$ is a directed acyclic graph with a set of nodes $\boldsymbol{V}$ and a set of directed edges $\boldsymbol{E}$ where:
•Each node in $\boldsymbol{V}$ corresponds to a random variable.
•There is a directed edge from node $\boldsymbol{X}$ to node $\boldsymbol{Y}$ if variable $\boldsymbol{X}$ has a direct (causal) influence on variable $\boldsymbol{Y}$
•Each node in $\boldsymbol{V}$ has a CPT associated with it. The CPT specifies the conditional distribution of the node given its parents, that is, $\boldsymbol{P}(\boldsymbol{x_i}|\boldsymbol{Parents}(\boldsymbol{x_i}))$.
•The parents of a node $\boldsymbol{X}$ are all those nodes that have arrows pointing to node $\boldsymbol{X}$.

Every entry in the full joint probability distribution can be calculated from the belief network alone. We will illustrate the process of using a BBN with the help of an example. Figure 16 depicts a typical BBN. A student whenever studied hard is highly likely to get good grades. Sometimes, but not very often, a student who is lucky will also get good grades. A student with good grades can get a good job. Independently, the student can go for higher studies too.

The following dependencies can be identified from the BBN:

1.Studying hard and being lucky are independent.
2.Getting good grades is dependent both on studying hard and to a lesser extent on being lucky.
3.Getting a job and going in for higher studies are conditionally independent given good grades.
4.Further, getting a job and going in for higher studies are only indirectly dependent on either studying hard or being lucky.

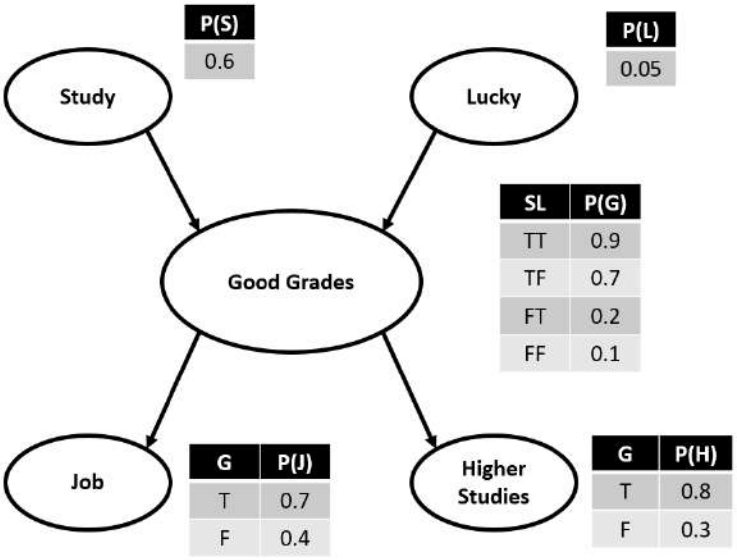
Further, the BBN also indicates the conditional distributions. In Fig. 16, each distribution is shown as a CPT. This form of table is used for discrete variables. Each row in a CPT contains the conditional probability of each node value for a conditioning case. A conditioning case is simply a combination of values for the parent nodes and can be seen as a miniature atomic event. The sum of each row must be 1, because the entries represent an exhaustive set of cases for the variable. For Boolean variables, we know that if the probability of a true variable is p and the probability of a false value is $\mathbf{1-p}$. For reasons of efficiency, we have omitted the second value in the CPTs.

Thus, in the figure, we have in the table for the job: $\boldsymbol{p(J)}$ when $\boldsymbol{g(good\,grades\,is\,true) = 0.8}$; we can infer $\boldsymbol{P(Not\,Job)}$ when $\boldsymbol{G}$ is false to be $\mathbf{1 - 0.8 = 0.2}$. In a similar fashion, in the CPT associated with good grades, we have

$\boldsymbol{P(good\ grades\ (G)\ when\ study\ hard\ (S)\ is\ true\ and\ lucky\ (L)\ is\ true)}$ to be 0.9

we have the complement $\boldsymbol{P(not\ good\ grades)}$ as 0.1. To make the picture clearer, we have omitted these negative probabilities. The rest of the entries can be similarly interpreted.

For these five variables binary in nature, we would have needed the computation of $\mathbf{2^5 = 32}$ joint probabilities but with the help of the BBN we need only 10 different probabilities.



P(S)
0.6

P(L)
0.05

SL	P(G)
TT	0.9
TF	0.7
FT	0.2
FF	0.1

G	P(J)
T	0.7
F	0.4

G	P(H)
T	0.8
F	0.3

An example BBN.

The purpose of the BBN model is that we can determine the posterior probability distribution for a set of query variables given a set of observed events. We achieve the posterior probability distribution by using inference within the network. Any entry in the joint probability distribution can be calculated from the belief network CPTs. For example, consider:

$$\boldsymbol{P(J,H,G,\neg S,\neg L) = P(J|H,G,\neg S,\neg L)P(H,G,\neg S,\neg L)}$$

$$\boldsymbol{= P(J|G)P(H|G,\neg S,\neg L)P(G,\neg S,\neg L)}$$

$$\boldsymbol{= P(J|G)P(H|G)P(G|\neg S,\neg L)P(\neg S,\neg L)}$$

$$\boldsymbol{= P(J|A)P(H|G)P(G|\neg S,\neg L)P(\neg S)P(\neg L)}$$

We are attempting to evaluate the probability of the event of a student getting a job; also a set for higher studies with good grades when he does not study hard nor is he lucky.

1.The first step/line is done using the chain rule.
2.The second step first applies conditional independence to change $P(J|H,G,\neg S,\neg L)$ to $P(J|G)$.
3.Also, another rule is applied to line 2 as well. This is the chain rule, which states that $\boldsymbol{P(H,G,\neg S,\neg L) = P(H|G,\neg S,\neg L)P(\neg S,\neg L)}$,
4.Similarly, on the third line, both conditional independence and the chain rule are applied.

5.On the last (fourth) line, $\boldsymbol{P(\neg S, \neg L)} = \boldsymbol{P(\neg S)P(\neg L)}$ because they are assumed to be absolutely independent.
6.Note that all of the probabilities on the last line of the proof can be looked up in the CPTs associated with the nodes in the belief network:

$$P(X_1 = x_1, \dots, X_n = x_n) = P(X_1)\prod_{i=2}^{n} P(\, X_i = x_i | Parents(X_i))$$

Two important results are used to perform inference in a BBN:

First is the chain rule, which is derived from repeated applications of the product rule, specific for a Bayesian network (because of the use of the term “$\boldsymbol{Parents}()$”) . It assumes conditional independence of siblings and ancestors, given parents. It also assumes node $\boldsymbol{X_1}$has no parents (it is at the “top” of the network).

The other result is a lemma which can be used to introduce new variables when there is no apparent dependency among the variables. For instance, if A and Care independent in the presence of B, we can add B to the computation as in the following equation:

$$\boldsymbol{P(A|C) = P(A \mid B \wedge C)P(B \mid C) + P(A|\neg B \wedge C)P(\neg B \mid C)}$$

The other important issue in a BBN is learning and constructing the BBN. Methods for learning Bayesian networks are an active area of current research.

The following two situations explain the learning of Bayesian networks:

1.The structure of the network is known: only the CPTs must be learned.
2.The structure of the network is unknown and must be learned.

If the network structure is known and given to us by a domain expert then it is a very easy to learn the probabilities. Use the data set giving values for all the attributes, simply compute the entries for the CPTs by counting, just as for NBC.

If data is absent for some variables, then the job is considerably more difficult. The space of possible probability assignments must be searched for the one that maximizes the likelihood of the available data.

Learning the structure of a Bayesian network can be considered a specific example of the general problem of selecting a probabilistic model that explains a given set of data. Although this is a difficult task, it is generally considered an appealing option, as constructing a structure by hand might be hard or even impossible if the dependent variables are not known by domain experts.

Support Vector Machines

9.Explain the concept and use of Support Vector Machines [CO2-L1]

Support Vector Machines and Kernel Methods

One major area in data analytics is “pattern analysis”. The general task of pattern analysis is to find and study general types of relations (e.g., clusters, rankings, principal components, correlations and classifications) in data sets. One popular tool for performing such pattern analysis is the “support vector machine” (SVM).

SVMs are supervised learning models with associated learning algorithms that analyze the data used for classification and regression analysis. Given a set of labelled training examples (data points), each belonging to one of two categories, an SVM training algorithm builds a model that is capable of assigning a category value to any new example. This makes SVMs a non-probabilistic binary linear classifier.

An SVM model represents the given data as points in space, mapped so that the examples of separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the “kernel trick”, implicitly mapping their inputs into high-dimensional feature spaces.

In the case of unlabelled data, when supervised learning is not possible, we need an unsupervised learning method that can find natural clusters of data points. This clustering mechanism should be able to map new data points to any of the clusters formed. SVMs can be extended to what is called support vector clustering (SVC) and this is commonly used in industrial applications either when data is not labelled or when only some data is labelled as a pre-processing for a classification pass. Thus, we see that SVMs are a very versatile tool for several types of data analysis.

In this section, we give an overview of the general SVMs and also discuss a few kernel-based extensions to the basic SVM.

Importance of SVM

SVMs became particularly popular when they were first used in handwriting recognition and produced results comparable to neural networks. SVMs can be used to solve various real-world problems:

1.SVMs are helpful in text and hypertext categorization as their application can significantly reduce the need for labelled training instances.
2.Classification of images can also be performed using SVMs. Experimental results show that SVMs achieve significantly higher search accuracy than traditional query refinement schemes after just three to four rounds of relevance feedback.
3.SVMs are also useful in medical science to classify proteins with up to 90% of the compounds classified correctly.

SVMs follow the concepts of statistical learning theory. In statistical learning theory, the problem of supervised learning is formulated as follows:

1.We are given a set of training data $\{(\boldsymbol{x_1}, \boldsymbol{y_1}) \ldots (\boldsymbol{x_n}, \boldsymbol{y_n})$ in $\boldsymbol{R^n} \times \boldsymbol{R}$.
2.This data is sampled according to unknown probability distribution $\boldsymbol{P}(\boldsymbol{x}, \boldsymbol{y})$, and a loss function $\boldsymbol{V}(\boldsymbol{y}, \boldsymbol{f}(\boldsymbol{x}))$.
3.The loss function measures the error when for a given $\boldsymbol{x}$, $\boldsymbol{f}(\boldsymbol{x})$ is “predicted” instead of the actual value $\boldsymbol{y}$.
4.The problem consists in finding a function $\boldsymbol{f}$ that minimizes the expectation of the error on new data that is, finding a function $\boldsymbol{f}$ that minimizes the expected error:

$$\int V\big(y, f(x)\big)P(x, y)dx\, dy$$

5.In statistical modelling, we would choose a model from the hypothesis space, which is closest (with respect to some error measure) to the underlying function in the target space.

10.Explain briefly about SVM and how does it construct a hyperplane. [CO2-L1]

Support Vector Machine

Classification is a popular data analytics task. In classification, we are given a large set of data points, where each data point is associated with one of the two classes, and the goal is to decide which class a new data point will be associated.

Example

We are given a large data set made up of student records in a university for the past 15 years. These records contain student data such as grades, language skills, programming skills, etc. Each student data point is associated with one status attribute which indicates whether the student was successful in the college placement drive. “Yes” indicates the student acquired a job in the placement drive;

“No” indicates the student could not get a job in the placement drive. Classification builds a model over this data which can then be used to predict the placement outcome for a new unknown student.

SVMs, as a classification method, view a data point as a $\boldsymbol{p}$-dimensional vector (a list of $\boldsymbol{p}$ attributes associated with each data point), and attempts to separate such points using a $(\boldsymbol{p} - \mathbf{1})$ dimensional hyperplane. This is called a linear classifier. There are many hyperplanes that might classify the data. One reasonable choice as the best hyperplane is the one that represents the largest separation, or margin, between the two classes. Hence, we choose that hyperplane such that the distance from it to the nearest data point on either side is maximized. If such a hyperplane exists, it is known as the maximum-margin hyperplane and the linear classifier it defines is known as a ***maximum-margin classifier***. The ability of a model to correcdy classify data not in the training set is known as its generalization.

More formally, an SVM constructs a hyperplane or set of hyperplanes in a high- or infinite dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data point of any class (so-called functional margin), since in general the larger the margin, the lower the generalization error of the classifier. Given a data set of points, we want to know which the best linear classifier of the type is

$$f(\vec{x}) = \vec{W}^{T}x + b$$

$$= (w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_nx_n + b)$$

which best explains the data point most accurately. While there can be an infinite number of hyperplanes that achieve 100% accuracy on training data, for generalization, we need the hyperplane that is optimal with respect to the accuracy of test data. Figure 17 shows a few possible hyperplanes that can separate the given data into two different classes.

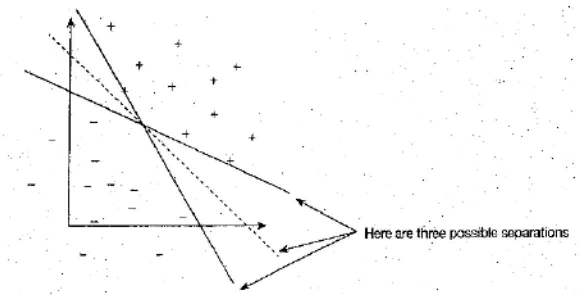


Figure 17 Choice of separating hyperplanes.

The common sense solution indicates that we should increase the gap (margin) between positive and negative cases as much as possible. The best linear classifier is the hyperplane in the middle of the gap (see Fig. 18).

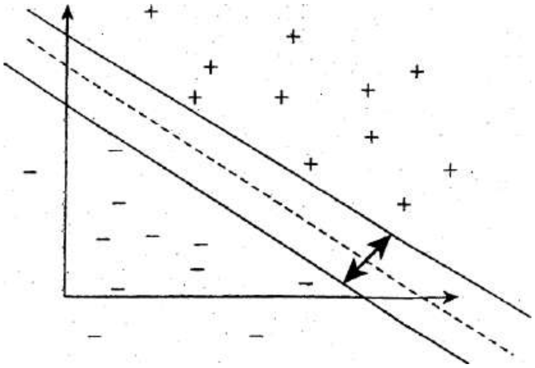


Figure 18 Middle hyperplane is the best separator.

An interesting question is why we need the maximum margin? The main reason is for better generalization. By keeping the maximum margin, even if we have made a small error in the location of the boundary, this gives us the least chance of causing a misclassification. The other advantage would be avoiding local minima and better classification performance.

11.Provide Mathematical Description for SVM and illustrate with suitable figures. [CO2-L1]

Mathematical Description

The following is the mathematical detail for a linear SVM:

We are given a training data set of n points of the form

$$(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$$

where the $\boldsymbol{y_i}$s are either +1 or -1, each indicating the class to which the point $\boldsymbol{x_i}$, belongs. Each $\boldsymbol{x_i}$, is a p-dimensional real vector. We want to find the "maximum-margin hyperplane" that divides the group of points $\vec{\boldsymbol{x}}_{\boldsymbol{i}}$, for which $\boldsymbol{y_i = 1}$, from the group of points for which $\boldsymbol{y_i = -1}$. This is defined so that the distance between the hyperplane and the nearest point $\vec{\boldsymbol{x}}_{\boldsymbol{i}}$ from either group is maximized.

Any hyperplane can be written as the set of points $\vec{\boldsymbol{x}}_{\boldsymbol{i}}$ satisfying $\vec{\boldsymbol{w}}\vec{\boldsymbol{x}} - \boldsymbol{b} = \boldsymbol{0}$. Let us say $\boldsymbol{f(x)}$ is the classification model. Given $\boldsymbol{f(x)}$, the classification is obtained as

$$\hat{y} = sign(f(x)) = \begin{cases} +1\,, f(x) > 0 \\ -1, f(x) < 0 \end{cases}$$

Note: Different $\boldsymbol{w}$ and $\boldsymbol{b}$ can result in identical classification. For example, we can apply any scalar $\boldsymbol{\alpha}$ such that

$$\hat{y} = sign(\alpha(W^T X + b))$$

$$= sign(W^T X + b)$$

Therefore, there are many identical solutions. Thus, in an SVM, we need to find $\boldsymbol{f(x)} = (\boldsymbol{w^T x} + \boldsymbol{b})$ with maximal margin, such that for points closest to the separating hyperplane, we have

$|\boldsymbol{w^T x_i} + \boldsymbol{b}| = \boldsymbol{1}$ (also called the support vectors)

and for other points:

$|\boldsymbol{w^T x_i} + \boldsymbol{b}| > 1$

Figure below illustrates this process.

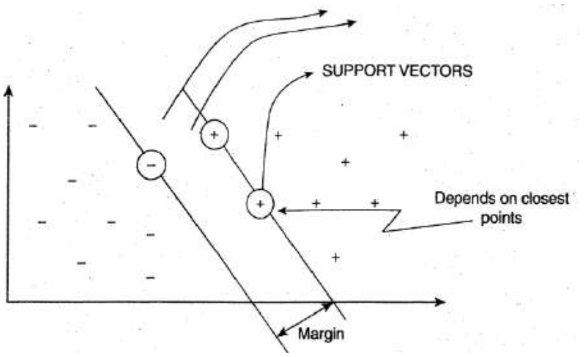
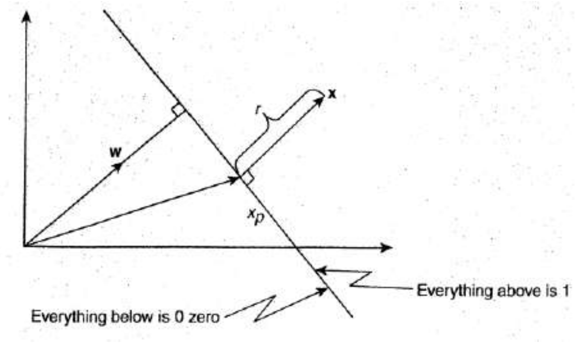


Illustration of SVM.

We need to calculate the length of the margin as a function of $\boldsymbol{w}$. The following diagram shows a point $\boldsymbol{x}$ and its projection $\boldsymbol{X_p}$ to the separating hyperplane. $\boldsymbol{r}$ is defined as the distance between data point $\boldsymbol{x}$ and the hyperplane.



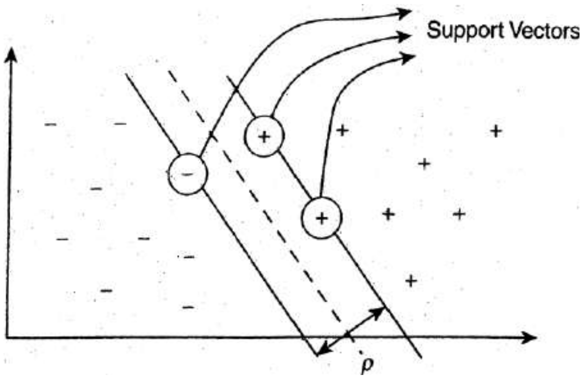
Computing the margin.

Note that $\boldsymbol{w}$ is a vector perpendicular to the hyperplane, so we have

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}\left(\boldsymbol{x_p} + \frac{\boldsymbol{W}}{\|\boldsymbol{w}\|}.\boldsymbol{r}\right)$$

$$= \boldsymbol{W^T X_p} + \boldsymbol{W^T}\frac{\boldsymbol{W}}{\|\boldsymbol{W}\|}\boldsymbol{r} + \boldsymbol{b}$$

$$= \|\boldsymbol{W}\|.\boldsymbol{r} \text{ since } \boldsymbol{W^T X_p} + \boldsymbol{b} = \boldsymbol{0}$$



Calculating the margin.

Therefore

$$r = \frac{f(x)}{\|W\|}$$

Figure 21 illustrates this. Now, solving for margin length $\boldsymbol{\rho}$:

$$\boldsymbol{\rho} = \frac{\boldsymbol{f(x_{\oplus})}}{\|\boldsymbol{W}\|} - \frac{\boldsymbol{f(x_{\Theta})}}{\|\boldsymbol{W}\|} = \frac{\mathbf{2}}{\|\boldsymbol{W}\|}$$

Conclusion: Maximizing the margin is equivalent to minimizing $\|\boldsymbol{W}\|$ (since we can ignore the constant 2 above).

12.Explain the learning problem using SVM in detail [CO2-L1]

Support Vector Machines: Learning Problem

Assuming a linearly separable data set, the task of learning coefficients $\boldsymbol{w}$ and $\boldsymbol{b}$ of SVM $\boldsymbol{f(x)} = (\boldsymbol{w^T x_i + b})$ reduces to solving the following constrained optimization problem:

Find $\boldsymbol{w}$ and $\boldsymbol{b}$ that minimize $\frac{\mathbf{1}}{\mathbf{2}}\|\boldsymbol{W}\|^2$ subject to constraints: $\boldsymbol{y_i(w^T x_i + b) \geq 1},\ \forall \boldsymbol{i}$

This optimization problem can be solved by using the Lagrangian function defined as

$\boldsymbol{L(w,b,\alpha)} = \frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{W^T W} - \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \boldsymbol{\alpha_i[y_i(W^T x_i + b) - 1]}$ such that $\boldsymbol{\alpha_i \geq 0}, \forall \boldsymbol{i}$

where $\boldsymbol{\alpha_1, \alpha_2, \dots, \alpha_N}$ are Lagrange multipliers and $\boldsymbol{\alpha} = [\boldsymbol{\alpha_1, \alpha_2, \dots, \alpha_N}]^{\boldsymbol{T}}$. The solution of the original constrained optimization problem is determined by the saddle point of $\boldsymbol{L(w,b,\alpha)}$ which has to be minimized with respect to $\boldsymbol{w}$ and $\boldsymbol{b}$ and maximized with respect to $\boldsymbol{\alpha}$. Without going into too many details that would be beyond the scope of this book we have the following results:

1.If $\boldsymbol{y_i(w^T x_i + b)} > 1$, the value of $\boldsymbol{\alpha_i}$ that maximizes $\boldsymbol{L(w,b,\alpha)}$ is $\boldsymbol{\alpha_i = 0}$.
2.If $\boldsymbol{y_i(w^T x_i + b)} < 1$, the value of $\boldsymbol{\alpha_i}$ that maximizes $\boldsymbol{L(w,b,\alpha)}$ is $\boldsymbol{\alpha_i} = +\infty$. However, since $\boldsymbol{w}$ and $\boldsymbol{b}$ are trying to minimize$\boldsymbol{L(w,b,\alpha)}$, they will be changed in such a way to make $\boldsymbol{y_i(w^T x_i + b)}$ at least equal to $+\mathbf{1}$.
3.We have $\boldsymbol{\alpha_i\{y_i(w^T x_i + b) - 1\} = 0},\ \forall \boldsymbol{i}$ where data points $\boldsymbol{x_i}$ with $\boldsymbol{\alpha_i} > 0$ are called the support vectors.
4.The necessary conditions for the saddle point of $\boldsymbol{L(w,b,\alpha)}$ are

$$\frac{\boldsymbol{\partial L}}{\boldsymbol{\partial w_j}} = \mathbf{0}, \forall \boldsymbol{j}$$

$$\frac{\boldsymbol{\partial L}}{\boldsymbol{\partial \alpha_j}} = \mathbf{0}, \forall \boldsymbol{j}$$

or stated a different way,$\Delta_w L = 0, \Delta_\alpha L = 0$.

Solving for the necessary conditions results in

$$W = \sum_{i=1}^{N} \alpha_i y_i x_i \; ; \sum_{i=1}^{N} \alpha_i y_i = 0$$

By replacing $\boldsymbol{\sum_{i=1}^{N} \alpha_i y_i x_i}$; into the Lagrangian function and by using $\boldsymbol{\sum_{i=1}^{N} \alpha_i y_i = 0}$ as a new constraint the dual optimization problem can be constructed as:

Find $\boldsymbol{\alpha}$ that maximizes $\boldsymbol{\sum_i \alpha_i - \frac{1}{2} \sum_i \sum_i \alpha_i \alpha_j y_i y_j x_j{}^T \, x_j}$

subject to $\boldsymbol{\sum_{i=1}^{N} \alpha_i y_i = 0 \, , \alpha_i \geq 0, \forall i}$

This equation is a convex quadratic programming problem, so there is a global minimum. There are a number of optimization routines capable of solving this optimization problem. These are beyond the scope of this book.

In summary, to use the SVM as a predictor, we have the following:

Given the values $\boldsymbol{\alpha_1, \alpha_2, \ldots, \alpha_N}$obtained by solution of the dual problem, the final SVM predictor can be expressed as

$$f(x) = W^T X_i + b = \sum_{i=1}^{N} \alpha_i y_i X_i{}^T \, X + b$$

where

$$b = \frac{1}{\left|I_{support}\right|} \sum_{i \in I_{support}} \left(y_i - \sum_i \alpha_j y_j X_j{}^T \, X_i \right)$$

and I_{support} is the set of support vectors.

Note:

1.To obtain the prediction, all data points from the training data are consulted.
2.Since $\boldsymbol{\alpha_i \neq 0}$ only for the support vectors, only support vectors are used in giving a prediction.
3.$X_j{}^T X$ is a scalar.

13.Explain how the SVM handles Linearly Separable and Non-Separable cases [CO2-L1]

Support Vector Machine

Support vector machines (**SVMs**, also called as **support vector networks**) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on

Support Vector Machine: Linearly Non-Separable Case

So far we have discussed the construction of SVMs on linearly separable training data. This is a very strong assumption that is unrealistic in most real-life applications. Figure 22 illustrates an example of data points that are not linearly separable.

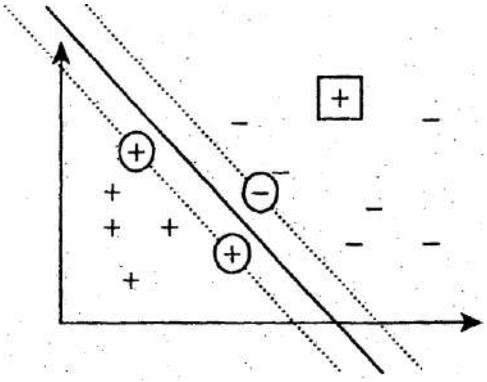


Figure 22 Illustrating non-linearly separable case.

Solution: Introducing the slack variables $\boldsymbol{\xi_i, i = 1, 2, \ldots, N}$ to relax the constraint $\boldsymbol{y_i(w^T x_i + b)} \geq \mathbf{1}$ to $\boldsymbol{y_i(w^T x_i + b)} \geq \mathbf{1} - \boldsymbol{\xi_i, \xi_i} \geq \mathbf{0}$. Ideally, one would prefer all slack variables to be zero and this would correspond to the linearly separable case. Therefore, the optimization problem for construction of SVM on linearly non-separable data is defined as

Find $\boldsymbol{w}$ and $\boldsymbol{b}$ that minimize $\frac{\mathbf{1}}{\mathbf{2}}\|\boldsymbol{W}\|^{\mathbf{2}} + \boldsymbol{C}\sum_{\boldsymbol{i}} \boldsymbol{\xi_i}^{\mathbf{2}}$

subject to $\boldsymbol{y_i(w^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0, \forall i}$

where $\boldsymbol{C} > 0$ is an appropriately selected parameter. The additional term $C\sum_i \boldsymbol{\xi_i}^{\boldsymbol{2}}$ enforces all slack variables to be as close to zero as possible.

