MCAS4110: DATA WAREHOUSING AND MINING

AIM: To studies the basic principles of data mining and data warehousing architecture.

Unit-I **11 periods**

Data Mining: Introduction – Information and production factor – Data mining Vs Query tools – Data and machine learning- Machine learning and statistics-Data Mining in marketing – Data Mining and ethics- Nuggets and data mining- Database Mining – A performance and database Perspective- Self learning computer systems – Concept learning – Data mining and the Data Warehousing

Unit-II **13 periods**

Knowledge Discovery Process : Knowledge discovery process – Data selection – Cleaning – Enrichment – Coding – Preliminary analysis of the data set using traditional query tools – Visualization techniques – Knowledge representation- Decision trees – Classification rules- Association rules –Rules with exceptions- rules involving relations- Trees for numeric - Instance-based representation- Neural Networks – Genetic Algorithms – Clustering - KDD (Knowledge Discovery in Databases) Environment.

Unit-III **13 periods**

Dataware House – Architecture: Data warehouse Architecture – System Process – Process Architecture – Design – Database Schema – Partitioning Strategy – Aggregations – Data Marting – Meta Data – System and Data Warehouse Process Managers.

Unit-IV **12 periods**

Hardware and Operational Design: Hardware and operational design of Data Warehouse – Hardware Architecture – Physical Layout – Security – Backup and Recovery – Service – Level Agreement – Operating the Warehouse.

Unit-V **11 periods**

Planning- Tuning and Testing: Capacity planning – Tuning the Data Warehouse – Testing Warehouses – Data Warehouse Features.

Text Books:

1. Pieter Adriaans, Dolf zantinge, “Data Mining”, Pearson Education, 2007.
1. Sam Anahory, Dennis Murray, “Data Warehousing in the real world – A Practical Guide for Building Decision Support Systems”, Pearson Education, 2006.

Reference Books:

1. Ian.H.Witten & Eibe Frank, “Data Mining – Practical Machine Learning Tools and Techniques, Morgan Kaufmann Publishers, 2006.
2. Jiawei Han, Micheline Kamber, “Data Mining: Concepts and Techniques” Morgan Kaufmann Publishers, 2000.
3. Hanand J and M. Kamber, “Data Mining: Concepts and Techniques”, Second Edition, Morgan Kaufman, 2006.

DATA WAREHOUSING AND MINING

Unit-I

Data Mining:

Extraction of Hidden Information from large Database.

Data mining is the process of analyzing hidden patterns of data according to different perspectives for categorization into useful information, which is collected and assembled in common areas, such as data warehouses, for efficient analysis, data mining algorithms, facilitating business decision making and other information requirements to ultimately cut costs and increase revenue.

Data mining is also known as data discovery and knowledge discovery

E.g.: Pharmacy Company, Credit Card Company, Transport Company etc…,

Information and Production Factor

Evolution of Data Mining:

In 1960, Collection of Data

In 1980, Data accessing

In 1990, Data Warehousing

Today, Data Mining

	8 bits	1 byte
1000	1000 bytes	1 kilobyte
1000^2	1000 000	1 megabyte
1000^3	1000 000 000	1 gigabyte
1000^4	1000 000 000 000	1 terabyte
1000^5	1000 000 000 000 000	1 petabyte
1000^6	1000 000 000 000 000 000	1 exabyte
1000^7	1000 000 000 000 000 000 000	1zettabyte
1000^8	1000 000 000 000 000 000 000 000	1 yottabyte
1000^9	1000 000 000 000 000 000 000 000 000	1 brontobyte
1000^{10}	1000 000 000 000 000 000 000 000 000 000	1 geopbyte

Types of Data

Data mining can be performed on following types of data

Relational databases

Data warehouses

Advanced DB and information repositories

Object-oriented and object-relational databases

Transactional and Spatial databases

Heterogeneous and legacy databases

Multimedia and streaming database

Text databases

Text mining and Web mining

DATA MINING IN MARKETTING

Although it is still a relatively new technology, businesses from all industry verticals i.e. healthcare, manufacturing, financial, transportation, etc. have invested in data mining technology to take advantage of historical data. Data mining techniques in CRM assist your business in finding and selecting the relevant information that can then be used to get a holistic view of the customer life-cycle; this comprises of four stages: customer identification, attraction, retention, and development. The more data there is in the database, the more accurate the models will be created and their subsequent use will result in more business value.

Data mining typically involves the use of predictive modeling, forecasting and descriptive modeling techniques as its key elements. Exploiting CRM in this age of **data analytics** enables an organization to manage customer retention, select the right prospects & customer segments, set optimal pricing policies, and objectively measure and rank the suppliers best suited for their needs.

Basket Analysis

Ascertain which items customers tend to purchase together. This knowledge can improve stocking, store layout strategies, and promotions.

