UNIT I

INTRODUCTION AND LIFE CYCLE

1.1 INTRODUCTION

In 21st century rapid growth happened in the field of Information Technology (IT). In every part of day to day life IT plays an important role such as health, business/industries, education, finance etc. In today's era to sustain in current competitive world, we need to use IT and its various applications. Information Technology is nothing but a computer based information system built by study, design, development, application, implementation, support and manages to serve requirement current business processes.

Due to use of this information technology system, huge amount of data is being generated every data an alarming rate. This huge amount of data is called as BIG DATA.

This data is generated through various sources as mobile phones, Social media like Facebook, Twitter and Instagram, Video Surveillance, Medical Imagining, Gene Sequencing and Geographical data shown in Fig. 1.1.

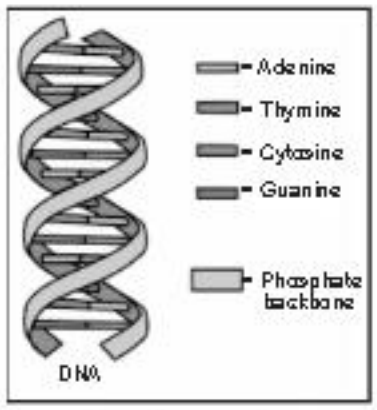
This Big Data is creating good opportunities for IT industries and other businesses to improve quality, efficiency, product services, level of customer satisfaction and profit. This is also good domain for academia and researchers to contribute in the field of data analysis.

Mobile Phones

Social Media Like Facebook, Twitter, Instagram

Video Surveillance

Medical Imagining



Gene Sequencing

Geographical Data

Fig. 1.1 : Sources of data

1.2 BIG DATA OVERVIEW

Imagine every day, every hour, every minute and every second data is generated by various resources. For Example -

- Every day around 1.5 millions payments are done by using PayPal.
- Every hour Walmart does more than 1 million customer transactions.
- Every minute millions of peoples are doing comments, status updates and photo uploads on Facebook.
- Every second thousands of tweets come on twitter.
- This shows that data is growing from Terabytes to Exabytes. This data have Volume, Variety, Velocity and veracity as shown in Fig. 1.2. This data have different forms as structured, semi structured, quasi structured and unstructured. This data is homogenous as well as heterogeneous in nature so such data is called as BIG DATA.
- Several industries are using this Big Data for various applications such as credit card fraud identification, attractive promotional offers to gold and platinum customers, recommendations of different products based on browsing history of individuals on social networking websites.

1.2.1 Big Data Characteristics

Three Attributes Defining Big Data Characteristics :

1. **Volume :**

   Big data is big in volume i.e. huge in volume. It has billions of rows and millions of columns.

2. **Data Complexity and Structures :**

   Big data have variety of sources, formats and structures. It also includes digital traces on web and other digital repositories.

3. New Data Creation Speed and Growth :

Big Data have high velocity data i.e. rapidly growing in nature.

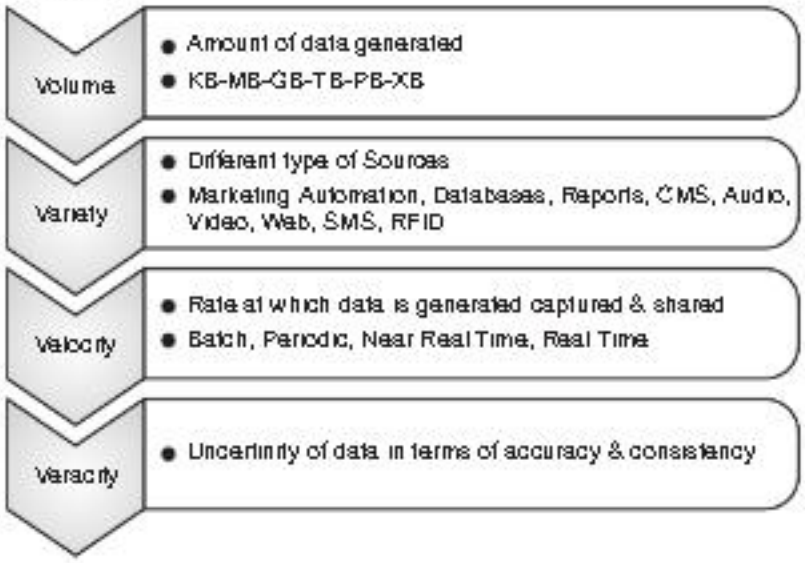


Fig. 1.2 : Big data V's

Due to above characteristics Big Data cannot be fully analyzed by traditional database technologies. Fig. 1.3 shows various components of Big Data. It needs new tools and technologies to store, manage and analyze. These tools and techniques designed and developed specifically to handle large data sets and to extract appropriate business knowledge.

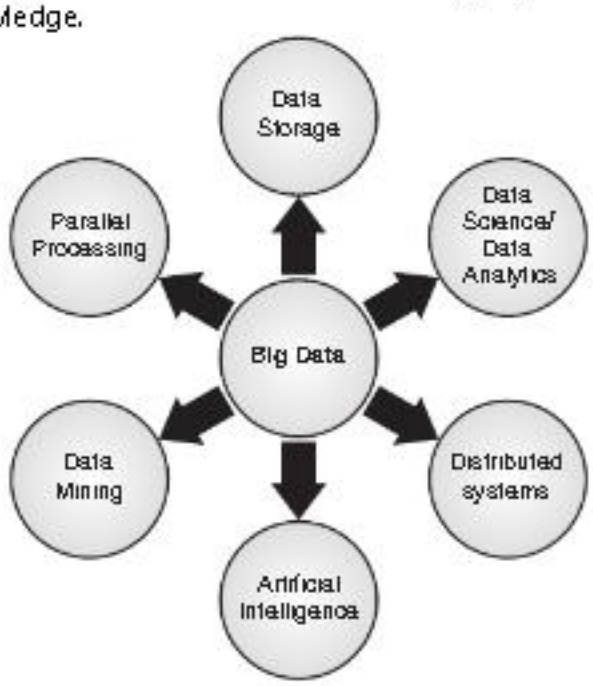


Fig. 1.3 : Components of big data

1.2.2 Data Structures

- Big data can be various forms as structured, semi structured, quasi structured and unstructured as shown in Fig. 1.4. Example : textual data, multimedia data (Audio/Video), financial sheets, genetic mappings etc.
- To process this unstructured or semi structured data, there is need of distributed and massively parallel processing environment.

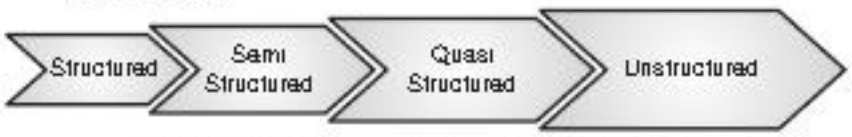


Fig. 1.4 : Data structures (Data is growing from structured to unstructured)

A. Structured Data

Structured data follows a predefined format, data type and schema (transaction data, Online analytical processing [OLAP] data cubes, traditional RDBMS, CSV files, and simple spreadsheets).

Example :

Roll_No	FName	MName	LName	Mathematics	Science	English	Marathi	Hindi
1	Suresh	Nagesh	Kulkarni	80	89	87	91	92
2	Kalish	Kumar	Joshi	99	98	92	92	95

B. Semi Structured Data

Semi structured data is self describing structures as Extensible Markup Language [XML] data files that are self-describing and defined by an XML schema.

C. Quasi Structured Data

Quasi structures data is textual data with inconsistent data formats. These formats are formatted with effort, tools, and time. Example : Web clickstream data that may contain inconsistencies in data values and formats.

D. Unstructured Data

Unstructured data has no predefined structure/schema. It may include text documents, PDFs, images, and video.

1.2.3 Data Repositories : Analyst Perspespective

Data repositories play an important role in data analysis. Some of them are as follows

1. Spread Sheets :

Spread sheets used by business users to create simple logic on data structured in rows and columns and create their own analyses of business problems. Database administer training is not required to create these sheets. They are easy to share but there can be problem of many versions and who have most recent updated data with logic.

2. Data Warehouses :

Data warehouse is type of repository to manage growing data centrally. It provides security, fault tolerance. It is a single repository where users are getting data from official and authenticated resources. Multidimensional data can be stored using OLAP cubes and BI Analytical tools are used to process this data. Advanced techniques such as regressions and neural networks are used to perform in-depth analytics.

Enterprise Data Warehouses (EDWs) are solving problem of multiple versions of a spreadsheet. EDW and BI strategies provides centrally store, manage and security to data feeds.

EDW is managed by data administrators which leads data analyst to wait for longer time to get data. EDW have strict rules which restrict analyst to create own dataset. Locally created data sets are not secured, managed and backed up properly. Analyst perspective EDW and BI provides data accuracy and availability but introduces problem of flexibility and agility.

3. Analytical Sandbox :

Analytical sandboxes are also called as workspaces or stand alone analytical data marts. They are designed to enable teams to explore many datasets in a controlled fashion. They are not used for enterprise level financial reporting and sales dashboards.

In these sandboxes data sets are gathered from multiple sources and technologies for analysis. These sandboxes enable flexible, high-performance analysis in a nonproduction environment. Analytical Sandboxes can store variety of data, such as raw data, textual data, unstructured data, without interfering with financially critical databases.

The key components of an analytical sandbox as shown in Fig. 1.5.

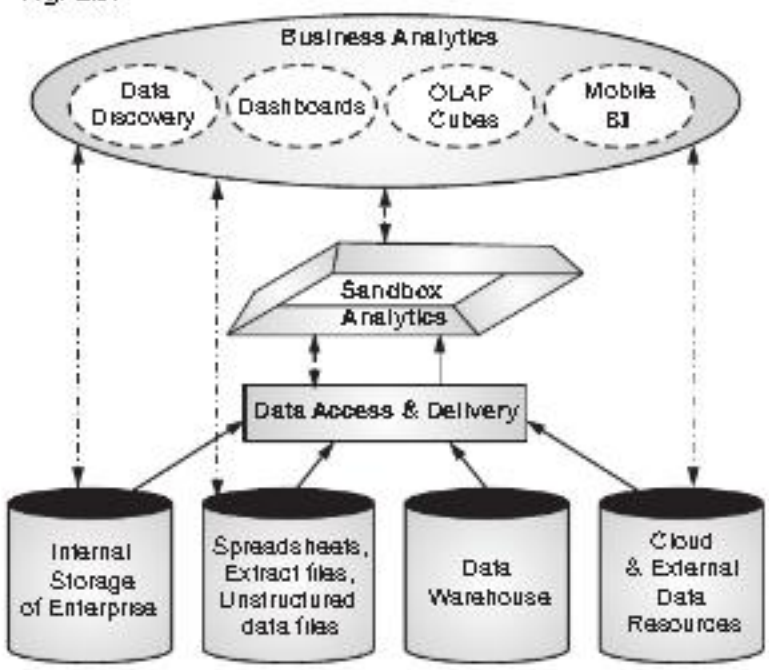


Fig. 1.5 : Components of analytical sandbox

- **Business Analytics** : This layer contains the self-service BI tools. These tools are used for data discovery, data visualization, OLAP, dash boards, Mobile BI, reports,
- **Analytical Sandbox Platform** : This platform hosts many architectural options for the processing, storage and networking capabilities. Ex BI appliances, cloud infrastructure, Databases (SQL/NOSQL)
- **Data Access and Delivery** : This layer is able to access, filter, augment and integrate data from a variety of data sources and data types.
- **Data Sources** : In this layer data is sourced from within and outside the organization. Data can be structured or unstructured e.g., extracts, feeds, messages, spreadsheets and documents.

1.3 STATE OF THE PRACTICE IN ANALYTICS

Current business is more data driven and more opportunities are involved in data analytics.

There are different 4 categories of business problems where data analytics need to inculcate.

1. **Optimize Business Operations** ex. analyze sale, price, profit and efficiency.
2. **Identify Business Risk** ex. Identify fraud, reduce customer churn.
3. **Predict New Business Opportunities** ex. cross-sell, up-sell and best new customer prospects.
4. Comply with laws or regulatory requirements ex. Many compliance and regulatory laws added or changed every year which adds complexity and data requirements for organizations. Anti-Money Laundering (AML) and fraud prevention need to handle by using analytical techniques.

1.3.1 BI V/s Data Science

Table 1.1 describes difference in Traditional Business Intelligence and emerging Data Science.

Table 1.1 : Business Intelligence V/s Data Science

Sr. No.	Parameters	Business Intelligence	Data Science
1.	Perspective	BI has current or backward-looking approach based on information. BI tries to solve question : "What happened?"	Data Science has forward approach. It is predicting future based on information. Data Science to solve question : "what will happen if we do X?"
2.	Focus	BI gives detail reports, key performance indicators and trends.	Data science predicts howthis data may look in future i.e. patterns and experimentations.
3.	Process	BI process is static and comparative.	Data Science gives scope to exploration and experimentation to decide howto gather and manage data.
4.	Data sources	BI data sources are predefined due to static nature and data gathering is slow.	Data science has flexible structure. Data is adding rapidly due flexible nature.
5.	Transform	BI helps you to solve known questions.	Data Science encourages new questions due to its growing data.
6.	Storage	BI uses SQL and Warehouse data.	Data Science uses less structured (logs, blogs, SQL, NoSQL, cloud data etc.)
7.	Data quality	BI provides single version of quality data.	Data science provides more precision, confidence level and much wider probabilities with its findings.
8.	IT owned vs business owned	BI is owned and managed by IT department.	Data Science is owned and managed by Analyst.
9.	Methods	BI uses Analytical methods.	Data Science uses scientific methods.
10.	Tools	BI uses statics and visualization tools	Data Science uses Statistics, Machine Learning, NLP and Graph analysis tools

1.3.2 Current Analytical Architecture

Most of organizations have typical analytical architecture as shown in Fig. 1.6. Most of the organizations are using data warehouse which provides good support for reporting and simple data analysis. It fails to support more complex analysis.

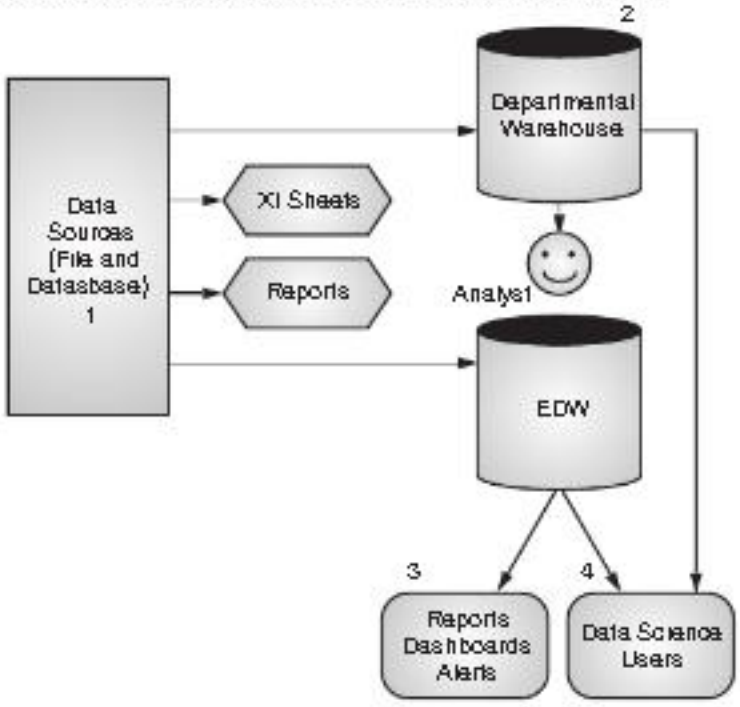


Fig. 1.6 : Current analytical architecture

1. **Data Sources :**
   - Data is loaded into the data warehouse.
   - Data needs to be well defined, structured and normalized with the appropriate data type definitions.
   - Data need to go through significant processing and checkpoints.
   - This data is centrally managed, secured and backed up.
2. **Departmental Warehouses :**
   - Local systems like Departmental warehouses and data marts may used by analyst to accommodate need of flexible analysis.
   - Local data marts may not have any security and structural constraints as EDW.
   - This allows users to do in-depth analysis.
3. **Data Analysis Reporting :**
   - In the data warehouse, data is read by additional applications across the enterprise for BI and reporting purposes.
   - Reports, Dashboards and Alerts are high-priority operational processes getting critical data feeds from the data warehouses and repositories.
4. **Data Science Users**
   - Analyst creates extract from EDW.
   - They analyze data offline by using R tool or any local analytical tools.
   - This is in-memory analytics as they are analyzing sample data rather than whole data.

Implications of Typical Analytic Architecture for Data Scientist

- EDW is designed for central data management and reporting so data for analysis are generally prioritized after operational processes.
- High-value data is hard to reach and leverage so predictive analytics and data mining activities are last in line for data.
- Data moves in batches from EDW to local analytical tools. This means that data scientists are performing in-memory analytics (such as with R, SAS, SPSS, or Excel).
- Analysts are working on only restricted size of data this can skew model accuracy.
- Data Science projects are not centrally managed, they will remain isolated and ad hoc. Due to this data science projects will exists as non standard initiatives and not aligned with business goals and strategies.

1.3.3 Drivers of Big Data

Drivers of big data are considered from two different angles : Business and Technology. Business entails market, sales and financial side of things, whereas, Technology has indicator/driver targeted towards technology and IT infrastructure side of things.

- **Data Driven Initiatives :** They are primarily categorized into 3 broad areas :
  1. **Data Driven Innovation :** I particularly like the innovation aspect with being data driven. Imagine being able to learn from your customer first what they need and having the ability to drive innovation through those uber targeted data indicators.
  2. **Data Driven Decision Making :** Data driven decision-making is the inherent ability of analytics to sieve through globs of data and identify the best path forward. Whether in terms of finding the best route to validating the current route and estimating the success/failure in current strategy. It takes decision making away from gut and focus on data backed reasoning for higher chances of success.
  3. **Data Driven Discovery :** Your data know a whole lot about you than you image. Having a discovery mechanism will help you to understand hidden insights that were not visible through traditional means.
- **Data Science as a Competitive Advantage :** I had the fortune of interacting with couple of mid size company's executives from commodity businesses. There had been a consistent outcry on having to build big data as a capability to add to their competitive advantage. With a proper data driven framework, businesses could build sustainable capabilities and further leverage these capabilities as a competitive edge. If businesses were able to master big data driven capabilities, businesses could use these capabilities to establish secondary source of revenues by selling it to other businesses.

- **Sustained Processes :** Data driven approach creates sustainable processes, which gives a huge endorsement to big data analytics strategy as a go for enterprise adoption. Randomness kills businesses and adds scary risks, while data driven strategy reduces the risk by bringing statistical models, which are measurable.
- **Cost Advantages of Commodity Hardware and Open Source Software :** How about the savings your IT will enjoy from moving things to commodity hardware and leverage more open source platforms for cost effective ways to achieve enterprise level computations and beyond. No more overpaying of premium hardware when similar or better analytical processing could be done using commodity and open source systems.
- **Quick Turnaround and less Bench Times :** Have you dealt with IT folks in your company? Mo and mo people, complex processes and communication charter gives you hard time connecting with someone who could get the task done. Things take forever long and cost fortunes with substandard quality. A good bigdata and analytics strategy could reduce the proof of concept time smoothly and substantially. It reduces the burden on IT and gets more high quality, fast and cost effective solutions baked. So, you will waste less time waiting for analysis / insights and more time digging through mo and mo data, and use it for better insights and analyses which was never heard of before.
- **Automation to Backfill Redundant/Mundane Tasks :** How about doing something to the 80% of time that is wasted in data cleaning and preprocessing. There is great deal of automation that could be take part and sky rocket enterprise efficiency. Less manual time spent on data prep and more time is spent on doing analysis that would have substantial ROI compared to mundane data preps and monotonous tasks.
- **Optimize Workforce to Leverage High Talent Cost :** This is an interesting area that I am keeping a close eye on. Businesses already have right talent pools that would solve some pieces of the big data puzzle on data science. Businesses have BI, Modelers and IT people working in harmony in some shape or form. So, a good big data & analytics strategy ensures current workforce is leveraged to it's core in handling enterprise big data and also ensures right number of data scientists are involved with clearer sight to their contribution and their ROI.

Technical

- **Data Continues to Grow Exponentially :** Whether you like it or not, data is increasing. One key technological push is the increasing data and the threat of not being able to use this exploding enterprise data for insights. Having a good strategy puts a pacifier to growing unutilized data concerns.
- **Data is everywhere and in Many Formats :** Besides being able to sieve through data in huge volumes, having a stream of disparate data also poses its threats. Text, voice, video, logs and other emerging formats make it harder to gain insights using traditional tools. So, businesses need to drive their big data toolkit to prep for this exploding data type that is entering corporate data DNA.
- **Alternate, Multiple Synchronous and Asynchronous Data Streams :** Data coming through multiple silos in realtime, creating problem in keeping up with this data in existing data systems. These multiple streams put pressure on businesses to have an effective strategy on handling these sources. With tools out there to handle such situations, it has become important to acquire such capabilities before the competition does.
- **Low Barrier to Entry :** As with any business, low barrier to entry poses one great leverage for businesses to try different technologies and come up with the best strategy. Easy frameworks & paradigms have made available lots of tools, which are relatively easier to deploy. These tools could deliver, a phenomenal computing horsepower.
- **Traditional Solutions Failing to Catch up with New Market Conditions :** Big data has given rise to exploding volume, velocity and variety of data. These 3Vs are difficult to handle and demand cutting edge technologies. New requirements have emerged from changing market dynamics that could not be addressed by old tools, but demands new big data tools. Hence, a big data and analytics strategy to embrace these tools before business goes obsolete.

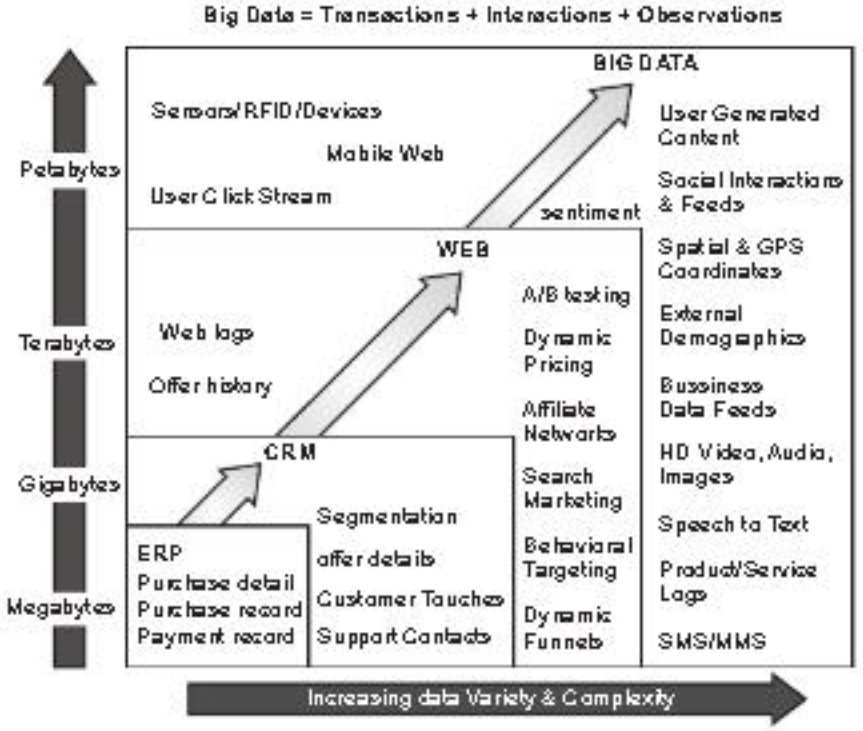


Fig. 1.7

1.3.4 Emerging Big Data Ecosystem and New Approach

New Big data Ecosystem have four major interconnected components as shown in Fig. 1.8.

1. **Data Devices**
   - Data devices adding new data every day. They are generating gigabytes of data and petabytes of Meta data.
   - Consider example of playing online video game through PC, game console or smart phones. Game provider captures all data generated by game as difficulty levels which help them to fine-tune more difficulty of the game. The log data is used to recommend similar type of games of their interest.
   - Another example is of smart phones which is important resource of data. Smart phones are used to store and transmit information about internet usage, SMS usage and real time location. By using this information GPS device provides real time traffic updates and suggest altemative routes to avoid traffic.
2. **Data Collectors**
   - Government, Medical, Retail, Internet, Phone/TV, Financial companies are different entities which collects data from different data device.
3. **Data Aggregators**
   - Data Aggregators make sense of the data collected by data collectors from various IOT Devices.
   - Data Aggregators are combining all data and preparing datasets out of it for further analysis.
4. **Data Users/Buyers**
   - Data users/buyers are getting direct benefit of the aggregated data.

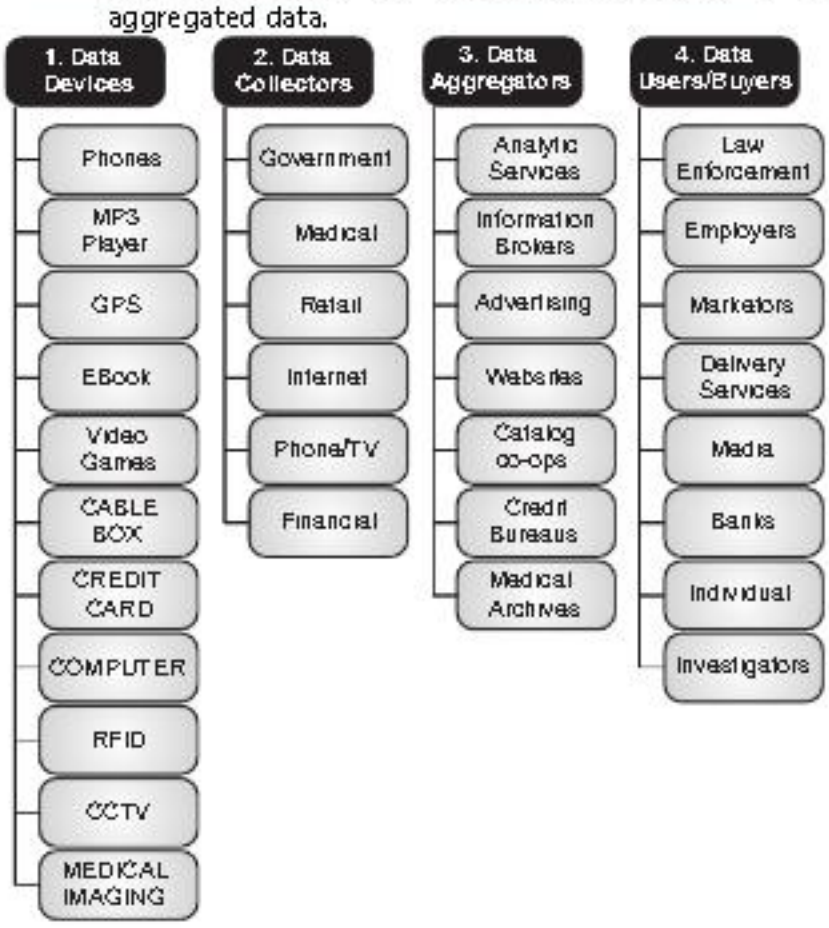


Fig. 1.8 : New big data ecosystem

   - Retail Bank is example of data buyers. They purchase data from data aggregators to know about customer's can able to apply for Home loan or any other type of loans. This data includes demographic information, credit history, loan information and personal bill payment history etc.
   - The unstructured data available on web helps to perform sentiment analysis about a particular product or event.

1.4 OVERVIEW OF DATA ANALYTICS LIFE CYCLE

- The Data Analytics Lifecycle is specifically designed for Big Data problems and data science projects.
- Data Analytics lifecycle has six phases and project work can occur in several phases at once.
- In life cycle, most of the phase's movement is either forward or backward.

Key Stake Holders for a Successful Analytics Project

The various roles and key stakeholders of an analytics project are shown in Fig. 1.9. Each Stake holder plays a important role in a successful analytics project. These 7 stakeholders are as follows

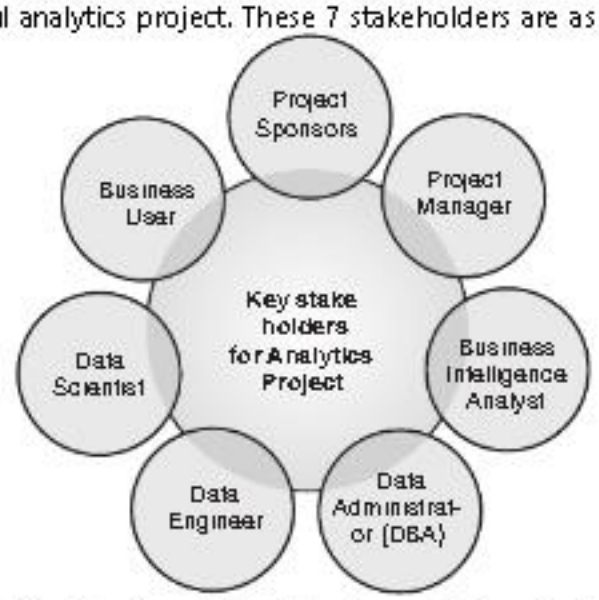


Fig. 1.9 : Key stake holders for analytics project

1. **Business User :**
   - Business user understands the domain area and usually benefits from the results.
   - Business user can consult and advise the project team on the context of the project.
   - Business user can decide the value of the results and how the outputs will be operationalized.
   - Business users are business analyst, line manager or deep subject matter expert in the project domain.
2. **Project Sponsors :**
   - Project Sponsors are responsible for the genesis of the project.
   - They provide the need and requirements for the project.
   - They define the actual business problem.
   - They provide the funding of the project.
   - They set the priorities for the project and clarify the desired outputs.

3. **Project Manager :**
   - The Project manager ensures about key milestones and objectives are achieved in defined time limit.
   - The project manager also ensures about quality of project.
4. **Business Intelligence Analyst :**
   - Business Intelligence Analyst provides business domain expertise based on a deep understanding of the data, Key Performance Indicators (KPIs), key metrics, and business intelligence from a reporting perspective.
   - Business Intelligence Analysts generally create dashboards and reports and have knowledge of the data feeds and sources.
5. **Database Administrator (DBA) :**
   - Database Administrator (DBA) configures the database environment to support the analytics needs of the working team.
   - DBA responsibilities may include
     - Providing access to key databases or tables
     - Ensuring the appropriate security levels related to the data repositories.
6. **Data Engineer :**
   - Data Engineer have deep technical skills to assist with tuning SQL queries for data management and data extraction
   - This person provides support for data ingestion into the analytic sandbox.
   - The DBA sets up and configures the databases to be used; the data engineer executes the actual data extractions and performs substantial data manipulation to facilitate the analytics.
   - The data engineer works closely with the data scientist to help shape data in the right ways for analyses.
7. **Data Scientist :**
   - Data Scientist provides expertise for analytical techniques, data modeling and they apply valid analytical techniques to given business problems.
   - Data Scientist ensures overall analytics objectives are met.
   - Data Scientist designs and executes analytical methods and approaches with the data available to the project.

1.4.1 Phases of Data Analytical Cycle

Data Analytics life cycle have total 6 phases as shown in Fig. 1.10. This iterative nature of the lifecycle closely portray a real project as the project move forward and may return to earlier stages as new information is uncovered and team members learn more about various stages of the project.

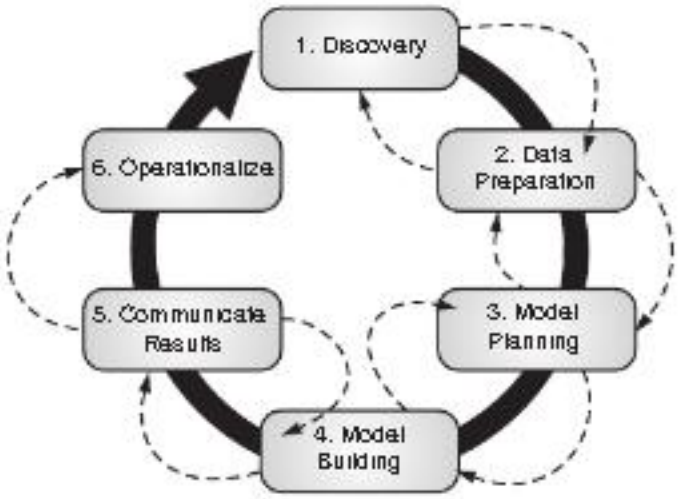


Fig. 1.10: Phases of data analytics life cycle

Overview of these Phases is as Follows :

Phase 1 : Discovery : Do I have enough Information to draft an analytic plan and share for peer review?

- In Discovery phase, the team learns the business domain its relevant history. They learn about similar type of previous implemented projects by the organization or business unit.
- They verify resources available to support the project in terms of people, technology, time, and data.
- Framing the business problem as an analytics challenge is main activity of this phase that can be addressed in subsequent phases and formulating Initial Hypotheses (IHs) to test and begin learning the data.

Phase 2 : Data Preparation : Do I have enough good quality data to start building the model?

- Data Preparation phase requires an analytic sandbox, in which the team can work with data and perform analytics for the duration of the project.
- The team executes Extract, Load, and Transform (ELT) or Extract, Transform and Load (ETL) to get data into the sandbox. The ELT and ETL are called as ETLT.
- To work on data and analyze it, data should be transformed in the ETLT process.
- The team study data in depth and apply various conditions and constraints on it.

Phase 3 : Model Planning : Do I have a good Idea about the type of model to try? Can I refine the analytic plan?

- In Model planning phase, team identify the methods, techniques and workflow which follows the subsequent model building phase.
- The team identifies relationships between variables which helps them to select key variables and the most suitable models.

Phase 4 : Model Building : Is the model robust enough? Have we failed for sure?

- The team develops data sets for testing, training, and production purposes in Model Building phase.

- In addition to this the team builds and executes models based on the work done in the model planning phase.
- The team performs comparison between existing tools and more robust environment as fast hard ware or parallel programming to execute model and work flow.

Phase 5 : Communicate Results :

- In this phase, the team collaborates with key stakeholders to determine success or failure of the project based on the criteria developed in Phase 1.
- The team should identify key findings, quantify the business value, and develop a narrative to summarize and convey findings to stakeholders.

Phase 6 : Opearationalize :

- In Opearationalize phase, the team delivers final reports, briefings, code, and technical documents.
- Team also runs a pilot project to implement the models in a production environment.

1.4.2 Phase 1- Discovery

In Discovery phase, the team learns the business domain its relevant history. In this phase data scientist team need to perform some activities as shown in Fig. 1.11.

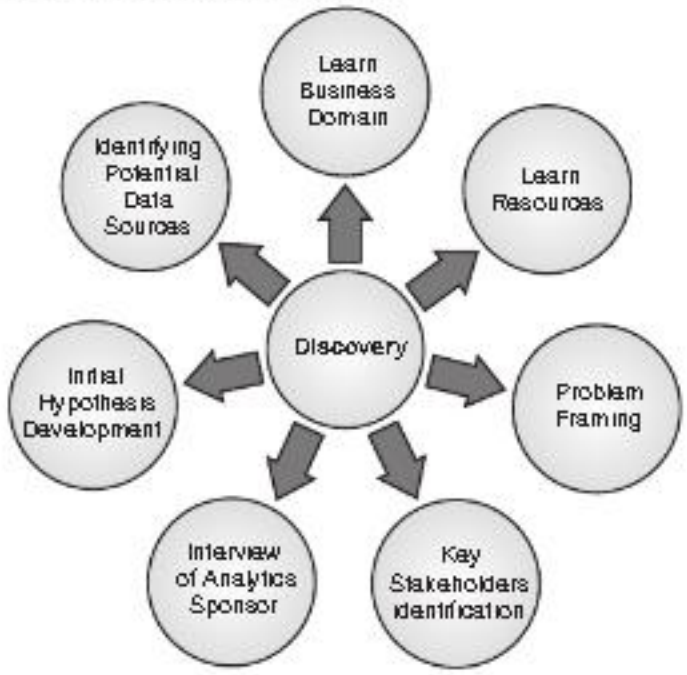


Fig. 1.11 : Phases 1 : Discovery

Learn Business Domain

- Understanding the domain area of the problem is essential activity for data scientist.
- Many data scientist have deep computational and quantitative knowledge that can apply for many cases.
- Some Data scientists may have deep knowledge of the methods, techniques, and ways for applying heuristics to a variety of business and conceptual problems and some of them are good in domain knowledge.
- The team needs to decide how much domain knowledge needs to build model in Phases 3 and 4. The team should have balance of domain knowledge experts and technical experts.

Learn Resources

- Team need to learn about all available resources such as technology, tools, systems, data, and people.
- Team also needs to identify different types of systems needed for later phases to operationalize the models.
- Team also needs to identify gap between existing tools, technologies and skills.
- Team needs to identify sufficient data is available or need to collect additional data, purchase it from outside sources.
- Team need to ensure the project team has the right mix of domain experts, customers, analytic talent, and project management to be effective.
- Team need to evaluate how much time is required if the team has the right breadth and depth of skills.

Problem Framing

- Problem framing is the process of stating the analytics problem to be solved.
- The best practice to write down the problem statement and share it with the key stakeholders.
- Each team member may have own perspective about problem and may have different solutions for the problem.
- Essentially, the team needs to consider the current situation and its main challenges.
- In this process team needs to identify
  - What are main objectives of the project?
  - What needs to be achieved in business terms?
  - What needs to be done to meet the needs?
  - What will be outcome of the project?

Key Stakeholders Identification

- The important step is to identify the key stakeholders and their interests in the project.
- During discussions with stake holders team can identify success criteria, key risks, and stakeholder's benefits.
- When interviewing stakeholders, team can learn about the domain area and any relevant history. For example, expected result of each stake holder and success parameters.
- Team can decide the type of participation expected from stakeholders in the project.
- Team can set clear expectations with the participants and avoid delays for approval or advice about the project.

Interview of Analytics Sponsor

- To interview project sponsors need to team need to be prepared well as they are providing funding to project and they may have different requirement and expectations
  - Prepare questions and review with colleagues.
  - Prefer open-ended questions and avoid asking leading questions.

- Probe for details and pose follow-up questions
- Give sufficient time to person to think.
- Let the sponsors express their ideas and ask clarifying questions, such as "Why? Is that correct? Is this idea on target? Is there anything else?"
- Use active listening techniques; repeat back what he said and try to summarize it.
- Try to avoid expressing the team's opinions, which can introduce bias; instead, focus on listening.
- Be mindful of the body language of the interviewers and stakeholders; use eye contact where appropriate and be attentive, Minimize distractions.
- Document what the team heard, and review it with the sponsors.

Some Common Questions

- What business problem is the team trying to solve?
- What is the desired outcome of the project?
- What data sources are available?
- What industry issues may impact the analysis?
- What timelines need to be considered?
- Who could provide insight into the project?
- Who has final decision-making authority on the project?
- How will the focus and scope of the problem change if the following dimensions change?

  Time : Analyzing 1 year or 10 years' worth of data?

  People : Assess impact of changes in resources on project timeline.

  Risk : Conservative to aggressive

  Resources : None to unlimited (tools, technology, systems)

  Size and Attributes of Data : Including internal and external data sources

Initial Hypothesis Development

- The main task of discovery phase is developing a set of IHs. The ideas can test with data. Team can come up with a few primary hypotheses to test and then be creative about developing several more. Hypothesis testing from a statistical perspective can be done in later phases also.
- The team can compare its answers with the outcome of an experiment or test and can generate additional possible solutions to problems.
- In this process gathering and assessing hypotheses from stakeholders and domain experts as they may have their own perspective on what the problem is, what the solution should be, and how to arrive at a solution.
- These stakeholders would know the domain area well and can offer suggestions on ideas to test as the team formulates hypotheses during this phase. All ideas will also give the team opportunities to expand the project scope into adjacent spaces.

Identifying Potential Data Sources

The Data Scientist team should perform five main activities during discovery phase :

1. **Identify Data Sources :**
   - Team need to make a list of candidate data sources, this data can be used to test the initial hypotheses outlined in this phase.
   - Team need to make an inventory currently available datasets and can be purchased from outside resources.
2. **Capture Aggregate Data Sources :**
   - Current aggregated data sources helps team to preview data and provide in dept understanding of it.
   - It helps team to decide further explorations and complex investigations on the data.
3. **Review the Raw Data :**
   - Raw data can be obtained from preliminary data sources as data feeds.
   - This data provides detail understanding of interdependencies among the data attributes.
   - Team can learn about content of the data, its quality, and its limitations.
4. **Evaluate the Data Structures and Tools Needed :**
   - The data type and structure helps to decide which data analysis tools team can use to analyze the data.
   - It also helps team to decide good technologies for project implementation.
5. **Scope the Sort of Data Infrastructure Needed for this Type of Problem :**
   - Team can decide the kind of infrastructure that's required, such as disk storage and network capacity etc.

1.4.3 Phase 2- Data Preparation

Data preparation phase of the Data Analytics Lifecycle includes the steps to explore, preprocess, and condition data prior to modeling and analysis. The team needs to create a robust environment to explore the data that is separate from a production environment. In this phase data scientist team need to perform some activities as shown in Fig. 1.12

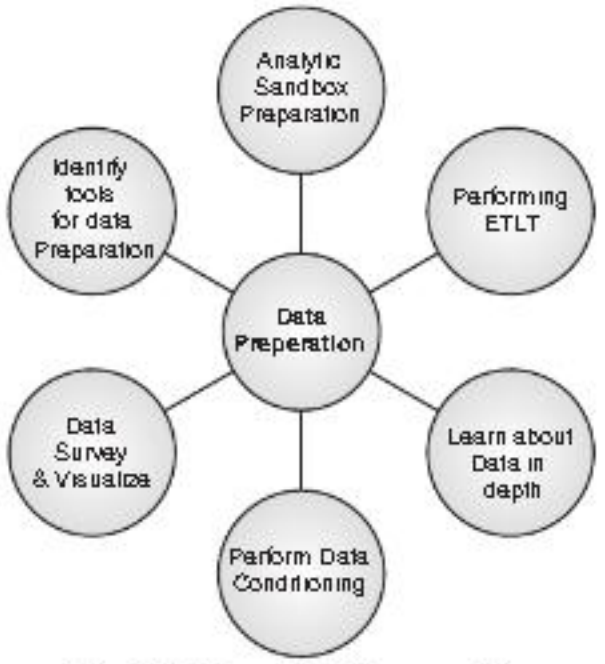


Fig. 1.12: Phases 2: Data preparation

Analytic Sandbox Preparation

- Data preparation requires the team to prepare an analytic sandbox also called as workspace in which the team can explore the data without interfering with live production databases.
- Consider an example If team need financial data to work, that data can be obtained from analytic sandbox rather than interacting with the product ion version of the organization's main database, because that will be tightly controlled and needed for financial reporting.
- To develop analytic sandbox, best practice is to collect all kinds of data so team members can access high volumes and varieties of data for a Big Data analytics project. Data can include summary-level aggregated data, structured data, raw data feeds and unstructured text data from call logs or web logs.
- During this collection of data, the data science team needs to give a justification to IT department how analytical sandbox data is different from traditional IT-controlled warehouses.
- The analytic sandbox enables organizations to work on more challenging data science projects. Organization can move beyond doing traditional data analysis and Business Intelligence to perform more robust and advanced predictive analytics.
- The sandbox size is large as it contains other raw data, unstructured data which is less required for organization but useful for data science project. The sandbox size is again depend on complexity of project.
- A good rule is to plan for the sandbox to be at least 5-10 times the size of the original data sets because partial copies of the data may be created that serve as specific tables or data stores for specific kinds of analysis in the project.
- The analytics sandbox must have ample of bandwidth and reliable network connections to the underlying data sources to enable uninterrupted read and write to perform various transformations.

Perfor ming ETLT

ETLT is combination of ETL (Extract, Transform and Load) and ELT (Extract, Load and Transform) as shown in Fig. 1.13.

Fig. 1.13: ETLT

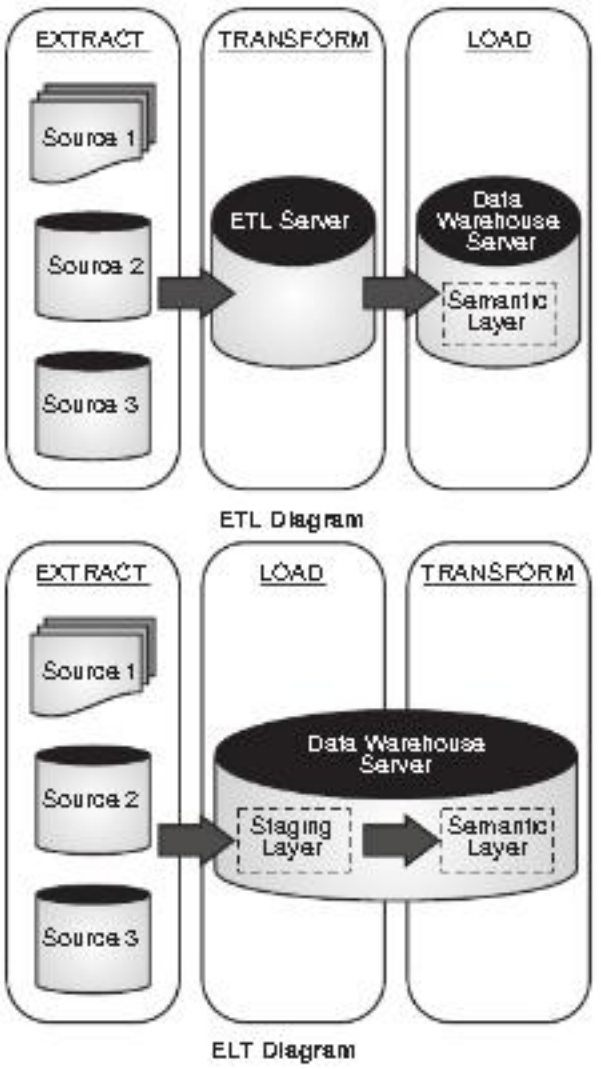


Fig. 1.14: ETL and ELT process

What is ETL?

In ETL, users perform extract, transform, and load processes to extract data from a data store, perform data transformations, and load the data back into the data store as shown in Fig. 1.14.

What is Sandbox Box Approach?

- Sandbox suggests extract, load, and then transform. In sandbox the data is extracted in its raw form and loaded into the datastore, where analysts can choose to transform the data into a new state or leave it in its original, raw condition. There is significant value for raw and including it in sandbox before performing transformation on it.
- Consider example of fraud detection on credit card usage. Outliers in this data population can represent higher-risk transactions which may be fraudulent credit card activity. In case of ETL these outliers may get filtered out. But in case of ELT all data is present in sandbox so can perform analysis of fraud detection.

What is ETLT Process?

- Consider a scenario where the team may want clean data and aggregated data.
- They also need to keep a copy of the original data to compare against or look for hidden patterns that may have existed in the data before the cleaning stage. This process is called as ETLT.

Learn about Data in-Depth

- Learning data in depth is a critical aspect of data preparation. This activity accomplishes following goals,
  - Clarifies the data that the data science team has access to at the start of the project.
  - Highlights gap in existing data sets of organization and team can trigger activity of new data collection from organization.
  - Identifies datasets outside the organization which can obtain, through open APIs, data sharing, or purchasing data to supplement already existing datasets.

Perform Data Conditioning

- The process of cleaning data, normalizing datasets, and performing transformations on the data is called as Data Conditioning.
- It is data pre-processing step in data analysis as it performs many operations on data set before model development.
- Generally this process is performed by IT, data owners, a DBA or a data engineer but, involvement of data scientist is also important in this step because many decisions are made in the data conditioning phase that affects subsequent analysis. The data science team must know decision about which data to keep or which data to transform or discard.
- To retrace the discarded data, team need to add some questions as follows :

  What are the data sources?

  What are the target fields (for example, columns of the tables)?

  How consistent are the contents and files?

Data Survey and Visualize

- To get overview of data, data visualization process carried out. The high-level patterns of data clarify characteristics of it.
- Data visualization used to examine data quality, such as whether the data contains many unexpected values or other indicators of dirty data. Another example is skewness, such as if the majority of the data is heavily shifted toward one value or end of a continuum.

Identify Tools for Data Preparation

Tools used commonly for this activity are

1. **Hadoop**

   Hadoop performs massively parallel processing. It performs web traffic parsing, GPS location analytics, genomic analysis, and combining of massive unstructured data feeds from multiple sources.

2. **Alpine Miner**

   Alpine Miner provides a Graphical User Interface (GUI) for creating analytic work flows. It includes data manipulations and a series of analytic events such as staged data-mining.

3. **Open Refine**

   Open Refine is a free, open source, powerful tool for working with messy data. It has a popular GUI for performing data transformations.

4. **Data Wrangler**

   Data Wrangler is an interactive tools used for data cleaning and transformation. It was developed by Stanford University.

1.4.4 Phase 3- Model Planning

In Model planning phase as shown in Fig. 1.15, the data science team decides candidate models to apply to the data for clustering, classifying, or finding relationships in the data depending on the goal of the project. During this phase the team refers to the hypotheses developed in Phase 1. These hypotheses help them to frame the analytics to execute in Phase 4 and select the right methods to achieve its objectives.

Some of the Activities to Consider in this Phase Include the Following :

- Assess the structure of the dataset; it dictates the tools and analytical techniques for the model building phase. Different tools and approaches are required for different types of data.
- Ensure that the analytical techniques enable the team to meet the business objectives and accept or reject the working hypotheses.
- Determine if the situation needs a single model or a series of techniques as part of a larger analytic workflow.
- Do research about other analyst work on same problem. Need to find out which methods, techniques they used to solve same type of problem.
- Data Exploration and Variable Selection.
- The data exploration carried out in phase 2 is focus mainly on data hygiene and on assessing the quality of the data itself. In this phase, it is carried out to understand the relationships among the variables to inform selection of the variables and methods and to understand the problem domain.
- The main aim to capture the most essential predictors and variables rather than considering every possible variable that people think may influence the outcome. This approach requires iterations and testing to identify the most essential variables for the intended analyses. The team need to test a range of variables to include in the model and then focus on the most important and influential variables.

Model Selection

- In model selection phase, the team can make list of suitable analytical techniques to fulfill end goal of the project. They can observe real world situations and try to map to the current problem for model construction.

- In machine learning and data mining, several such techniques such as classification, association rules, and clustering are available. Team also needs to identify techniques suitable for Big Data for structured data, unstructured data, or a hybrid approach.
- Initially these models can be created by using statistical software package such as R, SAS, or Matlab. As these tools are designed for data mining and machine learning algorithms, may have limitations for Big Data so team need to redesign algorithms as per requirement.

Common Tools for the Model Planning Phase

Many tools are available to assist in this phase. Here are several of the more common ones :

- R provides statistical analysis interface and graphical representation of data. It has many data mining and machine learning algorithm packages with data can be used for data analysis. R can connect with several structured and unstructured databases like SQL, MongoDB etc. and can perform analysis on that data.
- SQL Analysis services used to perform in database analytics of common data mining functions, involved aggregations, and basic predictive models.
- SAS/ACCESS provides integration between SAS and the analytics sandbox via multiple data connectors such as OBDC, JDBC and OLE DB. SAS itself is generally used on file extracts, but with SAS/ACCESS, users can connect to relational databases (such as Oracle or Teradata) and data warehouse appliances (such as Green plum or Aster), files, and enterprise applications (such as SAP and Salesforce.com).

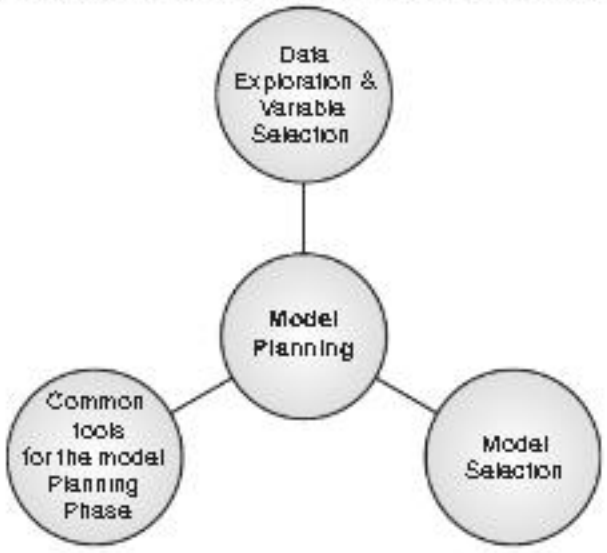


Fig. 1.15: Phases 3: Model planning

1.4.5 Phase 4- Model Building

- In building model phase data scientist team needs to develop training, testing and production datasets. Training data set used to train developed analytical model, test data for testing the model. The training datasets and testing datasets are robust for model.
- Training datasets : Initial Experiments ,Testing datasets : Validation of approach
- In this model team needs to iterate backward and forward to decide the final model. Phase 3 and 4 are short in span but conceptually complex, here the data science team needs to execute the models defined in Phase 3.
- In this phase team needs to consider following questions :
  - Does the model appear valid and accurate on the test data?
  - Does the model output/behavior make sense to the domain experts?
  - Is the model giving answers that make sense in this context?
  - Do the parameter values of the fitted model make sense in the context of the domain?
  - Is the model sufficiently accurate to meet the goal?
  - Does the model avoid intolerable mistakes?
- Common Tools for the Model Building Phase are SAS Enterprise Miner, SPSS Modeler, Matlab ,Alpine Miner , STATISTICA and Mathematica , R and PL/R , Octave, WEKA, Python (scikit-learn, numpy, scipy, pandas, and related data visualization using matplotlib), MADlib.

1.4.6 Phase 5- Communicate Results

- After executing the model, the team needs to compare the outcomes of the modeling to the criteria established for success and failure. The team considers how to convey the findings and outcomes to the various team members and stakeholders.
- The team needs to determine if it succeeded or failed in its objectives. The best practice in this phase is to record all the findings and then select the three most significant ones and share them with the stakeholders. Here the team needs to reflect implications of these findings and measure the business value.
- Team need to consider possible improvements and suggest them in future work or existing process.

1.4.7 Phase 6-Opearationalize

- In this phase the successfully executed model deployed in commercial environment to work on real data. This phase can bring in a new set of team members- usually the engineers responsible for the production environment. This technical group needs to ensure that running the model fits smoothly into the production environment and can integrate into related business processes.
- Operationalzing phase includes creating a mechanism to perform real time monitoring of model accuracy and, if accuracy degrades, finding ways to retrain the model.

- The key output from all stakeholders as described in Table 1.2 which helps to create main deliverables.

Table 1.2

Sr. No.	Stakeholders	Role	Key Output
1.	Business User	Business User tries to determine the benefits and implications of the findings to the business.	Presentation for project sponsors.
2.	Project Sponsor	Project sponsor typically asks questions related to the business impact of the project, the risks and return on investment (ROI).	Presentation for project sponsors.
3.	Project Manager	Project Manager needs to determine if the project was completed on time and within budget.	--
4.	Business Intelligence Analyst	Business Intelligent Analyst needs to know if the reports and dashboards he manages will be impacted and need to change.	Presentation for analysts.
5.	Data Engineer and Database Administrator	Data Engineer and Database Administrator needs to share the code from the analytical project and create technical documents that describe how to implement the code.	Code for technical people, Technical specifications of implementing the code.
6.	Data Scientists	Data Scientists need to share the code and explain the model to their peers, managers, and other stakeholders.	Code for technical people, Technical specifications of implementing the code.

1. Presentation for Project Sponsors :

- This contains high-level takeaways for executive-level stakeholders. It contains a few key messages to aid their decision-making process.
- Focus on clean, easy visuals for the presenter to explain and for the viewer to grasp.

2. Presentation for Analysts :

- This describes changes to business processes and reports.
- Data scientists reading this presentation are comfortable with technical graphs such as histograms etc.

3. Code for Technical People: such as engineers and technical people in production environment

4. Technical Specifications : For implementing the code

1.5 CASE STUDY : GINA-GLOBAL INNOVATION NETWORK AND ANALYSIS

- In 2012 EMC's new director wanted to improve the company's engagement of employees across the Global Centers of Excellence (GCE) to drive innovation, research, and university partnerships
- This project was created to accomplish
  - Store formal and informal data.
  - Track research from global technologists.
  - Mine the data for patterns and insights to improve the team's operations and strategy.

Phase 1 : Discovery

- Team members and roles,
  - Business user, project sponsor, project manager – Vice President from Office of CTO
  - BI analyst – Person from IT
  - Data engineer and DBA – People from IT
  - Data scientist – Distinguished engineer
- The data fell into two categories,
  - Five years of idea submissions from internal innovation contests
  - Minutes and notes representing innovation and research activity from around the world
- Hypotheses grouped into two categories
  1. Descriptive analytics of what is happening to spark further creativity, collaboration, and asset generation
  2. Predictive analytics to advise executive management of where it should be investing in the future
- **The 10 Main IHs that the GINA Team Developed Were as Follows :**
  1. Innovation activity in different geographic regions can be mapped to corporate strategic directions.
  2. The length of time it takes to deliver ideas decreases when global knowledge transfer occurs as part of the idea delivery process.
  3. Innovators who participate in global knowledge transfer deliver ideas more quickly than those who do not.
  4. An idea submission can be analyzed and evaluated for the likelihood of receiving funding.
  5. Knowledge discovery and growth for a particular topic can be measured and compared across geographic regions.
  6. Knowledge transfer activity can identify research-specific boundary spanners in disparate regions.
  7. Strategic corporate themes can be mapped to geographic regions.
  8. Frequent knowledge expansion and transfer events reduce the time it takes to generate a corporate asset from an idea.
  9. Lineage maps can reveal when knowledge expansion and transfer did not (or has not) resulted in a corporate asset.
  10. Emerging research topics can be classified and mapped to specific ideators, innovators, boundary spanners, and assets.

Phase 2 : Data Preparation

- Set up an analytics sandbox.
- Discovered that certain data needed conditioning and normalization and that missing datasets were critical.
- Team recognized that poor quality data could impact subsequent steps.

- They discovered many names were misspelled and problems with extra spaces.
- These seemingly small problems had to be addressed.

Phase 3 : Model Planning

- The study included the following considerations.
  - Identify the right milestones to achieve the goals.
  - Trace how people move ideas from each milestone towards the goal.
  - Tract ideas that die and others that reach the goal.
  - Compare times and outcomes using a few different methods.

Phase 4 : Model Building

- Several analytic method were employed
  - NLP on textual descriptions.
  - Social network analysis using R and Rstudio.
  - Developed social graphs and visualizations.

Social Graph of Data Submitters and Finalists

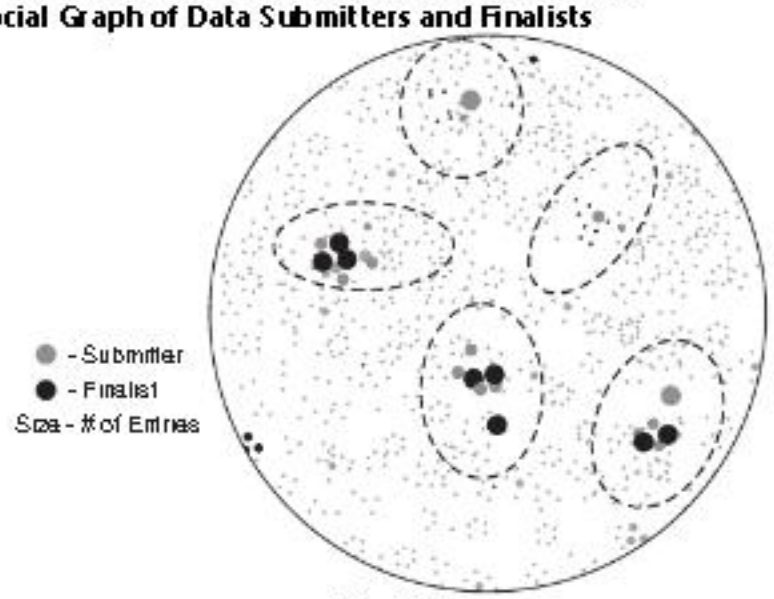


Fig. 1.16

Social Graph of Top Innovation Influencers

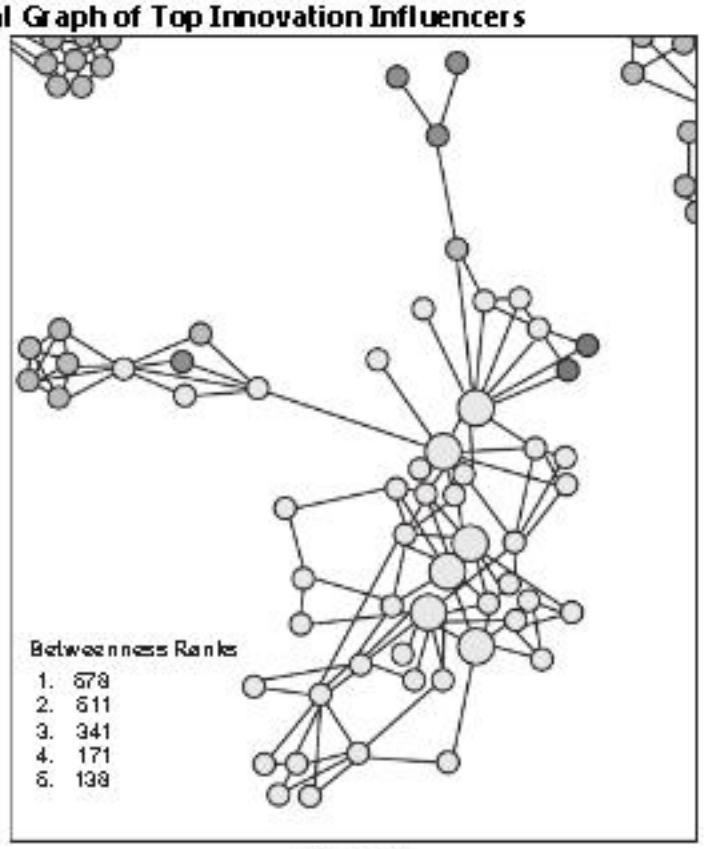


Fig. 1.17

Phase 5 : Communicate Results

- Study was successful in identifying hidden innovators
  - Found high density of innovators in Cork, Ireland
- The CTO office launched longitudinal studies

Phase 6 : Operationalize

- Deployment was not really discussed
- Key findings
  - Need more data in future
  - Some data were sensitive
  - A parallel initiative needs to be created to improve basic BI activities
  - A mechanism is needed to continually reevaluate the model after deployment

Table 1.3

Components of Analytic Plan	GINA Case Study
Discovery Business Problem Framed	Tracking global knowledge growth, ensuring effective knowledge transfer, and quickly converting it into corporate assets. Executing on these three elements should accelerate innovation.
Initial Hypotheses	An increase in geographical knowledge transfer improves the speed of idea delivery.
Data	Five years of innovation idea submissions and history; six months of textual notes from global innovation and research activities.
Model Planning Analytic Technique	Social network analysis, social graphs, clustering and regression analysis
Result and Key Findings	1. Identified hidden, high-value innovators and found ways to share their knowledge 2. Informed investment decisions in university research projects 3. Created tools to help submitters improve ideas with idea recommender systems

EXERCISE

1. What are the three characteristics of Big Data?
2. What is an analytic sandbox, and why is it important?
3. Explain the differences between BI and Data Science.
4. Discuss about current analytical architecture and its implications for data scientist?
5. Discuss about new Big data Ecosystem with all components?
6. In which phase would the team expect to invest most of the project time and in which phase expect less time? Why?
7. Discuss case study of GINA for data analytics life cycle.
8. What kinds of tools would be used in the following phases
   a. Data Preparation
   b. Model building

UNIT II

BASIC DATA ANALYTIC METHODS

2.1 STATISTICAL METHODS FOR EVALUATION

- Statistical methods have importance in data analytics.
- Statistical methods are mathematical formulas, models, and techniques that are used in statistical analysis of raw research data.
- Examples of such statistical methods are Mean, Standard Deviation, Regression, Sample size determination, Hypothesis testing etc.
- The application of statistical methods extracts information from research data and provides different ways to assess the robustness of research outputs.
- These methods are used throughout Data Analysis life cycle such as initial data exploration and data preparation, model building, evaluation of the final models.
- Statistical models are also used for assessment of how the new models improve the situation when deployed in the field.
- Statistic help to solve following three problems of each phase as follows

1. Model Building and Planning

- What are the best input variables for the model?
- Can the model predict the outcome given the input?