Dual problem: As in the linearly separable problem, this optimization problem can be converted to its dual problem:

Find $\boldsymbol{\alpha}$ that maximizes

$$\sum_i \boldsymbol{\alpha_i} - \frac{\boldsymbol{1}}{\boldsymbol{2}}\sum_i \sum_j \boldsymbol{\alpha_i \alpha_j y_j y_j X_i}^{\boldsymbol{T}} \boldsymbol{X_j}$$

subject to

$$\sum_{i=1}^{N} \boldsymbol{\alpha_i y_i = 0}, \ \ \boldsymbol{0 \leq \alpha_i \leq C}, \ \ \forall \boldsymbol{i}$$

The parameter helps to constrain the range of acceptable values of Lagrange multipliers $\boldsymbol{\alpha_i}$. The most appropriate choice for $\boldsymbol{C}$ will depend on the specific data set available.

Support Vector Machine: Non-Linear Case with Kernels

SVMs represented with a linear function $\boldsymbol{f}(\boldsymbol{x})$ (i.e. a separating hyperplane) have very limited representational power. As such, they could not be very useful in practical classification problems. It is often the case that the data is far from linear and the data sets are inseparable. The good news is that with a slight modification, SVM could solve highly non-linear classification problems. The solution is to use a “kernel trick”. Kernels are used to non-linearly map the input data to a high-dimensional space. The new mapping is then linearly separable. Figure 23 indicates data points that are non-linearly separable.

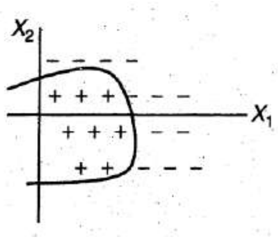
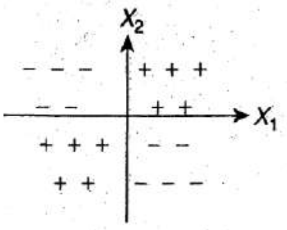


Figure 23 Non-linear separable data.

Example 10
XOR Problem



By constructing a new attribute:

$$X_1{}' = X_1 X_2$$

the XOR problem becomes linearly separable by the new attribute $X_1{}'$.

Example 11

Taylor Expansion

Value of a multi-dimensional function $F(x)$ at point x can be approximated as

$$\boldsymbol{F(\vec{x}) = F(\vec{x}_0) + \nabla F(\vec{x}_0)(\vec{x}-\vec{x}_0) + (\vec{x}-\vec{x}_0)^T \nabla^2 F(\vec{x}_0)(\vec{x}-\vec{x}_0) + O(\|\vec{x}-\vec{x}_0\|^3)}$$

Therefore, $\boldsymbol{F(x)}$ can be considered as a linear combination of complex attributes derived from the original ones:

$$\boldsymbol{F(\vec{x}) = F(\vec{x}_0) + \sum_{i=1}^{m} a_i x_i + \sum_{i,j=1}^{m} a_{ij} x_i x_j + \sum_{i,j,k=1}^{m} a_{ijk} x_i x_j x_k + O(\|\vec{x}-\vec{x}_0\|^3)}$$

14.What is referred to as Curse of Dimensionality in SVM? Explain. [CO2-L1]

SVM and Curse-of-Dimensionality

If the original attribute space is transformed into a very high-dimensional space, the likelihood of being able to solve the non-linear classification increases. However, one is likely to quickly encounter the curse-of-dimensionality problem.

The strength of SVM lies in the theoretical justification that margin maximization is an effective mechanism for alleviating the curse-of-dimensionality problem (i.e. SVM is the simplest classifier that solves the given classification problem). Therefore, SVMs are able to successfully solve classification problems with extremely high attribute dimensionality.

SVM Solution for Classification

As stated earlier, we use kernels to non-linearly map the input data to a high-dimensional space. The new mapping is then linearly separable denoting $\Phi: \Re^M \longrightarrow F$as a mapping from the original M- dimensional attribute space to the highly dimensional attribute space F.

By solving the following dual problem:

Find $\boldsymbol{\alpha}$ that maximizes

$$\sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \Phi(x_i)^T \Phi(x_j)$$

subject to

$$\sum_{i=1}^{N} \alpha_i y_i = 0, \qquad 0 \le \alpha_i \le C, \qquad \forall i$$

the resulting SVM is of the form:

$$f(x) = w^T \Phi(x_i) + b = \sum_{i=1}^{N} \alpha_i y_i \Phi(x_i)^T \, \Phi(\mathrm{x}) + b$$

Although SVMs are successful in dealing with highly dimensional attribute spaces, the fact that the SVM training scales linearly with the number of attributes, and considering limited memory space, could largely limit the choice of mapping Φ.

Solution: Kernel Trick

It allows computing scalar products (e.g. $\Phi(x_i)^T$, $\Phi(\mathrm{x})$) in the original attribute space. It follows from a result called Mercers theorem which states:

“There is a class of mappings Φ that has the following property: $\boldsymbol{\Phi}(\boldsymbol{x})^{\boldsymbol{T}}\boldsymbol{\Phi}(\boldsymbol{y}) = \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{K}$ is a corresponding kernel function.”

Examples of kernel function:

1.Gaussian Kernel: $K(x, y) = e^{\frac{\|x-y\|^2}{A}}$, A is a constant.
2.Polynomial Kernel: $K(x, y) = (x^T y + 1)^B$, B is a constant.

By introducing the kernel trick:

The dual problem is find $\boldsymbol{\alpha}$ that maximizes

$$\sum_{i} \alpha_i - \frac{1}{2}\sum_{i}\sum_{j} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

Subject to

$$\sum_{i=1}^{N} \alpha_i y_i = 0, \qquad 0 \le \alpha_i \le C, \qquad \forall i$$

The resulting SVM is

$$f(x) = w^T \Phi(x_i) + b = \sum_{i=1}^{N} \alpha_i y_i k(x_i, x) + b$$

15.List out and explain some of the applications of SVM in detail [CO2-L1]

SVM Applications

SVM is a useful technique for data classification. A classification task usually involves training and testing data which consist of some data instances. Each instance in the training set contains one target value and several attributes. The goal of SVM is to produce a model which predicts target value of data instances in the testing set which are given only the attributes. Thus, classification in SVM is an example of supervised learning. SVM can also be applied to regression problems by the introduction of an alternative loss function. The loss function must be modified to include a distance measure. The regression can be linear and non-linear. Similar to classification problems, a non-linear model is usually required to adequately model the data. In the same manner, as the non-linear SVC approach, a non-linear mapping can be used to map the data into a high-dimensional feature space where linear regression is performed. The kernel approach is again employed to address the curse-of-dimensionality. SVM has been found to be successful when used for pattern classification problems.

Applying the support vector approach to a particular practical problem involves resolving a number of questions based on the problem definition and the design involved with it. One of the major challenges is that of choosing an appropriate kernel for the given application. There are standard choices such as a Gaussian or polynomial kernel that are the default options, but if these prove ineffective or if the inputs are discrete structures more elaborate kernels will be needed. Once the choice of kernel and optimization criterion has been made, the key components of the system are in place.

For example, consider the text categorization problem. This is the classification of natural text documents into a fixed number of predefined categories based on their content. Since a document can be assigned to more than one category, this is not a multi-class classification problem, but can be viewed as a series of binary classification problems, one for each category. One of the standard representations of text for the purposes of information retrieval provides an ideal feature mapping for constructing a Mercer kernel.

Traditional classification approaches perform poorly when working directly because of the high dimensionality of the data, but SVMs can avoid the pitfalls of very high dimensional representations. A very similar approach to the techniques described for text categorization can also be used for the task of image classification, as in that case linear hard margin machines are frequently able to generalize well. The first real-world task on which support vector machines were tested was the problem of handwritten character recognition.

The major strengths of SVM are that the training is relatively easy. There is no problem of a local optimum. It scales relatively well to high-dimensional data and the trade-off between classifier complexity and error can be controlled explicitly. The weakness includes the need for a good kernel function.

Time Series Analysis

16.Explain the role of time series analysis in data analytics constructing a sample time series model. [CO2-L2]

Time Series Analysis

Time series modelling is a popular and recent research area which has attracted attention of researcher community over last few decades. The main aim of time series analysis is to collect and analyze the past observations to develop an appropriate model which can then be used to generate future values for the series, that is, to make forecasts. Time series forecasting is based on the idea that the history of occurrences over time can be used to predict the future. Time series models base their prediction of the future upon past values of a variable and/or past error without attempting to discover the factors affecting the behaviour of the series.

Time series forecasting is very popularly used in several practical fields such as business, economics, finance, science and engineering, etc. Several techniques exist to analyze time series and fit a model to explain them. Successful time series forecasting depends on an appropriate model fitting. Several researchers have contributed a plethora of methods for time series forecasting. In this section, we provide an overview of such techniques.

The Time Series Model

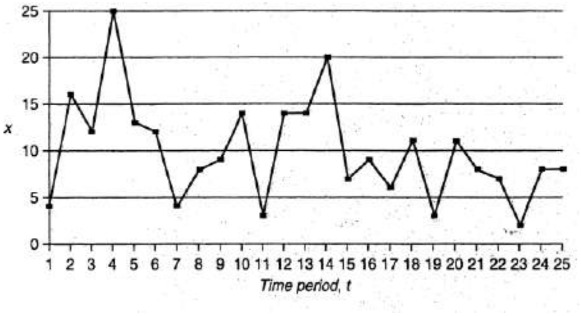
A time series is a sequential set of data points, measured typically at successive times. It is mathematically defined as a set of vectors $\boldsymbol{x}(\boldsymbol{t}), \boldsymbol{t} = \mathbf{0}, \mathbf{1}, \mathbf{2}, \ldots$ where $\boldsymbol{t}$ represents the time elapsed. The variable $\boldsymbol{x}(\boldsymbol{t})$ is treated as a random variable.

A time series model will generally reflect the fact that observations close together in time will be more closely related than observations further apart. In addition, time series models will often make use of the natural one-way ordering of time so that values for a given period will be expressed as deriving in some way from past values, rather than from future values.

The data might represent the weekly demand for some product. We use $\boldsymbol{x}$ to indicate an observation and t to represent the index of the time period. The observed demand for time t is specifically designate $\boldsymbol{dxt}$. The data from 1 through $\boldsymbol{T}$ is:$\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_t}$. The lines connecting the observations on the figure are provided only to clarify the picture and otherwise have no meaning.

Time	**Observation**									
1-10	4	16	12	25	13	12	4	8	9	14
11 -20	3	14	14	20	7	9	6	11	3	11
21-30	8	7	2	8	8	10	7	16	9	4

Weekly demand for weeks 1 through 30.

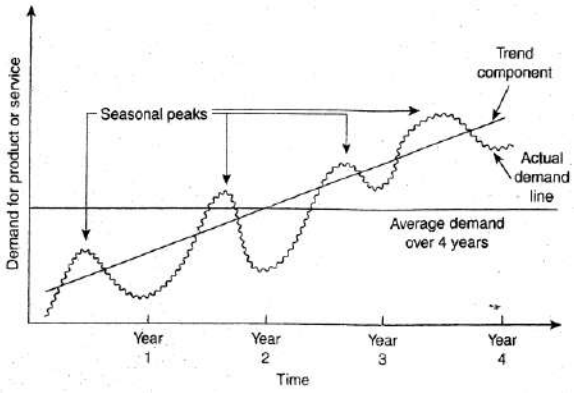


A time series of weekly demand.

Any time series is a composition of many individual underlying component time series. Some of these components are predictable whereas other components may be almost random which can be difficult to predict. This calls for decomposition methods that will generate individual components series from the original series. Decomposing a series into such components enables us to analyze the behaviour of each component and this can help to improve the accuracy of the final forecast. A typical sales time series can be considered to be a combination of four components (see Fig. 26).

Time series models are characterized of four components:

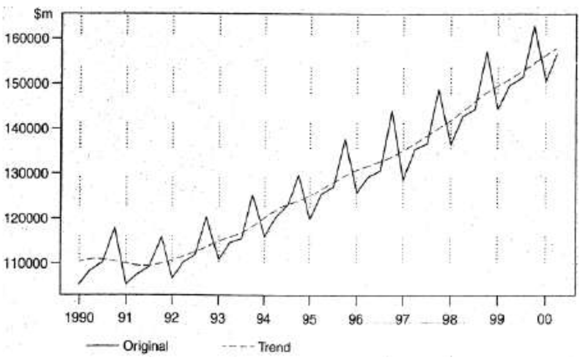
1.Trend component.
2.Cyclical component.
3.Seasonal component.
4.Irregular component.



Components of time series.

1.Trend

The trend is defined as the “long term” movement in a time series without calendar-related and irregular effects, and is a reflection of the underlying level. It is the result of influences such as population growth, price inflation and general economic changes. The graph above depicts a series in which there is an obvious upward trend over time.



Quarterly gross domestic product (upward trend)

2.Seasonality:

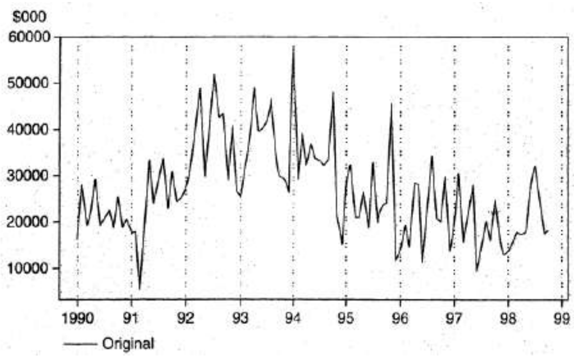
The component of the time series that captures the variability in the data due to seasonal fluctuations is called the seasonal component. Seasonal commodities are the best examples for seasonal components. The seasonal component consists of effects that are reasonably stable with respect to timing, direction and magnitude. It arises from systematic, calendar-related influences.

3.Cycles:

Any regular pattern of the sequence of values above and below the trend line. Any recurring sequence of points above and below the trend line that last for more than a year is considered to constitute the cyclical component of the time series.

4.Irregular:

The irregular component of the time series cannot be predicted in advance. The random variations in the time series are caused by short-term, unanticipated and non-recurring factors that affect the time series. The irregular component (sometimes also known as the residual) is what remains after the seasonal and trend components of a time series have been estimated and removed. It results from short-term fluctuations in the series which are neither systematic nor predictable. In a highly irregular series, these fluctuations can dominate movements, which will mask the trend and seasonality. The graph below is of a highly irregular time series.



Monthly value of building approvals, Australian Capital Territory (highly irregular time series).

17.Write a short note on the Decomposition Models of Time Series Analysis [CO2-L1]

Decomposition Model

Mathematical representation of the decomposition approach is

$$Y_t = f(S_t, T_t, E_t)$$

where

$\boldsymbol{Y_t}$ is the time series value (actual data) at period $\boldsymbol{t}$;

$\boldsymbol{S_t}$ is the seasonal component (index) at period $\boldsymbol{p}$

$\boldsymbol{T_t}$ is the trend-cycle component at period $\boldsymbol{t}$

$\boldsymbol{E_t}$ is the irregular (remainder) component at period $\boldsymbol{t}$.

There are two types of models:

1.Additive decomposition.
2.Multiplicative decomposition.

Additive decomposition

In some time series, the amplitude of both the seasonal and irregular variations do not change as the level of the trend rises or falls. In such cases, an additive model is appropriate. The additive model implies components are independent. Hence, if the magnitude of the seasonal fluctuation does not vary with the level of the series an additive model is used. Figure 29 depicts a typically additive series. The underlying level of the series fluctuates but the magnitude of the seasonal spikes remains approximately stable.

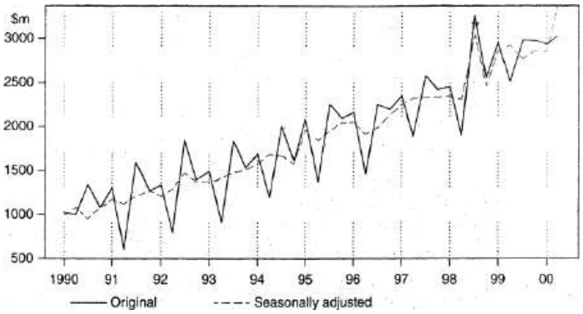


Figure 29 General government transfers to other sectors (additive series).

In the additive model, the observed time series ($\boldsymbol{Y_t}$) is considered to be the sum of three independent components: the seasonal $\boldsymbol{S_t}$, the trend $\boldsymbol{T_t}$ and the irregular $\boldsymbol{I_t}$ That is

$$\boldsymbol{Y_t = T_t + S_t + I_t}$$

Each of the three components has the same units as the original series. The seasonally adjusted series is obtained by estimating and removing the seasonal effects from the original time series.

Multiplicative decomposition

In many time series, the amplitude of both the seasonal and irregular variations increase as the level of the trend rises. In this situation, a multiplicative model is usually appropriate. In the multiplicative model, the original time series is expressed as the product of trend, seasonal and irregular components:

$$\boldsymbol{Y_t = T_t \times S_t \times I_t}$$

Neural Networks

18.Explain the structure of neural networks and the mathematical functions used in the data analysis process [CO2-L1]

Neural Networks

A complex algorithm used for predictive analysis, the *neural network*, is biologically inspired by the structure of the human brain. A neural network provides a very simple model in comparison to the human brain, but it works well enough for our purposes.

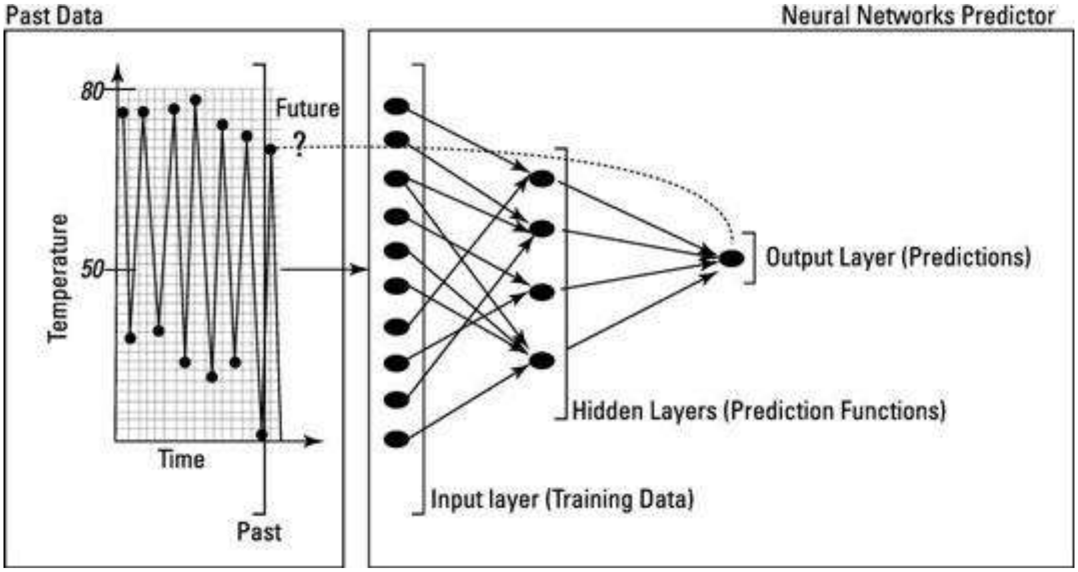
Widely used for data classification, neural networks process past and current data to estimate future values — discovering any complex correlations hidden in the data — in a way analogous to that employed by the human brain.

Neural networks can be used to make predictions on time series data such as weather data. A neural network can be designed to detect pattern in input data and produce an output free of noise.

Structure of Neural Networks

The structure of a neural-network algorithm has three layers:

- **The input layer** feeds past data values into the next (hidden) layer. The black circles represent *nodes* of the neural network.
- **The hidden layer** encapsulates several complex functions that create predictors; often those functions are hidden from the user. A set of nodes (black circles) at the hidden layer represents mathematical functions that modify the input data; these functions are called *neurons*.
- **The output layer** collects the predictions made in the hidden layer and produces the final result: the model's prediction.


Past Data
Neural Networks Predictor
80
50
Temperature
Time
Future
?
Past
Output Layer (Predictions)
Hidden Layers (Prediction Functions)
Input layer (Training Data)

Here's a closer look at how a neural network can produce a predicted output from input data. The hidden layer is the key component of a neural network because of the neurons it contains; they work together to do the major calculations and produce the output.

Each neuron takes a set of input values; each is associated with a *weight* (more about that in a moment) and a numerical value known as *bias*. The output of each neuron is a function of the output of the weighted sum of each input plus the bias.

Mathematical Function

Most neural networks use mathematical functions to activate the neurons. A *function* in math is a relation between a set of inputs and a set of outputs, with the rule that each input corresponds to an output.

For instance, consider the negative function where a whole number can be an input and the output is its negative equivalent. In essence, a function in math works like a black box that takes an input and produces an output.

Neurons in a neural network can use *sigmoid* functions to match inputs to outputs. When used that way, a sigmoid function is called a *logistic function* and its formula looks like this:

$f(\mathit{input}) = 1/(1+e^{\mathit{output}})$

Here *f* is the *activation function* that activates the neuron, and *e* is a widely used mathematical constant that has the approximate value of 2.718.

You might wonder why such a function is used in neurons. Well, most sigmoid functions have derivatives that are positive and easy to calculate. They’re continuous, can serve as types of smoothing functions, and are also bounded functions.

This combination of characteristics, unique to sigmoid functions, is vital to the workings of a neural network algorithm — especially when a derivative calculation — such as the weight associated with each input to a neuron — is needed.

The weight for each neuron is a numerical value that can be derived using either supervised training or unsupervised training such as data clustering.

- In the case of supervised training, weights are derived by feeding sample inputs and outputs to the algorithm until the weights are *tuned* (that is, there’s a near-perfect match between inputs and outputs).
- In the case of unsupervised training, the neural network is only presented with inputs; the algorithm generates their corresponding outputs. When the algorithms are presented with new-but-similar inputs and the algorithm produces new outputs that are similar to previous outputs, then the neurons’ weights have been tuned.

Neural networks tend to have high accuracy even if the data has a significant amount of noise. That’s a major advantage; when the hidden layer can still discover relationships in the data despite noise, you may be able to use otherwise-unusable data.

One disadvantage of the neural-network algorithms is that the accuracy of the prediction may be valid only within the time period during which the training data was gathered.

19.What is Principal Component Analysis and how is it useful in explaining data patterns? [CO2-L1]

Principal Component Analysis

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

PCA is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by some projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on.

Consider a data matrix, **X**, with column-wise zero empirical mean (the sample mean of each column has been shifted to zero), where each of the *n* rows represents a different repetition of the experiment, and each of the *p* columns gives a particular kind of feature (say, the results from a particular sensor).

Mathematically, the transformation is defined by a set of *p*-dimensional vectors of weights or *loadings* $w_{(k)} = (w_1, \dots, w_p)_{(k)}$ that map each row vector $x_{(i)}$of **X** to a new vector of Principal component *scores* $t_{(i)} = (t_1, \dots, t_k)_{(i)}$, given by

$$\boldsymbol{t_{k(i)} = X_i \cdot W_{(k)}}$$

in such a way that the individual variables of **t** considered over the data set successively inherit the maximum possible variance from **x**, with each loading vector **w** constrained to be a unit vector.

Example

A consumer products company wants to analyze customer responses to several characteristics of a new shampoo: color, smell, texture, cleanliness, shine, volume, amount needed to lather, and price. They perform a principal components analysis to determine whether they can form a smaller number of uncorrelated variables that are easier to interpret and analyze. The results identify the following patterns:

- Color, smell, and texture form a "Shampoo quality" component.
- Cleanliness, shine, and volume form an "Effect on hair" component.
- Amount needed to lather and price form a "Value" component.

Factor Analysis

Factor analysis is a method for explaining the structure of data by explaining the correlations between variables. Factor analysis summarizes data into a few dimensions by condensing a large number of variables into a smaller set of latent variables or factors. It is commonly used in the social sciences, market research, and other industries that use large data sets.

Consider a credit card company that creates a survey to assess customer satisfaction. The survey is designed to answer questions in three categories: timeliness of service, accuracy of the service, and courteousness of phone operators. The company can use factor analysis to ensure that the survey items address these three areas before sending the survey to a large number of customers. If the survey does not adequately measure the three factors, then the company should revaluate the questions and retest the survey before sending it to customers.

Fuzzy Logics

20.What purpose does the Fuzzy logics serve in the field of data analysis? Explain. [CO2-L2]

Fuzzy Logics

Fuzzy logic is a form of many-valued logic in which the truth values of variables may be any real number between 0 and 1, considered to be "fuzzy". By contrast, in Boolean logic, the truth values of variables may only be 0 or 1, often called "crisp" values. Fuzzy logic has been employed to handle the concept of partial truth, where the truth value may range between completely true and completely false. Furthermore, when linguistic variables are used, these degrees may be managed by specific (membership) functions.

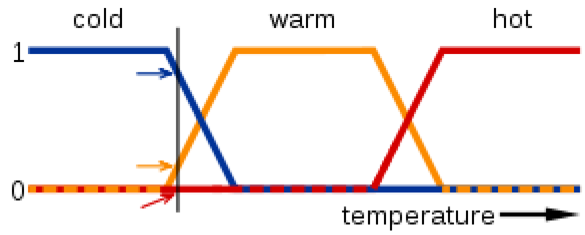
Boolean logic only permits conclusions which are either true or false. However, there are also propositions with variable answers, such as one might find when asking a group of people to identify a color. In such instances, the truth appears as the result of reasoning from inexact or partial knowledge in which the sampled answers are mapped on a spectrum.

Humans and animals often operate using fuzzy evaluations in many everyday situations. In the case where someone is tossing an object into a container from a distance, the person does not compute exact values for the object weight, density, distance, direction, container height and width, and air resistance to determine the force and angle to toss the object. Instead the person instinctively applies quick “fuzzy” estimates, based upon previous experience, to determine what output values of force, direction and vertical angle to use to make the toss.

Both degrees of truth and probabilities range between 0 and 1 and hence may seem similar at first. For example, let a 100 ml glass contain 30 ml of water. Then we may consider two concepts: empty and full. The meaning of each of them can be represented by a certain fuzzy set.one might define the glass as being 0.7 empty and 0.3 full. Note that the concept of emptiness would be subjective and thus would depend on the observer or designer. Another designer might, equally well, design a set membership function where the glass would be considered full for all values down to 50 ml. It is essential to realize that fuzzy logic uses degrees of truth as a mathematical model of vagueness, while probability is a mathematical model of ignorance.

Applying truth values

A basic application might characterize various sub-ranges of a continuous variable. For instance, a temperature measurement for anti-lock brakes might have several separate membership functions defining particular temperature ranges needed to control the brakes properly. Each function maps the same temperature value to a truth value in the 0 to 1 range. These truth values can then be used to determine how the brakes should be controlled.