Sales Forecasting

Examining time-based patterns helps businesses make re-stocking decisions. Furthermore, it helps you in supply chain management, financial management and gives complete control over internal operations.

Database Marketing

Retailers can design profiles of customers based on demographics, tastes, preferences, buying behavior, etc. It will also aid the marketing team in designing personalized marketing campaigns and promotional offers. This will result in enhanced productivity, optimal allocation of the company's resources and desirable ROI.

Predictive Life-Cycle Management

Data mining helps an organization predict each customer's lifetime value and to service each segment appropriately.

Market Segmentation

Learn which customers are interested in purchasing your products and design your marketing campaigns and promotions keeping their tastes and preferences in mind. This will increase efficiency and result in the desired ROI since you won't be targeting customers who show little to no interest in your product.

Product Customization

Manufacturers can customize products according to the exact needs of customers. In order to do this, they must be able to predict which features should be bundled to meet customer demand.

Fraud Detection

By analyzing past transactions that were later determined to be fraudulent, a business can take corrective measures and stop such events from occurring in the future. Banks and other financial institutions will benefit from this feature immensely, by reducing the number of bad debts.

Warranties

Manufacturers need to predict the number of customers who will make warranty claims and the average cost of those claims. This will ensure efficient and effective management of company funds.

TECHNIQUES FOR DATA MINING IN CRM

Anomaly Detection

Searching for information that doesn’t match expected behavior or a projected pattern is called anomaly detection. Anomalies can provide actionable information because they deviate from the average in the data set.

Association Rule Learning

Discover relations between data items in huge databases. With Association Rule Learning, hidden patterns can be uncovered and the information gained may be used to better understand customers, learn their habits, and predict their decisions.

Clustering

Identify similar data sets and understand both the similarities and the differences within the data. Data sets that have similar traits can be used for conversion rate increases. For example, if the buying behavior of one group of customers is similar to that of another group, they can both be targeted with similar services or products.

Classification

This technique is used for gathering information about data so that the data sets can be placed into proper categories. One example is the classification of email as either regular, acceptable email or as spam.

Regression

Regression analysis is one of the advanced data mining techniques in CRM. The objective is to find the dependency between different data items and map out which variables are affected by other variables. This technique is used to determine customer satisfaction levels and its impact on customer loyalty.

Data mining and Ethics

1. Selecting the wrong problem for data mining

2. Ignoring, What your sponsor thinks data mining is, and what it can and cannot do

4. Leaving insufficient time for data preparation

5. Looking only at aggregated results, never at individual records.

6. Being nonchalant about keeping track of mining procedure and results

7. Ignoring suspicious findings in your haste to move on

7. Running mining algorithms repeatedly without thinking hard enough about the next

Stages of the data analysis

8. Believing everything you are told about the data

9. Believing everything you are told about your own data mining analyses

10. Measuring your results differently from the way your own sponsor will measure them

Nuggets of Data Mining

· User friendly, intuitive interface
· Available for desktop or as on line real time data mining engine
· Power to extract knowledge from data that other methods can not
· Automatic rule generation in English “if-then” rules
· Ability to model up to 50,000 variables (without using clustering)
· Employs machine learning (No statistics used)
· Automatic binning of numeric variables
· Binning of nominal variables
· Ability to handle complex non-linear relationships with no statistical requirements
· Handles missing data

- Handles noisy data
- Assists in finding data errors
- Provides validation module
- Provides predictions for new data
- Reverse engineers information implicit in databases

Data Mining Technologies

- Allows stratified sampling for training files
- Unique feature that resolves rule conflicts for better predictions
- Computes attribute significance without limitation of correlated variables

Areas of Potential Application

The following list includes only a few of the possible applications.

Business

- CRM
- Banking -- mortgage approval, loan underwriting, fraud analysis and detection
- Finance -- analysis and forecasting of business performance, stock and bond analysis
- Insurance -- bankruptcy prediction, risk analysis, credit and collection models
- Web Marketing – personalization, targeted banner ads and cross sell/upsell opportunities
- Direct Marketing – response models, churn models, optimum creative, next to buy analysis
- Government – threat assessment, terrorist profiling

Manufacturing

- Fault analysis, quality control, preventive maintenance scheduling, automated systems

Medicine

- Gene analysis, epidemiological studies, toxicology, diagnosis, drug interactions, risk factor analysis,
quality control, retrospective drug studies

Scientific Research

- General modeling of all types

Self learning

Self-learning as machine learning paradigm was introduced in 1982 along with a neural network capable of self-learning named Crossbar Adaptive Array (CAA). It is learning with no external rewards and no external teacher advices. The CAA self-learning algorithm computes, in a crossbar fashion, both decisions about actions and emotions (feelings) about consequence situations. The system is driven by the interaction between cognition and emotion. [29] The self-learning algorithm updates a memory matrix $W =||w(a,s)||$ such that in each iteration executes the following machine learning routine:

```
In situation s perform action a;
Receive consequence situation s';
Compute emotion of being in consequence situation v(s');
```

```
Update crossbar memory  $w'(a,s) = w(a,s) + v(s')$.
```