2. Model Evaluation

- Is the model accurate?
- Does the model perform better than an obvious guess?
- Does the model perform better than another candidate model?

3. Model Deployment

- Is the prediction sound?
- Does the model have the desired effect (such as reducing the cost)?

2.2 HYPOTHESIS TESTING

1. Hypothesis

- A statistical hypothesis also called as confirmatory data analysis is test on the basis of process observation that is modeled via a set of random variables.
- Hypothesis is nothing but some formal questions that analyst want to resolve using available data.

Hypothesis Statement is as Follows :

If (do this to an independent variable) then (this will happen to the dependent variable)

Example : If I give supplement medicine to patient then he will cure in less time.

In Hypothesis Statement :

- If then statement is used.
- Includes both independent and dependent variables.
- It can be test by experiment, survey or other techniques.
- It is based on information/research.
- Can take appropriate decision.

Properties of Hypothesis :

- Hypothesis should be precise and clear.
- It should be enough capable for testing.
- It should represent variable relationship if hypothesis is relational type.
- It should be limited in scope and be very specific.
- It should be written in simple words, understandable to all team members.
- It should be consistent with all known facts.
- It should be tested in reasonable time.

2. Hypothesis Testing

- Hypothesis testing in statistics is way to test the results of a survey or experiment check meaningful results.

$$z = \frac{\hat{p} - p}{\sqrt{p\,q\,/n}}$$

- When comparing populations, example testing or evaluating the difference of the means from two samples of data, a common method used to assess the difference or the significance of the difference is called as hypothesis testing. Basically in hypothesis testing, we form an assertion and test it with data.

Steps for Hypothesis testing

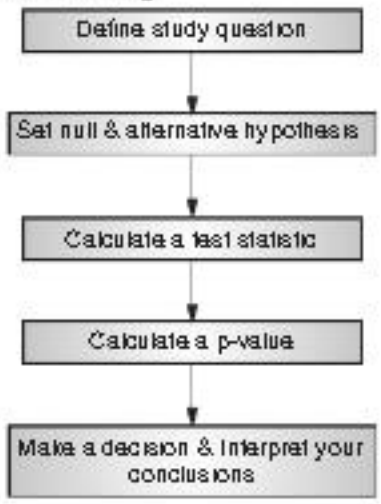


Fig. 2.1

Null Hypothesis and Alternative Hypothesis

- **Null Hypothesis H_0 :** While performing hypothesis testing, common assumption is that there is no difference between two samples. This assumption is used as the default position for building the test or conducting a scientific experiment.
  Example : DNA is shaped like a double helix.
- **Alternative Hypothesis H_1 :** If there is difference between two samples then it is called as Alternative Hypothesis. In hypothesis testing process it is important to state the null hypothesis and alternative hypothesis. A hypothesis test leads to either rejecting the null hypothesis in favor of the alternative or not rejecting the null hypothesis.

  Example : Identify the effect of drug M compared to drug N on animals.

  - **H_0 :** Drug M and drug N have the same effect on animals.
  - **H_1 :** Drug N has a greater effect than drug M on animals.

  or

  - **H_1 :** Drug M has a greater effect than drug N on animals.

Examples in data analysis life cycle

1. **To Test Accuracy of Forecast :**
   - **H_0 :** Model P does not predict better than the existing model.
   - **H_1 :** Model X predicts better than the existing model.
2. **To Test Recommendation Engine**
   - **H_0 :** Algorithm J does not produce better recommendations than the current algorithm being used.
   - **H_1 :** Algorithm J produces better recommendations than the current algorithm being used.

2.3 DIFFERENCE OF MEANS

- The mean difference, or difference in means, measures the absolute difference between the mean values in two different groups or two different populations.
- **Example :** In clinical trials analyst can identify how much difference there is between the averages of the experimental group and control groups.

Consider Two Hypotheses

H_0 : $\mu 1 = \mu 2$

H_1 : $\mu 1 \neq \mu 2$

$\mu 1$ = mean of group 1, $\mu 2$= mean of group 2

- This approach is to compare sample means X_1 and X_2, corresponding to each group.
- If the values of X_1 and X_2 are approximately equal to each other, the distributions of X_1 and X_2 overlap substantially and the null hypothesis is supported.
- A large observed difference between the sample means indicates that the null hypothesis should be rejected as shown in Fig. 2.2.
- Difference in mean can be tested by using Student-t test or Welch's t-test.

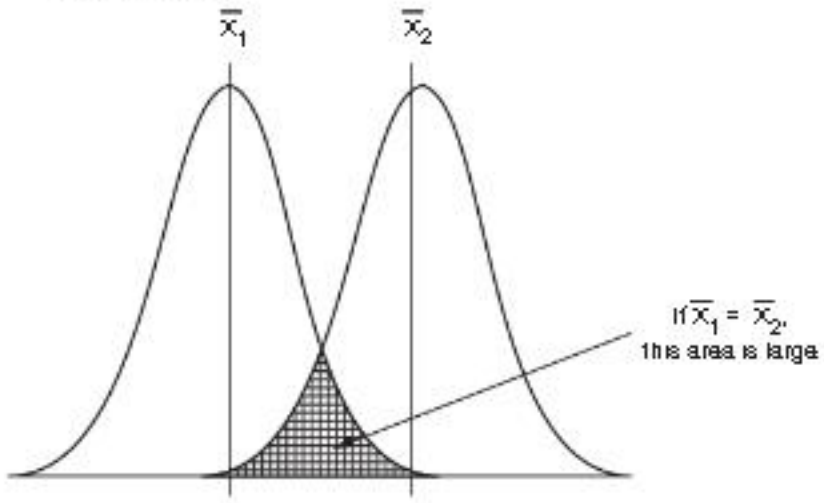


Fig. 2.2 : Overlap of the two distributions

Student-t Test :

- It is statistical Hypothesis test introduced in 1908 by William Sealy Gosset, a chemist working for the Guinness brewery in Dublin, Ireland. "Student" was his pen name.
- A *t*-test is used when the test statistic would follow a normal distribution if the value of a scaling term in the test statistic were known.

T Score

The t score is a ratio between the difference between two groups and the difference within the groups. The larger the t-score, the more difference there is between groups means groups are different and smaller the t-core, the more similarity there is between groups means the groups are similar.

Equal or Unequal Sample Sizes, Equal Variance

- It assumes that distributions of the two groups have equal but unknown variances.
- Suppose X_1 and X_2 samples are randomly and independently selected from two groups' grp1 and grp2 respectively.
- If each group is normally distributed with the same mean ($\mu 1 = \mu 2$) and with the same variance, then T (the t-statistic), given in Equation, follows a t - distribution with $n_1 + n_2 - 2$ degrees of freedom (df).

$$T = \frac{\overline{X}_1 - \overline{X}_2}{S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

$$S_p^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}$$

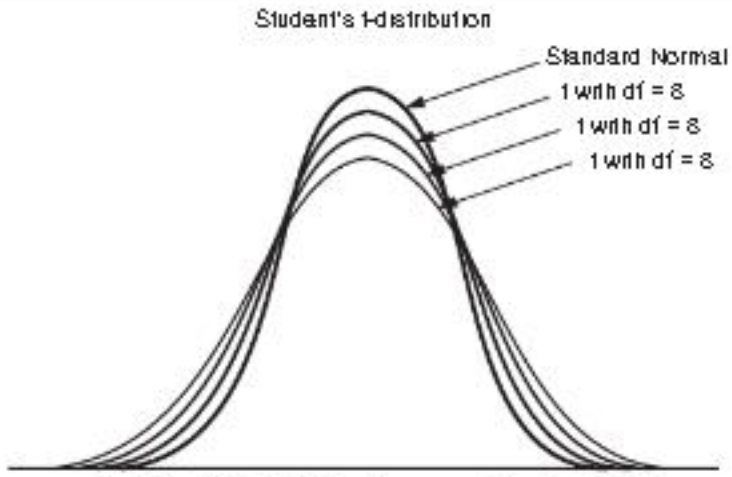


Fig. 2.3 : The t-distribution, used for the t-test

Source : Carnegie Mellon

- T-distribution is similar to the normal distribution. If degrees of freedom (df) approaches 30 or more, the t-distribution is identical to the normal distribution.
- As the numerator of T is the difference of the sample means, if the observed value of T is far enough from zero such that the probability of observing such a value of T is unlikely, one would reject the null hypothesis that the population means are equal.
- Thus, for a small probability, say a= 0.05, T* is determined such that P(|T|≥ T*) = 0.05. After the samples are collected and the observed value of T is calculated according to above Equation, the null hypothesis (μ1 = μ2) is rejected if |T|≥ T*
- In hypothesis testing α (small probability) is known as the **Significance Level** of the test.
- The significance level of the test is the probability of rejecting the null hypothesis, when the null hypothesis is actually TRUE.
- In other words, for α = 0.05, if the means from the two populations are truly equal, then in repeated random sampling, the observed magnitude of T would only exceed T* 5% of the time.
- In following R code n : 15 and 25 random samples selected from x and y respectively. Standard deviation SD=5 and mean is 100 and 105 respectively.

  R code for student T test

1. Generate 2 Groups using Normal Distributions as Follows

```
x <- morm(15, mean=100, sd=5) # normal distribution centered at 100
y <- rnorm(25, mean=105, sd=5) # normal distribution centered at 105
x
y
> x
[1] 100.03278 104.07315 103.42691 97.63273 99.19685 103.65574 113.17236 97.77431
[9] 97.49657 92.24864 97.54787 106.90654 103.81307 102.75919 101.91236
> y
[1] 108.76320 104.96821 104.96149 113.32078 102.96498 95.78091 105.45557 104.55679
[9] 99.21011 106.37135 105.58392 105.13484 106.42137 104.59500 96.13259 101.77601
[17] 115.42279 108.71601 102.44341 102.02338 97.86939 103.45484 100.12086 109.50642
[25] 107.16043
```

2. Run Students t Test

```
t.test(x, y, var.equal=TRUE) # run the Student's t-test
```

Two Sample t-Test

data : x and y

t = -1.9562, df = 38, p-value = 0.05783

alternative hypothesis : true difference in means is not equal to 0

95 percent confidence interval :

-6.2375385 0.1069102

sample estimates :

mean of x mean of y

101.4433 104.5086

T test shows t= -1.9562. Negative sign means x samples are less than y samples.

By using qt() function in R we can obtain T = 2.024394

by using significant level value 0.05

```
# obtain t value for a two-sided test at a 0.05 significance level
qt(p=0.05/2, df=38, lower.tail= FALSE)
```

2.024394

- The null hypothesis is not rejected as T statistic is less than the T value corresponding to the 0.05 significance level (|-1.9562|<2.024394).
- As alternative hypothesis μ1 ≠ μ2 so μ1 ≥ μ2 and μ1 ≤ μ2 both need to consider so this is called as two sided hypothesis test. So we used p=0.05/2 in qt() to obtain appropriate t value.
- P Value : A P-value is the probability that the results from your sample data occurred by chance. Every t-value has a p-value to go with it. Here p value is 0.05783 which is sum of P(T≤-1.9562) and P(T≥1.9562).
- If we use significant value 0.10 instead of 0.05 then null hypothesis may get rejected.
- The t-statistic for the area under the tail of a t-distribution. The –t and t are the observed values of the t-statistic.
- In the R output, t = 1.95628. The left shaded area corresponds to the P(T≤-1.9562), and the right shaded area corresponds to the P(T≥1.9562).

Welch's t-Test

- When equal population variance is not justified then Welch's t –test can be used for difference of means.
- Formula for Welch's test or unequal variance t test is as follows

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{S_1^2}{N_1} + \frac{S_2^2}{N_2}}}$$

Here $\bar{X}$ = Sample Mean

S^2 = Sample Variance

N = Sample Size

For Welch's t Test Consider Example as Follows

R code for **Welch's t test**

1. Generate 2 groups using normal distributions as follows

```
x <- rnorm(15, mean=100, sd=5) # normal distribution centered at 100
y <- morm(25, mean=105, sd=5) # normal distribution centered at 105
x
y
```

```
> x
100.03278 104.07315 103.42691 97.63273 99.19685 103.65574 113.17236 97.77431
97.49657 92.24864 97.54787 106.90654 103.81307 102.75919 101.91236
> y
108.76320 104.96821 104.96149 113.32078 102.96498 95.78091 105.45557 104.55679
99.21011 106.37135 105.58392 105.13484 106.42137 104.59500 96.13259 101.77601
115.42279 108.71601 102.44341 102.02338 97.86939 103.45484 100.12086 109.50642
107.16043
```

2. Run Welch's t Test

```
> t.test(x, y, var.equal=FALSE) # run the Welch's t-test
Welch Two Sample t-test
data : x and y
t = -2.67, df = 23.732, p-value = 0.01347
altemative hypothesis : true difference in means is not equal to 0
95 percent confidence interval :
-7.919728 -1.011709
sample estimates :
mean of x mean of y
 100.4713  104.9370
```

- In this particular example now p value 0.01347 is greater than student's t tests p values to null hypothesis rejection at significant level 0.10 is not possible.
- **Confidence Interval :** The confidence interval provides an interval estimate of the difference of the population means. In both examples confidence interval is 95 %, it means 95 % time CI contain true value of population parameter and 5% fail to contain true values.

2.4 WILCOX ON RANK–SUM TEST

What is Non Parametric Test ?

- The test based on ranks or signs are called as Non parametric tests.
- Data is in order and ranked.
- Analysis is performed on rank not on data.

When Non Parametric Tests are Used?

- When data is ordinal type
- When data is not following any shape or distribution
- When data is not meeting requirement of parametric tests.
- When data plot is appeared much skewed.
- When data samples are too small.
- When potential influential outliers are in data set.

What is Wilcox on Rank-Sum Test?

- It is a nonparametric hypothesis test that checks whether two populations are identically distributed.
- It is used to *if* two independent samples came from the same or equal populations.
- It inferences about whole population.
- Here assuming that the two populations are identically distributed.
- One would expect that the ordering of any sampled observations would be evenly intermixed among themselves it means large number of observations from one population grouped together.
- Assume A and B are two populations with independently random samples of size n_1 and n_2 respectively. So the total number of observations is then $N = n_1 + n_2$

Step 1 : Rank Observations

- The first step of the Wilcoxon test is to rank the set of observations from the two groups as if they came from one large group.
- The smallest observation receives a rank of 1, the second smallest observation receives a rank of 2, and so on with the largest observation being assigned the rank of N.

Step 2 : Apply Wilcox in Rank Sum Test

- The sum of ranks for smaller samples n_1 and n_2.

- The smaller of the two sums *TA* is used to compute the test statistic from :

  Wilcoxon for $n_1 \geq 10$ and $n_2 \geq 10$:

Test statistic :

$$Z = \frac{T_A - \frac{n_1(n_1 + n_2 + 1)}{2}}{\sqrt{\frac{n_1 n_2 (n_1 + n_2 + 1)}{12}}}$$

Example

Sample 1 : 11, 22, 14, 21

Sample 2 : 20, 9, 12, 10

Test

H0 : $\mu 1 = \mu 2$ vs HA : $\mu 1 \neq \mu 2$

Step 1 : Rank Sample Values

Rank	Value	Sample
1	9	2
2	10	2
3	11	1
4	12	2
5	14	1
6	20	2
7	21	1
8	22	1

Step 2 : Compute the Sum of Ranks for each Sample

Here RS(1) =3 + 5 + 7 + 8 = 23 and RS(2) = 1 + 2 + 4 + 6 = 13.

R Code

```
> samp1 = c(11, 22, 14, 21)
> samp2 = c(20, 9, 12, 10)
> wilcox.test(samp1, samp2)
    Wilcoxon rank sum test
data : samp1 and samp2
W = 13, p-value = 0.2
Alternative Hypothesis : True location shift is not equal to 0
```

2.5 TYPE 1 AND TYPE 2 ERRORS

- A hypothesis results in two type of errors based on acceptance or rejection of Null Hypothesis.
- These two errors are known as type I and type II errors.
- A **Type I Error** is the rejection of the null hypothesis when the null hypothesis is TRUE. The probability of the type I error is denoted by the Greek letter α.

α = P(reject H_0 | H_0 is true)

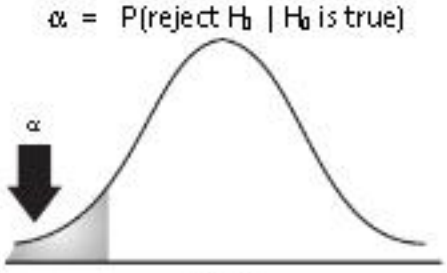


Fig. 2.4

- A **Type II Error** is the acceptance of a null hypothesis when the null hypothesis is FALSE. The probability of the type II error is denoted by the Greek letter β.

β = P(fail to reject the H_0 | H_A is true)

	H_0 True	H_0 False
H_0 Accepted	Correct Outcome	Type II Error
	Probability = 1- α	Probability= β
H_0 Rejected	Type I Error	Correct Outcome
	Probability= β	Probability = 1- α

- In other words Type I error means rejection of hypothesis which is accepted and Type II error means accepting hypothesis which is rejected.

2.6 POWER AND SAMPLE SIZE

- **Definition :** The probability that you reject the null hypothesis given that the alternative hypothesis is true. This is what we want to happen.
- Power = P(reject H_0 | H_A is true) = 1 - β, where β is the probability of a type II error.
- Power describes the test's ability to minimize type-II errors (false negatives).
- Power of test increases as sample size increases power is used to determine the necessary sample size.
- In the difference of means, the power of a hypothesis test depends on the true difference of the population means.
- A fixed significance level, a larger sample size is required to detect a smaller difference in the means.
- In general, the magnitude of the difference is known as the effect size. The sample size becomes larger, it is easier to detect a given effect size.

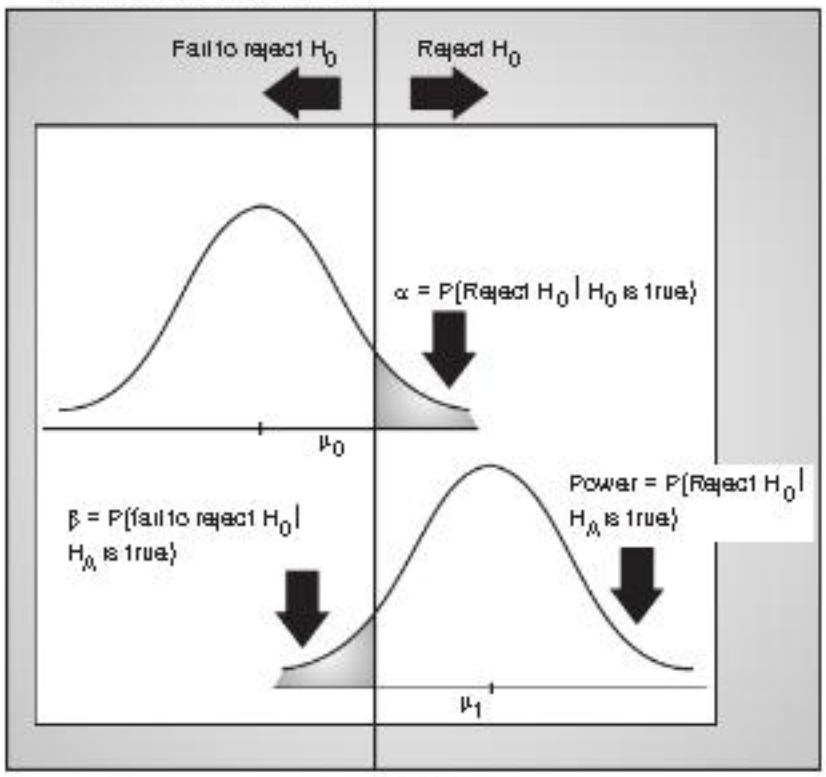


Fig. 2.5

2.7 ANOVA

Consider an example of testing the impact of nutrition and exercise on 80 candidates between age 25 and 50.

The candidates are randomly split into six groups, each assigned with a different weight loss strategy, and the goal is to determine which strategy is the most effective.

Group 1 only eats junk food.

Group 2 only eats healthy food.

Group 3 eats junk food and does cardia exercise every other day.

Group 4 eats healthy food and does cardia exercise every other day.

Group 5 eats junk food and does both cardia and strength training every other day.

Group 6 eats healthy food and does both cardia and strength training every other day.

Multiple t-tests could be applied to each pair of weight loss strategies.

Here the weight loss of Group 1 is compared with the weight loss of Group 2, 3, 4, 5, or 6 and the weight loss of Group 2 is compared with that of the next 4 groups. So total of 15 t-tests would be performed.

Multiple test may not we performed because

- The number of t-tests increases as the number of group's increases; analysis using the multiple t-tests becomes cognitively more difficult.
- By doing a greater number of analyses, the probability of committing at least one type I error somewhere in the analysis greatly increases.

What is ANVOA?

- Analysis of Variance (ANOVA) is used when need to compare more groups.
- This technique is extremely useful for researchers in the field of economics, education, psychology, and business/industry and in several other disciplines.
- ANOVA is used to test the difference among different groups for data for homogeneity.
- There may be variation between samples and also within sample items. ANOVA consists in splitting the variance for analytical purpose.
- The null hypothesis of ANOVA is that all the population means are equal.
- The alternative hypothesis is that at least one pair of the population means is not equal.

  H_0 : $\mu1=\mu2=.....=\mu n$ all population means are same

  H_1 : $\mu i \neq \mu j$ for at least one pair of i,j all population means are not same.

- ANOVA measures two sources of variation in the data and compares their relative sizes.
  1. **Variation BETWEEN Groups :** For each data value look at the difference between its group mean and the overall mean.
  2. **Variation WITHIN Groups :** For each data value we look at the difference between that value and the mean of its group.

  So

  Fc = between sample variance /within sample variance
- Between sample variance is large when the effect of all the treatments are different in such case Fc is large so chances of rejection of null hypothesis.
- The goal is to test whether the clusters formed by each population are more tightly grouped than the spread across all the populations.
- The total number of samples N is randomly split into the k groups. The number of samples i-th group is denoted as n1, and the mean of the group is X1 where i [l,k]. The mean of all the samples is denoted as X0.
- **ONE WAY ANOVA :** The between-groups mean sum of squares, S_B^2 is an estimate of the between-groups variance. It measures how the population means vary with respect to the grand mean, or the mean spread across all the populations.

$$S_B^2 = \frac{1}{k-1}\sum_{i=1}^{t} n_i \cdot (\bar{X}_i - \bar{X}_0)^2$$

- The within-group mean sum of squares, S_w^2, is an estimate of the within-group variance. It quantifies the spread of values within groups.

$$S_w^2 = \frac{1}{n-k}\sum_{i=1}^{t}\sum_{j=1}^{p} (X_0 - \bar{X}_j)^2$$

- The F-test statistic is defined as the ratio of the between-groups mean sum of squares and the within group mean sum of squares.

$$F_c = S_B^2 S_W^2$$

Example :

Customer visits shop gets one of two offers or not get any promotional offer at all. Here we can use ANOVA . The goal is to see if making the promotional offers makes a difference.

H_0 : no promotional offer makes difference.

```
o<- sample(c("offerx", "offery", "nopromo"), size=500,
replace=T)
# Simulated 500 observations of purchase sizes on the 3
offer options
```

```
purchase_size <- ifelse(o=="offerx", rnorm(500, mean=80,
sd=30),
    ifelse(o=="offery", rnorm(500, mean=85, sd=30),
    rnorm(500, mean=40, sd=30)))
# create a data frame of offer option and purchase size
offer_test <-
data.frame(offer=as.factor(o),purchase_amt =purchase_size)
```

After applying summary on offertest we can see that offerx=170, offery=161 and nopromo=169 offers have been made.

The aov ( ) function performs the ANOVA on purchase size and offer options.

```
Final_model<- aov(purchase_amt ~ o, data=offer_test)
```

After applying summary on Final_model we can see

The degree of freedom for offers is 2 (for k-1) and the degree of freedom for residuals is 497 (for n-k).

$$S_b^2 = 112611$$

$$S_w^2 = 862$$

$$\text{P-value} = < 2e\text{-}16$$

The F-test statistic is much greater than 1 with a p-value much less than 1. Thus, the null hypothesis that the means are equal should be rejected.

2.8 ADVANCED ANALYTICAL THEORY AND METHODS

- Unsupervised learning is Hthe training of an Artificial Intelligence (AI) algorithm using information that is neither classified nor labeled and allowing the algorithm to act on that information without supervisor or guidance.
- Unsupervised learning form group of unsorted information based on similarities or differences.

Unsupervised Learning Problems are Divided as Follows :

- **Clustering :** A clustering problem is used to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.
- **Association :** An association rule learning problem is used to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

What is Cluster Analysis?

- Cluster Hanalysis divides data Hinto Hgroups based Hon information found in the data that describes the objects and their relationships.
- Here main goal is that the objects within groups be similar to one another and different from the objects in other group.
- The greater the similarity within a cluster and the greater the difference between clusters, the better or more distinct the clustering.

Different Types of Clusterings

Clustering means entire collection of clusters. There are various types of clusters as follows

1. Hierarchical Clustering

- It is set of nested clusters that can organize as tree.
- Every node i.e. cluster in a tree except the leaf node is the union of its children i.e. subclusters. The leaves are always singleton clusters with single object.
- The root if the tree is cluster containing all the objects as shown in Fig. 2.6

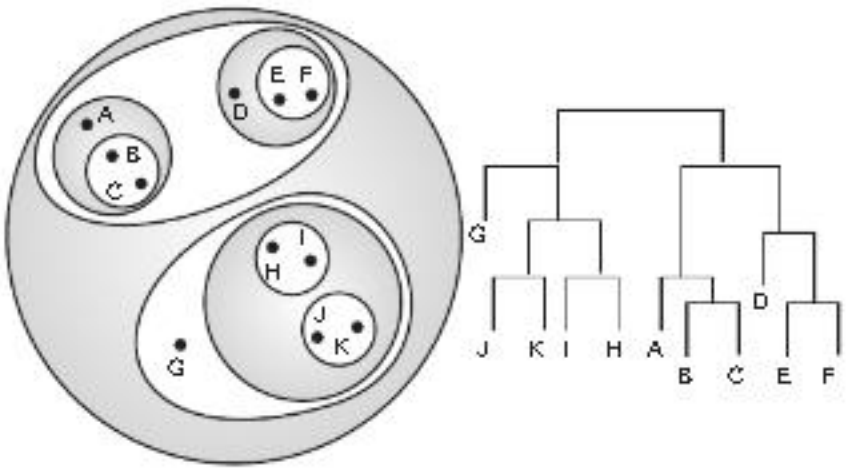


Fig. 2.6 : Hierarchical clustering

2. Partitional Clustering

It is a division of the set of data objects into non-overlapping subsets i.e. clusters such that each data object is in exactly one subset as shown in Fig. 2.7

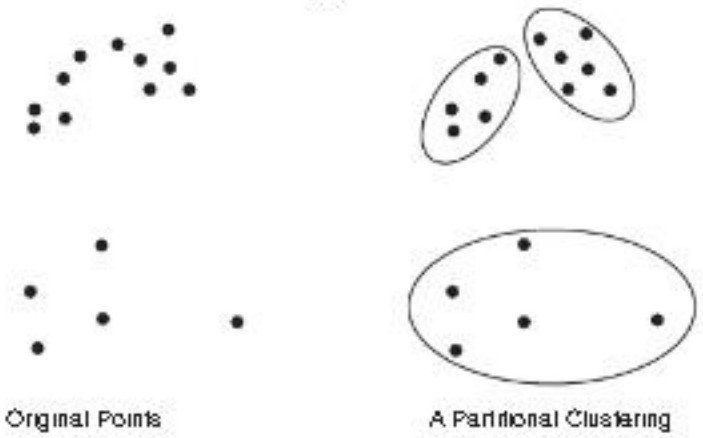


Fig. 2.7 : Partitional clustering

3. Exclusive Clustering

In Exclusive clustering each object belongs to single cluster, not to several as shown in Fig. 2.8.

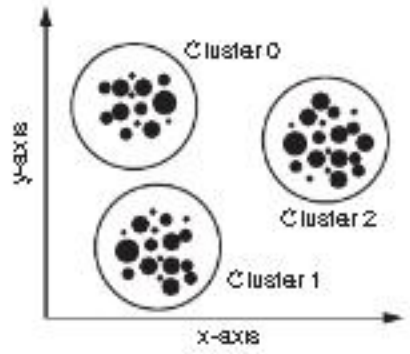


Fig. 2.8 : Exclusive clustering

4. Overlapping Clustering

In Overlapping Clustering objects are simultaneously belongs to different clusters with different degrees of association among each other as shown in Fig. 2.9.

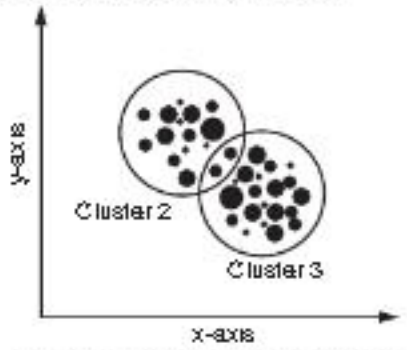


Fig. 2.9 : Overlapping Clustering

5. Fuzzy Clustering

In Fuzzy clustering ,every object belongs to every duster with a membership weight that is between 0 to 1 where 0 means absolutely doesn't belongs and 1 is absolutely belongs .

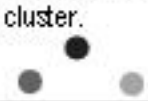
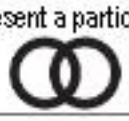
6. Complete Clustering

Complete clustering assigns every object to a cluster.

7. Partial Clustering

Partial Clustering doesn't assign every object to cluster. Such objects Hare may be Houtliers, Hnoise or Huninteresting background.

Different Types of Clusters :

Sr. No.	Type	Details
1.	Well Separated	A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster. 
2.	Prototype Based/Center Based	A cluster is a set of objects such that an object in a cluster is closer (more similar) to the "center" of a cluster, than to the center of any other cluster The center of a cluster is often a centroid, the average of all the points in the cluster, or a medoid, the most "representative" point of a cluster. 
3	Density Based	A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density. Used when the clusters are irregular or intertwined, and when noise and outliers are present. 
4	Shared Property or Conceptual Clusters	Finds clusters that share some common property or represent a particular concept. 
5	Contiguity Based	Each point is closer to at least one point in its cluster than to any point in another cluster. 

2.9 K MEANS

- K Means is a prototype based partitional clustering technique that attempts to find user defined clusters (K).
- It defines a prototype in terms of centroid which is mean of group of points and applied to objects in a continuous n-dimensional space.

a. K Means Method–Overview

Basic K-means algorithm

1. Select K points as initial centroids.
2. Repeat
   a. Form K clusters by assigning each point to its closest centroid.
   b. Recompute the centroid of each cluster.
3. Until Centroids do not change.

What is Centroid and Medoid?

- Given a cluster Km of N points { tm1,tm2, tmk} , the centroid or middle of the cluster computed as,

  Centroid = Cm = $\sum$ tmi / N is considered as the representative of the cluster (there may not be any corresponding object)
- Some algorithms use as representative a centrally located object called Medoid

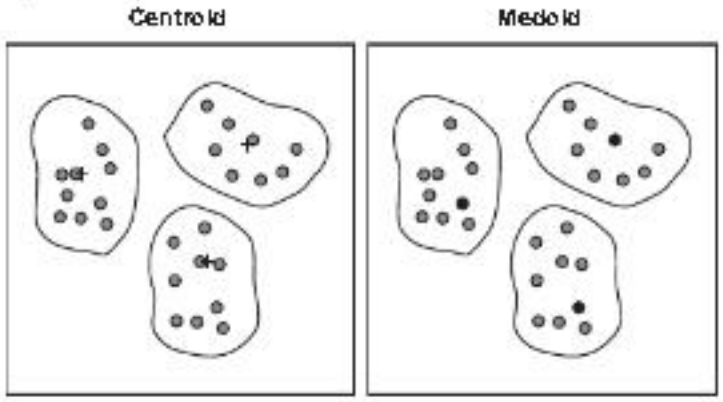


Fig. 2.10: Centroid and Medoid

Problem Statement : Illustrate the method to find k clusters from a collection of M objects with n attributes, the two dimensional case (n = 2) is examined.

Each object has two attributes, each object corresponding to the point (x_i, y_i), where x and y denote the two attributes

i = 1, 2 ... M.

For a given cluster of m points (m ~ M), i.e Centroid

Step 1 : Choose the value of k and the initial guesses for the centroids. Consider k=3 and the initial centroids are indicated by the points shaded in red, green, and blue as shown in Fig. 2.11

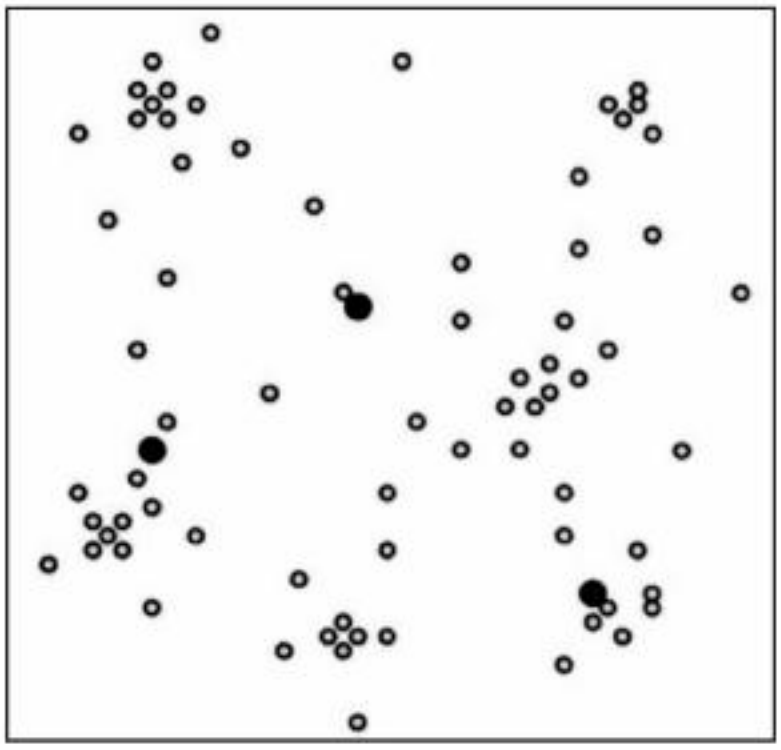


Fig. 2.11 : Initial starting points for the centroids

Step 2 : Compute the distance from each data point (x_i, y_i), to each centroid, and assign each point to the closest centroid .

This step is used to find out first k clusters association.

Now the distance d between two points (x_1, y_1) and (x_2, y_2) is calculated by following formula

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

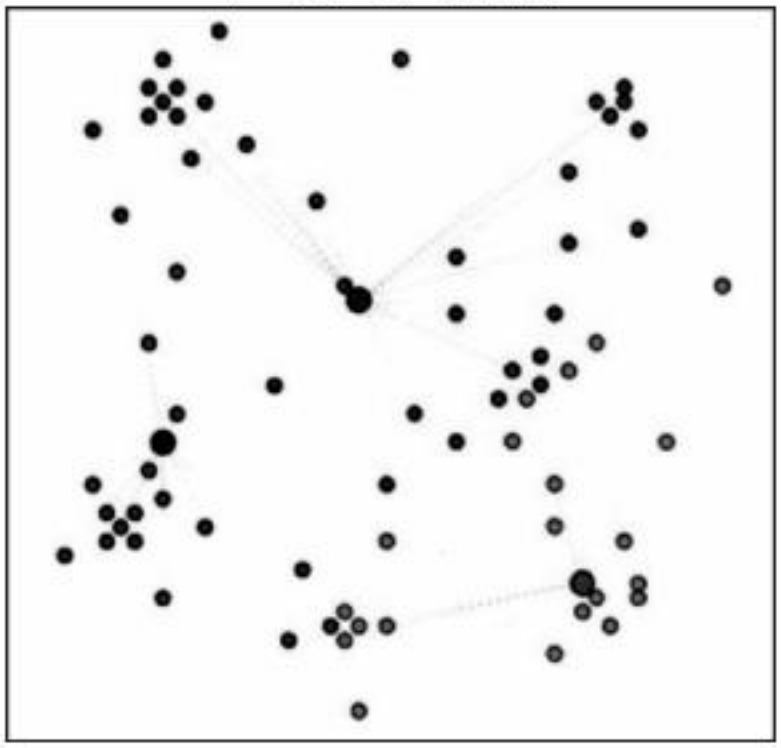


Fig. 2.12 : Points are assigned to closest centroid

Step 3 : Compute the centroid of each newly defined cluster from step 2

The centroid (x_c, y_c) is calculated as

$$(x_c, y_c) = \left[\frac{\sum_{i=1}^{m} x_i}{m}, \frac{\sum_{j=1}^{m} y_j}{m} \right]$$

In Fig. 2.13 the computed centroid are shown by using light shaded dots

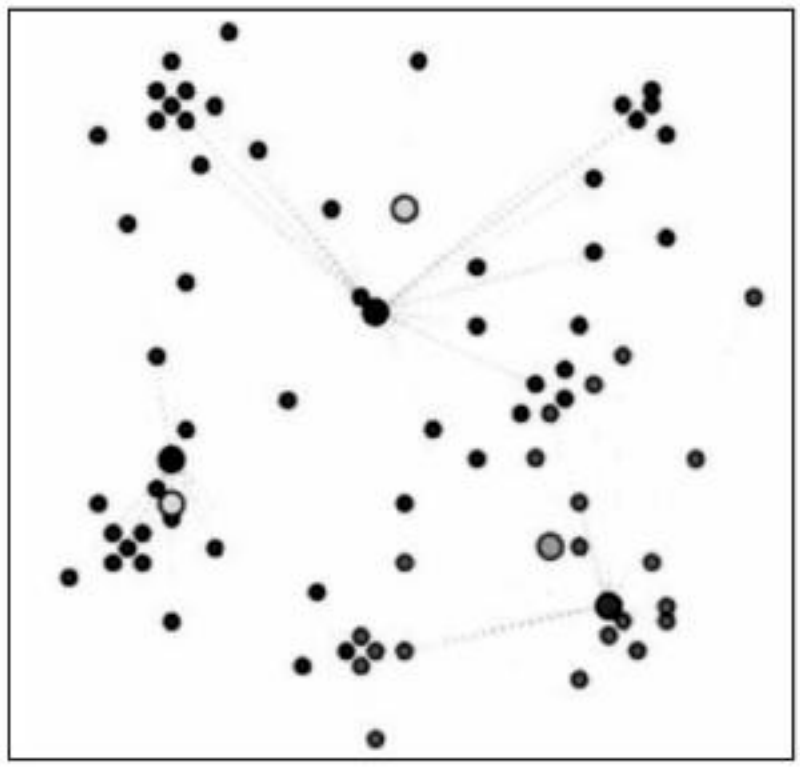


Fig. 2.13 : Compute the mean of each cluster

Step 4 : Repeat Steps 2 and 3 until the algorithm converges to an answer.

- Convergence occurs when the centroids do not change or when the centroids oscillate back and forth.
- This can occur when one or more points have equal distances from the centroid centers.

Example of K-Means

Consider 4 medicines as our training data point's object where each medicine has 2 attributes.

Each attribute represents coordinate of the object.

We have to determine which medicines belong to cluster 1 and which medicines belong to the other cluster.

Given Data

Object	Attribute 1 (X) : Weight Index	Attribute 2 (Y) : pH
Medicine A	1	1
Medicine B	2	1
Medicine C	4	3
Medicine D	5	4

Step 1 :

Consider initial centroids are $c_1 = (1, 1)$, $c_2 = (2, 1)$ and ploted as follows

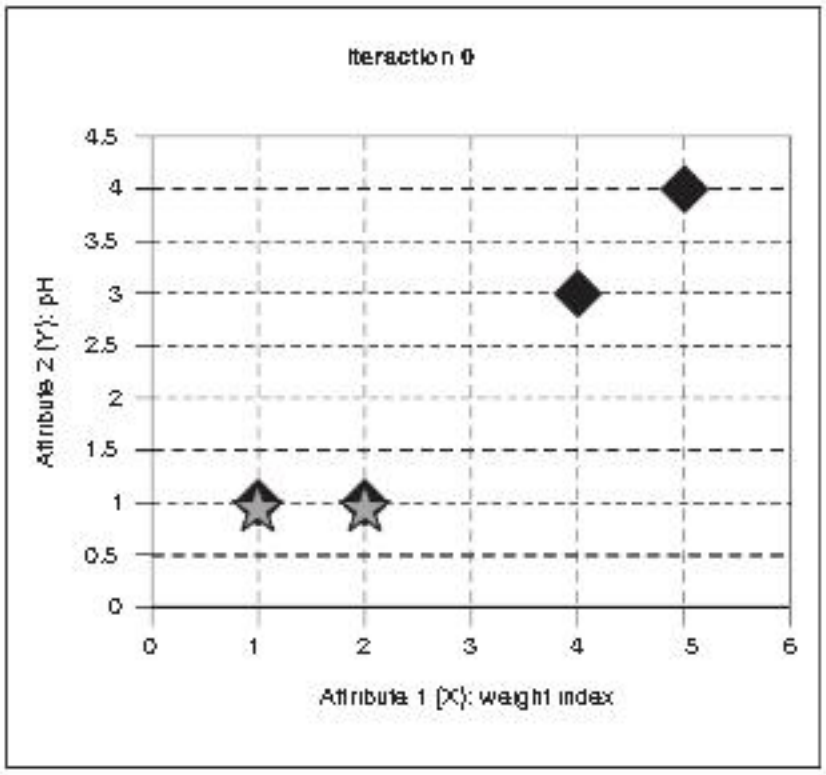


Fig. 2.14

Step 2 : Now compute distance between cluster centroid to each point

$$D^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \begin{matrix} c_1 = (1, 1) \text{ group} - 1 \\ c_2 = (2, 1) \text{ group} - 2 \end{matrix}$$

$$\begin{matrix} & A & B & C & D \end{matrix}$$

$$= \begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \begin{matrix} X \\ Y \end{matrix}$$

The first row of the distance matrix corresponds to the distance of each object to the first centroid and the second row is the distance of each object to the second centroid.

For example, distance from medicine C = (4, 3) to the first centroid $c_1 = (1, 1)$ is,

$$\sqrt{(4-1)^2 + (3-1)^2} = 3.61$$

and its distance to the second centroid is $c_2 = (2, 1)$ is

$$\sqrt{(4-2)^2 + (3-1)^2} = 2.83 \text{ etc.}$$

Now assign each object based on the minimum distance.

Medicine A is assigned to group 1, medicine B to group 2, medicine C to group 2 and medicine D to group 2.

The elements of Group matrix below is 1 if and only if the object is assigned to that group.

$$G^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \begin{matrix} \text{group} - 1 \\ \text{group} - 2 \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

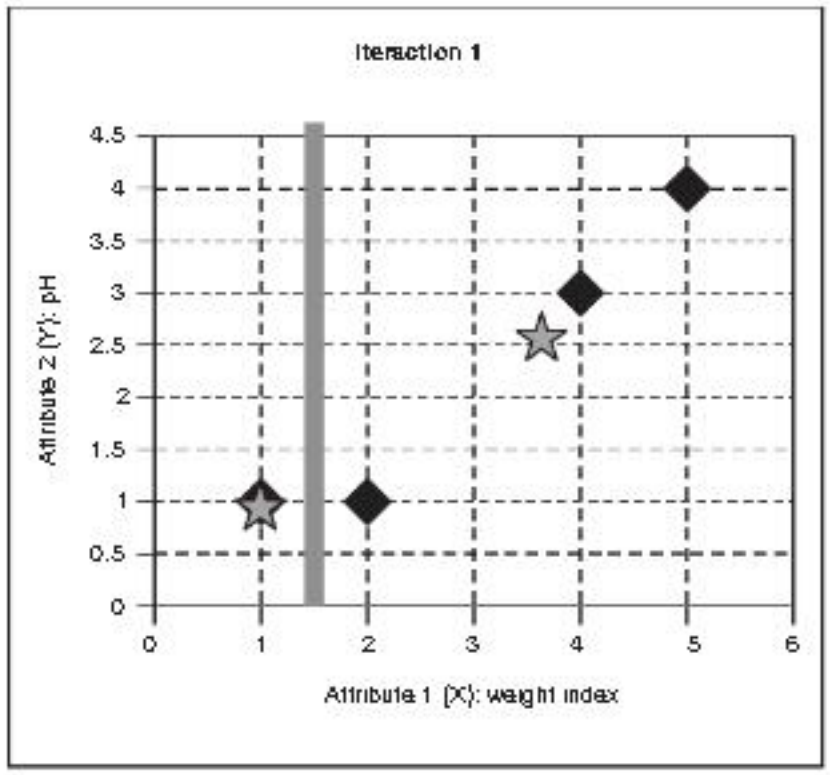


Fig. 2.15

Iteration 1 :

The next step is to compute the distance of all objects to the new centroids.

Similar to step 2, we have distance matrix at iteration 1 is

$$D^1 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} \begin{matrix} c_1 = (1, 1) \text{ group} - 1 \\ c_2 = \left(\frac{11}{3}, \frac{8}{3}\right) \text{ group} - 2 \end{matrix}$$

$$\begin{matrix} & A & B & C & D \end{matrix}$$

$$= \begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \begin{matrix} X \\ Y \end{matrix}$$

Based on the new distance matrix, we move the medicine B to Group 1 while all the other objects remain. The Group matrix is shown below,

$$G^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{matrix} \text{group} - 1 \\ \text{group} - 2 \end{matrix}$$

$$A \; B \; C \; D$$

Iteration 2 :

$$D^2 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \begin{matrix} c_1 = \left(1\frac{1}{2}, 1\right) \text{ group} - 1 \\ c_2 = \left(4\frac{1}{2}, 3\frac{1}{2}\right) \text{ group} - 2 \end{matrix}$$

$$\begin{matrix} & A & B & C & D \end{matrix}$$

$$= \begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \begin{matrix} X \\ Y \end{matrix}$$

$$G^2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{matrix} \text{group} - 1 \\ \text{group} - 2 \end{matrix}$$

$$A \; B \; C \; D$$

Here we can observe that $G^2 = G^1$ so we can summarize points as

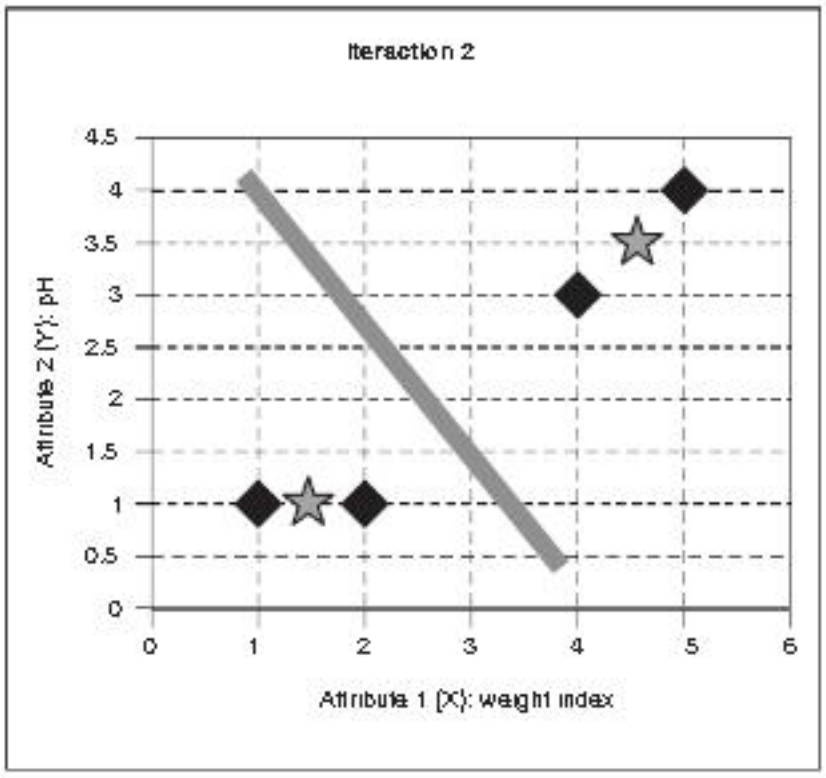


Fig. 2.16

Object	Feature 1(X) : Weight Index	Feature 2 (Y) : pH	Group (Result)
Medicine A	1	1	1
Medicine B	2	1	1
Medicine C	4	3	2
Medicine D	5	4	2

Generalize K Means for n-Dimensions

- To generalize the prior algorithm to n dimensions, suppose there are M objects, where each object is described by n attributes or property values (P_1, P_2, ... P_n). Then object i is described by for (P_{i1}, P_{i2}, ..., P_{in}), for i = 1, 2, ..., M.
- For a given point, R, at (R_1, P_{i2}, ..., R_n) and a centroidm, q, located at (q_1, q_2, ..., q_n), the distance, d, between Pi and q, is expressed as shown in

$$d(p, q) = \sqrt{\sum_{j=1}^{n} (p_{ij} - q_j)^2}$$

- The centroid q of a cluster of m points, (R_1, P_{i2}, ..., P_{in}), is calculated as shown in

$$(q_1, q_2, \ldots, q_n) = \left(\frac{\sum_{i=1}^{m} P_{i1}}{m}, \frac{\sum_{i=1}^{m} P_{i2}}{m}, \ldots, \frac{\sum_{i=1}^{m} P_{in}}{m}\right)$$

2.10 USE CASES

Several Problems can be Solved by Using K Means Algorithm Discussed as Follows :

Image Processing

- K-Means can be used for object detection for each frame of a video.
- For each frame, the task is to determine which pixels are most similar to each other.
- The attributes of each pixel can include brightness, color and location, the x and y coordinates in the frame.
- With security video images, changes in cluster are observed to indicate unauthorized access to a facility.

Medical

- In medical data patient attributes such as age, height, weight, systolic and diastolic blood pressures, cholesterol level, and other attributes can identify naturally occurring clusters
- These clusters can be used for specific preventive measures or clinical trial participation these clusters can be used.

Customer Segmentation

- To identify customers who have similar behaviors and spending patterns, marketing and sales groups use k-means approach.
- Consider example of wireless service provider.
- **They may Look at the Following Customer Attributes :**
  - Monthly bill,
  - Number of text messages,
  - Data volume consumed,
  - Minutes used during various daily periods
  - Years as a customer.
- Above data can be used by company to offer promotional offers to old customers and to gain new business.

2.11 DETERMINING NUMBER OF CLUSTERS

- k clusters can be identified in a given dataset, but what value of k should be selected?
- The value of k can be chosen based on a reasonable guess or some predefined requirement.
- How to know better or worse having k clusters versus k – 1 or k + 1 clusters

Solution :

- Use heuristic – e.g., Within Sum of Squares (WSS)
- WSS metric is the sum of the squares of the distances between each data point and the closest centroid
- The process of identifying the appropriate value of k is referred to as finding the "elbow" of the WSS curve

WSS Method

- Compute clustering algorithm (e.g., k-means clustering) for different values of k. For instance, by varying k from 1 to 10 clusters.
- For each k, calculate the total within-cluster sum of square (WSS).
- Plot the curve of WSS according to the number of clusters k.
- The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters.

$$\sum_{k=1}^{t} W(C_k) = \sum_{k=1}^{t} \sum_{x_i \in C_k} (x_i - \mu_k)^2$$

- where : xi -is a data point belonging to the cluster Ck
- μk is the mean value of the points assigned to the cluster Ck

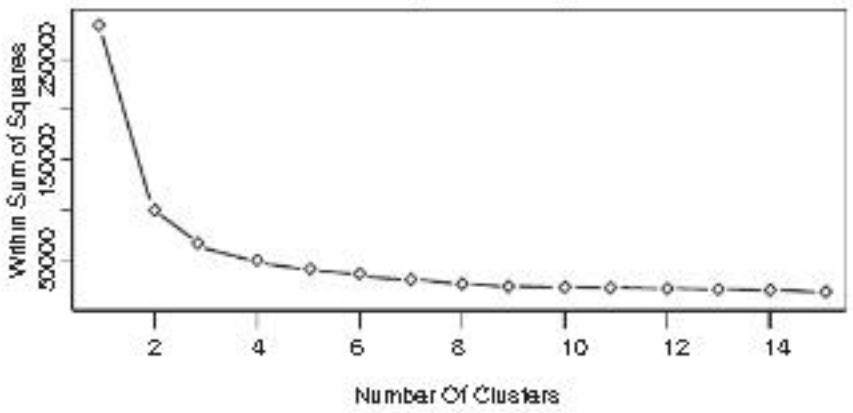


Fig. 2.17: WSS vs Number of clusters curve

In Fig. 2.17 the curve gradually decreases from one to two. At k=3 elbow forms and after that k>3 curve is linear so for k means algorithm k=3 is cluster value.

2.12 DIAGNOSTICS

- WSS can support heuristic which can then offer at least a number of potential k values which can be used.
- When the attributes are comparatively smaller in number, a common method to make more perfect selection of k value is to plot the data to decide how distinct the recognized clusters are from all the others.
- **While Exercising doing this the Following Aspects should be Attended to :**
  - Are clusters well detached from each other?
  - Does any of the clusters have only a small number of points?
  - Does any of the centroids look to be too close to the other?
  - Consider the following figure where four clusters are plotted having n=2.
- The four recognized clusters have enough space maintained between them.
- The Fig. 2.18 shows that clusters may be adjacent to each other, and the difference may not be so noticeable.

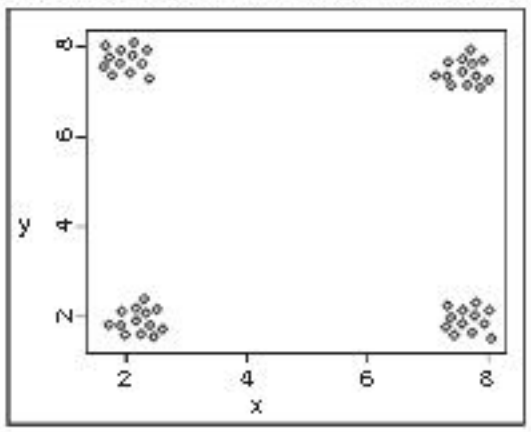


Fig. 2.18: Example of distinct clusters

- Such cases essentially demand to find out whether the results may or may not be effected using number of clusters.

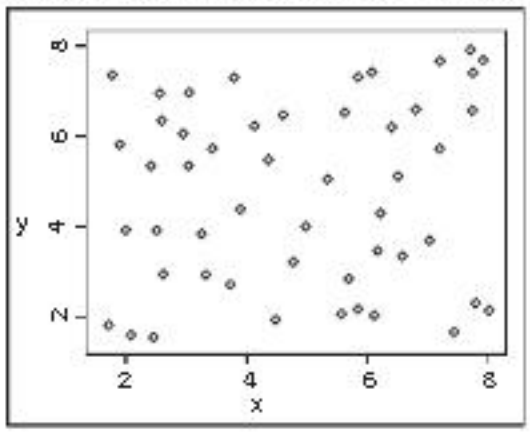


Fig. 2.19: less obvious cluster

- Example is the, following figure that makes use of 6 clusters to define the same dataset defined in the Fig. 2.20.

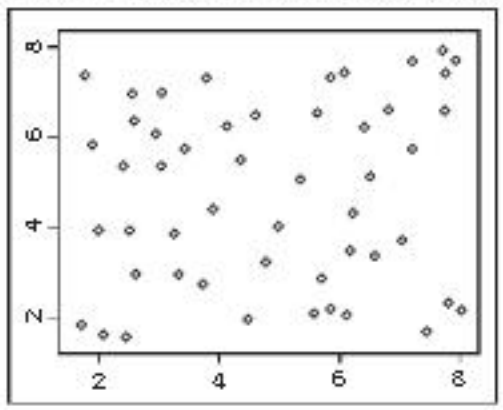


Fig. 2.20: Six clusters applied to the points

- When number of clusters are used it is occasionally not easy to distinguish, the groups whereas using less clusters, well distinguished groups can be easily understood.

2.13 REASONS TO CHOOSE AND CAUTIONS

There are Following Reasons to Choose the Cautions :

1. Rescaling
2. Object attributes
3. Additional consideration
4. Units of measure

These can be explained further

1. Rescaling

- Attributes we used to represent in dollars are common in clustering analysis and they can vary according to their magnitude from the other attributes.
- Assume that, if PI is stated in dollars and age is stated in years, the income attribute, often more than $20,000, can easily control the distance calculation with ages normally below the 100 years.
- Finally the resultant attribute will have a standard deviation which is = 1 and having no units.
- Dividing every attribute value by the suitable standard deviation and carrying out the k-means analysis yields the result as shown in Fig. 2.21.

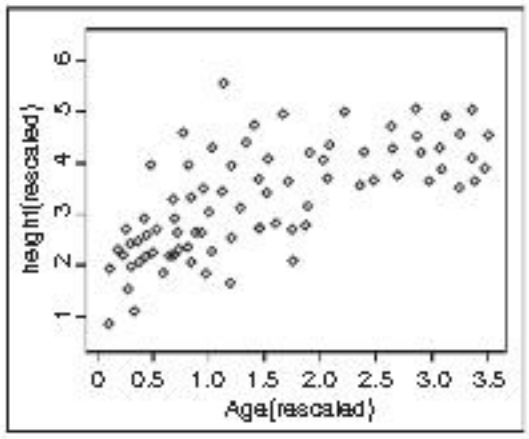


Fig. 2.21 : Clusters with rescaled attributes

2. Object Attributes :

- It is necessary to know, that which attributes be required during the assignment of new object to a cluster, and then to proceed to have the object that attributes to use in the analysis.
- Consider an example, information on present customers' fulfillment or purchase activity may be available, bust such information may not exists for possible customers.
- The data Expert/User can have more choices regarding the attributes to use in the clustering analysis.
- Whenever possible and also on the basis of the data, it is best to reduce the numerous attributes to the amount possible.
- In addition, to above lots of attributes can reduce the impact of the most significant variables.
- Besides, the use of number of same kinds of attributes can place highest degree of significance on single type of attributes.
- As an example, if 5 attributes associated to personal wealth are involved in a clustering analysis, the wealth attributes control the analysis and perhaps cover the significance of other attributes, like age.
- When dealing with the problem of number of attributes. It may be noted that one valuable method is to recognize any highly associated attributes and use only one or two of the associated attributes in the clustering analysis.

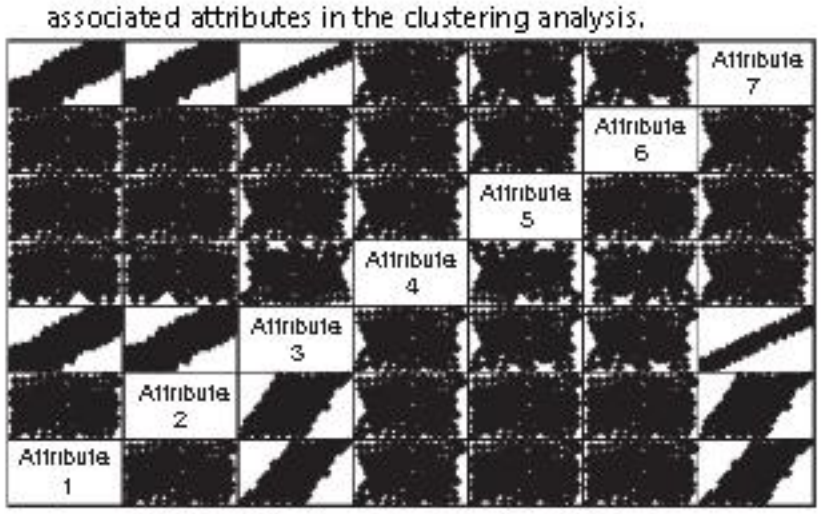


Fig. 2.22: Scatterplot matrix for 7 attributes

- In Fig. 2.22 a scatter plot matrix is used as a valuable tool to imagine the pair-wise relationships among the attributes.
- The Hrobust Hrelationship His Hdiscovered Hto Hbe Hamong Attribute 3 and Attribute 7.
- If value on any one of these attribute is identified then value of other attribute is identified with a close certainty.
- An option to decrease the numerous attributes is to combine number of attributes into single measure.

Units of Measure

- From a view of computational point the k-means algorithm is slightly unconcerned to the units of measure for a specified attributes.
- On the other hand, the algorithm will recognize various clusters based on the selection of the units of measure.
- For instance, assume that k-means is used to cluster patients based on age in years and height in centimeters.
- For k = 2, demonstrates the two clusters that would be determined for a specified dataset as shown in Fig. 2.23 below.

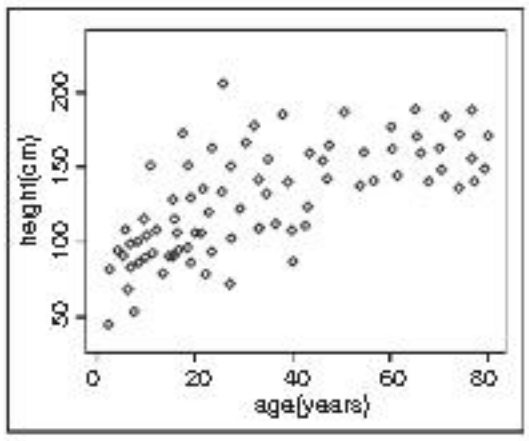


Fig. 2.23: Clusters with height expressed in centimeters

- When the height is rescaled from centimeters to meters the resulting clusters would be somewhat different, as shown in Fig. 2.24.