Fuzzy logic temperature

In this image, the meanings of the expressions *cold*, *warm*, and *hot* are represented by functions mapping a temperature scale. A point on that scale has three "truth values" — one for each of the three functions. The vertical line in the image represents a particular temperature that the three arrows (truth values) gauge. Since the red arrow points to zero, this temperature may be interpreted as "not hot". The orange arrow (pointing at 0.2) may describe it as "slightly warm" and the blue arrow (pointing at 0.8) "fairly cold".

Linguistic variables

While variables in mathematics usually take numerical values, in fuzzy logic applications, the non-numeric are often used to facilitate the expression of rules and facts.

A linguistic variable is *young* or its antonym *old*. However, the value of linguistic variables is that they can be modified via linguistic hedges applied to primary terms. These linguistic hedges can be associated with certain functions.

Fuzzification operations can map mathematical input values into fuzzy membership functions. And the opposite de-fuzzifying operations can be used to map a fuzzy output membership functions into a “crisp” output value that can be then used for decision or control purposes.

Forming a consensus of Inputs and Fuzzy Rules

Since the fuzzy system output is a consensus of all of the inputs and all of the rules, Fuzzy logic systems can be well behaved when input values are not available or are not trustworthy. Weightings can be optionally added to each rule in the rulebase and weightings can be used to regulate the degree to which a rule affects the output values. These rule weightings can be based upon the priority, reliability or consistency of each rule. These rule weightings may be static or can be changed dynamically, even based upon the output from other rules.

The Fuzzy Logic Process

1. Fuzzify all input values into fuzzy membership functions.
2. Execute all applicable rules in the rulebase to compute the fuzzy output functions.
3. De-fuzzify the fuzzy output functions to get "crisp" output values.

21.In what way the stochastic approach help in optimizing deterministic model [CO2-L1]

Stochastic Optimization

Stochastic optimization (**SO**) methods are optimization methods that generate and use random variables. For stochastic problems, the random variables appear in the formulation of the optimization problem itself, which involve random objective functions or random constraints. For example, Stochastic optimization methods also include methods with random iterates. Some stochastic optimization methods use random iterates to solve stochastic problems, combining both meanings of stochastic optimization. Stochastic optimization methods generalize deterministic methods for deterministic problems.

Randomized search methods

On the other hand, even when the data set consists of precise measurements, some methods introduce randomness into the search-process to accelerate progress. Such randomness can also make the method less sensitive to modeling errors. Further, the injected randomness may enable the method to escape a local optimum and eventually to approach a global optimum. Indeed, this randomization principle is known to be a simple and effective way to obtain algorithms with *almost certain* good performance uniformly across many data sets, for many sorts of problems. Stochastic optimization methods of this kind include:

- Simulated Annealing
- Quantum Annealing
- Probability Collectives
- Reactive Search Optimization (RSO)
- Cross-Entropy Method
- Random Search
- Informational Search
- Stochastic Tunnelling
- Parallel Tempering
- Stochastic Hill Climbing
- Swarm Algorithms

- Evolutionary Algorithms
  - Genetic Algorithms
  - Evolution Strategies

Simulated Annealing

There are certain optimization problems that become unmanageable using combinatorial methods as the number of objects becomes large. A typical example is the **traveling salesman problem,** which belongs to the **NP-complete** class of problems. For these problems, there is a very effective practical algorithm called simulated annealing (thus named because it mimics the process undergone by misplaced atoms in a metal when its heated and then slowly cooled). While this technique is unlikely to find the optimum solution, it can often find a very good solution, even in the presence of noisy data.

The traveling salesman problem can be used as an example application of simulated annealing. In this problem, a salesman must visit some large number of cities while minimizing the total mileage traveled. If the salesman starts with a random itinerary, he can then pairwise trade the order of visits to cities, hoping to reduce the mileage with each exchange. The difficulty with this approach is that while it rapidly finds a **local minimum,** it cannot get from there to the **global minimum**.

Simulated annealing improves this strategy through the introduction of two tricks. The first is the so-called "Metropolis algorithm", in which some trades that do not lower the mileage are accepted when they serve to allow the solver to "explore" more of the possible space of solutions. Such "bad" trades are allowed using the criterion that

$$e^{-\Delta D/T} > R(0, 1),$$

where $\Delta D$is the change of distance implied by the trade (negative for a "good" trade; positive for a "bad" trade), $T$is a "synthetic temperature," and $R(0, 1)$is a random number in the interval $[0, 1]$. $D$is called a "cost function," and corresponds to the free energy in the case of annealing a metal (in which case the temperature parameter would actually be the kT, where $k$is Boltzmann's Constant and $T$is the physical temperature, in the Kelvin absolute temperature scale). If $T$is large, many "bad" trades are accepted, and a large part of solution space is accessed. Objects to be traded are generally chosen randomly, though more sophisticated techniques can be used.

The second trick is, again by analogy with annealing of a metal, to lower the "temperature." After making many trades and observing that the cost function declines only slowly, one lowers the temperature, and thus limits the size of allowed "bad" trades. After lowering the temperature several times to a low value, one may then "quench" the process by accepting only "good" trades in order to find the local minimum of the cost function. There are various "annealing schedules" for lowering the temperature, but the results are generally not very sensitive to the details.

There is another faster strategy called threshold acceptance. In this strategy, all good trades are accepted, as are any bad trades that raise the cost function by less than a fixed threshold. The threshold is then periodically lowered, just as the temperature is lowered in annealing. This eliminates exponentiation and random number generation in the Boltzmann criterion. As a result, this approach can be faster in computer simulations.

Unit – III

Introduction

Part – A

1.What factors lead to Concept Drift? [CO3-L1]

The constantly changing patterns in data streams can affect the induced data mining models in multiple ways such as

✓Changes in the class label of an existing data variable,

✓Change in the available feature information.

Both these lead to a phenomenon Called **Concept drift**.

2.How are continuous queries evaluated? [CO3-L1]

Continuous queries are evaluated continuously as data streams continue to arrive. The answer to a continuous query is produced over time, always reflecting the stream data seen so far.

Continuous query answers may be stored and updated as new data arrives, or they may be produced as data streams themselves.

3.What are called Ad-Hoc Queries? [CO3-L1]

An ad-hoc query is issued online after the data streams have already begun. Ad-hoc queries can be either one-time queries or continuous queries. Ad-hoc queries are basically questions asked once about the current state of a stream or streams.

4.List out few challenges of data mining algorithms. [CO3-L2]

Data streams pose several challenges for data mining algorithm design, The most important of them are,

- Algorithms must make use of limited resources (time and memory).
- Algorithms must deal with data whose distribution changes over time.

5.Why traditional data mining algorithms could not be used on data streams? [CO3-L1]

Many traditional data mining algorithms can be modified to work with larger datasets, but they cannot handle continuous supply of data.

If a traditional algorithm has learnt and induced a model of the data seen until now, it cannot immediately update the model when new information keeps arriving at continuous intervals. Instead, the entire training process must be repeated with the new examples included.

6.What are one time queries? [CO3-L1]

One-time queries are queries that are evaluated once over a point-in-time snapshot of the data set, with the answer returned to the user.

For example, a stock price checker may alert the user when a stock price crosses a particular price point.

7.What are continuous queries? [CO3-L1]

Continuous queries are evaluated continuously as data streams continue to arrive. The answer to a continuous query is produced over time, always reflecting the stream data seen so far.

Continuous query answers may be stored and updated as new data arrives, or they may be produced as data streams themselves.

8.What is a predefined query? [CO3-L1]

A pre-defined query is one that is supplied to the DSMS before any relevant data has arrived. Pre-defined queries are most commonly continuous queries.

9.List out the major issues in Data Stream Query Processing. [CO3-L2]

The major issues in Data Stream Query Processing are as follows,

- Unbounded Memory Requirements
- Approximate Query Answering
- Sliding Windows
- Batch Processing, Sampling and Synopses
- Blocking Operators

10.What is Reservoir Sampling? [CO3-L1]

Reservoir sampling is a family of randomized algorithms for randomly choosing k samples from a list of n items, where n is either a very large or unknown number. Typically n is large enough that the list doesn’t fit into main memory. For example, a list of search queries in Google

11.How is Biased Reservoir Sampling different from Reservoir Sampling? [CO3-L1]

Biased reservoir sampling is a bias function to regulate the sampling from the stream. This bias gives a higher probability of selecting data points from recent parts of the stream as compared to distant past. This bias function is quite effective since it regulates the sampling in a smooth way so that the queries over recent horizons are more accurately resolved.

12.What does the term “Filtering a Data Stream” mean? [CO3-L1]

“Filtering” tries to observe an infinite stream of data, look at each of its items and quantify whether the item is of interest and should be stored for further evaluation.

Hashing has been the popular solution to designing algorithms that approximate some value that we would like to maintain.

13.What is a Bloom Filter? [CO3-L1]

A Bloom filter is a space-efficient probabilistic data structure. It is used to test whether an element is a member of a set. False positive matches are possible, but false negatives are not, thus a Bloom filter has a 100% recall rate.

14.What is a cardinality estimation problem? [CO3-L1]

The count-distinct problem, also known as ***cardinality estimation problem***. It is the problem of finding the number of distinct elements in a data stream with repeated elements.

This is a well-known problem with numerous applications.

15.What is Flajolet–Martin algorithm used for? [CO3-L1]

The Flajolet–Martin algorithm is an algorithm for approximating the number of distinct elements in a stream with a single pass and space-consumption which is logarithmic in the maximum number of possible distinct elements in the stream.

16.How are “moments” estimated? [CO3-L1]

The method of moments is a method of estimation of population parameters. One starts with deriving equations that relate the population moments (i.e., the expected values of powers of the random variable under consideration) to the parameters of interest.

17.What is called the decay of insight? [CO3-L1]

The length of time that analytic insight has value is rarely considered in big data and analytics projects.

The concept of the half-life of insight can be used to understand the expectation of the magnitude of insight after a period of time. It gives the expectation $M(t)$ of the magnitude of the insight after time t.

18.What is Real-Time Analysis? [CO3-L1]

Real-time analytics is the use of, or the capacity to use, all available enterprise data and resources when they are needed. It consists of dynamic analysis and reporting, based on data entered into a system less than one minute before the actual time of use. Real-time analytics is also known as real-time data analytics

19.What is a Data Stream Management System? [CO3-L1]

A Data stream management system (DSMS) is a computer program to manage continuous data streams. It is similar to a database management system (DBMS), which is, however, designed for static data in conventional databases. A DSMS also offers a flexible query processing so that the information need can be expressed using queries.

20.What is Event Stream Processing? [CO3-L1]

Event stream processing, or ESP, is a set of technologies designed to assist the construction of event-driven information systems. ESP technologies include event visualization, event databases, event-driven middleware, and event processing languages, or complex event processing (CEP).

21.What is called Data Stream Mining? [CO3-L1]

Data Stream Mining is the process of extracting knowledge structures from continuous, rapid data records. A data stream is an ordered sequence of instances that in many applications of data stream mining can be read only once or a small number of times using limited computing and storage capabilities.

Part – B

1.Explain data stream management systems in detail. [CO3-L2]

Data Streams

In the recent years, a new class of data-intensive applications has become widely recognized, that is, applications in which the data is modelled best as transient data streams instead of traditional Database.

Data stream real-time analytics are needed to manage the data currently generated, at an ever increasing rate from these applications. *Examples of such applications include financial applications, network monitoring, security, telecommunications data management, web applications, manufacturing, sensor networks, call detail records, email, blogging, twitter posts and others*.

In the data stream model, individual data items may be relational tuples. *Example: network measurements, call records, web page visits, sensor readings, etc.*

The continuous arrival of data in multiple, rapid, time-varying, unpredictable and unbounded streams create new research problems.

In all of the applications mentioned above, it is not feasible to simply load the arriving data into a traditional database management system (DBMS) and operate on it there.

Traditional DBMSs are not designed for rapid and continuous loading of individual data items, and they do not directly support the ***continuous queries*** that are typical of data stream applications.

Also the data in a data stream is lost forever if it is not processed immediately or stored.

But the data in most data streams arrive so rapidly that it is not feasible to store it all in active storage (i.e., in a conventional database), and then interact with it at later time.

Therefore the algorithms that process data streams must work under very strict constraints of space and time.

In addition to this data streams pose several challenges for data mining algorithm design.

•Algorithms must make use of limited resources (time and memory).

•Algorithms must deal with data whose distribution changes over time.

Data Stream Management Systems

Traditional relational databases store and retrieve records of data that are static in nature.

It does not keep track of time unless time is added as an attribute to the database during designing the schema itself.

This model was sufficient for most of the legacy applications (old applications) and older repositories of information.

But many current and emerging applications require support for online analysis of rapidly arriving and changing data streams. This has created a huge number of research activity which attempts to build new models to manage streaming data.

This has resulted in **Data Stream Management Systems (DSMS)**, with importance on ***continuous query languages and query evaluation***.

The generic model for such a DSMS is as follows.

Data Stream Model

A data stream is a real-time, continuous and ordered sequence of items. The ordering may be done implicitly using arrival time or explicitly by using time-stamp.

It is not possible to control the order in which the items arrive. It is also not feasible to locally store a stream fully in any memory device.

Moreover, a query made over streams will actually run continuously over a period of time and incrementally return new results as new data arrives. Therefore, these are known as ***long-running, continuous, standing and persistent queries***.

Any generic model that attempts to store and retrieve data streams must have the following characteristics,

1.The data model and query processor must allow both **order-based** and **time-based** operations

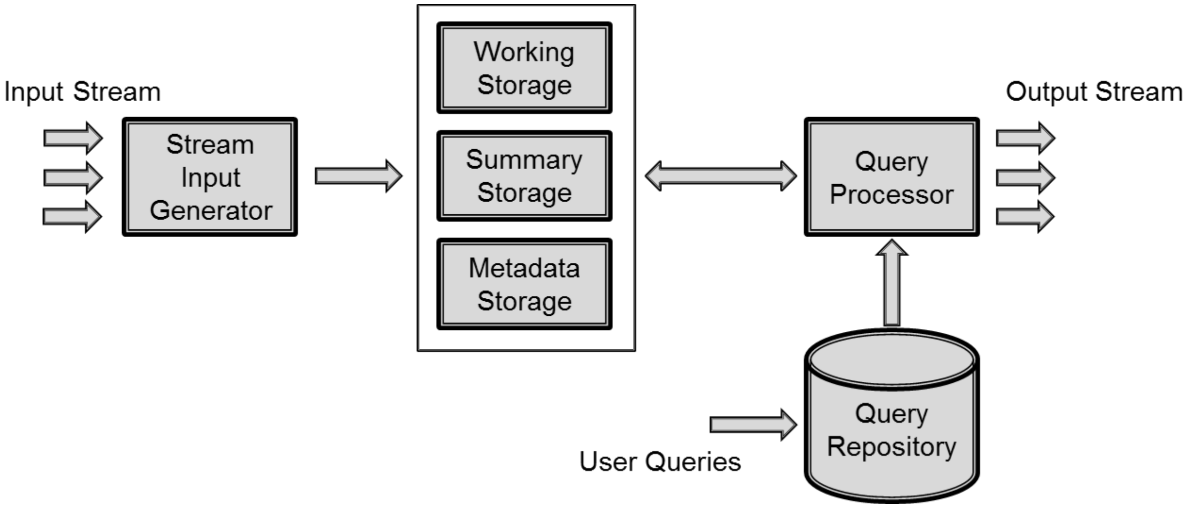
2.Because of the inability to store a complete stream some approximate summary structures must be used. As a result of this summarization ***queries over the summaries may not return exact answers***.

3.Streaming query must not use any operators that require the entire input before any results are produced. Such operators will block the query processor ***indefinitely***.

4.Any query that requires backtracking over a data stream is infeasible. This is due to the storage and performance constraints imposed by a data stream. Thus any online stream algorithm is restricted to make only one pass over the data.

5.Applications that monitor streams in real-time must react quickly to unusual data values. Thus, long-running queries must be prepared for changes in system conditions any time during their execution lifetime (e.g., they may encounter variable stream rates).
6.As per the Scalability requirements parallel and shared execution of many continuous queries must be possible.
An abstract architecture for a typical DSMS is depicted in the figure below.
An input monitor may regulate the input rates, possibly by dropping packets.
Data are typically stored in three partitions:

1.Temporary working storage (e.g., for window queries).
2.Summary storage.
3.Static storage for meta-data (e.g., physical location of each source).
Long-running queries are registered in the query repository and placed into groups for shared processing. It is also possible to pose one-time queries over the current state of the stream.
The query processor communicates with the input monitor and may re-optimize the query plans in response to changing input rates.
Results are streamed to the users or temporarily buffered.



2.Explain Sampling in Data Streams and its types[CO3-L2]

Sampling in Data Streams

Sampling is a common practice for selecting a subset of data to be analysed. Instead of dealing with an entire data stream, we select instances at periodic intervals. Sampling is used to compute statistics of the stream.
Sampling methods reduce the amount of data to process the computational costs, but they can also be a source of errors.

The main problem is to ***obtain a representative sample***, a subset of data that has approximately the same properties of the original data.

Reservoir Sampling

Many mining algorithms can be applied if only we can draw a representative sample of the data from the stream.

Imagine there is a really large stream of data elements.

The goal is to efficiently return a random sample of $\mathbf{1000}$ elements evenly distributed from the original stream.

A simple way is to generate random integers between $\mathbf{0}$ and $(\boldsymbol{N} - \mathbf{1})$, then retrieving the elements at those indices and you have your answer.

To make this sampling without replacement, we simply needs to note whether or not our sample already has that random number and if so, choose a new random number.

This can make the algorithm very expensive if the sample size is very close to N.

Further in the case of a data stream we don't know $\boldsymbol{N}$, the size of the stream in advance and we cannot index directly into it.

We can count it, but that requires making two passes of the data that is not possible.

Thus, the general sampling problem in the case of a data stream is, "How to ensure such a sample is drawn uniformly, given that the stream is continuously growing?"

For example, if we want to draw a sample of 100 items and the stream has length of only 1000, then we want to sample roughly one in ten items. But if a further million items arrive, we must ensure that the probability of any item being sampled is more like one in a million. If we retain the same 100 items, then this is cannot be considered a representative sample.

Several solutions are possible to ensure that we continuously maintain a uniform sample from the stream.

Reservoir-based methods were originally proposed for one-pass access of data from magnetic storage devices such as tapes. Similarly to the case of streams, the number of records $\boldsymbol{N}$ is not known in advance and the sampling must be performed dynamically as the records from the tape are read.

Assume that we wish to obtain an unbiased sample of size $\boldsymbol{n}$ from the data stream and we maintain a reservoir of size $\boldsymbol{n}$ from the data stream. The first $\boldsymbol{n}$ points in the data streams are added to the reservoir for initialization.

Then, when the $(\boldsymbol{t} + \mathbf{1})^{\mathbf{th}}$ point from the data stream is received, it is added to the reservoir with probability $\boldsymbol{n}/(\boldsymbol{t} + \mathbf{1})$.

In order to make room for the new point, any of the current points in the reservoir are sampled with equal probability and subsequently removed.

If we draw a sample of size $\mathbf{1}$, we initialize the sample with the first item from the stream.

Biased Reservoir Sampling

In many cases, the stream data may evolve over time, and the corresponding data mining or query results may also change over time. Thus, the results of a query over a more recent window may be quite different from the results of a query over a more distant window.

Similarly, the entire history of the data stream may not relevant for use in a repetitive data mining application much as classification.

The simple reservoir sampling algorithm can be adapted to a ***sample from a moving window*** over data streams. This is useful in many data stream applications where a small amount of recent history is more relevant than the entire previous stream.

However, this can sometimes be an extreme solution, since for some applications we may need to sample from varying lengths of the stream history.

While recent queries may be more frequent, it is also not possible to completely disregard queries over more distant horizons in the data stream.

Biased reservoir sampling is a bias function to regulate the sampling from the stream. This bias gives a higher probability of selecting data points from recent parts of the stream as compared to distant past. This bias function is quite effective since it regulates the sampling in a smooth way so that the queries over recent horizons are more accurately resolved.

Concise Sampling

Many a time, the size of the reservoir is sometimes restricted by the available main memory. It is desirable to increase the sample size within the available main memory restrictions.

For this purpose, the technique of concise sampling is quite effective. Concise sampling uses the fact that the number of distinct values of an attribute is often significantly smaller than the size of the data stream.

In many applications, sampling is performed based on a single attribute in multi-dimensional data. For example, customer data in an e-commerce site sampling may be done based on only customer ids. The number of distinct customer ids is definitely much smaller than $\boldsymbol{n}$ the size of the entire stream.

The repeated occurrence of the same value can be exploited in order to increase the sample size beyond the relevant space restrictions.

We note that when the number of distinct values in the stream is smaller than the main memory limitations, the entire stream can be maintained in main memory, and therefore, sampling may not even be necessary.

For current systems in which the memory sizes may be of several gigabytes, very large sample sizes can be in main memory as long as the number of distinct values does not exceed the memory constraints.

On the other hand, for more challenging streams with an unusually large number of distinct values, we can use the following approach.

1.The sample is maintained as a set $\boldsymbol{S}$ of $< value, count >$ pairs.
2.For those pairs in which the value of count is one, we do not maintain the count explicitly, but we maintain the value as a singleton.
3.The number of elements in this representation is referred to as the footprint and is bounded above by $\boldsymbol{n}$.
4.We use a threshold parameter $\boldsymbol{X}$ that defines the probability of successive sampling from the stream. The value of $\boldsymbol{X}$ is initialized to be $\mathbf{1}$.
5.As the points in the stream arrive, we add them to the current sample with probability lit.
6.We note that if the corresponding value count pair is already included in the set 5, then we only need to increment the count by 1. Therefore, the footprint size does not increase.
7.On the other hand, if the value of the current point is distinct from all the values encountered so far, of it exists as a singleton then the footprint increases by 1. This is because either a singleton needs to be added, or a singleton gets converted to a value count pair with a count of 2.
8.The increase in footprint size may potentially require the removal of an element from sample S in order to make room for the new insertion.
9.When this situation arises, we pick a new (higher) value of the threshold $\acute{\boldsymbol{\tau}}$, and apply this threshold to the footprint in repeated passes.
10. In each pass, we reduce the count of a value with probability $(\boldsymbol{\tau}/\boldsymbol{\tau}^{\prime})$ until at least one value count pair reverts to a singleton or a singleton is removed.
11. Subsequent points from the stream are sampled with probability $(\mathbf{1}/\boldsymbol{\tau}^{\prime})$.

In practice, $\boldsymbol{\tau}^{\prime}$ may be chosen to be about 10% larger than the value of $\boldsymbol{\tau}$. The choice of different values of $\boldsymbol{\tau}$ provides different trade-offs between the average (true) sample size and the computational requirements of reducing the footprint size.

3.Explain the process of Data Stream Mining with suitable examples[CO3-L2]

Data Stream Mining

Data Stream Mining is the process of extracting useful knowledge from continuous, rapid data streams.

Many traditional data mining algorithms can be modified to work with larger datasets, but they cannot handle continuous supply of data.

For example, if a traditional algorithm has learnt and induced a model of the data seen until now, it cannot immediately update the model when new information keeps arriving at continuous intervals. Instead, the entire training process must be repeated with the new examples included.

With big data, this limitation is both undesirable and highly inefficient.

Mining big data streams faces three principal challenges:

➢Volume
➢Velocity
➢Volatility

Volume and velocity require a high volume of data to be processed in limited time.

From the beginning, the amount of available data constantly increases from zero to potentially infinity. This requires incremental approaches that incorporate information as it becomes available, and online processing if not all data can be kept.

Volatility indicates that environment is never stable and has constantly changing patterns.

In this scenario, old data is of limited use, even if it could be saved and processed again later. The constantly changing patterns can affect the induced data mining models in multiple ways:

✓Changes in the class label of an existing data variable,
✓Change in the available feature information.

Both these lead to a phenomenon Called **Concept drift**.

Example of Concept Drift

Stock Market Application

Consider a stock market application which labels a particular stock as “hold” or “sell” can change the labels rapidly based on a current stream of input information. Changes in the available feature information can arise when new features become available.

Weather Forecasting Application

Consider a continuous weather forecasting application may now need to consider more attributes because of adding new sensors continuously. Existing features might need to be excluded due to regulatory requirements, or a feature might change in its scale, if data from a more precise instrument becomes available.

Thus, CONCEPT DRIFT is a phenomenon that occurs because of feature changes or changes in behaviour of the data itself.

This indicates that one important ingredient to mining data streams is **ONLINE MINING OF CHANGES**. This means we are looking to manage data that arrives online, often in real-time, forming a stream which is potentially infinite.

Even if the patterns are discovered in snapshots of data streams the changes to the patterns may be more critical and informative. With data streams, people are often interested in mining queries like

➢“Compared to the past few days, what are the distinct features of the current status?

➢“What are the relatively stable factors overtime?

Clearly to answer the above queries, we have to examine the changes.

Further mining data streams is challenging in the following two respects.

On one hand, random access to fast and large data streams may be impossible. Thus, multi-pass algorithms (i.e., ones that load data items into main memory multiple times) are often infeasible.

On the other hand, the exact answers from data streams are often too expensive.

The main assumption of data stream processing is that training examples can be briefly inspected a single time only, that is, they arrive in a high speed stream, and then must be discarded to make room for subsequent data.

The algorithm processing the stream has no control over the order of the data seen, and must update its model incrementally as each data element is inspected.

Another desirable property is that we must be able to apply the algorithm at any point of time even in between successive arrivals of data elements in the stream.

All these challenges have resulted in creating a new set of algorithms written exclusively for data streams. These algorithms can naturally cope with very large data sizes and can tackle challenging real-time applications not previously tackled by traditional data mining. The most common data stream mining tasks are CLUSTERING, CLASSIFICATION and FREQUENT PATTERN MINING.

4.Explain and analyse the Bloom Filter in detail with the algorithm[CO3-L2]

The Bloom Filter

A Bloom filter is a space-efficient probabilistic data structure, conceived by Burton Howard Bloom in 1970, that is used to test whether an element is a member of a set. False positive matches are possible, but false negatives are not, thus a Bloom filter has a 100% recall rate. In other words, a query returns either “possibly in set” or -definitely not in set”. Elements can be added to the set, but not removed (though this can be addressed with a “counting” filter). The more elements that are added to the set,, the larger the probability of false positives.

The Bloom filter for a set is much smaller than the set itself, which makes it appropriate for storing large amounts of data required when filtering on a data stream in main memory itself

The advantage of this data structure is that it uses considerably less space than any exact method, but pays for this by introducing a small probability of error. Depending on the space available, this error can be made arbitrarily small.

Let us assume we want to represent n-element sets $S = S_1, S_2, \ldots, S_n$ from a very large universe U, with $|U| = u \gg n$

We want to support insertions and membership queries (as to say "Given $x \in U$ is $x \in S$?") so that:
If the answer is No, then $x \in S$.
If the answer is Yes, then x may or may not be in S, but the probability that $x \notin S$ (false positive) is low.
Both insertions and membership queries should be performed in a constant time. A Bloom filter is a bit vector B of m bits, with k independent hash functions $h_{1,} \dots, h_{k,}$ that map each key in $\boldsymbol{U}$ to the set $R_m = 0, 1, \dots, (m - 1)$. We assume that each hash function h_1 maps a uniformly at random chosen key $x \in S$ to each element of R_m with equal probability. Since we assume the hash functions are independent, it follows that the vector $h_1(x), \dots, h_k(x)$ is equally likely to be any of the m_kk-tuples or elements from R_m.