It is a system with only one input, situation s, and only one output, action (or behavior) a. There is neither a separate reinforcement input nor an advice input from the environment. The back propagated value (secondary reinforcement) is the emotion toward the consequence situation. The CAA exists in two environments, one is behavioral environment where it behaves, and the other is genetic environment, wherefrom it initially and only once receives initial emotions about situations to be encountered in the behavioral environment. After receiving the genome (species) vector from the genetic environment, the CAA learns a goal seeking behavior, in an environment that contains both desirable and undesirable situations. [30]

What is Data warehouse?

A data warehouse is a technique for collecting and managing data from varied sources to provide meaningful business insights. It is a blend of technologies and components which allows the strategic use of data.

Data Warehouse is electronic storage of a large amount of information by a business which is designed for query and analysis instead of transaction processing. It is a process of transforming data into information and making it available to users for analysis.

What Is Data Mining?

Data mining is looking for hidden, valid, and potentially useful patterns in huge data sets. Data Mining is all about discovering unsuspected/ previously unknown relationships amongst the data.

It is a multi-disciplinary skill that uses machine learning, statistics, AI and database technology.

The insights extracted via Data mining can be used for marketing, fraud detection, and scientific discovery, etc.

Data Mining Vs Data Warehouse: Key Differences

Data Mining	**Data Warehouse**

Data mining is the process of analyzing unknown patterns of data.	A data warehouse is database system which is designed for analytical instead of transactional work.
Data mining is a method of comparing large amounts of data to finding right patterns.	Data warehousing is a method of centralizing data from different sources into one common repository.
Data mining is usually done by business users with the assistance of engineers.	Data warehousing is a process which needs to occur before any data mining can take place.
Data mining is the considered as a process of extracting data from large data sets.	On the other hand, Data warehousing is the process of pooling all relevant data together.
One of the most important benefits of data mining techniques is the detection and identification of errors in the system.	One of the pros of Data Warehouse is its ability to update consistently. That's why it is ideal for the business owner who wants the best and latest features.
Data mining helps to create suggestive patterns of important factors. Like the buying habits of customers, products, sales. So that, companies can make the necessary adjustments in operation and production.	Data Warehouse adds an extra value to operational business systems like CRM systems when the warehouse is integrated.
The Data mining techniques are never 100% accurate and may cause serious	In the data warehouse, there is great chance that the data which was required for analysis by the organization may not

consequences in certain conditions.	be integrated into the warehouse. It can easily lead to loss of information.
The information gathered based on Data Mining by organizations can be misused against a group of people.	Data warehouses are created for a huge IT project. Therefore, it involves high maintenance system which can impact the revenue of medium to small-scale organizations.
After successful initial queries, users may ask more complicated queries which would increase the workload.	Data Warehouse is complicated to implement and maintain.
Organisations can benefit from this analytical tool by equipping pertinent and usable knowledge-based information.	Data warehouse stores a large amount of historical data which helps users to analyze different time periods and trends for making future predictions.
Organisations need to spend lots of their resources for training and Implementation purpose. Moreover, data mining tools work in different manners due to different algorithms employed in their design.	In Data warehouse, data is pooled from multiple sources. The data needs to be cleaned and transformed. This could be a challenge.
The data mining methods are cost-effective and efficient compares to other statistical data applications.	Data warehouse's responsibility is to simplify every type of business data. Most of the work that will be done on user's part is inputting the raw data.
Another critical benefit of data mining	Data warehouse allows users to access

techniques is the identification of errors which can lead to losses. Generated data could be used to detect a drop-in sale.	critical data from the number of sources in a single place. Therefore, it saves user's time of retrieving data from multiple sources.
Data mining helps to generate actionable strategies built on data insights.	Once you input any information into Data warehouse system, you will unlikely to lose track of this data again. You need to conduct a quick search, helps you to find the right statistic information.

Why use Data Warehouse?

Some most Important reasons for using Data warehouse are:

- Integrates many sources of data and helps to decrease stress on a production system.
- Optimized Data for reading access and consecutive disk scans.
- Data Warehouse helps to protect Data from the source system upgrades.
- Allows users to perform master Data Management.
- Improve data quality in source systems.

Why use Data mining?