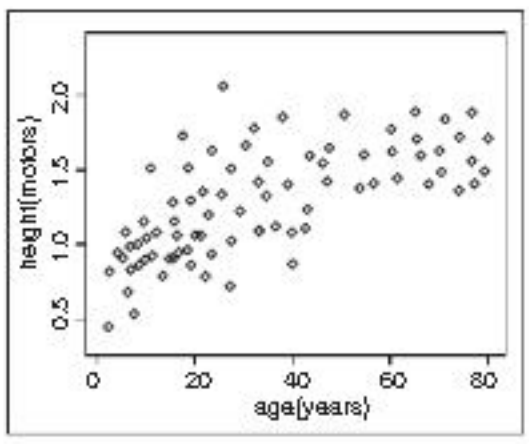


Fig. 2.24: Clusters with height expressed in meters

- When the height is displayed in meters, the magnitude of the ages controls the distance computations among two points.

3. Additional Considerations

- The k-means algorithm is complex to the beginning positions of the initial centroid.
- Therefore, it is essential to rerun the k-means analysis number of times for a specific value of k to make sure the cluster results offer the completely least WSS.
- A Euclidean distance function is used for assigning the points to the closest centroids.
- For doing above task other possible functions can also be used such as cosine similarity and the Manhattan distance functions.
- The cosine similarity function is frequently selected to compare Htwo Hdocuments Hwhich are Hdepend Hon Hthe occurrence of every word that present in every document.
- The following function is Manhattan distance function for two points p and q which are ranging from p_1 and q_1 to p_n and q_n respectively.

$$d_1(p, q) = \sum_{i=1}^{n} |p_i - q_i|$$

- The Manhattan distance function is similar to the distance covered by a car in a city, where the streets are positioned in a rectangular grid.

EXERCISE

1. HWhat is Type I and Type II errors?
2. HWhat is difference between Students-t method and Welcha's t method?
3. HExplain ANOVA with suitable example.
4. HWhat is Type I and Type II errors?
5. HWhat is difference between Students-t method and Welcha's t method?
6. HExplain ANOVA with suitable example.
7. HExplain Clustering Hand different Htype of Hclustering approaches.
8. HExplain different types of cluster type.
9. HExplain use cases of K means.
10. HWhat is K means? Explain with suitable example.
11. HExplain WSS method to decide number of clusters?

UNIT III

ASSOCIATION RULES AND REGRESSION

3.1 INTRODUCTION

Most of the established companies have accumulated masses of data from their customers for decades. With the e-commerce applications growing rapidly, the companies will have a significant amount of data in months not in years. Data Mining, also known as Knowledge Discovery in Databases (KDD), is to find trends, patterns, correlations, anomalies in these databases which can help us to make accurate future decisions. Mining Association Rules is one of the main application areas of Data Mining.

Given a set of customer transactions on items, the aim is to find correlations between the sales of items.

In this unit we are going to consider an example of such analysis, commonly known as market basket analysis. Market Basket Analysis is the discovery of relations or correlations among a set of items. Along with this we are also going to see frequent item sets, closed item sets, association rules and algorithms such as apriori and FP growth to generate association rules.

3.2 MARKET BASKET ANALYSIS

Association rules are of the form if X then Y. For example : 60% of those who buy comprehensive motor insurance also buy health insurance; 80% of those who buy books on-line also buy music on-line; 50% of those who have high blood pressure and are overweight have high cholesterol. These rules are actionable in that they can be used to target customers for marketing, or for product placing, or more generally to inform decision making.

Examples of Areas in which Association Rules have been Used Include :

- **Credit Card Transactions :** Items purchased by credit card give insight into other products the customer is likely to purchase.
- **Supermarket Purchases :** Common combinations of products can be used to inform product placement on supermarket shelves.
- **Telecommunication Product Purchases :** Commonly associated options (call waiting, caller display, etc) help to determine how to structure product bundles which maximize revenue.
- **Banking Services :** The patterns of services used by retail customers are used to identify other services they may wish to purchase.
- **Insurance Claims :** Unusual combinations of insurance claims can be a sign of fraud.
- **Medical Patient Histories :** Certain combinations of conditions can indicate increased risk of various complications.

Fig. 3.1 : Market Basket Analysis

We consider how to derive association rules directly from historical data, as opposed to via customer surveys or other means. Such data is characterized by being readily available in large quantities (and thus cheap), though it is often of poor quality or incomplete.

Let's discuss this problem in the context of supermarket purchases, which is where the terminology "Market Basket Analysis" comes from. The data consists of a number of transaction records, each containing a set of items purchased by that customer. Each row in this table corresponds to a transaction, which contain a unique identifier labeled TID and a set of items bought by a given customer. Retailers are interested in analyzing the data to learn about the purchasing behavior of their customers. Such valuable information can be used to support a variety of business related applications such as marketing promotions, inventory management, and customer relationship management.

Customer Purchases

Table 3.1 : An Example of Market Basket Transactions

TID	ITEMS
1	{ Tiling Cement, Tiles }
2	{ Paint, White Spirit}
3	{ Paint, Wallpaper, Plaster}
4	{ Paint, Plaster, Tiling Cement, Tiles }

- This unit presents a methodology known as association analysis which is useful in discovering interesting relationships hidden in large datasets. The uncovered relationships can be represented in the form of association rules or sets of frequent items. For example, The following rule can be extracted from the data set shown in Table 3.1 :

  {Tiling Cement, Tiles}

- The rule suggests that a strong relationship exists between the sale of Tiling Cement and Tiles because many customers who buy Tiling Cement also buy Tiles. Retailers can use this type of rules (information) to help them identify new opportunities for cross selling their products to the customers.
- Market Basket Transaction can also be represented in a binary format as shown in Table 3.2. In this table, each row corresponds to a transaction and each column corresponds to an item. An item can be treated as binary variable whose value is one if the item is present in a transaction and value is zero otherwise.

Table 3.2 : A Binary 0/1 Representation of Market Basket Transaction

TID	Tiling Cement	Tiles	Paint	White Spirit	Wallpaper	Plaster
1	1	1	0	0	0	0
2	0	0	1	1	0	0
3	0	0	1	0	1	1
4	1	1	1	0	0	1

- Because the presence of an item in a transaction is more important than its absence, an item is an asymmetric binary variable. This representation gives very simplistic view of market basket transaction as it ignores important aspects of the data such as quantity of the items sold, price paid to purchase.
- Apart from the market basket data, association analysis is also applicable to other application domains such as bioinformatics, medical diagnosis, web mining and scientific data analysis. In the analysis of Earth science data, the association patterns may reveal interesting relationships among ocean, land and atmospheric processes. Such information may help Earth scientist to develop a better understanding of how the different elements of the Earth system interact with each other.
- There are two important issues that need to be addressed when applying association analysis to market basket data. First, discovering patterns from a large transaction data set is a challenging task; Second, some of the discovered patterns are potentially bogus because they may happen simply by chance.

- Let S be the set of all possible purchases and let n be the number of transactions. Each transaction record is a subset of S. We consider rules of the form "$(x_1, x_2, ..., x_j)$ implies $(y_1, y_2, ..., y_k)$" where

  $x_1, x_2, ..., y_1, y_2, ...$ are elements of S. The collection $(x_1, x_2, ..., x_j)$ is called an itemset; read this as

  "x_1 and x_2 and ... and x_j".
- The support of the rule is defined as

  $$\text{Supp}((x_1, x_2, \ldots) \text{ implies } (y_1, y_2, \ldots)) = \frac{\text{No. transactions containing } x_1, x_2, \ldots \text{ and } y_1, y_2 \ldots}{n}$$

- More generally we define the support of an itemset as

  $$\text{Supp}(x_1, x_2, \ldots) = \frac{\text{No. transaction containign } x_1, x_2, \ldots}{n}$$

- The confidence of the rule is

$$\text{Conf}((x_1, x_2, \ldots) \text{ implies } (y_1, y_2, \ldots)) = \frac{\text{Supp}((x_1, x_2, \ldots) \text{ implies } (y_1, y_2, \ldots))}{\text{Supp}(x_1, x_2, \ldots)}$$

- To consider a rule, we impose a minimum support, indicating a reasonable amount of data about the rule. The confidence measures how good a predictor the rule is. If we specify a minimum support s_0 and a minimum confidence c_0, then a strong rule is one which has Supp$((x_1, x_2, ...)$ implies $(y_1, y_2, ...)) > s_0$ and Conf$((x_1, x_2, ...)$ implies $(y_1, y_2, ...)) > c_0$.
- Support and a high confidence do not necessarily mean that a rule is interesting. The lift or improvement of the rule is

$$\text{Lift}(x_1, x_2, \ldots) \text{ implies } (y_1, y_2, \ldots)) = \frac{\text{Supp}(x_1, x_2, \ldots \text{ and } y_1, y_2, \ldots)}{(\text{Supp}(x_1, x_2, \ldots) \text{ Supp}(y_1, y_2, \ldots))}$$

- The lift is > 1 if the association between $(x_1, x_2, ...)$ and $(y_1, y_2, ...)$ is due to more than just chance. It corresponds to positive correlation between the events "purchased $x_1, x_2, ...$" and "purchased $y_1, y_2, ...$".

  For the data above we have, for example

  Supp(Paint implies White Spirit) = 1/4

  Conf(Paint implies White Spirit) = 1/3

  Lift(Paint implies White Spirit) = 4/3

  Supp(Paint and Plaster implies Wallpaper) = 1/4

  Conf(Paint and Plaster implies Wallpaper) = 1/2

  Lift(Paint and Plaster implies Wallpaper) = 2

Note : This can all be rephrased in terms of conditional probability using counting measure. Let Ω be the set of records and for any single record ω we put $P(\omega) = 1/|\Omega| = 1/n$. Define event E_A to be the set of records containing item set A, then Supp (A) = $P(E_A)$ and Conf (A implies B) = $P(E_B \mid E_A)$. We see that Lift (A implies B) is > 1 if and only if knowing that A is in a record increase the probability that B in the record.

- We would like to find all rules with good lift. In practice there are too many rules to search through for this to be practical, however it turns out that if we restrict ourselves to strong rules then the problem becomes tractable. (Refer to Apriori algorithm)
- An alternative to using the lift to measure the interest of a rule is to use the significance. The significance is calculated using a 2*2 contingency table. This gives the observed frequencies of all possible combinations of transactions containing item sets $(x_1, x_2, ...)$ and transactions containing item sets $(y_1, y_2, ...)$.

Table 3.3

	$(x_1, x_2, ...)$	**Not** $(x_1, x_2, ...)$	**Total**
$(y_1, y_2, ...)$	Supp($x_1, x_2, ...$ and $y_1, y_2, ...$)	Supp($y_1, y_2, ...$) – Supp ($x_1, x_2, ...$ and $y_1, y_2, ...$)	Supp($y_1, y_2, ...$)
Not $(y_1, y_2, ...)$	Supp($x_1, x_2, ...$) – Supp($x_1, x_2, ...$ and $y_1, y_2, ...$)	1 – Supp ($y_1, y_2, ...$) + Supp($x_1, x_2, ...$ and $y_1, y_2, ...$)	1 – Supp ($y_1, y_2, ...$)
Total	Supp($x_1, x_2, ...$)	1 – Supp($x_1, x_2, ...$)	1

- If the occurrence of $(x_1, x_2, ...)$ and $(y_1, y_2, ...)$ is independent, then we would expect Supp $(x_1, x_2, ...$and $y_1, y_2, ...)$ = Supp$(x_1, x_2, ...)$* Supp$(y_1, y_2, ...)$.
- Association rules are not always useful, even if they have high support, confidence and lift > 1. For example the rule "Customers who purchase maintenance agreements also purchase large appliances" might have good confidence and lift, but is still not useful. We can classify rules as useful, trivial and inexplicable.
- Useful rules are the ones we want, with high quality actionable information. Trivial rules will already be known by anyone familiar with the business. Inexplicable rules are those which have no apparent explanation and do not suggest a course of action. An example of the latter is the famous "Men who buy nappies on Thursdays also buy beer" rule.
- There is no automatic way of identifying trivial or inexplicable rules. In practice one needs to experiment with the choice of the minimum levels of support and confidence, s_0 and c_0, to find all the interesting rules without including too many others. Typically rules where the "consequent" $(y_1, y_2, ...)$ consists of a single item y are the most useful.

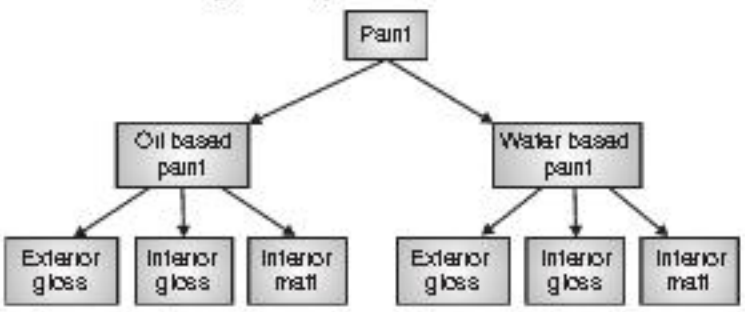


Fig. 3.2 : Association rules : examples

- Association rules can also be improved by combining purchase items. Items often fall into natural hierarchies. For example, in many cases better rules can be obtained by grouping items together according to this taxonomy. That is, rather than consider "red oil based exterior gloss" and "blue oil based exterior gloss" as separate items we combine them as "oil based exterior gloss", or even as "oil based paint" or just "paint". As a rule of thumb, market basket analysis tends to work better when individual items have roughly the same level of support.
- Another way of extracting good rules from bad is to consider negations. If the rule "$(x_1, x_2, ...)$ implies $(y_1, y_2, ...)$" has lift < 1 then the rule "$(x_1, x_2, ...)$ implies not $(y_1, y_2, ...)$" has lift > 1. One should note however that such a rule is often not actionable, in that it does not lead to a useful course of action.
- Association analysis results should be interpreted with caution. The inference made by an association rule does not necessarily imply causality. Instead it suggests a strong co-occurrence relationship between items in the antecedent and consequent of the rule. Causality on the other hand requires knowledge about the causal and effect of attributes in the data and typically involves relationships occurring over time.

Importance of Support and Confidence

- Support is an important measure because a rule that has very low support may occur simply by chance. A low support rule is also likely to be unimportant from business perspective because it may not be profitable to promote items that customers buy together. For these reasons, support is often used eliminate unimportant rules.
- Confidence on the other hand, measures the reliability of the inference made by a rule. For a given rule, X->Y, higher the confidence, more likely it is for Y to be present in the transactions that contain X. Confidence also provides the estimate of the conditional probability of Y given X.

3.3 FREQUENT ITEMSETS

- Frequent sets play an essential role in many Data Mining tasks that try to find interesting patterns from databases, such as association rules, correlations, sequences, episodes, classifiers and clusters. The mining of association rules is one of the most popular problems of all these.
- The identification of sets of items, products, symptoms and characteristics, which often occur together in the given database, can be seen as one of the most basic tasks in Data Mining.
- The original motivation for searching frequent sets came from the need to analyze so called supermarket transaction data, that is, to examine customer behavior in terms of the purchased products. Frequent sets of products describe how often items are purchased together.

- Formally let I be the set of items.
- A transaction over I is a couple T = (tid, I) where tid is the transaction identifier and I is the set of items from I.
- A database D over I is a set of transactions over I such that each transaction has a unique identifier.
- A transaction T = (tid, I) is said to support a set X, if X C I. The cover of a set X in D consists of the set of transaction identifiers of transactions in D that supports X. The support of a set X in D is the number of transactions in the cover of X in D. The frequency of a set X in D is the probability that X occurs in a transaction, or in other words, the support of X divided by the total number of transactions in the database. A set is called frequent if its support is no less than a given absolute minimal support threshold min_sup with $0 \leq min\_sup_{abs} \leq |D|$. When working with frequencies of sets instead of their support, we use the relative minimal frequency threshold min_sup_{rel}, with $0 > min\_sup_{rel} > 1$. Obviously $min\_sup_{abs} = [min\_sup_{rel} * |D|$. Here we will mostly use the absolute minimal support threshold and omit the subscript abs. Let D be a database of transactions over a set of items I, and min_sup the minimal support threshold. The collection of frequent sets in D with respect to min_sup is denoted by
  F(D, min_sup) :={X C I | support(X, D) >= min_sup} or simply F if D and min_sup are clear from the context.
- Given a set of items I, a database of transactions D over I, and a minimal support threshold min_sup, find F (D, min_sup).
- In practice we are not only interested in the set of sets F, but also in the actual supports of these sets.
- For example, consider the database shown in the following table over the set of items I = {beer, chips, pizza, wine} :

Table 3.4

TID	Itemsets
100	{beer, chips, wine}
200	{beer, chips}
300	{pizza, wine}
400	{chips, pizza}

The Table 3.5 shows all frequent sets in D with respect to a minimal support threshold equal to 1, their cover in D, plus their support and frequency :

Table 3.5

Set	Cover	Support	Frequency
{}	{100,200,300,400}	4	100%
{beer}	{100,200}	2	50%
{chips}	{100,200,400}	3	75%
{pizza}	{300,400}	2	50%
{wine}	{100,300}	2	50%
{beer, chips}	{100,200}	2	50%
{beer, wine}	{100}	1	25%
{chips, pizza}	{400}	1	25%
{chips, wine}	{100}	1	25%
{pizza, wine}	{300}	1	25%
{beer, chips, wine}	{100}	1	25%

- If we are given the support threshold min_sup, then every frequent set X also represents the trivial rule X=> {} which holds with 100% confidence.
- The task of discovering all frequent sets is quite challenging. The search space is exponential in the number of items occurring in the database and the targeted databases tend to be massive, containing millions of transactions. Both these characteristics make it a worthwhile effort to seek the most efficient techniques to solve this task.

3.3.1 Generating Association Rules from Frequent Itemsets

- Once the frequent itemsets from transactions in a database D have been found, it is straightforward to generate strong association rules from them, where strong association rules satisfy both minimum support and minimum confidence. This can be done using the following equation :

  confidence(A=>B) = P(B|A) = support_count(AUB) / support_count(A)

- The conditional probability is expressed in terms of itemset support count, where :
  support_count(AUB) is the number of transactions containing the itemsets AUB and
  support_count(A) is the number of transactions containing the itemset A. Based on this equation, association rules can be generated as follows :
  - For each frequent itemset l, generate all nonempty subsets of l.
  - For every nonempty subset s of l, output the rule "s => (l-s)" if support_count(l) / support_count(s)>=min_conf, where min_conf is the minimm confidence threshold.
- Let's try an example based on the transactional data shown on Table 3.6. Suppose the data contain the frequent itemset l = {I1, I2, I5}. The nonempty subsets of l are : {I1, I2}, {I1, I5}, {I2, I5}, {I1}, {I2} and {I5}. The resulting association rules are as shown below each listed with its confidence

Table 3.6

I1 and I2=>I5	Conf=2/4=50%
I1 and I5=>I2	Conf=2/2=100%
I2 and I5=> I1	Conf=2/2=100%
I1=>I2 and I5	Conf=2/6=33%
I2=>I1 and I5	Conf=2/7=29%
I5=>I1 and I2	Conf=2/2=100%

- If the minimum confidence threshold is 70%, then only the second, third and last rules above are output, because these are the only ones generated that are strong.

3.4 ASSOCIATION RULE MINING

- Association rule mining is one of the most important and well researched techniques of data mining. It aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in the transaction databases or other data repositories.
- Association rules are widely used in various areas such as telecommunication networks, market and risk management, inventory control etc. Various association mining techniques and algorithms will be briefly introduced and compared later.
- Association rule mining is to find out association rules that satisfy the predefined minimum support and confidence from a given database. The problem is usually decomposed into two subproblems. One is to find those itemsets whose occurrences exceed a predefined threshold in the database; those itemsets are called frequent or large itemsets. The second problem is to generate association rules from those large itemsets with the constraints of minimal confidence.
- Suppose one of the large itemsets is L_k, $L_k = \{I_1, I_2, \dots, I_k\}$, association rules with this itemsets are generated in the following way : the first rule is $\{I_1, I_2, \dots, I_{k-1}\} \Rightarrow \{I_k\}$, by checking the confidence this rule can be determined as interesting or not. Then other rule are generated by deleting the last items in the antecedent and inserting it to the consequent, further the confidences of the new rules are checked to determine the interestingness of them.
- Those processes iterated until the antecedent becomes empty. Since the second subproblem is quite straight forward, most of the researches focus on the first subproblem. The first sub-problem can be further divided into two sub-problems : candidate large itemsets generation process and frequent itemsets generation process.
- We call those itemsets whose support exceed the support threshold as large or frequent itemsets, those itemsets that are expected or have the hope to be large or frequent are called candidate itemsets. In many cases, the algorithms generate an extremely large number of association rules, often in thousands or even millions. Further, the association rules are sometimes very large.
- It is nearly impossible for the end users to comprehend or validate such large number of complex association rules, thereby limiting the usefulness of the data mining results.
- Several strategies have been proposed to reduce the number of association rules, such as generating only "interesting" rules, generating only "nonredundant" rules, or generating only those rules satisfying certain other criteria such as coverage, leverage, lift or strength.
- Let $I = I_1, I_2, \dots, I_m$ be a set of m distinct attributes, T be transaction that contains a set of items such that $T \subseteq I$, D be a database with different transaction records Ts. An association rule is an implication in the form of $X \Rightarrow Y$, where $X, Y \subset I$ are sets of items called itemsets, and $X \cap Y = \phi$. X is called antecedent while Y is called consequent, the rule means X implies Y.
- There are two important basic measures for association rules, support(s) and confidence(c). Since the database is large and users concern about only those frequently purchased items, usually thresholds of support and confidence are predefined by users to drop those rules that are not so interesting or useful.
- The two thresholds are called minimal support and minimal confidence respectively. Support(s) of an association rule is defined as the percentage/fraction of records that contain $X \cup Y$ to the total number of records in the database.
- Suppose the support of an item is 0.1%, it means only 0.1 percent of the transaction contain purchasing of this item.
- Confidence of an association rule is defined as the percentage/fraction of the number of transactions that contain $X \cup Y$ to the total number of records that contain X.
- Confidence is a measure of strength of the association rules, suppose the confidence of the association rule $X \Rightarrow Y$ is 80%, it means that 80% of the transactions that contain X also contain Y together.
- In general, a set of items (such as the antecedent or the consequent of a rule) is called an itemset. The number of items in an itemset is called the length of an itemset. Itemsets of some length k are referred to as k-itemsets. Generally, an association rules mining algorithm contains the following steps :
  - The set of candidate k-itemsets is generated by 1-extensions of the large (k – 1)- itemsets generated in the previous iteration.
  - Supports for the candidate k-itemsets are generated by a pass over the database.
  - Itemsets that do not have the minimum support are discarded and the remaining itemsets are called large k-itemsets.
- This process is repeated until no more large itemsets are found. The AIS algorithm was the first algorithm proposed for mining association rule. In this algorithm only one item consequent association rules are generated, which means that the consequent of those rules only contain one item, for example we only generate rules like $X \cap Y \Rightarrow Z$ but not those rules as $X \Rightarrow Y \cap Z$.

- The main drawback of the AIS algorithm is too many candidate itemsets that finally turned out to be small are generated, which requires more space and wastes much effort that turned out to be useless. At the same time this algorithm requires too many passes over the whole database.
- Apriori is more efficient during the candidate generation process. Apriori uses pruning techniques to avoid measuring certain itemsets, while guaranteeing completeness. These are the itemsets that the algorithm can prove will not turn out to be large. However there are two bottlenecks of the Apriori algorithm. One is the complex candidate generation process that uses most of the time, space and memory.
- Another bottleneck is the multiple scan of the database. Based on Apriori algorithm, many new algorithms were designed with some modifications or improvements.
- Increasing the Efficiency of Association Rules Algorithms The computational cost of association rules mining can be reduced in four ways :
  1. By reducing the number of passes over the database
  2. By sampling the database
  3. By adding extra constraints on the structure of patterns
  4. Through parallelization.
- In recent years much progress has been made in all these directions.
- Reducing the number of passes over the database FP-Tree, frequent pattern mining, is another milestone in the development of association rule mining, which breaks the main bottlenecks of the Apriori. The frequent itemsets are generated with only two passes over the database and without any candidate generation process. FP-tree is an extended prefix-tree structure storing crucial, quantitative information about frequent patterns.
- Only frequent length-1 items will have nodes in the tree, and the tree nodes are arranged in such a way that more frequently occurring nodes will have better chances of sharing nodes than less frequently occurring ones.
- FP-Tree scales much better than Apriori because as the support threshold goes down, the number as well as the length of frequent itemsets increase dramatically.
- The candidate sets that Apriori must handle become extremely large, and the pattern matching with a lot of candidates by searching through the transactions becomes very expensive. The frequent patterns generation process includes two sub processes : constructing the FT-Tree, and generating frequent patterns from the FP-Tree.
- The mining result is the same with Apriori series algorithms. To sum up, the efficiency of FP-Tree algorithm account for three reasons.
  1. First the FP-Tree is a compressed representation of the original database because only those frequent items are used to construct the tree, other irrelevant information are pruned.
  2. Secondly this algorithm only scans the database twice.
  3. Thirdly, FP-Tree uses a divide and conquer method that considerably reduced the size of the subsequent conditional FP-Tree. Every algorithm has his limitations, for FP-Tree it is difficult to be used in an interactive mining system.
- During the interactive mining process, users may change the threshold of support according to the rules. However for FP-Tree the changing of support may lead to repetition of the whole mining process. Another limitation is that FP-Tree is that it is not suitable for incremental mining.
- Since as time goes on databases keep changing, new datasets may be inserted into the database, those insertions may also lead to a repetition of the whole process if we employ FP-Tree algorithm.

3.4.1 Categories of Databases in which Association Rules are Applied

- Transactional database refers to the collection of transaction records, in most cases they are sales records. With the popularity of computer and e-commerce, massive transactional databases are available now. Data mining on transactional database focuses on the mining of association rules, finding the correlation between items in the transaction records.
- One of data mining techniques, generalized association rule mining with taxonomy, is potential to discover more useful knowledge than ordinary flat association rule mining by taking application specific information into account. In particular in retail one might consider as items particular brands of items or whole groups like milk, drinks or food. The more general the items chosen the higher one can expect the support to be.
- Thus one might be interested in discovering frequent itemsets composed of items which themselves form a taxonomy. Earlier work on mining generalized association rules ignore the fact that the taxonomies of items cannot be kept static while newtransactions are continuously added into the original database.
- How to effectively update the discovered generalized association rules to reflect the database change with taxonomy evolution and transaction update is a crucial task.

- Spatial association rules describe the relationship between one set of features and another set of features in a spatial database. The spatial operations that used to describe the correlation can be within, near, next to, etc. The form of spatial association rules is also X⇒ Y, where X, Y are sets of predicates and of which some are spatial predicates, and at least one must be a spatial predicate.
- Temporal association rules can be more useful and informative than basic association rules. For example rather than the basic association rule {diapers}⇒ {beer}, mining from the temporal data we can get a more insight rule that the support of {diapers} ⇒{beer} jumps to 50% during 6pm to 9pm everyday, obviously retailers can make more efficient promotion strategy by using temporal association rule.

3.5 APRIORI ALGORITHM

- In data mining, association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases. Based on the concept of strong rules, Agrawal introduced association rules for discovering regularities between products in large scale transaction data recorded by Point-Of-Sale (POS) systems in supermarkets.
- For example, the rule {onion,potatoes}=>{burger} found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, he or she is likely to also buy burger. Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements.
- In addition to the above example from market, analysis association rules are employed today in many application areas including Web usage mining, intrusion detection and bioinformatics.
- In computer science and data mining, Apriori is a classic algorithm for learning association rules.
- Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation). Other algorithms are designed for finding association rules in data having no transactions having no timestamps the problem of association rule mining is defined as :
- Let I = $\{i_1, i_2, ..., i_n\}$ be a set of n binary attributes called items. Let D = $\{t_1, t_2, ..., t_n\}$ be a set of transactions called the database. Each transaction in D has a unique transaction ID and contains a subset of the items in I. A rule is defined as an implication of the form $X \rightarrow Y$ where $X, Y \subseteq I$ and $X \cap Y = \phi$. The sets of items (for short itemsets) X and Y are called antecedent (left-hand-side or LHS) and consequent (right-hand-side or RHS) of the rule respectively.
- To illustrate the concepts, we use a small example from the supermarket domain. The set of items is I = {milk, bread, butter, beer} and a small database containing the items (1 codes presence and 0 absence of an item in a transaction) is shown in the table below. An example rule for the supermarket could be {milk,bread}=>{butter} meaning that if milk and bread is bought, customers also buy butter.

Note : this example is extremely small. In practical applications, a rule needs a support of several hundred transactions before it can be considered statistically significant, and datasets often contain thousands or millions of transactions. To select interesting rules from the set of all possible rules, constraints on various measures of significance and interest can be used. The best-known constraints are minimum thresholds on support and confidence.

Table 3.7

TID	Milk	Bread	Butter	Beer
1	1	1	0	0
2	0	1	1	0
3	0	0	0	1
4	1	1	1	0
5	0	1	0	0
6	1	0	0	0
7	0	1	1	1
8	1	1	1	1
9	0	1	0	1
10	1	1	0	0
11	1	0	0	0
12	0	0	0	1
13	1	1	1	0
14	1	0	1	0
15	1	1	1	1

Support

- The support supp(X) of an itemset X is defined as the proportion of transactions in the data set which contain the itemset.

  supp(X) = no. of transactions which contain the itemset X / total no. of transactions
- In the example database, the itemset {milk,bread,butter} has a support of 4 /15 = 0.26 since it occurs in 26% of all transactions. To be even more explicit we can point out that 4 is the number of transactions from the database which contain the itemset {milk,bread,butter} while 15 represents the total number of transactions.

Confidence

- The confidence of a rule is defined as :
- The rule {milk,bread}=>{butter} we have the following confidence :

  supp({milk,bread,butter}) / supp({milk,bread}) = 0.26 / 0.4 = 0.65

  This means that for 65% of the transactions containing milk and bread the rule is correct.
- Confidence can be interpreted as an estimate of the probability P(Y | X), the probability of finding the RHS of the rule in transactions under the condition that these transactions also contain the LHS.

Lift

- The lift of a rule is defined as :

  The rule {milk,bread}=>{butter} has the following lift :

  supp({milk,bread,butter}) / supp({butter}) x supp({milk,bread})= 0.26/0.46 x 0.4= 1.4

Conviction

- The conviction of a rule is defined as :

  The rule {milk,bread}=>{butter} has the following conviction :

  1 – supp({butter})/ 1- conf{milk,bread}=>{butter}) = 1-0.46/1-0.65 = 1.54
- The conviction of the rule X=>Y can be interpreted as the ratio of the expected frequency that X occurs without Y (that is to say, the frequency that the rule makes an incorrect prediction) if X and Y were independent divided by the observed frequency of incorrect predictions.
- In this example, the conviction value of 1.54 shows that the rule {milk,bread} => {butter} would be incorrect 54% more often (1.54 times as often) if the association between X and Y was purely random chance.

Apriori Algorithm

General Process

- Association rule generation is usually split up into two separate steps :
  1. First, minimum support is applied to find all frequent itemsets in a database.
  2. Second, these frequent itemsets and the minimum confidence constraint are used to form rules.

  While the second step is straightforward, the first step needs more attention.
- Finding all frequent itemsets in a database is difficult since it involves searching all possible itemsets (item combinations). The set of possible itemsets is the power set over I and has size $2^n - 1$ (excluding the empty set which is not a valid itemset). Although the size of the powerset grows exponentially in the number of items n in I, efficient search is possible using the downward-closure property of support (also called anti-monotonicity) which guarantees that for a frequent itemset, all its subsets are also frequent and thus for an infrequent itemset, all its supersets must also be infrequent.
- Exploiting this property, efficient algorithms (e.g., Apriori and Eclat) can find all frequent itemsets.

Apriori Algorithm Pseudocode

```
procedure Apriori (T, minSupport) { //T is the database and minSupport is the minimum support
L1= {frequent items};
for (k= 2; Lk-1 !=∅ ; k++) {
Ck= candidates generated from Lk-1
//that is cartesian product Lk-1 x Lk-1 and eliminating any k-1 size itemset that is not
//frequent
for each transaction t in database do{
#increment the count of all candidates in Ck that are contained in t
Lk = candidates in Ck with minSupport
}//end for each
}//end for
return Uk Lk ;
}
```

- As is common in association rule mining, given a set of itemsets (for instance, sets of retail transactions, each listing individual items purchased), the algorithm attempts to find subsets which are common to at least a minimum number C of the itemsets. Apriori 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. The algorithm terminates when no further successful extensions are found.
- Apriori uses breadth-first search and a tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length k – 1. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.
- Apriori, while historically significant, suffers from a number of inefficiencies or trade-offs, which have spawned other algorithms. Candidate generation generates large numbers of subsets (the algorithm attempts to load up the candidate set with as many as possible before each scan). Bottom-up

subset exploration (essentially a breadth-first traversal of the subset lattice) finds any maximal subset S only after all $2^{|S|} - 1$ of its proper subsets.

3.5.1 Frequent Itemset Generation in the Apriori Algorithm

- Apriori is the first association rule mining algorithm that uses support-based pruning to systematically control the exponential growth of candidate itemsets.

Table 3.8

Candidate 1 Itemset	
Item	**Count**
Beer	3
Bread	4
Cola	2
Diaper	4
Milk	4
Eggs	1

- From the table items – Cola and Eggs are removed because of low of support

Table 3.9

Candidate 2 Itemset	
Itemset	**Count**
{Beer, Bread}	2
{Beer, Diaper}	3
{Beer, Milk}	2
{Bread, Diaper}	3
{Bread, Milk}	3
{Diaper, Milk}	3

- From the table {Beer, Bread} and {Beer, Milk} itemsets are removed because of low support.

Table 3.10

Candidate 3 Itemset	
Item	**Count**
{Bread, Diaper, Milk}	3

- We assume that the support threshold is 60% which is equivalent to a minimum support count equal to 3.
- Initially every item is considered as a candidate 1-itemset. After counting their supports, the candidate itemsets{Cola} and {Eggs} are removed as they appear in fewer than three transactions. In the next iteration, candidate 2 itemsets are generated using only the frequent 1-itemsets because the Apriori principle ensures that all supersets of the infrequent 1-itemsets must be infrequent.
- Because there are only four frequent 1 itemsets, the number of candidate 2-itemsets generated by the algorithm is $\binom{4}{2} = 6$. Two of these six candidates {Beer, Bread} and {Beer, Milk} are subsequently found to be infrequent after computing their support their values. The remaining four candidates are frequent and will be used to generate candidate 3-itemsets. Without support based pruning, there are $\binom{6}{3} = 20$ candidate 3-itemsets that can be formed using six items given in the example. With the Apriori algorithm, we only need to keep candidate 3-itemsets whose subsets are frequent. The only candidate that has this property is {Bread, Diaper, Milk}
- The effectiveness of the Apriori pruning strategy can be shown by counting the number candidate itemsets generated. A brute-force strategy of calculating all itemsets (up to size 3) as candidates will produce

$$\binom{6}{1} + \binom{6}{2} + \binom{6}{3} = 6 + 15 + 20 = 41 \text{ candidates}$$

With the Apriori algorithm this number decreases to

$$\binom{6}{1} + \binom{4}{2} + 1 = 6 + 6 + 1 = 13 \text{ candidates.}$$

This represents 68% reduction in the number of candidate itemsets

3.6 CASE STUDY-TRANSACTIONS IN GROCERY STORE

- Retailing is an industry with high level of competition. It is a customer-based industry which depends on how it could be aware of what the customers' needs and requirements are. The very first groceries displayed their products in an industrial approach which have produced the present day grocery store layouts based on 'sectors' as fruits, vegetables, magazines, CDs, etc.
- This approach is company oriented and it fails to respond to the needs of the time-pressured consumer. In the new market, satisfying the customer needs is one of the important tasks for the retailers. This is required from company to move from the traditional store layout to new store layout which concern about the buying behaviour of the customer 'customer-oriented store layout'.
- The store layout and the promotional campaign are huge tasks for retail managers. The complexity of these tasks lies in the relationships between categories on sale as well as on the impact that it produces on the consumer spatial behaviour and in-store traffic.
- One possibility to do so is to make the store layout construction and the promotional campaign through the introduction of market basket analysis (Cil et al., 2009). Market basket analysis has the objective of individuating

products, or groups of products, that tend to occur together (are associated) in buying transactions (baskets).

- The knowledge obtained from a market basket analysis can be very valuable, and it can be employed by a supermarket to redesign the layout of the store to increase the profit through placing interdependencies products near to each other and to satisfy customers through saving time and personalized the store layout.
- Another strategy, items that are associated can be put near to each other; it increases the sales of other items due to complementarily effects. If the customers see them, it has higher probability that they will purchase them together.
- The knowledge obtained from a market basket analysis can also be used to improve the efficiency of a promotional campaign : products that are associated should not be put together.

Description of the Methodology

The methodology consists of six phases to extract the association rules from the transactional database using our scheme, starting with business understanding, data assembling, data preprocessing, model building (using the new scheme), post-processing of association rules and results interpretation. Those phases are explained in the following sections.

3.6.1 Business Understanding

- The concern of this stage is to identify the problem area and describe the problem in general terms. In another words, the enterprise decision makers need to formulate goals that the data mining process is expected to achieve.
- Then the first step in the methodology is to clearly defined business problem. The business analyst specifies the problem in the business.

3.6.2 Data Assembling

- Data mining required access to data. The data may be represented as volumes of records in several database files or the data may contain only few hundred records in a single file. To build effective model a data mining algorithm must be presented with thousands or millions of instances.
- Then the second step in our scheme is to collect the data which could come from many resources (OLAP, data warehouse, relational database and flat file).

3.6.3 Data Preprocessing

- Applying data preprocessing techniques before mining, can substantially enhance the overall quality of the patterns mined and/or the time required for the actual mining, low-quality data will lead to low-quality mining results.
- The preparation of data set is one of the most critical steps in a data mining process. This stage is concerned with selecting data, and mapping it.
- Selecting data refers for removing unnecessary information. When the data is drawn from different sources, it is possible that the same information is represented in different sources in different format. Mapping the data, it is the process of transfer the values of the selected variables. In our scheme, we are going to deal with numeric attribute. Using the numeric attributes will reduce the consumption of the memory.
- Therefore, we need to map or to eliminate the nominal attributes from the dataset. Consequently, this stage reconfigures the data to ensure consistent format, as there is possibility of inconsistent formats.

3.6.4 Model Building

- This stage is concerned with extraction of patterns for the data. The core of this research is mainly focused on model building. This phase concerns various view points and different aspects that should be given attention in order to yield sufficient results.

Input of Scheme :

- The set of the data (D)
- The support value (S)
- The confidence value (C)

Output of Scheme :

- Set of association rules

Step 1 : Compute the frequency of itemsets in D;

Compute the support (sup) of itemset, //where sup= frequency $(X \cup Y)/|D|$; If(sup(itemsets)>=S)

{Generate the frequent itemsets (FI);}

Step 2 : For each frequent itemsets (FI)

{Find the subset item sets x and y;}

Step 3 : Compute the confidence (conf) of x===>y, //where conf=frequency $(X \cup Y)$/frequency(X);

For each x and y

If(conf(x===>y) >=C)

{Compute the correlation (cor) of x===>y, /* where cor is one of three measurements

$$\text{correlation coefficient} = \sqrt{\frac{P(AB) - P(A)\,P(B)}{P(A)P(B)P(A)P(B)}}$$

$$\text{Cosine} = \frac{P(AB)}{P(A)\,P(B)}$$

$$\text{Interest} = \frac{P(AB)}{P(A)P(B)}$$

*/ Output x===>y {sup(x,y) , conf(x,y), cor(x,y)};}

3.6.5 Post-Processing of Association Rules

- This stage depends on the result of the previous ones. Actually it is about measure the interestingness of the items in the obtained model or pattern. Most likely the most significant problem with association rules, which so far remains largely unsettled, is the 'interestingness' of association rules. Indeed, the main strength of association rule mining is that, since it discovers all association rules that exist in a database, it can reveal valuable and unexpected information.
- These strengths, however, are also its weakness; i.e. the number of discovered rules can be huge, hundreds or even thousands of rules, which makes manual examination of those rules practically infeasible.
- In other words, association rule results sometimes create a new data mining problem of the second order.
- This makes post-processing of these rules very significant, i.e. we need good methods to reduce the number of association rules to the most interesting ones.
- The reasons for this problem of interestingness can be found in the limitations of the support-confidence framework, adopted by almost all.
- After learning system induces models from the data their evaluation should take place, there are several measurement for this purpose, support, confidence, correlation, cosine and interest.
- All objective measure of interestingness for association rules is based on the statistical notion of correlation between the items in the antecedent and the consequent of the rule.
- The idea is to construct a contingency table from the association rule results and test the interdependence between the antecedent and the consequent of the rule.
- We utilize three popular objective measurements which are correlation, cosine and interest (Tan et al., 2006) to see which is the most suitable for our data set. Therefore, a good strategy is to perform the correlation coefficient analysis first, and when the result shows that they are weakly positively/negatively correlated, other analyses can be performed to assist in obtaining a more complete picture.

3.6.6 Interpretation and Explanation of the Results

- The patterns obtained in the stage of model building and refined in the stage of post-processing of association rules are converted into knowledge, which in turn, is used to support the decision-making.
- After the rules are mined out of the database, the rules are used to understand the problem better. To summarize the obtain rules there are four methods which are : By ranking the rules by their supports value help the decision maker to know what the most ubiquitous rules are.
- By ranking the rules by their confidence value highly confidence value imply strong relationships. By summarizing all the rules that have certain value for consequent, it can be used to understand what is the associated with the consequent and perhaps what affects the consequent.
- Now we may use the acquired knowledge directly for predication or in an expert system shell as a knowledge base. If the knowledge discovery process is performed for an end-user, we usually document the derived results. Another possibility is to visualize the knowledge, or to transform it to an understandable form for the user-end. In this stage, the main concern is to summarize the obtain rules and present them to the decision makers.

3.7 VALIDATION AND TESTING

- Validation is the process of assessing how well your mining models perform against real data. It is important that you validate your mining models by understanding their quality and characteristics before you deploy them into a production environment.

3.7.1 Methods for Testing and Validation of Association Rules

- There are many approaches for assessing the quality and characteristics of a data mining model.
  - Use various measures of statistical validity to determine whether there are problems in the data or in the model.
  - Separate the data into training and testing sets to test the accuracy of predictions.
  - Ask business experts to review the results of the data mining model to determine whether the discovered patterns have meaning in the targeted business scenario
  - All of these methods are useful in data mining methodology and are used iteratively as you create, test, and refine models to answer a specific problem. No single comprehensive rule can tell you when a model is good enough, or when you have enough data.

3.7.2 Definition of Criteria for Validating Association Rules

- Measures of data mining generally fall into the categories of accuracy, reliability, and usefulness.
- Accuracy is a measure of how well the model correlates an outcome with the attributes in the data that has been provided. There are various measures of accuracy, but all measures of accuracy are dependent on the data that is used. In reality, values might be missing or approximate, or the data might have been changed by multiple processes.
- Particularly in the phase of exploration and development, you might decide to accept a certain amount of error in the data, especially if the data is fairly uniform in its characteristics.

- For example, a model that predicts sales for a particular store based on past sales can be strongly correlated and very accurate, even if that store consistently used the wrong accounting method. Therefore, measurements of accuracy must be balanced by assessments of reliability.
- Reliability assesses the way that a data mining model performs on different data sets. A data mining model is reliable if it generates the same type of predictions or finds the same general kinds of patterns regardless of the test data that is supplied. For example, the model that you generate for the store that used the wrong accounting method would not generalize well to other stores, and therefore would not be reliable.
- Usefulness includes various metrics that tell you whether the model provides useful information. For example, a data mining model that correlates store location with sales might be both accurate and reliable, but might not be useful, because you cannot generalize that result by adding more stores at the same location.
- Moreover, it does not answer the fundamental business question of why certain locations have more sales. You might also find that a model that appears successful in fact is meaningless, because it is based on cross-correlations in the data.

3.7.3 Tools for Testing and Validation of Mining Models

- Analysis Services supports multiple approaches to validation of data mining solutions, supporting all phases of the data mining test methodology.
  - Partitioning data into testing and training sets.
  - Filtering models to train and test different combinations of the same source data.
  - Measuring lift and gain. A lift chart is a method of visualizing the improvement that you get from using a data mining model, when you compare it to random guessing.
  - Performing cross-validation of data sets
  - Generating classification matrices. These charts sort good and bad guesses into a table so that you can quickly and easily gauge how accurately the model predicts the target value.
  - Creating scatter plots to assess the fit of a regression formula.
  - Creating profit charts that associate financial gain or costs with the use of a mining model, so that you can assess the value of the recommendations.
- These metrics do not aim to answer the question of whether the data mining model answers your business question; rather, these metrics provide objective measurements that you can use to assess the reliability of your data for predictive analytics, and to guide your decision of whether to use a particular iterate on the development process.

3.8 DIAGNOSTICS

- Even though the Apriori algorithm is considered as simple to understand and implement, few of the rules generated by this algorithm may not be interesting even or even may be practically useless.
- As such generation of some rules may be due to coincidental relationships which are between the variables.
- For the purpose of throwing light on this problem, one has to use the measures such as confidence, lift, and leverage.
- One more problem coupled with association rules is that, in Phase 3 as well as Phase 4 of the Data Analytics Lifecycle, the team has to mention the minimum support before the process of model execution, which may otherwise generate extremely more or extremely less rules.
- In the further research, it is possible for a variant of the algorithm to use a default target range for the number of rules.
- This will help the algorithm to adjust the minimum support accordingly.
- The Apriori algorithm is known to be one of the earliest and the most fundamental algorithms for the purpose of generating association rules.
- A Apriori algorithm helps to lower the computational workload by the process of examining itemsets which comply with the precise minimum threshold.
- On the other side the Apriori algorithm may be computationally expensive as it depends upon the size of the dataset.
- For each and every stage of support, the algorithm needs a process of scanning of the whole database to get the result.
- The time of computation in each run increase based on the increase in database.
- Following are some approaches to enhance Apriori's efficiency :

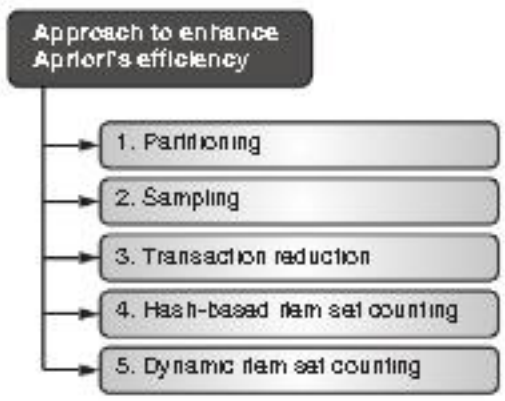


Fig. 3.3 : Approaches to enhance Apriori's efficiency

1. **Partitioning :**
   - It is a must for an item set to be frequent in minimum one partition of the transaction database if that item set is potentially frequent in a transaction database.
2. **Sampling :**
   - Sampling means to out a subset of the data which is having lower support threshold and takes the help of the subset to carry out association rule mining.
3. **Transaction Reduction :**
   - A transaction in which frequently remain abset k-itemsets is considered as useless in the process of subsequent scans and therefore can be ignored.
4. **Hash-Based Item Set Counting :**
   - If hashing bucket count of a k-itemset coming subsequently is lower than a specific threshold, then it is confirmed that the k-itemset cannot be frequent.
5. **Dyanmic Itemset Counting :**
   - The itemsets can be added only when all of the subsets of new candidate itemsets are anticipated to be frequent.

3.9 REGRESSION

- Regression analysis can imply a broader range of techniques which are useful for data analysis. Statisticians commonly define regression so that the goal is to understand "as far as possible with the available data how the conditional distribution of some response y varies across subpopulations determined by the possible values of the predictor or predictors".
- For example, if there is a single categorical predictor such as male or female, a legitimate regression analysis has been undertaken if one compares two income histograms, one for men and one for women.
- Or, one might compare summary statistics from the two income distributions : The mean incomes, the median incomes, the two standard deviations of income, and so on.
- One might also compare the shapes of the two distributions with a normal distribution. There is no requirement in regression analysis for there to be a "model" by which the data were supposed to be generated.
- There is no need to address cause and effect. And there is no need to undertake statistical tests or construct confidence intervals.
- The definition of a regression analysis can be met by pure description alone. Construction of a "model," often coupled with causal and statistical inference, are supplements to a regression analysis, not a necessary component.
- Given such a definition of regression analysis, a wide variety of techniques and approaches can be applied.
- There is a continuous random variable called the dependent variable, Y, and a number of independent variables, $x_1, x_2, \ldots, x_p$.
- Our purpose is to predict the value of the dependent variable (also referred to as the response variable) using a linear function of the independent variables.
- The values of the independent variables (also referred to as predictor variables, regressors or covariates) are known quantities for purposes of prediction.
- Let x be an instance and let y be its real–valued label. For linear regression, x must be a vector of real numbers of fixed length. Remember that this length p is often called the dimension, or dimensionality, of x. Write $x = (x_1, x_2, x_3, x_4 \ldots\ldots x_p)$. The linear regression model is

$$y = b_0 + b_1x_1 + b_2x_2 + \ldots\ldots\ldots + b_px_p :$$

- The righthand side above is called a linear function of x. The linear function is defined by its coefficients b_0 to b_p. These coefficients are the output of the data mining algorithm.
- The coefficient b_0 is called the intercept. It is the value of y predicted by the model if $x = 0$ for all i. Of course, it may be completely unrealistic that all features x have value zero. The coefficient bi is the amount by which the predicted y value increases if x_i increases by 1, if the value of all other features is unchanged. For example, suppose x_1 is a binary feature where $x_1 = 0$ means female and $x_1 = 1$ means male, and suppose $b_1 = -2.5$.
- Then the predicted y value for males is lower by 2.5, everything else being held constant. Suppose that the training set has cardinality n, i.e. it consists of n examples of the form (x_i, y_i), where $x_i = x_{i1}, \ldots\ldots x_{ip}$. Let b be any set of coefficients. The predicted value for x_i is

$$\hat{y}_i = f(x_i; b) = b_0 + \sum_{j=1}^{p} b_j x_j$$

- The semicolon in the expression f(x, b) emphasizes that the vector x_i is a variable input, while b is a fixed set of parameter values. If we define $x_{i0} = 1$ for every i, then we can write

$$\hat{y}_i = \sum_{j=0}^{p} b_j x_{ij}$$

The constant $x_{i0} = 1$ can be called a pseudo–feature.

- Finding the optimal values of the coefficients b_0 to b_p is the job of the training algorithm. To make this task well–defined, we need a definition of what "optimal" means. The standard approach is to say that optimal means minimizing the sum of squared errors on the training set, where the squared error on training example i is square of $(y_i - \hat{y}_i)$. The training algorithm then finds.

$$\hat{b} = \text{argmin}_b \sum_{i=1}^{n} (f(x_i, b) - y_i)^2$$

- The objective function $\sum_i (y_i - \sum_j b_j x_{ij})^2$ is called the sum of squared errors, or SSE for short. Note that during training the n different xi and yi values are fixed, while the parameters b are variable.
- The optimal coefficient values ^b are not defined uniquely if the number n of training examples is less than the number p of features.
- Even if n > p is true, the optimal coefficients have multiple equivalent values if some features are themselves related linearly. Here, "equivalent" means that the different sets of coefficients achieve the same minimum SSE.
- For an intuitive example, suppose features 1 and 2 are temperature measured in degrees Celsius and degrees Fahrenheit respectively.
- Then $x_2 = 32 + 9(x_1/5) = 32 + 1.8x_1$, and the same model can be written in many different ways :

$$y = b_0 + b_1x_1 + b_2x_2$$

$$y = b_0 + b_1x_1 + b_2(32 + 1.8 \times 1)$$

$$= [b_0 + 32b_2] + [b_1(1 + 1.8b_2)] \times 1 + 0 \times 2$$

and an infinite number of other ways. In the extreme, suppose $x_1 = x_2$. Then all models $y = b_0 + b_1x_1 + b_2x_2$ are equivalent for which $b_1 + b_2$ equals a constant.

- When two or more features are approximately related linearly, then the true values of the coefficients of those features are not well determined. The coefficients obtained by training will be strongly influenced by randomness in the training data.
- Regularization is a way to reduce the influence of this type of randomness. Consider all models $y = b_0 + b_1x_1 + b_2x_2$ for which $b_1 + b_2 = c$. Among these models, there is a unique one that minimizes the function $b_1^2 + b_2^2$. This model has $b_1 = b_2 = c/2$.
- We can obtain it by setting the objective function for training to be the Sum of Squared Errors (SSE) plus a function that penalizes large values of the coefficients. A simple penalty function of this type is $\sum_{j=1}^{p} b_j^2$. A parameter λ can control the relative importance of the two objectives, namely SSE and penalty :

$$\hat{b} = \text{argmin}_b \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \frac{1}{p} \sum_{j=1}^{p} b_j^2$$

If $\lambda = 0$ then one gets the standard least–squares linear regression solution. As λ increases, the penalty on large coefficients gets stronger, and the typical values of coefficients get smaller. The parameter λ is often called the strength of regularization.

- The fractions 1/n and 1/p do not make an essential difference. They can be used to make the numerical value of λ easier to interpret.

3.9.1 Linear Regression

- The penalty function $\sum_{j=1}^{p} b_j^2$ is the square of the L_2 norm of the vector b. Using it for linear regression is called ridge regression. Any penalty function that treats all coefficients bj equally, like the L_2 norm, is sensible only if the typical magnitudes of the values of each feature are similar; this is an important motivation for data normalization.
- Note that in the formula

$$\sum_{j=1}^{p} b_j^2$$

the sum excludes the intercept coefficient b_0. One reason for doing this is that the target y values are typically not normalized.

- Linear regression is a supervised machine learning technique which aims to build a learning model and tries to establish relationship between two variables by best fitting a linear line to input data. This line follows the equation of line :

$$y = mx + c$$

- Here y variable is considered to be a dependent or output variable explanatory variable, x is considered to be a independent variable and c represents intercept. y is a variable to be predicted and also known as criterion variable. Prediction of y is based on values of x hence x is called as predictor variable. Predictor variable could be one or more.
- When there is only one predictor variable, the prediction method is called simple regression and when there are more than one predictor variable, the prediction method is called multi regression.

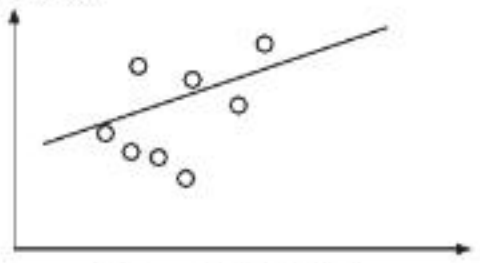


Fig. 3.4 : Linear regression model

- As prediction is core logic of regression, we are making use of regression technique to solve the problem of stock market movement prediction.

- Before using linear regression model for prediction one must determine if there exists a relationship between variable to be predicted/observed and input data. There must significant amount of association between input and output variable. Strength of relation between these two variables can be identified using scatter plot.
- Correlation coefficient is an important numerical measure which measures how strongly two variables are associated or correlated with each other. Correlation coefficient takes value between -1 and 1 indicating the strength of the association of the observed data for the two variables.
- Linear regression is a technique which consists of searching for the best-fitting straight line through the input data points. The best-fitting line is called a regression line. Once the regression line is modeled for group of data, data values which lie away from line are called as outliers.
- Outliers represent erroneous data and may indicate a poorly fitting regression line. With such erroneous data values accuracy of prediction can reduce to great extent. Outliers can have huge effects on the linear regression.
- Linear regression always searches for the linear relationship in the form of straight line between output variable and input variable/variables. Linear regression is always based on the assumption that there exists a linear relationship between output variable and input variable, which is not always true.
- Linear regression does not provide best fit for nonlinear data. This makes sense to consider Polynomial regression for prediction of stock movement.

3.9.2 Logistics Regression

- Here, we are going to deal with a dependent variable that is binary (a categorical variable that has two values such as "yes" and "no") rather than continuous.
- Logistic regression can also be applied to ordered categories (ordinal data), that is, variables with more than two ordered categories, such as what you find in many surveys. However, we won't be dealing with that in this course and you probably will never be taught it. If our dependent variable has several unordered categories (e.g., suppose our DV was state of origin in the U.S.), then we can use something called discriminant analysis, which will be taught to you in a course on multivariate statistics.
- It is customary to code a binary DV either 0 or 1. For example, we might code a successfully kicked field goal as 1 and a missed field goal as 0 or we might code yes as 1 and no as 0 or admitted as 1 and rejected as 0 or Cherry Garcia flavor ice cream as 1 and all other flavors as zero. If we code like this, then the mean of the distribution is equal to the proportion of 1s in the distribution.
- For example if there are 100 people in the distribution and 30 of them are coded 1, then the mean of the distribution is .30, which is the proportion of 1s. The mean of the distribution is also the probability of drawing a person labeled as 1 at random from the distribution. That is, if we grab a person at random from our sample of 100 that I just described, the probability that the person will be a 1 is .30.
- Therefore, proportion and probability of 1 are the same in such cases. The mean of a binary distribution so coded is denoted as P, the proportion of 1s. The proportion of zeros is (1-P), which is sometimes denoted as Q. The variance of such a distribution is PQ, and the standard deviation is Sqrt(PQ).
- Suppose we want to predict whether someone is male or female (DV, M=1, F=0) using height in inches (IV). We could plot the relations between the two variables as we customarily do in regression.

The Plot Might Look Something Like this :

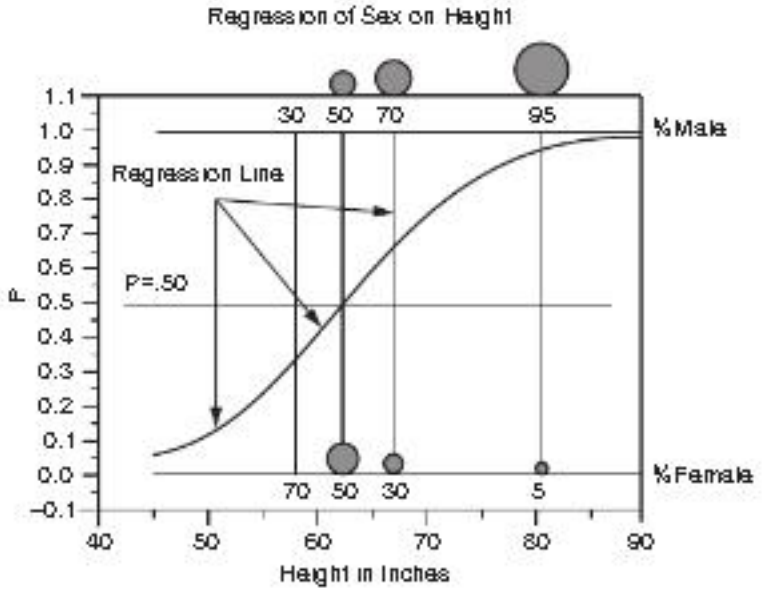


Fig. 3.5

Points to Notice about the Graph (Data are Fictional) :

- The regression line is a rolling average, just as in linear regression. The Y-axis is P, which indicates the proportion of 1s at any given value of height. (review graph)
- The regression line is nonlinear. (review graph)
- None of the observations --the raw data points-- actually fall on the regression line. They all fall on zero or one. (review graph)

Why use Logistic Regression rather than Ordinary Linear Regression?

Now a days, people didn't use logistic regression with a binary DV. They just used ordinary linear regression instead. Statisticians won the day, however, and now most psychologists use logistic regression with a binary DV for the following reasons :

- If you use linear regression, the predicted values will become greater than one and less than zero if you move far enough on the X-axis. Such values are theoretically inadmissible.
- One of the assumptions of regression is that the variance of Y is constant across values of X (homoscedasticity). This cannot be the case with a binary variable, because the variance is PQ. When 50 percent of the people are 1s, then the variance is .25, its maximum value. As we move to more extreme values, the variance decreases. When P=.10, the variance is .1*.9 = .09, so as P approaches 1 or zero, the variance approaches zero.
- The significance testing of the *b* weights rest upon the assumption that errors of prediction (Y-Y') are normally distributed. Because Y only takes the values 0 and 1, this assumption is pretty hard to justify, even approximately. Therefore, the tests of the regression weights are suspect if you use linear regression with a binary DV.

The Logistic Curve

- The logistic curve relates the independent variable, X, to the rolling mean of the DV, P ($\bar{Y}$). The formula to do so may be written either

$$P = \frac{e^{a+bX}}{1+e^{a+bX}}$$

Or

$$P = \frac{1}{1+e^{-(a+bX)}}$$

where P is the probability of a 1 (the proportion of 1s, the mean of Y), *e* is the base of the natural logarithm (about 2.718) and *a* and *b* are the parameters of the model. The value of *a* yields P when X is zero, and *b* adjusts how quickly the probability changes with changing X a single unit (we can have standardized and unstandardized b weights in logistic regression, just as in ordinary linear regression).

- Because the relation between X and P is nonlinear, *b* does not have a straightforward interpretation in this model as it does in ordinary linear regression.

Loss Function

- A loss function is a measure of fit between a mathematical model of data and the actual data. We choose the parameters of our model to minimize the badness-of-fit or to maximize the goodness-of-fit of the model to the data. With least squares (the only loss function we have used thus far), we minimize SS_{res} the sum of squares residual. This also happens to maximize SS_{reg} the sum of squares due to regression. With linear or curvilinear models, there is a mathematical solution to the problem that will minimize the sum of squares, that is,

$$b = (X'X)^{-1}X'y$$

Or

$$\beta = R^{-1}r$$

- With some models, like the logistic curve, there is no mathematical solution that will produce least squares estimates of the parameters. For many of these models, the loss function chosen is called maximum likelihood. A likelihood is a conditional probability (e.g., P(Y|X), the probability of Y given X). We can pick the parameters of the model (a and b of the logistic curve) at random or by trial-and-error and then compute the likelihood of the data given those parameters (actually, we do better than trail-and-error, but not perfectly).
- We will choose as our parameters, those that result in the greatest likelihood computed. The estimates are called maximum likelihood because the parameters are chosen to maximize the likelihood (conditional probability of the data given parameter estimates) of the sample data.
- The techniques actually employed to find the maximum likelihood estimates fall under the general label numerical analysis. There are several methods of numerical analysis, but they all follow a similar series of steps. First, the computer picks some initial estimates of the parameters.
- Then it will compute the likelihood of the data given these parameter estimates. Then it will improve the parameter estimates slightly and recalculate the likelihood of the data. It will do this forever until we tell it to stop, which we usually do when the parameter estimates do not change much (usually a change .01 or .001 is small enough to tell the computer to stop).