Algorithm

Initially all m bits of B are set to 0.
Insert x into S. Compute $h_1(x), \dots, h_k(x)$ and set
$B[h_1(x)] = B[h_2(x)] = \cdots = B[h_k(x)] = 1$
Query if $x \in S$, Compute $h_1(x), \dots, h_k(x)$.
If $B[h_1(x)] = B[h_2(x)] = \cdots = B[h_k(x)] = 1$, then answer Yes else answer No.
Clearly, if an item is inserted into the filter, it is found when searched for. Hence, there is no false negative error. The only possibility for error is when an item is not inserted into the filter, buf each of the locations that the hash functions map to it are all turned on. We will show that this error can be kept small while using considerably less space than any exact method.

Analysis of the Bloom Filter

If a key value is in 5, then the element will surely pass through the Bloom filter. However, if the key value is not in S, it might still pass. This is called a false positive. We need to understand how to calculate the probability of a false positive, as a function of n, the bit-array length, m the number of members of $\boldsymbol{S}$, and $\boldsymbol{k}$, the number of hash functions. The computation can be done as follows:

1.The probability that one hash do not set a given bit is $(\mathbf{1} - \mathbf{1}/\boldsymbol{m})$ given the setting in previous section.
2.The probability that it is not set by any of the k hash functions is $(\mathbf{1} - \mathbf{1}/\boldsymbol{m})^{\boldsymbol{k}}$.
3.Hence, after all $\boldsymbol{n}$ elements of $\boldsymbol{S}$ have been inserted into the Bloom filter, the probability that a specific bit is still $\mathbf{0}$ is $f = (\mathbf{1} - \mathbf{1}/\boldsymbol{m})^{\boldsymbol{kn}} = e^{-kn/m}$. (Note that this uses the assumption that the hash functions are independent and perfectly random.)
4.The probability of a false positive is the probability that a specific set of h bits are 1, which is

5.$(\mathbf{1}-(\mathbf{1}\ -\ \mathbf{1}/\boldsymbol{m})^{\boldsymbol{kn}})\boldsymbol{k}=(\mathbf{1}-\boldsymbol{e}^{-\boldsymbol{kn}/\boldsymbol{m}})^{\boldsymbol{k}}$

6.Thus, we can identify three performance metrics for Bloom filters that can be adjusted to tune its performance. First the computation time (corresponds to the number $\boldsymbol{k}$ of hash functions), second the size (corresponds to the number $\boldsymbol{m}$ of bits), and finally probability of error (corresponds to the false positive rate).

7.Suppose we are given the ratio *min* and want to optimize the number $\boldsymbol{k}$ of hash functions to minimize the false positive rate $\boldsymbol{f}$. Note that more hash functions increase the precision but also the number of $\mathbf{1}\boldsymbol{s}$ in the filter, thus making false positives both less and more likely at the same time.

8.Let $\boldsymbol{g}-\ \boldsymbol{ln}(\boldsymbol{f})-\boldsymbol{kln}(\boldsymbol{l}-\boldsymbol{e}^{-\boldsymbol{kn}/\boldsymbol{m}})^{\boldsymbol{k}}$. Suppose $\boldsymbol{p}=\boldsymbol{e}^{-\boldsymbol{kn}/\boldsymbol{m}}$. Then, the derivative of $\boldsymbol{g}$ is

$$\frac{\boldsymbol{dg}}{\boldsymbol{dk}}=\ln(\mathbf{1}-\boldsymbol{p})+\frac{\boldsymbol{kn}}{\boldsymbol{m}}\frac{\boldsymbol{p}}{\mathbf{1}-\boldsymbol{p}}$$

To find the optimal $\boldsymbol{k}$, or right number of hash functions to use, we should equate the above derivative to $\mathbf{0}$. The solution is

$$\boldsymbol{k}=\left[(\ln 2)\frac{\boldsymbol{m}}{\boldsymbol{n}}\right]$$

So for this optimal value of $\boldsymbol{k}$, the false positive rate is

$$\left(\frac{\mathbf{1}}{\mathbf{2}}\right)\boldsymbol{k}=(\mathbf{0.6185})^{\boldsymbol{m}/\boldsymbol{n}}$$

or

$$\boldsymbol{k}=\frac{\boldsymbol{m}}{\boldsymbol{n}}\log_{\boldsymbol{e}}\mathbf{2}$$

As $\boldsymbol{m}$ grows in proportion to $\boldsymbol{n}$, the false positive rate decreases.
Some reasonable values for $\boldsymbol{m}$, $\boldsymbol{n}$ and $\boldsymbol{k}$ are:

a.$m/n\ =\ 6, k\ =\ 4, Pr\ [False\ positive]\ =\ 0.05$
b.$min\ =\ 8, k\ =\ 6, Pr\ [False\ positive]\ =\ 0.02$

Bloom filters have some nice properties that make it ideal for data streaming applications. When an item is inserted into the filter, all the locations can be updated “blindly” in parallel. The reason for this is that the only operation performed is to make a bit 1, so there can never be any data race conditions. Also for an insertion, there need to be a constant (that is, k) number of writes and no reads. Again, when searching for an item, there are at most a constant number of reads. These properties make Bloom filters very useful for high-speed applications, where memory access can be the bottleneck.

5.Explain the Flajolet-Martin Algorithm in detail[CO3-L2]

The Flajolet-Martin Algorithm

The principle behind the FM algorithm is that, it is possible to estimate the number of distinct elements by hashing the elements of the universal set-to a bit-string that is sufficiently long.

This means that the $\boldsymbol{e}$ length of the bit-string must be such that there are more possible results of the hash function than there are elements of the universal set.

Before we can estimate the number of distinct elements, we first choose an upper boundary; of distinct elements $\boldsymbol{m}$. This boundary gives us the maximum number of distinct elements that we might be able to detect. Choosing $\boldsymbol{m}$ to be too small will influence the precision of our measurement. Choosing $\boldsymbol{m}$ that is far bigger than the number of distinct elements will only use too much memory. Here, the memory that is required is $\boldsymbol{O}|\boldsymbol{log}\ (\boldsymbol{m})|$.

For most applications, $\boldsymbol{64-bit}$ is a sufficiently large bit-array. The array needs to be initialized to zero. We will then use one or more adequate hashing functions. These hash functions will map the input to a number that is representable by our bit-array. This number will then be analyzed for each record. If the resulting number contained $\boldsymbol{k}$ trailing zeros, we will set the $\boldsymbol{k^{th}}$ bit in the bit array to one.

Finally, we can estimate the currently available number of distinct elements by taking the index of the first zero bit in the bit-array. This index is usually denoted by R. We can then estimate the number of unique elements N to be estimated by $\boldsymbol{N = 2^R}$. The algorithm is as follows:

Algorithm

Pick a hash function $\boldsymbol{h}$ that maps each of the $\boldsymbol{n}$ elements to at least $\boldsymbol{log_2 n}$ bits-

For each stream element $\boldsymbol{n}$ let $\boldsymbol{r(a)}$ be the number of trailing 0s in $\boldsymbol{h(a)}$.

Record $\boldsymbol{R}$ = the maximum $\boldsymbol{r(a)}$ seen.

Estimate = $\boldsymbol{2^R}$,

Example

$\boldsymbol{r(a)}$ = position of first 1 counting from the right; say $\boldsymbol{h(d) = 12}$, then 12 is 1100 in binary, so $\boldsymbol{r(a) = 2}$

Example

Suppose the stream is 1, 3, 2, 1, 2, 3, 4, 3, 1, 2, 3, 1 .... Let $\boldsymbol{h(x) = 3x + 1\ mod\ 5}$.

So the transformed stream ($\boldsymbol{h}$ applied to each item) is 4,5,2,4,2,5,3,5,4,2,5,4

Each of the above element is converted into its binary equivalent as 100, 101, 10, 100, 10, 101, 11, 101, 100, 10,.101, 100

We compute $\boldsymbol{r(a)}$ of each item in the above stream: 2, 0, 1, 2, 1, 0, 0, 0, 2, 1, 0, 2

So $\boldsymbol{R = maximum\ r(a)}$, which is 2. Output $\boldsymbol{2^2 = 4}$.

A very simple and heuristic intuition as to why Fiajokt-Martin works can be explained as follows:

1.h(a) hashes a with equal probability to any of N values
2.Then h(a) is a sequence of log_2N bits, where 2^{-r} fraction of all a s have a tail of i zeros
•About 50% of as hash to ***0
•About 25% of as hash to **00
•So, if we saw the longest tail of $r = 2$ (i.e., item hash ending *100) then we have probably seen about four distinct items so far
3.So, it takes to hash about 2^r items before we see one with zero-suffix of length r.
More formally we can see that the algorithm works, because the probability that a given hash function $h(a)$, ends in at least r zeros is 2^{-r}. In case of m different elements, the probability that $R \geq r$ (R is max. tail length seen so far) is given by

$$P(R \geq r) = 1 - (1 - 2^{-r})^m$$

Since 2^{-r} is small, $1 - (1 - 2^{-r})m \approx 1 - e^{-m2^{-r}}$
If $2^r \ll m$, $1 - (1 - 2^{-r})^m \approx 1 - (1 - m2^{-r}) \approx m/2r \approx 0$.
if $2^r \ll m, 1 - \left(1 - 2^{-r}\right)^m \approx 1 - e^{-m2^{-r}} \approx 1$
Thus, 2^R will almost always be around m.

Variations to the FM Algorithm

There are reasons why the simple FM algorithm won't work with just a single hash function. The expected value of $[2^R]$ is actually infinite. The probability halves when R is increased to $R + 1$, however, the value doubles. In order to get a much smoother estimate, that is also more reliable, we can use many hash functions. Another problem with the FM algorithm in the above form is that the results vary a lot. A common solution is to run the algorithm multiple times with k different hash-functions, and combine the results from the different runs.
One idea is to take the mean of the k results together from each hash-function, obtaining a single estimate of the cardinality. The problem with this is that averaging is very susceptible to outliers (which are likely here).
A different idea is to use the median which is less prone to be influenced by outliers. The problem with this is that the results can only take form as some power of **2**. Thus, no matter how many hash functions we use, should the correct value of m be between two powers of **2**, say **400**, then it will be impossible to obtain a close estimate. A common solution is to combine both the mean and the median:

1.Create $k \cdot l$ hash-functions and split them into k distinct groups (each of size l).
2.Within each group use the median for aggregating together the l results
3.Finally take the mean of the k group estimates as the final estimate.

Sometimes, an outsized $2R$ will bias some of the groups and make them too large. However, taking the median of group averages will reduce the influence of this effect almost to nothing.

Moreover, if the groups themselves are large enough, then the averages can be essentially any number, which enable us to approach the true value m as long as we use enough hash functions.

Groups should be of size at least some small multiple of $log_2 m$.

6.Discuss in detail the role of Decaying Windows in data stream analysis[CO3-L2]

Decaying Windows

Pure sliding windows are not the only way by which the evolution of data streams can be taken into account during the mining process. A second way is to introduce a decay factor into the computation. Specifically, the weight of each transaction is multiplied by a factor of $f < 1$, when a new transaction arrives. The overall effect of such an approach is to create an exponential decay function on the arrivals in the data stream. Such a model is quite effective for evolving data stream, since recent transactions are counted more significantly during the mining process. Specifically, the decay factor is applied only to those itemsets whose counts are affected by the current transaction. However, the decay factor will have to be applied in a modified way by taking into account the last time that the itemset was touched by an update. This approach works because the counts of each itemset reduce by the same decay factor in each iteration, as long as a transaction count is not added to it. Such approach is also applicable to other mining problems where statistics are represented as the sum of decaying values.

We discuss a few applications of decaying windows to find interesting aggregates over data streams.

The Problem of Most-Common Elements

Suppose we have a stream whose elements are the movie tickets purchased all over the world, with the name of the movie as part of the element. We want to keep a summary of the stream that is the most popular movies “currently.” While the notion of “currently” is imprecise, intuitively, we want to discount the popularity of an older movie that may have sold many tickets, but most of these decades ago. Thus, a newer movie that sold n tickets in each of the last 10 weeks is probably more popular than a movie that sold 2n tickets last week but nothing in previous weeks.

One solution would be to imagine a bit stream for each movie. The i^{th} bit has value 1 if the i^{th} ticket is for that movie, and 0 otherwise. Pick a window size N, which is the number of most recent tickets that would be considered in evaluating popularity. Then, use the method of the DGIM algorithm to estimate the number of tickets for each movie, and rank movies by their estimated counts.

This technique might work for movies, because there are only thousands of movies, but it would fail if we were instead recording the popularity of items sold at Amazon, or the rate at which different Twitter-users tweet, because there are too many Amazon products and too many tweeters. Further, it only offers approximate answers.

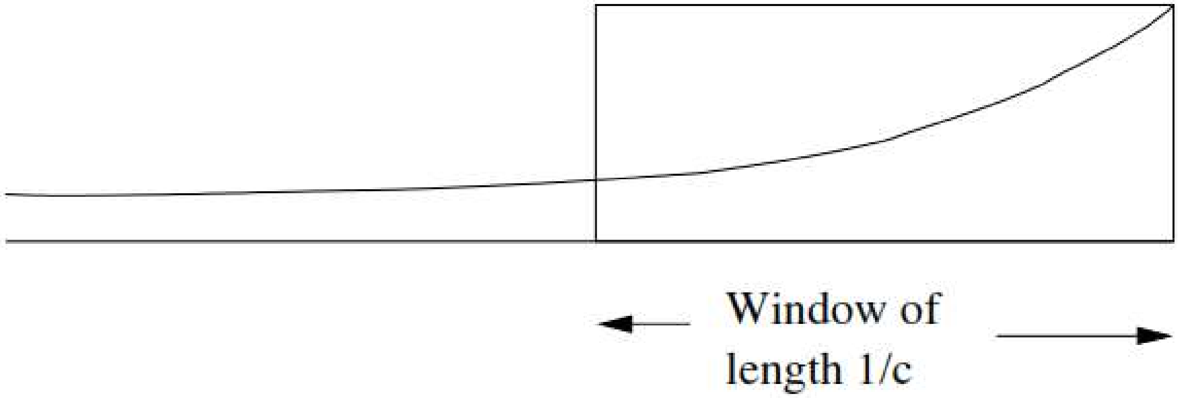
Describing a Decaying Window

One approach is to re-define the question so that we are not asking for a simple count of 1s in a window. We compute a smooth aggregation of all the 1s ever seen in the stream, but with decaying weights. The further in the past a 1 is found the lesser is the weight given to it. Formally, let a stream currently consist of the elements $a_1, a_2, \ldots, a_t$where a_1 is the first element to arrive and a_t is the current element. Let $\boldsymbol{c}$ be a small constant, such as ***10⁻⁶*** or ***10⁻⁹***. Define the exponentially decaying window for this stream to be the sum

$$\sum_{i=0}^{t-1} a_{t-1}(1-c)^t$$

The effect of this definition is to spread out the weights of the stream elements as far back in time as the stream goes. In contrast, a fixed window with the same sum of the weights, $\mathbf{1/c}$, would put equal weight $\mathbf{1}$ on each of the most recent He elements to arrive and weight $\mathbf{0}$ on all previous elements. This is illustrated in Figure below

It is much easier to adjust the sum in an exponentially decaying window than in a sliding window of fixed length. In the sliding window, we have to somehow take into consideration the element that falls out of the window each time a new element arrives. This forces us to keep the exact elements along with the sum, or to use some approximation scheme such as DGIM. But in the case of a decaying window, when a new element a_{t+1} arrives at the stream input, all we need to do is the following:



Illustrating decaying windows.

1.Multiply the current sum by $\mathbf{1-c}$.

2.Add a_{t+1}.

The reason this method works is that each of me previous elements has now moved one position further from the current element, so its weight is multiplied by $\mathbf{1-c}$. Further, the weight on the current element is $(\mathbf{1-c})^{\mathbf{0}}=\mathbf{1}$, so adding Add $\boldsymbol{a_{t+1}}$is the correct way to include the new elements contribution.

Now we can try to solve the problem of finding the most popular movies in a stream of ticket sales. We can use an exponentially decaying window with a constant c, say $\boldsymbol{10^{-9}}$. We are approximating a sliding window that holds the last one billion ticket sales. For each movie, we can imagine a separate stream with a 1 each time a ticket for that movie appears in the stream, and a 0 each time a ticket for some other movie arrives. The decaying sum of the $\mathbf{1}\boldsymbol{s}$; thus, it measures the current popularity of the movie.

To optimize this process, we can avoid performing these counts for the unpopular movies. If the popularity score for a movie goes below 1, its score is dropped from the counting. A good threshold value to use is (1/2).

When a new ticket arrives on the stream, do the following:

1.For each movie whose score is currently maintained multiply its score by (1 - c)

2.Suppose the new ticket is for movie M. If there is currently a score for M, add 1 to that score. If there is no score for M, create one and initialize it to 1.

3.If any score is below the threshold 1/2, drop that score.

A point to be noted is that the sum of all scores is $\mathbf{1}/\boldsymbol{c}$. Thus, there cannot be more than $\mathbf{2}/\boldsymbol{c}$ movies with score of $\mathbf{1/2}$ or more, or else the sum of the scores would exceed lie. Thus, lie is a limit on the number of movies being counted at any time. Of course in practice, the number of actively counted movies would be much less than $\mathbf{2}/\boldsymbol{c}$. If the number of items is very large then other more sophisticated techniques are required.

7.Explain the generic design of RTAP in detail[CO3-L2]

Real-Time Analytics Platform (RTAP)

Real-time analytics makes use of all available data and resources when they are needed. It consists of dynamic analysis and reporting, based on data entered into a system less than one minute before the actual time of use. Real-time denotes the ability to process data as it arrives, rather than storing the data and retrieving it at some point in the future.

For example, consider an e-merchant like Flipkart or Snapdeal; real time means the time elapsed from the time a customer enters the website to the time the customer logs out. Any analytics procedure, like providing the customer with recommendations or offering a discount based on current value in the shopping car, etc., will have to be done within this timeframe which may be a about 15 minutes to an hour.

But from the point of view of a military application where there is constant monitoring say of the air space, time needed to analyze a potential threat pattern and make decision maybe a few milliseconds.

“Real-Time Analytics” is thus discovering meaningful patterns in data for something urgent. There are two specific and useful types of real-time analytics - On-Demand and Continuous.

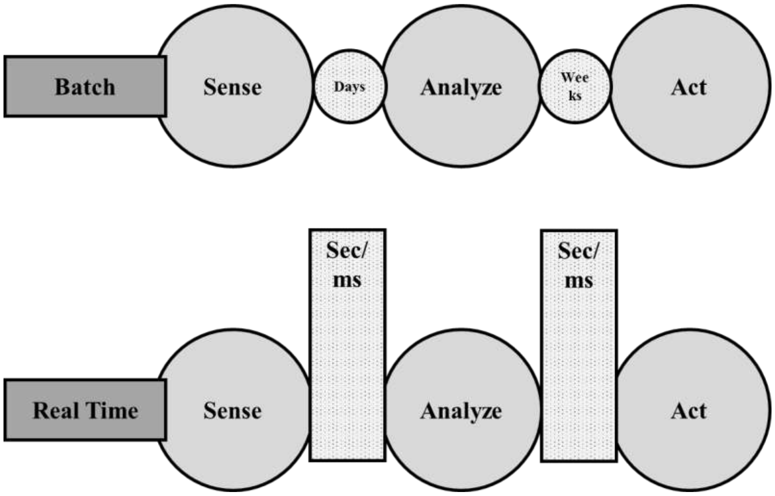
1.On-Demand Real-Time Analytics is reactive because it waits for users to request a query and then delivers the analytics. This is used when someone within a company needs to take a pulse on what is happening right this minute. For instance, a movie producer may want to monitor the tweets and identify sentiments about his movie on the first day first show and be prepared for the outcome.

2.Continuous Real-Time Analytics is more proactive and alerts users with continuous updates in real time. The best example could be monitoring the stock market trends and provide analytics to help users make a decision to buy or sell all in real time.

Real-Time Analytics Applications

Analytics falls along a spectrum. On one end of the spectrum sit hatch analytical applications, which are used for complex, long-running analyses. They tend to have slower response times (up to minutes, hours, or days) and lower requirements for availability.

Examples of batch analytics include Hadoop- based workloads. On the other end of the spectrum sit real-time analytical applications, which provide lighter-weight analytics very quickly. Latency is low (sub-second) and availability requirements are high (e.g., 99.99%). Figure below illustrates this.



Batch versus real-time analytics.

Example applications include:

1.Financial Services: Analyse ticks, tweets, satellite imagery, weather trends, and any other type of data to inform trading algorithms in real time.

2.Government: Identify social program fraud within seconds based on program history, citizen profile, and geospatial data.

3.E-Commerce Sites: Real-time analytics will help to tap into user preferences as people are on the site or using a product. By knowing what users like at run time can help the site to decide relevant content to be made available to that user. This can result in a better customer experience overall leading to increase in sales. Let us take a look at how this works for these companies. For example, Amazon recommendations change after each new product you view so that they can upsell customers throughout the session. Real-time recommendations create a personal shopping experience for each and every customer. With more insight into, their customers on an individual level, Amazon is able to effectively upsell and cross-sell products at every interaction point.

4.Insurance Industry: Digital channels of customer interaction (such as online channels) as well as conversations online (such as social media) have created new streams of real-time event data. Insurance, being a data-rich industry and a high customer lifetime value business, can gain immensely from real-time analytics. The following are a few scenarios where an insurance firm can benefit from real-time analytics:

•A prospective customer visits the website looking to get a quote. Real-time analytics can be used to predict the propensity of the customer to leave the site without applying for a quote. This, in turn, can be used to trigger actions like free consultation, some more schemes, etc.

•In the insurance industry, fast-tracking of claims improves customer satisfaction significantly. However, this can increase the risk of fraud. Real-time analytics can be used to reduce the risk of fraud even while accelerating the speed of processing claims.

•Some auto insurers are already collaborating with automobile firms to gather real-time information from vehicles on a continuous basis. With GPS-enabled telemetry devices in place, insurers can devise innovative policies where the premium could be thought of as a car gas-tank that is filled up at a station, just as the actual consumption of gas changes dynamically based on a variety of conditions, the premium can be “consumed” in real time based on driving behaviour - say if one drives safely premium must be extended longer than when driving rashly.

The list of applications for real-time analytics is endless. At the end of this section we shall discuss two popular applications, real-time sentiment analysis and real-time stock predictions, in greater detail.

Generic Design of an RTAP

Companies like Facebook and Twitter generate petabytes of real-time data. This data must be harnessed to provide real-time analytics to make better business decisions. Further in today’s context, billions of devices are already connected to the internet, with more connecting each day. With the evolution of the Internet of Things (IOT), we have a large number of new data sources such as smart meters, sensors and wearable medical devices. Real-time analytics will leverage information from ail these devices to apply analytics algorithms and generate automated actions within milliseconds of a trigger.
To create an environment where you can do “Real-Time Analytics”, the following three aspects of data flows to your system are important:

1.Input: An event happens (new sale, new customer, someone enters a high security zone etc.).
2.Process and Store Input: Capture the data of the event, and analyze the data without leveraging resources that are dedicated to operations.
3.Output: Consume this data without disturbing operations (reports, dashboard, etc.).

Unit – IV

Introduction

Part – A

1.Define the term Market Basket. [CO4-L2]

“Market-Baskets” is an abstraction that models any many-many relationship between two concepts: “items” and “baskets”. Since the term a “market-basket” has its origin in retail applications, it is sometimes called “transactions”.

2.What are share and support of an Itemset? [CO4-L1]

Support, supp(X) of an itemset X is the ratio of transactions in which an itemset appears to the total number of transactions.

Share of an itemset is the ratio of the count of items purchased together to the total count of items purchased in all transactions.

3.What is a Frequent Itemset? [CO4-L1]

A set of items that appears in many baskets is said to be “frequent.” Assume there is a number s, called the support threshold. If I is a set of items, the support for I is the number of baskets for which I is a subset. We say I is frequent if its support is s or more.

4.What is the association rule of frequent items? [CO4-L1]

Associative rule is often represented as a collection of if–then rules. The form of an association rule is $I \rightarrow j$, where I is a set of items and j is an item. The implication of this association rule is that if all of the items in I appear in some basket, then j is “likely” to appear in that basket as well.

5.Define interest of an associative rule.

The interest of an association rule $I \rightarrow j$ is the difference between its confidence and the fraction of baskets that contain j.

If I has no influence on j, then we would expect that the fraction of baskets including I that contain j would be exactly the same as the fraction of all baskets that contain j.

6.What is a singleton? [CO4-L1]

In statistics, a singleton is a set with a single element. An example is the set $\boldsymbol{S}$ of all items $\boldsymbol{s}$ that are neither positive nor negative.

$$\boldsymbol{S} = \{\boldsymbol{0}\}$$

7.What is the role of monotonicity? [CO4-L1]

Monotonicity provides a way to compact the information about frequent itemsets. If we are given a support threshold $\boldsymbol{s}$, then we say an itemset is *maximal* if no superset is frequent.

8.What is an A-Priori Algorithm? [CO4-L1]

The **A-Priori Algorithm** is an influential **algorithm** for mining frequent itemsets for boolean association rules. **A-Priori** uses a "bottom up" approach, where frequent subsets are extended one item at a time.

The A-Priori Algorithm is designed to reduce the number of pairs that must be counted, by performing two passes over data, rather than one pass.

9.What is a frequent-items table? [CO4-L1]

The second pass of A-Priori, creates a new numbering from $\boldsymbol{1\ to\ m}$ only for the frequent items. This table is an ***array indexed*** $\boldsymbol{1\ to\ n}$, and the entry for $\boldsymbol{i}$ is either $\boldsymbol{0}$, if item $\boldsymbol{i}$ is not frequent, or a unique integer in the range $\boldsymbol{1\ to\ m}$ if item $\boldsymbol{i}$ is frequent. This table is called as the frequent-items table.

<u>Part – B</u>

1.Write a short note on Market-Basket Model with a suitable example.

Market-Basket Model[CO4-L1]

“Market-Baskets” is an abstraction that models any many-many relationship between two concepts: “items” and “baskets”. Since the term a “market-basket” has its origin in retail applications, it is sometimes called “transactions”.

Each basket consists of a set of items (an itemset), and usually we assume that the number of items in a basket is small - much smaller than the total number of items. The number of baskets is usually assumed to-be very large, bigger than what can fit in main memory.

Items need not be “contained” in baskets. We are interested in the co-occurrences of items related to a basket, not vice-versa. For this purpose, we define basket data in general terms. Let $I = \{i_1, \ldots, i_k\}$ be a set of k elements, called items.