Some most important reasons for using Data mining are:

- Establish relevance and relationships amongst data. Use this information to generate profitable insights
- Business can mak informed decisions quickly
- Helps to find out unusual shopping patterns in grocery stores.
- Optimize website business by providing customize offers to each visitor.
- Helps to measure customer's response rates in business marketing.
- Creating and maintaining new customer groups for marketing purposes.
- Predict customer defections, like which customers are more likely to switch to another supplier in the nearest future.
- Differentiate between profitable and unprofitable customers.
- Identify all kind of suspicious behavior, as part of a fraud detection process.

Summary:

- A data warehouse is a blend of technologies and components which allows the strategic use of data. It is a process of centralizing data from different sources into one common repository.
- Data mining is looking for hidden, valid, and potentially useful patterns in huge data sets.
- Data Warehouse helps to protect Data from the source system upgrades.
- Data warehouses are used by data scientists, business intelligence developers, to analyze data.

Data mining technology helps businesses to reach closer to their objectives.

Database Mining

Data part of it, needs no introduction. For a data science product data is enough but for a good Data Science product good and sufficient data is needed and that is primary task of Data Mining which we will discuss in detail

Data mining is a very first step of Data Science product. Data mining is a field where we try to identify patterns in data and come up with initial insights.

E.g., you got the data and you identified missing values then you saw that missing values are mostly coming from recordings taken manually.

Few people mistake Data mining with data extraction. Data mining comes into play once you have collected data.

Companies use powerful data mining techniques coupled with advanced tools to extract valuable information out of large amount of data.

E.g., Walmart collects point of sales data from their 3,000+ stores across the world and stores it into their Data Warehouse. Walmart suppliers have access to this database and they identify the buying patterns among Walmart customers and use this to maintain their inventory in future. Walmart data warehouse processes more than a million such queries every year.
Data mining uses power of machine learning, statistics and database techniques to mine large databases and come up with patterns.

Mostly data mining uses cluster analysis, anomaly detection, association rule mining etc. to find out patterns in data.

In short Data Mining is finding out hidden and interesting patterns stored in large data warehouses using the power of statistics, artificial intelligence, machine learning and database management techniques.

Statistics
Statistics is the base of all Data Mining and Machine learning algorithms.

Statistics is the study of collecting, analyzing and studying data and come up with inferences and prediction about future.

Major task of a statistician is to estimate population from sample metrics. Statistics also deal with designing surveys and experiments in order to get quality data which can further be used to make estimation about the population. If we have to formally list down the task of statistics, it will be as follows

- Designing surveys and experiments
- Summarizing and understanding data
- Estimating population behavior
- Prediction or estimation of future

Statistics is used to summarize numbers for example finding out descriptive statistics like Mean, Median, Mode, Standard Deviation, Variance, Percentiles, Testing hypotheses etc.

Machine Learning

Machine learning is a part of data science which majorly focuses on writing algorithms in a way such that machines (Computers) are able to learn on their own and use the learnings to tell about new dataset whenever it comes in.Machine learning uses power of statistics and learns from the training dataset. For example, we use regressions, classifications etc. to learn from training data and use those learnings to estimate test dataset.

- Machine Learning and Statistics both are concerned on how we learn from data but statistics is more concerned about the inference that can be drawn from the model whereas machine learning focuses on optimization and performance.
- Statistical learning involves forming a hypothesis (making assumptions that are validated before building models) before building a model. In machine learning models, the machine learning algorithms are directly run on the model making the data speak instead of guiding it in a specific direction with initial hypothesis.
- Statistics is all about sample, population, and hypothesis whereas machine learning is all about predictions, supervised and unsupervised learning.
- Machine Learning is about building algorithms that help machines emulate human learning whereas Statistics is about converting the data into aggregate numbers which help understand the structure in data.

Concept Learning

Inferring a boolean-valued function from training examples of its input and output. Acquiring the definition of a general category from given sample positive and negative training examples of the category. Concept Learning can seen as a problem of searching through a predefined space of potential hypotheses for the hypothesis that best fits the training examples.

Unit II

What is the KDD Process?