Where on Earth Did This Stuff Come From?

Suppose we only know a person's height and we want to predict whether that person is male or female. We can talk about the probability of being male or female, or we can talk about the odds of being male or female. Let's say that the probability of being male at a given height is .90. Then the odds of being male would be

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

In our example, the odds would be .90/.10 or 9 to one. Now the odds of being female would be .10/.90 or 1/9 or .11. This asymmetry is unappealing, because the odds of being a male should be the opposite of the odds of being a female. We can take care of this asymmetry though the natural logarithm, *ln*. The natural log of 9 is 2.217 (*ln*(.9/.1)= 2.217). The natural log of 1/9 is -2.217 (*ln*(.1/.9)=-2.217), so the log odds of being male is exactly opposite to the log odds of being female.

The Natural Log Function Looks like this :

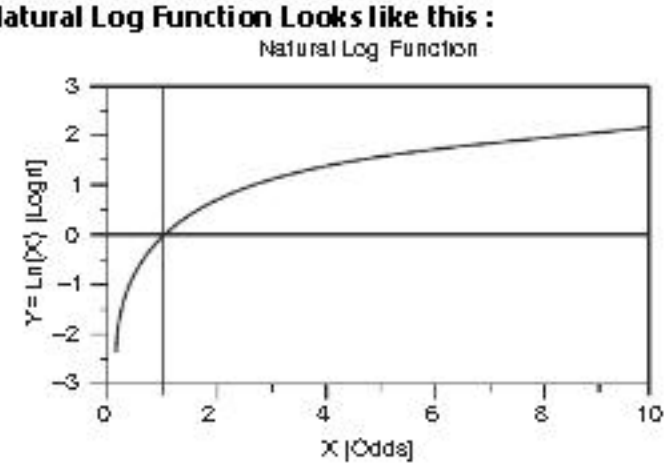


Fig. 3.6

Note that the natural log is zero when X is 1. When X is larger than one, the log curves up slowly. When X is less than one, the natural log is less than zero, and decreases rapidly as X approaches zero. When P = .50, the odds are .50/.50 or 1, and *ln*(1) =0. If P is greater than .50, *ln*(P/(1-P) is positive; if P is less than .50, *ln*(odds) is negative. [A number taken to a negative power is one divided by that number, e.g. $e^{-10} = 1/e^{10}$. A logarithm is an exponent from a given base, for example $ln(e^{10}) = 10$.]

Back to Logistic Regression.

In logistic regression, the dependent variable is a logit, which is the natural log of the odds, that is,

$$\log(\text{odds}) = \text{logit}(P) = ln\left(\frac{P}{1-P}\right)$$

So a logit is a log of odds and odds are a function of P, the probability of a 1. In logistic regression, we find

$$\text{logit}(P) = a + bX,$$

Which is assumed to be linear, that is, the log odds (logit) is assumed to be linearly related to X, our IV. So there's an ordinary regression hidden in there. We could in theory do ordinary regression with logits as our DV, but of course, we don't have logits in there, we have 1s and 0s. Then, too, people have a hard time understanding logits. We could talk about odds instead. Of course, people like to talk about probabilities more than odds. To get there (from logits to probabilities), we first have to take the log out of both sides of the equation. Then we have to convert odds to a simple probability :

$$ln\left(\frac{P}{1-P}\right) = a + bX$$

$$\frac{P}{1-P} = e^{a+bX}$$

$$P = \frac{e^{a+bX}}{1+e^{a+bX}}$$

The simple probability is this ugly equation that you saw earlier. If log odds are linearly related to X, then the relation between X and P is nonlinear, and has the form of the S-shaped curve you saw in the graph and the function form (equation) shown immediately above.

An Example

Suppose that we are working with some doctors on heart attack patients. The dependent variable is whether the patient has had a second heart attack within 1 year (yes = 1). We have two independent variables, one is whether the patient completed a treatment consistent of anger control practices (yes=1). The other IV is a score on a trait anxiety scale (a higher score means more anxious).

Our Data :

Person	2nd Heart Attack	Treatment of Anger	Trait Anxiety
1	1	1	70
2	1	1	80
3	1	1	50
4	1	0	60
5	1	0	40
6	1	0	65
7	1	0	75
8	1	0	80
9	1	0	70
10	1	0	60
11	0	1	65
12	0	1	50
13	0	1	45
14	0	1	35
15	0	1	40
16	0	1	50
17	0	0	55
18	0	0	45
19	0	0	50
20	0	0	60

Our Correlation Matrix :

	Heart	Treat	Anx
Heart	1		
Treat	-.30	1	
Anx	.59**	-.23	1
Mean	.50	.45	57.25
SD	.51	.51	13.42

Note that half of our patients have had a second heart attack. Knowing nothing else about a patient, and following the best in current medical practice, we would flip a coin to predict whether they will have a second attack within 1 year. According to our correlation coefficients, those in the anger treatment group are less likely to have another attack, but the result is not significant. Greater anxiety is associated with a higher probability of another attack, and the result is significant (according to *r*).

Now let's look at the logistic regression, for the moment examining the treatment of anger by itself, ignoring the anxiety test scores. SAS prints this :

Response Variable : HEART

Response Levels : 2

Number of Observations : 20

Link Function : Logit

Response Profile

Ordered

Value HEART Count

1 0 10

2 1 10

SAS tells us what it understands us to model, including the name of the DV, and its distribution.

Then we calculate probabilities with and without including the treatment variable.

Model Fitting Information and Testing Global Null Hypothesis BETA= 0

Criterion Intercept Intercept Chi-sq

Only and

Covariates

-2 LOG L 27.726 25.878 1.848

1df (p=.17)

- The computer calculates the likelihood of the data. Because there are equal numbers of people in the two groups, the probability of group membership initially (without considering anger treatment) is .50 for each person. Because the people are independent, the probability of the entire set of people is $.50^{20}$, a very small number.
- Because the number is so small, it is customary to first take the natural log of the probability and then multiply the result by -2. The latter step makes the result positive. The statistic -2LogL (minus 2 times the log of the likelihood) is a badness-of-fit indicator, that is, large numbers mean poor fit of the model to the data. SAS prints the result as -2 LOG L.
- For the initial model (intercept only), our result is the value 27.726. This is a baseline number indicating model fit. This number has no direct analog in linear regression. It is roughly analogous to generating some random numbers and finding R^2 for these numbers as a baseline measure of fit in ordinary linear regression.
- By including a term for treatment, the loss function reduces to 25.878, a difference of 1.848, shown in the chi-square column. The difference between the two values of -2LogL is known as the likelihood ratio test.
- When taken from large samples, the difference between two values of -2LogL is distributed as chi-square :

$$\chi^2 = LL_R - (-2\,LL_B) = -2\,ln\left(\frac{likehood_R}{likelihood_B}\right)$$

- Recall that multiplying numbers is equivalent to adding exponents (same for subtraction and division of logs).
- This says that the (-2Log L) for a restricted (smaller) model - (-2LogL) for a full (larger) model is the same as the log of the ratio of two likelihoods, which is distributed as chi-square. The full or larger model has all the parameters of interest in it. The restricted is said to be *nested* in the larger model.
- The restricted model has one or more of parameters in the full model restricted to some value (usually zero). The parameters in the nested model must be a proper subset of the parameters in the full model. For example, suppose we have two IVs, one categorical and once continuous, and we are looking at an ATI design.
- A full model could have included terms for the continuous variable, the categorical variable and their interaction (3 terms). Restricted models could delete the interaction or one or more main effects (e.g., we could have a model with only the categorical variable). A nested model cannot have as a single IV, some other categorical or continuous variable not contained in the full model.

- If it does, then it is no longer nested, and we cannot compare the two values of -2LogL to get a chi-square value. The chi-square is used to statistically test whether including a variable reduces badness-of-fit measure. This is analogous to producing an increment in R-square in hierarchical regression. If chi-square is significant, the variable is considered to be a significant predictor in the equation, analogous to the significance of the *b* weight in simultaneous regression.

For our example with anger treatment only, SAS produces the following :

Analysis of Maximum Likelihood Estimates

Variable	DF	Par Est	Std Err	Wald Chisq	Pr > Chi-sq	Stand . Est	Odds Ratio
Intercept	1	-.5596	.6268	.7972	.3719	.	.
Treatment	1	1.2528	.9449	17566	.1849	.3525	3.50

- The intercept is the value of *a*, in this case -.5596. As usual, we are not terribly interested in whether a is equal to zero. The value of *b* given for Anger Treatment is 1.2528. The chi-square associated with this *b* is not significant, just as the chi-square for covariates was not significant. Therefore we cannot reject the hypothesis that *b* is zero in the population. Our equation can be written either :

$$\text{Logit}(P) = -.5596 + 1.2528X$$

Or

$$P = \frac{1}{1 + e^{-(-.5596 + 1.2528X)}}$$

The main interpretation of logistic regression results is to find the significant predictors of Y. However, other things can sometimes be done with the results.

The Odds Ratio

Recall that the odds for a group is :

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

Now the odds for another group would also be P/(1-P) for that group. Suppose we arrange our data in the following way :

	Anger Treatment		
Heart Attack	**Yes (1)**	**No (0)**	**Total**
Yes (1)	3 (a)	7 (b)	10 (a+b)
No (0)	6 (c)	4 (d)	10 (c+d)
Total	9 (a+c)	11 (b+d)	20 (a+b+c+d)

- Now we can compute the odds of having a heart attack for the treatment group and the no treatment group. For the treatment group, the odds are 3/6 = 1/2.
- The probability of a heart attack is 3/(3+6) = 3/9 = .33. The odds from this probability are .33/(1-.33) = .33/.66 = 1/2. The odds for the no treatment group are 7/4 or 1.75. The odds ratio is calculated to compare the odds across groups.

$$\text{OR} = \frac{a/c}{b/d} = \frac{ad}{bc}$$

- If the odds are the same across groups, the odds ratio (OR) will be 1.0. If not, the OR will be larger or smaller than one. People like to see the ratio be phrased in the larger direction. In our case, this would be 1.75/.5 or 1.75*2 = 3.50.
- Now if we go back up to the last column of the printout where is says odds ratio in the treatment column, you will see that the odds ratio is 3.50, which is what we got by finding the odds ratio for the odds from the two treatment conditions. It also happens that $e^{1.2528} = 3.50$. Note that the exponent is our value of *b* for the logistic curve.

3.9.3 Reasons to Choose and Cautions

- Linear regression is considered to be appropriate when the input variables are in continuous or discrete form with categorical data types, while the outcome variable is continuous.
- Logistic regression is suitable when the outcome variable is categorical.
- In both the modules, linear additive function of the input variables is taken into consideration.
- The performance of both regression techniques will be poor in case the above assumptions do not hold true.
- In addition to the linear regression, the supposition of usually distributed error terms with a constant variance is considered as significant for number of the statistical interferences which can be taken into account.
- If it is found that the several assumptions that we hold, it is necessary to apply appropriate transformations on that data.
- Even if a set of input variables is a good forecaster for the outcome variable, the analyst should not suppose that the input variables straightforwardly result in an outcome.
- E.g. it might be recognized that the people who have usual dentist visits may have a less chances of heart attacks.
- Although, only sending an individual to the dentist almost not sure to has no impact on the person's possibility of having a heart attack.
- Frequent dentist visits may point to a person's overall health as well as dietary choices, which may show direct effect on a person's health.
- The above example describes the usually known expression, "Correlation does not imply causation."

- One has to take care while applying previously fitted model to data which falls outside the dataset used to train the model.
- In a regression model, it may be possible that linear relationship may no longer hold at values outside the training dataset.
- In a linear regression model, the state of residence gives an easy additive term to the income model; however there will not be any effect on the coefficient of the other input variables, like Age and Education.
- On the other hand, if state makes impact on the other variable's effect to the income model, substitute approach will generate fifty distinct linear regression models : one model for each state.
- When lot of input variables are highly associated with to each other, it cause to condition known as multicollinearity.
- The multicollinearity may usually generate coefficient estimates which are comparatively big in absolute magnitude and might be of wrong direction with –ve or +ve sign.
- Whenever possible, it is necessary to remove majority of these associated variables from the model or replaced by a new variable.
- For example in a regression application related to medical field, height and weight may are important input variables, but it may be possible that these variables tend to be correlated.

3.9.4 Additional Regression Models

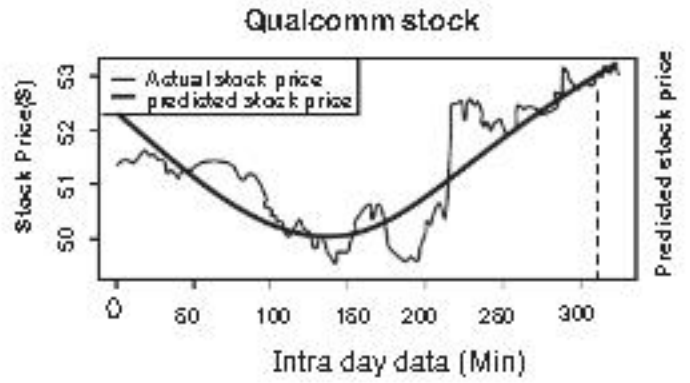
1. Polynomial Regression

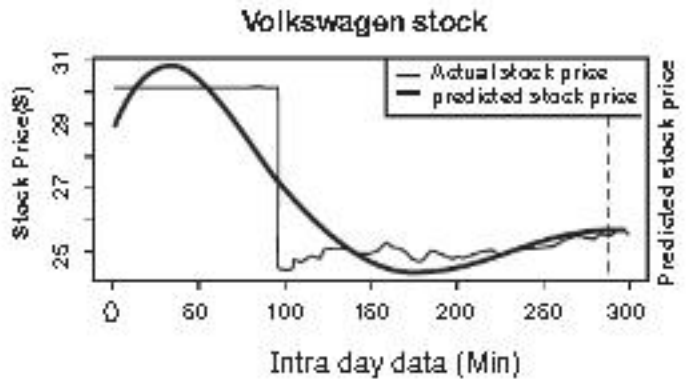
- Sometimes, a graph of the independent versus a dependent variable may suggest there is a nonlinear relationship. Such kind of relationship can be understood with the help of a polynomial regression model. Polynomial regression model for a single variable can be given as :

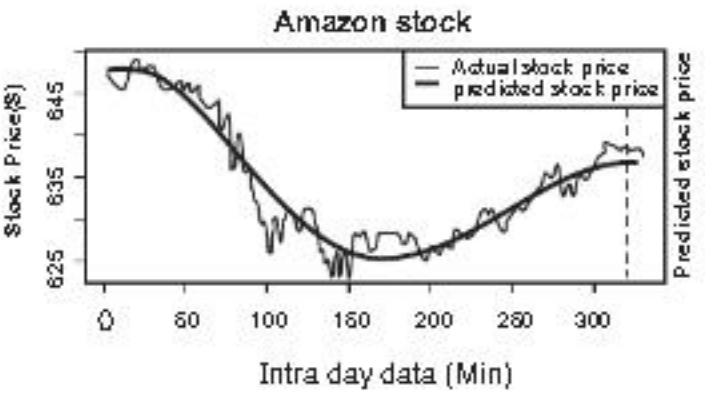
$$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m,\ m < n$$

where, m is called the degree of the polynomial. Although polynomial regression represents nonlinear behavior, still it is considered as linear regression with regression coefficients $a_0, a_1 \ldots a_m$.

- Great thing about polynomial regression is that there could be more than one independent variable and these variables may need to have interaction with them in order to predict dependent variable Y.







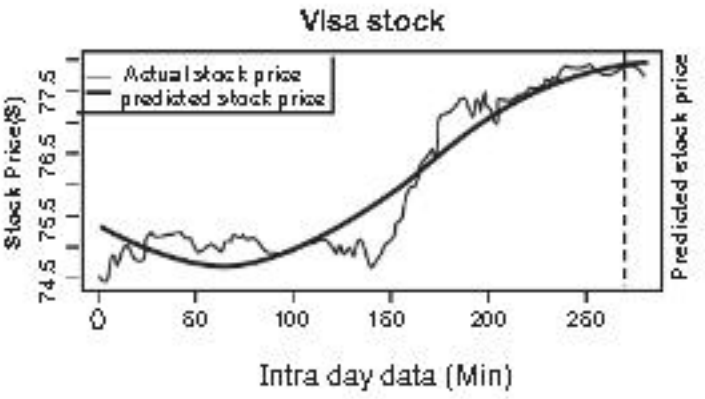


Fig. 3.7 : Polynomial regression for prediction of stock price movement

- With experimental work performed here in the context of polynomial regression, modeler has to keep mind general principles otherwise polynomial regression may produce meaningless results beyond the scope of the model. These general principles are given here.
  - The fitted curve is more reliable only if model is trained with data of size much larger.

- Do not extrapolate beyond confines of observed values otherwise due to nonlinear nature, polynomial regression may produce meaningless results beyond the scope of the model.
- Values of independent variables must not be too large causing overflow for high degree polynomials. Therefore input variables X need to scale down.

2. Support Vector Regression (SVR)

- A lot of study has been done over applying support vector machine for stock market analysis. However support vector machine solves binary classification problem whereas SVR acknowledges the presence of non-linearity in the data and provides a proficient prediction model.
- SVR uses the same basic idea as Support Vector Machine (SVM), a classification algorithm, but applies it to predict real values rather than a class.

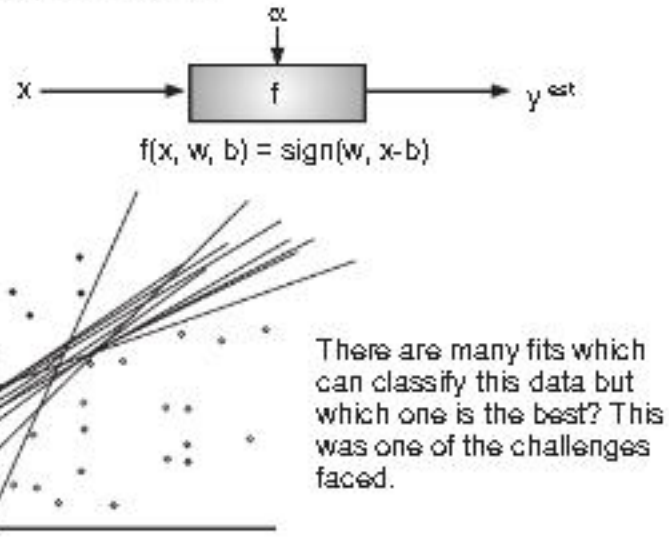


Fig. 3.8 : Number of hyperplanes possible for classification

- Support vector machine is specialized supervised machine learning technique which not only forms a hyper-plane separating two classes but it identifies a just right hyper-plane which segregates two classes better. SVM identifies only one hyper-plane that maximizes the margin.
- In SVM, kernel functions are set of mathematical functions that take data as input and transform this input from one form to the required form. Kernel function basically performs mapping of low dimensional input space to a higher dimensional.
- There are different kernel functions like linear, nonlinear, polynomial, Radial Basis Function (RBF) which can be used with different SVM algorithms. Kernel function transforms extremely complex data and finds out the process to classify the data based on the predefined labels or classes.

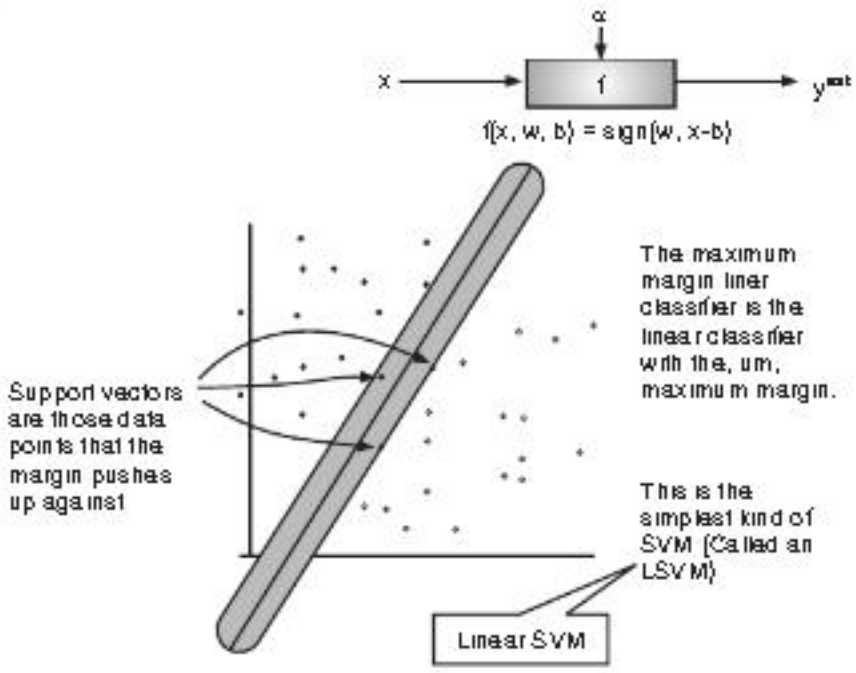


Fig. 3.9 : Support vector machine

Following are Some Commonly used Kernel Functions :

- **Polynomial :** A polynomial kernel is a popular function for non-linear modeling. Polynomial kernels are typically used in image processing applications.

$$k(x,x') = (xx')^d$$

$$k(x,x') = (xx' + 1)^d$$

Here d is known as degree of polynomial function.

- **Gaussian Radial Basis Function (GRBF) :** GRBF are general purpose kernel functions and most commonly used with a Gaussian form. GRBP are particularly used when there no prior knowledge about the data.

$$k(x,x') = \exp\left(\frac{\|(x-x')^2\|}{2\sigma^2}\right)$$

- **Exponential Radial Basis Function :** When there are discontinuities in the input data space or when data points are discrete in nature, a radial basis function produces a separate linear solution and makes these discontinuities to be acceptable.

$$k(x,x') = \exp\left(\frac{\|(x-x')^2\|}{2\sigma^2}\right)$$

- **Multi-Layer Perceptron (MLP) :** MLP is based on Neural Networks with a single hidden layer which can also be used to represent kernel function.

$$k(x,x') = \tanh(p(x,x') + \sigma)$$

EXERCISE

1. Write short note on market basket analysis problem
2. What do you mean by frequent item set ? How association rules can be generated from frequent item sets?
3. What is regression? Explain linear regression and logistic regression with suitable example.
4. Differentiate between support vector machine and regression.
5. Explain the apriori algorithm for generation of association rules ? How candidate keys are generated in apriori algorithm

UNIT IV

CLASSIFICATION

4.1 INTRODUCTION

- Classification is a task of assigning an object to a certain class based on its similarity to previous examples of other objects. Classification can be done with reference to original data or it is based on a model of that data. Certainty is a factor in classification. As with most data mining solutions, a classification usually comes with a degree of certainty.
- It might be the probability of the object belonging to the class or it might be some other measure of how closely the object resembles other examples from that class.
- Classification consists of predicting a certain outcome based on a given input. In order to predict the outcome, the algorithm processes a training set containing a set of attributes and the respective outcome, usually called goal or prediction attribute.
- The algorithm tries to discover relationships between the attributes that would make it possible to predict the outcome. Next the algorithm is given a data set not seen before, called prediction set or testing dataset, which contains the same set of attributes, except for the prediction attribute – not yet known. The algorithm analyses the input and produces a prediction.
- The prediction accuracy defines how "good" the algorithm is. For example, in a medical database, the training set would have relevant patient information recorded previously, where the prediction attribute is whether or not the patient had a heart problem. Table 4.1 below illustrates the training and prediction sets of such database.

Table 4.1 : Training Set

Age	Heart Rate	Blood Pressure	Heart Problem
65	78	150/70	Yes
37	83	112/76	No
71	67	108/65	No

Table 4.2 : Prediction Set

Age	Heart Rate	Blood Pressure	Heart Problem
43	98	147/89	?
65	58	106/63	?
84	77	150/65	?

- Among several types of knowledge representation present in the literature, classification normally uses prediction rules to express knowledge. Prediction rules are expressed in the form of IF–THEN rules, where the antecedent (IF part) consists of a conjunction of conditions and the rule consequent (then part) predicts a certain predictions attribute value for an item that satisfies the antecedent. Using the example above, a rule predicting the first row in the training set may be represented as following : IF (Age=65 AND Heart rate>70) OR (Age>60 AND Blood pressure>140/70) THEN Heart problem=yes In most cases the prediction rule is immensely larger than the example above.
- Conjunction has a good property for classification; each condition separated by OR's defines smaller rules that captures relations between attributes. Satisfying any of these smaller rules means that the consequent is the prediction. Each smaller rule is formed with AND's which facilitates narrowing down relations between attributes. How well predictions are done is measured in percentage of predictions hit against the total number of predictions.
- A decent rule ought to have a hit rate greater than the occurrence of the prediction attribute. In other words, if the algorithm is trying to predict rain in Seattle and it rains 80% of the time, the algorithm could easily have a hit rate of 80% by just predicting rain all the time. Therefore, 80% is the base prediction rate that any algorithm should achieve in this case. The optimal solution is a rule with 100% prediction hit rate, which is very hard, when not impossible, to achieve.
- Therefore, except for some very specific problems, classification by definition can only be solved by approximation algorithms.

4.2 CLASSIFICATION REQUIREMENT

- Classification is a two step process. In the first model is constructed using training data, and in the second test model built is tested using testing dataset. The difference between training and testing dataset is that final classes in which objects or instances to be classified are already known whereas in testing dataset the final classes are unknown.
- So first classification model is built or trained using training dataset, and then it is evaluated using testing dataset. Detail description of both the steps is given below.

Classification Process

Step 1 : Model Construction

- Model construction starts with describing a set of predetermined classes. predetermined classes are the classes

which are previously known Each record is assumed to belong to a predefined class, as determined by the class label attribute.

- The set of records used for model construction is called as training set. The classification model is represented by classification rules, decision trees, or some mathematical formulae.

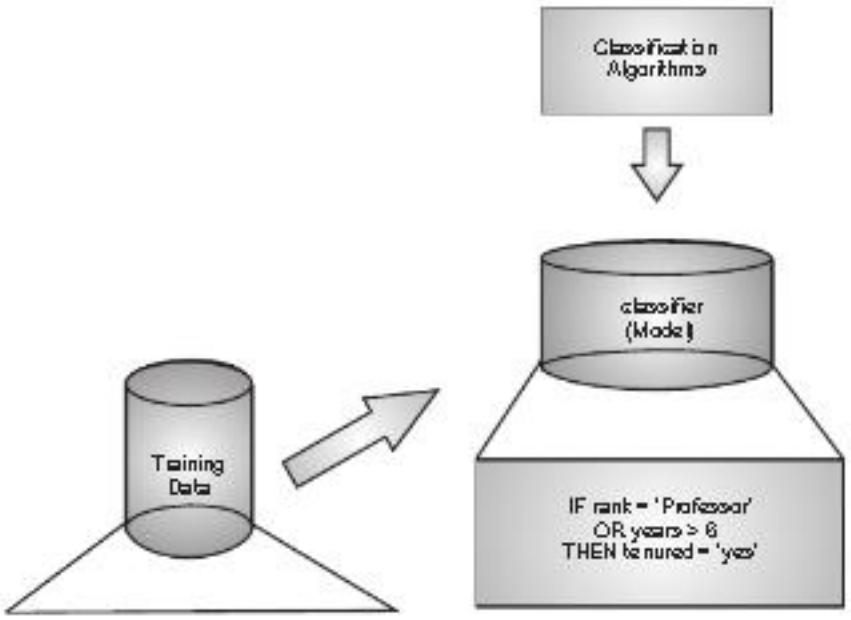


Fig. 4.1

Name	Rank	Years	Tenured
Shweta	Assistant Prof.	3	No
Nitin	Assistant Prof.	7	Yes
Subhash	Professor	2	Yes
Chetan	Associate Prof.	7	Yes
Mayuresh	Assistant Prof.	6	No
Swati	Associate Prof.	3	No

Step 2 : Model Usage

- Classification model is then used for classifying future or unknown objects. The set of these unknown objects is known as testing dataset. Classes are not predetermined but it is the responsibility of the model to accurately classify each record/object in to respective class.
- Estimating the accuracy of the model is important. The known label of test sample is compared with the classified result from the model.
- Accuracy rate is the percentage of test set samples that are correctly classified by the model. There are number of metrics which are used to estimate the classification accuracy which we will discuss in section.
- Also remember that test set is independent of training set.

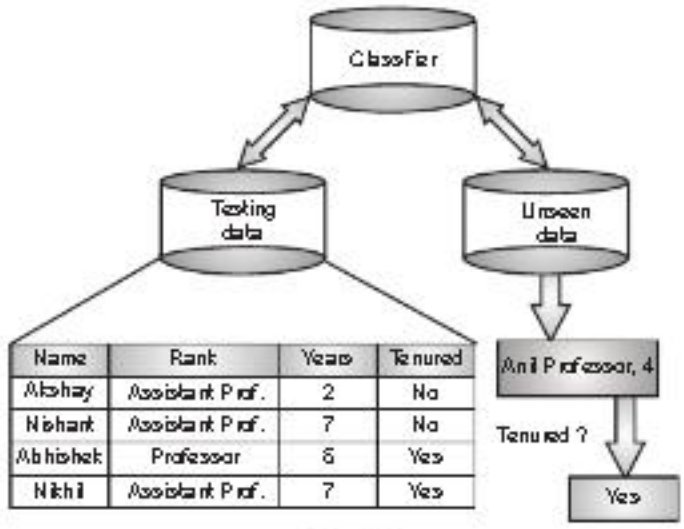


Fig. 4.2

- Classification is a classical problem extensively studied by statisticians and machine learning researchers.

Following are Some Issues Need to be Addressed by Classification Technique :

- **Speed and Scalability :** Classifying data sets with millions of examples and hundreds of attributes with reasonable speed.
- **Predictive Accuracy :** Accuracy of prediction, also called as classification accuracy should be as high as possible.
- **Time for Construction :** Time required to construct the model should be less.
- **Time for Usage :** Time required to use the model for classification of unknown records should be less.
- **Robustness :** Classification model should be able to perform some data preprocessing steps on its own. It should be robust in handling noise and missing values.
- **Interpretability :** Rules formed by classification model should be easy to understand, they should be compact and their correctness should be high.

4.3 SUPERVISED LEARNING

- Supervised learning is the kind of learning where the training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations and new data is classified based on the training set. Final class labels are known before hand in supervised learning.
- Typical example of supervised learning is classification technique. In Unsupervised learning (clustering), the class labels of training data are unknown.
- Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data. Typical example of unsupervised learning is clustering.
- **Supervised Learning** is the machine learning task of building a function from supervised training data. The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal).

- A supervised learning algorithm analyzes the training data and produces an inferred function, which is called a classifier or classification model.
- The inferred function should predict the correct output value for any valid input object. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way.
- Supervised machine learning is the search for algorithms that reason from externally supplied instances to produce general hypotheses, which then make predictions about future instances.
- In other words, the goal of supervised learning is to build a concise model of the distribution of class labels in terms of predictor features. The resulting classifier is then used to assign class labels to the testing instances where the values of the predictor features are known, but the value of the class label is unknown.

4.3.1 Supervised Learning Overview

In Order to Solve a Given Problem of Supervised Learning, One has to Perform the Following Steps

1. **Determine the Type of Training Examples :**

   Before doing anything else, the engineer should decide what kind of is to be used as an example. For instance, this might be a single handwritten character, an entire handwritten word, or an entire line of handwriting.

2. **Gather a Training Set :**

   The training set needs to be representative of the real–world use of the function. Thus, a set of input objects is gathered and corresponding outputs are also gathered, either from human experts or from measurements.

3. **Determine the Input Feature Representation of the Learned Function :**
   - The accuracy of the learned function depends strongly on how the input object is represented. Typically, the input object is transformed into a feature vector, which contains a number of features that are descriptive of the object. The number of features should not be too large, because of the curse of dimensionality; but should contain enough information to accurately predict the output.
   - Determine the structure of the learned function and corresponding learning algorithm.
   - For example, the user may choose to use support vector machines or decision trees.

4. **Complete the Design :**
   - Run the learning algorithm on the gathered training set. Some supervised learning algorithms require the user to determine certain control parameters. These parameters may be adjusted by optimizing performance on a subset (called a validation set) of the training set, or via cross–validation.

5. **Evaluate the Accuracy of the Learned Function**
   - After parameter adjustment and learning, the performance of the resulting function should be measured on a test set that is separate from the training set. A wide range of supervised learning algorithms is available, each with its strengths and weaknesses There is no single learning algorithm that works best on all supervised learning problems.

There are Four Major Issues to Consider in Supervised Learning :

1. **Bias–Variance Tradeoff**
   - A first issue is the tradeoff between bias and variance. Imagine that we have available several different, but equally good, training data sets. A learning algorithm is biased for a particular input x if, when trained on each of these data sets, it is systematically incorrect when predicting the correct output for x.
   - A learning algorithm has high variance for a particular input x if it predicts different output values when trained on different training sets. The prediction error of a learned classifier is related to the sum of the bias and the variance of the learning algorithm.
   - Generally, there is a tradeoff between bias and variance. A learning algorithm with low bias must be "flexible" so that it can fit the data well. But if the learning algorithm is too flexible, it will fit each training data set differently, and hence have high variance.
   - A key aspect of many supervised learning methods is that they are able to adjust this tradeoff between bias and variance (either automatically or by providing a bias/variance parameter that the user can adjust).

2. **Function Complexity and Amount of Training Data**
   - The second issue is the amount of training data available relative to the complexity of the "true" function (classifier or regression function). If the true function is simple, then an "inflexible" learning algorithm with high bias and low variance will be able to learn it from a small amount of data.

- But if the true function is highly complex (e.g., because it involves complex interactions among many different input features and behaves differently in different parts of the input space), then the function will only be learnable from a very large amount of training data and using a "flexible" learning algorithm with low bias and high variance.
- Good learning algorithms therefore automatically adjust the bias/variance tradeoff based on the amount of data available and the apparent complexity of the function to be learned.

3. Dimensionality of the Input Space

- A third issue is the dimensionality of the input space. If the input feature vectors have very high dimension, the learning problem can be difficult even if the true function only depends on a small number of those features.
- This is because the many "extra" dimensions can confuse the learning algorithm and cause it to have high variance. Hence, high input dimensionality typically requires tuning the classifier to have low variance and high bias.
- In practice, if the engineer can manually remove irrelevant features from the input data, this is likely to improve the accuracy of the learned function. In addition, there are many algorithms for feature selection that seek to identify the relevant features and discard the irrelevant ones.
- This is an instance of the more general strategy of dimensionality reduction, which seeks to map the input data into a lower dimensional space prior to running the supervised learning algorithm.

4. Noise in the Output Values

- A fourth issue is the degree of noise in the desired output values (the supervisory targets). If the desired output values are often incorrect (because of human error or sensor errors), then the learning algorithm should not attempt to find a function that exactly matches the training examples.
- This is another case where it is usually best to employ a high bias, low variance classifier.

4.3.2 Other Issues in Supervised Learning

1. Heterogeneity of the Data

- If the feature vectors include features of many different kinds (discrete, discrete ordered, counts, continuous values), some algorithms are easier to apply than others.
- Many algorithms, including Support Vector Machines, linear regression, logistic regression, neural networks, and nearest neighbor methods, require that the input features be numerical and scaled to similar ranges (e.g., to the [–1,1] interval).
- Methods that employ a distance function, such as nearest neighbor methods and support vector machines with Gaussian kernels, are particularly sensitive to this. An advantage of decision trees is that they easily handle heterogeneous data.

2. Redundancy in the Data.

- If the input features contain redundant information (e.g., highly correlated features), some learning algorithms (e.g., linear regression, logistic regression, and distance based methods) will perform poorly because of numerical instabilities.
- These problems can often be solved by imposing some form of regularization.

3. Presence of Interactions and Non–Linearities.

- If each of the features makes an independent contribution to the output, then algorithms based on linear functions (e.g., linear regression, logistic regression, Support Vector Machines, naive Bayes) and distance functions (e.g., nearest neighbor methods, support vector machines with Gaussian kernels) generally perform well.
- However, if there are complex interactions among features, then algorithms such as decision trees and neural networks work better, because they are specifically designed to discover these interactions. Linear methods can also be applied, but the engineer must manually specify the interactions when using them.

4.3.3 Generalizations of Supervised Learning

There are several ways in which the standard supervised learning problem can be generalized :

1. Semi–Supervised Learning :

In this setting, the desired output values are provided only for a subset of the training data. The remaining data is unlabeled.

2. Active Learning :

Instead of assuming that all of the training examples are given at the start, active learning algorithms interactively collect new examples, typically by making queries to a human user. Often, the queries are based on unlabeled data, which is a scenario that combines semi–supervised learning with active learning.

3. Structured Prediction :

When the desired output value is a complex object, such as a parse tree or a labeled graph, then standard methods must be extended.

4. Learning to Rank :

When the input is a set of objects and the desired output is a ranking of those objects, then again the standard methods must be extended.

4.4 DECISION TREES

- Classification of data objects based on a predefined knowledge of objects is data mining. There are many classification algorithms available but decision tree is the most commonly used. Decision Tree learning is one of the most widely used and practical methods for inductive inference over supervised data.
- A decision tree represents a procedure for classifying categorical data based on their attributes. It is also efficient for processing large amount of data, so is often used in data mining application.
- The construction of decision tree does not require any domain knowledge or parameter setting, and therefore appropriate for exploratory knowledge discovery.
- Their representation of acquired knowledge in tree form is intuitive and easy to assimilate by humans.
- Decision tree is one of the classification technique used in decision support system and machine learning process. A decision tree is a predictive modeling technique that is used in classification, clustering and predictive task.
- Decision tree uses a divide and conquer technique to split the problem search space into subsets. The most important feature of decision tree classifier is their ability to break down a complex decision making process into collection of simpler decision, thus providing solution which is easier to interpret.
- A decision tree is a classifier expressed as a recursive partition of the instance space. The decision tree consists of nodes that form a rooted tree, meaning it is a directed tree with a node called "root" that has no incoming edges. All other nodes have exactly one incoming edge.
- A node with outgoing edges is called an internal or test node. All other nodes are called leaves (also known as terminal or decision nodes). In a decision tree, each internal node splits the instance space into two or more sub–spaces according to a certain discrete function of the input attributes values.
- In the simplest and most frequent case, each test considers a single attribute, such that the instance space is partitioned according to the attribute's value. In the case of numeric attributes, the condition refers to a range.
- Each leaf is assigned to one class representing the most appropriate target value. Alternatively, the leaf may hold a probability vector indicating the probability of the target attribute having a certain value. Instances are classified by navigating them from the root of the tree down to a leaf, according to the outcome of the tests along the path.
- Fig. 4.3 describes a decision tree that reasons whether or not a person is suffering from fever. Internal nodes are represented as circles, whereas leaves are denoted as triangles.
- Note that this decision tree incorporates both nominal and numeric attributes. Given this classifier, the analyst can predict the response of a potential customer (by sorting it down the tree), and understand the behavioral characteristics of the entire potential customers population regarding direct mailing. Each node is labeled with the attribute it tests, and its branches are labeled with its corresponding values.

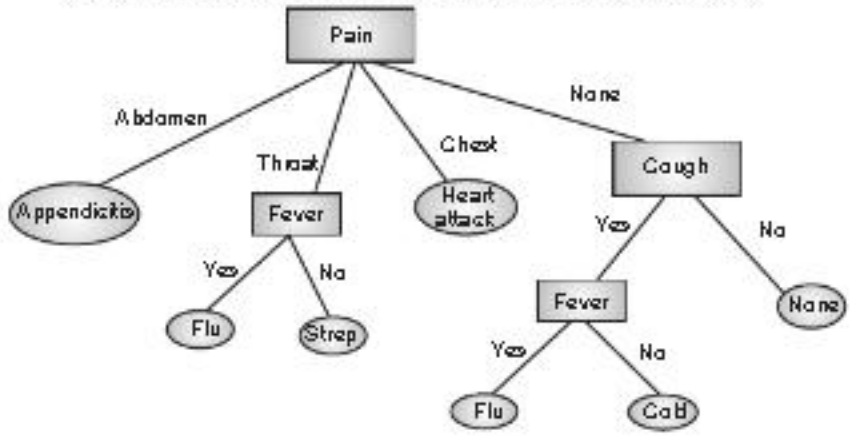


Fig. 4.3

- Decision trees are trees that classify instances by sorting them based on feature values. Each node in a decision tree represents a feature in an instance to be classified, and each branch represents a value that the node can assume. Instances are classified starting at the root node and sorted based on their feature values.
- In case of numeric attributes, decision trees can be geometrically interpreted as a collection of hyper planes, each orthogonal to one of the axes. Naturally, decision–makers prefer less complex decision trees, since they may be considered more comprehensible.
- The tree complexity is explicitly controlled by the stopping criteria used and the pruning method employed. Usually the tree complexity is measured by one of the following metrics : the total number of nodes, total number of leaves, tree depth and number of attributes used. Decision tree induction is closely related to rule induction.
- Each path from the root of a decision tree to one of its leaves can be transformed into a rule simply by conjoining the tests along the path to form the antecedent part, and taking the leaf's class prediction as the class value.

The construction of the decision tree involves the following three main phases.

1. **Construction Phase :** The initial decision tree is constructed in this phased, based on the entire training data set. It requires recursively partitioning the training set into two or more, sub–partition using a splitting criterion, until a stopping criteria is met.
2. **Pruning Phase :** the tree constructed in the previous phase may not result in the best possible set of rules due to over–fitting. The pruning phase removes some of the lower branches and nodes to improve its performance.
3. **Processing the Pruned Tree :** to improve understandability.

4.4.1 Decision Tree Algorithm

A Decision Tree (DT) Model is a Computational Model Consisting of Three Parts :

1. A decision tree is defined.
2. An algorithm to create the tree.
3. An algorithm that applies the tree to data and solves the problem under consideration.

Algorithm :

Input : T// Decision Tree

D// Input Database

Output :M// Model Prediction

DT Proc algorithm :

// simplest algorithm to illustrate prediction technique using DT.

For each t ε D do

n = root node of T;

While n not leaf node do

Obtain answer to question on n applied to t;

Identify arc from t, which contains correct answer;

n = node at end of this arc;

Make prediction for t based on label of n;

4.4.2 Strengths

(a) Decision Tree are Able to Generate Understandable Rules : The ability of decision tree to generate rules that can be translated into comprehensive English or SQL is the greatest strength of this technique.

(b) Ability to Clearly Indicate Best Field : Decision tree building algorithms put the field that does the best job of splitting the training records at the root node of the tree

(c) Decision Tree Able to Handle Both Continuous and Categorical Variables : Continuous variable are equally easy to split by picking a number somewhere in their range of values. Categorical variables, which pose problems for neural networks for statistical techniques, come ready–made with their own splitting criteria : One branch for each category.

4.4.2 Weaknesses

- **Decision :** Tree are less appropriate for estimation tasks where the goal is to predict the value of continuous such as income, blood pressure, or interest rate.
- **Decision Tree are also Problematic for Time :** Series data values a lot of effort is put into presenting the data in such a way that tends and sequential patterns are made visible.
- **Error–Prone with too Many Classes :** Some decision tree algorithm can only deal with binary–valued target classes(yes/no, accept/reject);others are able to assign records to an arbitrary number of classes, but are error prone when the number of training examples per class gets small.
- **Computationally Expensive to Train :** The process of growing a decision tree is computationally expensive. At each node, each candidate splitting field must be sorted before its best split can be found. Pruning algorithm can also be expensive since many candidate sub trees must be formed and compared.

4.4.3 Attribute Selection Measures

An attribute selection measure is a heuristic for selecting the splitting criterion that best separates a given data partition, D of class–labeled training tuple into individual classes. It is also known as splitting rules because they determine how the tuple at a given node are to be split. The attribute having the best score for the measure is chosen as splitting attribute for a given tuple.

Three Popular Attribute Selection Measures :

1. **Information Gain :** It is defined as the difference between the original information requirement and the new requirement.

   Gain (A) = Info (D) – Info A (D)

   ; Where D is data partition, be a training set of class–labeled tuple.

   ; Info (D) is Entropy of D; InfoA (D) is expected information required to classify tuple from D based on the partitioning by A.

2. **Gain Ratio :** It applies a kind of normalization to information gain using split info valued defined analogously with Info (D).

   Gain Ratio (A) = Gain (A)/ split Info (A); attribute with maximum gain ratio is selected as the splitting attribute.

3. **Gini Index :** The Gini Index measures the impurities of D, data partition or set of training tuples.

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

   pi is probability that an arbitrary tuple in D belongs to class Ci.;

   pi = |Ci,D|/|D|. The attribute that have maximum the reduction in impurity is selected as the splitting criteria.

- Many irrelevant attributes may be present in data to be mined. So they need to be removed. Also many mining algorithms don't perform well with large amounts of features or attributes. Therefore feature selection techniques needs to be applied before any kind of mining algorithm is applied.
- The main objectives of feature selection are to avoid overfitting and improve model performance and to provide faster and more cost–effective models.
- The selection of optimal features adds an extra layer of complexity in the modeling as instead of just finding optimal parameters for full set of features, first optimal feature subset is to be found and the model parameters are to be optimized.
- Attribute selection methods can be broadly divided into filter and wrapper approaches. In the filter approach the attribute selection method is independent of the data mining algorithm to be applied to the selected attributes and assess the relevance of features by looking only at the intrinsic properties of the data.
- In most cases a feature relevance score is calculated, and low scoring features are removed. The subset of features left after feature removal is presented as input to the classification algorithm.
- Advantages of filter techniques are that they easily scale to high dimensional datasets are computationally simple and fast, and as the filter approach is independent of the mining algorithm so feature selection needs to be performed only once, and then different classifiers can be evaluated.
- Disadvantages of filter methods are that they ignore the interaction with the classifier and that most proposed techniques are univariate which means that each feature is considered separately, thereby ignoring feature dependencies, which may lead to worse classification performance when compared to other types of feature selection techniques.
- In order to overcome the problem of ignoring feature dependencies, a number of multivariate filter techniques were introduced, aiming at the incorporation of feature dependencies to some degree. Wrapper methods embed the model hypothesis search within the feature subset search.
- In the wrapper approach the attribute selection method uses the result of the data mining algorithm to determine how good a given attribute subset is. In this a search procedure in the space of possible feature subsets is defined, and various subsets of features are generated and evaluated.
- The major characteristic of the wrapper approach is that the quality of an attribute subset is directly measured by the performance of the data mining algorithm applied to that attribute subset.
- The wrapper approach tends to be much slower than the filter approach, as the data mining algorithm is applied to each attribute subset considered by the search.
- In addition, if several different data mining algorithms are to be applied to the data, the wrapper approach becomes even more computationally expensive. Advantages of wrapper approaches include the interaction between feature subset search and model selection, and the ability to take into account feature dependencies.
- A common drawback of these techniques is that they have a higher risk of overfitting than filter techniques and are very computationally intensive. Another category of feature selection technique was also introduced, termed embedded technique in which search for an optimal subset of features is built into the classifier construction, and can be seen as a search in the combined space of feature subsets and hypotheses. Just like wrapper approaches, embedded approaches are thus specific to a given learning algorithm.
- Embedded methods have the advantage that they include the interaction with the classification model, while at the same time being far less computationally intensive than wrapper methods.

4.4.4 Tree Pruning

When a decision tree is built, many of the branches will reflect anomalies in the training data due to noise or outliers. Tree pruning methods address this problem of over fitting the data.

There are Two Common Approaches to Tree Pruning:

1. **Pre–Pruning :** In the pre–pruning approach, a tree is "pruned" by halting its construction early. Upon halting, the node becomes leaf. The leaf may hold most frequent class among the subsets tuples or the probability distribution of those tuples. When constructing a tree, attribute selection measures such as statistical significance, information gain, gini index and so on can be used to access the goodness of a split. If partitioning the tuple at node would result in a split that falls below a prespecified threshold, then further partitioning of a given subset is halted. There are difficulties, however in choosing an appropriate threshold. High threshold could result in oversimplified trees; whereas low thresholds could result in very little simplification.
2. **Post–Pruning:** Post–pruning removes sub trees from a "fully grown" tree. A sub tree at a given node is pruned by removing its branches and replacing it with leaf. The leaf is labeled with the most frequent class among the sub tree being replaced. The "best" pruned tree is one that minimizes the encoding bits. This method adopts MDL (Minimum Description Length) principle. The basic idea is that the simplest solution is preferred. Alternatively, pre–pruning and

post–pruning may be interleaved for a combined approach. Post–pruning requires more computation than pre–pruning, yet generally leads to a more reliable tree. Although pruned tree tends to be more compact than their unpruned counterparts, they may still be rather large and complex. Decision tree can suffer from repetition and replication.

4.4.5 Pruning Techniques

- Although the decision trees are accurate and efficient, they often suffer the disadvantage of providing very large trees that make them incomprehensible to experts. To solve this problem, researchers in the field have considerable interest in tree pruning.
- Tree pruning methods convert a large tree into a small tree, making it easier to understand. Such methods typically use statistical measures to remove the least reliable branches, generally resulting in faster classification and an improvement in the ability of the tree to correctly classify independent test data".
- It is necessary to know the advantage and disadvantage of every decision tree pruning method before it is decided that which pruning method will be selected.

The Following are Some Main Methods to Simplify Decision Trees.

4.4.6 Reduced Error Pruning

- This pruning technique is developed by Quinlan. Reduced error pruning is the simplest and most understandable pruning method in decision tree pruning. For every non–leaf subtree S of the original decision tree, the change in misclassification over the test set is examined.
- The misclassification would occur if this subtree were replaced by the best possible leaf which is the majority of leaf. If the error rate of the new tree would be equal to or smaller than that of the original tree and that subtree S contains no subtree with the same property, S is replaced by the leaf.
- Otherwise, stop the process. The constraint that the subtree S contains no subtree with the same property guarantees reduced error pruning in bottom–up induction.
- Since each node is visited only once to evaluate the opportunity of pruning it, the advantage of this method is its linear computational complexity. However, this method requires a test set separate from the cases in the training set from which the tree was constructed. When the test set is much smaller than the training set, this method may lead to over pruning.
- Many researchers found that Reduced Error Pruning performed as well as most of the other pruning methods in terms of accuracy and better than most in terms of tree size.

4.4.7 Pessimistic Error Pruning

- It is optimistic for user to use a training set to test the error rate of a decision tree, because decision trees have been tailored to the training set. In this case, the error rate can be 0. But some data other than the training set is used; the error rate will increase dramatically.
- To solve this problem, Quinlan used continuity correction for the binomial distribution to get an error rate which is more realistic. In statistics, continuity correction is a useful method in the application of the normal distribution to the computation of binomial probabilities.
- When the normal distribution (a continuous distribution) is used to find approximate answers to problems arising from the binomial distributions (discrete distribution), an adjustment is made for the mismatch of types of distribution. This is called the "**Continuity Correction**". Quinlan uses the following equations to obtain the number of mis-classifications :

$$n'(t) = e(t) + (1/2) \quad \ldots(1)$$

$$n'(Tt) = e(Tt) + (NT/2) \quad \ldots(2)$$

Equation (1) is the number of misclassifications for node t and (2) is the number of misclassifications for subtree T.

Where :

NT is the number of leaves for subtree T,

e(t) is the number of misclassifications at node t,

e(Tt) is the number of misclassifications for subtree T.

The 1/2 in the equation (1) and (2) is a constant which indicates the contribution of a leaf to the complexity of the tree.

- This pruning method only keeps the subtree if its corrected (from equation 2) is more than one standard error better than the figure for the node (from equation 1). This method is much faster than Reduced Error Pruning and also provides higher accuracies. Its disadvantage is that, in the worst case, when the tree does not need pruning at all, each node will still be visited once.

4.4.8 Cost–Complexity Pruning

- This method was proposed by Breiman. It takes of both the number of errors and the complexity of the tree. The size of the tree is used to represent the complexity of the tree. It is also known as the CART pruning method and can be described it in two steps

1. Selection of a parametric family of subtrees of $\{T_0; T_1;\ldots; T_t\}$, according to some heuristics. T_0 is the original decision tree and each T_{i+1} is obtained by replacing one or more subtrees of T_i with leaves by pruning those branches that show the lowest increase in apparent error rate per pruned leaf until the final tree T_t is just a leaf.

2. Choice of the best tree according to an estimate of the true error rates of the trees in the parametric family. For example, consider subtree T used to classify each of the N cases in the training set and E of N examples are wrongly classified if subtree T is replaced by the best leaf. Let NT be the number of leaves in subtree T, the following equation is used to define the total cost–complexity of subtree T :

$$\text{Cost–complexity} = \left(\frac{E}{N}\right) + \alpha \times NT \quad \ldots(3)$$

where α is the cost of one extra leaf in the tree and gives the reduction in error per leaf.

3. If the subtree is pruned, the new tree would misclassify M more of the cases in the training set but would contain NT –1 fewer leaves.
The same cost–complexity will be obtained when

$$\alpha = \left(\frac{M}{(N \times (NT - 1))}\right) \quad \ldots(4)$$

4. From the above equation, α can be calculated for each subtree and the subtree(s) with the smallest value of α is selected for pruning. Continue to process this until the leaf is obtained. The next job is to select one of the trees. The standard error (SE) of the misclassification rate is

$$SE = \frac{(R \times (100 - R))}{N} \quad \ldots(5)$$

where :

R = misclassification rate of the pruned tree,

N = number of examples in the test data.

5. The smallest tree whose observed number of errors on the test set does not exceed R + SE is selected.
6. This method requires a pruning set distinct from the original training set. Its disadvantage is that it can only choose a tree in the set $\{T_0, T_1, \ldots, T_U\}$, which is obtained in the first step, instead of the set of all possible subtrees.
7. It also seems anomalous that the cost–complexity model used to generate the sequence of subtrees is abandoned when the best tree is selected.

Minimum Error Pruning

- This method was developed by Niblett and Brotko. It is a bottom–up approach which seeks a single tree that minimizes the expected error rate on an independent data set.
- Assume that there are k classes for a set of data which number is n and nc is the class c with the greatest number of data. If it is predicted that all future examples will be in class c, the following equation is used to predict the expected error rate :

$$Ek = \frac{(n - nc + k - 1)}{(n + k)} \quad \ldots(6)$$

where :

k is the number of classes for all data,

Ek is the expected error rate if we predict that all future examples will be in class c.

The Method Consists of Three Steps :

1. At each non–leaf node in the decision tree, use equation (6) to calculate the expected error rate if that subtree is pruned.
2. Calculate the expected error rate if the node is not pruned, combined by weighting according to the proportion of observations along each branch.
3. If pruning the node leads to a greater expected error, then keep the subtree; otherwise, prune it.

Critical Value Pruning

- This method was proposed by Mingers. In this method, a threshold, named the critical value, is set to estimate the importance or strength of a node. When the node does not reach the critical value, it will be pruned.
- But when a node meets the pruning condition but its children do not all meet the pruning condition, this branch should be kept because it contains relevant nodes. If a larger critical value is selected, a smaller resulting tree will be obtained because of the more drastic pruning.

Mingers Describes the Critical Value Pruning as Two Main Steps :

1. Prune subtree for increasing critical values,
2. Measure the significance of the pruned trees as a whole and their predictive ability and choose the best tree among them.

- The disadvantage of this method is its strong tendency to under prune and this method selects trees with comparatively low predictive accuracy

Optimal Pruning

- Breiman introduce a convenient terminology used to state and verify the mathematical properties of optimal pruning.
- They also introduce an algorithm to select a particular optimally pruned subtree from among the k candidates .
Bratko, Bohanec, and Almuallim address the issue of finding optimal pruning in another way. Bohanes introduced an algorithm guaranteeing Optimal Pruning (OPT), and Almuallim further improved OPT in terms of the computational complexity.
- Their motivation for simplifying decision trees is different from the typical motivation for pruning decision trees when learning from noisy data. Both of them assume that the initial, unpruned decision trees are completely correct.
- However, in learning from noisy data, which is our case, it is assumed that the initial, unpruned decision tree is inaccurate and appropriate pruning would improve its accuracy.

Cost–Sensitive Decision Tree Pruning

- One main problem for many decision tree pruning methods is that when a decision tree is pruned, it is always assumed that all the classes are equally probable and equally important. However, in real–world classification problems, there is also a cost associated with misclassifying examples from each class.

- Currently, the most common method for cost–sensitive pruning method is to use techniques in statistics to deal with the problem. The use of probability models and statistical methods for analyzing data has become common practice in virtually all scientific disciplines. For example, M. Jordan used a statistical approach to build a decision tree model. A parameter can be estimated from sample data either by a single number (a point estimate) or an entire interval of plausible values (a confidence interval).
- Frequently, however, the objective of an investigation is not to estimate a parameter but to decide which of two contradictory claims about the parameter is correct (some cost–sensitive pruning method makes use of this. Methods for accomplishing this comprise the part of statistical inference called hypothesis testing. The null hypothesis, denoted by H_0, is the claim about one or more population characteristics that is initially assumed to be true.
- The alternative hypothesis, denoted by Ha, is the assertion that is contradictory to H_0. The null hypothesis will be rejected in favor of the alternative hypothesis only if sample evidence suggests that H_0 is false. If the sample does not strongly contradict H_0, it will continue to believe in the truth of the null hypothesis.

The Many Benefits in Data Mining that Decision Trees Offer are :

- Decision trees are self–explanatory and when compacted they are also easy to follow. In other words if the decision tree has a reasonable number of leaves, it can be grasped by non–professional users. Furthermore decision trees can be converted to a set of rules. Thus, this representation is considered as comprehensible.
- Able to handle a variety of input data : nominal, numeric and textual
- Able to process datasets that may have errors or missing values
- High predictive performance for a relatively small computational effort
- Available in many data mining packages over a variety of platforms
- Useful for various tasks, such as classification, regression, clustering and feature selection.
- Decision trees are capable of handling datasets that may have errors.
- Decision trees are capable of handling datasets that may have missing values.

4.5 ID3

- Data mining is the technique to extract the hidden predictive data from large databases; it is an influential technology and used by lot of companies because of very abnormal fallouts. Data mining is very supportive technique to analyse the forthcoming prediction with the help of historical behaviour of data and statistics, these features of data mining sanction proactive business and it is called knowledge–driven decisions.
- This automation, prospective scrutinizing and exploration of past events work as demonstration tool and implement a DSS (Decision Support System).
- The construction of decision trees from data is a longstanding discipline.
- Decision trees are a very effective method of supervised learning. It aims is the partition of a dataset into groups as homogeneous as possible in terms of the variable to be predicted. It takes as input a set of classified data, and outputs a tree that resembles to an orientation diagram where each end node (leaf) is a decision (a class) and each non– final node (internal) represents a test. Each leaf represents the decision of belonging to a class of data verifying all tests path from the root to the leaf.
- The tree is simpler, and technically it seems easy to use. In fact, it is more interesting to get a tree that is adapted to the probabilities of variables to be tested. Mostly balanced tree will be a good result.
- If a sub–tree can only lead to a unique solution, then all sub–tree can be reduced to the simple conclusion, this simplifies the process and does not change the final result. Ross Quinlan worked on this kind of decision trees.
- Data mining techniques can rapidly implement on existing software and hardware and intensify the quality of service of them.

4.5.1 Information Theory

- Theories of Shannon are at the base of the ID3 algorithm. Entropy Shannon is the best known and most applied. It first defines the amount of information provided by an event : The higher the probability of an event is low (it is rare), the more information it provides is great. (In the following all logarithms are base2).

(A) Shannon Entropy

In general, if we are given a probability distribution $P = (P_1, P_2,..., P_n)$ and a sample **S** then the **Information** carried by this distribution, also called the **Entropy of P** is giving by :

$$\text{Entropie (P)} = -\sum_{i=1}^{n} p_i \times \log (p_i)$$

(B) The Gain Information G (p, T)

We have functions that allow us to measure the degree of mixing of classes for all sample and therefore any position of the tree in construction. It remains to define a function to select the test that must label the current node.

It defines the gain for a test **T** and a position **p**

$$\text{Gain (p, T)} = \text{Entropie (p)} - \sum_{i=1}^{n} (p_j \times \text{Entropie (p}_j\text{)})$$

where values (pj) is the set of all possible values for attribute T. We can use this measure to rank attributes and build the decision tree where at each node is located the attribute with the highest information gain among the attributes not yet considered in the path from the root.

4.5.2 ID3 Algorithm

- ID3 is a simple decision tree erudition algorithm developed by Ross Quinlan (1983). The basic idea of ID3 algorithm is to create a decision tree of given set, by using top–down greedy search to check each attribute at every tree node.
- To select the most useful attribute using classification technique, we present a metric information gain and to catch an optimal way to classify an erudite set, we need to minimize the depth of the tree.
- Thus, we need some function which should be able to measure the most balanced splitting.
- The information gain metric is such a function that we can use for efficient balanced splitting. In direction to define information gain exactly, we need to deliberate entropy. First, let's assume that the resulting decision tree classifies instance into two classes without loss of simplification and we would call them P (positive) and N (negative).
- Given set S, containing these positive and negative targets, the entropy of S related to this Boolean classification is :

  $\text{Entropy(S)} = -P(\text{positive}) \log_2 P(\text{positive}) - P(\text{negative}) \log_2 P(\text{negative})$

  P (positive) : Proportion of positive examples in S

  P (negative) : Proportion of negative examples in S
- So concerning Points are as we discussed, to minimize the decision tree depth; we need to select the optimal attribute for splitting the tree node, so that we can easily imply the attribute with the maximum entropy reduction.
- The attribute that can help in maximum entropy reduction is the optimal attribute for splitting. We define Information Gain as the predictable reduction of entropy related to specified attribute when splitting a decision tree node.

The information gain, Gain(S, A) of an attribute A,

Gain(S, A) = Entropy(S) –Sum for v from 1 to n of (|Sv|/|S|) × Entropy (Sv).

- ID3 is a supervised learning algorithm, builds a decision tree from a fixed set of examples. The resulting tree is used to classify future samples. ID3 algorithm builds tree based on the information (information gain) obtained from the training instances and then uses the same to classify the test data. ID3 algorithm generally uses nominal attributes for classification with no missing values.
- The pseudo code of this algorithm is very simple. Given a set of attributes not target $C_1, C_2, ..., C_n$, C the target attribute, and a set S of recording learning.

Inputs : R : a set of non– target attributes, C : the target **attribute**, S : training data.