Let $\boldsymbol{B} = \{\boldsymbol{b_1}, \ldots, \boldsymbol{b_n}\}$ be a set of $\boldsymbol{n}$ subsets of $\boldsymbol{I}$. We call each $\boldsymbol{b_i} \subset \boldsymbol{I}$ a basket of items. For example, in a retail market-basket application, the set $\boldsymbol{I}$ consists of the items stocked by a retail outlet and each basket is the set of purchases from one register transaction; on the other hand, in a document basket application, the set $\boldsymbol{I}$ contains all dictionary words and proper nouns, while each basket is a single document in the corpus and each basket consists of all words that Occur in that document.

Frequent-Itemset Mining

Let $\boldsymbol{I} = \{\boldsymbol{i_1}, \ldots, \boldsymbol{i_k}\}$ be a set of items. Let $\boldsymbol{D}$, the task-relevant data, be a set of database transactions where each transaction $\boldsymbol{T}$ is a set of items such that $\boldsymbol{T} \subseteq \boldsymbol{I}$.

Each transaction is associated with an identifier, called TID.

Let $\boldsymbol{A}$ be a set of items.

A transaction $\boldsymbol{T}$ is said to contain $\boldsymbol{A}$ if and only if $\boldsymbol{A} \subseteq \boldsymbol{T}$

A set of items is referred to as an itemset. An itemset that contains $\boldsymbol{k}$ items is a $\boldsymbol{k}$-itemset. For example, consider a computer store with computer-related items in its inventory.

The set ***{computer, anti-virus software, printer, flash-drive}*** is a 4-itemset.

The occurrence frequency of an itemset is the number of transactions that contain the itemset. This is also known, simply, as the frequency, support count, or count of the itemset.

We can call an itemset $\boldsymbol{I}$ a “frequent itemset” only if its support count is sufficiently large. We prescribe a minimum support $\boldsymbol{s}$ and any $\boldsymbol{I}$ which has support greater than or equal to $\boldsymbol{s}$ is a frequent itemset.

Applications

A supermarket chain may have 10,000 different items in its inventory. Daily millions of customers will use their shopping carts (“market-baskets”) to the checkout section where the cash register records the set of items they purchased and give out a bill.

Each bill represents one market-basket or one transaction.

In this scenario, the identity of the customer is not strictly necessary to get useful information from the data. Retail organizations analyze the market-basket data to learn what typical customers buy together.

Consider a retail organization that has several floors, where soaps are in floor 1 and items like towels and other similar goods are in floor 10.

Analysis of the market-basket shows a large number of baskets containing both soaps and towels. This information can be used by the supermarket manager in several ways:

Actually, many people walk from where the soaps, are to where the towels is which means they have to move form floor 1, catch the elevator to move to floor 10. The manger could choose to put a small shelf in floor 1; consisting of an assortment of towels and some other bathing accessories that, might also be bought along with soaps and towels, for example, shampoos, bath mats etc. Doing so can generate additional “on the spot” sales.

The store can run a sale on soaps and at the same time raise the price of towels (without advertising that fact). People will come to the store for the cheap soaps, and many will need towels too. It is not worth the trouble to go to another store for cheaper towels, so they buy that too. The store makes back on towels what it loses on soaps, and gets more customers into the store.

While the relationship between soaps and towels seems somewhat obvious, market-basket analysis may identify several pairs of items that occur together frequently but the connections between them may be less obvious.

For example, the analysis could show chocolates being bought with movie CDs. But we need some rules to decide when a fact about co-occurrence of sets of items can be useful. Firstly any useful set (need not be only pairs) of items must be bought by a large number of customers. It is not even necessary that there be any connection between purchases of the items, as long as we know that: lots of customers buy them.

2.Explain the two-pass A-Priori Algorithm in detail[CO4-L2]

The A-Priori Algorithm

The **A-Priori Algorithm** is an **influential algorithm** for mining frequent itemsets for Boolean association rules.

A-Priori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation, and groups of candidates are tested against the data.

If we have enough main memory to count all pairs; using either triangular matrix or triples, then it is a simple matter to read the file of baskets in a single pass.

For each basket, we use a double loop to generate all the pairs. Each time we generate a pair, we add 1 to its count. At the end, we examine all pairs to see which have counts that are equal to or greater than the support threshold s; these are the frequent pairs.

However, this naive approach fails if there are too many pairs of items to count them all in the main memory.

The Apriori algorithm uses the monotonicity property to, reduce the number of pairs that must be counted, at the expense of performing two passes over data, rather than one pass.

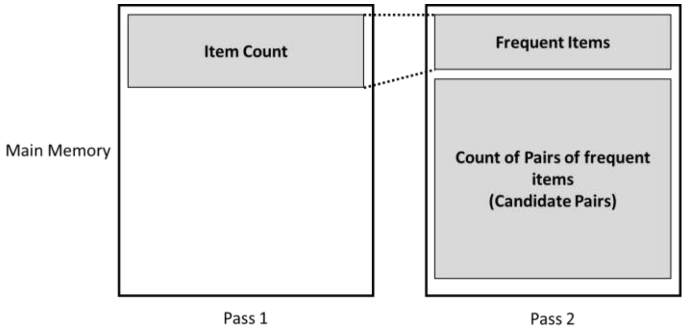
The Apriori algorithm for finding frequent pairs is a two-pass algorithm that limits the amount of main memory needed by using the downward-closure property of support to avoid counting pairs that will turn out to be infrequent at the end.

Let $\boldsymbol{s}$ be the minimum support required. Let $\boldsymbol{n}$ be the number of items.

In the ***first pass***, we ***read the baskets and count in main memory the occurrences of each item***. We then remove all items whose frequency is lesser than $\boldsymbol{s}$ to get the set of frequent items. This requires memory proportional to $\boldsymbol{n}$.

In the ***second pass***, we ***read the baskets again and count in main memory only those pairs where both items are frequent items***. This pass will require memory proportional to square of frequent items only (for counts) plus a list of the frequent items (so you know what must be counted).

Figure below indicates the main memory in the two passes of the Apriori algorithm.



Main memory map in A-Priori algorithm.

The First Pass of A-priori

Two tables are created in this pass. The first table, if necessary, translates item names into integers from 1 to n, as it will be optimal to use numbers than long varying length string names.

The other table is an array of counts; the **i**th array element counts the occurrences of the item numbered ***i***.

Initially, the counts for all the items are 0. As we read baskets, we look at each item in the basket and translate its name into an integer. Next, we use that integer to index into the array of counts, and we add 1 to the integer found there.

After the first pass, we examine the counts of the items to determine which of them are frequent as singletons. It will normally be the case that the number of frequent singletons will definitely be much lesser than the number of items. Thus, we employ a trick for the next pass.

For the second pass of A-priori, we create a new numbering from 1 to m for just the frequent items. This table is an array indexed 1 to n, and the entry for i is either 0, if item i is not frequent, or a unique integer in the range 1 to m if item i is frequent. We shall refer to this table as the ***frequent-items table***.

The Second Pass of A-priori

During the second pass, we count all the pairs that consist of two frequent items. The space required on the second pass is $2m^2$ bytes, rather than $2n^2$ bytes, if we use the triangular- matrix method for counting.

Renumbering of just the frequent items is necessary if we are to use a triangular matrix of the right size. The complete set of main-memory structures used in the first and second passes of this improved Apriori algorithm is shown in the figure below.

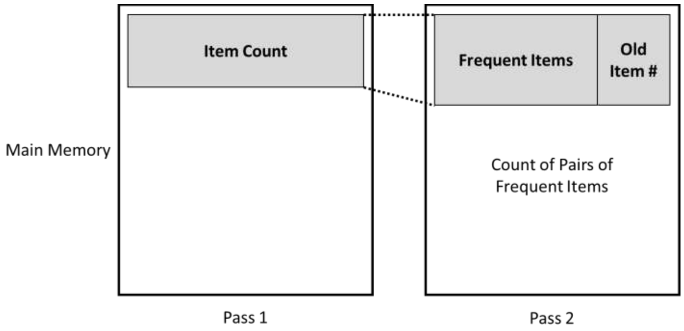
The benefit of eliminating infrequent items;

•If only half the items are frequent we need one quarter of the space to count.
•If we use the triples method, we need to count only those pairs of two frequent items that occur in at least one basket.

The mechanics of the second pass are as follows:

1.For each basket, look in the frequent-items table to see which of its items are frequent.
2.In a double loop, generate all pairs of frequent items in that basket.
3.For each such pair, add one to its count in the data structure used to store counts.

Finally, at the end of the second pass, examine the structure of counts to determine which pairs are frequent.

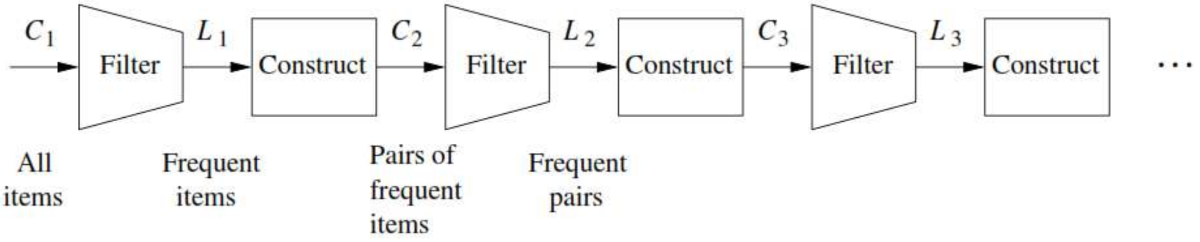


Improved A-priori algorithm.

Finding All Frequent Itemsets the same technique as used for finding frequent pairs without counting all pairs can be extended to find larger frequent itemsets without an exhaustive count of all sets. In the Apriori algorithm, one pass is taken for each set-size k. If no frequent itemsets of a certain size are found, then monotonicity tells us there can be no larger frequent itemsets, so we can stop. The pattern of moving from one size k to the next size k + 1 can be summarized as follows. For each size k, there are two sets of itemsets:

1.Q is the set of candidate itemsets of size k - the itemsets that we must count in order to determine whether they are in fact frequent.
2.is the set of truly frequent itemsets of size k.

The pattern of moving from one set to the next and one size to the next is depicted in Fig. 5.



3.Illustrate how PARK-CHEN-YU Algorithm handles large Datasets. [CO4-L2]

Handling Larger Datasets in Main Memory

The initial candidate set generation for the large 2-itemsets is the key to improve the performance of the A-priori algorithm. This algorithm is fine as long as the step for the counting of the candidate pairs has enough memory that it can be accomplished without excessive moving of data between disk and main memory.

Another performance related issue is the amount of data that has to be scanned during large itemset discovery.

A straightforward implementation would require one pass over the database of all transactions for each iteration.

Note that as k increases, not only is there a smaller number of large k itemsets, but here are also fewer transactions containing any large k itemsets. Reducing the number of transactions to be scanned and trimming the number of items in each transaction can improve the data mining efficiency in later stages.

Several algorithms have been proposed to cut down on the size of candidate set C2. The Park-Chen-Yu (PCY) algorithm, takes advantage of the fact that in the first pass of Apriori there is typically lots of main memory not needed for the counting of single items.

Algorithm of Park-Chen-Yu

In the Apriori algorithm, in each pass we use the set of large itemsets L- to form the set of candidate large itemsets ***C_i+1*** by joining ***L_i*** with ***L_i*** on ***(i-1)*** common items for the next pass.

We then scan the database and count the support of each item set in Ci+ j so as to determine + l.

As a result, in general, the more itemsets in C- the higher the processing cost for determining ***L_i*** will be.

An algorithm designed by Park, Chen, and Yu called as ***DHP (Direct Hashing and Pruning)*** is designed to reduce the number of itemsets to be explored in ***Q*** in initial iterations significantly.

The corresponding processing cost to determine from C- is, therefore, reduced.

The PCY algorithm uses the technique of hashing to filter out unnecessary itemsets for next candidate itemset generation.

When the support of candidate ***k-itemsets is counted by scanning the database***, PCY accumulates information about candidate ***(k+1)-itemsets*** in advance in such a way that all possible (k-+ l) itemsets of each transaction after some pruning are hashed to a hash table.

Each bucket in the hash table consists of a number to represent how many itemsets have been hashed to this bucket thus far. We note that based on the resulting hash table a hit vector can be constructed, where ***the value of one bit is set to 1 if the number in the corresponding entry of the hash table is greater than or equal to r***, the minimum support value.

This ***bit vector*** can be used later to greatly reduce the number of itemsets in ***Q***

The PCY algorithm ***does not need any extra space*** to store the hash tables or the bit vectors. It exploits the observation that there may be much unused space in main memory on the first pass.

If there are a million items and gigabytes of main memory, which is quite typical these days, we ***do not need more than 10% of the main memory for the two tables***, one a translation table from item names to small integers and second an array to count those integers.

The PCY algorithm then uses the still available space for an unusual type of hash table. The buckets of this **hash table store integers** rather than sets of keys. Pairs of items are hashed to buckets of this hash table.

As we examine a basket during the first pass, we not only add 1 to the count for each item in the basket, but also generate all possible pairs (2-itemsets). We hash each pair using a hash function, and we add 1 to the bucket into which that pair hashes.

Note that the pair itself does not go into the bucket; the pair only affects the single integer in the bucket.

Algorithm

```
FOR (each basket):
FOR (each item in the basket):
add 1 to items count;
FOR (each pair of items):
{hash the pair to a bucket;
add 1 to the count for that bucket;}
```

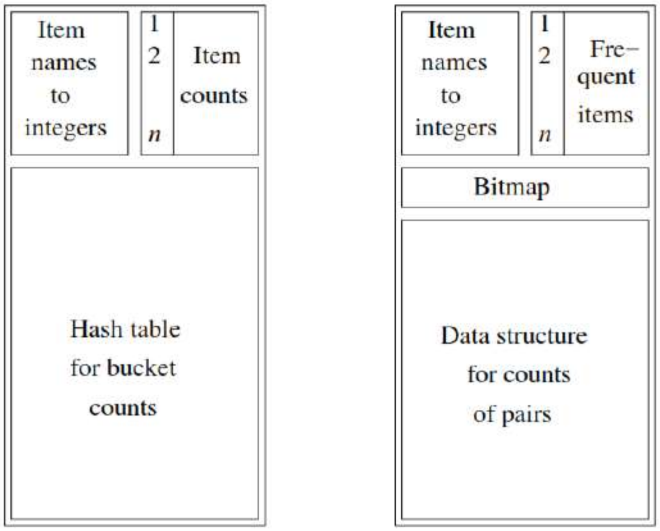
At the end of the first pass, each bucket has a count, which is the sum of the counts of all the pairs that hash to that bucket. If the ***count of a bucket is at least as great as the support threshold s, it is called a frequent, bucket***. We can say nothing about the pairs that hash to a frequent bucket; they could all be frequent pairs from the information available to us.

But if the count of the bucket is less than (an *infrequent bucket*), we know no pair that hashes to this bucket can be frequent, even if the pair consists of two frequent items. This fact gives us an advantage on the second pass.

In Pass 2 we only count pairs that hash to frequent buckets.

Depending on the data and the amount of available main memory, there may or may not be a benefit in using the hash table on pass 1. In the worst case, all buckets are frequent, and the PCY algorithm counts exactly the same pairs as Apriori does on the second pass.

However, most of the buckets will be infrequent. In that case, PCY reduces the memory requirements of the second pass.



Pass 1 Pass 2

4.Explain how SON algorithm outperforms A-Priory Algorithm with Proof. [CO4-L2]

Limited Pass Algorithms

The algorithms for frequent itemsets discussed so far use one pass for each size of itemset we investigate. Recall that Apriori algorithm requires k passes to find frequent itemsets of size k. If the main memory is too small to hold the data and the space needed to count frequent itemsets of one size, there does not seem to be any way to avoid k passes to compute the exact collection of frequent itemsets.

In many applications, however, finding each and every frequent itemset is not important. In many cases, it may be impractical to consider too many frequent itemsets. All of them cannot result in some action taken by the enterprise. For instance, if we are looking for items purchased together at a supermarket, we are not going to run a sale or change the layout based on every frequent itemset we find. So it is quite sufficient to find most, but not all of the frequent itemsets.

In this section we shall discuss some approximate algorithms that have been proposed to find all or most frequent itemsets using at most two passes. We begin with a simple sampling based algorithm. To get more accurate answers we discuss the Algorithm of Savasere, Omiecinski and Navathe (SON) which use two passes to get the exact answer. Further this algorithm can be easily implemented by MapReduce. Finally, Toivonens algorithm uses two passes on average, gets an exact answer, but may, rarely, not terminate in finite time.

The Algorithm of Savasere, Omiecinski and Navathe

The idea behind Savasere, Omiecinski and Navathe (SON) Algorithm also called as Partition algorithm is summarized below. If we are given a database with a small number of potential large itemsets, say, a few thousand, then the support for all of them can be tested in one scan and the actual large itemsets can be discovered. Clearly, this approach will work only if the given set contains all actual large itemsets.

Partition algorithm accomplishes this in two scans of the database. In one scan, it generates a set of all potentially large itemsets by scanning the database once. This set is a superset of all large itemsets, that is, it may contain false positives. But no false negatives are reported. During the second scan, counters for each of these itemsets are set up and their actual support is measured in one scan of the database.

The SON algorithm divides the database into non-overlapping subsets; these are individually considered as separate databases and all large itemsets for that partition, called local frequent item- sets, are generated in one pass.

The Apriori algorithm can then be used efficiently on each partition if it fits entirely in main memory. Partitions are chosen in such a way that each partition can be accommodated in the main memory. As such, a partition is read only once in each pass.

The only limitation with the partition method is that the minimum support used for each partition has a slightly different meaning from the original value. The minimum support is based on the size of the partition rather than the size of the database for determining local frequent (large) itemsets. The actual support threshold value is the same as given earlier, but the support is computed only for a partition.

At the end of pass one, we take, the union of all frequent itemsets from each partition. This forms the global candidate frequent itemsets for the entire database. When these lists are merged, they may contain some false positives.

That is, some of the itemsets that are frequent (large) in one partition may not qualify in several other partitions and hence may not exceed the minimui support when the original database is considered. Note that there are no false negatives; no large itemsets will be missed.

The global candidate large itemsets identified in pass one are verified in pass two; that is, their actual support is measured for the entire database.

At the end of phase two, ail global large itemsets are identified. The Partition algorithm lends itself naturally to a parallel or distributed implementation for better efficiency. Further improvements to this algorithm have been suggested in the literature.

The implementation details for the SON algorithm are as follows:

1.The first pass of the SON algorithm is very similar to the first pass of the random sampling algorithm.
2.Instead of randomly sampling the dataset with a fixed probability p, the SON algorithm divided the dataset into chunks of size $\mathbf{1/p}$. It then loads the baskets into the main memory and runs th Apriori or one of its improvements in the main memory. Given a support threshold s, an itemsc is frequent in a chunk if its support is at least $\boldsymbol{ps}$ in the chunk.

The fact that this algorithm does not generate any false negatives arises from the following theorer and its corollary.

Theorem
If an itemset $\boldsymbol{I}$ is infrequent in every chunk, $\boldsymbol{I}$ is infrequent in the full dataset.

Proof: Since $\boldsymbol{I}$ is infrequent in every chunk, supp $(\boldsymbol{I}) < ps$ in each chunk. Since there are $\mathbf{1/p}$ chunks

$$supp\ (I) < \left(\frac{1}{p}\right) ps$$

in the full dataset.

The contrapositive of the theorem follows:

Corollary: If an itemset $\boldsymbol{I}$ is frequent in the full dataset, $\boldsymbol{I}$ is frequent in at least one chunk.

This guarantees that all truly frequent itemsets are in the set of itemsets that are frequent in at lea one chunk. Therefore, if we store all itemsets that are frequent in at least one chunk, we know that v will not have any false negative.

The second pass of the SON algorithm; is exactly the same as the second pass of the random san pling algorithm: the algorithm goes over the full dataset and makes sure that all the itemsets that are frequent in at least one chunk are frequent in the full dataset.

5.What are Frequent Items in a stream and how can they be counted? Explain[CO4-L2]

Counting Frequent Items in a Stream

We had seen that data streams possess distinct computational characteristics, such as unknown o unbounded length, possibly very fast arrival rate, inability to backtrack over previously arrived item (only one sequential pass over the data is permitted), and a lack of system control over the order ii which the data arrive. As data streams are of unbounded length, it is intractable to store the entire data into main memory.

Finding frequent itemsets in data streams lends itself to many applications of Big Data. In man such applications, one is normally interested in the frequent itemsets in the recent period of time

example, in the telecommunication network fault analysis, it is important to know what are the luent itemsets happening before the faults on the network during the recent period of time*The ling of frequent itemsets is also essential to a wide range of emerging applications, such as web log click-stream mining, network traffic analysis, trend analysis, e-business and stock market analysis, ;or networks, etc.

With the rapid emergence of these new application domains, it has become chalking to analyze data over fast-arriving and large data streams in order to capture interesting trends, erns and exceptions.

In a time-varying data stream, the frequency of itemsets may change with time. This will make lid some old frequent itemsets and also introduce new ones.

When changes are detected (e.g., a ~ fraction of old frequent itemsets are found no longer valid during the test against new arriving), the frequent itemsets need to be re-computed to reflect the features of new data and the result ild be available as quickly as possible.

In a data stream application instead of a file of market-baskets, we have a stream of baskets from that we want to mine the frequent itemsets.

A clear distinction between streams and files, when frequent itemsets are considered, is that there is no end to a stream, so eventually an itemset is going to d the support threshold as long as it appears repeatedly in the stream.

As a result, for streams, we think of the support threshold s as a fraction of the baskets in which an itemset must appear in r to be considered frequent. Even with this adjustment, we still have several options regarding the on of the stream over which that fraction is measured.

Sampling Methods for Streams

s assume that stream elements are baskets of items. One simple approach to maintaining a current ate of the frequent itemsets in a stream is to collect some number of baskets and store it as a file, any one of the frequent-itemset algorithms discussed earlier in this chapter, meanwhile ignoring ream elements that keep arriving, or storing them as another file to be analyzed later.

When the ent-itemsets algorithm finishes, we have an estimate of the frequent itemsets in the stream.

rc can use this collection of frequent itemsets for whatever application is at hand, but start running er iteration of the chosen frequent-itemset algorithm immediately. The following two methods e used:

Use the file that was collected while the first iteration of the algorithm was running. At the same time, collect yet another file to be used at another iteration of the algorithm when this current iteration finishes.

Start collecting another file of baskets now, and run the algorithm when an adequate number of baskets has been collected.

We can continue to count the numbers of occurrences of each of these frequent itemsets, along with the total number of baskets seen in the stream, since the counting started. If any itemset is discovered to occur in a fraction of the baskets that is significantly below the threshold fraction s, then this set can be dropped from the collection of frequent itemsets.

When computing the fraction, it is important to include the occurrences from the original file of baskets from which the frequent itemsets were derived. If not, we run the risk that we shall encounter a short period in which a truly frequent itemset does not appear sufficiently frequently and throw it out. We should also allow some way for new frequent itemsets to be added to the current collection.

Periodically gather a new segment of the baskets in the stream and use it as the data file for another iteration of the chosen frequent-itemsets algorithm.

The new collection of frequent items is formed from the result of this iteration and the frequent itemsets from the previous collection that have survived the possibility of having been deleted for becoming infrequent.

Add some random itemsets to the current collection, and count their fraction of occurrences for a while, until one has a good idea of whether or not they are currently frequent.

Rather than choosing the new itemsets completely at random, one might focus on sets with items that appear in many itemsets already known to be frequent; For instance, a good choice can be to pick new itemsets from the negative border of the current set of frequent itemsets.

Frequent Itemsets in Decaying Windows

A decaying window on a stream is formed by picking a small constant c and giving the Ith element prior to the most recent element the weight (1 - c)', or approximately e~ct. We considered, for each item, a stream that had 1 if the item appeared at a certain stream element and 0 if not.

We defined the “score” for that item to be the sum of the weights where its stream element was 1. We were constrained to record all items whose score was at least 1/2.

We cannot use a score threshold above 1, because we do not initiate a count for an item until the item appears in the stream, and the first time it appears, its score is only 1 [since 1, or (1 - c)°, is the weight of the current item].

If we wish to adapt this method to streams of baskets, we consider two modifications. Stream elements are baskets rather than individual items, so many items may appear at a given stream element.

Treat each of those items as if they were the “current” item and add 1 to their score after multiplying all current scores by 1 - c. If some items in a basket have no current score, initialize the scores of those items to 1.

The second modification is trickier. We want to find all frequent itemsets, not just singleton itemsets, If we were to initialize a count for an itemset whenever we saw it, we would have too many counts. For example, one basket of 20 items has over a million subsets, and all of these would have to be initiated for one basket. On the other hand, as we mentioned, if we use a requirement

Above 1 for initiating the; scoring of an itemset, then we would never get any itemsets started, and the method would not work.

A way of dealing with this problem is to start scoring certain itemsets as soon as we see one instance, ut be conservative about which itemsets we start.

We may borrow from the Apriori trick and only start n itemset I if all its immediate proper subsets are already being scored. The consequence of this restriction is that if it is truly frequent, eventually we shall begin to count it, but we never start an itemset unless it would at least be a candidate in the sense used in the Apriori algorithm.

6.Write a short note on clustering techniques and its applications. [CO4-L2]

Clustering Techniques

In this section, we will review the basic concepts of clustering and introduce the different terminologies used. We will give an overview of traditional clustering techniques and discuss the so-called “Curse of Dimensionality” that dictates the design of different set of algorithms for high dimensional data.

Basic Concepts

Clustering datasets consist of objects (measurements, events) that are normally represented as points (vectors) in a multi-dimensional space. Each dimension represents a distinct attribute (variable, measurement) describing the object.

For simplicity, it is usually assumed that values are present for all attributes. The set of objects form an m by n matrix, where there are m rows, one for each object, and n columns, one for each attribute.

Since clustering is the grouping of similar instances/objects, some measures that can determine whether two objects are similar or dissimilar are required. The two main types of measures normally used to estimate this relation are distance measures and similarity measures.

Many clustering methods use distance measures to determine the similarity or dissimilarity between any pair of objects. It is useful to denote the distance between two instances xi and x- as d(xiy Xj).

We can recall that a valid distance measure should be symmetric and attain zero value when the two vectors are identical. The distance measure is called a metric distance measure if it also satisfies the following properties:

Triangle inequality d(xt,xk)< d{xi,Xj) + d(xiyxk)V xi,xjyxk e 5

^(^,^) = 0=^x- -x}V xitXj € S

However, current clustering situations require more sophisticated measures. They may involve Euclidean spaces of very high dimension or spaces that are not Euclidean at all. In such cases, objects are sometimes, represented by more complicated data structures than the vectors of attributes. Good examples include text documents, images, or graphs.

Determining the similarity (or differences) of two objects in such a situation is more complicated, but if a reasonable similarity (dissimilarity) measure exists, then a clustering analysis can still be performed. Such measures include the Jaccard distance, Cosine distance, Hamming distance, and Edit distance.