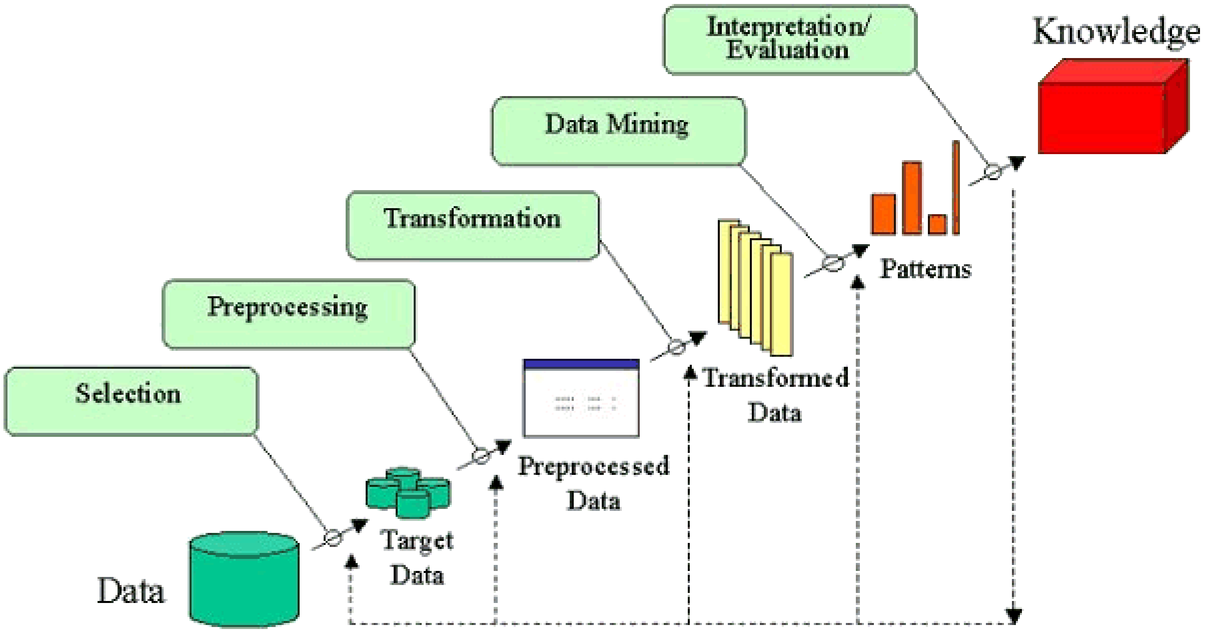
The term Knowledge Discovery in Databases, or KDD for short, refers to the broad process of finding knowledge in data, and emphasizes the "high-level" application of particular data mining methods. It is of interest to researchers in machine learning, pattern recognition, databases, statistics, artificial intelligence, knowledge acquisition for expert systems, and data visualization.

The unifying goal of the KDD process is to extract knowledge from data in the context of large databases.

It does this by using data mining methods (algorithms) to extract (identify) what is deemed knowledge, according to the specifications of measures and thresholds, using a database along with any required preprocessing, subsampling, and transformations of that database.

Some people don't differentiate data mining from knowledge discovery while others view data mining as an essential step in the process of knowledge discovery. Here is the list of steps involved in the knowledge discovery process

An Outline of the Steps of the KDD Process



The overall process of finding and interpreting patterns from data involves the repeated application of the following steps:

1. Developing an understanding of
    - the application domain
    - the relevant prior knowledge
    - the goals of the end-user
2. Creating a target data set: selecting a data set, or focusing on a subset of variables, or data samples, on which discovery is to be performed.
3. Data cleaning and preprocessing.
    - Removal of noise or outliers.
    - Collecting necessary information to model or account for noise.
    - Strategies for handling missing data fields.
    - Accounting for time sequence information and known changes.

Data Preprocessing

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues.

Data preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format.

Steps Involved in Data Preprocessing:

1. Data Cleaning:

The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.

- **(a). Missing Data:**
  This situation arises when some data is missing in the data. It can be handled in various ways.
  Some of them are:
    1. **Ignore the tuples:**
       This approach is suitable only when the dataset we have is quite large and multiple values are missing within a tuple.
    2. **Fill the Missing values:**
       There are various ways to do this task. You can choose to fill the missing values manually, by attribute mean or the most probable value.
- **(b). Noisy Data:**
  Noisy data is a meaningless data that can’t be interpreted by machines.It can

be generated due to faulty data collection, data entry errors etc. It can be handled in following ways :

1. **Binning Method:**
   This method works on sorted data in order to smooth it. The whole data is divided into segments of equal size and then various methods are performed to complete the task. Each segmented is handled separately. One can replace all data in a segment by its mean or boundary values can be used to complete the task.
2. **Regression:**
   Here data can be made smooth by fitting it to a regression function.The regression used may be linear (having one independent variable) or multiple (having multiple independent variables).
3. **Clustering:**
   This approach groups the similar data in a cluster. The outliers may be undetected or it will fall outside the clusters.

2. Data Transformation:

This step is taken in order to transform the data in appropriate forms suitable for mining process. This involves following ways:

1. **Normalization:**
   It is done in order to scale the data values in a specified range (-1.0 to 1.0 or 0.0 to 1.0)
2. **Attribute Selection:**
   In this strategy, new attributes are constructed from the given set of attributes to help the mining process.
3. **Discretization:**
   This is done to replace the raw values of numeric attribute by interval levels or conceptual levels.
4. **Concept Hierarchy Generation:**
   Here attributes are converted from level to higher level in hierarchy. For Example-The attribute “city” can be converted to “country”.