Output : returns a decision tree

Start

Initialize to empty tree;

If S is empty then

Return a single node failure value

End If

If S is made only for the values of the same target **then**

Return a single node of this value

End if

If R is empty **then**

Return a single node with value as the most common value of the target attribute values found in S

End if

D ← the attribute that has the largest Gain (D, S) among all the attributes of R

{dj j = 1, 2, .., m} ← Attribute values of D

{Sj with j = 1, 2, .., m} ←The subsets of S respectively constituted of dj records attribute value D

Return a tree whose root is D and the arcs are labeled by d1, d2, .., dm and going to sub–trees ID3 (R–{D}, C, S1),

ID3 (R–{D} C, S2), ., ID3 (R–{D}, C, Sm)

End

- Suppose we want to use the ID3 algorithm to decide if the time ready to play ball.
- During two weeks, the data are collected to help build an ID3 decision tree (Table 4.3).
- The classification of the target is "should we play ball?" which can be Yes or No.
- Weather attributes outlook, temperature, humidity and wind speed.

They can take the following values :

Outlook = {Sun, Overcast, Rain}

Temperature = {Hot, Sweet, Cold}

Humidity = {High, Normal}

Wind = {Low, High}

Examples of the set S are :

Table 4.3 : Dataset S

Day	Outlook	Temperature	Humidity	Wind	Play
D1	Sun	Hot	High	Low	No
D2	Sun	Hot	High	High	No
D3	Overcase	Hot	High	Low	Yes
D4	Rain	Sweet	High	Low	Yes
D5	Rain	Cold	Normal	Low	Yes
D6	Rain	Cold	Normal	High	No
D7	Overcast	Cold	Normal	High	No
D8	Sun	Sweet	High	Low	No
D9	Sun	Cold	Normal	Low	Yes
D10	Rain	Sweet	Normal	Low	Yes
D11	Sun	Sweet	Normal	High	Yes
D12	Overcast	Sweet	High	High	Yes
D13	Overcast	Hot	Normal	Low	Yes
D14	Rain	Sweet	High	High	No

We need to find the attribute that will be the root node in our decision tree. The gain is calculated for the four attributes.

The entropy of the set S :

$$\text{Entropy (S)} = -\frac{9}{14}\times \log_2\left(\frac{9}{14}\right)-\frac{5}{14}\times \log_2\left(\frac{5}{14}\right) = 0.94$$

Calculation for the first attribute

$$\text{Gain(S, Outlook)} = \text{Entropy (S)} - \frac{5}{14}\times \text{Entropy (SSun)} - \frac{4}{14} \times \text{Entropy (SRain)} - 5/14^* \text{ Entropy (SOvercast)}$$

$$= 0.94 - \frac{5}{14}\times 0.9710 - \frac{4}{14}\times 0 - \frac{5}{14}\times 0.9710$$

$$\text{Gain(S, Outlook)} = 0.246$$

Calculation of entropies :

$$\text{Entropy (SSunl)} = -\frac{2}{5}\times \log_2\left(\frac{2}{5}\right)-\frac{3}{5}\times \log_2\left(\frac{3}{5}\right) = 0.9710$$

$$\text{Entropy (SRain)} = -\frac{4}{4}\times \log_2\left(\frac{4}{4}\right) - 0\times \log_2(0) = 0$$

$$\text{Entropy (Sovercast)} = -\frac{3}{5}\times \log_2\left(\frac{3}{5}\right) - \times \log_2\left(\frac{2}{5}\right) = 0.9710$$

As well we find for the other variables :

Gain(S, Wind) = 0.048

Gain(S, Temperature) = 0.0289

Gain(S, Humidity) = 0.1515

Outlook attribute has the highest gain, so it is used as a decision attribute in the root node of the tree.

Since Visibility has three possible values, the root node has three branches (Sun, Rain and Overcast).

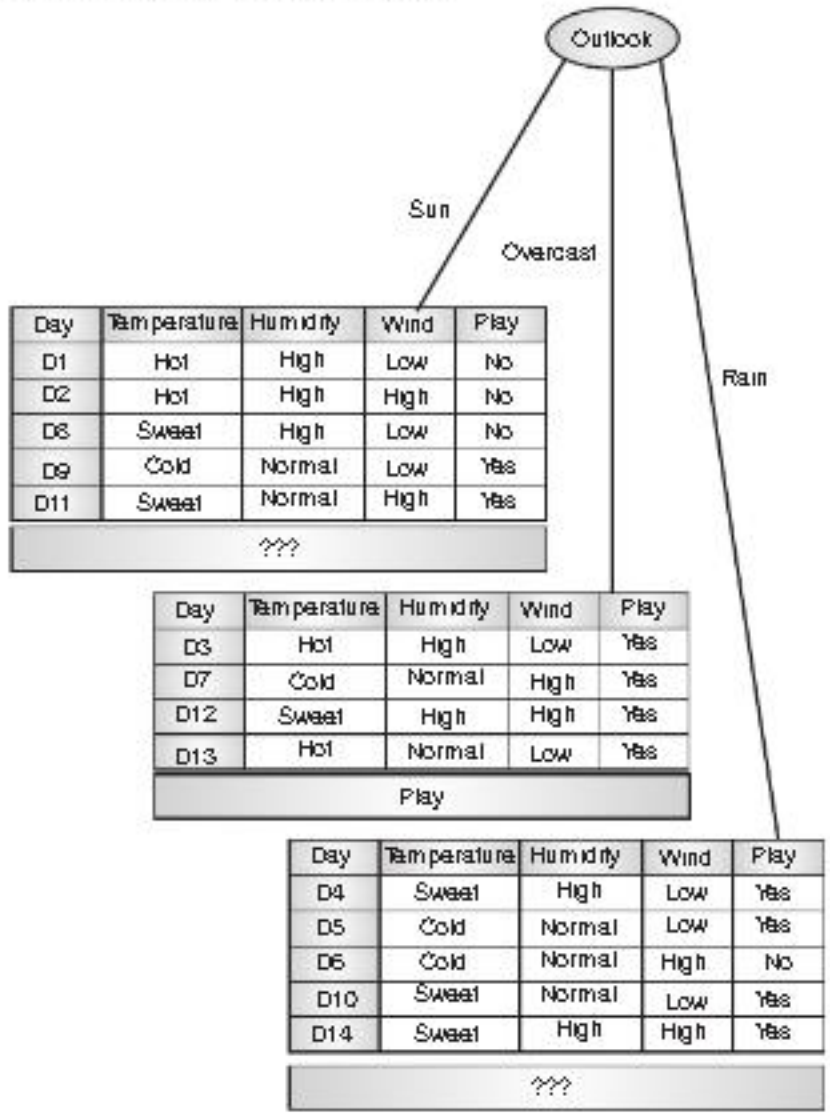


Fig. 4.4 : (a) ID3 final tree

So by using the three new sets, the information gain is calculated for the temperature, humidity, until we obtain subsets Sample containing (almost) all belonging examples to the same class.

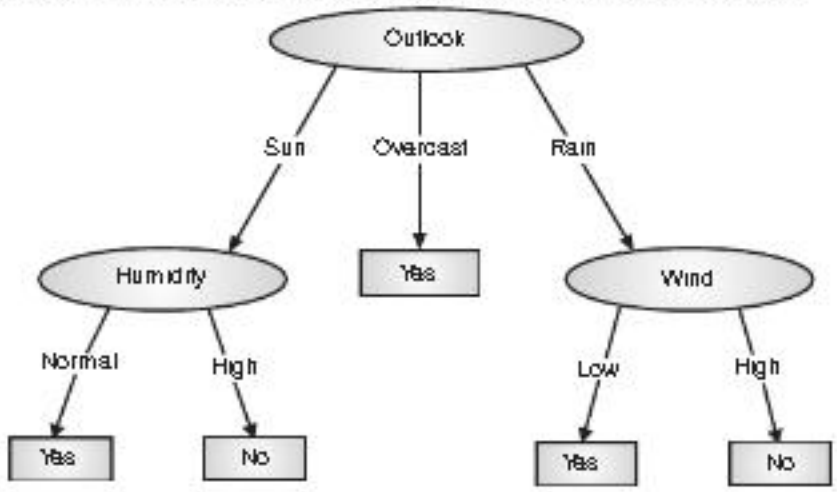


Fig. 4.4 : (b) ID3 final tree

4.6 SCALABLE DECISION TREES

- Classification has been identified as an important problem in the emerging field of data mining.
- In classification, we are given a set of example records, called a training set, where each record consists of several fields or attributes. Attributes are either continuous, coming from an ordered domain, or categorical, coming from an unordered domain. One of the attributes, called the classifying attribute, indicates the class to which each example belongs. The objective of classification is to build a model of the classifying attribute based upon the other attributes.
- Once a model is built, it can be used to determine the class of future unclassified records. Applications of classification arise in diverse fields, such as retail target marketing, customer retention, fraud detection and medical Diagnosis. Decision trees can be constructed relatively fast compared to other methods. Another advantage is that decision tree models are simple and easy to understand. Moreover, trees can be easily converted into SQL statements that can be used to access databases efficiently.
- Finally, decision tree classifiers obtain similar and sometimes better accuracy when compared with other classification methods. A decision tree is a class discriminator that recursively partitions the training set until each partition consists entirely or dominantly of examples from one class.
- Each non–leaf node of the tree contains a split point which is a test on one or more attributes and determines how the data is partitioned.

Problems with Traditional Serial Decision Tree

- Random sampling is often used to handle large datasets when building a classifier. Previous work on building tree classifiers from large datasets includes Catlett's two methods whose emphasis is for improving the time taken to develop a classifier. The first method used data sampling at each node of the decision tree, and the second discretized continuous attributes.
- However, Catlett only considered datasets that could fit in memory; the largest training data had only 32,000 examples. Chan and Ytolfo considered partitioning the data into subsets that fit in memory and then developing a classifier on each subset in parallel. The output of multiple classifiers is combined using various algorithms to reach the final classification.
- Their studies showed that although this approach reduces running time significantly, the multiple classifiers did not achieve the accuracy of a single classifier built using all the data. However, the cumulative cost of classifying data incrementally can sometimes exceed the cost of classifying the entire training set once.
- To overcome this situation a new algorithm, SLIQ was proposed. Like ID3, SLIQ assumes that the entire dataset can fit in real memory and does not address issues such as disk I/O.
- The algorithms presented here require processor communication to evaluate any given split point, limiting the number of possible partitioning schemes the algorithms can efficiently consider for each leaf.
- The SLIQ classification algorithm addressed several issues in building a fast scalable classifier. SLIQ gracefully handles disk–resident data that is too large to fit in memory.
- It does not use small memory–sized datasets obtained via sampling or partitioning, but builds a single decision tree using the entire training set. However, SLIQ does require that some data per record stay memory–resident all the time.
- Since the size of this in–memory data structure grows in direct proportion to the number of input records, this limits the amount of data that can be classified by SLIQ.
- Another decision–tree–based classification algorithm, called SPRINT, that removes all of the memory restrictions, and is fast and scalable. The algorithm has also been designed to be easily parallelized. SPRINT has shown excellent scale up, speedup and size up properties.
- The combination of these characteristics makes SPRINT an ideal tool for data mining. We already know that general decision tree algorithm works in two phases : Growing and Pruning.
- The tree growth phase is computationally much more expensive than pruning, since the data is scanned multiple times in this part of the computation. Pruning requires access only to the fully grown decision tree.
- SLIQ has proven that pruning phase typically takes less than 1% of the total time needed to build a classifier. We therefore focus only on the tree growth phase. For pruning, algorithm used in SLIQ, which is based on the Minimum Description Length principle is considered.

Partition(Data S)

if (all points in 5′ are of the same class) then

return;

for each attribute A do

evaluate splits on attribute A;

Use best split found to partition S into S1 and Sz;

Partition(&);

Initial call : Partition(TrainingData)

From Fig. 4.4. General Tree–growth Algorithm

There are Two Major Issues that have Critical Performance Implications in the Tree–Growth Phase :

1. How to find split points that define node tests.
2. Having chosen a split point, how to partition the data.

- The well–known decision tree classifiers, ID3 example, grow trees depth–first and repeatedly sort the data at every node of the tree to arrive at the best splits for numeric attributes. SLIQ, on the other hand, replaces this repeated sorting with one–time sort by using separate lists for each attribute SLIQ uses a data structure called a class list which must remain memory resident at all times.
- The size of this structure is proportional to the number of input records, and this is what limits the number of input records that SLIQ can handle. Scalable Parallelizable Induction of Decision Trees (SPRINT) addresses the above two issues differently from previous algorithms; it has no restriction on the size of input and yet is a fast algorithm. It shares with SLIQ the advantage of a one–time sort, but uses different– data structures. In particular, there is no structure like the class list that grows with the size of input and needs to be memory–resident.

Attribute Lists

- SPRINT initially creates an attribute list for each attribute in the data. Entries in these lists, which we will call attribute records, consist of an attribute value, a class label, and the index of the record (tid) from which these value were obtained.
- Initial lists for continuous attributes are sorted by attribute value once when first created. If the entire data does not fit in memory, attribute lists are maintained on disk.
- The initial lists created from the training set are associated with the root of the classification tree. As the tree is grown and nodes are split to create new children, the attribute lists belonging to each node are partitioned and associated with the children.
- When a list is partitioned, the order of the records in the list is preserved; thus, partitioned lists never require resorting.

Parallelizing Classification

- We now turn to the problem of building classification trees in parallel. We again focus only on the growth phase due to its data intensive nature. The pruning phase can easily be done off–line on a serial processor as it is computationally inexpensive, and requires access to only the decision–tree grown in the training phase.
- In parallel tree growth, the primary problems remain finding good split–points and partitioning the data using the discovered split points. As in any parallel algorithm, there are also issues of data placement and workload balancing that must be considered.
- Fortunately, these issues are easily resolved in the SPRINT algorithm. 'SPRINT was specifically designed to remove any dependence on data structures that are either centralized or memory–resident; because of these design goals, SPRINT parallelizes quite naturally and efficiently.

Data Placement and Workload Balancing

- Recall that the main data structures used in SPRINT are the attribute lists and the class histograms.
- SPRINT achieves uniform data placement and workload balancing by distributing the attribute lists evenly over N processors of a shared nothing machine. This allows each processor to work on only I/N of the total data.
- The partitioning is achieved by first distributing the training set examples equally among all the processors.
- Each processor then generates its own attribute list partitions in parallel by projecting out each attribute from training–set examples it was assigned. Lists for categorical attributes are therefore evenly partitioned and require no further processing.
- However, continuous attribute lists must now be sorted and repartitioned into contiguous sorted sections. The result of this sorting operation is that each processor gets a fairly equalized sorted sections of each attribute list.

Deciding Splitting Criterion

- Finding split points in parallel SPRINT is very similar to the serial algorithm. In the serial version, processors scan the attribute lists either evaluating split points for continuous attributes or collecting distribution counts for categorical attributes.
- This does not change the parallel algorithm, no extra work or communication is required while each processor is scanning its attribute list partitions.
- We get the full advantage of having N processors simultaneously and independently processing I/N of the total data.
- The differences between the serial and parallel algorithms appear only before and after the attribute list partitions are scanned.

Continuous Attributes

- For continuous attributes, the parallel version of SPRINT differs from the serial version in how it initializes the Cbelow and Cabove class–histograms. In a parallel environment, each processor has a separate contiguous section of a "global" attribute list.
- Thus, a processor's Cbelow, and Cabove histograms must be initialized to reflect the fact that there are sections of the attribute list on other processors. Specifically, Cbelow must initially reflect the class distribution of all sections of an attribute list assigned to processors of lower rank.

- The Cabove and Cbelowhistograms must likewise initially reflect the class distribution of the local section as well as all sections assigned to processors of higher rank. As in the serial version, these statistics are gathered when attribute lists for new leaves are created. After collecting statistics, the information is exchanged between all the processors and stored with each leaf, where it is later used to initialize that leaf's Cbelow and Cabove class histograms.
- Once all the attribute list sections of a leaf have been processed, each processor will have what it considers to be the best split for that leaf. The processors then communicate to determine which of the N split points has the lowest cost.

Categorical Attributes

- For categorical attributes, the difference between the serial and parallel versions arises after an attribute–list section has been scanned to build the count matrix for a leaf.
- Since the count matrix built by each processor is based on "local" information only, we must exchange these matrices to get the "global" counts.
- This is done by choosing a coordinator to collect the count matrices from each processor. The coordinator process then sums the local matrices to get the global count–matrix.
- As in the serial algorithm, the global matrix is used to find the best split for each categorical attribute.

Performing the Splits

- Having determined the winning split points, splitting the attribute lists for each leaf is nearly identical to the serial algorithm with each processor responsible for splitting its own attribute list partitions.
- The only additional step is that before building the probe structure, we will need to collect rids from all the processors. (Recall that a processor can have attribute records belonging to any leaf.)
- Thus, after partitioning the list of a leaf's splitting attribute, the rids collected during the scan are exchanged with all other processors. After the exchange, each processor continues independently, constructing a probe–structure with all the rids and using it to split the leaf's remaining attribute lists.

Parallelizing SLIQ

- The attribute lists used in SLIQ can be partitioned evenly across multiple processors as is done in parallel SPRINT. However, the parallelization of SLIQ is complicated by its use of a centralized, memory–resident data–structure the class list. Because the class list requires random access and frequent updating, parallel algorithms based on SLIQ require that the class list be kept memory resident.
- This leads us to two primary approaches for parallelizing SLIQ : one where the class list is replicated in the memory of every processor, and the other where it is distributed such that each processor's memory holds only a portion of the entire list.

Attribute list in SLIQ

Training Data

Age	Salary	Class
30	65	G
23	15	B
40	75	G
55	40	B
55	100	G
45	60	G

Attribute Lists

Age	Class List Index
23	2
30	1
40	3
45	6
55	5
55	4

Salary	Class List Index
15	2
40	4
60	6
65	1
75	3
100	5

Class List

Index	Class	Leaf
1	G	N1
2	B	N1
3	G	N1
4	B	N1
5	G	N1
6	G	N1

Attribute and Class List in SLIQ

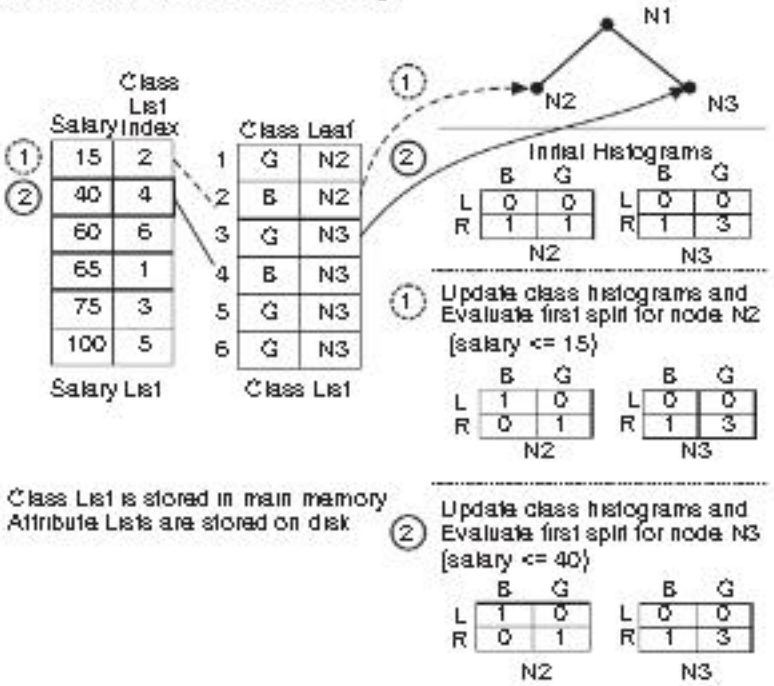


Fig. 4.5

Issues in Scalability

- We know that there are two major operations in decision tree building. First is to evaluate splits for each attribute, and select the best split. This step is perhaps the most costly step. Second step is to create partitions using the best split.
- To find the best split in an attribute, we need to sort training examples on the attribute. This sorting needs to be conducted for each node. Sorting can be costly when the database cannot be held in main memory.

Rule Extraction from Decision Trees

- The input data for a classification task is a collection of records. Each record is also known as an instance or example, is characterized by a tuple (x, y), where x is the set of attributes and y is a special attribute, designated as class label.
- Classification technique is a systematic approach for building classification models from the input data set. J48 is a decision tree algorithm. Decision tree built using this algorithm can be used for classification.
- It builds the tree from a set of training instances. Each sample is represented as $S_i = x_1, x_2 \ldots x_4$, Where $x_1, x_2, \ldots x_4$ is a set of attributes. $C = c_1, c_2, \ldots$ represent the class to which each sample belongs.
- At each node of the tree, this algorithm chooses one attribute of the dataset that most effectively splits the dataset so that each sample get correctly classified. Its criterion is the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. The attribute with highest normalized information gain is chosen to make decision.
- Let p(i|t) denote the fraction of records belonging to class I at a given node t.

$$\text{Entropy (t)} = \sum_{i=0}^{c-1} p(i|t) \log_2 p(i|t)$$

- The measures for selecting best split are often based on degree of impurity of the child nodes. The smaller the degree of impurity, more inaccurate is the class distribution. This degree of impurity is given using Gini Index.

$$\text{Gini(t)} = 1 - \sum_{i=0}^{c-1} [p(i \mid t)]^2$$

4.7 DECISION RULES

- 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.

Rule Extraction using ID3 Decision Tree Algorithm

- ID3 is a decision tree algorithm for constructing decision tree from given dataset. Each node of the tree corresponds to splitting point of the tree, known as splitting attribute. Each branch is a possible value for that attribute.
- At each node splitting attribute is selected which provides highest information gain. This attribute is known to be the most informative attribute among rest of the attributes.
- This algorithm uses the criterion of the information gain to determine goodness of the split. The attribute with the highest information gain is taken as splitting attribute and the dataset is split for all distinct values of the attribute.

Table 4.4 : Weather Dataset

ID	Outlook	Temperature	Humidity	Wind	Play
X1	Sunny	Hot	High	Weak	No
X2	Sunny	Hot	High	Strong	No
X3	Overcast	Hot	High	Weak	Yes
X4	Rain	Mild	High	Weak	Yes
X5	Rain	Cool	Normal	weak	Yes
X6	Rain	Cool	Normal	Strong	No
X7	Overcast	Cool	Normal	Strong	Yes
X8	sunny	Mild	High	Weak	No
X9	sunny	Cool	Normal	Weak	Yes
X10	rain	Mild	Normal	Weak	Yes
X11	sunny	Mild	Normal	Strong	Yes
X12	overcast	Mild	High	Strong	Yes
X13	overcast	Hot	Normal	Weak	Yes
X14	rain	mild	High	Strong	No

- The weather dataset is relatively a small dataset and it is fictitious. It indicates the conditions that are suitable for playing some gain.
- The conditional attributes from this dataset are {Outlook, Temperature, Humidity, Wind} and decision attribute is Play which denotes weather to play or not.
- All four attributes are categorical attribute. Values for first conditional attribute– Outlook are {sunny, overcast, rain}. Values for second conditional attribute– temperature are {cold, mild, hot}, values for third conditional attribute– humidity are {High, Normal} and values for fourth conditional attribute–wind are {Strong, Weak}.
- The values of all four attributes produce $3 \times 3 \times 2 \times 2 = 36$ possible combinations. But as a input we will consider only 14 values. Rule extraction and decision tree for weather dataset

- Weather dataset given in table consists of 14 examples with 9 Yes and 5 NO examples.

$$\text{Entropy}(s) = -\frac{9}{14}\times \log_2\left(\frac{9}{14}\right) - \frac{5}{14}\times \log_2\left(\frac{5}{14}\right) = 0.940$$

- There are four conditional attributes present in the table. In order to find the attribute which can be served as the root node of the decision tree to be induced, information gain is calculated for each attribute.
- For attribute wind, there are 8 occurrences of wind = weak and 6 occurrences of wind = strong. Out of 8 occurrences of wind=weak, 6 examples are yes and 2 examples are no. Out of 6 occurrences of wind=weak, 3 examples are yes and 3 examples are no.

$$\text{Gain}(s, \text{wind}) = \text{Entropy}(s) - \frac{8}{14}\times \text{Entropy}(S_{weak}) - \frac{6}{14} \times \text{Entropy}(S_{strong})$$

$$= 0.940 - \frac{8}{14}\times 0.811 - \frac{6}{14}\times 1.0$$

$$= 0.048$$

$$\text{Entropy}(S_{weak}) = -\left(\frac{6}{8}\right)\times \log_2\left(\frac{6}{8}\right) - \frac{2}{8}\times \log_2\left(\frac{2}{8}\right) = 0.811$$

$$\text{Entropy}(S_{strong}) = -\left(\frac{3}{6}\right)\times \log_2\left(\frac{3}{6}\right) - \frac{3}{6}\times \log_2\left(\frac{3}{6}\right) = 1.0$$

Similarly

Gain(s, outlook) = 0.246

Gain(s, temperature) = 0.029

Gain(s, humidity) = 0.151

- Outlook has the highest gain so it is used as decision attribute in the root node. Since outlook has three values, there are three branches from root node. So the next question is which of the remaining attributes should be tested at the branch node sunny $S_{sunny} = \{x_1, x_2, x_8, x_9, x_{11}\}$. So there are 5 examples from table 4.4 with outlook = sunny.

Gain (S_{sunny}, Humidity) = 0.970

Gain(S_{sunny}, Temperature) = 0.570

Gain(S_{sunny}, Wind) = 0.019

- Thus above calculations show that the attribute humidity shows the highest gain, therefore, it should be used as the next decision node for the branch sunny.

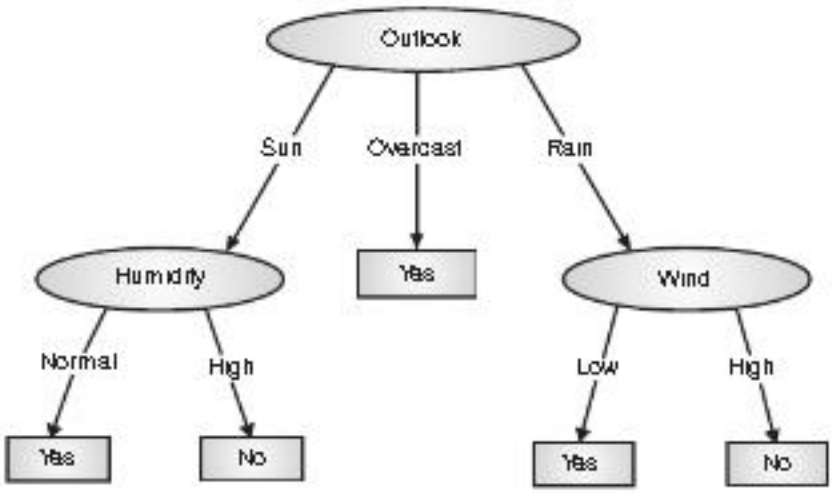


Fig. 4.6

- The process is repeated until all data are classified perfectly or no attribute is left for the child nodes further down the tree.
- Decision tree generated for the weather dataset example corresponding rules are
  - If outlook = sunny and humidity = high then play = no
  - If outlook = sunny and humidity = normal then play = yes
  - If outlook = overcast then play = yes
  - If outlook = overcast and wind = weak then play = yes
  - If outlook = overcast and wind = strong then play = no.
- The disadvantage of decision tree is that it is more time consuming as it first builds the tree and then do the calculations.

Regression

- Regression analysis can imply a broader range of techniques which are useful for data analysis. Statisticians commonly define regression so that the goal is to understand "as far as possible with the available data how the conditional distribution of some response y varies across subpopulations determined by the possible values of the predictor or predictors".
- For example, if there is a single categorical predictor such as male or female, a legitimate regression analysis has been undertaken if one compares two income histograms, one for men and one for women.
- Or, one might compare summary statistics from the two income distributions : The mean incomes, the median incomes, the two standard deviations of income, and so on.
- One might also compare the shapes of the two distributions with a normal distribution. There is no requirement in regression analysis for there to be a "model" by which the data were supposed to be generated.
- There is no need to address cause and effect. And there is no need to undertake statistical tests or construct confidence intervals.
- The definition of a regression analysis can be met by pure description alone. Construction of a "model," often coupled with causal and statistical inference, are supplements to a regression analysis, not a necessary component.
- Given such a definition of regression analysis, a wide variety of techniques and approaches can be applied.
- There is a continuous random variable called the dependent variable, Y, and a number of independent variables, $x_1, x_2, \ldots, x_p$.
- Our purpose is to predict the value of the dependent variable (also referred to as the response variable) using a linear function of the independent variables.

- The values of the independent variables(also referred to as predictor variables, regressors or covariates) are known quantities for purposes of prediction.
- Let x be an instance and let y be its real–valued label. For linear regression, x must be a vector of real numbers of fixed length. Remember that this length p is often called the dimension, or dimensionality, of x. Write $x = (x_1, x_2, x_3, x_4 \ldots\ldots x_p)$. The linear regression model is

$$y = b_0 + b_1x_1 + b_2x_2 + \ldots\ldots\ldots + b_px_p :$$

- The righthand side above is called a linear function of x. The linear function is defined by its coefficients b_0 to b_p. These coefficients are the output of the data mining algorithm.
- The coefficient b_0 is called the intercept. It is the value of y predicted by the model if $x_i = 0$ for all i. Of course, it may be completely unrealistic that all features x have value zero. The coefficient bi is the amount by which the predicted y value increases if x_i increases by 1, if the value of all other features is unchanged. For example, suppose x_1 is a binary feature where $x_1 = 0$ means female and $x_1 = 1$ means male, and suppose $b_1 = –2.5$.
- Then the predicted y value for males is lower by 2.5, everything else being held constant. Suppose that the training set has cardinality n, i.e. it consists of n examples of the form (x_i, y_i), where $x_i = x_{i1} \ldots\ldots x_{ip}$. Let b be any set of coefficients. The predicted value for x_i is

$$\hat{y}_i = f(x_i; b) = b_0 + \sum_{j=1}^{p} b_jx_{ij}$$

- The semicolon in the expression f(x, b) emphasizes that the vector x_i is a variable input, while b is a fixed set of parameter values. If we define $x_{i0} = 1$ for every i, then we can write

$$\hat{y}_i = \sum_{j=0}^{p} b_jx_{ij}$$

The constant $x_{i0} = 1$ can be called a pseudo–feature.

- Finding the optimal values of the coefficients b_0 to bp is the job of the training algorithm. To make this task well–defined, we need a definition of what "optimal" means. The standard approach is to say that optimal means minimizing the sum of squared errors on the training set, where the squared error on training example i is square of $(y_i – y_i)$. The training algorithm then finds.

$$\hat{b} = \text{argmin}_b \sum_{i=1}^{n} (f(x_i, b) – y_i)^2$$

- The objective function $\sum_i (y_i – \sum_j b_jx_{ij})^2$ is called the sum of squared errors, or SSE for short. Note that during training the n different xi and yi values are fixed, while the parameters b are variable.
- The optimal coefficient values ^b are not defined uniquely if the number n of training examples is less than the number p of features.
- Even if n > p is true, the optimal coefficients have multiple equivalent values if some features are themselves related linearly. Here, "equivalent" means that the different sets of coefficients achieve the same minimum SSE.
- For an intuitive example, suppose features 1 and 2 are temperature measured in degrees Celsius and degrees Fahrenheit respectively.
- Then $x_2 = 32 + 9(x_1/5) = 32 + 1.8x_1$, and the same model can be written in many different ways :

$$y = b_0 + b_1x_1 + b_2x_2$$

$$y = b_0 + b_1x_1 + b_2(32 + 1.8 \times 1)$$

$$= [b_0 + 32b_2] + [b_1(1 + 1.8b_2)] \times 1 + 0 \times 2$$

and an infinite number of other ways. In the extreme, suppose $x_1 = x_2$. Then all models $y = b_0 + b_1x_1 + b_2x_2$ are equivalent for which $b_1 + b_2$ equals a constant.

- When two or more features are approximately related linearly, then the true values of the coefficients of those features are not well determined. The coefficients obtained by training will be strongly influenced by randomness in the training data.
- Regularization is a way to reduce the influence of this type of randomness. Consider all models $y = b_0 + b_1x_1 + b_2x_2$ for which $b_1 + b_2 = c$. Among these models, there is a unique one that minimizes the function $b_1^2 + b_2^2$. This model has $b_1 = b_2 = c/2$.
- We can obtain it by setting the objective function for training to be the Sum of Squared Errors (SSE) plus a function that penalizes large values of the coefficients. A simple penalty function of this type is $\sum_{j=1}^{p} b_j^2$. A parameter λ can control the relative importance of the two objectives, namely SSE and penalty :

$$\hat{b} = \text{argmin}_b \frac{1}{n} \sum_{i=1}^{n} (y_i – \hat{y}_i)^2 + \lambda \frac{1}{p} \sum_{j=1}^{p} b_j^2$$

If $\lambda = 0$ then one gets the standard least–squares linear regression solution. As λ increases, the penalty on large coefficients gets stronger, and the typical values of coefficients get smaller. The parameter λ is often called the strength of regularization.

- The fractions 1/n and 1/p do not make an essential difference. They can be used to make the numerical value of λ easier to interpret.

- The penalty function $\sum_{j=1}^{p} b_j^2$ is the square of the L_2 norm of the vector b. Using it for linear regression is called ridge regression. Any penalty function that treats all coefficients bj equally, like the L_2 norm, is sensible only if the typical magnitudes of the values of each feature are similar; this is an important motivation for data normalization.
- Note that in the formula

$$\sum_{j=1}^{p} b_j^2$$

the sum excludes the intercept coefficient b_0. One reason for doing this is that the target y values are typically not normalized.

4.8 EVALUATING A DECISION TREE

A decision tree can help you to make tough choices between different paths and outcomes, but only if you evaluate the model correctly. Decision trees are graphic models of possible decisions and all related possible outcomes in a tree form, with the outcomes shown as "branches" off each choice. You can use a decision tree to help you to make all kinds of business decisions, including new product development, new marketing strategies and workforce changes.

Features

- Assign a numerical value to each possible outcome on the tree. Use dollar amounts for outcomes. For example, if one outcome would gain the company $100,000, mark "$100,000" next to it. Estimate company value for outcomes without a specific price tag.
- Label the likelihood of each outcome. Use whole percentages for each outcome on the same branch. The outcome percentages must total 100 for each set of possible outcomes for the same branch. For example, for a decision branch with four possible outcomes, the percentages of all four outcomes must equal 100. Use past experience and data to estimate the possibility of each outcome.
- Make a separate list for each decision and its possible outcomes. Calculate an adjusted outcome value by multiplying the likelihood percentage by the value. For example, label an outcome worth $300,000 that has a 30-percent chance of succeeding as $90,000 on your list.
- Review each branch on the tree for costs. You must factor in the costs of the decisions when looking at outcome values. Subtract the cost for each decision from your adjusted outcome values. Label the results "Final Outcomes."

Evaluation

- Look at the possible decisions on the tree for all "Final Outcomes." Mark the decisions that carry a lot of risk and the decisions that have a low probability of a successful outcome.
- Look at the risky decisions. Consider whether your business can tolerate the amount of risk. For example, a decision with an outcome of $150,000 that costs $20,000 to attempt is dangerous if your business can't afford to lose $20,000. Eliminate outcomes with unaffordable attached risks.
- Look at the decisions for the outcomes with the lowest chance of success. Consider whether possible outcomes justify the implementation expenses for each decision. Eliminate choices that carry a high cost or a lot of risk without a significant outcome.
- Consider the remaining decisions. Refer to the "Final Outcomes" list for reference. Select the path leading to a significant final outcome that has a high chance of succeeding.

4.9 BAYES CLASSIFICATION

- A **Naive Bayes Classifier** is a simple probabilistic classifier based on applying Bayes theorem (from Bayesian statistics) with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model".
- The Bayesian Classification represents a supervised learning method as well as a statistical method for classification. Assumes an underlying probabilistic model and it allows us to capture uncertainty about the model in a principled way by determining probabilities of the outcomes. It can solve diagnostic and predictive problems.
- This Classification is named after Thomas Bayes (1702–1761), who proposed the Bayes Theorem.
- Bayesian classification provides practical learning algorithms and prior knowledge and observed data can be combined. Bayesian Classification provides a useful perspective for understanding and evaluating many learning algorithms. It calculates explicit probabilities for hypothesis and it is robust to noise in input data.
- In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 4" in diameter.
- Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple.

- Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without believing in Bayesian probability or using any Bayesian methods.
- In spite of their naive design and apparently over–simplified assumptions, naive Bayes classifiers have worked quite well in many complex real–world situations. In 2004, analysis of the Bayesian classification problem has shown that there are some theoretical reasons for the apparently unreasonable efficacy of naive Bayes classifiers. Still, a comprehensive comparison with other classification methods in 2006 showed that Bayes classification is outperformed by more current approaches, such as boosted trees or random forests.
- An advantage of the naive Bayes classifier is that it only requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix.

The Naive Bayes Probabilistic Model

- Abstractly, the probability model for a classifier is a conditional model

$$p(C|F_1, \ldots, F_n)$$

over a dependent class variable C with a small number of outcomes or classes, conditional on several feature variables through F_1 to F_n. The problem is that if the number of features n is large or when a feature can take on a large number of values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more tractable.

- Using Bayes' theorem, we write

$$p(C \mid F_1, \ldots, F_n) = \frac{p(C)\ p(F_1, \ldots, F_n|C)}{p(F_1, \ldots, F_n)}$$

In plain English the above equation can be written as

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

- In practice we are only interested in the numerator of that fraction, since the denominator does not depend on and the values of the features are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model.

$$p(C, F_1, \ldots, F_n)$$

which can be rewritten as follows, using repeated applications of the definition of conditional probability

$$p(C, F_1, \ldots, F_n) = p(C)\ p(F_1, \ldots, F_n|C)$$

$$= p(C)\ p(F_1|C)\ p(F_2, \ldots, F_n|C, F_1)$$

$$= p(C)\ p(F_1|C)\ p(F_2, F_1)\ p(F_3, \ldots, F_1, F_2)\ p(F_4 \ldots, F_n|C, F_1, F_2, F_3)$$

$$= p(C)\ p(F_1|C)\ p(F_2|C, F_1)\ p(F_3|C, F_1, F_2) \ldots p(F_n|C, F_1, F_2, F_3, \ldots F_{n-1})$$

- Now the "native" conditional independence assumptions come into play : assume that each feature F_i is conditionally independent of every other feature F_j for $j \neq i$. This means that

$$p(F_i \mid C, F_j) = p(F_i \mid C)$$

for $i \neq j$, and so the joint model can be expressed as

$$p(C, F_1, \ldots, F_n) = p(C)\ p(F_1|C)\ p(F_2|C)\ p(F_3|C)$$

for i = j, and so the joint model can be expressed as

$$p(C, F_1, \ldots, F_n) = p(C)\ p(F_1|C)\ p(F_2C)\ p(F_3|C) \ldots$$

$$= p(C) \prod_{t=1}^{n} p(F_i \mid C)$$

- This means that under the above independence assumptions, the distribution over the class variable C can be expressed like this :

$$p(C \mid F_1, \ldots, F_n) = \frac{1}{Z} p(C) \prod_{t=1}^{n} p(F_i|C)$$

where Z (the evidence) is a scaling factor dependent only on $F_1, \ldots, F_n$, i.e. a constant if the values of the feature variables are known

- Models of this form are much more manageable, since they factor into a so-called class prior p(C) and independent probability distributions $p(F_i|C)$. If there are k classes and if a model for each $p(F_i|C = c)$ can be expressed in terms of r parameters, then the corresponding native Bayes model has $(k - 1) + nrk$ parameters. In practice, often k = 2 (binary classification) and r = 1 (Bernoulli variables as features) are common, and so the total number of parameters of the native Bayers model is 2n + 1, where model is 2n + 1, where n is the number of binary features used for classification and prediction.

Parameter Estimation

- All model parameters (i.e. class priors and feature probability distributions) can be approximated with relative frequencies from the training set. These are maximum likelihood estimates of the probabilities. A class prior may be calculated by assuming equiprobable classes (i.e. (prior for a given class) = (number of samples in the class) / (total number of samples)).
- To estimate the parameters for a features from the training set. If one is dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution.

- For example, suppose the training data contains a continuous attribute, x. We first segment the data by the class, and then compute the mean and variance of x in each class. Let μ_c be the mean of the values in x associated with class c, and let σ_c^2 be the variance of the values of the values in x associated with class c. Then, the probability of some value given a class, P (x = v/c), can be computed by plugging v into the equation for a Normal distribution parameterized by μ_c and σ_c^2 That is,

$$p(x = v|c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{-\frac{(v-\mu_c)^2}{2\sigma_c^2}}$$

- Another common technique for handling continuous values if to use binning to discretize the values. In general, the distribution method is a better choice if there is small amount of training data, or if the precise distribution of the data is known. The discretization method tends to do better if there is a large amount of training data because it will learn to fit the distribution of the data.
- Since naive Bayes is typically used when a large amount of data is available (as more computationally expensive models can generally achieve better accuracy), the discretization method is generally preferred over the distribution method.

Sample Correction

- If a given class and feature value never occur together in the training set then the frequency–based probability estimate will be zero. This is problematic since it will wipe out all information in the other probabilities when they are multiplied.
- It is therefore often desirable to incorporate a small–sample correction in all probability estimates such that no probability is ever set to be exactly zero.

Constructing a Classifier from the Probability Model

- The discussion so far has derived the independent feature model, that is, the naive Bayes probability model. The naive Bayes classifier 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 is the function defined as follows :

$$\text{classify}(f_1, \ldots, f_n) = \operatorname{argmax}_c p(C = c) \prod_{i=1}^{n} p(F_i = f_i \mid C = c)$$

Naive Bayes Classification Example

It is based on the Bayesian theorem It is particularly suited when the dimensionality of the inputs is high. Parameter estimation for naive Bayes models uses the method of maximum likelihood. In spite over–simplified assumptions, it often performs better in many complex real world situations

Advantage : Requires a small amount of training data to estimate the parameters

Rec	Age	Income	Student	Credit_rating	Buys_computer
r1	<=30	High	No	Fair	No
r2	<=30	High	No	Excellent	No
r3	31…40	High	No	Fair	Yes
r4	>40	Medium	No	Fair	Yes
r5	>40	Medium	No	Fair	Yes
r6	>40	Low	Yes	Excellent	No
r7	31–40	Low	Yes	Excellent	Yes
r8	<=30	Medium	No	Fair	No
r9	<=30	Low	Yes	Fair	Yes
r10	>40	Medium	Yes	Fair	Yes
r11	<=30	Medium	Yes	Excellent	Yes
r12	32…40	Medium	No	Excellent	Yes
r13	31…40	High	Yes	Fair	Yes
r14	>40	Medium	No	Excellent	No

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

So here classification problem is a person belonging to tuple X will buy a computer?

Derivation :

D : Set of tuples

Where each Tuple is an 'n' dimensional attribute vector

X : $(x_1, x_2, x_3, \ldots x_n)$

Let there be 'm' Classes : $C_1, C_2, C_3 \ldots C_m$

Naïve Bayes classifier predicts X belongs to Class Ci if and only if

$$P(C_i/X) > P(C_j/X) \text{ for } 1 <= j <= m$$

Maximum Posteriori Hypothesis

$$P(C_i/X) = P(X/C_i)\, P(C_i) / P(X)$$

Maximize $P(X/C_i)\, P(C_i)$ as P(X) is constant

With many attributes, it is computationally expensive to evaluate $P(X/C_i)$.

Naive Assumption of "class conditional independence"

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

$$P(X/C_i) = P(x_1/C_i) \times P(x_2/C_i) \times \ldots \times P(x_n/C_i)$$

Theory applied on previous example :

$$P(C_1) = P(\text{buys\_computer} = \text{yes}) = \frac{9}{14} = 0.643$$

$$P(C_2) = P(\text{buys\_computer} = \text{no}) = \frac{5}{14} = 0.357$$

$$P(\text{age=youth/buys\_computer} = \text{yes}) = \frac{2}{9} = 0.222$$

$P(age=youth / buys\_computer = no) = \frac{3}{5} = 0.600$

$P(income=medium / buys\_computer = yes) = \frac{4}{9} = 0.444$

$P(income=medium / buys\_computer = no) = \frac{2}{5} = 0.400$

$P(student=yes / buys\_computer = yes) = \frac{6}{9} = 0.667$

$P(student=yes / buys\_computer = no) = \frac{1}{5} = 0.200$

$P(credit\ rating=fair / buys\_computer = yes) = \frac{6}{9} = 0.667$

$P(credit\ rating=fair / buys\_computer = no) = \frac{2}{5} = 0.400$

$P(X/Buys\ a\ computer = yes) = P(age=youth / buys\_computer = yes) \times P(income=medium / buys\_computer = yes) \times P(student=yes / buys\_computer = yes) \times P(credit\ rating = fair / buys\_computer = yes) = 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$

$P(X/Buys\ a\ computer = No) = 0.600 \times 0.400 \times 0.200 \times 0.400 = 0.019$

Find class Ci that Maximizes $P(X/G) \times P(G)$

$=> P(X/Buys\ a\ computer = yes) * P(buys\_computer = yes) = 0.028$

$=> P(X/Buys\ a\ computer = No) * P(buys\_computer = no) = 0.007$

Prediction : Buys a computer for Tuple X

Another example of Naïve Bayes classification

The Bayes Naive classifier selects the most likely classification V_{nb} given the attribute values $a_1, a_2, \ldots\ldots, a_n$.

This results in :

$$V_{nb} = \arg\max_{v_j} P(v_j) \prod P(a_i \mid v_j)$$

We generally estimate $P(a_i|v_j)$ using m–estimates :

$$P(a_i \mid v_j) = \frac{n_c + mp}{n + m}$$

where :

n = the number of training examples for which v = vj

nc = number of examples for which v = vj and a = ai

p = a priori estimate for $P(a_i \mid v_j)$

m = the equivalent sample size

Car Theft Example

Attributes are Color, Type, Origin, and the subject, stolen can be either yes or no.

Data Set

Example No.	Color	Type	Origin	Stolen?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

Training Example

We want to classify a Red Domestic SUV. Note there is no example of a Red Domestic SUV in our data set. We need to calculate the probabilities

P(Red|Yes), P(SUV|Yes), P(Domestic|Yes),

P(Red|No) , P(SUV|No), and P(Domestic|No)

and multiply them by P(Yes) and P(No) respectively. Estimation of these values is done as follows :

Yes	No
Red	Red
n = 5	n = 5
n_c = 3	n_c = 2
p = .5	p = .5
m = 3	m = 3
SUV	SUV
n = 5	n = 5
n_c = 1	n_c = 3
p = .5	p = .5
m = 3	m = 3
Domestic	Domestic
n = 5	n = 5
n_c = 2	n_c = 3
p = .5	p = .5
m = 3	m =3

Looking at P(Red|Yes), we have 5 cases where y = Yes , and in 3 of those cases a_i = Red. So for P(Red|Yes), n = 5 and n_c = 3. Note that all attribute are binary (two possible values). We are assuming no other information so, p = 1 / (number–of–attribute–

values) = 0.5 for all of our attributes. Our m value is arbitrary, (We will use m = 3) but consistent for all attributes. Now using the pre–computed values of n, nc, p, and m.

$$P(\text{Red} \mid \text{Yes}) = \frac{3 + 3 \times 0.5}{5 + 3} = 0.56$$

$$P(\text{Red} \mid \text{No}) = \frac{2 + 3 \times 0.5}{5 + 3} = 0.43$$

$$P(\text{SUV} \mid \text{Yes}) = \frac{1 + 3 \times 0.5}{5 + 3} = 0.31$$

$$P(\text{SUV} \mid \text{No}) = \frac{3 + 3 \times 0.5}{5 + 3} = 0.56$$

$$P(\text{Domestic} \mid \text{Yes}) = \frac{2 + 3 \times 0.5}{5 + 3} = 0.43$$

$$P(\text{Domestic} \mid \text{No}) = \frac{3 + 3 \times 0.5}{5 + 3} = 0.56$$

We have P(Yes) = .5 and P(No) = 0.5, so we can apply equation. For v = Yes, we have

$$P(\text{Yes}) \times P(\text{Red} \mid \text{Yes}) \times P(\text{SUV} \mid \text{Yes}) \times P(\text{Domestic} \mid \text{Yes})$$
$$= 0.5 \times 0.56 \times .31 \times 0.43 = 0.037$$

and for v = No, we have

$$P(\text{No}) \times P(\text{Red} \mid \text{No}) \times P(\text{SUV} \mid \text{No}) \times P(\text{Domestic} \mid \text{No})$$
$$= 0.5 \times 0.43 \times 0.56 \times 0.56 = 0.069$$

Since 0 : 069 > 0 :037, our example gets classified as 'NO'

4.10 SMOOTHING

- The use of an algorithm to remove noise from a data set, allowing important patterns to stand out. Data smoothing can be done in a variety of different ways, including random, random walk, moving average, simple exponential, linear exponential and seasonal exponential smoothing. Data smoothing can be used to help predict trends, such as trends in securities prices.
- Data smoothing techniques are used to eliminate "noise" and extract real trends and patterns. Below are some of the available smoothing methods.

Random

- This method is best when each period's data has no relationship to the pattern in the previous data. Under this condition, the best prediction for the next value in a series is simply the average of all previous data points.

$$y'_t = \frac{1}{k-1} \sum_{j=1}^{t-1} y_j$$

Random Walk

A random walk exists if the next data point is equal to the last data point plus some random deviation. Many financial securities move in this manner. Under this condition, the best prediction for the next value in a series is simply the last value.

$$y'_t = y_{t-1}$$

Moving Average

This method works well if the data contains no trend or cyclic pattern.

$$y'_t = \frac{1}{n} \sum_{j=t-n}^{t-1} y_j$$

n is a user-supplied constant greater than zero defining the number of consecutive points to average. Higher values cause greater smoothing.

Simple Exponential Smoothing

This method works well if the data contains no trend or cyclic pattern and the most recent data points are more significant than earlier points.

$$y'_t = ay_{t-1} + (1 - a)y'_{t-1}$$

a is the smoothing constant.

Linear Exponential Smoothing (Holt's Method)

This method works well if the data contains a trend but no cyclic pattern.

$$y'_t = ay_{t-1} + (1 - a)(y'_{t-1} + t_{t-1})$$
$$t_t = b(y'_t - y'_{t-1}) + (1 - b)t_{t-1}$$

a is the level smoothing constant and *b* is the trend smoothing constant.

Seasonal Exponential Smoothing (Winter's method)

This method works well if the data contains a trend and a cyclic pattern.

$$y'_t = a(y_{t-1} / s_{t-p}) + (1 - a)(y'_{t-1} + t_{t-1})$$
$$t_t = b(t'_t - y'_{t-1}) + (1 - b)t_{t-1}$$
$$s_t = c(y_{t-1} / y'_t) + (1 - c)s_{t-p}$$

a is the level smoothing constant, *b* is the trend smoothing constant, *c* is the seasonal smoothing constant, and *p* is the season period.

4.11 DIAGNOSTICS

- Naïve Bayes classifiers are capable to handle missing values which is not possible to logistic regression.
- It is also considered as robust regarding the irrelevant variables, which are scattered between all the classes whose impacts are not pronounced.
- Even if libraries are not available, it is easy to implement the Naïve Bayes classifier.
- The base of prediction is counting the occurrences of events, making the classifier competent to execute.
- Naïve Bayes is considered as efficient computationally and can also handle high-dimensional data in an efficient manner.
- By studying, it is indicate that the naive Bayes classifier in number of cases is competitive with several learning algorithms such as decision trees and neural networks.

- In come of the cases, output of naïve Bayes is superior to other methods.
- Naïve Bayes classifiers are able to handle categorical variables, which is not possible to logistic regression.
- By handling categorical variables it is also possible in decision tree, but increase in number of levels may result in a deep tree.
- Overall performance of naïve Bayes classifier is more better than decision trees regarding the categorical values with many levels.
- Naïve Bayes is more resistant to overfitting as compared to decision trees, particularly when there is involvement of a smoothing technique.
- The variables in the data are considered conditionally independent. Hence, correlating variables becomes sensitive since the algorithm may double the count effects.
- For example consider the individuals having low income and low credit tend to default, if there is any requirement of scoring "default" based on both income as well as credit as two different attributes, there may be double-counting effect in naïve Bayes on the default outcome. which will affect the accuracy of the prediction.
- Usually the naïve Bayes classifiers are not considered as very trustworthly for probability estimation and their use should be limited to assigning class labels. Even if probabilities are offered as part of the output for the purpose of prediction.
- We can use the Naïve Bayes only with categorical variables. If there are nay continuous variables then they must be converted into categorical variables by using the process called as discretization.

4.12 DIAGNOSTICS OF CLASSIFIERS

Performance evaluation metrics treat every class equally. There is no biasing. However it may not be suitable for analyzing imbalanced dataset, where the rare class is considered as more important than all other classes. For binary classification problem (binary class – only two classes are present in which we want to classify the instances), the rare class is often called as positive class, while other class is known as negative class.

- **Confusion Matrix**

Confusion matrix contains information about actual and predicted classifications done by a classification system.

A Confusion Matrix for Binary Classification Problem

	Predicted Class	
Actual Class	TP	FN
	FP	TN

Confusion Matrix for Crime – Suspect Identification

A	B	Classified as
173	52	Suspect
12	63	Innocent

- **Terminologies used in Confusion Matrix**
  - **True Positive (TP)**

    If the outcome from a prediction is p and the actual value is also p, then it is called as true positive.

    It is calculated as diagonal element / sum of relevant row

    The counts in the confusion matrix can also be expressed in terms of percentage.

    True positive rate which is also known as sensitivity is given as :

    $$TPR = TP / (TP + FN)$$

  - **False Negative (FN)**

    It corresponds to number of positive instances which are wrongly predicted as negative.

    False negative rate is the fraction of positive instances predicted as a negative class.

    $$FNR = FN / (TP + FN)$$

  - **True Negative (TN)**

    It corresponds to number of negative instances which are correctly predicted as negative.

    True negative rate which is also known as specificity is defined as fraction of negative instances correctly classified as negative class.

    $$TNR = TN / (TN + FP)$$

  - **False Positive (FP)**

    If the outcome from a prediction is p and the actual value is n, then it is called as false positive.

    It is calculated as diagonal element / sum of relevant row

    It corresponds to number of negative instances which are wrongly predicted as positive.

    False positive rate which is also known as specificity is defined as fraction of negative instances correctly classified as positive class.

    $$FPR = FP / (TN + FP)$$

  - **For Class Suspect**

    J48 classifies 173 suspects as true positives out of 225 so for class suspect and J48 algorithm true positive rate becomes 0.769.

  - **For Class Innocent**

    J48 classifies 63 innocent as true negatives out of 75 so for class innocent and J48 algorithm true negative rate becomes 0.840

- **For Class Suspect**

  J48 misclassifies 12 suspects as false positives out of 75 so for class suspect and J48 algorithm false positive rate becomes 0.160.

- **For Class Innocent**

  J48 classifies 52 innocent as false negatives out of 225 so for class innocent and J48 algorithm false negative rate becomes 0.231.

- **Precision and Recall**

  Precision and recall are most widely used performance metrics. They are used in a classification problem when successful detection of one of the classes is more significant than detection of other classes.

  Precision is the fraction of retrieved instances that are relevant. It determines the fraction of instances that are actually positive and classifier has also classified them as positive. Higher the precision, lower the number of false positive errors committed by classifier.

  Precision is basically the measure of exactness or quality.

  It is calculated as diagonal element / sum of relevant column.

  Precision (P) = TP / (TP + FP)

  For class suspect precision value is 0.935.

  For class innocent precision value is 0.548.

- **Recall**

  Recall is the fraction of relevant instances that are retrieved. It measures the fraction of positive examples correctly predicted by the classifier.

  Recall is basically the measure of completeness.

  Recall is same as true positives for both classes.

  Recall (R) = TP / (TP+FN)

  F_measure

  F_measure is used when both Precision and Recall are important to measure accuracy.

  F_measure = 2*precision*recall/ (precision + recall)

  F_measure = 2*P*R / (P + R)

  For class suspect F_measure is 0.844 and for class innocent F_measure is 0.663.

- **Mathew's Correlation Coefficient (MCC)**

  MCC is a factor which indicates accuracy of final class predictions for bi–variant classification problem. Its value ranges from –1 to +1. –1 value indicates the complete disagreement between observed and predicted values, 0 indicates random prediction and +1 indicates perfect prediction. Value of MCC in our classification problem is 0.542.

Table 4.4 : Performance Metric for Crime Suspect Identification Problem

Factor\Class	Suspect	Innocent
TP Rate	0.769	0.840
FP Rate	0.160	0.231
Precision	0.935	0.548
Recall	0.769	0.840
F_measure	0.844	0.663
MCC	0.542	0.542

4.13 ADDITIONAL CLASSIFICATION METHODS

There are two forms of data analysis that can be used for extracting models describing important classes or to predict future data trends.

These Two Forms are as Follows :

1. Classification
2. Prediction

Classification models predict categorical class labels; and prediction models predict continuous valued functions. For example, we can build a classification model to categorize bank loan applications as either safe or risky, or a prediction model to predict the expenditures in dollars of potential customers on computer equipment given their income and occupation.

What is Classification?

Following are the examples of cases where the data analysis task is Classification –

- A bank loan officer wants to analyze the data in order to know which customer (loan applicant) are risky or which are safe.
- A marketing manager at a company needs to analyze a customer with a given profile, who will buy a new computer.

In both of the above examples, a model or classifier is constructed to predict the categorical labels. These labels are risky or safe for loan application data and yes or no for marketing data.

What is Prediction?

Following are the examples of cases where the data analysis task is Prediction –

Suppose the marketing manager needs to predict how much a given customer will spend during a sale at his company. In this example we are bothered to predict a numeric value. Therefore the data analysis task is an example of numeric prediction. In this case, a model or a predictor will be constructed that predicts a continuous-valued-function or ordered value.

Note : Regression analysis is a statistical methodology that is most often used for numeric prediction.

How Does Classification Works?

With the help of the bank loan application that we have discussed above, let us understand the working of classification.

The Data Classification Process Includes Two Steps :

1. Building the Classifier or Model
2. Using Classifier for Classification

1. Building the Classifier or Model

- This step is the learning step or the learning phase.
- In this step the classification algorithms build the classifier.
- The classifier is built from the training set made up of database tuples and their associated class labels.
- Each tuple that constitutes the training set is referred to as a category or class. These tuples can also be referred to as sample, object or data points.

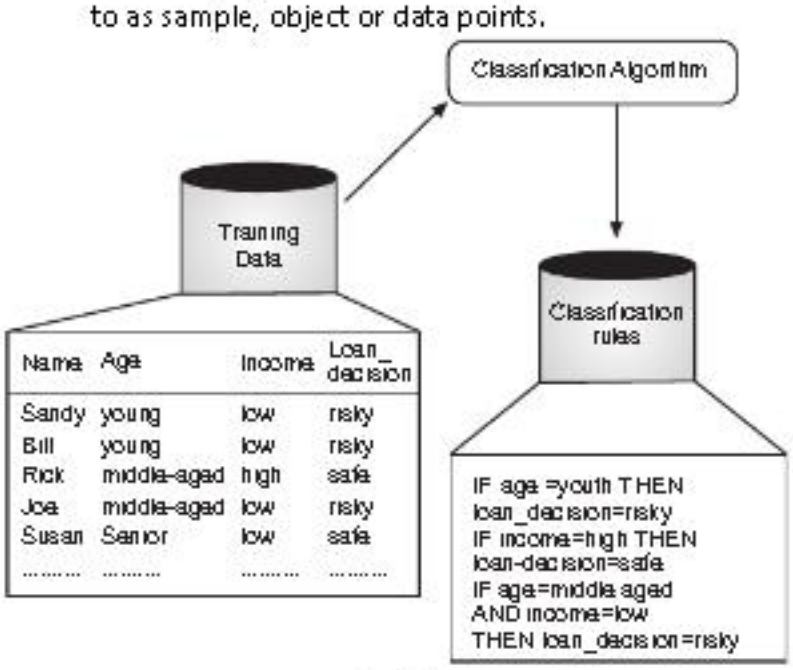


Fig. 4.7

2. Using Classifier for Classification

- In this step, the classifier is used for classification. Here the test data is used to estimate the accuracy of classification rules. The classification rules can be applied to the new data tuples if the accuracy is considered acceptable.

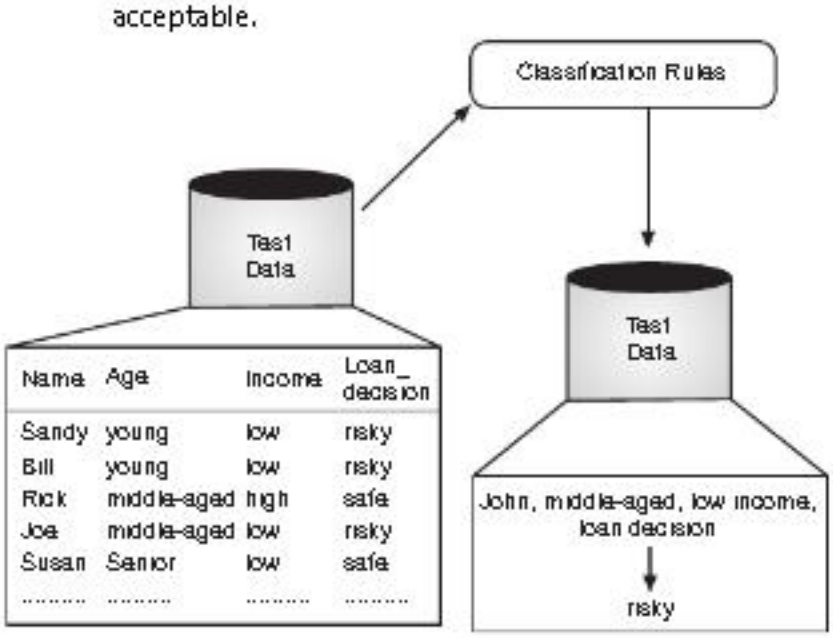


Fig. 4.8

- Classification techniques in data mining are capable of processing a large amount of data. It can be used to predict categorical class labels and classifies data based on training set and class labels and it can be used for classifying newly available data. The term could cover any context in which some decision or forecast is made on the basis of presently available information.
- Classification procedure is recognized method for repeatedly making such decisions in new situations. Here if we assume that problem is a concern with the construction of a procedure that will be applied to a continuing sequence of cases in which each new case must be assigned to one of a set of pre defined classes on the basis of observed features of data. Creation of a classification procedure from a set of data for which the exact classes are known in advance is termed as pattern recognition or supervised learning. Contexts in which a classification task is fundamental include, for example, assigning individuals to credit status on the basis of financial and other personal information, and the initial diagnosis of a patient's disease in order to select immediate treatment while awaiting perfect test results. Some of the most critical problems arising in science, industry and commerce can be called as classification or decision problems. Three main historical strands of research can be identified: statistical, machine learning and neural network. All groups have some objectives in common. They have all attempted to develop procedures that would be able to handle a wide variety of problems and to be extremely general used in practical settings with proven success.