Clustering Applications

Some common application domains in which the clustering problem arises are as follows:

Intermediate step for other fundamental data mining problems: Since a clustering can be considered a form of data summarization, it often serves as a key intermediate step for many fundamental data mining problems, such as classification or outlier analysis. A compact summary of the data is often useful for different kinds of application-specific insights.

Collaborative filtering:

In collaborative filtering methods of clustering provides a summarization of like-minded users.

The ratings provided by the different users for each other are used in order to perform the collaborative filtering. This can be used to provide recommendations in a variety of applications.

Customer segmentation:

This application is quite similar to collaborative filtering, since it creates groups of similar customers in the data.

The major difference from collaborative filtering is that instead of using rating information, arbitrary attributes about the objects may be used for clustering purposes.

Data summarization:

Many clustering methods are closely related to dimensionality reduction methods. Such methods can be considered as a form of data summarization.

Data summarization can be helpful in creating compact data representations that are easier to process and interpret in a wide variety of applications.

Dynamic trend detection:

Many forms of dynamic and streaming algorithms can be used to perform trend detection in a wide variety of social networking applications.

In such applications, the data is dynamically clustered in a streaming fashion and can be used in order to determine the important patterns of changes.

Examples of such streaming data could be multi-dimensional data, text streams, streaming time-series data, and trajectory data. Key trends and events in the data can be discovered with the use of clustering methods.

Multimedia data analysis:

A variety of different kinds of documents, such as images, audio, or video, fall in the general category of multimedia data.

The determination of similar segments has numerous applications, such as the determination of similar snippets of music or similar photographs.

In many cases, the data may be multi-modal and may contain different types. In such cases, the problem becomes even more challenging...

Biological data analysis:

Biological data has become pervasive in the last few years, because of the success of the human genome effort and the increasing ability to collecting different kinds of gene expression data.

Biological data is usually structured either as sequences or as networks. Clustering algorithms provide good ideas of the key trends in the data, as well as the unusual sequences.

Social network analysis:

In these applications, the structure of a social network is used in order to determine the important communities in the underlying network.

Community detection has important applications in social network analysis, because it provides an important understanding of the community structure in the network.

Clustering also has applications to social network summarization, which is useful in a number of applications.

The above-mentioned list of applications represents a good cross-section of the wide diversity of problems that can be addressed with the clustering algorithms.

7.Explain what strategies are commonly used to perform clustering. [CO4-L2]

Clustering Strategies

Distinction in clustering approaches that is between hierarchical and partitioned approaches is highlighted below:

1.Hierarchical techniques produce a nested arrangement of partitions, with a single cluster at the top consisting of all data points and singleton clusters of individual points at the bottom. Each intermediate level can be viewed as combining (splitting) two clusters from the next lower (next higher) level. Hierarchical clustering techniques which start with one cluster of all the points and then keep progressively splitting the clusters till singleton clusters are reached are called “divisive” clustering. On the other hand, approaches that start with singleton clusters and go on merging close clusters at every step until they reach one cluster consisting of the entire dataset are called “agglomerative” methods. While most hierarchical algorithms just join two clusters or split a cluster into two sub-clusters at every step, there exist hierarchical algorithms that can join more than two clusters in one step or split a cluster into more than two sub-clusters.

2.Partitional techniques create one-level (un-nested) partitioning of the data points. If K is the desired number of clusters, then partitional approaches typically find all K clusters in one step. The important issue is we need to have predefined value of K, the number of clusters we propose to identify in the dataset.

Of course, a hierarchical approach can be used to generate a flat partition of K clusters, and likewise, the repeated application of a partitional scheme can provide a hierarchical clustering.

There are also other important distinctions between clustering algorithms as discussed below:

1.Does a clustering algorithm use all attributes simultaneously (polythetic) or use only one attribute at a time (monothetic) to compute the distance?
2.Does a clustering technique use one object at a time (incremental) or does the algorithm consider all objects at once (non-incremental)?
3.Does the clustering method allow a point to belong to multiple clusters (overlapping) or does it insist that each object can belong to one cluster only (non-overlapping)? Overlapping clusters are not to be confused with fuzzy clusters, as in fuzzy clusters objects actually belong to multiple classes with varying levels of membership.

Algorithms for clustering big data can also be distinguished by

1.Whether the dataset is treated as a Euclidean space, and if the algorithm can work for any arbitrary distance measure. In a Euclidean space where data is represented as a vector of real numbers, the notion of a Centroid which can be used to summarize a collection of data points is very natural - the mean value of the points. In a non-Euclidean space, for example, images or docu- . .ments where data is a set of words or a group of pixels, there is no notion of a Centroid, and we are forced to develop another way to summarize clusters.
2.Whether the algorithm is based on the assumption that data will fit in main memory, or whether data must reside in secondary memory, primarily. Algorithms for large amounts of data often must take shortcuts, since it is infeasible to look at all pairs of points. It is also necessary to summarize the clusters in main memory itself as is common with most big data algorithms.

8.Explain how Hierarchical clustering is performed in Euclidean Space and Non-Euclidean Space. [CO4-L2]

Hierarchical Clustering

Hierarchical clustering builds a cluster hierarchy or, in other words, a tree of clusters, also known as a dendrogram. Every cluster node contains child clusters; sibling clusters partition the points covered by their common parent. Such an approach allows exploring data on different levels of granularity. Hierarchical clustering methods are categorized into agglomerative (bottom-up) and divisive (top-down).

The advantages of hierarchical clustering include: embedded flexibility regarding the level of granularity; ease of handling of any forms of similarity or distance; can be extended to be applicable with any attribute-i-ypes. But these algorithms also suffer from the followingdiaw&acks: ambiguity in termination criteria; the fact that most hierarchical algorithms cannot disturb an earlier intermediate cluster even for improvement of cluster quality..

Hierarchical clustering can also be differentiated based on whether we are dealing with Euclidean or non-Euclidean space.

Hierarchical Clustering in Euclidean Space

These methods construct the clusters by recursively partitioning the instances in either a top-down or bottom-up fashion. These methods can be subdivided as following:

1.Agglomerative hierarchical clustering: Each object initially represents a cluster of its own. Then clusters are successively merged until the desired clustering is obtained.
2.Divisive hierarchical clustering: All objects initially belong to one cluster. Then the cluster is divided into sub-clusters, which are successively divided into their own sub-clusters. This process continues until the desired cluster structure is obtained.

The result of the hierarchical methods is a dendrogram, representing the nested grouping of objects and similarity levels at which groupings change. A clustering of the data objects is obtained by cutting the dendrogram at the desired similarity level. Figure 9 showrs a simple example of hierarchical clustering. The merging or division of clusters is to be performed according to some similarity measure, chosen so as to optimize some error criterion (such as a sum of squares).

The hierarchical clustering methods can be further divided according to the manner in which inter- cluster distances for merging are calculated.

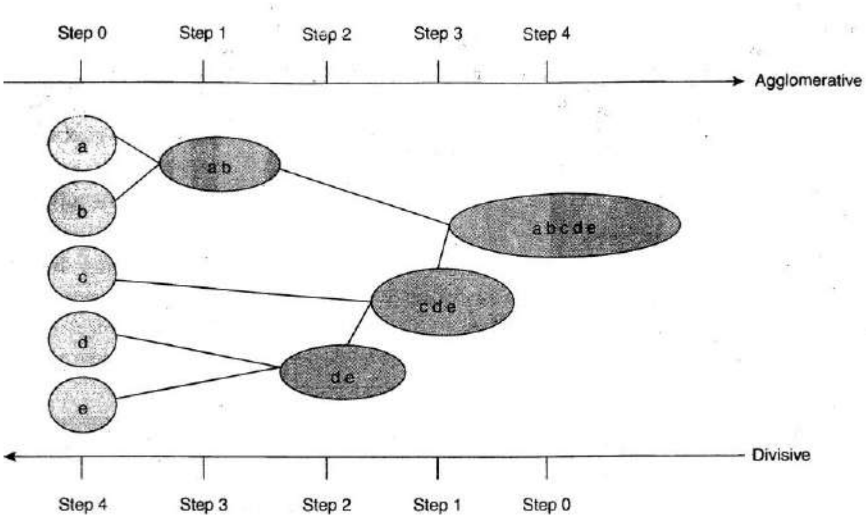


Figure 9 Hierarchical clustering,

1.Single-link clustering; Here the distance between the two clusters is taken as the shortest distance from any member of one duster to any member of the other cluster. If the data consist of similarities, the similarity between a pair of clusters is considered to be equal to the greatest similarity from any member of one cluster to any member of the other cluster.
2.Complete-link clustering: Here the distance between the two clusters is the longest distance from any member of one cluster to any member of the other cluster.
3.Average-link clustering: Here the distance between two clusters is the average of all distances computed between every pair of two points one from each cluster.
4.Centroid link clustering: Here the distance between the clusters is computed as the distance between the two mean data points (average point) of the clusters. This average point of a cluster is called its Centroid. At each step of the clustering process we combine the two clusters that have the smallest Centroid distance. The notion of a Centroid is relevant for Euclidean space only, since all the data points have attributes with real values. Figure 10 shows this process.

Another decision point for hierarchical clustering would be the stopping criteria. Some choices are: stop merging (or splitting) the clusters when

1.Some pre-determined number of clusters is reached. We know beforehand that we need to find a fixed number of clusters “K” and we stop when we have K clusters.
2.If across a series of merges or splits, very little change occurs to the clustering, it means clustering has reached some stable structure.
3.If the maximum distance between any two points in a cluster becomes greater than a pre-specified value or threshold we can stop further steps.

4.Combination of the above conditions.

The main disadvantages of the hierarchical methods are as follows:

1.Inability to Scale well - The time complexity of hierarchical algorithms is at least 0(m2) (where m is the total number of instances), which is non-linear with the number of objects. Clustering a large number of objects using a hierarchical algorithm is also characterized by huge I/O costs.
2.Hierarchical methods can never undo what was done previously. Namely there is no backtracking capability.

Hierarchical Clustering in Non-Euclidean Space

When the space is non-Euclidean, as is the case when data points are documents or images etc., we need to use some distance measure, such as Jaccard, cosine, or edit distance. We have seen in Chapter 5 how to compute such distances between two such data points. We need some mechanism for finding inter-cluster distance using Centroids. A problem arises when we need to represent a cluster, because we cannot replace a collection of points by their Centroid.

This representative point is called the “Clustroid” of the cluster. We can select the Clustroid in various ways. Common choices include selecting as the Clustroid the point that minimizes:

1.The sum of the distances to the other points in the cluster.
2.The maximum distance to another point in the cluster.
3.The sum of the squares of the distances to the other points in the cluster.

9.Write a short not on K-Means Algorithm. [CO4-L2]

K-Means Algorithm

The $\boldsymbol{k}$-means algorithm discovers K (non-overlapping) clusters by finding centroids (“central” points) and then assigning each point to the cluster associated with its nearest centroid. A cluster centroid is typically the mean or median of the points in its cluster and “nearness” is defined by a distance or similarity function. In the ideal case, the Centroids are chosen to minimize the total “error”, where the error for each point is given by a function that measures the dispersion of a point from its cluster Centroid, for example, the squared distance.

The algorithm starts with an initial set of cluster Centroids chosen at random or according to some heuristic procedure. In each iteration, each data point is assigned to its nearest cluster Centroid. Nearness is measured using the Euclidean distance measure. Then the cluster Centroids are re-computed. The Centroid of each duster is calculated as the mean value of all the data points hat are assigned to that cluster.

Several termination conditions are possible. For example, the search may stop when the error that is computed at every iteration does not reduce because of reassignment of the Centroids. This indicates that the present partition is locally optimal. Other stopping criteria can be used also such as stopping the algorithms after a pre-defined number of iterations.

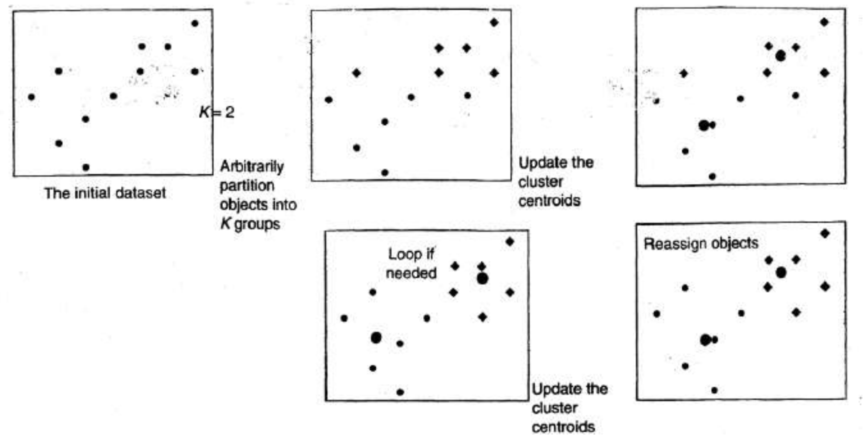

K = 2
The initial dataset
Arbitrarily partition objects into K groups
Update the cluster centroids
Reassign objects
Loop if needed
Update the cluster centroids

Figure 11 Illustrating A-means.

The $\boldsymbol{K}$-means algorithm is a gradient-decent procedure which begins with an initial set of if cluster- centers and iteratively updates it so as to decrease the error function. The complexity of T iterations of the A-means algorithm performed on a sample size of m instances, each characterized by $\boldsymbol{N}$ attributes, is $\boldsymbol{O}(\boldsymbol{T} * \boldsymbol{K} * \boldsymbol{m} * \boldsymbol{N})$. This linear complexity is one of the reasons for die popularity of the K-means algorithms. Even if the number of data points is sufficiently large, this algorithm is computationally attractive. Other reasons for the algorithms popularity are its ease of interpretation, simplicity of implementation, speed of convergence and adaptability to sparse data.

The major disadvantage of the A-means algorithm involves the selection of the initial partition or the initial Centroids. The algorithm is very sensitive to this selection, which may make the difference between the algorithm converging at global or local minimum.

Further, the K-means algorithm works well only on datasets having circular clusters. In addition, this algorithm is sensitive to noisy data and outliers; a single outlier can increase the squared error dramatically. It is applicable only when mean is defined (namely, for numeric attributes) and it requires the number of clusters in advance, which is not trivial when no prior knowledge is available.

Since the K-means algorithm requires Euclidean space that is often limited to the numeric attributes to find the Centroid, the K-prototypes algorithm was proposed. This algorithm is similar to the K-means algorithm but can process attributes that are non-numeric too. Instead of using the Centroid it uses a prototype which captures the essence of the cluster to which it belongs. The algorithm then clusters objects with numeric and categorical attributes in a way similar to the K-means algorithm. The similarity measure on numeric attributes is the square Euclidean distance; the similarity measure on the categorical attributes is the number of mismatches between objects and the cluster prototypes.

One such algorithm is the $\boldsymbol{k}$-medoids or partition around medoids (PAM). Each cluster is represented by the central object in the cluster, rather than by the mean that may not even belong to the cluster. The $\boldsymbol{k}$-medoids method is more robust than the $\boldsymbol{K}$-means algorithm in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean. However, its processing is more costly than the $\boldsymbol{K}$-means method. Both methods require the user to specify $\boldsymbol{k}$, the number of clusters.

10.Explain how K-Means Algorithm can be adapted to handle Large-Scale Data[CO4-L2]

K-Means For Large-Scale Data

We shall now examine an extension of $\boldsymbol{K}$-means that is designed to deal with datasets so large that they cannot fit in main memory. This algorithm does not output the clustering per se but just determines the cluster Centroids. If we require the final clustering of points, another pass is made through the dataset, assigning each point to its nearest Centroid which is then returned as the result.

This algorithm, called the BFR algorithm, for its authors (Bradley, Fayyad and Reina), assumes the data points are from an $\boldsymbol{n}$-dimensional Euclidean space. Therefore, we can use the Centroids to represent clusters, as they are being formed. The BFR algorithm also assumes that the quality of a cluster can be measured by the variance of the points within a cluster; the variance of a cluster is the average of the square of the distance of a point in the cluster from the Centroid of the cluster.

This algorithm for reasons of optimality does not record the Centroid and variance, but rather stores the following (2 n + t) summary statistics for each cluster:

1.N, the number of points in the cluster.
2.For each dimension $\boldsymbol{i}$, the sum of the ith coordinates of the points in the cluster, denoted SUM;.
3.For each dimension $\boldsymbol{i}$, the sum of the squares of the ***i***th coordinates of the points in the cluster, denoted as ***SUMSQ***$_i$.

The reasoning behind using these parameters comes from the fact that these parameters are easy to compute when we merge two clusters. We need to just add the corresponding values from the two clusters. Similarly we can compute the Centroid and variance also very easily from these values as:

1.The ***i***th coordinate of the Centroid is $\frac{SUM_i}{N}$
2.The variance in the ***i***th dimension is $\frac{SUMSQ_i}{N} - \left(\frac{SUM_i}{N}\right)^2$.
3.The standard deviation in the ***i***th dimension is the square root of the variance in that dimension.

Initially, the BFR algorithm selects k points, either randomly or using some preprocessing methods to make better choices. In the next step the data file containing the points of the dataset are in chunks. These chunks could be from data stored in a distributed file system or there may be one monolithic huge file which is then divided into chunks of the appropriate size. Each chunk consists of just so many points as can be processed in the main memory. Further some amount of main memory is also required to store the summaries of the k dusters and other data, so the entire 'memory is not available to store a chunk.

The data stored in the main-memory other than the chunk from the input consists of three types of objects:

1.The discard set: The points already assigned to a cluster. These points do not appear in main memory. They are represented only by the summary statistics for their cluster.
2.The compressed set: There are several groups of points that are sufficiently close to each other for us to believe they belong in the same cluster, but at present they are not close to any current Centroid. In this case we cannot assign a cluster to these points as

we cannot ascertain to which cluster they belong. Each such group is represented by its summary statistics, just like the clusters are, and the points themselves do not appear in main memory.

3.The retained set: These points are not close to any other points; they are “outliers.” They will eventually be assigned to the nearest cluster, but for the moment we have to retain each such point in main memory

These sets will change as we bring in successive chunks of data into the main memory. Figure 12 indicates the state of the data after a few chunks of data have been processed by the BFR algorithm.

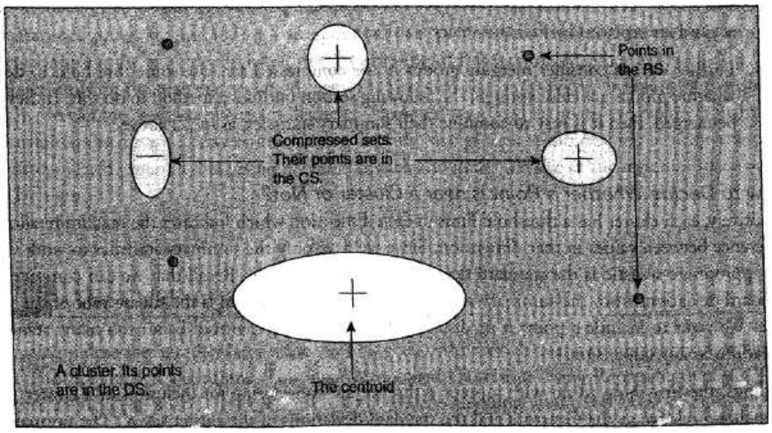

Points in the RS
Compressed sets. Their points are in the CS.
A cluster. Its points are in the DS.
The centroid

Figure 12 The three sets of points identified by BFR.

Processing a Memory Load of Points

We shall now describe how one chunk of data points are processed. We assume that the main memory currently contains the summary statistics for the K clusters and also for zero or, more groups of points that are in the compressed set. The main memory also holds the current set of points in the retained set. We then perform the following steps:

1.For all points $(x_1,x_2,\ldots,x_n)$ that are “sufficiently close” (based on distance threshold) to the Centroid of a cluster, add the point to that cluster. The point then is added to the discard set. We add 1 to the value N in the summary statistics for that cluster indicating that this cluster has grown by one point. We also add X_j to SUM_i and add X_j^2 to $SUMSQ_i$, for that cluster.

2.If this is the last chunk of data, merge each group from the compressed set and each point of the retained set into its nearest cluster. We have seen earlier that it is very simple and easy to merge dusters and groups using their summary statistics. Just add the counts N, and add corresponding components of the SUM and SUM5Q vectors. The algorithm ends at this point.

3.Otherwise (this was not the last chunk), use any main-memory clustering algorithm to cluster the remaining points from this chunk, along with all points in the current retained set. Set a threshold on the distance values that can occur in the cluster, so we do not merge points unless they are reasonably close.

4.Those points that remain isolated as clusters of size 1 (i.e., they are not near any other point) become the new retained set. Clusters of more than one point become groups in the compressed set and are replaced by their summary statistics.

5.Further we can consider merging groups in the compressed set. Use some threshold to decide whether groups are close enough; the following section outlines a method to do this. If they can be merged, then it is easy to combine their summary statistics, as in (2) above.

10.How to Decide Whether a Point is near a Cluster or Not? [CO4-L2]

Intuitively, each cluster has a threshold limit in each dimension which indicates the maximum allowed difference between values in that dimension. Since we have only the summary statistics to work with, the appropriate statistic is the standard deviation in that dimension. Recall that we can compute the standard deviations from the summary statistics; the standard deviation is the square root of the variance. We want to include a point if its distance from the cluster Centroid is not too many standard deviations in any dimension.

Thus, the first thing to do with a point p that we are considering for inclusion in a cluster is to normalize p relative to the Centroid and the standard deviations of the cluster say ***p'***. The normalized distance of p from the centroid is the absolute distance of ***p'*** from the origin. This distancd is sometimes called the Mahalanobis distance.

We also need to decide whether to merge groups in the compressed set. We can make a decision using only the summary statistics for the two groups. Choose an upper bound on the sum of the variances of the combined group in each dimension. Recall that we compute the summary statistics for the combined group by just adding corresponding components, and compute the variance in each dimension using the formula discussed in the beginning of this section. Another way would be, put an upper limit on the diameter in any dimension. Since we do not know the locations of the points exactly, we cannot compute the exact diameter. However, we could estimate the diameter in the i^{th} dimension as the distance between the Centroids of the two groups in dimension i plus the standard deviation of each group in dimension i. This approach also limits the size of the region of space occupied by a group.

11.Explain briefly about CURE Algorithm and its implimentation[CO4-L2]

The CURE Algorithm

This section describes another large-scale-clustering algorithm which uses a combination of partition- based and hierarchical algorithms. This algorithm, called Clustering Using Representatives (CURE), assumes a Euclidean space. However, it does not assume anything about the shape of clusters; they need not be normally distributed, and can even have strange curves, S-shapes, or even rings. Instead of representing the clusters by their Centroid, it uses a collection of representative points, as is implied by the name of the algorithm.

Overview of the Algorithm

One way to manage clustering in large datasets would be to use one single data point to represent a cluster. Conceptually, we implicitly assume that each cluster has a spherical shape and thr ? its radius point is representative of the entire cluster. This is not true for many real-life applications where the clusters can exist in many other complicated shapes. At the other extreme, we can keep all the data points within each cluster, which is expensive both computation wise well as space wise, especially for large datasets. To address this issue, CURE proposes to use a set of well-scattered data points to represent a cluster.

CURE is essentially a hierarchical clustering algorithm. It starts by treating each data point as a single cluster and then recursively merges two existing clusters into one until we have only k clusters. In order to decide which two clusters to be merged at each iteration, it computes the minimum distance between all the possible pairs of the representative points from the two clusters. CURE uses two major data structures to perform this operation efficiently. First, it uses a heap to track the distance of each existing cluster to its closest cluster. Additionally, it also uses a k-d tree to store all the representative points for each cluster.

In order to speed up the computation, CURE first draws a sample of the input dataset and then runs the above procedure on the sampled data. When the original dataset is large, the different clusters might overlap each other, which in turn will require a large sample size. To overcome this problem, CURE uses partitions to speed up. It first partitions the n sampled data points partitions.

Within each partition, it then runs a partial hierarchical clustering algorithm until either a pre-defined number of clusters is reached or the distance between the two clusters to be merged exceeds some threshold. After that, it runs another clustering pass on all the partial clusters from all the p partitions ("global clustering"). Finally, each non-sampled data point is assigned to a cluster based on its distance to the representative point ("labeling").

CURE Implementation

Initialization Step

Start with an initial sample of the data and cluster it using any traditional main memory clustering technique. Although any clustering method can be used, since CURE is designed more to handle oddly shaped clusters, hierarchical methods are the better choices. Agglomerative hierarchical methods work the best with CURE.

Select a small set of points from each cluster to be -representative points. Choose these representatives so that they are wide apart, that is, as far from each other as possible.

Move each of the representative points by some fixed fraction of the distance between its location and the Centroid of its cluster. In effect we are shrinking these representative points towards the Centroid of the cluster by a fraction. This fraction could be about 20% or 30% of the original distance. (Note that this step requires a Euclidean space, since otherwise, we cannot define a line joining two points). The distance between any two clusters is now computed as the distance between the closest pair of representative points - one belonging to each of the two clusters. Thus, only the representative points of a cluster are used to compute its distance from other clusters.

The representative points attempt to capture the physical shape and geometry of the cluster. Furthermore, shrinking the scattered points toward the mean by a factor gets rid of any anomalies caused by outliers or random noise points. The reason for this is that outliers will normally be for off from the cluster center, and thus shrinking would cause the outliers to come closer to die center while the remaining representative points would undergo very small shifts. The larger movements in the outliers would thus reduce their ability to cause the wrong clusters to be merged. The shrinking parameter can also be used to control the shapes of clusters. A smaller value shrinks the scattered points very little and, thus, favours the elongated clusters. On the other hand, with larger values the scattered points get located closer to the mean, resulting in compact clusters.

Completion Phase

Once the initialization step is complete, we now have to cluster the remaining points and output the final clusters:

1.The final step of CURE is to merge two clusters if they have a pair of representative points, one from each cluster, that are sufficiently close. The user may pick the distance threshold at which points may be considered “close”. This merging step can be repeated until there are no more sufficiently close clusters.
2.Each point p is brought horn secondary storage and compared with the representative points. We assign p to the cluster of the representative point that is closest to p.

12.Illustrate how clustering could be performed on Streams[CO4-L2]

Clustering Streams

Most of the classical methods in clustering cannot be simply extended to be used with very large datasets or data streams. The stream scenario brings a unique set of challenges with it, as follows:

1.Data Streams are normally massive in size and thus they cannot be stored explicitly even on disk. Therefore, the data needs to be processed in a single pass, in which all the summary information required for the clustering process must be collected and stored. The time needed to process each record must be minimum and constant. Otherwise, the summarization process will not be in synchronization with the data arriving in the stream.
2.The patterns in the data stream may continuously evolve and change over time. For the purpose of stream clustering this means that we cannot have a static clustering model but we need a model capable of evolving with changes in the stream data. Further at any instant of time a current clustering must be generated so that it can be used y an analytics application. This is required also because data streams may never end.
3.Based on the domain of origin of the data stream, clustering streams face different challenges. For .example, if the data values consist of very high Jimensiona! data made up of discrete attributes, it may become impossible to store summary representations of the clusters efficiently. Therefore, space-efficient methods are needed when data for clustering such data streams.