3. Data Reduction:

Since data mining is a technique that is used to handle huge amount of data. While working with huge volume of data, analysis became harder in such cases. In order to get rid of this, we uses data reduction technique. It aims to increase the storage efficiency and reduce data storage and analysis costs.

The various steps to data reduction are:

1. **Data Cube Aggregation:**
   Aggregation operation is applied to data for the construction of the data cube.
2. **Attribute Subset Selection:**
   The highly relevant attributes should be used, rest all can be discarded. For performing attribute selection, one can use level of significance and p- value of the attribute.the attribute having p-value greater than significance level can be discarded.

3. **Numerosity Reduction:**
   This enable to store the model of data instead of whole data, for example: Regression Models.
4. **Dimensionality Reduction:**
   This reduce the size of data by encoding mechanisms.It can be lossy or lossless. If after reconstruction from compressed data, original data can be retrieved, such reduction are called lossless reduction else it is called lossy reduction. The two effective methods of dimensionality reduction are:Wavelet transforms and PCA (Principal Componenet Analysis).

4. Data reduction and projection.
   - Finding useful features to represent the data depending on the goal of the task.
   - Using dimensionality reduction or transformation methods to reduce the effective number of variables under consideration or to find invariant representations for the data.
5. Choosing the data mining task.
   - Deciding whether the goal of the KDD process is classification, regression, clustering, etc.
6. Choosing the data mining algorithm(s).
   - Selecting method(s) to be used for searching for patterns in the data.
   - Deciding which models and parameters may be appropriate.
   - Matching a particular data mining method with the overall criteria of the KDD process.
7. Data mining.
   - Searching for patterns of interest in a particular representational form or a set of such representations as classification rules or trees, regression, clustering, and so forth.
8. Interpreting mined patterns.
9. Consolidating discovered knowledge.

The terms *knowledge discovery* and *data mining* are distinct.

NEURAL NETWORKS

Neural networks are an approach to computing that involves developing mathematical structures with the ability to learn. The methods are the result of academic investigations to model nervous system learning. Neural networks have the remarkable ability to derive meaning from complicated or imprecise data and can be used to extract patterns and detect trends that arc too complex to be noticed by either humans or other computer techniques. There are several difference forms of neural network.

- Perceptrons
- Back propagation networks
- Kohonen self-organizing map

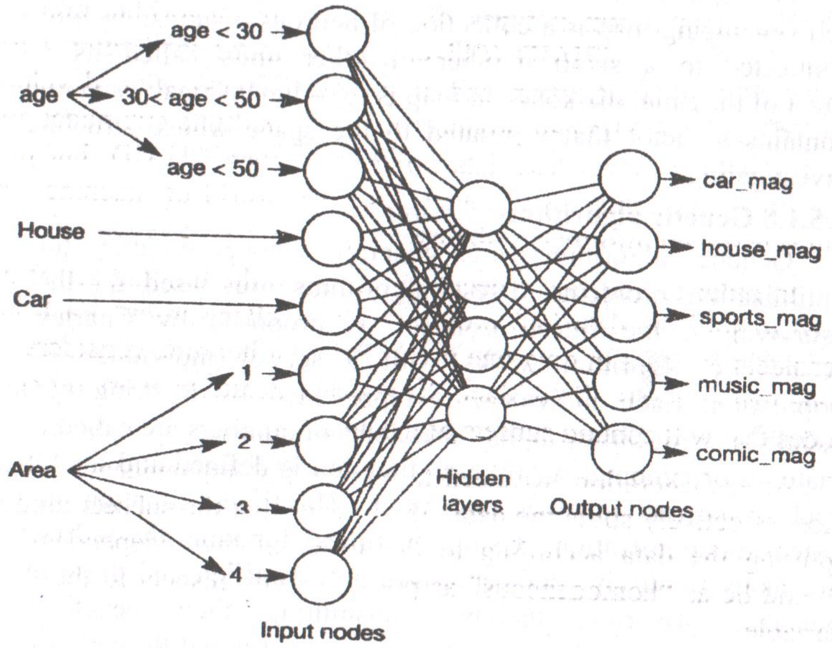


Figure — Neural network architecture for simple data mining

A perceptron consists of simple three-layered network with input units called photo-receptors; intermediate units called associators, and output units called responders. The perceptron could learn simple categories and thus could be used to perform simple classification tasks.

A major improvement was the introduction of hidden layers in the so called hack propagation networks.

A back propagation network not only has input and output nodes, but also a set of intermediate layers with hidden nodes. In its initial stage a back propagation network, expose it to a training set input data. The input nodes are wholly interconnected to the hidden nodes, and the nodes are wholly interconnected to the output nodes. In an untrained network the branches between the nodes have equal weights. During the training stage the network receives examples of input and output pairs corresponding to records in the database, and adapts the weights of the different branches until all the inputs match the appropriate outputs.