Statistical Procedure Based Approach

- Two main phases of work on classification can be identified within the statistical community. The first "classical" phase concentrated on extension of Fisher's early work on linear discrimination. The second, "modern" phase concentrated on more flexible classes of models many of which attempt to provide an estimate of the joint distribution of the features within each class which can in turn provide a classification rule.
- Statistical procedures are generally characterized by having an precise fundamental probability model which provides a probability of being in each class instead of just a classification.
- Also it is usually assumed that the techniques will be used by statisticians and hence some human involvement is assumed with regard to variable selection and transformation and overall structuring of the problem.

Machine Learning Based Approach

- Machine Learning is generally covers automatic computing procedures based on logical or binary operations that learn a task from a series of examples.

- Here we are just concentrating on classification and so attention has focused on decision-tree approaches in which classification results from a sequence of logical steps.
- These classification results are capable of representing the most complex problem given sufficient data. Other techniques such as genetic algorithms and inductive logic procedures (ILP) are currently under active improvement and its principle would allow us to deal with more general types of data including cases where the number and type of attributes may vary.
- Machine Learning approach aims to generate classifying expressions simple enough to be understood easily by the human and must mimic human reasoning sufficiently to provide insight into the decision process. Like statistical approaches background knowledge may be used in development but operation is assumed without human interference.

4.13.1 Neural Network

- The field of Neural Networks has arisen from diverse sources ranging from understanding and emulating the human brain to broader issues of copying human abilities such as speech and can be use in various fields such as banking, legal, medical, news, in classification program to categories data as intrusive or normal.
- Generally neural networks consist of layers of interconnected nodes where each node producing a non-linear function of its input and input to a node may come from other nodes or directly from the input data. Also, some nodes are identified with the output of the network.
- On the basis of this example there are different applications for neural networks that involve recognizing patterns and making simple decisions about them.
- In airplanes we can use a neural network as a basic autopilot where input units reads signals from the various cockpit instruments and output units modifying the plane's controls appropriately to keep it safely on course. Inside a factory we can use a neural network for quality control.

4.13.2 Classification Algorithms

- Classification is one of the Data Mining techniques that is mainly used to analyze a given data set and takes each instance of it and assigns this instance to a particular class such that classification error will be least.
- It is used to extract models that accurately define important data classes within the given data set. Classification is a two step process.
- During first step the model is created by applying classification algorithm on training data set then in second step the extracted model is tested against a predefined test data set to measure the model trained performance and accuracy. So classification is the process to assign class label from data set whose class label is unknown.

(A) ID3 Algorithm

- ID3 calculation starts with the original set as the root hub. On every cycle of the algorithm it emphasizes through every unused attribute of the set and figures the entropy (or data pick up IG(A)) of that attribute. At that point chooses the attribute which has the smallest entropy (or biggest data gain) value. The set is S then split by the selected attribute (e.g. marks < 50, marks < 100, marks >= 100) to produce subsets of the information. The algorithm proceeds to recurs on each and every item in subset and considering only items never selected before.

Recursion on a Subset may bring to a Halt in One of these Cases :

- Every element in the subset belongs to the same class (+ or -), then the node is turned into a leaf and labelled with the class of the examples
- If there are no more attributes to be selected but the examples still do not belong to the same class (some are + and some are -) then the node is turned into a leaf and labelled with the most common class of the examples in that subset.
- If there are no examples in the subset, then this happens when parent set found to be matching a specific value of the selected attribute. For example if there was no example matching with marks >=100 then a leaf is created and is labelled with the most common class of the examples in the parent set.

Working Steps of Algorithm is as Follows :

- Calculate the entropy for each attribute using the data set S.
- Split the set S into subsets using the attribute for which entropy is minimum (or, equivalently, information gain is maximum)
- Construct a decision tree node containing that attribute in a dataset.
- Recurse on each member of subsets using remaining attributes.

(B) C4.5 Algorithm

- C4.5 is an algorithm used to produce a decision tree which is an expansion of prior ID3 calculation. It enhances the ID3 algorithm by managing both continuous and discrete properties, missing values and pruning trees after construction. The decision trees created by C4.5 can be used for grouping and often referred to as a statistical classifier.
- C4.5 creates decision trees from a set of training data same way as Id3 algorithm. As it is a supervised learning algorithm it requires a set of training examples which can be seen as a pair: input object and a desired output value (class).
- The algorithm analyzes the training set and builds a classifier that must have the capacity to accurately arrange both training and test cases. A test example is an input object and the algorithm must predict an output value.

- Consider the sample training data set $S = S_1, S_2, \ldots S_n$ which is already classified. Each sample S_i consist of feature vector $(x_{1i}, x_{2i}, \ldots, x_{ni})$ where x_j represent attributes or features of the sample and the class in which S_i falls.
- At each node of the tree C4.5 selects one attribute of the data that most efficiently splits its set of samples into subsets such that it results in one class or the other. The splitting condition is the normalized information gain (difference in entropy) which is a non-symmetric measure of the difference between two probability distributions P and Q. The attribute with the highest information gain is chosen to make the decision.

General Working Steps of Algorithm is as Follows :

1. Assume all the samples in the list belong to the same class. If it is true, it simply creates a leaf node for the decision tree so that particular class will be selected.
2. None of the features provide any information gain. If it is true, C4.5 creates a decision node higher up the tree using the expected value of the class.
3. Instance of previously-unseen class encountered. Then, C4.5 creates a decision node higher up the tree using the expected value.

(C) C. K. Nearest Neighbors Algorithm

- The closest neighbor (NN) rule distinguishes the classification of unknown data point on the basis of its closest neighbor whose class is already known. M. Cover and P. E. Hart purpose k nearest neighbour (KNN) in which nearest neighbor is computed on the basis of estimation of k that indicates how many nearest neighbors are to be considered to characterize class of a sample data point.
- It makes utilization of the more than one closest neighbor to determine the class in which the given data point belongs to and consequently it is called as KNN. These data samples are needed to be in the memory at the run time and hence they are referred to as memory-based technique.
- T. Bailey and A. K. Jain enhance KNN which is focused on weights. The training points are assigned weights according to their distances from sample data point. But at the same time the computational complexity and memory requirements remain the primary concern dependably.
- To overcome memory limitation size of data set is reduced. For this the repeated patterns which don't include additional data are also eliminated from training data set. To further enhance the information focuses which don't influence the result are additionally eliminated from training data set. The NN training data set can be organized utilizing different systems to enhance over memory limit of KNN.
- The KNN implementation can be done using ball tree, k-d tree, nearest feature line (NFL), principal axis search tree and orthogonal search tree. The tree structured training data is further divided into nodes and techniques like NFL and tunable metric divide the training data set according to planes.
- Using these algorithms we can expand the speed of basic KNN algorithm. Consider that an object is sampled with a set of different attributes. Assuming its group can be determined from its attributes; different algorithms can be used to automate the classification process.

In Pseudo Code K-Nearest Neighbor Classification Algorithm can be Expressed :

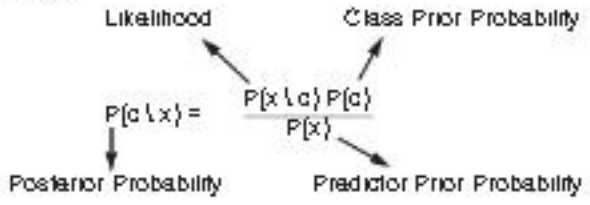
K ← number of nearest neighbors

- For each object X in the test set do calculate the distance D(X,Y) between X and every object Y in the training set neighborhood ← the k neighbors in the training set closest to X

X.class ← SelectClass (neighborhood) End for

Naïve Bayes Algorithm

- The Naive Bayes Classifier technique is based on Bayesian theorem and is particularly used when the dimensionality of the inputs is high. The Bayesian Classifier is capable of calculating the most possible output based on the input. It is also possible to add new raw data at runtime and have a better probabilistic classifier.
- A naive Bayes classifier considers that the presence (or absence) of a particular feature(attribute) of a class is unrelated to the presence (or absence) of any other feature when the class variable is given.
- For example, a fruit may be considered to be an apple if it is red, round. Even if these features depend on each other or upon the existence of other features of a class, a naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple. Algorithm works as follows,
- Bayes theorem provides a way of calculating the posterior probability, P(c|x), from P(c), P(x), and P(x|c). Naive Bayes classifier considers that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors.



$$P(c \setminus X) = P(x_1 \setminus c) \times P(x_2 \setminus c) \times \ldots\ldots \times P(x_n \setminus c) \times P(c)$$

- P(c|x) is the posterior probability of class (target) given predictor (attribute) of class.
- P(c) is called the prior probability of class.
  - P(x|c) is the likelihood which is the probability of predictor of given class.
  - P(x) is the prior probability of predictor of class.

SVM Algorithm

- SVM have attracted a great deal of attention in the last decade and actively applied to various domains applications. SVMs are typically used for learning classification, regression or ranking function.
- SVM are based on statistical learning theory and structural risk minimization principal and have the aim of determining the location of decision boundaries also known as hyperplane that produce the optimal separation of classes.
- Maximizing the margin and thereby creating the largest possible distance between the separating hyperplane and the instances on either side of it has been proven to reduce an upper bound on the expected generalization error.
- Efficiency of SVM based classification is not directly depend on the dimension of classified entities. Though SVM is the most robust and accurate classification technique, there are several problems.
- The data analysis in SVM is based on convex quadratic programming, and it is computationally expensive, as solving quadratic programming methods require large matrix operations as well as time consuming numerical computations.
- Training time for SVM scales quadratically in the number of examples, so researches strive all the time for more efficient training algorithm, resulting in several variant based algorithm.
- SVM can also be extended to learn non-linear decision functions by first projecting the input data onto a high-dimensional feature space using kernel functions and formulating a linear classification problem in that feature space.
- The resulting feature space is much larger than the size of dataset which are not possible to store in popular computers. Investigation on this issues leads to several decomposition based algorithms.
- The basic idea of decomposition method is to split the variables into two parts: set of free variables called as working set, which can be updated in each iteration and set of fixed variables, which are fixed at a particular value temporarily. This procedure is repeated until the termination conditions are met. Originally, the SVM was developed for binary classification, and it is not simple to extend it for multi-class classification problem. The basic idea to apply multi classification to SVM is to decompose the multi class problems into several two class problems that can be addressed directly using several SVMs.

ANN Algorithm

- Artificial Neural Networks (ANNs) are types of computer architecture inspired by biological neural networks (Nervous systems of the brain) and are used to approximate functions that can depend on a large number of inputs and are generally unknown. Artificial neural networks are presented as systems of interconnected "neurons" which can compute values from inputs and are capable of machine learning as well as pattern recognition due their adaptive nature.
- An artificial neural network operates by creating connections between many different processing elements each corresponding to a single neuron in a biological brain. These neurons may be actually constructed or simulated by a digital computer system. Each neuron takes many input signals then based on an internal weighting produces a single output signal that is sent as input to another neuron. The neurons are strongly interconnected and organized into different layers. The input layer receives the input and the output layer produces the final output. In general one or more hidden layers are sandwiched in between the two. This structure makes it impossible to forecast or know the exact flow of data.
- Artificial neural networks typically start out with randomized weights for all their neurons. This means that initially they must be trained to solve the particular problem for which they are proposed. A back-propagation ANN is trained by humans to perform specific tasks. During the training period, we can evaluate whether the ANN's output is correct by observing pattern. If it's correct the neural weightings that produced that output are reinforced; if the output is incorrect, those weightings responsible can be diminished.
- Implemented on a single computer, an artificial neural network is normally slower than more traditional solutions of algorithms. The ANN's parallel nature allows it to be built using multiple processors giving it a great speed advantage at very little development cost. The parallel architecture allows ANNs to process very large amounts of data very efficiently in less time. When dealing with large continuous streams of information such as speech recognition or machine sensor data ANNs can operate considerably faster as compare to other algorithms.
- An artificial neural network is useful in a variety of real-world applications such as visual pattern recognition and speech recognition that deal with complex often incomplete data. In addition, recent programs for text-to-speech have utilized ANNs. Many handwriting analysis programs (such as those used in popular PDAs) are currently using ANNs.

Table 4.5

Sr. No.	Algorithm	Features	Limitations
1.	C4.5 Algorithm	• Build models can be easily interpreted. • Easy to implement. • Can use both discrete and continuous values. • Deals with noise.	• Small variation in data can lead to different decision trees. • Does not work very well on a samll training data set. • Overfitting.
2.	ID3 Algorithm	• It produces the more accuracy result than the C4.5 algorithm. • Detection rate is increase and space consumption is reduced.	• Requires large seaching time. • Sometimes it may generate very long rules which are very hard to prune. • Requires large amount of memory to store tree.
3.	K-Nearest neighbor Algorithm	• Classes need not be linearly separable. • Zero cost of the learning process. • Sometimes it is Robust with regard to noisy traning data. • Well suited for multimodal classes.	• Time to find the neaest Neighbours in a large traning data set can be excessive. • It is sensitive to noisy or irrelevent attributes. • Perfocrmance of algorithm depends on the number of dimension used.
4.	Naive Bayes Algorithm	• Simple to implement. • Great computational efficienty and classification rate. • It predicts accurate results fpr most of the classification and prediction problems.	• The precision of algorithm decreses if the amount of data is less. • For obtaining good results it requires a very large number of records.
5.	Support vector machine algorithm	• High accuracy. • Work well even if data is not linearly separable in the base feature space.	• Speed and size requirement both in taining and testing is more. • High complexity and extensive memory requirements for classification in many cases.
6.	Artificial Neural Network Algorithm	• It is easy to use, with few parameters to adjust. • A neural network leans and reprogramming is not needed. • Easy to implement. • Applicable to a wide range of problems in real life.	• Requires high processing time if neural network is large. • Difficult to know how many neurons and layers are necessary. • Learning can be slow.

EXERCISE

1. Differentiate between supervised learning and unsupervised learning ?
2. What do you mean by classification ? What are the various requirements for classification ?
3. Explain decision tree classification algorithm with
   (a) Attribute selection phase
   (b) Tree pruning phase
4. Explain ID3 algorithm with suitable example
5. Define following terms
   (a) Entropy
   (b) Gini index
6. Explain various scalable decision tree algorithms ? How performance can be improved with the help of scalable decision tree algorithms
7. Explain linear regression with suitable example
8. What is Bayes' theorem ? Explain Naïve Bayes classification technique with suitable example
9. Explain following metrics of performance measurement
   (a) Confusion matrix
   (b) TP rate, FP rate, TN rate, FN rate
   (c) Precision and Recall
   (d) MCC

UNIT V

BIG DATA VISUALIZATION

5.1 INTRODUCTION TO DATA VISUALIZATION

- To understand current and future trends in the field of data visualization, it helps to begin with some historical context. Despite the fact that predecessors to data visualization date back to the 2nd century AD, most developments have occurred in the last two and a half centuries, predominantly during the last 30 years.
- Visualization is systematic pictorial representation of technique. Anything which is represented in graphical form with the help of diagrams, charts, pictures, flowcharts, etc. is called as Visualization.
- Data visualization is pictorial or graphical representation of data with the help of graphs, bar, histogram, tables, pie charts, minds maps etc. Depend on complexity of data and aspect of analysis, dimensions can vary (1D/2D/3D/n-D). All visuals are used to present different data sets.
- Now a day's visualization-based data discovery methods allow business users to combine different data sources to create custom analytical views.
- By using data visualization, business owners can understand their large data in a simple format. These methods are also time saving so business does not have to spend much time to make a report or solve a query.

Following Points Need to Consider for Data Visualization :

- **Do Not Forget the Metadata :** Data about data can be very revealing.
- **Participation Matters :** Visualization tools should be interactive and user participation is very important.
- **Encourage Interactivity :** Well interactive tools may lead to discovery.
- Conventional visualization, methods are not sufficient to represent big data so need to choose the most effective way to visualize the data to surface any insights it may contain. Big data are high volume, high velocity and high variety datasets that require new processing techniques to enable enhanced process optimization, insight discovery and decision making.

Visualization Pipeline

- The input is a collection of data that can be structured or unstructured.
- The data transformation and analysis module is tasked with extracting structured data from the input data. If the input data collection is too large to fit into computer memory, a data reduction technique is applied first.
- For unstructured data, some data mining techniques such as clustering or categorization can be adopted to extract related structure data for visualization.
- With the structured data, this module then removes noise by applying a smoothing filter, interpolating missing values, or correcting erroneous measurements.
- The output of this module is then sent to the filtering module, which automatically or semi-automatically selects data portions to be visualized (focus data).
- Given the results produced by the filtering module, the mapping module maps the focus data to geometric primitives (e.g., points, lines) and their attributes (e.g., color, position, size).With the rendering module, geometric data are transformed into image data.
- Users can then interact with the generated image data through various UI controls to explore and understand the data from multiple perspectives.

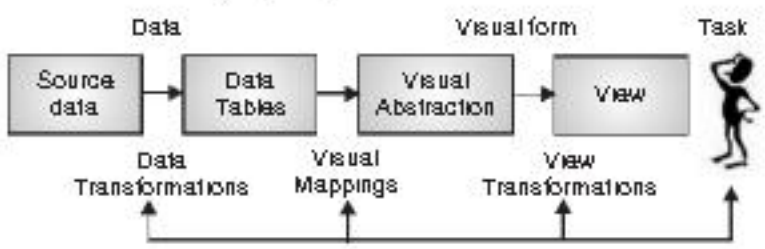


Fig. 5.1 : Visualization pipeline

5.2 CHALLENGES TO BIG DATA VISUALIZATION

- Big data visualization with diversity and heterogeneity i.e. structured, semi-structured, and unstructured is a big problem. Speed is an important aspect for the big data analysis.
- Combination of cloud computing advanced graphical user interface and big data can be used for the better management of big data scalability.

Following are the Challenges for Big Data Visualization :

1. **Visual Noise**
   - Most of the objects in dataset are too close to each other so users cannot divide them as separate objects on the screen.
   - For example, presentation of whole array of data can become a total mess on a screen. Sometimes we can see only one big spot, consisting of points, which represent each data row.

2. Large Image Perception

- Usually output of visualization methods is dependent on device resolution so there is a limit on the number of points to show per visualization.
- Here solution is to replace visualization device for a modern one or a group of devices for partial data visualization, allowing us to present a more detailed image with a larger number of data points.
- Repetition of this process for an infinite number of times, we will meet a human perception limitation.
- As data grows it is difficult for human being to understand data and perform data analysis so data visualization methods are limited not only by aspect ratio and resolution of device but also by physical perception limits.

3. Information Loss

- Reduction of visible data sets leads to of information loss.
- Such approaches can mislead the analyst, when he cannot notice some interesting hidden objects, and some complex aggregation process can consume a large amount of time and performance resources in order to get the accurate and required information.

4. High Performance Requirements

- The graphical analysis is not limited to only static image visualization but need for dynamic visualization. For dynamic data visualization, speed is required i.e. high performance requirement.

5. High Rate of Image Change

- Users observe data and cannot react to the number of data change or the intensity of displaying it.
- The simple decrease of changing rate cannot provide the desired result, as the reaction speed of the human being directly depends on it.

6. Data Quality

- In Big Data visualization only accurate data i.e. quality data is being visualized.
- If this data is inaccurate then appropriate people and processes need to be put in place to manage corporate data, metadata, data sources, and any transformations or data cleaning that are performed before storage.

7. Availability of Visualization Specialists

- Usually Big data visualization tools are designed in way so any one from organization can use it. But to retrieve all possible aspects and results organization needs to hire data visualization specialist.

8. Perceptual and Interactive Scalability

- Visualizing every data point may lead to over-plotting and may overwhelm users perceptual and cognitive capacities.
- Performing data reduction using sampling or filtering can hide some interesting feature or outliers.
- Querying large data stores can result in high latency, disrupting fluent interaction.

9. Parallelization

- Massive parallelization is a challenge in visualization due to size of data. The challenge in parallel visualization algorithms is decomposing a problem into independent tasks that can be run concurrently.

10. Visualization Hardware Resources

- Big Data visualization requires powerful computer hardware, fast storage systems, or even a move to cloud.

5.3 CONVENTIONAL DATA VISUALIZATION TOOLS

Interactive visualization can be performed through approaches such as zooming (zoom in and zoom out), overview and detail, zoom and pan, and focus and context or fish eye. The steps for interactive visualization are as follows

- **Selecting :** Interactive selection of data entities or subset or part of whole data or whole data set according to the user interest.
- **Linking :** It is useful for relating information among multiple views.
- **Filtering :** It helps users adjust the amount of information for display. It decreases information quantity and focuses on information of interest.
- **Rearranging or Remapping :** Because the spatial layout is the most important visual mapping, rearranging the spatial layout of the information is very effective in producing different insights.

Some Conventional Data Visualization Tools are as Follows :

Simple Table

- Simple table is structured representation to show data.
- Column in table specifies field, parameter, attribute, property etc and Rows in table have actual information.

Pie Charts

- A pie chart or circle graph as shown in Fig. 5.2, is divided into number of sectors, each circle describe a proportion in a whole quantity.
- Wedge in pie chart represents the part of data that has common feature or characteristics.
- It can be labeled to identify different data points generally shown in percentage.

- Two varieties of pie charts are Doughnut charts and Exploding pie charts.
- Doughnut charts are similar to standard pie charts except they have hollow center.
- While in Exploding charts wedges or segment or sector can be extracted from the rest of the wedges. These sectors or wedges provide good interactions Multi-level Pie, Radial tree, or Ring chart is another variation of Pie chart.

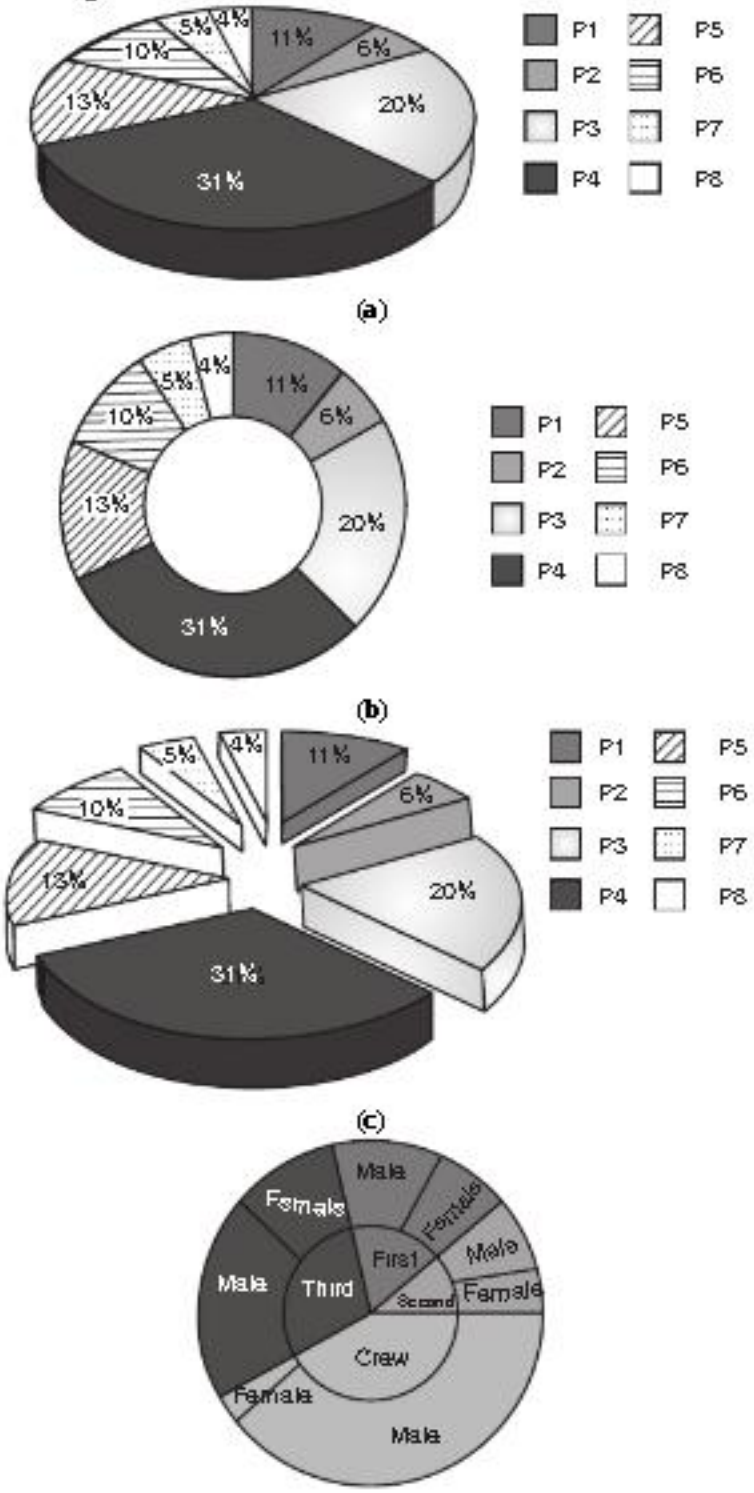
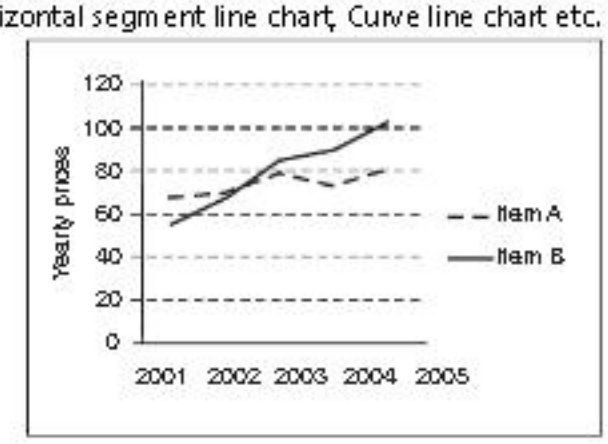


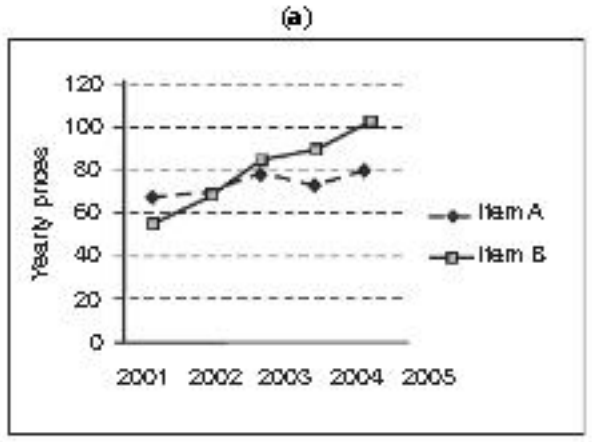
Fig. 5.2 : (a) Standard pie chart, (b) Doughnut pie chart, (c) Exploding pie chart, (d) Multi-level pie chart

Line Charts :

- Line chart as shown in Fig. 5.3 is used to display information in connected points. These points are connected through continuous or straight line. Data points can be represented by icons or symbols, or can also draw simple line without icons.
- The line chart is often used to visualize a trend in data over time interval, means it use to show tendency in the data set and illustrate the data behaviour with the passage of time or over a specific interval of time.
- There are many form or Variations of line chart or line graph, depends on the data points to be plot, for example; Step line chart, Reverse step line chart, vertical segment line chart, Horizontal segment line chart, Curve line chart etc.



(a)



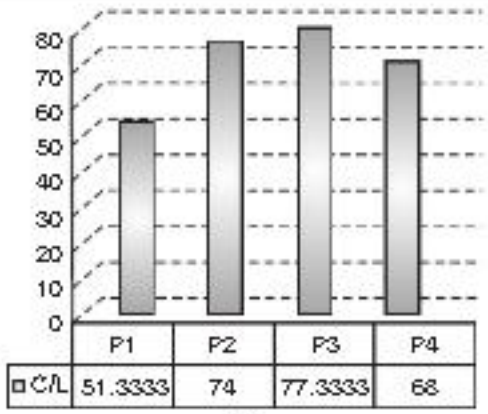
(b)

Fig. 5.3 : (a) Simple line chart,

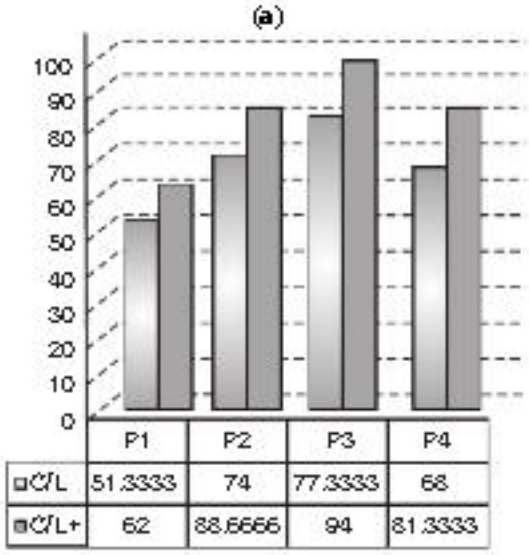
(b) Line chart with symbolic data point

Bar Charts :

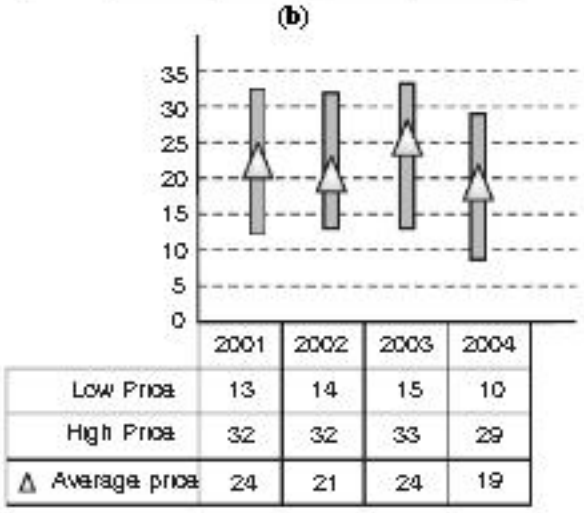
- Bar chart as shown in Fig. 5.4 is most of the time use for discrete data not for continuous data.
- Bar Chart represents data in horizontal bars, the vertical length of the bar represent the values.
- Bar chart is use to represent a single data series and related data points are group in one series. Another alternative of horizontal bar chart is vertical bar chart, which is use in the same way as horizontal bar chart.
- Floating column chart or High Low Open Close chart is another type of bar chart. In floating bar chart represent the low bound and high bound of the bar unlike simple bar chart where there is only high limit of the bar.
- Candlestick Chart is another type of bar chart, in which top and bottom vertical line represents the low and high values, while the filled box represent the range of opening and closing values.



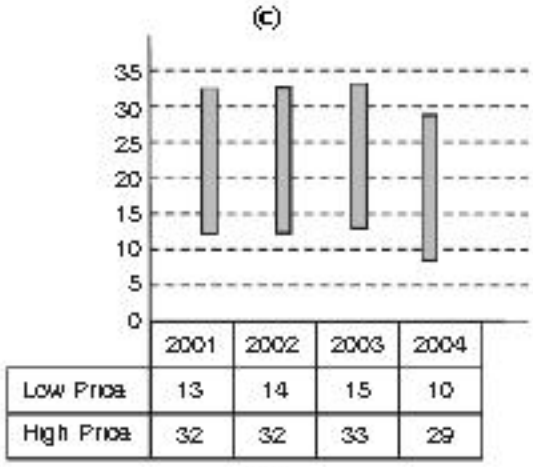
(a)



(b)



(c)



(d)

Fig. 5.4 : (a) Single bar chart, (b) Multi bar chart, (c) Floating bar chart, (d) High low open close bar chart

Histograms

- Histogram as shown in Fig. 5.5 is common and vital technique use in statistics and data analysis and is a graphical representation which represents the distribution of data.
- It is used for the distribution of continuous data.
- A histogram is the fall of specific items in class of variable, which consist of consecutive, non overlapping horizontal slabs.
- These slabs are mostly of the same size and adjacent, these may be of different sizes as well.

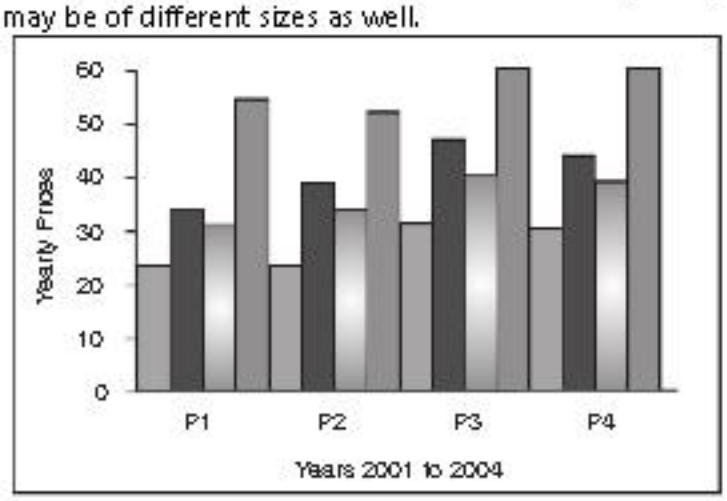
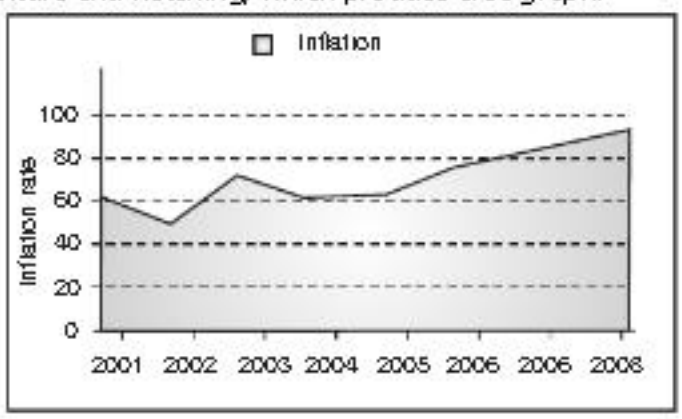


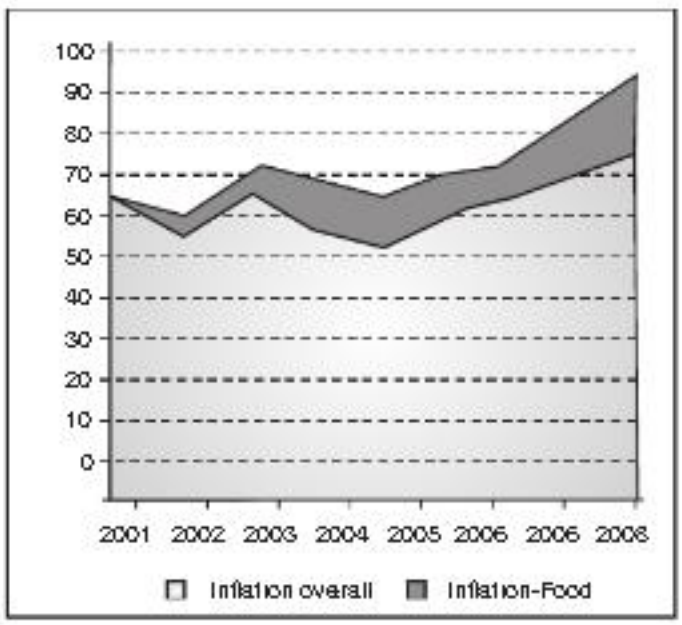
Fig. 5.5 : Histogram

Area Chart :

- Area charts as shown in Fig. 5.6 is also called area graph, use to display quantitative data graphically.
- Area chart control is use to represent data in bounded area.
- The bounded area is based on the line graph, the line is generated and the area below is shaded with colors, different texture and hatching, which produce area graph.



(a)



(b)

Fig. 5.6 : (a) Area chart, (b) Multi series area chart

Scatter Plot

- Scatter Plot as shown in Fig. 5.7 is also known as plot, plot chart, scatter chart, scatter gram, scatter diagram or scatter graph.
- Scatter plot is graphical display of set of data in Cartesian coordinate, shows the relationship between two variables, one variable represent horizontal distance (independent variable) and second variable vertical distance (dependent variable) of data point from the coordinate axis.
- Scatter plot shows how strong the relationship is between the variables, and determines whether their exits any outlier in the data or not.
- It is used to look at how the data is dispersed. Scatter plot is useful to determine trend in the data and identify outliers easily.

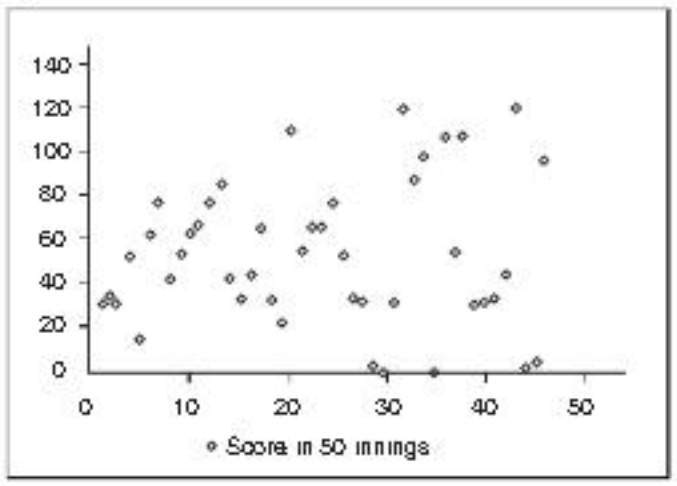
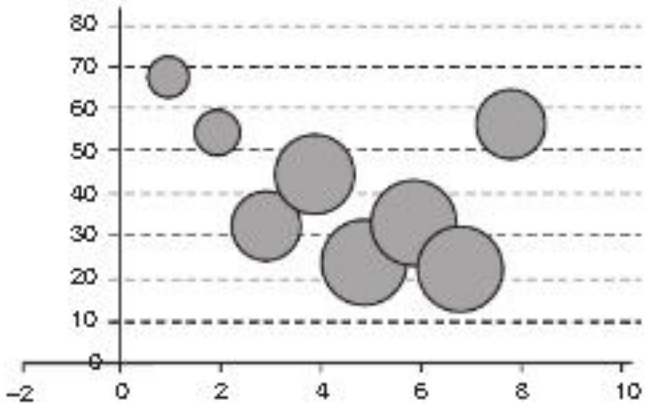


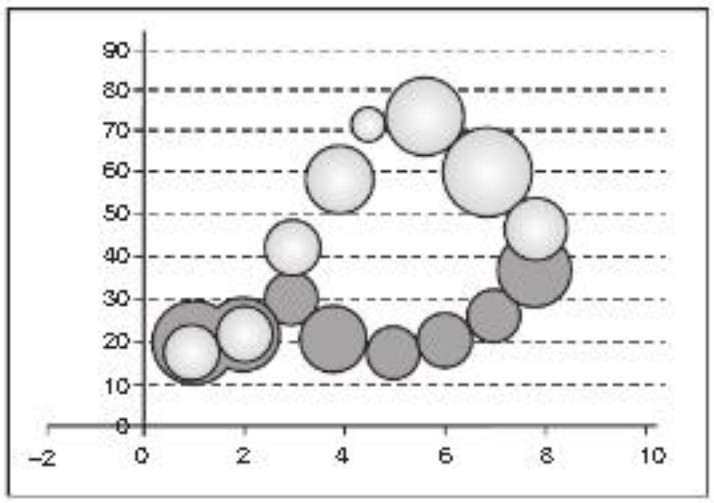
Fig. 5.7 : Scatter plot

Bubble Plots

- Bubble Plots as shown in Fig. 5.8 are defined by three different numeric parameters. These three numeric values represent data point i.e. one value determines its position along x-axis, one along y-axis, and third valuere present the size of the bubble in the chart.
- In bubble chart a bubble is differentiated from other bubbles in term of its size and in term of its position.



(a)



(b)

Fig. 5.8 : (a) Time series bubble chart, (b) Multi series bubble chart

5.4 TECHNIQUES FOR VISUAL DATA REPRESENTATIONS

Techniques used for visual data representations are as follows

1. Isoline

- It is a 2 dimensional data representation of curved line that moves constantly on surface of a graph.
- Its plotting is based on arrangement of data rather than data visualisation. Isoline is as shown in Fig. 5.9.

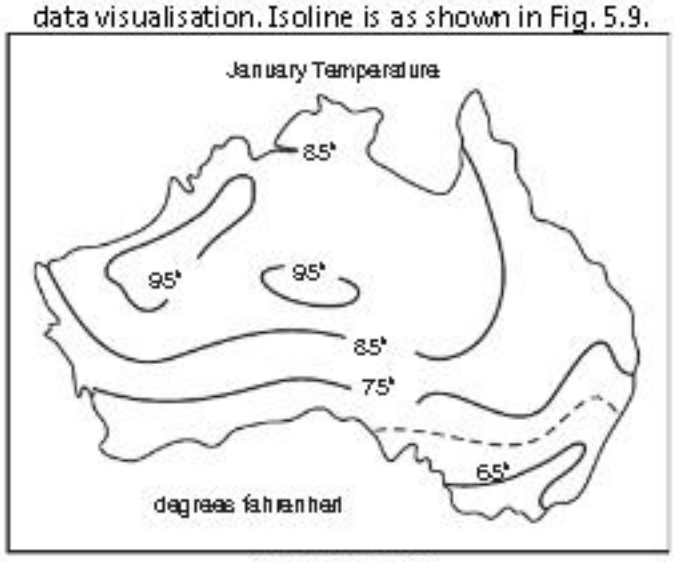


Fig. 5.9 : Isoline

2. Isosurface

- Isosurface as shown in Fig. 5.10 is 3 dimensional representation of an isoline.

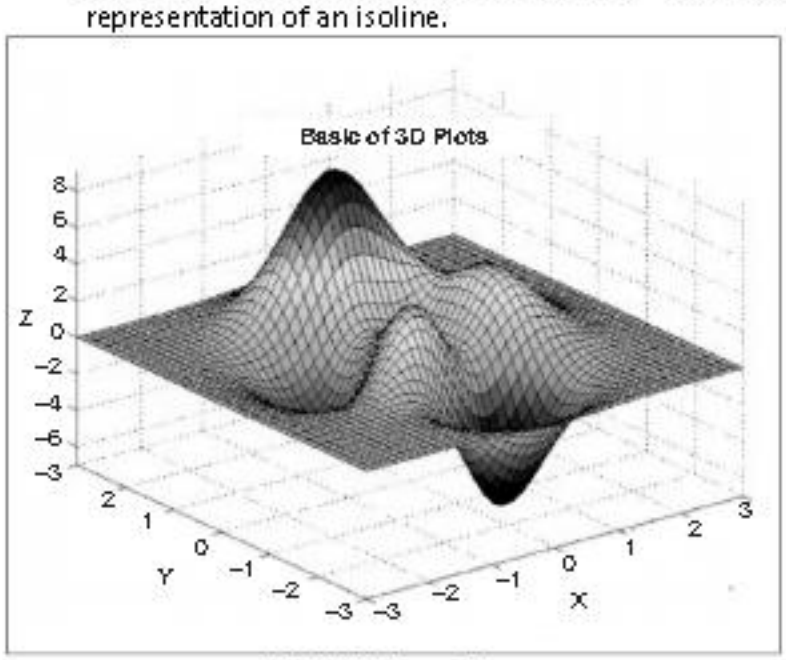


Fig. 5.10 : Isosurface

- It is designed to represent points that are bounded by a constant value in a volume of space i.e. in a domain that covers 3D space.

3. Direct Volume Rendering (DVR)

- DVR as shown in Fig. 5.11 is a method used for performing 2 dimensional projections for a 3 dimensional dataset.
- It means 3D record is projected in 2D for more transparent visualization.

Fig. 5.11 : Direct volume rendering

4. Stream Line

- Stream line as shown in Fig. 5.12 describes data flow.
- They Changes with time when data flow is not steady.
- It is a field line that results from velocity vector field.

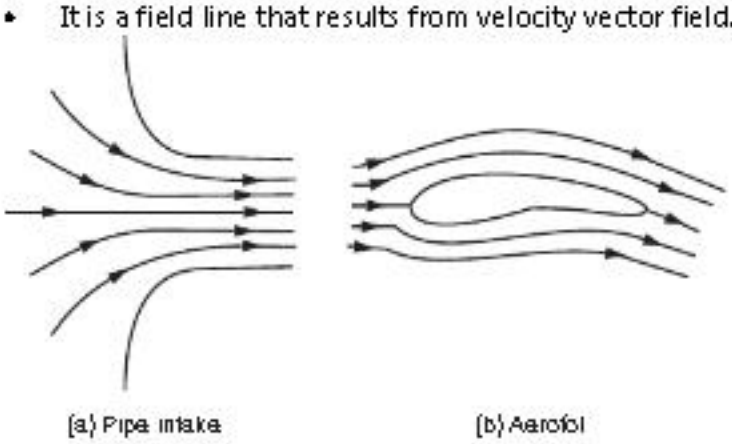


Fig. 5.12: Stream line

5. Map

- Map as shown in Fig. 5.13 is visual representation of geographic locations.
- It is depicted on planar surface.

Fig. 5.13: Google map of Mumbai

6. Parallel Coordinate Plot

- It is used to represent multidimensional data as shown in Fig. 5.14.

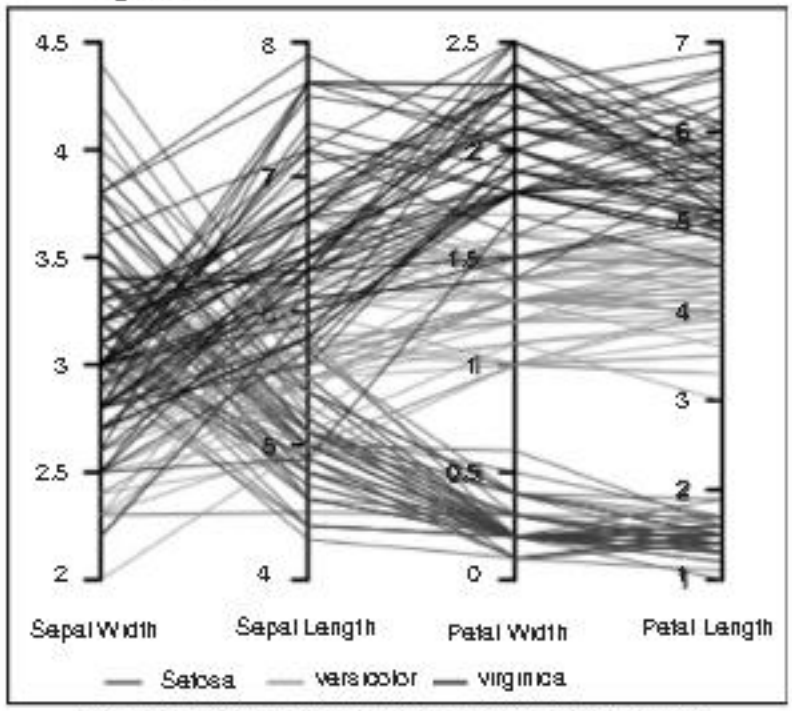


Fig. 5.14 : Parallel coordinate plot for fisher iris data

2. Time Line

- Chronological display of events is shown by using time line as shown in Fig. 5.15.

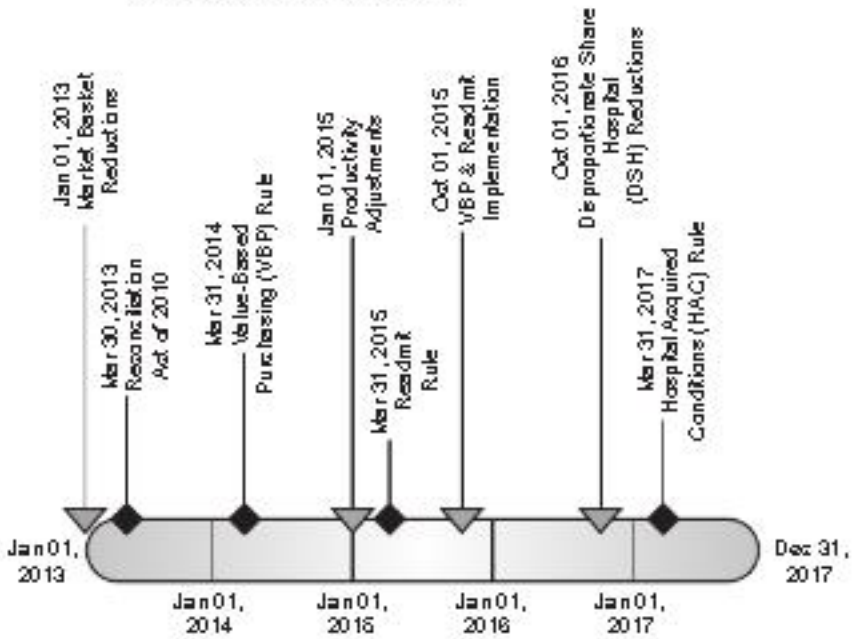


Fig. 5.15: Time line

8. Venn Diagram

- It shows logical relations between finite collections of sets as shown in Fig. 5.16.

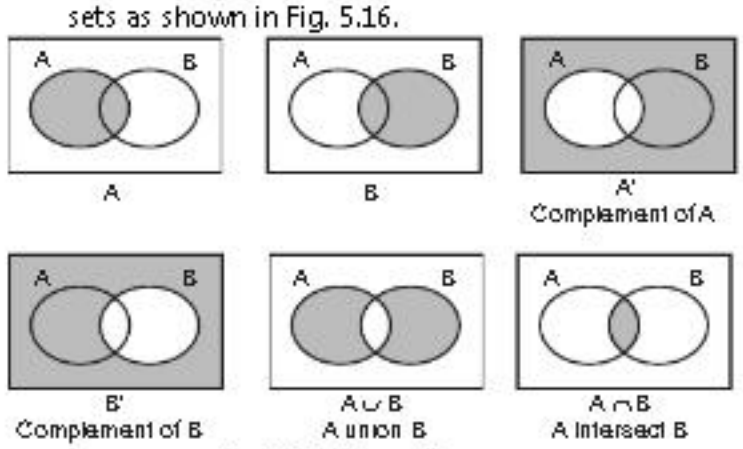


Fig. 5.16: Venn diagrams

9. Euler Diagram

- It represents relationship between sets as shown in Fig. 5.17

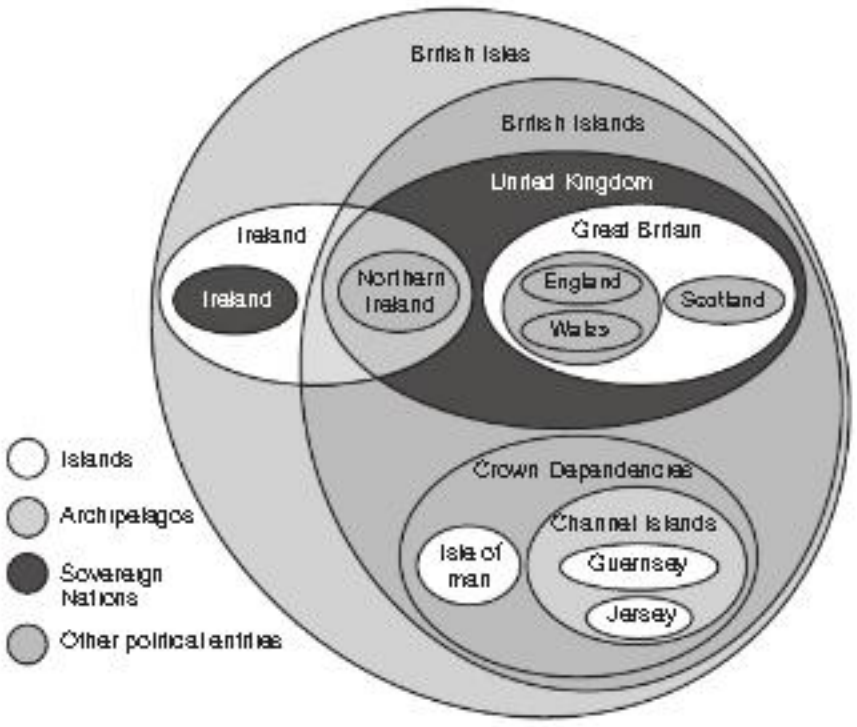


Fig. 5.17: Euler diagram

10. Hyperbolic Trees

- Fig. 5.18 shows hyperbolic trees which represent graph drawn using hyperbolic geometry.

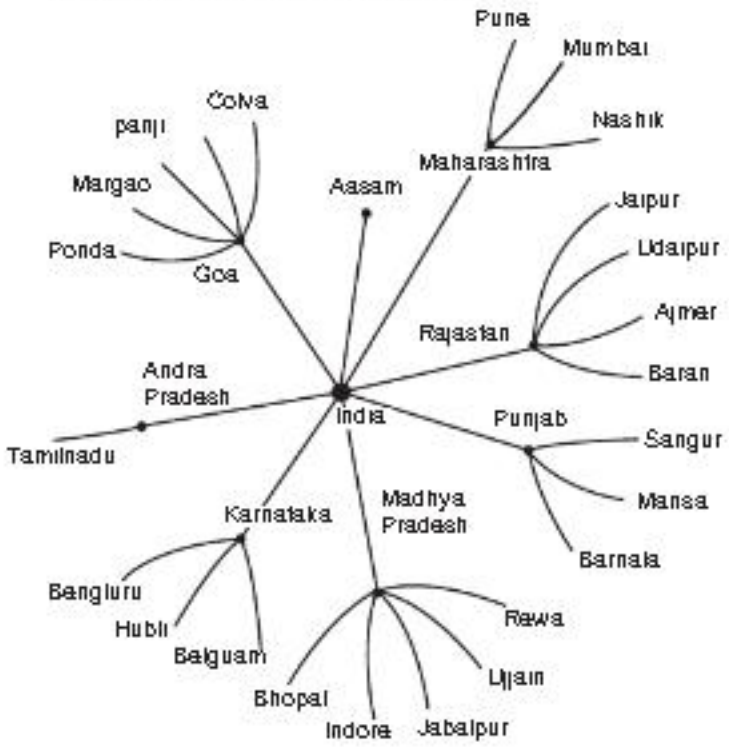


Fig. 5.18: Hyperbolic trees

11. Cluster Diagram

- Cluster is represented by cluster diagram as shown in Fig. 5.19. Ex. Cluster of astronomic entity

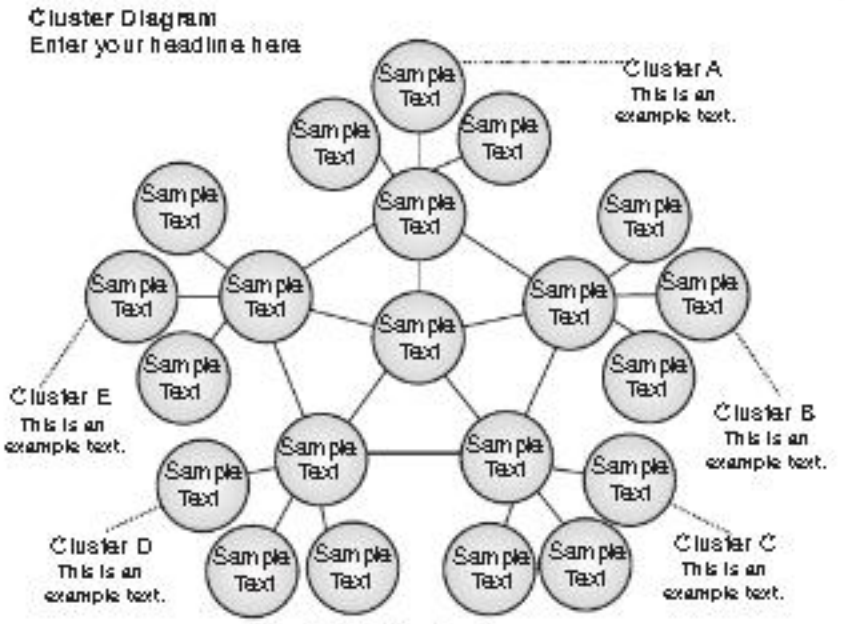
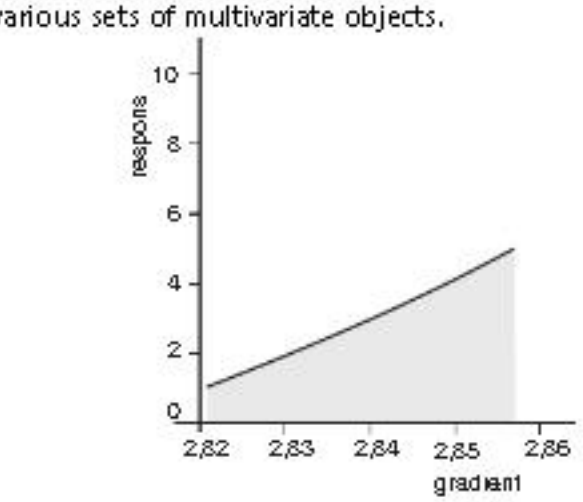


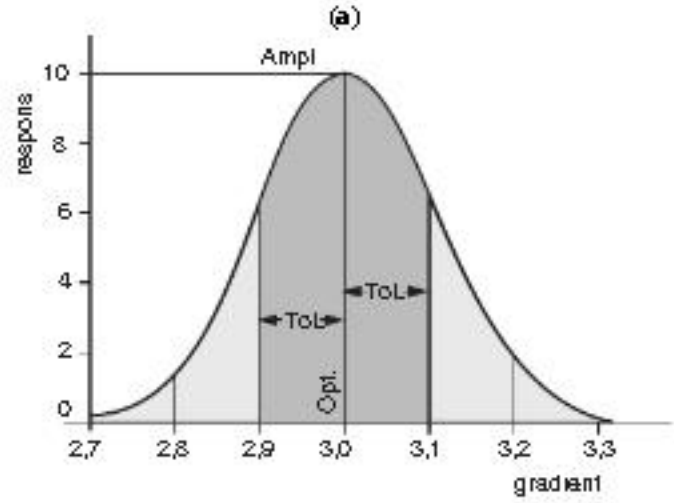
Fig. 5.19: Cluster diagram

12. Ordinogram

Ordinogram is as shown in Fig. 5.20, which is used to analyze various sets of multivariate objects.



(a)



(b)

Fig. 5.20: Ordinogram

5.5 TYPES OF DATA VISUALIZATION

Following table shows the different types of data visualization

Table 5.1

Sr. No.	Name of Type	Details	Used Tools
1.	1D/Linear	Lists of data items, organized by a single feature (e.g., alphabetical order)	No Visualization tools used

Sr. No.	Name of Type	Details	Used Tools
2.	2D/Planar	2D area types of data visualisation are usually geospatial, meaning that they relate to the relative position of things on the earth's surface. Example : Choropleth Maps Dasymetric map Cartogram Isopleth maps self-organizing map (SOM) A dot distribution	Google Charts GeoCommons Google Maps API Polymaps Many eyes Google Fusion Tables Tableau Public
3.	3D/Volumetric	3D types of visualisation are generally 3D computer models, volume/surface rendering and computer simulations.	TrueSpace AutoQ3D AC3D
4.	nD/Multidimensional	Multidimensional data elements are those with two or more dimensions. Example : Histogram, Pie chart, Tag cloud, Bubble cloud, Bar chart, Scatter plot, Heat map, etc.	Google Charts Many eyes Google Fusion Tables Tableau Public
5.	Temporal	Temporal visualisations are similar to one-dimensional linear visualisations, but differ because they have a start and finish time and items that may overlap each other. Example Timeline Time series Connected scatter plot Polar Area Diagram Gantt chart	Excel Time plot TimeFlow Time line JS Time Searcher Google Charts Google Fusion Tables Tableau Public
6.	Tree/Hierarchical	Data relationships need to be shown in the form of hierarchies. Tree/Hierarchical have Tree Diagram Ring Chart Dendrogram Radial Tree Hyperbolic Tree	Google Charts d3 Network Workbench/Sci
7.	Network	It is used to represent data relations that are too complex to be represented in the form of hierarchies. Node-link diagram Dependency graph Alluvial Diagram Matrix chart	Gephi NodeXL VOSviewer UCINET GUESS Network Workbench/Sci, sigma.js d3/Protovis Many Eyes Google Fusion Tables

5.6 VISUALIZING BIG DATA

- Traditional visualization techniques are not sufficient for Big Data Visualization. Big data have structured as well as unstructured data collected from various resources. It is difficult for traditional tools to handle heterogeneity of data sources, data streaming and real time data.
- Challenges faced by business to handle Big Data in terms of
  - Unstructured data format
  - Unable to analyze real time data
  - Huge amount of data generated
  - Lack of efficient tools and techniques
- **Volume :** The methods are developed to work with an immense number of datasets and enable to derive meaning from large volumes of data.
- **Variety :** The methods are developed to combine as many data sources as needed.
- **Velocity :** With the methods, businesses can replace batch processing with real-time stream processing.
- **Value :** The methods not only enable users to create attractive infographics and heatmaps, but also create business value by gaining insights from big data.
- All above factor leads IT companies to focus on research and development of robust algorithm, software and tools to analyze data which is scattered in Internet. An advance visual analytics technique allows business owners and researchers to explore data.
- Visualization helps to identify data patterns in form of graphs or charts which allows deriving helpful information from raw data.
- Visual data mining performs integration of information visualization and human computer interaction.

- Visualization produces cluttered images that are filtered with the help of clutter reduction techniques i.e. Uniform Sampling and dimension reduction.
- Visual data reduction performs automated data analysis to measure density, outlier and their differences.
- These measures are used as quality metrics to evaluate data reduction activity.
- **Categories of Visual Quality Metrics are as**
  - Size metrics (no of data points)
  - Visual effectiveness metrics (data density, collisions)
  - Feature preservation metrics (discovering and preserving data density difference)
- In Big Data applications, it is difficult to conduct data visualization because of the large size and high dimension of big data.
- Most of current Big Data visualization tools have poor performances in scalability, functionalities, and response time.
- Uncertainty can result in a great challenge to effective uncertainty-aware visualization and arise during a visual analytics process.
- Potential solutions to some challenges or problems about visualization and big data were presented :

1. **Meeting the Need for Speed :** One possible solution is hardware. Increased memory and powerful parallel processing can be used. Another method is putting data in-memory but using a grid computing approach, where many machines are used.
2. **Understanding the Data :** One solution is to have the proper domain expertise in place.
3. **Addressing Data Quality :** It is necessary to ensure the data is clean through the process of data governance or information management.
4. **Displaying Meaningful Results** : One way is to cluster data into a higher-level view where smaller groups of data are visible and the data can be effectively visualized.
5. **Dealing with Outliers** : Possible solutions are to remove the outliers from the data or create a separate chart for the outliers.