The problem of data stream clustering can occur in two flavors: Firstly, we want to be able to access large streaming sources like network traffic, satellite imagery, streaming videos, etc., which, because of their sheer volume and very high velocity of arrival, cannot be stored for offline processing. Thus clustering can aggregate and summarize data in real time from these streams and then discard the actual data. Secondly, stream clustering methods can also be used to access very fast changing databases like web databases; Googles MapReduce computational framework is one example which uses stream clustering techniques.

In this section, we provide a very brief overview of some stream clustering algorithms. The reader is referred to the references for further information on Stream Clustering.

A Simple Streaming Model

A simple Stream Clustering model is based on the k-medians (k-means) clustering methodology. The central scheme is to divide stream into chunks, which can be managed and stored in main memory. We assume that the original data stream D is divided into chunks D_1 ..., D_r,..., each of which contains at most m data points. The value of m is dictated by the size of the main memory. Since each chunk fits in main memory, we can use complex clustering algorithms on each chunk of data.

Each stream element is represented as one point in some dimensional space. A sliding window consists of the most recent N points. Stream clustering attempts to group subsets of the points in the stream, so that we can quickly answer queries which need the clustering structure of the last m points, for any m < N. Further given a value A" we may need the last m points to be grouped into K clusters.

Further there is no restriction regarding the space in which the points of the stream may be represented. It may be a Euclidean space, in which case clustering will work with Centroids (means) of the selected clusters. If the space is non-Euclidean, clustering algorithms will process Clustroids (medians) of the selected clusters.

Further, for ease of designing efficient streaming algorithms it is assumed that the stream elements are fairly stable and their summary statistics that do not vary very much as the stream proceeds. This implies that it is sufficient to consider only a sample of the stream to estimate the clusters, and we can afford to ignore the stream after a while. However, this assumption may not be true in general, which may require further complex algorithms to cluster. For example, the Centroids of the clusters may migrate slowly as time goes on, or clusters may expand, contract, divide or merge.

13.Explain shortly how Clustering algorithms could be modified to handle Streams. [CO4-L2]

A Stream Clustering Algorithm

We discuss a simplified version of an algorithm referred to as BDMO (for the authors, B. Babcock, M. Datar, R. Motwani and L. O'Callaghan). The true version of the algorithm provides better performance guarantees in the worst case but is quite complex and out of the scope of this book.

The BDMO algorithm builds on the methodology for counting ones in a stream that was described in Chapter 6. Some of the key similarities and differences are as follows:

1.Similar to the earlier algorithm, the points of the stream are partitioned into, and summarized by, buckets whose sizes are a power of two. Here, the size of a bucket is the number of points it represents, rather than the number of stream elements that are 1.
2.As before, the sizes of buckets follow the rule that there may be only one or two buckets of each size, up to some limit. But unlike the previous algorithm, the sequence of possible bucket sizes need not start at 1. They are required only to form a sequence where each size is twice the previous size, for example, 3, 6, 12, 24,....
3.Like before Bucket sizes have to be non-decreasing as we go back in time. As was discussed in Chapter 6 we can conclude that there will be 0(log N) buckets.

The contents of a bucket are as follows:

1.The size of the bucket.
2.The timestamp of the bucket, this is, the most recent point that contributes to the bucket. As before we can record the timestamps with modulo N.
3.A collection of records that represent the clustering of the points that are in this bucket. These records contain: the number of points in the cluster; the Centroid or Clustroid of the cluster; other necessary parameters useful for merging clusters and maintaining approximations to the fall set of parameters for the merged cluster.

Initializing Buckets

The smallest bucket size will bep, a power of 2. Thus, for everyp stream elements, we have to construct a new bucket, with the most recent p points. The timestamp for this bucket is the timestamp of the most recent point in the bucket, We can leave each point to form a cluster by itself, or we may apply some clustering strategy on the entire bucket of points. For instance, if we choose a AT-means algorithm, then (assuming K<p) we cluster the points into Kclusters.

After performing the initial clustering we compute the Centroids or Clustroids for the clusters and count the points in each cluster. This information is added to the record for each cluster. We also compute the other needed parameters for merging clusters ahead.

Merging Buckets

Whenever we create a new bucket, we need to review the sequence of buckets.

1.First, if some bucket has a timestamp that is more than Artime units prior to the current time, then nothing of that bucket is in the window, and we may drop it from the list.
2.Second, we may have created three buckets of size p, in which case we must merge the oldest two of the three. The merger may create two buckets of size 2p, in which case we may have to merge buckets of increasing sizes, recursively. (Similar to out earlier algorithm.)

To merge two consecutive buckets:

1.Record the size of the bucket as twice the size of the two buckets being merged.
2.The timestamp for the merged bucket is the timestamp of the more recent of the two consecutive buckets.
3.If we decide to merge the clusters we need to compute the parameters of the newly formed cluster.

Answering Queries

One main goal of stream clustering is to be able to answer any queries posed on a current window of data. One common query is a request for the clustering structure of the most recent m points in the stream, where m < N. The above discussed strategy involves combining of buckets as we go back in time, and so we may not always be able to find a set of buckets that covers exactly the last m points. However, if we choose the smallest set of buckets that cover the last m points, we can easily see that the construction of these buckets restricts the number of points to no more than the last 2m points. This query will provide as answer, the Centroids or Clustroids of all the points in the selected buckets.

Unit – V

Introduction

Part – A

1.Explain about the process of inter cluster data copying. [CO5-L2]

HDFS provides a distributed data copying facility through the DistCP from source to destination. If this data copying is within the hadoop cluster then it is referred to as inter cluster data copying. DistCP requires both source and destination to have a compatible or same version of hadoop.

2. How can you overwrite the replication factors in HDFS? CO5-L1]

The replication factor in HDFS can be modified or overwritten in 2 ways-

1)Using the Hadoop FS Shell, replication factor can be changed per file basis using the below command-

$hadoop fs –setrep –w 2 /my/test_file (test_file is the filename whose replication factor will be set to 2)

2)Using the Hadoop FS Shell, replication factor of all files under a given directory can be modified using the below command-

3)$hadoop fs –setrep –w 5 /my/test_dir (test_dir is the name of the directory and all the files in this directory will have a replication factor set to 5)

3. Explain the difference between NAS and HDFS. [CO5-L2]

NAS runs on a single machine and thus there is no probability of data redundancy whereas HDFS runs on a cluster of different machines thus there is data redundancy because of the replication protocol.

NAS stores data on a dedicated hardware whereas in HDFS all the data blocks are distributed across local drives of the machines.

In NAS data is stored independent of the computation and hence Hadoop MapReduce cannot be used for processing whereas HDFS works with Hadoop MapReduce as the computations in HDFS are moved to data.

4. Explain what happens if during the PUT operation, HDFS block is assigned a replication factor 1 instead of the default value 3. [CO5-L2]

Replication factor is a property of HDFS that can be set accordingly for the entire cluster to adjust the number of times the blocks are to be replicated to ensure high data availability. For every block that is stored in HDFS, the cluster will have n-1 duplicated blocks. So, if the replication factor during the PUT operation is set to 1 instead of the default value 3, then it will have a single copy of data. Under these circumstances when the replication factor is set to 1 ,if the DataNode crashes under any circumstances, then only single copy of the data would be lost.

4A. What is the process to change the files at arbitrary locations in HDFS? _ [CO5-L1]

HDFS does not support modifications at arbitrary offsets in the file or multiple writers but files are written by a single writer in append only format i.e. writes to a file in HDFS are always made at the end of the file.

5. Explain about the indexing process in HDFS. _ [CO5-L2]

Indexing process in HDFS depends on the block size. HDFS stores the last part of the data that further points to the address where the next part of data chunk is stored.

6. Explain about the SMB Join in Hive. [CO5-L2]

In SMB join in Hive, each mapper reads a bucket from the first table and the corresponding bucket from the second table and then a merge sort join is performed. Sort Merge Bucket (SMB) join in hive is mainly used as there is no limit on file or partition or table join. SMB join can best be used when the tables are large. In SMB join the columns are bucketed and sorted using the join columns. All tables should have the same number of buckets in SMB join.

7 . How can you connect an application, if you run Hive as a server? CO5-L1]

When running Hive as a server, the application can be connected in one of the 3 ways-
ODBC Driver-This supports the ODBC protocol
JDBC Driver- This supports the JDBC protocol
Thrift Client- This client can be used to make calls to all hive commands using different programming language like PHP, Python, Java, C++ and Ruby.

8. What does the overwrite keyword denote in Hive load statement? CO5-L1]

Overwrite keyword in Hive load statement deletes the contents of the target table and replaces them with the files referred by the file path i.e. the files that are referred by the file path will be added to the table when using the overwrite keyword.

9. What is SerDe in Hive? How can you write your own custom SerDe? CO5-L1]

SerDe is a Serializer DeSerializer. Hive uses SerDe to read and write data from tables. Generally, users prefer to write a Deserializer instead of a SerDe as they want to read their own data format rather than writing to it. If the SerDe supports DDL i.e. basically SerDe with parameterized columns and different column types, the users can implement a Protocol based DynamicSerDe rather than writing the SerDe from scratch.

10 . What are the stable versions of Hadoop? CO5-L1]

Release 2.7.1 (stable)
Release 2.4.1
Release 1.2.1 (stable)

11. What is Apache Hadoop YARN? CO5-L1]
YARN is a powerful and efficient feature rolled out as a part of Hadoop 2.0.YARN is a large scale distributed system for running big data applications.

12. Is YARN a replacement of Hadoop MapReduce? CO5-L1]
YARN is not a replacement of Hadoop but it is a more powerful and efficient technology that supports MapReduce and is also referred to as Hadoop 2.0 or MapReduce 2
.

13. What are the additional benefits YARN brings in to Hadoop? CO5-L1]
Effective utilization of the resources as multiple applications can be run in YARN all sharing a common resource.In Hadoop MapReduce there are seperate slots for Map and Reduce tasks whereas in YARN there is no fixed slot. The same container can be used for Map and Reduce tasks leading to better utilization.
YARN is backward compatible so all the existing MapReduce jobs.
Using YARN, one can even run applications that are not based on the MaReduce model

14. How can native libraries be included in YARN jobs? CO5-L1]
There are two ways to include native libraries in YARN jobs-
1) By setting the -Djava.library.path on the command line but in this case there are chances that the native libraries might not be loaded correctly and there is possibility of errors.
2) The better option to include native libraries is to the set the LD_LIBRARY_PATH in the .bashrc file.

15. Explain the differences between Hadoop 1.x and Hadoop 2.x CO5-L2]
In Hadoop 1.x, MapReduce is responsible for both processing and cluster management whereas in Hadoop 2.x processing is taken care of by other processing models and YARN is responsible for cluster management.
Hadoop 2.x scales better when compared to Hadoop 1.x with close to 10000 nodes per cluster.
Hadoop 1.x has single point of failure problem and whenever the NameNode fails it has to be recovered manually. However, in case of Hadoop 2.x StandBy NameNode overcomes the SPOF problem and whenever the NameNode fails it is configured for automatic recovery.
Hadoop 1.x works on the concept of slots whereas Hadoop 2.x works on the concept of containers and can also run generic tasks.

16. What are the core changes in Hadoop 2.0? CO5-L1]
Hadoop 2.x provides an upgrade to Hadoop 1.x in terms of resource management, scheduling and the manner in which execution occurs. In Hadoop 2.x the cluster resource management capabilities work in isolation from the MapReduce specific programming logic. This helps Hadoop to share resources dynamically between multiple parallel processing frameworks like Impala and the core MapReduce component. Hadoop 2.x Hadoop 2.x allows workable and fine grained resource configuration leading

to efficient and better cluster utilization so that the application can scale to process larger number of jobs.

17. Differentiate between NFS, Hadoop NameNode and JournalNode. [CO5-L1]

HDFS is a write once file system so a user cannot update the files once they exist either they can read or write to it. However, under certain scenarios in the enterprise environment like file uploading, file downloading, file browsing or data streaming –it is not possible to achieve all this using the standard HDFS. This is where a distributed file system protocol Network File System (NFS) is used. NFS allows access to files on remote machines just similar to how local file system is accessed by applications.

Namenode is the heart of the HDFS file system that maintains the metadata and tracks where the file data is kept across the Hadoop cluster.

StandBy Nodes and Active Nodes communicate with a group of light weight nodes to keep their state synchronized. These are known as Journal Nodes.

18. What are the modules that constitute the Apache Hadoop 2.0 framework? CO5-L1]

Hadoop 2.0 contains four important modules of which 3 are inherited from Hadoop 1.0 and a new module YARN is added to it.

Hadoop Common – This module consists of all the basic utilities and libraries that required by other modules.

HDFS- Hadoop Distributed file system that stores huge volumes of data on commodity machines across the cluster.

MapReduce- Java based programming model for data processing.

YARN- This is a new module introduced in Hadoop 2.0 for cluster resource management and job scheduling.

19. How is the distance between two nodes defined in Hadoop? CO5-L1]

Measuring bandwidth is difficult in Hadoop so network is denoted as a tree in Hadoop. The distance between two nodes in the tree plays a vital role in forming a Hadoop cluster and is defined by the network topology and java interface DNStoSwitchMapping. The distance is equal to the sum of the distance to the closest common ancestor of both the nodes. The method getDistance(Node node1, Node node2) is used to calculate the distance between two nodes with the assumption that the distance from a node to its parent node is always 1.

20. What is the size of the biggest hadoop cluster a company X operates? CO5-L1]

Asking helps a hadoop job seeker understand the hadoop maturity curve at a company.Based on the answer of the interviewer, a candidate can judge how much an organization invests in Hadoop and their enthusiasm to buy big data products from various vendors. The candidate can also get an idea on the hiring needs of the company based on their hadoop infrastructure.

21. For what kind of big data problems, did the organization choose to use Hadoop? CO5-L3]

Asking to the interviewer shows the candidates keen interest in understanding the reason for hadoop implementation from a business perspective. gives the impression to the interviewer that the candidate is not merely interested in the hadoop developer job role but is also interested in the growth of the company.

22. Based on the answer to question no 1, the candidate can ask the interviewer why the hadoop infrastructure is configured in that particular way, why the company chose to use the selected big data tools and how workloads are constructed in the hadoop environment. [CO5-L3]

Asking to the interviewer gives the impression that you are not just interested in maintaining the big data system and developing products around it but are also seriously thoughtful on how the infrastructure can be improved to help business growth and make cost savings.

23. What kind of data the organization works with or what are the HDFS file formats the company uses? [CO5-L1]

The question gives the candidate an idea on the kind of big data he or she will be handling if selected for the hadoop developer job role. Based on the data, it gives an idea on the kind of analysis they will be required to perform on the data.

24. What is the most complex problem the company is trying to solve using Apache Hadoop? [CO5-L1]

Asking helps the candidate know more about the upcoming projects he or she might have to work and what are the challenges around it. Knowing this beforehand helps the interviewee prepare on his or her areas of weakness.

25. Will I get an opportunity to attend big data conferences? Or will the organization incur any costs involved in taking advanced hadoop or big data certification? [CO5-L1]

This is a very important question that you should be asking these the interviewer. This helps a candidate understand whether the prospective hiring manager is interested and supportive when it comes to professional development of the employee.

Stay Tuned to the blog for more updates on Hadoop Interview FAQ's!!!

We hope that these Hadoop Interview Questions and Answers have pre-charged you for your next Hadoop Interview.Get the Ball Rolling and share your hadoop interview experiences in the comments below.Please do! It's all part of our shared mission to ease Hadoop Interviews for all prospective Hadoopers.We invite you to get involved.

26. What are the main components of a Hadoop Application? [CO5-L1]

Hadoop applications have wide range of technologies that provide great advantage in solving complex business problems.

Core components of a Hadoop application are-
1) Hadoop Common
2) HDFS
3) Hadoop MapReduce
4) YARN
Data Access Components are - Pig and Hive
Data Storage Component is - HBase
Data Integration Components are - Apache Flume, Sqoop, Chukwa
Data Management and Monitoring Components are - Ambari, Oozie and Zookeeper.
Data Serialization Components are - Thrift and Avro
Data Intelligence Components are - Apache Mahout and Drill.

27. What is Hadoop streaming? [CO5-L1]
Hadoop distribution has a generic application programming interface for writing Map and Reduce jobs in any desired programming language like Python, Perl, Ruby, etc. This is referred to as Hadoop Streaming. Users can create and run jobs with any kind of shell scripts or executable as the Mapper or Reducers.

28. What is the best hardware configuration to run Hadoop? [CO5-L1]
The best configuration for executing Hadoop jobs is dual core machines or dual processors with 4GB or 8GB RAM that use ECC memory. Hadoop highly benefits from using ECC memory though it is not low - end. ECC memory is recommended for running Hadoop because most of the Hadoop users have experienced various checksum errors by using non ECC memory. However, the hardware configuration also depends on the workflow requirements and can change accordingly.

29. What are the most commonly defined input formats in Hadoop? [CO5-L1]
The most common Input Formats defined in Hadoop are:

30. What is a block and block scanner in HDFS?
Block - The minimum amount of data that can be read or written is generally referred to as a “block” in HDFS. The default size of a block in HDFS is 64MB.
Block Scanner - Block Scanner tracks the list of blocks present on a DataNode and verifies them to find any kind of checksum errors. Block Scanners use a throttling mechanism to reserve disk bandwidth on the datanode.

31. On what concept the Hadoop framework works? [CO5-L1]
Hadoop Framework works on the following two core components-
1)HDFS – Hadoop Distributed File System is the java based file system for scalable and reliable storage of large datasets. Data in HDFS is stored in the form of blocks and it operates on the Master Slave Architecture.
2)Hadoop MapReduce-This is a java based programming paradigm of Hadoop framework that provides scalability across various Hadoop clusters. MapReduce distributes the workload into various tasks that can run in parallel.

Hadoop jobs perform 2 separate tasks- job. The map job breaks down the data sets into key-value pairs or tuples. The reduce job then takes the output of the map job and combines the data tuples to into smaller set of tuples. The reduce job is always performed after the map job is executed.
Here is a visual that clearly explain the HDFS and Hadoop MapReduce Concepts-

32. Name some companies that use Hadoop. [CO5-L1]
Yahoo (One of the biggest user & more than 80% code contributor to Hadoop)
Facebook
Netflix
Amazon
Adobe
eBay
Hulu
Spotify
Rubikloud
Twitter

33. What are the core components of Hadoop framework? [CO5-L1]
Hadoop consists of the following components:

1. Hadoop Common
2. Hadoop Distributed File System(HDFS)
3. Hadoop MapReduce
4. Hadoop Yet Another Resource Negotiator(YARN)(MapReduce 2.0)

34. What is HDFS? Explain its Features. [CO5-L1]
HDFS is a distributed file system that provides a limited interface for managing the file system to allow it to scale and provide high throughput. HDFS creates multiple replicas of each data block and distributes them on computers throughout a cluster to enable reliable and rapid access.
<u>Main Features of Hadoop:</u>

1. HDFS creates multiple replicas of data blocks for reliability, placing them on the computer nodes around the cluster.
2. Hadoop's target is to run on clusters of the order of 10,000 nodes.
3. A File consists of many 64 MB blocks.

35. What is Namenode? [CO5-L1]
NameNode is the master that contains the metadata. In general, it maintains the directories and files and manages the blocks which are present on the DataNode. The following are the functions of NameNode:

1. Manages namespace of the file system in memory.
2. Maintains "inode" information.

- **What is Datanode?** [CO5-L1]

DataNodes are the slaves which provide the actual storage and are deployed on each machine. They are responsible for processing read and write requests for the clients. The following are the other functions of DataNode:

1. Handles block storage on multiple volumes and also maintain block integrity.
2. Periodically sends heartbeats and also the block reports to NameNode.

33. What is JobTrackers?

JobTracker is the master which manages the jobs and resources in the cluster. The JobTracker tries to schedule each map on the TaskTracker which is running on the same DataNode as the underlying block.

Part – B

1. Write a short note on the Hadoop Framework[CO5-L1]

Hadoop Framework

Apache Hadoop is a framework that allows distributed processing of large data sets across clusters of commodity computers using a simple programming model. It is designed to scale-up from single servers to thousands of machines, each providing computation and storage. Rather than rely on hardware to deliver high-availability, the framework itself is designed to detect and handle failures at the application layer, thus delivering a highly available service on top of a cluster of computers, each of which may be prone to failures.

In short, Hadoop is an open-source software framework for storing and processing big data in a distributed way on large clusters of commodity hardware. Basically, it accomplishes the following two tasks:

1. Massive data storage.
2. Faster processing.

Advantage of Hadoop

Problems in data transfer made the organizations to think about an alternate way. The following examples explain the use of Hadoop.

The transfer speed is around 100 MB/s and a standard disk is 1 TB. Time to read entire disk = 10,000 s or 3 h!. Then increase in processing time may not be very helpful because of two reasons:

- Network bandwidth is now more of a limiting factor.
- Physical limits of processor chips are reached.

If 100 TB of datasets are to be scanned on a 1000 node cluster, then in case of

- remote storage with 10 Mbps bandwidth, it would take 165 min.
- local storage with 50 Mbps, it will take 33 min.

So it is better to move computation rather than moving data.

Taking care of hardware failure cannot be made optional in Big Data Analytics but has to be made as a rule. In case of 1000 nodes, we need to consider say 4000 disks, 8000 core, 25 switches, 1000 NICs and 2000 RAMs (16 TB). Meantime between failures could be even less than a day since commodity hardware is. used. There is a need for fault tolerant store to guarantee reasonable availability.

Hadoop Goals

The main goals of Hadoop are listed below:

1. **Scalable**: It can scale up from a single server to thousands of servers.
2. **Fault tolerance:** It is designed with very high degree of fault tolerance.
3. **Economical:** It uses commodity hardware instead of high-end hardware.
4. **Handle hardware failures:** The resiliency of these clusters comes from the software’s ability to detect and handle failures at the application layer.

The Hadoop framework can store huge amounts of data by dividing the data into blocks and storing it across multiple computers, and computations can be run in parallel across multiple connected machines.

Hadoop gained its importance because of its ability to process huge amount of variety of data generated every day especially from automated sensors and social media using low-cost commodity hardware.

Since processing is done in batches, throughput is high but latency is low. Latency is the time (minutes/seconds or clock period) to perform some action or produce some result whereas throughput is the number of such actions executed or result produced per unit of time. The throughput of memory system is termed as memory bandwidth.

Hadoop Assumptions

Hadoop was developed with large clusters of computers in mind with the following assumptions:

1. Hardware will fail, since it considers a large cluster of computers.
2. Processing will be run in batches; so aims at high throughput as opposed to low latency.
3. Applications that run on Hadoop Distributed File System (HDFS) have large datasets typically from gigabytes to terabytes in size.
4. Portability is important.
5. Availability of high-aggregate data bandwidth and scale to hundreds of nodes in a single cluster.
6. Should support tens of millions of files in a single instance.
7. Applications need a write-once-read-many access model.

2. Explain the various core components of the Hadoop. [CO5-L2]

Core Components of Hadoop

Hadoop consists of the following components:

1. **Hadoop Common**: This package provides file system and OS level abstractions. It contains libraries and utilities required by other Hadoop modules.

2. **Hadoop Distributed File System (HDFS)**: HDFS is a distributed file system1 that provides a limited interface for managing the file system.
3. **Hadoop MapReduce**: MapReduce is the key algorithm that the Hadoop MapReduce engine uses to distribute work around a cluster.
4. **Hadoop Yet Another Resource Negotiator (YARN) (MapReduce 2.0)**: It is a resource- management platform responsible for managing compute resources in clusters and using them for scheduling of users' applications.

Hadoop Common Package

This consists of necessary Java archive (JAR) files and scripts needed to start Hadoop. Hadoop requires Java Runtime Environment (JRE) 1.6 or higher version. The standard start-up and shut-down scripts need Secure Shell (SSH) to be setup between the nodes in the cluster.

HDFS (storage) and MapReduce (processing) are the two core components of Apache Hadoop. Both HDFS and MapReduce work in unison and they are co-deployed, such that there is a single cluster that provides the ability to move computation to the data. Thus, the storage system HDFS is not physically separate from a processing system MapReduce.

Hadoop Distributed File System (HDFS)

HDFS is a distributed file system that provides a limited interface for managing the file system to allow it to scale and provide high throughput. HDFS creates multiple replicas of each data block and distributes them on computers throughout a cluster to enable reliable and rapid access. When a file is loaded into HDFS, it is replicated and fragmented into "blocks" of data, which are stored across the cluster nodes; the cluster nodes are also called the DataNodes. The NameNode is responsible for storage and management of metadata, so that when MapReduce or another execution framework calls for the data, the NameNode informs it where the data that is needed resides. Figure 1 shows the NameNode and DataNode block replication in HDFS architecture.

Main Features of Hadoop

1. HDFS creates multiple replicas of data blocks for reliability, placing them on the computer nodes around the cluster.
2. Hadoop's target is to run on clusters of the order of 10,000 nodes.
3. A file consists of many 64 MB blocks.

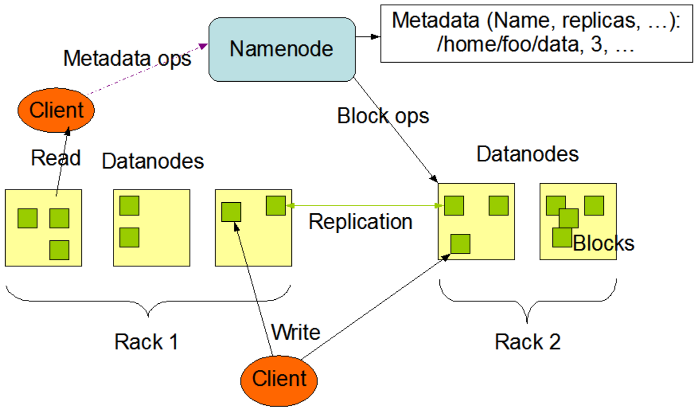
Main Components of HDFS NameNode

NameNode is the master that contains the metadata. In general, it maintains the directories and files and manages the blocks which are present on the DataNode. The following are the functions of NameNode:

1. Manages namespace of the file system in memory.
2. Maintains "inode" information.
3. Maps inode to the list of blocks and locations.

4. Takes care of authorization and authentication.
5. Creates checkpoints and logs the namespace changes.

HDFS Architecture



NameNode and DataNode block replication.

So the NameNode maps DataNode to the list of blocks, monitors status (health) of DataNode and replicates the missing blocks.

DataNodes

DataNodes are the slaves which provide the actual storage and are deployed on each machine. They are responsible for processing read and write requests for the clients. The following are the other functions of DataNode:

1. Handles block storage on multiple volumes and also maintain block integrity.
2. Periodically sends heartbeats and also the block reports to NameNode.

Figure 2 shows how HDFS handles job processing requests from the user in the form of Sequence Diagram. User copies the input files into DFS and submits the job to the client. Client gets the input file information from DFS, creates splits and uploads the job information to DFS. JobTracker puts ready job into the internal queue. JobScheduler picks job from the queue and initializes the job by creating job object. JobTracker creates a list of tasks and assigns one map task for each input split. TaskTrackers send heartbeat to JobTracker to indicate if ready to run new tasks. JobTracker chooses task from first job in priority-queue and assigns it to the TaskTracker.