In 1981 Tuevo Kohonen demonstrated a completely different version of neural networks that is currently known as Kohonen's self-organizing maps. These neural networks can be seen as the artificial counterparts of maps that exist in several places in the brain, such as visual maps, maps of the spatial possibilities of limbs, and so on. A Kohonen self-organizing map is a collection of neurons of units, each of which is connected to a small number of other units called its neighbors. Most of the time. the kohonen map is two-dimensional; each node or unit contains a factor that is related to the space whose structure we are investigating.

Decision Tress

Decision Trees are a type of Supervised Machine Learning

where the data is continuously split according to a certain parameter. The tree can be explained by two entities, namely decision nodes and leaves. The leaves are the decisions or the final outcomes. And the decision nodes are where the data is split. (Fitness example)

There are two main types of Decision Trees:

Classification trees (Yes/No types)

Regression trees (Continuous data types)

There are many algorithms out there which construct Decision Trees, but one of the best is called as ID3 Algorithm. ID3 Stands for

Iterative Dichotomiser 3.

Entropy

Entropy is the measures of impurity, disorder or uncertainty, messy of data

Entropy controls how a Decision Tree decides to split the data. It actually effects how a Decision Tree draws its boundaries.

Entropy, also called as Shannon Entropy is denoted by H(S) for a finite set S, is the measure of the amount of uncertainty or randomness in data.

$$H(S) = \sum_{x \in X} p(x) \log_2 \frac{1}{p(x)}$$

Intuitively, it tells us about the predictability of a certain event. Example, consider a coin toss whose probability of heads is 0.5 and probability of tails is 0.5. Here the entropy is the highest possible, since there’s no way of determining what the outcome might be. Alternatively, consider a coin which has heads on both the sides, the entropy of such an event can be predicted perfectly since we know beforehand that it’ll always be heads. In other words, this event has **no randomness** hence it’s entropy is zero.

In particular, lower values imply less uncertainty while higher values imply high uncertainty.

Information Gain

Information gain (IG) measures how much “information” a feature gives us about the class. Information gain is the main key that is used by Decision Tree Algorithms to construct a Decision Tree. Decision Trees algorithm will always tries to maximize Information gain. An attribute with highest Information gain will tested/split first.

Information gain is also called as Kullback-Leibler divergence denoted by IG(S,A) for a set S is the effective change in entropy after deciding on a particular attribute

A. It measures the relative change in entropy with respect to the independent variables.

$$IG(S, A) = H(S) - H(S, A)$$

Alternatively,

$$IG(S, A) = H(S) - \sum_{i=0}^{n} P(x) * H(x)$$

where IG(S, A) is the information gain by applying feature A. H(S) is the Entropy of the entire set, while the second term calculates the Entropy after applying the feature A, where P(x) is the probability of event x.

Let’s understand this with the help of an example

Consider a piece of data collected over the course of 14 days where the features are Outlook, Temperature, Humidity, Wind and the outcome variable is whether Cricket was played on the day. Now, our job is to build a predictive model which takes in above 4 parameters and predicts whether Cricket If will be played on the day. We’ll build a decision tree to do that using **ID3 algorithm.**

Day	**Outlook**	**Temperature**	**Humidity**	**Wind**	**Play**
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes

D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

ID3 Algorithm will perform following tasks recursively

1. Create root node for the tree
2. If all examples are positive, return leaf node 'positive'
3. Else if all examples are negative, return leaf node 'negative'
4. Calculate the entropy of current state H(S)
5. For each attribute, calculate the entropy with respect to the attribute 'x' denoted by H(S, x)
6. Select the attribute which has maximum value of IG(S, x)
7. Remove the attribute that offers highest IG from the set of attributes
8. Repeat until we run out of all attributes, or the decision tree has all leaf nodes.

Now we'll go ahead and grow the decision tree. The initial step is to calculate H(S), the Entropy of the current state. In the above example, we can see in total there are 5 No's and 9 Yes's.

Yes	No	Total
9	5	14

Entropy(S) = ∑ – p(I) . log$_2$p(I)

Entropy(Decision) = – p(Yes) . log$_2$p(Yes) – p(No) . log$_2$p(No)

Entropy(Decision) = – (9/14) . log$_2$(9/14) – (5/14) . log$_2$(5/14) = 0.940

Remember that the Entropy is 0 if all members belong to the same class, and 1 when half of them belong to one class and other half belong to other class that is perfect randomness. Here it's 0.94 which means the distribution is fairly random.

Now the next step is to choose the attribute that gives us highest possible Information Gain which we'll choose as the root node.