Advantages of Visualization Tools

- Simple to use even for non technical users.
- Interactive to connect with different data sources.
- Competent enough to create appropriate visual presentation of data.
- Able to interpret big data and can analyze it.
- Able to link different data values, restore missing data and polish data for further analysis.

5.7 TOOLS USED IN DATA VISUALIZATION

Tools used in data visualization are as follows,

1. Tag Galaxy

- Tag Galaxy provides a stunning ways to find flicker images as shown in Fig. 5.21.
- This site provides search tools which makes online combing process a memorable visual experience.

Fig. 5.21 : Tag galaxy

2. D3

- Data driven Documents called as D3 as shown in Fig. 5.22. It is a JavaScript library for manipulating documents based on data.
- It can bind arbitrary data to a Document Object Model (DOM) and applies data driven transformations to the document.
- D3 brings data in HTML, SVG, and CSS. D3's format.

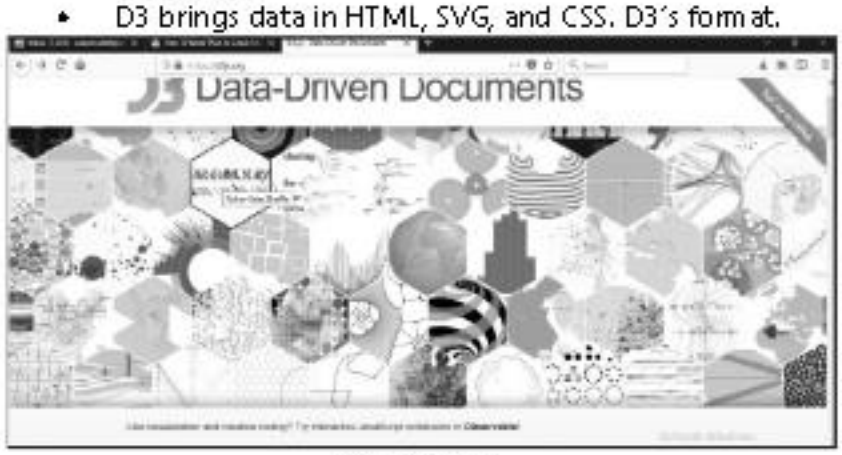


Fig. 5.22: D3

3. Rootzmap Mapping the Internet

- This tool generates a series of map on the basis of provided dataset by National Aeronautics and Space Administration.
- Fig. 5.23 represents Rootzmap Mapping the Internet.

Fig. 5.23: Rootzmap mapping the internet

4. Google Charts API

- This tool allows user to create dynamic web page embedded charts as shown in Fig. 5.24.
- A chart is obtained from the data and formatting parameters supplied by HTTP.

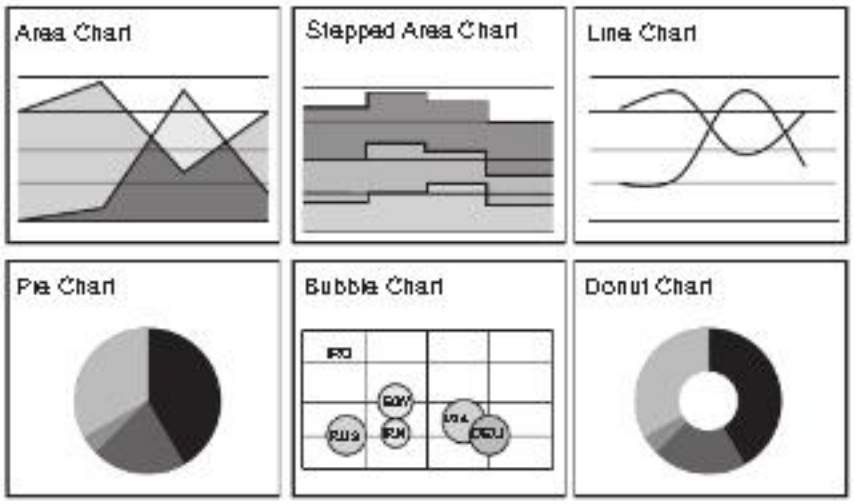


Fig. 5.24 : Google charts API

5. TwittEarth

Fig. 5.25: TwittEarth

- This tool helps to show live tweets from all over world on 3 D globe.
- This improves social media visualization and provides global image mapping in tweets as shown in Fig. 5.25.

6. Some More Tools

- **Last.Forward :** It is open source software used to analyze social music network provided by last.fm.
- **Digg.com :** It provides web based visualization tools.
- **Pics :** It is used to keep track of images on website.

7. Open Source Data Visualization Tools

Some open source tools for data visualization are VTK, ELKI, Gephi, IBM OpenDX, Tulip, Tableau Public etc.

5.8 ANALYTICAL TECHNIQUES USED IN BIG DATA VISUALIZATION

Analytical techniques are used to identify relationship among variables.

Some Commonly Used Analytical Techniques are as Follows :

1. Regression Analysis

- It is a statistical tool used for prediction.
- It used to predict continuous dependent variables from independent variables.
- This helps to find effect of one variable on another.
- Various types of regression analysis are Logistic regression, Ordinary least squares regression, Hierarchical linear modelling and Duration model.

2. Grouping Methods

- It is technique of making categories of observation into significant blocks.
- Discriminant analysis is performed to identify features to distinguish groups.

3. Multiple Equation Models

- To analyze pathway from independent variables to dependent variables carried out in multiple equation model.
- Path analysis and structural equation modelling are types of multiple equation models.

Tableau Public for Data Visualization

Tableau is a business intelligence and data visualization tool that turns data into actionable insights probably business insights.

In Simple words, Tableau helps to see and understand data.

Download and Installation Steps :

Tableau offers five main products catering to diverse visualization needs for professionals and organizations. They are :

1. **Tableau Desktop :** Made for individual use
2. **Tableau Server :** Collaboration for any organization
3. **Tableau Online :** Business Intelligence in the Cloud
4. **Tableau Reader :** Let you read files saved in Tableau Desktop.
5. **Tableau Public :** For journalists or anyone to publish interactive data online and it is free to use.

Features of Tableau

- Rapidly analyze data
- Browse and Explore
- User-friendly Dashboards
- Observe and Calculate

- Share and Interact Various positions for Tableau :
  1. Business Analyst
  2. Data Scientist
  3. Tableau Expert
  4. Tableau Developer
  5. BI Developer
  6. Lead Reporting

How to Access Data source to tableau? File Systems like CSV, Excel etc. Relational systems like Oracle, Sql Server, and DB2 etc. Cloud systems like Windows Azure, Google Big Query etc. Other Sources using ODBC

Tableau Server Components

The Following are the Components of Tableau Server Application Server :

- Application Server processes (wgserver.exe) handle browsing and permissions for the Tableau Server web and mobile interfaces. When a user opens a view in a client device, that user starts a session on Tableau Server. This means that an Application Server thread starts and checks the permissions for that user and that view
- **VizQL Server :** Once a view is opened, the client sends a request to the VizQL process (vizqlserver.exe). The VizQL process then sends queries directly to the data source, returning a result set that is rendered as images and presented to the user. Each VizQL Server has its own cache that can be shared across multiple users.
- **Data Server :** The Tableau Data Server lets you centrally manage and store Tableau data sources. It also maintains metadata from Tableau Desktop, such as calculations, definitions, and groups.

Tableau Environment Opening and Closing the Application

The first thing to understand is how to open and close the application. Open Tableau There are many ways to open Tableau from your desktop computer. Open the application by doing one of the following : Double-click the Tableau icon on your desktop.

Select Start > All Programs > Tableau.

Double-click a Tableau workbook or bookmark file.

Tableau files are typically stored in the My Tableau Repository folder of your My Documents folder.

Close Tableau When you are done working in Tableau you should save your work and close the application. Close the application by doing one of the following :

- Click the Close icon located in the right corner of the application title bar.
- Select File > Exit. If your workbook has not been saved, you will be asked whether you want to save it.

Tableau Workspace

- The Tableau workspace consists of menus, a toolbar, the Data window, cards that contain shelves and legends, and one or more sheets. Sheets can be worksheets or dashboards.
- Worksheets contain shelves, which are where you drag data fields to build views. You can change the default layout of the shelves and cards to suit your needs, including resizing, moving, and hiding them.
- Dashboards contain views, legends, and quick filters. When you first create a dashboard, the Dashboard is empty and all of the worksheets in the workbook are shown in the Dashboard window.

Data Blending

- Data blending is when you blend data from multiple data sources on a single worksheet. The data is joined on common dimensions.
- Data Blending does not create row level joins and is not a way to add new dimensions or rows to your data. Data blending should be used when you have related data in multiple data sources that you want to analyze together in a single view.
- To integrate data, you must first add one of the common dimensions from the primary data source to the view. For example, when blending Actual and Target sales data, the two data sources may have a Date field in common.
- The Date field must be used on the sheet. Then when you switch to the secondary data source in the Data window, Tableau automatically links fields that have the same name.
- If they don't have the same name, you can define a custom relationship that creates the correct mapping between fields.

Joining Tables

- Many relational data sources are made up of a collection of tables that are related by specific fields.
- For example, a data source for a publisher may have a table for authors that contain the first name, last name, phone number, etc. of clients. In addition, there may be another table for titles that contains the price, royalty, and title of published books. In order to analyze these two tables together, to answer questions like, how much was paid in royalties last year for a particular author, you would join the two tables using a common field such as Author ID. That way you can view and use the fields from both tables in your analysis.

EXERCISE

1. Explain Data Visualization and discuss various challenges in data visualization.
2. What are different conventional visualization tools?
3. What are different techniques for visual data representations?
4. Explain types of data visualization.
5. Write a note on Visualizing Big Data
6. Explain tools used in data visualization.
7. Write a note on Analytical techniques used in Big data visualization.

✠ ✠ ✠

UNIT VI

ADVANCED ANALYTICS-TECHNOLOGY AND TOOLS

6.1 ANALYTICS FOR UNSTRUCTURED DATA

- Big data is a combination of transactional data and interactive data. While technologies have mastered the art of managing volumes of transaction data, it is the interactive data that is adding variety and velocity characteristics to the ever-growing data reservoir and subsequently poses significant challenges to enterprises. Irrespective of how data is managed within an enterprise, if it is leveraged properly, it can deliver immense business values.
- Fig. 6.1 illustrates the value cycle of data, from raw data to decision making. In the early 2000s, the acceptance of concepts like Enterprise Data Warehouse (EDW), Business Intelligence (BI) and analytics, helped enterprises to transform raw data collections into actionable wisdom. Analytic applications such as customer analytics, financial analytics, risk analytics, product analytics, health-care analytics became an integral part of the business applications architecture of any enterprise. But all of these applications were dealing with only one type of data : Structured Data.

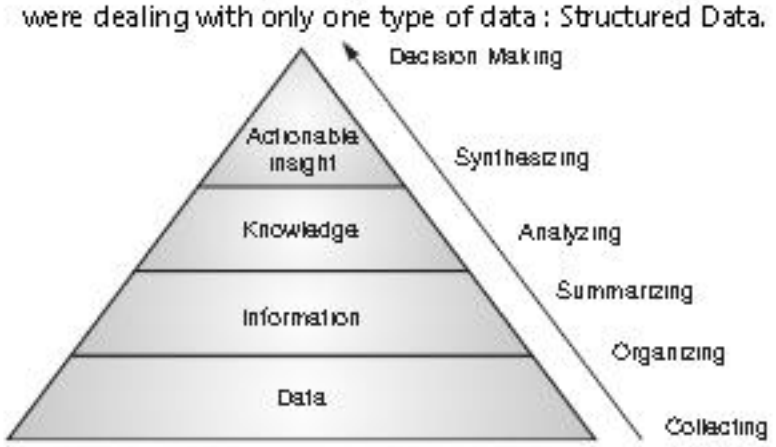


Fig. 6.1 : Transforming raw data into action-guiding wisdom

- The ubiquity of the Internet has dramatically changed the way enterprises function. Essentially most every business became a "digital" business. The result was a data explosion. New application paradigms such as web 2.0, social media applications, cloud computing, and software-as-a-service applications further contributed to the data explosion.
- These new application paradigms added several new dimensions to the very definition of data. Data sources for an enterprise were no longer confined to data stores within the corporate firewalls but also to what is available outside the firewalls. Companies such as LinkedIn, Facebook, Twitter, and Netflix took advantage of these newer data sources to launch innovative product offerings to millions of end users; a new business paradigm of "consumerism" was born. Data regardless of type, location, and source increasingly has become a core business asset for an enterprise and is now categorized as belonging to two camps : Internal Data (enterprise application data) and External data (e.g., web data).
- With that, a new term has emerged : Big Data. So, what is the definition of this all-encompassing arena called "Big Data"? To start with, the definition of big data veers into 3Vs (exploding data volumes, data getting generated at high velocity and data now offering more variety); however, if you scan the Internet for a definition of big data, you will find many more interpretations.
- There are also other interesting observations around big data : It is not only the 3Vs that need to be considered, rather when the scale of data poses real challenges to the traditional data management principles, it can then be considered a big data problem. The heterogeneous nature of big data across multiple platforms and business functions makes it difficult to be managed by following the traditional data management principles, and there is no single platform or solution that has answers to all the questions related to big data.
- On the other hand, there is still a vast trove of data within the enterprise firewalls that is unused (or underused) because it has historically been too voluminous and/or raw (i.e., minimally structured) to be exploited by conventional information systems, or too costly or complex to integrate and exploit. Big data is more a concept than a precise term. Some categorize big data as a volume issue, only to petabyte-scale data collections (> one million GB); some associate big data with the variety of data types even if the volume is in terabytes.
- These interpretations have made big data issues situational. Big data has been of concern in few selected industries and scenarios for some time : Physical Sciences (meteorology, physics), Life Sciences (genomics, biomedical research), Financial institutions (banking, insurance, and capital markets) and government (defense, treasury). For these industries, big data was primarily a data volume problem, and to solve these data-volume-related issues they had heavily relied on a mash-up of custom-developed technologies and a set of complex programs to collect and manage the data.

- But, when doing so, these industries and vendor products generally made the Total Cost of Ownership (TCO) of the IT infrastructure rise exponentially every year. CIOs and CTOs have always grappled with dilemmas like how to lower IT costs to manage the ever-increasing volumes of data, how to build systems that are scalable, how to address performance-related concerns to meet business requirements that are becoming increasingly global in scope and reach, how to manage data security, and privacy and data-quality-related concerns.
- The unstructured nature of big data has made the concerns increase in manifold ways : how does an industry effectively utilize the poly-structured nature of data (structured data like database content, semi-structured data like log files or XML files and unstructured content like text documents or web pages or graphics) in a cost effective manner? We have come a long way from the first mainframe era.
- Over the last few years, technologies have evolved, and now we have solutions that can address some or all of these concerns. Indeed a second mainframe wave is upon us to capture, analyze, classify, and utilize the massive amount of data that can now be collected. There are many instances where organizations, embracing new methodologies and technologies, effectively leverage these poly-structured data reservoirs to innovate. Some of these innovations are described below:
  - Search at scale
  - Multimedia content
  - Sentiment analysis
- Big data analytics is the process of examining large and varied data sets -- i.e., big data -- to uncover hidden patterns, unknown correlations, market trends, customer preferences and other useful information that can help organizations make more-informed business decisions. Driven by specialized analytics systems and software, big data analytics can point the way to various business benefits, including new revenue opportunities, more effective marketing, better customer service, improved operational efficiency and competitive advantages over rivals.
- Big data analytics applications enable data scientists, predictive modelers, statisticians and other analytics professionals to analyze growing volumes of structured transaction data, plus other forms of data that are often left untapped by conventional Business Intelligence (BI) and analytics programs.
- That encompasses a mix of semi-structured and unstructured data -- for example, internet clickstream data, web server logs, social media content, text from customer emails and survey responses, mobile-phone call-detail records and machine data captured by sensors connected to the internet of things.
- On a broad scale, data analytics technologies and techniques provide a means of analyzing data sets and drawing conclusions about them to help organizations make informed business decisions. BI queries answer basic questions about business operations and performance. Big data analytics is a form of advanced analytics, which involves complex applications with elements such as predictive models, statistical algorithms and what-if analyses powered by high-performance analytics systems.

6.2 USE CASES

Following are the Most Popular use Cases of Big Data Analytics :

1. 360° View of the Customer

- Many enterprises use big data to build a dashboard application that provides a 360° view of the customer. These dashboards pull together data from a variety of internal and external sources, analyze it and present it to customer service, sales and/or marketing personnel in a way that helps them do their jobs.
- For example, imagine the sort of dashboard an insurance company might create with information about its customers. Naturally, it would include demographic data, like customer's names, addresses, household income and family members, as well as sales information about which types of policies the customers hold.
- It could also pull information from the company's Customer Relationship Management (CRM) solution about the customer's past interactions with the firm and even provide links to transcripts of recent calls, email messages or chat sessions. It might also show which pages of the company website a particular customer had recently visited, providing valuable clues about the reason a customer might be calling.
- The dashboard could also pull in external information, such as the customer's recent social media posts. Or if an auto insurance customer had agreed to have a tracking device from the company installed, it might even provide details about the customer's current location and recent speed.
- All of that information would obviously help to prepare company staff to interact with the customer, but the most sophisticated dashboards don't stop there. If it uses advanced analytics or machine learning tools, the dashboard takes a guess about the reason for a customer call.
- It could suggest opportunities for cross-selling or upselling customers on products, or if it detects that a customer might be in danger of defecting to a

competitor, it might suggest potential discounts that could lower the customer's rate. Some tools can even analyze customer's language to detect their current emotions and suggest appropriate responses to sales or customer service agents.

- This might sound far-fetched and futuristic, but many companies today already have systems like this one in place, and they are using them to improve customer satisfaction and increase revenues and margins.

2. Fraud Prevention

- For credit card holders, fraud prevention is one of the most familiar use cases for big data. Even before advanced big data analytics became popular, credit card issuers were using rules-based systems to help them flag potentially fraudulent transactions.
- So, for example, if a credit card were used to rent a car in Hawaii, but the customer lived in Omaha, a customer service agent might call to confirm that the cardholder was on vacation and that someone hadn't stolen the card.
- Thanks to big data analytics and machine learning, today's fraud prevention systems are orders of magnitude better at detecting criminal activity and preventing false positives. In the example already mentioned, for instance, a sophisticated fraud prevention system might be able to see that the customer had recently purchased airline tickets, sunscreen and a new swimsuit before the rental car purchase. Based on historical patterns, a predictive analytics or machine learning system would be able to tell that the rental car was thus less likely to be a fraudulent purchase.
- But fraud prevention systems can get even more sophisticated than that. According to Experian, fraud tends to be concentrated in certain geographic regions—often near airports, which make it easy for criminals to move stolen goods. However, which zip codes are riskiest tends to change over time. Big data analytics can look at past records of fraudulent transaction and quickly identify changing trends. Credit card companies and retailers can then pay more attention to transactions in zip codes that are emerging as hotbeds for criminal activity.
- Credit card issuers are understandably hesitant about disclosing all the advanced analytic techniques that they use to detect and prevent fraud. However, many credit card firms and other consultants offer technology, advice and services to other firms to help them set up systems to stop criminal transactions.

3. Security Intelligence

- On the theme of criminal activity, organizations are also using big data analytics to help them thwart hackers and cyberattackers. Operating an enterprise IT department generates an enormous amount of log data. In addition, cyber threat intelligence data is available from external sources, such as law enforcement or security providers. Many organizations or now using big data solutions to help them aggregate and analyze all of this internal and external information to help them prevent, detect and mitigate attacks.
- Big data security solutions vary in sophistication and they are sold under a wide variety of names. For example, vendors sell log analytics tools that can detect anomalies in network data, Security Information and Event Management (SIEM) tools that offer real-time analysis of security alerts generated by other security software, and user and entity behavior analytics (UEBA) solutions that use analytics and machine learning to detect unusual patterns in device or user activity. Other big data security solutions are labelled as security intelligence offerings or network intelligence offerings.

4. Data Warehouse Offload

- One of the easiest — and potentially most cost-effective — ways for organizations to begin using big data tools is to remove some of the burden from their data warehouses. Even among the few organizations that haven't yet started experimenting with big data analytics, it is common to have a data warehouse that facilitates their Business Intelligence (BI) efforts.
- Unfortunately, data warehouse technology tends to be very costly to purchase and run. And as business leaders have begun demanding more reports and insights from their BI teams, the data warehouse solutions haven't always been able to provide the desired performance.
- To solve this problem, many enterprises use an open source big data solution like Hadoop to replace or compliment their data warehouses. Hadoop-based solutions often provide much faster performance while reducing licensing fees and other costs.

5. Price Optimization

- Both Business-to-Consumer (B2C) and Business-to-Business (B2B) enterprises are also using big data analytics to optimize the prices that they charge their customers. For any company, the goal is to set prices so that they maximize their income. If the price is too high, they will sell fewer products, decreasing their net returns. But if the price is too low, they may leave money on the table.

- Big data analytics allows companies to see which price points have yielded the best overall results under various historic market conditions. Businesses that are more sophisticated with their pricing analytics may also employ variable or dynamic pricing strategies. They use their big data solutions to segment their customer base and build models that show how much different types of customers will be willing to pay under different circumstances. B2C companies that have attempted this approach have met with mixed results, but it is fairly standard among B2B companies.

6. Operational Efficiency

- In addition to helping organizations optimize their pricing, big data analytics can also help companies identify other potential opportunities to streamline operations or maximize their profits. Often, this particular big data use case is the purview of BI or financial analysts.
- These staffers have long been running the weekly, monthly and quarterly reports that help executives track the bottom line. But as big data tools have become available and have improved in their sophistication, analysts are able to incorporate data from more sources and to update those reports much more frequently.
- For example, a nationwide retailer might want to track the hourly sales of a new product in all of its physical stores. Big data analytics could easily highlight potential problems, say, for instance, a particular store that hadn't sold any of the new product during the first few hours of the rollout. A quick phone call might then reveal that the store manager had forgotten to put the new product on display, and staff could remedy the situation before it became more costly for the company or led them to inaccurate conclusions about the popularity of the product.

7. Recommendation Engines

- Speaking of popularity, one of the most familiar use cases for big data is the recommendation engine. When you are watching a movie at Netflix or shopping for products from Amazon, you probably now take it for granted that the website will suggest similar items that you might enjoy. Of course, the ability to offer those recommendations arises from the use of big data analytics to analyze historical data.
- These recommendation engines have become so commonplace on the Web that many customers now expect them when they are shopping online. And organizations that haven't taken advantage of their big data in this way may lose customers to competitors or may lose out on upsell or cross-sell opportunities.

8. Social Media Analysis and Response

- The flood of posts that flow through social media outlets like Facebook, Twitter, Instagram and others is one of the most obvious examples of big data. Today, companies are expected to monitor what people are saying about them in social media and respond appropriately — and if they do not, they quickly lose customers.
- As a result, many enterprises are investing in tools to help them monitor and analyze social platforms in real-time. Sometimes these are standalone social media products, while at other times, they are part of a larger marketing intelligence or big data analytics solution.

9. Preventive Maintenance and Support

- Many of the big data use cases mentioned so far relate to retail or financial companies, but businesses in manufacturing, energy, construction, agriculture, transportation and similar sectors of the economy can also benefit from big data. In these examples, some of the biggest benefit might come from using big data to improve equipment maintenance.
- As the Industrial Internet of Things (IIoT) begins to become a reality, factories and other facilities that use expensive equipment are deploying sensors that can monitor that equipment and transmit relevant data over the Internet. They then use big data solutions to analyze that information — often in real time — to detect when a problem is about to occur. They can then perform preventive maintenance that may help prevent accidents or costly line shutdowns.

10. Internet of Things

- And enterprises in every industry are beginning to see the possibilities of the Internet of Things (IoT). As in the preventive maintenance example, they are using sensors to collect data that they can then analyze to achieve actionable insights. They might track customer or product movement, monitor the weather or keep an eye on security camera footage.
- As with big data itself, the number of ways in which analytics can be applied to IoT solutions seems to be endless.

Other Common Big Data Use Cases

- While these are ten of the most common and well-known big data use cases, there are literally hundreds of other types of big data solutions currently in use today. Companies routinely use big data analytics for marketing, advertising, human resource manage and for a host of other needs.

- Many organizations find that they can apply big data solutions to their unique, industry-specific needs. For example, healthcare organizations look for patterns in treatment that lead to the best outcomes for patients. Farmers use big data to find the best time to plant or harvest. Professional sports teams use analytics to decide who should be on the roster and to help improve player performance. The energy industry uses big data from smart meters to improve efficiency, and financial traders use big data to determine when to buy or sell.
- Perhaps it shouldn't be surprising then that once organizations begin to experiment with big data technology, they often find dozens of new uses for it that they hadn't originally considered. As time goes on and the big data tools become even more sophisticated, organizations and vendors will almost certainly discover new ways to use big data solutions that no one has even considered today.

6.3 MAPREDUCE

- MapReduce is a framework using which we can write applications to process huge amounts of data, in parallel, on large clusters of commodity hardware in a reliable manner.

What is MapReduce?

- MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.
- The major advantage of MapReduce is that it is easy to scale data processing over multiple computing nodes. Under the MapReduce model, the data processing primitives are called mappers and reducers. Decomposing a data processing application into mappers and reducers is sometimes nontrivial. But, once we write an application in the MapReduce form, scaling the application to run over hundreds, thousands, or even tens of thousands of machines in a cluster is merely a configuration change. This simple scalability is what has attracted many programmers to use the MapReduce model.

The Algorithm

- Generally MapReduce paradigm is based on sending the computer to where the data resides!
- MapReduce program executes in three stages, namely map stage, shuffle stage, and reduce stage.
  - **Map Stage :** The map or mapper's job is to process the input data. Generally the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data.
  - **Reduce Stage :** This stage is the combination of the Shufflestage and the Reduce stage. The Reducer's job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.
- During a MapReduce job, Hadoop sends the Map and Reduce tasks to the appropriate servers in the cluster.
- The framework manages all the details of data-passing such as issuing tasks, verifying task completion, and copying data around the cluster between the nodes.
- Most of the computing takes place on nodes with data on local disks that reduces the network traffic.
- After completion of the given tasks, the cluster collects and reduces the data to form an appropriate result, and sends it back to the Hadoop server.

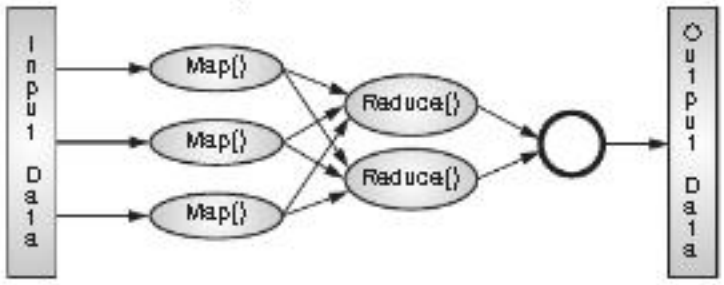


Fig. 6.2

Inputs and Outputs (Java Perspective)

- The MapReduce framework operates on <key, value> pairs, that is, the framework views the input to the job as a set of <key, value> pairs and produces a set of <key, value> pairs as the output of the job, conceivably of different types.
- The key and the value classes should be in serialized manner by the framework and hence, need to implement the Writable interface. Additionally, the key classes have to implement the Writable-Comparable interface to facilitate sorting by the framework. Input and Output types of a MapReduce job : (Input) <k1, v1> -> map -> <k2, v2>-> reduce -> <k3, v3>(Output).

	Input	Output
Map	<k1, v1>	list (<k2, v2>)
Reduce	<k2, list(v2)>	list (<k3, v3>)

Terminology :

- **PayLoad :** Applications implement the Map and the Reduce functions, and form the core of the job.
- **Mapper :** Mapper maps the input key/value pairs to a set of intermediate key/value pair.
- **NamedNode :** Node that manages the Hadoop Distributed File System (HDFS).
- **DataNode :** Node where data is presented in advance before any processing takes place.
- **MasterNode :** Node where JobTracker runs and which accepts job requests from clients.
- **SlaveNode :** Node where Map and Reduce program runs.
- **JobTracker :** Schedules jobs and tracks the assign jobs to Task tracker.
- **Task Tracker :** Tracks the task and reports status to JobTracker.
- **Job :** A program is an execution of a Mapper and Reducer across a dataset.
- **Task :** An execution of a Mapper or a Reducer on a slice of data.
- **Task Attempt :** A particular instance of an attempt to execute a task on a SlaveNode.

Example Scenario

Given below is the data regarding the electrical consumption of an organization. It contains the monthly electrical consumption and the annual average for various years.

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Avg
1979	23	23	2	43	24	25	26	26	26	26	25	26	25
1980	26	27	28	28	28	30	31	31	31	30	30	30	29
1981	31	32	32	32	33	34	35	36	36	34	34	34	34
1984	39	38	39	39	39	41	42	43	40	39	38	38	40
1985	38	39	39	39	39	41	41	41	00	40	39	39	45

If the above data is given as input, we have to write applications to process it and produce results such as finding the year of maximum usage, year of minimum usage, and so on. This is a walkover for the programmers with finite number of records. They will simply write the logic to produce the required output, and pass the data to the application written.

But, think of the data representing the electrical consumption of all the largescale industries of a particular state, since its formation.

When we write applications to process such bulk data,

- They will take a lot of time to execute.
- There will be a heavy network traffic when we move data from source to network server and so on.

To solve these problems, we have the MapReduce framework.

Input Data

The above data is saved as sample.txt and given as input. The input file looks as shown below.

1979	23	23	2	43	24	25	26	26	26	26	25	26	25
1980	26	27	28	28	28	30	31	31	31	30	30	30	29
1981	31	32	32	32	33	34	35	36	36	34	34	34	34
1984	39	38	39	39	39	41	42	43	40	39	38	38	40
1985	38	39	39	39	39	41	41	41	00	40	39	39	45

Example Program

Given below is the program to the sample data using MapReduce framework.

```
package hadoop;
import java.util.*;
import java.io.IOException;
import java.io.IOException;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.*;
import org.apache.hadoop.util.*;
public class ProcessUnits
{
//Mapper class
public static class E_EMapper extends MapReduceBase
implements
Mapper<LongWritable ,/*Input key Type */
Text,   /*Input value Type*/
Text,   /*Output key Type*/
IntWritable>  /*Output value Type*/
{
//Map function
public void map(LongWritable key, Text value,
OutputCollector<Text, IntWritable> output,
Reporter reporter) throws IOException
{
 String line = value.toString();
 String lasttoken = null;
 StringTokenizer s = new StringTokenizer(line,"\t ");
 String year = s.nextToken();
 while(s.hasMoreTokens())
 {
 lasttoken=s.nextToken();
 }
```

```
 int avgprice = Integer.parseInt(lasttoken);
 output.collect(new Text(year), new
IntWritable(avgprice));
}
}
//Reducer class
public static class E_EReduce extends MapReduceBase
implements
Reducer< Text, IntWritable, Text, IntWritable >
{
//Reduce function
public void reduce( Text key, Iterator <IntWritable>
values,
 OutputCollector<Text, IntWritable> output, Reporter
reporter) throws IOException
 {
  int maxavg=30;
  int val=Integer.MIN_VALUE;
  while (values.hasNext())
  {
  if((val=values.next().get())>maxavg)
  {
   output.collect (key, new IntWritable(val));
  }
  }
 }
}
//Main function
public static void main(String args[])throws Exception
{
JobConf conf = new JobConf(ProcessUnits.class);
conf.setJobName("max_eletricityunits");
conf.setOutputKeyClass(Text.class);
conf.setOutputValueClass(IntWritable.class);
conf.setMapperClass(E_EMapper.class);
conf.setCombinerClass(E_EReduce.class);
conf.setReducerClass(E_EReduce.class);
conf.setInputFormat(TextInputFormat.class);
conf.setOutputFormat(TextOutputFormat.class);
FileInputFormat.setInputPaths(conf, new Path(args[0]));
FileOutputFormat.setOutputPath(conf, new Path(args[1]));
JobClient.runJob(conf);
}
}
```

Save the above program as ProcessUnits.java. The compilation and execution of the program is explained below.

Compilation and Execution of Process Units Program

Let us assume we are in the home directory of a Hadoop user (e.g. /home/hadoop).

Follow the steps given below to compile and execute the above program.

Step 1

The following command is to create a directory to store the compiled java classes.

```
$ mkdir units
```

Step 2

Download Hadoop-core-1.2.1.jar, which is used to compile and execute the MapReduce program. Visit the following link http://mvnrepository.com/artifact/org.apache.hadoop/hadoop-core/1.2.1to download the jar. Let us assume the downloaded folder is /home/hadoop/.

Step 3

The following commands are used for compiling the ProcessUnits.javaprogram and creating a jar for the program.

```
$ javac -classpath hadoop-core-1.2.1.jar -d units
ProcessUnits.java
$ jar -cvf units.jar -C units/ .
```

Step 4

The following command is used to create an input directory in HDFS.

```
$HADOOP_HOME/bin/hadoop fs -mkdir input_dir
```

Step 5

The following command is used to copy the input file named sample.txtin the input directory of HDFS.

```
$HADOOP_HOME/bin/hadoop fs -put
/home/hadoop/sample.txt input_dir
```

Step 6

The following command is used to verify the files in the input directory.

```
$HADOOP_HOME/bin/hadoop fs -ls input_dir/
```

Step 7

The following command is used to run the Eleunit_max application by taking the input files from the input directory.

```
$HADOOP_HOME/bin/hadoop jar units.jar
hadoop.ProcessUnits input_dir output_dir
```

Wait for a while until the file is executed. After execution, as shown below, the output will contain the number of input splits, the number of Map tasks, the number of reducer tasks, etc.

```
INFO mapreduce.Job : Job job_1414748220717_0002
completed successfully
14/10/31 06 :02 :52
INFO mapreduce.Job : Counters : 49
File System Counters
FILE : Number of bytes read=61
FILE : Number of bytes written=279400
FILE : Number of read operations=0
FILE : Number of large read operations=0
FILE : Number of write operations=0
HDFS : Number of bytes read=546
HDFS : Number of bytes written=40
HDFS : Number of read operations=9
HDFS : Number of large read operations=0
HDFS : Number of write operations=2 Job Counters
 Launched map tasks=2
 Launched reduce tasks=1
 Data-local map tasks=2
 Total time spent by all maps in occupied slots (ms)=146137
 Total time spent by all reduces in occupied slots (ms)=441
 Total time spent by all map tasks (ms)=14613
 Total time spent by all reduce tasks (ms)=44120
 Total vcore-seconds taken by all map tasks=146137
 Total vcore-seconds taken by all reduce tasks=44120
 Total megabyte-seconds taken by all map tasks=149644288
 Total megabyte-seconds taken by all reduce
tasks=45178880
Map-Reduce Framework
Map input records=5
 Map output records=5
 Map output bytes=45
 Map output materialized bytes=67
 Input split bytes=208
 Combine input records=5
 Combine output records=5
 Reduce input groups=5
 Reduce shuffle bytes=6
 Reduce input records=5
 Reduce output records=5
 Spilled Records=10
 Shuffled Maps =2
 Failed Shuffles=0
 Merged Map outputs=2
 GC time elapsed (ms)=948
 CPU time spent (ms)=5160
 Physical memory (bytes) snapshot=47749120
 Virtual memory (bytes) snapshot=2899349504
 Total committed heap usage (bytes)=277684224
File Output Format Counters
 Bytes Written=40
```

Step 8

The following command is used to verify the resultant files in the output folder.

```
$HADOOP_HOME/bin/hadoop fs -ls output_dir/
```

Step 9

The following command is used to see the output in Part-00000 file. This file is generated by HDFS.

```
$HADOOP_HOME/bin/hadoop fs -cat output_dir/part-
00000
```

Below is the output generated by the MapReduce program.

```
1981 34
1984 40
1985 45
```

Step 10

The following command is used to copy the output folder from HDFS to the local file system for analyzing.

```
$HADOOP_HOME/bin/hadoop fs -cat output_dir/part-
00000/bin/hadoop dfs get output_dir /home/hadoop
```

Important Commands

All Hadoop commands are invoked by the $HADOOP_HOME/bin/hadoopcommand. Running the Hadoop script without any arguments prints the description for all commands.

Usage : hadoop [--config confdir] COMMAND

The following table lists the options available and their description.

Options	Description
namenode -format	Formats the DFS filesystem.
secondarynamenode	Runs the DFS secondary namenode.
namenode	Runs the DFS namenode.
datanode	Runs a DFS datanode.
dfsadmin	Runs a DFS admin client.
mradmin	Runs a Map-Reduce admin client.
fsck	Runs a DFS filesystem checking utility.
fs	Runs a generic filesystem user client.
balancer	Runs a cluster balancing utility.
oiv	Applies the offline fsimage viewer to an fsimage.
fetchdt	Fetches a delegation token from the NameNode.
jobtracker	Runs the MapReduce job Tracker node.
pipes	Runs a Pipes job.
tasktracker	Runs a MapReduce task Tracker node.
historyserver	Runs job history servers as a standalone daemon.

job	Manipulates the MapReduce jobs.
queue	Gets information regarding JobQueues.
version	Prints the version.
jar <jar>	Runs a jar file.
distcp <srcurl> <desturl>	Copies file or directories recursively.
distcp2 <srcurl> <desturl>	DistCp version 2.
archive -archiveName NAME -p	Creates a hadoop archive.
<parent path> <src>* <dest>	
classpath	Prints the class path needed to get the Hadoop jar and the required libraries.
daemonlog	Get/Set the log level for each daemon

How to Interact with MapReduce Jobs

Usage : hadoop job [GENERIC_OPTIONS]

The following are the Generic Options available in a Hadoop job.

GENERIC_OPTIONS	Description
-submit <job-file>	Submits the job.
-status <job-id>	Prints the map and reduce completion percentage and all job counters.
-counter <job-id> <group-name> <countername>	Prints the counter value.
-kill <job-id>	Kills the job.
-events <job-id> <fromevent-#> <#-of-events>	Prints the events' details received by jobtracker for the given range.
-history [all] <jobOutputDir> - history < jobOutputDir>	Prints job details, failed and killed tip details. More details about the job such as successful tasks and task attempts made for each task can be viewed by specifying the [all] option.
-list[all]	Displays all jobs. -list displays only jobs which are yet to complete.
-kill-task <task-id>	Kills the task. Killed tasks are NOT counted against failed attempts.
-fail-task <task-id>	Fails the task. Failed tasks are counted against failed attempts.
-set-priority <job-id> <priority>	Changes the priority of the job. Allowed priority values are VERY_HIGH, HIGH, NORMAL, LOW, VERY_LOW

To see the status of job

```
$HADOOP_HOME/bin/hadoop job -status <JOB-ID>
e.g.
$HADOOP_HOME/bin/hadoop job -status
job_201310191043_0004
```

To See the History of Job Output-Dir

```
$HADOOP_HOME/bin/hadoop job -history <DIR-NAME>
e.g.
$HADOOP_HOME/bin/hadoop job -history
/user/expert/output
```

To kill the job

```
$HADOOP_HOME/bin/hadoop job -kill <JOB-ID>
e.g.
$HADOOP_HOME/bin/hadoop job -kill
job_201310191043_0004
```

Working of Map Reduce

Hadoop Ecosystem component 'MapReduce' works by breaking the processing into two phases :

i. Map phase
ii. Reduce phase

- Each phase has key-value pairs as input and output. In addition, programmer also specifies two functions : map function and reduce function
- Map function takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Read Mapper in detail.
- Reduce function takes the output from the Map as an input and combines those data tuples based on the key and accordingly modifies the value of the key. Read Reducer in detail.

Features of MapReduce

- **Simplicity :** MapReduce jobs are easy to run. Applications can be written in any language such as java, C++, and python.
- **Scalability :** MapReduce can process petabytes of data.
- **Speed :** By means of parallel processing problems that take days to solve, it is solved in hours and minutes by MapReduce.
- **Fault Tolerance :** MapReduce takes care of failures. If one copy of data is unavailable, another machine has a copy of the same key pair which can be used for solving the same subtask.

6.4 APACHE HADOOP

- Hadoop is an open-source framework that allows to store and process big data in a distributed environment across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage.
- Due to the advent of new technologies, devices, and communication means like social networking sites, the amount of data produced by mankind is growing rapidly

every year. The amount of data produced by us from the beginning of time till 2003 was 5 billion gigabytes. If you pile up the data in the form of disks it may fill an entire football field. The same amount was created in every two days in 2011, and in every ten minutes in 2013. This rate is still growing enormously. Though all this information produced is meaningful and can be useful when processed, it is being neglected. 90% of the world's data was generated in the last few years.

6.4.1 Big Data

Big data means really a big data, it is a collection of large datasets that cannot be processed using traditional computing techniques. Big data is not merely a data, rather it has become a complete subject, which involves various tools, techniques and frameworks.

What Comes Under Big Data?

Big data involves the data produced by different devices and applications. Given below are some of the fields that come under the umbrella of Big Data.

- **Black Box Data :** It is a component of helicopter, airplanes, and jets, etc. It captures voices of the flight crew, recordings of microphones and earphones, and the performance information of the aircraft.
- **Social Media Data :** Social media such as Facebook and Twitter hold information and the views posted by millions of people across the globe.
- **Stock Exchange Data :** The stock exchange data holds information about the 'buy' and 'sell' decisions made on a share of different companies made by the customers.
- **Power Grid Data :** The power grid data holds information consumed by a particular node with respect to a base station.
- **Transport Data :** Transport data includes model, capacity, distance and availability of a vehicle.
- **Search Engine Data :** Search engines retrieve lots of data from different databases.

Fig. 6.3

Thus Big Data includes huge volume, high velocity, and extensible variety of data. The data in it will be of three types.

1. **Structured Data :** Relational data.
2. **Semi Structured Data :** XML data.
3. **Unstructured Data :** Word, PDF, Text, Media Logs.

Benefits of Big Data

Big data is really critical to our life and its emerging as one of the most important technologies in modern world. Follow are just few benefits which are very much known to all of us :

- Using the information kept in the social network like Facebook, the marketing agencies are learning about the response for their campaigns, promotions, and other advertising mediums.
- Using the information in the social media like preferences and product perception of their consumers, product companies and retail organizations are planning their production.
- Using the data regarding the previous medical history of patients, hospitals are providing better and quick service.

6.4.2 Big Data Technologies

- Big data technologies are important in providing more accurate analysis, which may lead to more concrete decision-making resulting in greater operational efficiencies, cost reductions, and reduced risks for the business.
- To harness the power of big data, you would require an infrastructure that can manage and process huge volumes of structured and unstructured data in realtime and can protect data privacy and security.
- There are various technologies in the market from different vendors including Amazon, IBM, Microsoft, etc., to handle big data. While looking into the technologies that handle big data, we examine the following two classes of technology :

1. **Operational Big Data**
   - This include systems like MongoDB that provide operational capabilities for real-time, interactive workloads where data is primarily captured and stored.
   - NoSQL Big Data systems are designed to take advantage of new cloud computing architectures that have emerged over the past decade to allow massive computations to be run inexpensively and efficiently. This makes operational big data workloads much easier to manage, cheaper, and faster to implement.
   - Some NoSQL systems can provide insights into patterns and trends based on real-time data with minimal coding and without the need for data scientists and additional infrastructure.

2. **Analytical Big Data**
   - This includes systems like Massively Parallel Processing (MPP) database systems and MapReduce that provide analytical capabilities for retrospective and complex analysis that may touch most or all of the data.
   - MapReduce provides a new method of analyzing data that is complementary to the capabilities provided by SQL, and a system based on MapReduce that can be scaled up from single servers to thousands of high and low end machines.

   These two classes of technology are complementary and frequently deployed together.

Operational vs. Analytical Systems

	Operational	Analytical
Latency	1 ms - 100 ms	1 min - 100 min
Concurrency	1000 - 100,000	1 - 10
Access Pattern	Writes and Reads	Reads
Queries	Selective	Unselective
Data Scope	Operational	Retrospective
End User	Customer	Data Scientist
Technology	NoSQL	MapReduce, MPP Database

Big Data Challenges

The Major Challenges Associated with Big Data are as Follows :

- Capturing data
- Curation
- Storage
- Searching
- Sharing
- Transfer
- Analysis
- Presentation

To fulfill the above challenges, organizations normally take the help of enterprise servers.

Traditional Approach

In this approach, an enterprise will have a computer to store and process big data. Here data will be stored in an RDBMS like Oracle Database, MS SQL Server or DB2 and sophisticated softwares can be written to interact with the database, process the required data and present it to the users for analysis purpose.

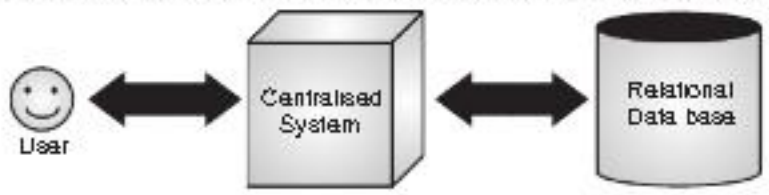


Fig. 6.4

Limitations

- This approach works well where we have less volume of data that can be accommodated by standard database servers, or up to the limit of the processor which is processing the data. But when it comes to dealing with huge amounts of data, it is really a tedious task to process such data through a traditional database server.

Google's Solution

Google solved this problem using an algorithm called MapReduce. This algorithm divides the task into small parts and assigns those parts to many computers connected over the network, and collects the results to form the final result dataset.

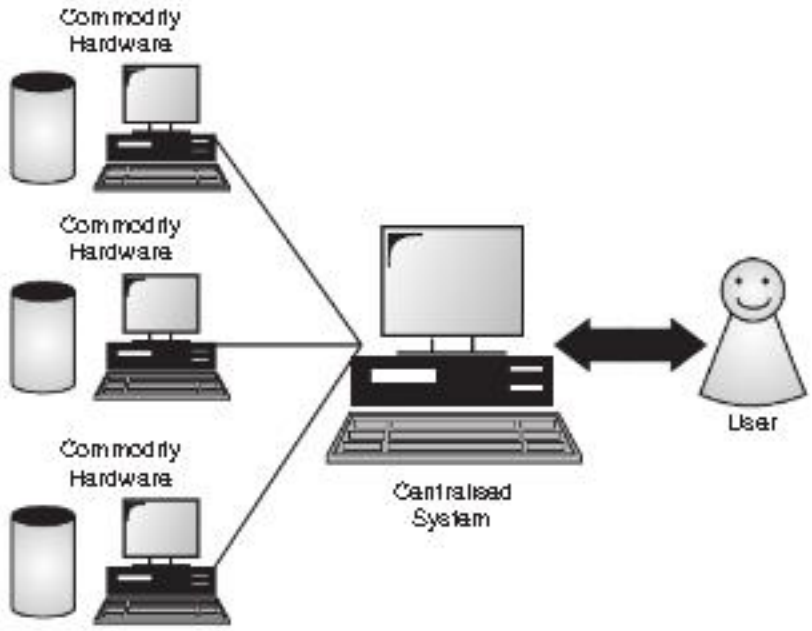


Fig. 6.5

Above diagram shows various commodity hardwares which could be single CPU machines or servers with higher capacity.

Hadoop

- Doug Cutting, Mike Cafarella and team took the solution provided by Google and started an Open Source Project called Hadoop in 2005 and Doug named it after his son's toy elephant. Now Apache Hadoop is a registered trademark of the Apache Software Foundation.
- Hadoop runs applications using the MapReduce algorithm, where the data is processed in parallel on different CPU nodes. In short, Hadoop framework is capable enough to develop applications capable of running on clusters of computers and they could perform complete statistical analysis for a huge amounts of data.

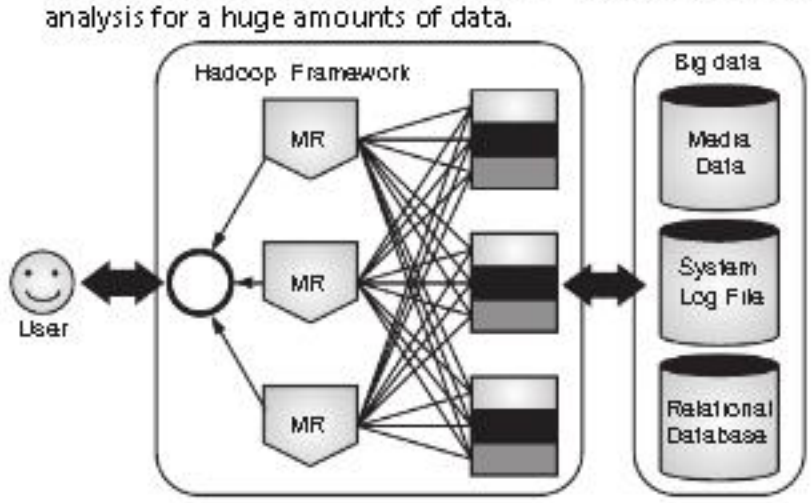


Fig. 6.6

- Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. A Hadoop frame-worked application works in an environment that provides distributed storage and computation across clusters of computers.
- Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage.

6.4.3 Hadoop Architecture

Hadoop Framework Includes Following Four Modules :

1. **Hadoop Common :** These are Java libraries and utilities required by other Hadoop modules. These libraries provides filesystem and OS level abstractions and contains the necessary Java files and scripts required to start Hadoop.
2. **Hadoop YARN :** This is a framework for job scheduling and cluster resource management.
3. **Hadoop Distributed File System (HDFS™) :** A distributed file system that provides high-throughput access to application data.
4. **Hadoop MapReduce :** This is YARN-based system for parallel processing of large data sets.

We can use following diagram to depict these four components available in Hadoop framework.

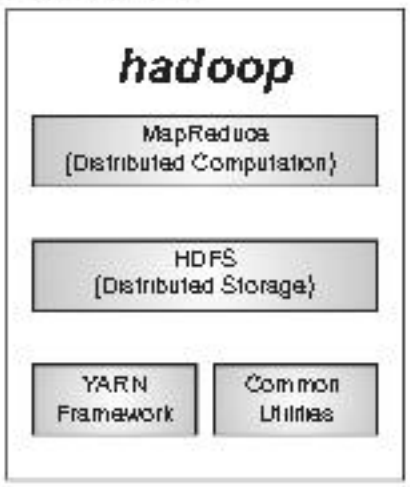


Fig. 6.7

- Since 2012, the term "Hadoop" often refers not just to the base modules mentioned above but also to the collection of additional software packages that can be installed on top of or alongside Hadoop, such as Apache Pig, Apache Hive, Apache HBase, Apache Spark etc.

MapReduce

- Hadoop MapReduce is a software framework for easily writing applications which process big amounts of data in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.

The Term MapReduce Actually Refers to the Following Two Different Tasks that Hadoop Programs Perform :

- **The Map Task :** This is the first task, which takes input data and converts it into a set of data, where individual elements are broken down into tuples (key/value pairs).
- **The Reduce Task :** This task takes the output from a map task as input and combines those data tuples into a smaller set of tuples. The reduce task is always performed after the map task.
- Typically both the input and the output are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks.
- The MapReduce framework consists of a single master JobTracker and one slave TaskTracker per cluster-node. The master is responsible for resource management, tracking resource consumption/availability and scheduling the jobs component tasks on the slaves, monitoring them and re-executing the failed tasks.
- The slaves TaskTracker execute the tasks as directed by the master and provide task-status information to the master periodically.
- The JobTracker is a single point of failure for the Hadoop MapReduce service which means if JobTracker goes down, all running jobs are halted.

6.4.4 Hadoop Distributed File System

- Hadoop can work directly with any mountable distributed file system such as Local FS, HFTP FS, S3 FS, and others, but the most common file system used by Hadoop is the Hadoop Distributed File System (HDFS).
- The Hadoop Distributed File System (HDFS) is based on the Google File System (GFS) and provides a distributed file system that is designed to run on large clusters (thousands of computers) of small computer machines in a reliable, fault-tolerant manner.
- HDFS uses a master/slave architecture where master consists of a single NameNode that manages the file system metadata and one or more slave DataNodes that store the actual data.
- A file in an HDFS namespace is split into several blocks and those blocks are stored in a set of DataNodes. The NameNode determines the mapping of blocks to the DataNodes. The DataNodes takes care of read and write operation with the file system. They also take care of block creation, deletion and replication based on instruction given by NameNode.
- HDFS provides a shell like any other file system and a list of commands are available to interact with the file system. These shell commands will be covered in a separate chapter along with appropriate examples.

How Does Hadoop Work?

Stage 1

A user/application can submit a job to the Hadoop (a hadoop job client) for required process by specifying the following items :

- The location of the input and output files in the distributed file system.

- The java classes in the form of jar file containing the implementation of map and reduce functions.
- The job configuration by setting different parameters specific to the job.

Stage 2

The Hadoop job client then submits the job (jar/executable etc) and configuration to the JobTracker which then assumes the responsibility of distributing the software/configuration to the slaves, scheduling tasks and monitoring them, providing status and diagnostic information to the job-client.

Stage 3

The TaskTrackers on different nodes execute the task as per MapReduce implementation and output of the reduce function is stored into the output files on the file system.

6.4.5 Advantages of Hadoop

- Hadoop framework allows the user to quickly write and test distributed systems. It is efficient, and it automatic distributes the data and work across the machines and in turn, utilizes the underlying parallelism of the CPU cores.
- Hadoop does not rely on hardware to provide Fault-Tolerance and High Availability (FTHA), rather Hadoop library itself has been designed to detect and handle failures at the application layer.
- Servers can be added or removed from the cluster dynamically and Hadoop continues to operate without interruption.
- Another big advantage of Hadoop is that apart from being open source, it is compatible on all the platforms since it is Java based.

Hadoop is supported by GNU/Linux platform and its flavors. Therefore, we have to install a Linux operating system for setting up Hadoop environment. In case you have an OS other than Linux, you can install a Virtualbox software in it and have Linux inside the Virtualbox.

Pre-Installation Setup

Before installing Hadoop into the Linux environment, we need to set up Linux using ssh (Secure Shell). Follow the steps given below for setting up the Linux environment.

Creating a User

At the beginning, it is recommended to create a separate user for Hadoop to isolate Hadoop file system from Unix file system. Follow the steps given below to create a user :

- Open the root using the command "su".
- Create a user from the root account using the command "useradd username".
- Now you can open an existing user account using the command "su username".

Open the Linux terminal and type the following commands to create a user.

```
$ su
password :
# useradd hadoop
# passwd hadoop
New passwd :
Retype new passwd
```

SSH Setup and Key Generation

- SSH setup is required to do different operations on a cluster such as starting, stopping, distributed daemon shell operations. To authenticate different users of Hadoop, it is required to provide public/private key pair for a Hadoop user and share it with different users.
- The following commands are used for generating a key value pair using SSH. Copy the public keys form id_rsa.pub to authorized_keys, and provide the owner with read and write permissions to authorized_keys file respectively.

```
$ ssh-keygen -t rsa
$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
$ chmod 0600 ~/.ssh/authorized_keys
```

Installing Java

- Java is the main prerequisite for Hadoop. First of all, you should verify the existence of java in your system using the command "java -version". The syntax of java version command is given below.

```
$ java -version
```

- If everything is in order, it will give you the following output.

```
java version "1.7.0_71"
Java(TM) SE Runtime Environment (build 1.7.0_71-b13)
Java Hot Spot (TM) Client VM (build 25.0-b02, mixed mode)
```

If java is not installed in your system, then follow the steps given below for installing java.

Step 1

Download java (JDK <latest version> - X64.tar.gz) by visiting the following link

http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads1880260.html.

Then jdk-7u71-linux-x64.tar.gz will be downloaded into your system.

Step 2

Generally you will find the downloaded java file in Downloads folder. Verify it and extract the jdk-7u71-linux-x64.gz file using the following commands.

```
$ cd Downloads/
$ ls
jdk-7u71-linux-x64.gz
$ tar zxf jdk-7u71-linux-x64.gz
$ ls
jdk1.7.0_71 jdk-7u71-linux-x64.gz
```

Step 3

To make java available to all the users, you have to move it to the location "/usr/local/". Open root, and type the following commands.

```
$ su
password :
# mv jdk1.7.0_71 /usr/local/
# exit
```

Step 4

For setting up PATH and JAVA_HOME variables, add the following commands to ~/.bashrc file.

```
export JAVA_HOME=/usr/local/jdk1.7.0_71
export PATH=$PATH :$JAVA_HOME/bin
```

Now apply all the changes into the current running system.

```
$ source ~/.bashrc
```

Step 5

Use the following commands to configure java alternatives :

```
# alternatives --install /usr/bin/java java
usr/local/java/bin/java 2
# alternatives --install /usr/bin/javac javac
usr/local/java/bin/javac 2
# alternatives --install /usr/bin/jar jar usr/local/java/bin/jar
2
# alternatives --set java usr/local/java/bin/java
# alternatives --set javac usr/local/java/bin/javac
# alternatives --set jar usr/local/java/bin/jar
```

Now verify the java -version command from the terminal as explained above.

Downloading Hadoop

Download and extract Hadoop 2.4.1 from Apache software foundation using the following commands.

```
$ su
password :
# cd /usr/local
# wget http ://apache.claz.org/hadoop/common/hadoop-2.4.1/
hadoop-2.4.1.tar.gz
# tar xzf hadoop-2.4.1.tar.gz
# mv hadoop-2.4.1/* to hadoop/
# exit
```

Hadoop Operation Modes

Once you have downloaded Hadoop, you can operate your Hadoop cluster in one of the three supported modes :

1. **Local/Standalone Mode :** After downloading Hadoop in your system, by default, it is configured in a standalone mode and can be run as a single java process.
2. **Pseudo Distributed Mode :** It is a distributed simulation on single machine. Each Hadoop daemon such as hdfs, yam, MapReduce etc., will run as a separate java process. This mode is useful for development.
3. **Fully Distributed Mode :** This mode is fully distributed with minimum two or more machines as a cluster. We will come across this mode in detail in the coming chapters.

Installing Hadoop in Standalone Mode

- Here we will discuss the installation of Hadoop 2.4.1 in standalone mode.
- There are no daemons running and everything runs in a single JVM. Standalone mode is suitable for running MapReduce programs during development, since it is easy to test and debug them.

Setting Up Hadoop

You can set Hadoop environment variables by appending the following commands to ~/.bashrc file.

```
export HADOOP_HOME=/usr/local/hadoop
```

Before proceeding further, you need to make sure that Hadoop is working fine. Just issue the following command :

```
$ hadoop version
```

If everything is fine with your setup, then you should see the following result :

```
Hadoop 2.4.1
Subversion https
://svn.apache.org/repos/asf/hadoop/common -r 1529768
Compiled by hortonmu on 2013-10-07T06 :28Z
Compiled with protoc 2.5.0
From source with checksum
79e53ce7994d1628b240f09af91e1af4
```

It means your Hadoop's standalone mode setup is working fine. By default, Hadoop is configured to run in a non-distributed mode on a single machine.

Example

- Let's check a simple example of Hadoop. Hadoop installation delivers the following example MapReduce jar file, which provides basic functionality of MapReduce and can be used for calculating, like Pi value, word counts in a given list of files, etc.

```
$HADOOP_HOME/share/hadoop/mapreduce/hadoop-
mapreduce-examples-2.2.0.jar
```

- Let's have an input directory where we will push a few files and our requirement is to count the total number of words in those files. To calculate the total number of words, we do not need to write our MapReduce, provided the .jar file contains the implementation for word count. You can try other examples using the same .jar file; just issue the following commands to check supported MapReduce functional programs by hadoop-mapreduce-examples-2.2.0.jar file.

```
$ hadoop jar
$HADOOP_HOME/share/hadoop/mapreduce/hadoop-
mapreduceexamples-2.2.0.jar
```

Step 1

Create temporary content files in the input directory. You can create this input directory anywhere you would like to work.

```
$ mkdir input
$ cp $HADOOP_HOME/*.txt input
$ ls -l input
```

It will give the following files in your input directory :

```
total 24
-rw-r--r-- 1 root root 15164 Feb 21 10 :14 LICENSE.txt
-rw-r--r-- 1 root root 101 Feb 21 10 :14 NOTICE.txt
-rw-r--r-- 1 root root 1366 Feb 21 10 :14 README.txt
```

These files have been copied from the Hadoop installation home directory. For your experiment, you can have different and large sets of files.

Step 2

Let's start the Hadoop process to count the total number of words in all the files available in the input directory, as follows :

```
$ hadoop jar
$HADOOP_HOME/share/hadoop/mapreduce/hadoop-
mapreduceexamples-2.2.0.jar wordcount input output
```

Step 3

Step-2 will do the required processing and save the output in output/part-r00000 file, which you can check by using :

```
$cat output/*
```

It will list down all the words along with their total counts available in all the files available in the input directory.

```
"AS 4
"Contribution" 1
"Contributor" 1
"Derivative 1
"Legal 1
"License" 1
"License"); 1
"Licensor" 1
"NOTICE" 1
"Not 1
"Object" 1
"Source" 1
"Work" 1
"You" 1
"Your") 1
"" 1
"control" 1
"printed 1
"submitted" 1
(50%) 1
(BIS), 1
(C) 1
(Don't) 1
(ECCN) 1
(INCLUDING 2
(INCLUDING, 2
.....
```

Installing Hadoop in Pseudo Distributed Mode

Follow the steps given below to install Hadoop 2.4.1 in pseudo distributed mode.

Step 1 : Setting Up Hadoop

You can set Hadoop environment variables by appending the following commands to ~/.bashrc file.

```
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export
HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/
lib/native
export PATH=$PATH :$HADOOP_HOME/sbin
:$HADOOP_HOME/bin
export HADOOP_INSTALL=$HADOOP_HOME
```

Now apply all the changes into the current running system.

```
$ source ~/.bashrc
```

Step 2 : Hadoop Configuration

You can find all the Hadoop configuration files in the location op infrastructure.

```
$ cd $HADOOP_HOME/etc/hadoop
```

In order to develop Hadoop programs in java, you have to reset the java environment variables in hadoop-env.sh file by replacing JAVA_HOME value with the location of java in your system.

```
export JAVA_HOME=/usr/local/jdk1.7.0_71
```

The following are the list of files that you have to edit to configure Hadoop.

core-site.xml

The core-site.xml file contains information such as the port number used for Hadoop instance, memory allocated for the file system, memory limit for storing the data, and size of Read/Write buffers.

Open the core-site.xml and add the following properties in between <configuration>, </configuration> tags.

```
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
```

hdfs-site.xml

The hdfs-site.xml file contains information such as the value of replication data, namenode path, and datanode paths of your local file systems. It means the place where you want to store the Hadoop infrastructure.