Secondary NameNode is responsible for performing periodic checkpoints. These are used to restart the NameNode in case of failure. MapReduce can then process the data where it is located.

MapReduce

The MapReduce algorithm aids in parallel processing and basically comprises two sequential phases: map and reduce.

1. In the **map phase**, a set of key-value pairs forms the input and over each key—value pair, the desired function is executed so as to generate a set of intermediate key—value pairs.
2. In the **reduce phase**, the intermediate key—value pairs are grouped by key and the values are combined together according to the reduce algorithm provided by the user. Sometimes no reduce phase is required, given the type of operation coded by the user.

MapReduce processes are divided between two applications, JobTracker and TaskTracker at the cluster level. JobTracker is responsible for scheduling job runs and managing computational resources across the cluster; hence it runs on only one node of the cluster. Each MapReduce job is split into a number, of tasks which are assigned to the various TaskTrackers depending on which data is stored on that node. So TaskTracker runs on every slave node in the cluster. JobTracker oversees the progress of each TaskTracker as they complete their individual tasks.

The following points summarize the above discussion:

1. Hadoop implements Googles MapReduce, using HDFS.
2. MapReduce divides applications into many small blocks of work.
3. Performs Sort/merge-based distributed computing.
4. Follows functional style programming and so naturally is parallelizable across a large cluster of workstations or PCs.

In the MapReduce paradigm, each job has a user-defined map phase followed by a user-defined reduce phase as follows:

1. Map phase is a parallel, share-nothing processing of input.
2. In the reduce phase, the output of the map phase is aggregated.

HDFS is the storage system for both input and output of the MapReduce jobs.

Main Components of MapReduce

The main components of MapReduce are listed below:

1. **JobTrackers:** JobTracker is the master which manages the jobs and resources in the cluster. The JobTracker tries to schedule each map on the TaskTracker which is running on the same DataNode as, the underlying block.
2. **TaskTrackers:** TaskTrackers are slaves which are deployed on each machine in the cluster. They are responsible for running the map and reduce tasks as instructed by the JobTracker.
3. **JobHistoryServer:** JobHistoryServer is a daemon that saves historical information about completed tasks/applications.

Note: If the map phase has M fragments and the reduce phase has R fragments, then M and R should be much larger than the number of worker machines. R is often decided by the users, because the output of each reduce task ends up in a separate output file. Typically (at Google), M = 2,00,000 and R = 5000, using 2000 worker machines.

Yet Another Resource Negotiator (YARN)

YARN addresses problems with MapReduce 1.0s architecture, specifically the one faced by JobTracker service.

Hadoop generally has up to tens of thousands of nodes in the cluster. Obviously, MapReduce 1.0 had issues with scalability, memory usage, synchronization, and also Single Point of Failure (SPOF) issues. In effect, YARN became another core component of Apache Hadoop.

Functions of YARN

It splits up the two major functionalities “resource management” and “job scheduling and monitoring” of the JobTracker into two separate daemons. One acts as a “global Resource Manager (RM)” and the other as a “ApplicationMaster (AM)” per application. Thus, instead of having a single node to handle both scheduling and resource management for the entire cluster, YARN distributes this responsibility across the cluster.

The RM and the NodeManager manage the applications in a distributed manner. The RM is the one that arbitrates resources among all the applications in the system. The per-application AM negotiates resources from the RM and works with the NodeManager(s) to execute and monitor the component tasks.

1. The RM has a scheduler that takes into account constraints such as queue capacities, user-limits, etc. before allocating resources to the various running applications.
2. The scheduler performs its scheduling function based on the resource requirements of the applications.
3. The NodeManager is responsible for launching rhe applications’ containers. It monitors the application’s resource usage (CPU, memory, disk, network) and reports the information to the RM.
4. Each AM runs as a normal container. It has the responsibility of negotiating appropriate resource containers from the scheduler, tracking their status and monitoring their progress.

3. Explain in detail the Ecosystem of the Hadoop Framework. [CO5-L2]

Hadoop Ecosystem

Apart from HDFS and MapReduce, the other components of Hadoop ecosystem are shown in the figure below. The main ecosystems components of Hadoop architecture are as follows:

1. Apache HBase: Columnar (Non-relational) database.
2. Apache Hive: Data access and query.

3. Apache HCatalog: Metadata services.
4. Apache Pig: Scripting platform.
5. Apache Mahout: Machine learning libraries for Data Mining.
6. Apache Oozie: Workflow and scheduling services.
7. Apache ZooKeeper: Cluster coordination.
8. Apache Sqoop: Data integration services.

These components are discussed in detail in the following subsections.

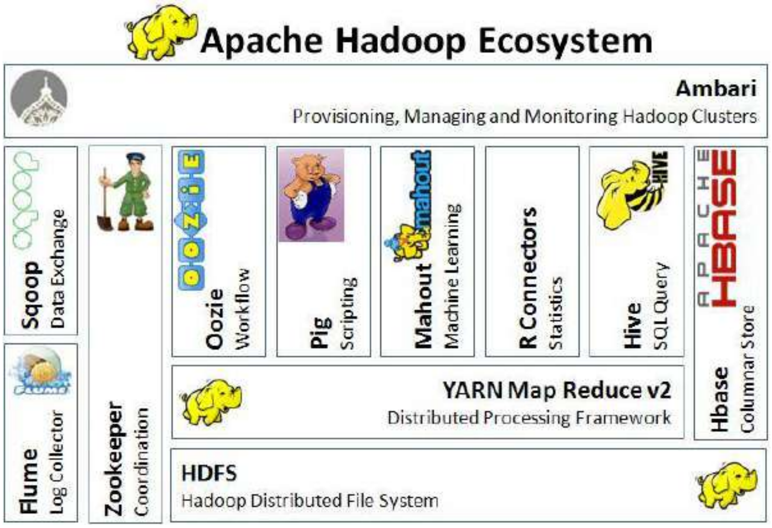


Figure 3 Hadoop ecosystem.

HBase

HBase “is an open-source, distributed, versioned, column-oriented store” that sits on top of HDFS. HBase is based on Google’s Bigtable. HBase is based on columns rather than rows. This essentially increases the speed of execution of operations if they are need to be performed on similar values across massive datasets; for example, read/write operations that involve all rows but only a small subset of all columns. HBase does not provide its own query or scripting language, but is accessible through Java, Thrift and REST APIs.

Hive

Hive provides a warehouse structure for other Hadoop input sources and SQL-like access for data in HDFS. Hive’s query language, HiveQL, compiles to MapReduce and also allows user-defined functions (UDFs). Hive’s data model is based primarily on three related data structures: tables, partitions and buckets. Tables correspond to HDFS directories that are divided into partitions, which in turn can be divided into buckets.

HCataiog

HCatalog is a metadata and table storage management service for HDFS. HCatalog’s goal is to simplify the user’s interaction with HDFS data and enable data sharing between tools and execution platforms.

Pig

Pig is a run-time environment that allows users to execute MapReduce on a Hadoop cluster. Pig Latin is a high-level scripting language on Pig platform. Like HiveQL in Hive, Pig Latin is a higher-level language that compiles to MapReduce.

Pig is more flexible with respect to possible data format than Hive due to its data model. Pig's data model is similar to the relational data model, but here tuples can be nested. For example, a table of tuples can have a table in the third field of each tuple. In Pig, tables are called bags. Pig also has a "map" data type, which is useful in representing semi-structured data such as JSON or XML."

Sqoop

Sqoop ("SQL-to-Hadoop") is a tool which transfers data in both ways between relational systems and HDFS or other Hadoop data stores such as Hive or HBase. Sqoop can be used to import data from external structured databases into HDFS or any other related systems such as Hive and HBase. On the other hand, Sqoop can also be used to extract data from Hadoop and export it to external structured databases such as relational databases and enterprise data warehouses.

Oozie

Oozie is a job coordinator and workflow manager for jobs executed in Hadoop. It is integrated with the rest of the Apache Hadoop stack. It supports several types of Hadoop jobs, such as Java map-reduce,

Streaming map-reduce, Pig, Hive and Sqoop as well as system-specific jobs such as Java programs and shell scripts. An Oozie workflow is a collection of actions and Hadoop jobs arranged in a Directed Acyclic Graph (DAG), since tasks are executed in a sequence and also are subject to certain constraints.

Mahout

Mahout is a scalable machine-learning and data-mining library. There are currently following four main groups of algorithms in Mahout:

1. Recommendations/Collective filtering.
2. Classification/Categorization.
3. Clustering.
4. Frequent item-set mining/Parallel frequent pattern mining.

Mahout is not simply a collection of pre-existing data mining algorithms. Many machine learning algorithms are non-scalable; that is, given the types of operations they perform, they cannot be executed as a set of parallel processes. But algorithms in the Mahout library can be executed in a distributed fashion, and have been written for MapReduce

ZooKeeper

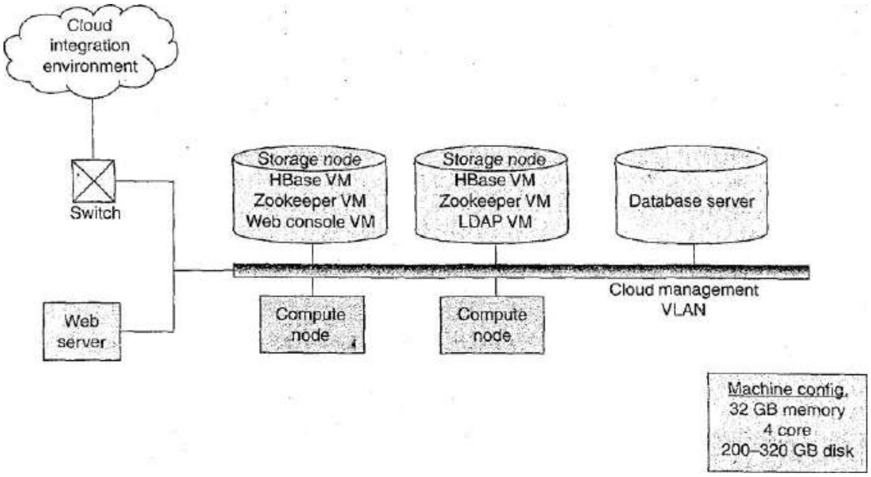
ZooKeeper is a distributed service with master and slave nodes for storing and maintaining configuration information, naming, providing distributed synchronization and providing group services in memory on ZooKeeper servers. ZooKeeper allows distributed processes to coordinate with each other through a shared hierarchical name space of data registers called znodes. HBase depends on ZooKeeper and runs a ZooKeeper instance by default.

4. Illustrate the physical architecture required to support Big Data Analytics[CO5-L2]

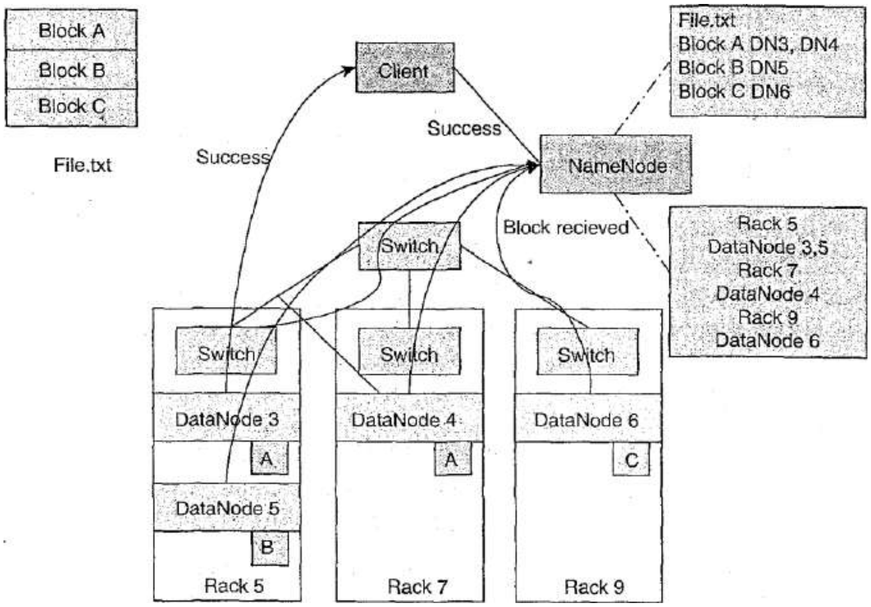
Physical Architecture

Organizations tend to store more and more data in cloud environments, since clouds offer business users scalable resources on demand. Combining processor-based servers and storage, along with networking resources used in cloud environments, with big data processing tools such as Apache Hadoop software, provides the high-performance computing power needed to analyse vast amounts of data efficiently and cost-effectively. The machine configuration for storage and computing servers typically are 32 GB memory, four core processors and 200—320 GB hard disk. Running Hadoop in virtualized environments continues to evolve and mature with initiatives from open-source software projects. The figure below shows cloud computing infrastructure required for Big Data Analytics.

Every Hadoop-compatible file system should provide location awareness for effective scheduling of work: the name of the rack or the network switch where a worker node is. Hadoop application uses this information to find the data node and run the task. HDFS replicates data to keep different copies of the data on different racks. The goal is to reduce the impact of a rack power or switch failure.



Cloud computing infrastructure to support Big Data Analytics.



Hadoop-compatible file system provides location awareness.

A small Hadoop cluster includes a single master and multiple worker nodes. The master node consists of a JobTracker, TaskTracker, NameNode and DataNode. A slave or worker node acts as both a DataNode and TaskTracker, though it is possible to have data-only worker nodes and compute-only worker nodes.

In case of a larger cluster, the HDFS is managed through a dedicated NameNode server to host the file system index, and a secondary NameNode that can generate snapshots of the NameNodes memory structures, thus reducing the impact of loss of data. Similarly, a standalone JobTracker server can manage job scheduling. In clusters where the Hadoop MapReduce engine is deployed against an alternate file system, the NameNode, secondary NameNode and DataNode architecture of HDFS are replaced by the file-system-specific equivalents.

HDFS stores large files in the range of gigabytes to terabytes across multiple machines. It achieves reliability by replicating the data across multiple hosts. Data is replicated on three nodes: two on the same rack and one on a different rack. Data nodes can communicate with each other to re-balance data and to move copies around. HDFS is not fully POSIX-compliant to achieve increased performance for data throughput and support for non-POSIX operations such as Append.

The HDFS file system includes a so-called secondary NameNode, which regularly connects with the primary NameNode and builds snapshots of the primary NameNode directory information, which the system then saves to local or remote directories. These check-pointed images can be used to restart a failed primary NameNode without having to replay the entire journal of file-system actions, then to edit the log to create an up-to-date directory structure.

An advantage of using HDFS is data awareness between the JobTracker and TaskTracker. The JobTracker schedules map or reduce jobs to TaskTrackers with an awareness of the data location. For example, if node A contains data (x, y, z) and node B contains data {a, b, c), the JobTracker schedules node B to perform map or reduce tasks on (a, b, c) and node A would be scheduled to perform map or reduce tasks on (x, y, z). This reduces the amount of traffic that goes over the network and prevents unnecessary data transfer.

When Hadoop is used with other file system, this advantage is not always available. This can have a significant impact on job-completion times, which has been demonstrated when running data-intensive jobs. HDFS was designed for mostly immutable files and may not be suitable for systems requiring concurrent write-operations.

5. List out and explain the major limitations of the Hadoop Farework [CO5-L2]

Hadoop Limitations

HDFS cannot be mounted directly by an existing operating system. Getting data into and out of the HDFS file system can be inconvenient. In Linux and other Unix systems, a file system in Userspace (FUSE) virtual file system is developed to address this problem.

File access can be achieved through the native Java API, to generate a client in the language of the users' choice (C++, Java, Python, PHP, Ruby, etc.), in the command-line interface or browsed through the HDFS-UI web app over HTTP.

The following are the primary limitations of Hadoop:

1. **Security Concerns**: Hadoop security model is disabled by default due to sheer complexity. Whoever's managing the platform should know how to enable it; else data could be at huge risk. Hadoop does not provide encryption at the storage and network levels, which is a major reason for the government agencies and others not to prefer to keep their data in Hadoop framework.
2. **Vulnerable By Nature**: Hadoop framework is written almost entirely in Java, one of the most widely used programming languages by cyber-criminals. For this reason, several experts have suggested dumping it in favor of safer, more efficient alternatives.
3. **Not Fit for Small Data**: While big data is not exclusively made for big businesses, not all big data platforms are suitable for handling small files. Due to its high capacity design, the HDFS lacks the ability to efficiently support the random reading of small files. As a result, it is not recommended for organizations with small quantifies of data.
4. **Potential Stability Issues**: Hadoop is an open-source platform necessarily created by the contributions of many developers who continue to work on the

project. While improvements are constantly being made, like all open-source software, Hadoop has stability issues. To avoid these issues, organizations are strongly recommended to make sure they are running the latest stable version or run it under a third-party vendor equipped to handle such problems.

5. **General Limitations**: Google mentions in its article that Hadoop may not be the only answer for big data. Google has its own Cloud Dataflow as a possible solution. The main point the article stresses is that companies could be missing out on many other benefits by using Hadoop alone.

6. Write a short note on Hive. [CO5-L2]

Hive

We have seen that the Hadoop ecosystem contains different sub-projects such as Sqoop, Pig and Hive that are used to help Hadoop modules. Let us recall Sqoop, Pig and Hive:

1. **Sqoop**: For importing and exporting data to and from between HDFS and relational database management system (RDBMS).
2. **Pig**: A procedural language platform for developing a script for MapReduce operations.
3. **Hive**: A platform for developing SQL typescripts to do MapReduce operations.

The following are the various ways to execute MapReduce operations:

1. The traditional approach uses Java MapReduce programme for structured, semi-structured and unstructured data.
2. The scripting approach is used for MapReduce to process structured and semi-structured data using Pig.
3. The HiveQL or HQL for MapReduce to process structured data using Hive.

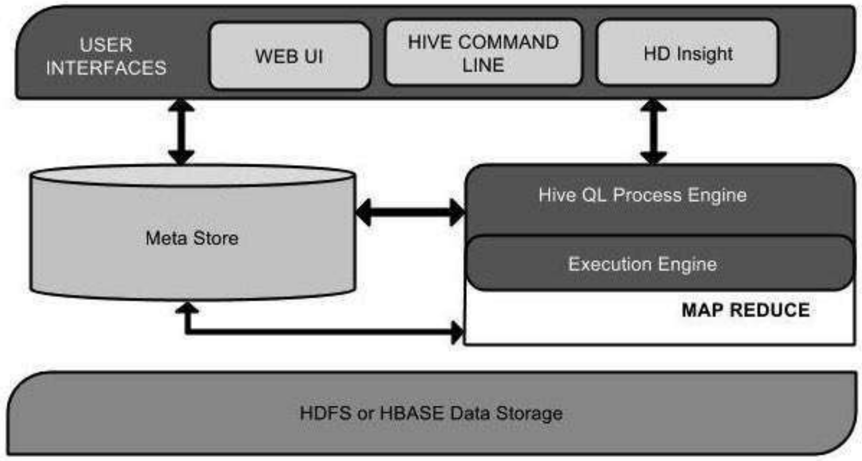
Features of Hive

Hive was originally developed by Facebook; later ASF further developed and released it as an open- source. Amazon uses it in Amazon Elastic MapReduce.

It is used to process structured data in Hadoop. It is a data warehouse infrastructure tool resides on top of Hadoop. It summarizes big data, and so querying and analyzing big data is made easy. Hive is not a relational database and so is not used for online transaction processing. It is designed for OLAP. The following explains the features of Hive:

1. It is familiar, fast, scalable and extensible.
2. It stores database schema; the processed data is stored into HDFS.
3. It provides SQL-type language for querying called HiveQL or HQL.

Architecture of Hive



1. **Hive User Interface**: Hive creates interaction between user and HDFS through Hive Web UI, Hive command line and in Windows server Hive HD insight.
2. **Metadata Store**: Hive stores the database schema and its HDFS mapping in the database server.
3. **HDFS/HBase**: HDFS or HBASE are the data storage techniques to store data into the file system.
4. **HiveQuery Processing Engine (HiveQL)**: HiveQL is used for querying on schema info on the metadata store. Instead of writing MapReduce programme in Java, a query can be written for MapReduce job and processed.
5. **Execution Engine**: The execution engine is used to processes the query and generates results as same as MapReduce results.

The following steps explain the working principles of Hive:

1. The Hive user interface (Web Ul/command line/HD insight) sends a query to database driver (JDBC/ODBC/) to execute.
2. The driver with the help of query compiler that parses the query, checks the syntax and the requirement of the query.
3. The compiler then sends the metadata request to database where the metadata is stored.
4. The database sends the response to the compiler.
5. The compiler sends the response to the driver which is passed to the execution engine.
6. The execution engine (MapReduce process) sends the job to task to JobTracker, which is in NameNode and it assigns this job to TaskTracker, which is in DataNode.
7. The execution engine will receive the results from DataNodes.
8. The execution engine then sends the results to the driver.

9. The driver sends it to UI.

7. **Explain how MapReduce collaborates with the Hadoop Software stack in the Data Analytics process. [CO5-L2]**

MapReduce and The Software Stack

Businesses and governments need to analyze and process a tremendous amount of data in a very short period of time. The processing is to be done on a large amount of data, which will take a huge amount of time if done on a single machine. So the idea is to divide the data into smaller chunks and send to a cluster of machines, where they can be processed simultaneously and then the results can be combined.

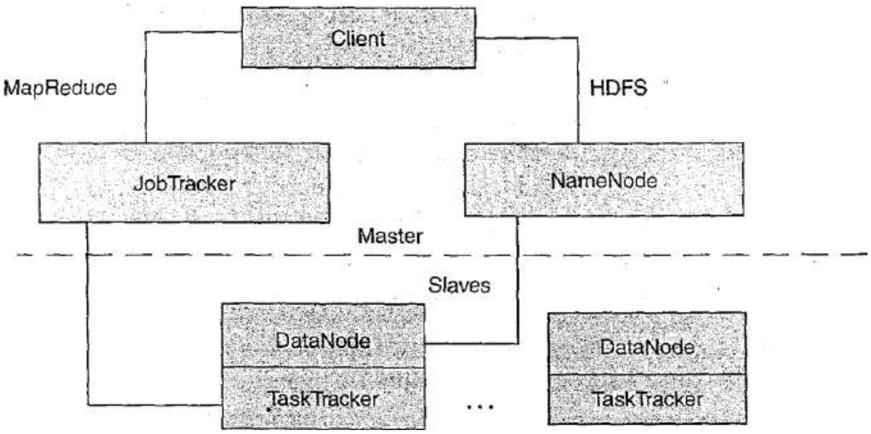
Huge increase in data generated from social network and other blogging sites, for example, “Friends” on social networking sites, has led to increase in the volume of graphic data with millions of nodes and edges. This led to the creation of a new software stack. This new software stack provides parallelism by using several commodity hardware connected by Ethernet or switches. Hadoop is a framework for large-scale distributed batch processing. Hadoop can be deployed on a single machine if the data can be handled by the machine, but it is mainly designed to efficiently distribute a large amount of data for processing across a set of machines. Hadoop includes a distributed file system (DFS) that splits the input data and sends these portions of the original data to several machines in the defined cluster to hold. Main focus of this new software stack is MapReduce, a high-level programming system. This helps in doing the computation of the problem in parallel using all the connected machines so that the output, results are obtained in an efficient manner. DFS also provides data replication up to three times to avoid data loss in case of media failures.

Figure 7 shows the role of client machines, Master and Slave Nodes in Hadoop deployment. The MasterNode stores the huge data Hadoop Distributed File System (HDFS) and runs parallel computations on all that data (MapReduce).

1. The NameNode coordinates and monitors the data storage function (HDFS), while the JobTracker coordinates the parallel processing of data using MapReduce.
2. SlaveNode does the actual work of storing the data and running the computations. Master- Nodes give instructions to their SlaveNodes. Each slave runs both a DataNode and aTaskTracker daemon that communicate with their respective MasterNodes.
3. The DataNode is a slave to the NameNode.
4. The TaskTracker is a slave to the JobTracker.

Distributed File Systems

Most scientific applications in the past, required to do parallel processing for fast computing, used special-purpose computers. Web services enabled the use of commodity nodes (having RAM, CPU and. hard disk) to execute the chosen services independently on the nodes and this reduced the cosi ul using special-purpose machines for parallel computing. In recent times, the new parallel-compminp architecture called cluster computing is in use. Compute nodes typically in the range of 8—64 are stored in racks and are connected with each other by Ethernet or switch to the network. Failure at the node level (disk failure) and at the rack level (network failure) is taken care of by replicating data in second ary nodes. All the tasks are completed independently and so if any task fails, it can be re-started withom affecting the other tasks.



File system stores data permanently. The system has logical drives and is layered on top of physical storage medium. It is addressed by a file name under a directory that supports hierarchical nesting. Access to the file is through file path consisting of drive, directory(s) and filename.

DFS supports access to files that are stored on remote servers. It also offers support for replication and local caching. Concurrent access to files read/write has to be taken care of using locking conditions, Different types of implementations are available based on the complexity of applications.

Google File System

Google had to store a massive amount of data. It needs a good DFS with cheap commodity computers to reduce cost. These commodity computers are unreliable, hence redundant storage is required to manage failures. Most of the files in Google file system (GFS) are written only once and sometimes appended. But it needs to allow large streaming reads and so high-sustained throughput is required over low latency. File sizes are typically in gigabytes and are stored as chunks of 64 MB each. Each of these chunks is replicated thrice to avoid information loss due to the failure of the commodity hardware. These chunks are centrally managed through a single master that stores the metadata information about the chunks. Metadata stored on the master has file and chunk namespaces, namely, mapping of file to chunks and location of the replicas of each chunk. Since Google users do a lot of streaming read of large data sets, caching has no importance or benefit. What if the master fails? Master is replicated in shadow master. Also the master involvement is reduced by not moving data through it; metadata from master is cached at clients. Master chooses one of the replicas of chunk as primary and delegates the authority for taking care of the data mutations.

Hadoop Distributed File System

HDFS is very similar to GFS. Here, the master is called NameNode and shadow master is called Secondary NameNode. Chunks are called blocks and chunk server is called DataNode. DataNode stores and retrieves blocks, and also reports the list of blocks it is storing to NameNode. Unlike GFS, only single-writers per file is allowed and no append record operation is possible. Since HDFS is an open-source, interface, libraries for different file systems are provided.

Physical Organization of Compute Nodes

Hadoop runs best on Linux machines. Hadoop is installed in client machines with all the cluster settings. The client machine loads data and MapReduce program into the cluster, and then retrieves or views the results once the program is executed. For smaller clusters, where the number of nodes is less than 40, a single physical server can host both JobTracker and NameNode. For medium and large clusters,both of them can be in different physical servers. The “server virtualization” or “hypervisor layer” adds to overhead and impedes the Hadoop performance. Hadoop does work in a virtual machine. Cluster (with a few nodes) can be up and running in VMware Workstation on a laptop machine.