Let us assume the following data.

```
dfs.replication (data replication value) = 1
(In the below given path /hadoop/ is the user name.
hadoopinfra/hdfs/namenode is the directory created by hdfs file system.)
namenode path = //home/hadoop/hadoopinfra/hdfs/namenode
(hadoopinfra/hdfs/datanode is the directory created by hdfs file system.)
datanode path = //home/hadoop/hadoopinfra/hdfs/datanode
```

Open this file and add the following properties in between the <configuration> </configuration> tags in this file.

```
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.name.dir</name>
<value>file:///home/hadoop/hadoopinfra/hdfs/namenode
</value>
</property>
<property>
<name>dfs.data.dir</name>
<value>file:///home/hadoop/hadoopinfra/hdfs/datanode
</value>
</property>
</configuration>
```

Note : In the above file, all the property values are user-defined and you can make changes according to your Hadoop infrastructure.

yarn-site.xml

This file is used to configure yarn into Hadoop. Open the yarn-site.xml file and add the following properties in between the <configuration>, </configuration> tags in this file.

```
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
</configuration>
```

mapred-site.xml

This file is used to specify which MapReduce framework we are using. By default, Hadoop contains a template of yarn-site.xml. First of all, it is required to copy the file from mapred-site.xml.template to mapred-site.xml file using the following command.

```
$ cp mapred-site.xml.template mapred-site.xml
```

Open mapred-site.xml file and add the following properties in between the <configuration>, </configuration> tags in this file.

```
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
```

Verifying Hadoop Installation

The following steps are used to verify the Hadoop installation.

Step 1 : Name Node Setup

Set up the namenode using the command "hdfs namenode -format" as follows.

```
$ cd ~
$ hdfs namenode -format
```

The expected result is as follows.

```
10/24/14 21:30:55 INFO namenode.NameNode :
STARTUP_MSG:
/************************************************************
********
STARTUP_MSG: Starting NameNode
STARTUP_MSG: host = localhost/192.168.1.11
STARTUP_MSG: args = [-format]
STARTUP_MSG: version = 2.4.1
...
```

```
...
10/24/14 21 :30 :56 INFO common.Storage : Storage
directory
/home/hadoop/hadoopinfra/hdfs/namenode has been
successfully formatted.
10/24/14 21 :30 :56 INFO
namenode.NNStorageRetentionManager : Going to
retain 1 images with txid >= 0
10/24/14 21 :30 :56 INFO util.ExitUtil : Exiting with status 0
10/24/14 21 :30 :56 INFO namenode.NameNode :
SHUTDOWN_MSG :
/************************************************************
********
SHUTDOWN_MSG : Shutting down NameNode at
localhost/192.168.1.11
************************************************************
*******/
```

Step 2 : Verifying Hadoop dfs

The following command is used to start dfs. Executing this command will start your Hadoop file system.

```
$ start-dfs.sh
```

The expected output is as follows :

```
10/24/14 21 :37 :56
Starting namenodes on [localhost]
localhost : starting namenode, logging to
/home/hadoop/hadoop
2.4.1/logs/hadoop-hadoop-namenode-localhost.out
localhost : starting datanode, logging to /home/hadoop/hadoop
2.4.1/logs/hadoop-hadoop-datanode-localhost.out
Starting secondary namenodes [0.0.0.0]
```

Step 3 : Verifying Yarn Script

The following command is used to start the yarn script. Executing this command will start your yarn daemons.

```
$ start-yarn.sh
```

The expected output as follows :

```
starting yarn daemons
starting resourcemanager, logging to /home/hadoop/hadoop
2.4.1/logs/yarn-hadoop-resourcemanager-localhost.out
localhost : starting nodemanager, logging to
/home/hadoop/hadoop
2.4.1/logs/yarn-hadoop-nodemanager-localhost.out
```

Step 4 : Accessing Hadoop on Browser

The default port number to access Hadoop is 50070. Use the following url to get Hadoop services on browser.

```
http ://localhost :50070/
```

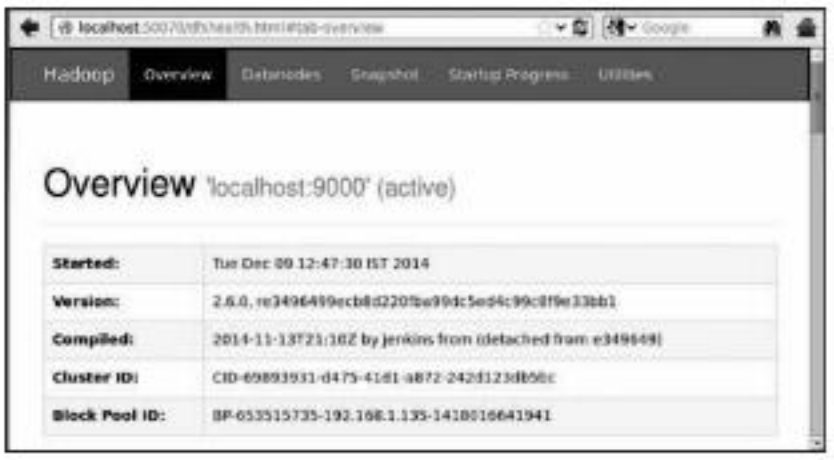


Fig. 6.8

Step 5 : Verify All Applications for Cluster

The default port number to access all applications of cluster is 8088. Use the following url to visit this service.

```
http ://localhost :8088/
```

Fig. 6.9

Hadoop File System was developed using distributed file system design. It is run on commodity hardware. Unlike other distributed systems, HDFS is highly faulttolerant and designed using low-cost hardware.

HDFS holds very large amount of data and provides easier access. To store such huge data, the files are stored across multiple machines. These files are stored in redundant fashion to rescue the system from possible data losses in case of failure. HDFS also makes applications available to parallel processing.

Features of HDFS

- It is suitable for the distributed storage and processing.
- Hadoop provides a command interface to interact with HDFS.
- The built-in servers of namenode and datanode help users to easily check the status of cluster.
- Streaming access to file system data.
- HDFS provides file permissions and authentication.

HDFS Architecture

Given below is the architecture of a Hadoop File System.

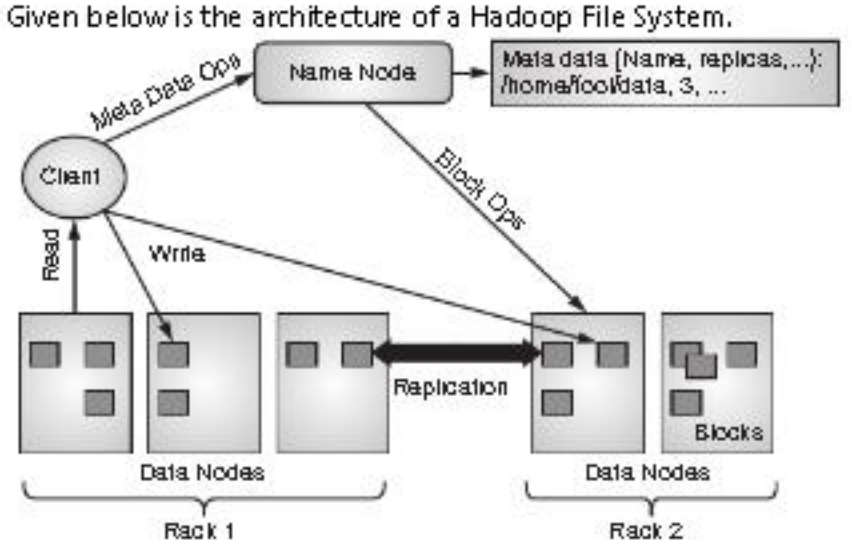


Fig. 6.10

HDFS follows the master-slave architecture and it has the following elements.

Namenode

The namenode is the commodity hardware that contains the GNU/Linux operating system and the namenode software. It is a software that can be run on commodity hardware. The system having the namenode acts as the master server and it does the following tasks :

- Manages the file system namespace.
- Regulates client's access to files.
- It also executes file system operations such as renaming, closing, and opening files and directories.

Datanode

- The datanode is a commodity hardware having the GNU/Linux operating system and datanode software. For every node (Commodity hardware/System) in a cluster, there will be a datanode. These nodes manage the data storage of their system.
- Datanodes perform read-write operations on the file systems, as per client request.
- They also perform operations such as block creation, deletion, and replication according to the instructions of the namenode.

Block

- Generally the user data is stored in the files of HDFS. The file in a file system will be divided into one or more segments and/or stored in individual data nodes. These file segments are called as blocks. In other words, the minimum amount of data that HDFS can read or write is called a Block. The default block size is 64MB, but it can be increased as per the need to change in HDFS configuration.

Goals of HDFS

- **Fault Detection and Recovery :** Since HDFS includes a large number of commodity hardware, failure of components is frequent. Therefore HDFS should have mechanisms for quick and automatic fault detection and recovery.
- **Huge Datasets :** HDFS should have hundreds of nodes per cluster to manage the applications having huge datasets.
- **Hardware at Data :** A requested task can be done efficiently, when the computation takes place near the data. Especially where huge datasets are involved, it reduces the network traffic and increases the throughput.

Starting HDFS

- Initially you have to format the configured HDFS file system, open namenode (HDFS server), and execute the following command.

```
$ hadoop namenode -format
```

- After formatting the HDFS, start the distributed file system. The following command will start the namenode as well as the data nodes as cluster.

```
$ start-dfs.sh
```

Listing Files in HDFS

- After loading the information in the server, we can find the list of files in a directory, status of a file, using 'ls'. Given below is the syntax of ls that you can pass to a directory or a filename as an argument.

```
$HADOOP_HOME/bin/hadoop fs -ls <args>
```

Inserting Data into HDFS

- Assume we have data in the file called file.txt in the local system which is ought to be saved in the hdfs file system. Follow the steps given below to insert the required file in the Hadoop file system.

Step 1

You have to create an input directory.

```
$HADOOP_HOME/bin/hadoop fs -mkdir /user/input
```

Step 2

Transfer and store a data file from local systems to the Hadoop file system using the put command.

```
$HADOOP_HOME/bin/hadoop fs -put /home/file.txt
/user/input
```

Step 3

You can verify the file using ls command.

```
$HADOOP_HOME/bin/hadoop fs -ls /user/input
```

Retrieving Data from HDFS

Assume we have a file in HDFS called outfile. Given below is a simple demonstration for retrieving the required file from the Hadoop file system.

Step 1

Initially, view the data from HDFS using cat command.

```
$HADOOP_HOME/bin/hadoop fs -cat /user/output/outfile
```

Step 2

Get the file from HDFS to the local file system using get command.

```
$HADOOP_HOME/bin/hadoop fs -get /user/output/
/home/hadoop_tp/
```

Shutting Down the HDFS

You can shut down the HDFS by using the following command.

```
$ stop-dfs.sh
```

There are many more commands in "$HADOOP_HOME/bin/hadoop fs" than are demonstrated here, although these basic operations will get you started. Running ./bin/hadoop dfs with no additional arguments will list all the commands that can be run with the FsShell system. Furthermore, $HADOOP_HOME/bin/hadoop fs -help commandName will display a short usage summary for the operation in question, if you are stuck.

A Table of all the Operations is shown below. The Following Conventions are used for Parameters :

"<path>" means any file or directory name.

"<path>..." means one or more file or directory names.

"<file>" means any filename.

"<src>" and "<dest>" are path names in a directed operation.

"<localSrc>" and "<localDest>" are paths as above, but on the local file system.

All other files and path names refer to the objects inside HDFS.

1.	ls <path> Lists the contents of the directory specified by path, showing the names, permissions, owner, size and modification date for each entry.
2.	lsr <path> Behaves like -ls, but recursively displays entries in all subdirectories of path.
3.	du <path> Shows disk usage, in bytes, for all the files which match path; filenames are reported with the full HDFS protocol prefix.
4.	dus <path> Like -du, but prints a summary of disk usage of all files/directories in the path.
5.	mv <src> <dest> Moves the file or directory indicated by src to dest, within HDFS.
6.	cp <src> <dest> Copies the file or directory identified by src to dest, within HDFS.
7.	rm <path> Removes the file or empty directory identified by path.
8.	rmr <path> Removes the file or directory identified by path. Recursively deletes any child entries (i.e., files or subdirectories of path).
9.	put <localSrc> <dest> Copies the file or directory from the local file system identified by localSrc to dest within the DFS.
10.	copyFromLocal <localSrc> <dest> Identical to -put
11.	moveFromLocal <localSrc> <dest> Copies the file or directory from the local file system identified by localSrc to dest within HDFS, and then deletes the local copy on success.
12.	get [-crc] <src> <localDest> Copies the file or directory in HDFS identified by src to the local file system path identified by localDest.
13.	getmerge <src> <localDest> Retrieves all files that match the path src in HDFS, and copies them to a single, merged file in the local file system identified by localDest.
14.	cat <filen-ame> Displays the contents of filename on stdout.
15.	copyToLocal <src> <localDest> Identical to -get
16.	moveToLocal <src> <localDest> Works like -get, but deletes the HDFS copy on success.
17.	mkdir <path> Creates a directory named path in HDFS. Creates any parent directories in path that are missing (e.g., mkdir -p in Linux).
18.	setrep [-R] [-w] rep <path> Sets the target replication factor for files identified by path to rep. (The actual replication factor will move toward the target over time)
19.	touchz <path> Creates a file at path containing the current time as a timestamp. Fails if a file already exists at path, unless the file is already size 0.
20.	test -[ezd] <path> Returns 1 if path exists; has zero length; or is a directory or 0 otherwise.
21.	stat [format] <path> Prints information about path. Format is a string which accepts file size in blocks (%b), filename (%n), block size (%o), replication (%r), and modification date (%y, %Y).
22.	tail [-f] <file2name> Shows the last 1KB of file on stdout.
23.	chmod [-R] mode,mode,... <path>... Changes the file permissions associated with one or more objects identified by path.... Performs changes recursively with R. mode is a 3-digit octal mode, or {augo}+/-{rwxX}. Assumes if no scope is specified and does not apply an umask.
24.	chown [-R] [owner][:[group]] <path>... Sets the owning user and/or group for files or directories identified by path.... Sets owner recursively if -R is specified.
25.	chgrp [-R] group <path>... Sets the owning group for files or directories identified by path.... Sets group recursively if -R is specified.
26.	help <cmd-name> Returns usage information for one of the commands listed above. You must omit the leading '-' character in cmd.

6.5 HADOOP ECOSYSTEM

- The objective of this Apache Hadoop ecosystem components tutorial is to have an overview what are the different components of Hadoop ecosystem that make Hadoop so powerful and due to which several Hadoop job roles are available now.

- We will also learn about Hadoop ecosystem components like HDFS and HDFS components, MapReduce, YARN, Hive, Apache Pig, Apache HBase and HBase components, HCatalogue, Avro, Thrift, Drill, Apache mahout, Sqoop, Apache Flume, Ambari, Zookeeper and Apache OOzie to deep dive into Big Data Hadoop and to acquire master level knowledge of the Hadoop Ecosystem.

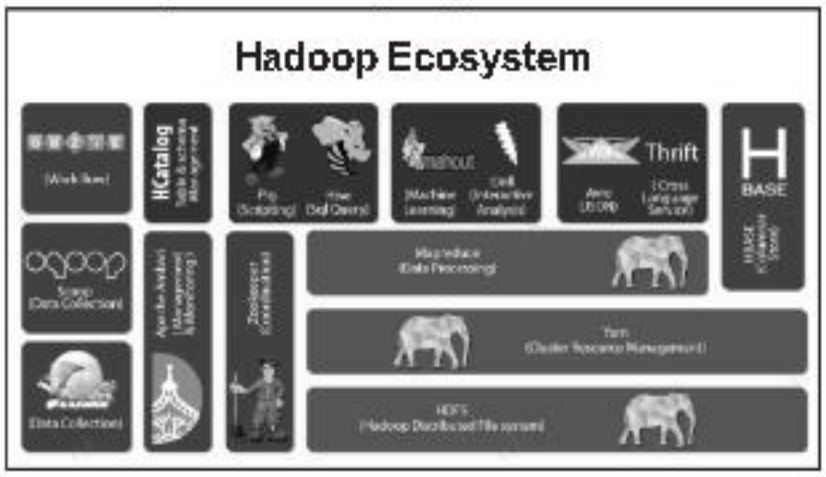


Fig. 6.11 : Hadoop ecosystem components diagram

Introduction to Hadoop Ecosystem

As we can see the different Hadoop ecosystem explained in the above Fig. 6.11 of Hadoop Ecosystem. Now we are going to discuss the list of Hadoop Components in this section one by one in detail.

6.6 HADOOP DISTRIBUTED FILE SYSTEM

It is the most important component of Hadoop Ecosystem. HDFS is the primary storage system of Hadoop. Hadoop distributed file system (HDFS) is a java based file system that provides scalable, fault tolerance, reliable and cost efficient data storage for Big data. HDFS is a distributed filesystem that runs on commodity hardware. HDFS is already configured with default configuration for many installations. Most of the time for large clusters configuration is needed. Hadoop interact directly with HDFS by shell-like commands.

HDFS Components :

There are two major components of Hadoop HDFS- NameNode and DataNode. Let's now discuss these Hadoop HDFS Components-

i. NameNode

It is also known as Master node. NameNode does not store actual data or dataset. NameNode stores Metadata i.e. number of blocks, their location, on which Rack, which Datanode the data is stored and other details. It consists of files and directories.

Tasks of HDFS NameNode

Manage file system namespace.

Regulates client's access to files.

Executes file system execution such as naming, closing, opening files and directories.

ii. DataNode

It is also known as Slave. HDFS Datanode is responsible for storing actual data in HDFS. Datanode performs read and write operation as per the request of the clients. Replica block of Datanode consists of 2 files on the file system. The first file is for data and second file is for recording the block's metadata. HDFS Metadata includes checksums for data. At startup, each Datanode connects to its corresponding Namenode and does handshaking. Verification of namespace ID and software version of DataNode take place by handshaking. At the time of mismatch found, DataNode goes down automatically.

Tasks of HDFS DataNode

DataNode performs operations like block replica creation, deletion, and replication according to the instruction of NameNode.

DataNode manages data storage of the system.

This was all about HDFS as a Hadoop Ecosystem component.

Refer HDFS Comprehensive Guide to read Hadoop HDFS in detail and then proceed with the Hadoop Ecosystem tutorial.

6.7 YARN

- Hadoop YARN (Yet Another Resource Negotiator) is a Hadoop ecosystem component that provides the resource management. Yarn is also one the most important component of Hadoop Ecosystem.
- YARN is called as the operating system of Hadoop as it is responsible for managing and monitoring workloads. It allows multiple data processing engines such as real-time streaming and batch processing to handle data stored on a single platform.

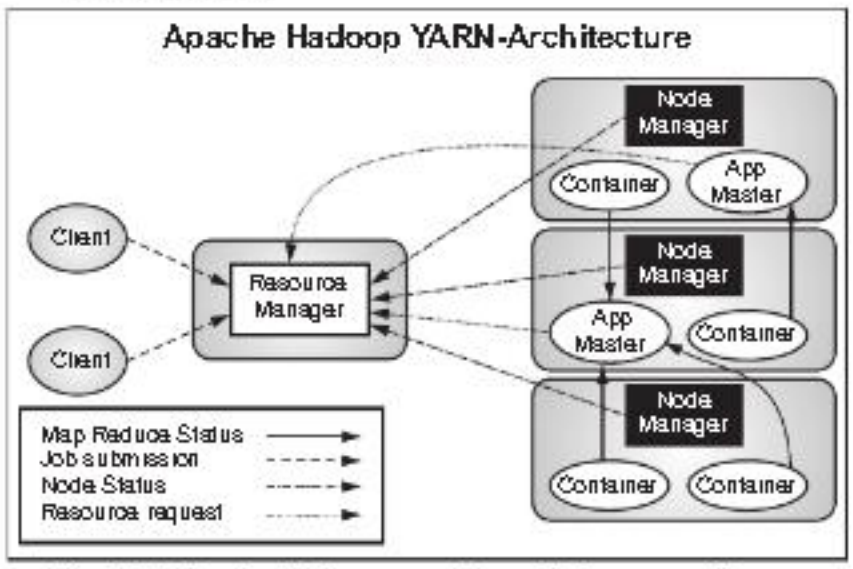


Fig. 6.12: Apache Hadoop ecosystem – Hadoop yarn diagram

YARN has been Projected as a Data Operating System for Hadoop2. Main Features of YARN are :

- **Flexibility :** Enables other purpose-built data processing models beyond MapReduce (batch), such as interactive and streaming. Due to this feature of YARN, other applications can also be run along with Map Reduce programs in Hadoop2.

- **Efficiency :** As many applications run on the same cluster, Hence, efficiency of Hadoop increases without much effect on quality of service.
- **Shared :** Provides a stable, reliable, secure foundation and shared operational services across multiple workloads. Additional programming models such as graph processing and iterative modeling are now possible for data processing.

6.8 PIG

- Apache Pig is a high-level language platform for analyzing and querying huge dataset that are stored in HDFS. Pig as a component of Hadoop Ecosystem uses PigLatin language. It is very similar to SQL.
- It loads the data, applies the required filters and dumps the data in the required format. For programs execution, pig requires Java runtime environment.

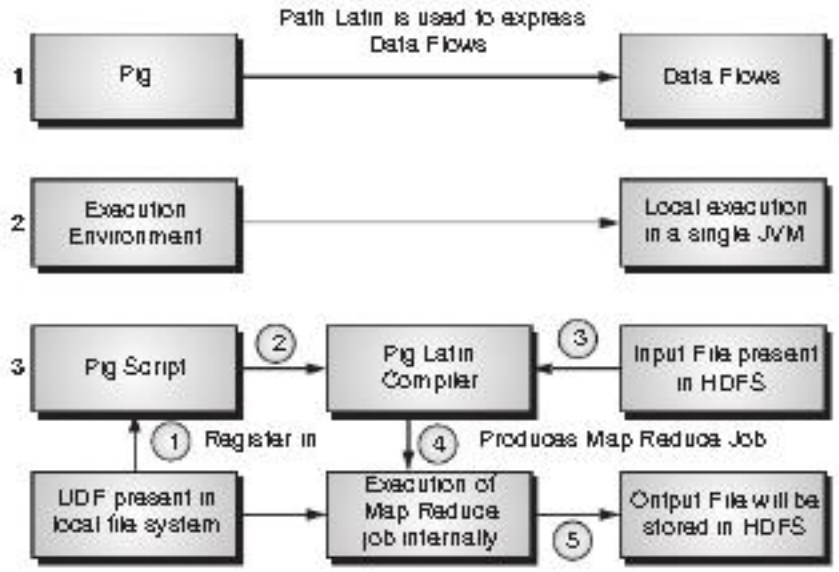


Fig. 6.13: Hadoop ecosystem tutorial – pig diagram

Features of Apache Pig:

- **Extensibility :** For carrying out special purpose processing, users can create their own function.
- **Optimization Opportunities :** Pig allows the system to optimize automatic execution. This allows the user to pay attention to semantics instead of efficiency.
- **Handles all Kinds of Data :** Pig analyzes both structured as well as unstructured.

6.9 HIVE

- The Hadoop ecosystem component, Apache Hive, is an open source data warehouse system for querying and analyzing large datasets stored in Hadoop files. Hive do three main functions : data summarization, query, and analysis.
- Hive use language called HiveQL (HQL), which is similar to SQL. HiveQL automatically translates SQL-like queries into MapReduce jobs which will execute on Hadoop.

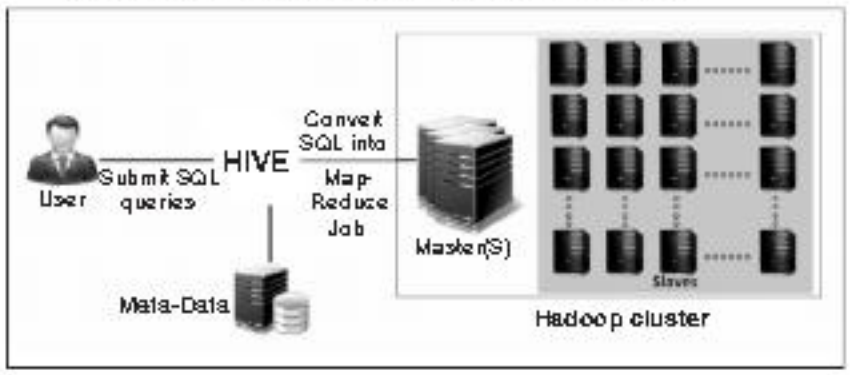


Fig. 6.14: Components of Hadoop ecosystem – hive diagram

Main Parts of Hive are :

- **Metastore :** It stores the metadata.
- **Driver :** Manage the lifecycle of a HiveQL statement.
- **Query Compiler :** Compiles HiveQL into Directed Acyclic Graph (DAG).
- **Hive Server :** Provide a thrift interface and JDBC/ODBC server.

6.10 HBASE

Apache HBase is a Hadoop ecosystem component which is distributed database that was designed to store structured data in tables that could have billions of row and millions of columns. HBase is scalable, distributed, and Nosql database that is built on top of HDFS. HBase, provide real time access to read or write data in HDFS.

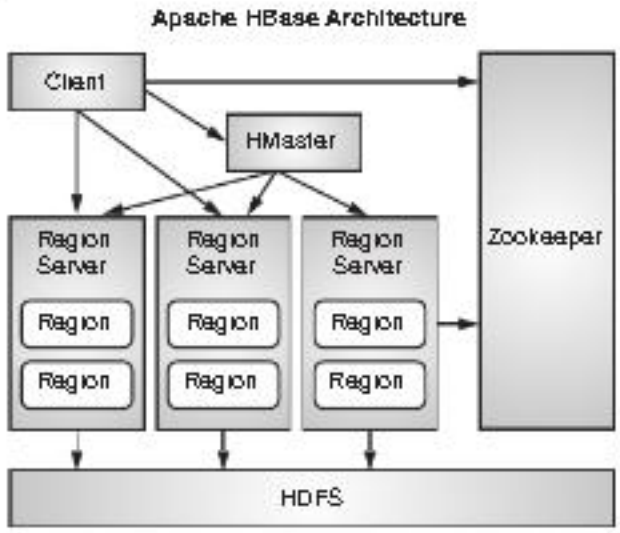


Fig. 6.15: Hadoop ecosystem components – HBase diagram

Components of Hbase

There are two HBase Components namely- HBase Master and RegionServer.

1. HBase Master

It is not part of the actual data storage but negotiates load balancing across all RegionServer.

Maintain and monitor the Hadoop cluster.

Performs administration (interface for creating, updating and deleting tables.)

Controls the failover.

HMaster handles DDL operation.

2. RegionServer

It is the worker node which handle read, write, update and delete requests from clients. Region server process runs on every node in Hadoop cluster. Region server runs on HDFS DateNode.

6.11 APACHE MAHOUT

Mahout is open source framework for creating scalable machine learning algorithm and data mining library. Once data is stored in Hadoop HDFS, mahout provides the data science tools to automatically find meaningful patterns in those big data sets.

Algorithms of Mahout are :

- **Clustering :** Here it takes the item in particular class and organizes them into naturally occurring groups, such that item belonging to the same group are similar to each other.
- **Collaborative Filtering :** It mines user behavior and makes product recommendations (e.g. Amazon recommendations)
- **Classifications :** It learns from existing categorization and then assigns unclassified items to the best category.
- **Frequent Pattern Mining :** It analyzes items in a group (e.g. items in a shopping cart or terms in query session) and then identifies which items typically appear together.

Apache Sqoop

Sqoop imports data from external sources into related Hadoop ecosystem components like HDFS, Hbase or Hive. It also exports data from Hadoop to other external sources. Sqoop works with relational databases such as teradata, Netezza, oracle, MySQL.

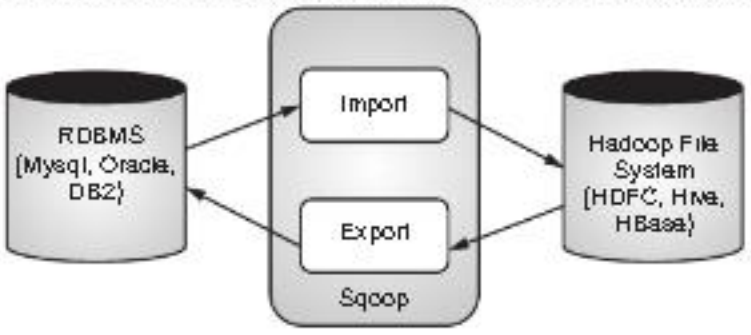


Fig. 6.16: Explain Hadoop ecosystem – apache sqoop diagram

Features of Apache Sqoop :

- **Import Sequential Datasets from Mainframe :** Sqoop satisfies the growing need to move data from the mainframe to HDFS.
- **Import Direct to ORC Files :** Improves compression and light weight indexing and improve query performance.
- **Parallel Data Transfer :** For faster performance and optimal system utilization.
- **Efficient Data Analysis :** Improve efficiency of data analysis by combining structured data and unstructured data on a schema on reading data lake.
- **Fast Data Copies :** From an external system into Hadoop.

Apache Flume

Flume efficiently collects, aggregate and moves a large amount of data from its origin and sending it back to HDFS. It is fault tolerant and reliable mechanism. This Hadoop Ecosystem component allows the data flow from the source into Hadoop environment. It uses a simple extensible data model that allows for the online analytic application. Using Flume, we can get the data from multiple servers immediately into hadoop.

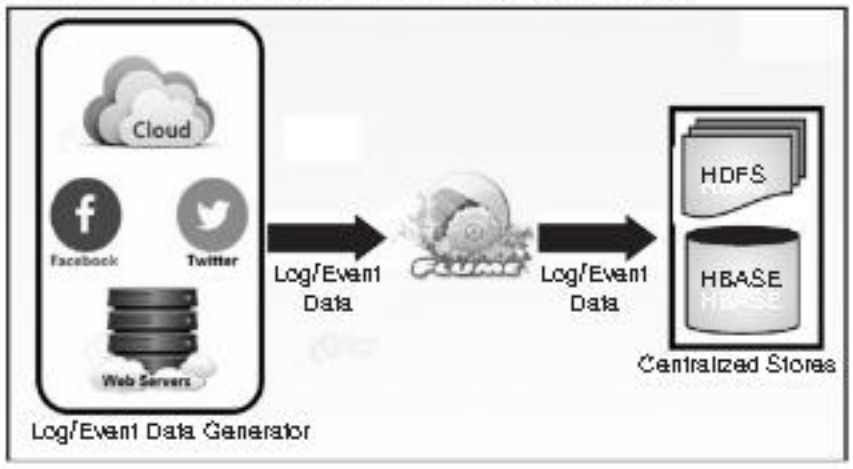


Fig. 6.17: Hadoop ecosystem components – apache flume

Refer Flume Comprehensive Guide for More Details

Ambari

Ambari, another Hadoop ecosystem component, is a management platform for provisioning, managing, monitoring and securing apache Hadoop cluster. Hadoop management gets simpler as Ambari provide consistent, secure platform for operational control.

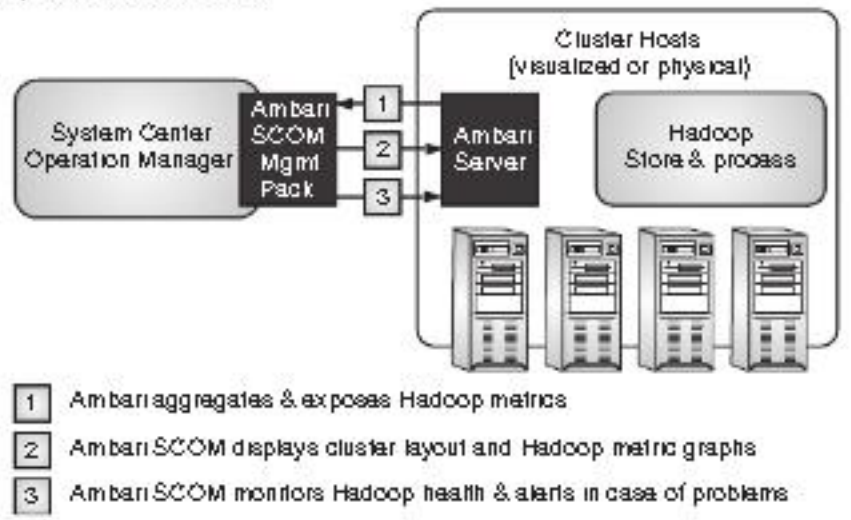


Fig. 6.18: Hadoop ecosystem tutorial – ambari diagram

Features of Ambari :

- **Simplified Installation, Configuration, and Management :** Ambari easily and efficiently create and manage clusters at scale.
- **Centralized Security Setup :** Ambari reduce the complexity to administer and configure cluster security across the entire platform.

- **Highly Extensible and Customizable :** Ambari is highly extensible for bringing custom services under management.
- **Full Visibility into Cluster Health :** Ambari ensures that the cluster is healthy and available with a holistic approach to monitoring.

Zookeeper

Apache Zookeeper is a centralized service and a Hadoop Ecosystem component for maintaining configuration information, naming, providing distributed synchronization, and providing group services. Zookeeper manages and coordinates a large cluster of machines.

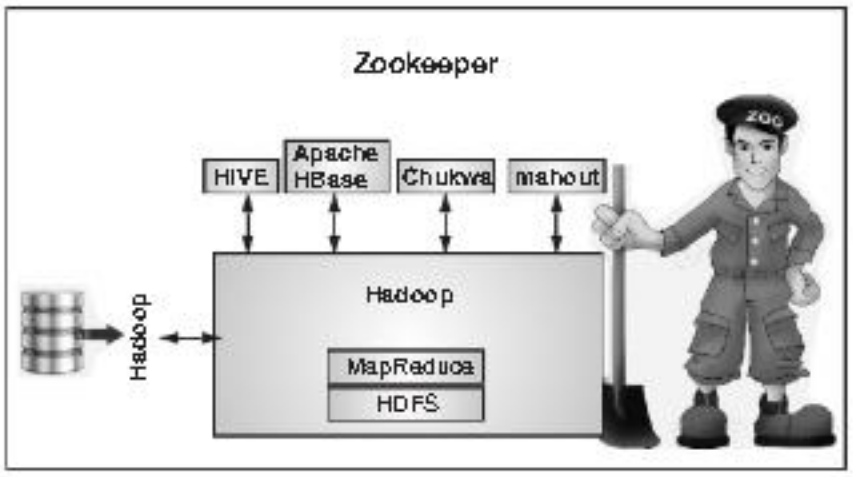


Fig. 6.19: Hadoop ecosystem explained – zookeeper diagram

Features of Zookeeper :

Fast : Zookeeper is fast with workloads where reads to data are more common than writes. The ideal read/write ratio is 10 :1.

Ordered : Zookeeper maintains a record of all transactions.

Oozie

- It is a workflow scheduler system for managing apache Hadoop jobs. Oozie combines multiple jobs sequentially into one logical unit of work. Oozie framework is fully integrated with apache Hadoop stack, YARN as an architecture center and supports Hadoop jobs for apache MapReduce, Pig, Hive, and Sqoop.

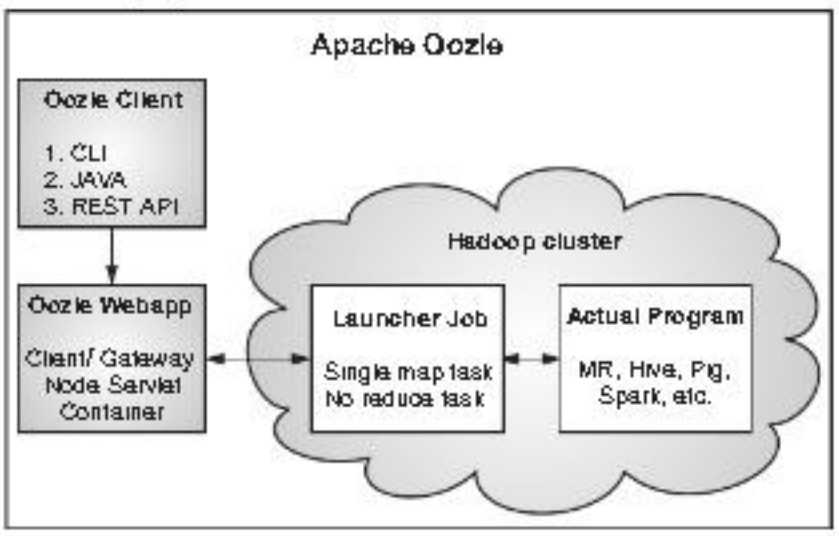


Fig. 6.20: Apache Hadoop ecosystem – Oozie diagram

- In Oozie, users can create Directed Acyclic Graph of workflow, which can run in parallel and sequentially in Hadoop. Oozie is scalable and can manage timely execution of thousands of workflow in a Hadoop cluster. Oozie is very much flexible as well. One can easily start, stop, suspend and rerun jobs. It is even possible to skip a specific failed node or rerun it in Oozie.

There are Two Basic Types of Oozie Jobs :

1. **Oozie Workflow :** It is to store and run workflows composed of Hadoop jobs e.g., MapReduce, pig, Hive.
2. **Oozie Coordinator :** It runs workflow jobs based on predefined schedules and availability of data.

6.12 NoSQL

- NoSQL is a non-relational database management systems, different from traditional relational database management systems in some significant ways. It is designed for distributed data stores where very large scale of data storing needs (for example Google or Facebook which collects terabits of data every day for their users). These type of data storing may not require fixed schema, avoid join operations and typically scale horizontally.
- In today's time data is becoming easier to access and capture through third parties such as Facebook, Google+ and others. Personal user information, social graphs, geo location data, user-generated content and machine logging data are just a few examples where the data has been increasing exponentially.
- To avail the above service properly, it is required to process huge amount of data. Which SQL databases were never designed. The evolution of NoSQL databases is to handle these huge data properly.

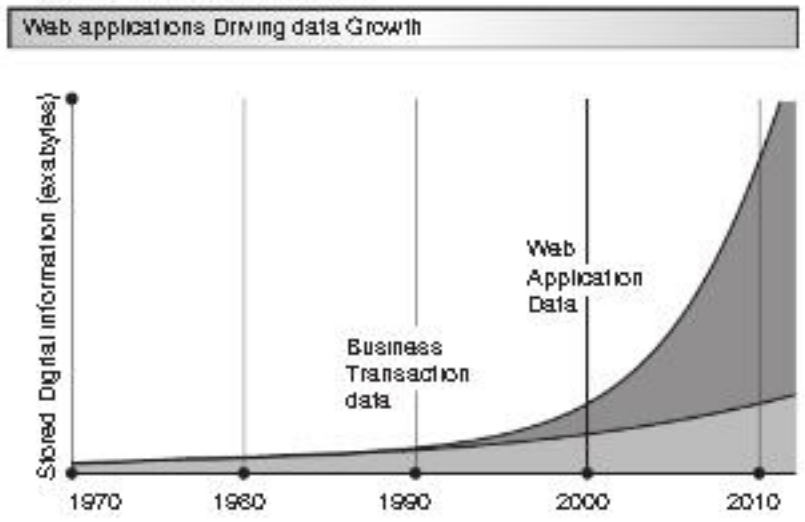


Fig. 6.21

RDBMS V/s NoSQL

RDBMS

- Structured and organized data
- Structured Query Language (SQL)

- Data and its relationships are stored in separate tables.
- Data Manipulation Language, Data Definition Language
- Tight Consistency

NoSQL

- Stands for Not Only SQL
- No declarative query language
- No predefined schema
- Key-Value pair storage, Column Store, Document Store, Graph databases
- Eventual consistency rather ACID property
- Unstructured and unpredictable data
- CAP Theorem
- Prioritizes high performance, high availability and scalability
- BASE Transaction

Brief History of NoSQL

- The term NoSQL was coined by Carlo Strozzi in the year 1998. He used this term to name his Open Source, Light Weight, DataBase which did not have an SQL interface.
- In the early 2009, when last.fm wanted to organize an event on open-source distributed databases, Eric Evans, a Rackspace employee, reused the term to refer databases which are non-relational, distributed, and does not conform to atomicity, consistency, isolation, durability - four obvious features of traditional relational database systems.
- In the same year, the "no :sql(east)" conference held in Atlanta, USA, NoSQL was discussed and debated a lot.
- And then, discussion and practice of NoSQL got a momentum, and NoSQL saw an unprecedented growth.

6.12.1 CAP Theorem (Brewer's Theorem)

You must understand the CAP theorem when you talk about NoSQL databases or in fact when designing any distributed system. CAP theorem states that there are three basic requirements which exist in a special relation when designing applications for a distributed architecture.

- **Consistency** : This means that the data in the database remains consistent after the execution of an operation. For example after an update operation all clients see the same data.
- **Availability** : This means that the system is always on (service guarantee availability), no downtime.
- **Partition Tolerance** : This means that the system continues to function even the communication among the servers is unreliable, i.e. the servers may be partitioned into multiple groups that cannot communicate with one another.

In theoretically it is impossible to fulfill all 3 requirements. CAP provides the basic requirements for a distributed system to follow 2 of the 3 requirements. Therefore all the current NoSQL database follow the different combinations of the C, A, P from the CAP theorem.

Here is the Brief Description of Three Combinations CA, CP, AP :

- **CA :** Single site cluster, therefore all nodes are always in contact. When a partition occurs, the system blocks.
- **CP :** Some data may not be accessible, but the rest is still consistent/accurate.
- **AP :** System is still available under partitioning, but some of the data returned may be inaccurate.

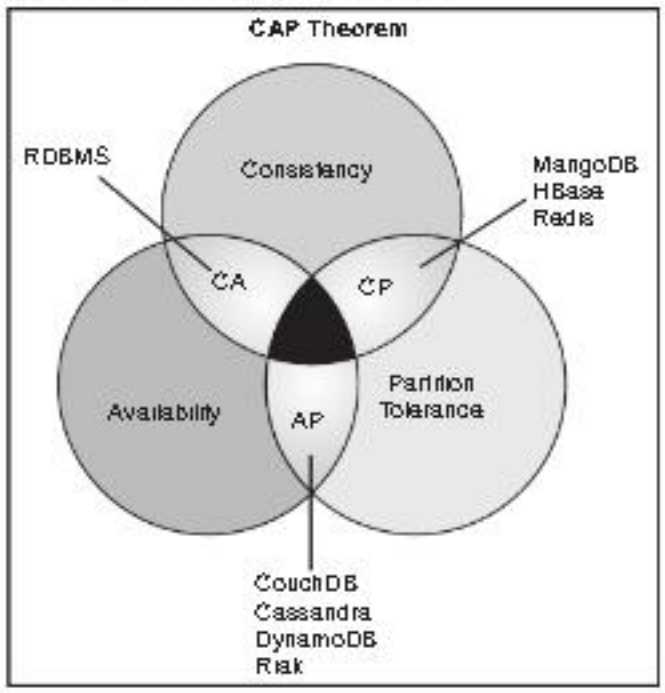


Fig. 6.22

NoSQL pros/cons

Advantages :

- High scalability.
- Distributed Computing.
- Lower cost.
- Schema flexibility, semi-structure data.
- No complicated Relationships.

Disadvantages

- No standardization.
- Limited query capabilities (so far).
- Eventual consistent is not intuitive to program for.

The BASE

The CAP theorem states that a distributed computer system cannot guarantee all of the following three properties at the same time :

1. Consistency
2. Availability
3. Partition tolerance

A BASE system gives up on consistency.

- Basically available indicates that the system does guarantee availability, in terms of the CAP theorem.
- Soft state indicates that the state of the system may change over time, even without input. This is because of the eventual consistency model.
- Eventual consistency indicates that the system will become consistent over time, given that the system doesn't receive input during that time.

ACID V/s BASE

ACID	BASE
Atomic	Basically Available
Consistency	Soft state
Isolation	Eventual consistency
Durable	

6.12.2 NoSQL Categories

There are four general types (most common categories) of NoSQL databases. Each of these categories has its own specific attributes and limitations. There is not a single solutions which is better than all the others, however there are some databases that are better to solve specific problems.

To clarify the NoSQL databases, lets discuss the most common categories :

1. Key-value stores
2. Column-oriented
3. Graph
4. Document oriented

1. Key-Value Stores

- Key-value stores are most basic types of NoSQL databases.
- Designed to handle huge amounts of data.
- Based on Amazon's Dynamo paper.
- Key value stores allow developer to store schema-less data.
- In the key-value storage, database stores data as hash table where each key is unique and the value can be string, JSON, BLOB (Binary Large OBjec) etc.
- A key may be strings, hashes, lists, sets, sorted sets and values are stored against these keys.
- For example a key-value pair might consist of a key like "Name" that is associated with a value like "Robin".
- Key-Value stores can be used as collections, dictionaries, associative arrays etc.
- Key-Value stores follow the 'Availability' and 'Partition' aspects of CAP theorem.
- Key-Values stores would work well for shopping cart contents, or individual values like color schemes, a landing page URI, or a default account number.

Example of Key-value store DataBase : Redis, Dynamo, Riak. etc.

Pictorial Presentation :

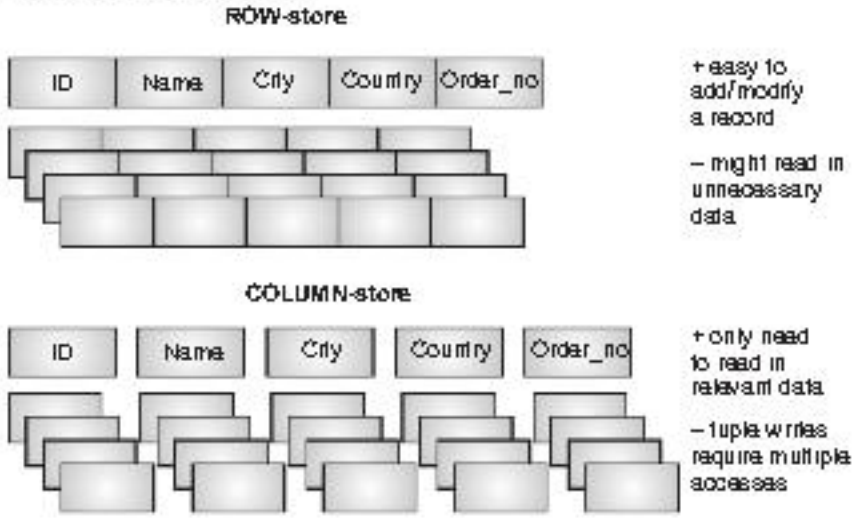


Fig. 6.23

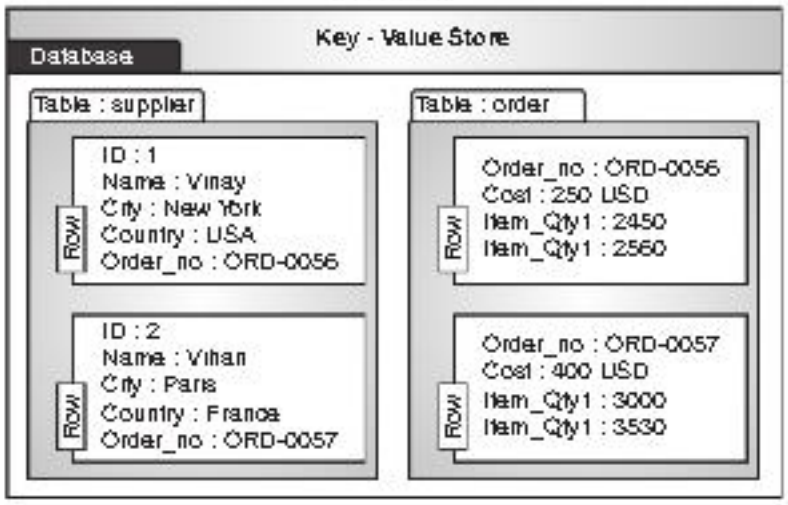


Fig. 6.24

Column-Oriented Databases

- Column-oriented databases primarily work on columns and every column is treated individually.
- Values of a single column are stored contiguously.
- Column stores data in column specific files.
- In Column stores, query processors work on columns too.
- All data within each column datafile have the same type which makes it ideal for compression.
- Column stores can improve the performance of queries as it can access specific column data.
- High performance on aggregation queries (e.g. COUNT, SUM, AVG, MIN, MAX).
- Works on data warehouses and business intelligence, Customer Relationship Management (CRM), Library card catalogs etc.

Example of Column-oriented databases : BigTable, Cassandra, SimpleDB etc.

Pictorial Presentation :

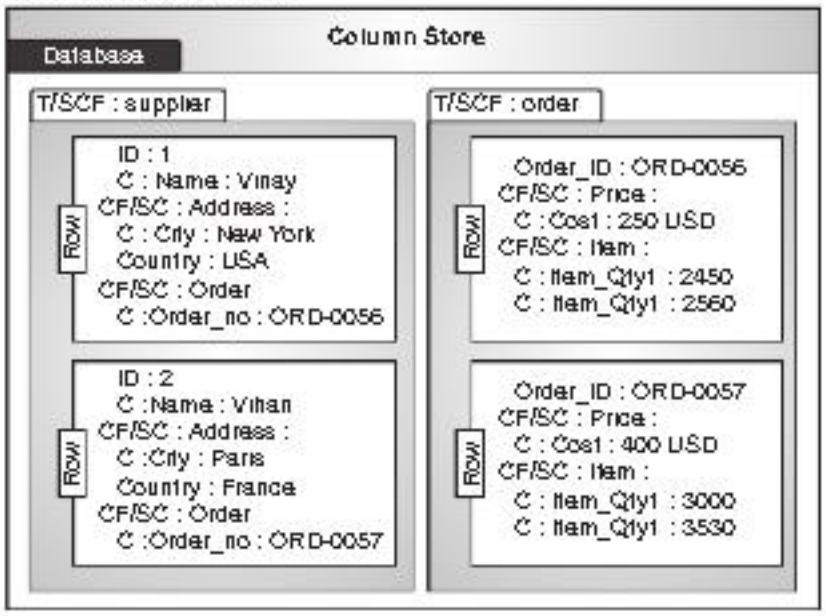


Fig. 6.25

Graph Databases

A graph data structure consists of a finite (and possibly mutable) set of ordered pairs, called edges or arcs, of certain entities called nodes or vertices.

The following picture presents a labeled graph of 6 vertices and 7 edges.

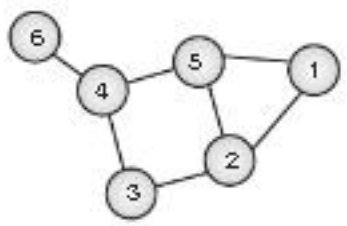


Fig. 6.26

What is a Graph Databases?

- A graph database stores data in a graph.
- It is capable of elegantly representing any kind of data in a highly accessible way.
- A graph database is a collection of nodes and edges
- Each node represents an entity (such as a student or business) and each edge represents a connection or relationship between two nodes.
- Every node and edge are defined by a unique identifier.
- Each node knows its adjacent nodes.
- As the number of nodes increases, the cost of a local step (or hop) remains the same.
- Index for lookups.

Here is a comparison between the classic relational model and the graph model :

Relational Model	Graph Model
Tables	Vertices and Edges set
Rows	Vertices
Columns	Key/value pairs
Joins	Edges

Example of Graph Databases : OrientDB, Neo4J, Titan.etc.

Pictorial Presentation :

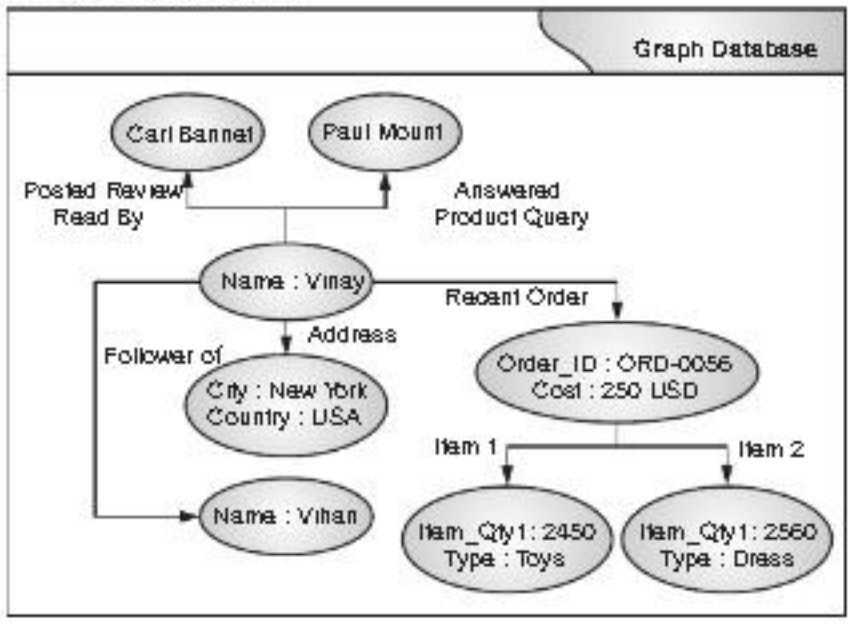


Fig. 6.27

Document Oriented Databases

- A collection of documents
- Data in this model is stored inside documents.
- A document is a key value collection where the key allows access to its value.
- Documents are not typically forced to have a schema and therefore are flexible and easy to change.
- Documents are stored into collections in order to group different kinds of data.
- Documents can contain many different key-value pairs, or key-array pairs, or even nested documents.

Here is a comparison between the classic relational model and the document model :

Relational Model	Document Model
Tables	Collections
Rows	Documents
Columns	Key/value pairs
Joins	not available

Example of Document Oriented databases : MongoDB, CouchDB etc.

Pictorial Presentation :

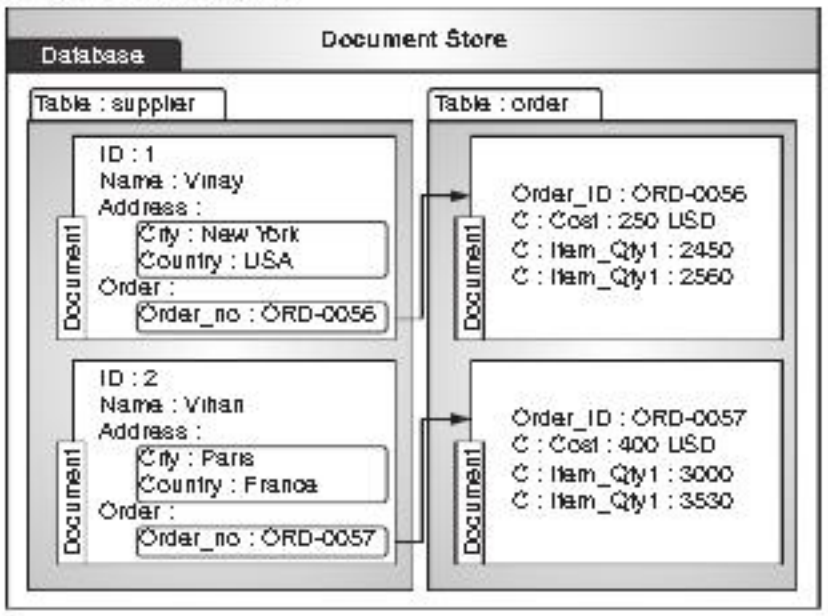


Fig. 6.28

6.13 AN ANALYTICS PROJECT- COMMUNICATING, OPERATIONALIZING, CREATING FINAL DELIVERABLES

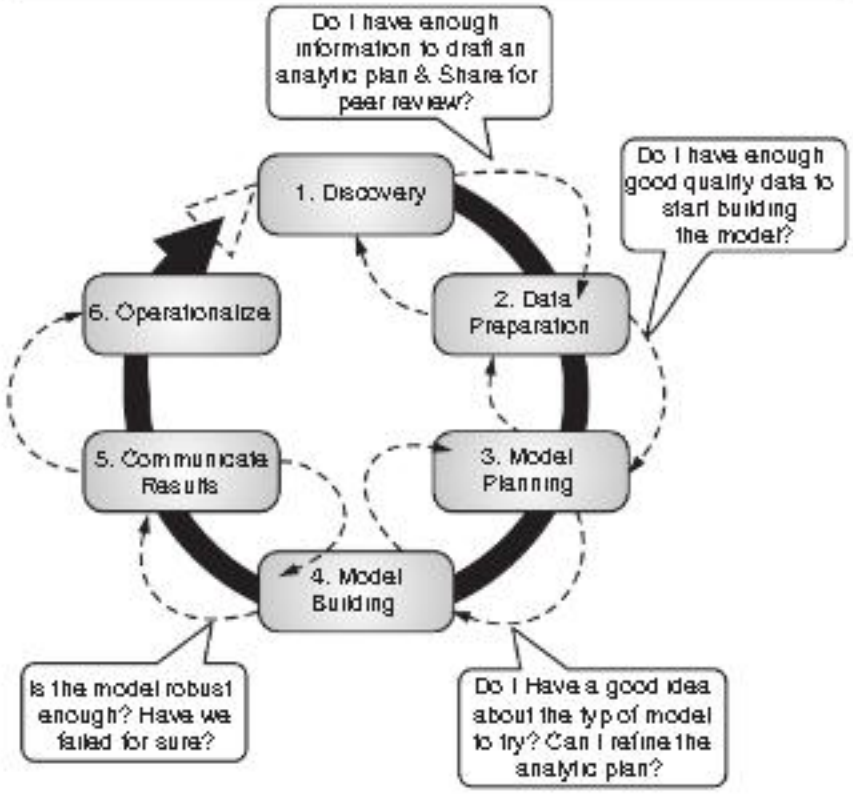


Fig. 6.29 : Data analytics lifecycle

Phase 1–Discovery : In Phase 1, the team learns the business domain, including relevant history such as whether the organization or business unit has attempted similar projects in the past from which they can learn. The team assesses the resources available to support the project in terms of people, technology, time, and data. Important activities in this phase include framing the business problem as an analytics challenge that can be addressed in subsequent phases and formulating Initial Hypotheses (IHs) to test and begin learning the data.

Phase 2–Data Preparation : Phase 2 requires the presence of an analytic sandbox, in which the team can work with data and perform analytics for the duration of the project. The team needs to execute extract, load, and transform (ELT) or extract, transform and load (ETL) to get data into the sandbox. The ELT and ETL are sometimes abbreviated as ETLT. Data should be transformed in the ETLT process so the team can work with it and analyze it. In this phase, the team also needs to familiarize itself with the data thoroughly and take steps to condition the data.

Phase 3–Model Planning : Phase 3 is model planning, where the team determines the methods, techniques, and workflow it intends to follow for the subsequent model building phase. The team explores the data to learn about the relationships between variables and subsequently selects key variables and the most suitable models.

Phase 4–Model Building : In Phase 4, the team develops datasets for testing, training, and production purposes. In addition, in this phase the team builds and executes models based on the work done in the model planning phase. The team also considers whether its existing tools will suffice for running the models, or if it will need a more robust environment for executing models and workflows (for example, fast hardware and parallel processing, if applicable).

Phase 5–Communicate Results : In Phase 5, the team, in collaboration with major stakeholders, determines if the results of the project are a success or a failure based on the criteria developed in Phase 1. The team should identify key findings, quantify the business value, and develop a narrative to summarize and convey findings to stakeholders.

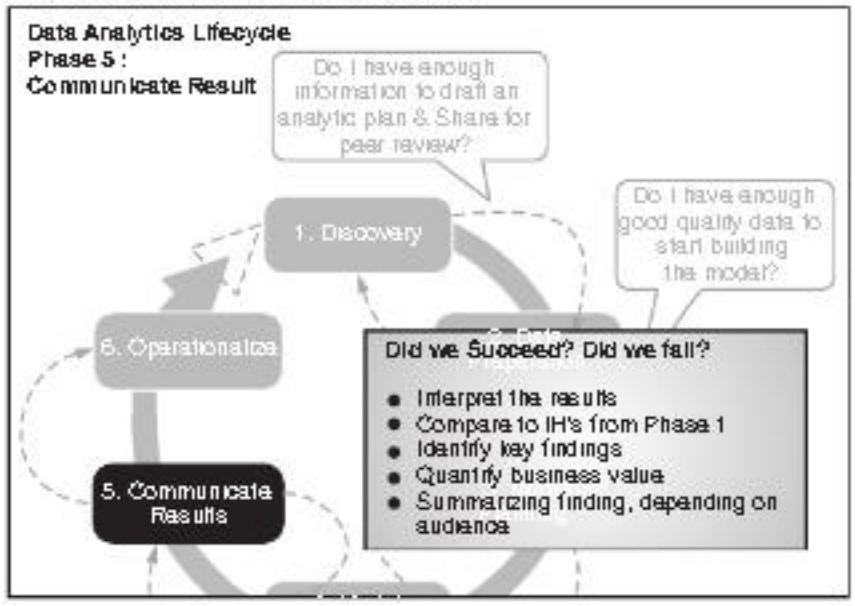


Fig. 6.30

Phase 6–Operationalize : In Phase 6, the team delivers final reports, briefings, code, and technical documents. In addition, the team may run a pilot project to implement the models in a production environment.

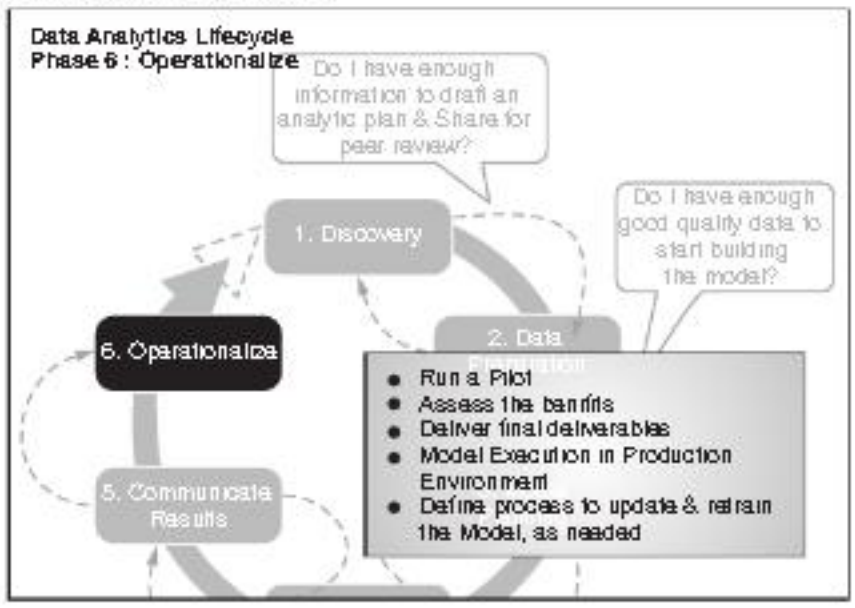


Fig. 6.31

EXERCISE

1. Explain in brief analytics for unstructured data.
2. Write a short note on MapReduce.
3. Explain apache Hadoop with architecture.
4. Explain Hadoop ecosystems.
5. Explain Pig with example.
6. Explain HIVE with its architecture.
7. What is Hbase data model.
8. Explain the data analytics lifecycle.
9. Explain use cases.
10. Explain YARN.
11. Explain key value store in NoSQL.
12. Write a note on Mahout.
13. Explain NoSQL with advantages.