Let's start with 'Wind'

$$IG(S, Wind) = H(S) - \sum_{i=0}^{n} P(x) * H(x)$$

where ‘x’ are the possible values for an attribute. Here, attribute ‘Wind’ takes two possible values in the sample data, hence x = {Weak, Strong}

We’ll have to calculate:

Amongst all the 14 examples we have **8 places where the wind is weak and 6 where the wind is Strong**.

Wind = Weak	Wind = Strong	Total
8	6	14

$$P(S_{weak}) = \frac{Number\ of\ Weak}{Total}$$

$$= \frac{8}{14}$$

$$P(S_{strong}) = \frac{Number\ of\ Strong}{Total}$$

$$= \frac{6}{14}$$

Now out of the 8 Weak examples, 6 of them were ‘Yes’ for Play cricket and 2 of them were ‘No’ for ‘Play Cricket’. So, we have,

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

$$= 0.811$$

Similarly, out of 6 Strong examples, we have **3 examples where the outcome was ‘Yes’ for Play Cricket and 3 where we had ‘No’ for Play Cricket**.

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

$$= 1.000$$

Remember, here half items belong to one class while other half belong to other. Hence we have perfect randomness.

Now we have all the pieces required to calculate the Information Gain,

$$IG(S, Wind) = H(S) - \sum_{i=0}^{n} P(x) * H(x)$$

$$IG(S, Wind) = H(S) - P(S_{weak}) * H(S_{weak}) - P(S_{strong}) * H(S_{strong})$$

$$= 0.940 - \left(\frac{8}{14}\right)(0.811) - \left(\frac{6}{14}\right)(1.00)$$

$$= 0.048$$

Which tells us the Information Gain by considering ‘Wind’ as the feature and give us information gain of **0.048**. Now we must similarly calculate the Information Gain for all the features.

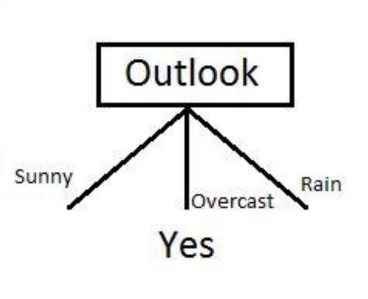
$$IG(S, Outlook) = 0.246$$

$$IG(S, Temperature) = 0.029$$

$$IG(S, Humidity) = 0.151$$

$IG(S, Wind) = 0.048$ (Previous example)

We can clearly see that IG(S, Outlook) has the highest information gain of 0.246, **hence we chose Outlook attribute as the root node**. At this point, the decision tree looks like.



Here we observe that whenever the outlook is Overcast, Play Cricket is always ‘Yes’, it’s no coincidence by any chance, the simple tree resulted because of **the highest information gain is given by the attribute Outlook**.

Now how do we proceed from this point? We can simply apply **recursion**, you might want to look at the algorithm steps described earlier.

Now that we’ve used Outlook, we’ve got three of them remaining Humidity, Temperature, and Wind. And, we had three possible values of Outlook: Sunny, Overcast, Rain. Where the Overcast node already ended up having leaf node ‘Yes’, so we’re left with two subtrees to compute: Sunny and Rain.

Next step would be computing$H(S_{sunny})$.

Table where the value of Outlook is Sunny looks like:

Temperature	Humidity	Wind	Play Cricket
Hot	High	Weak	No
Hot	High	Strong	No
Mild	High	Weak	No
Cool	Normal	Weak	Yes
Mild	Normal	Strong	Yes

$$H(S_{sunny}) = \left(\frac{3}{5}\right)\log_2\left(\frac{3}{5}\right) - \left(\frac{2}{5}\right)\log_2\left(\frac{2}{5}\right) = 0.96$$

In the similar fashion, we compute the following values

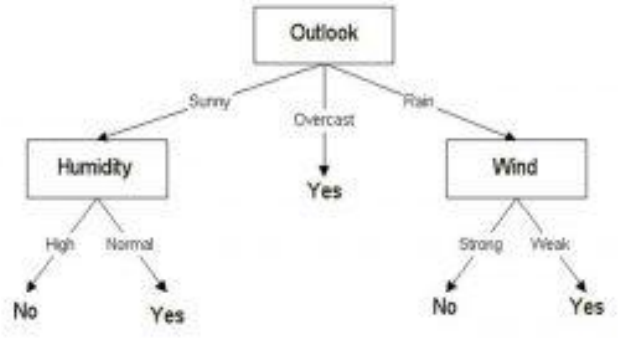
$$IG(S_{sunny}, Humidity) = 0.96$$

$$IG(S_{sunny}, Temperature) = 0.57$$

$$IG(S_{sunny}, Wind) = 0.019$$

As we can see the **highest Information Gain is given by Humidity**. Proceeding in the same way with S_{rain} will give us Wind as the one with highest information gain. The final Decision Tree looks something like this.

The final Decision Tree looks something like this.



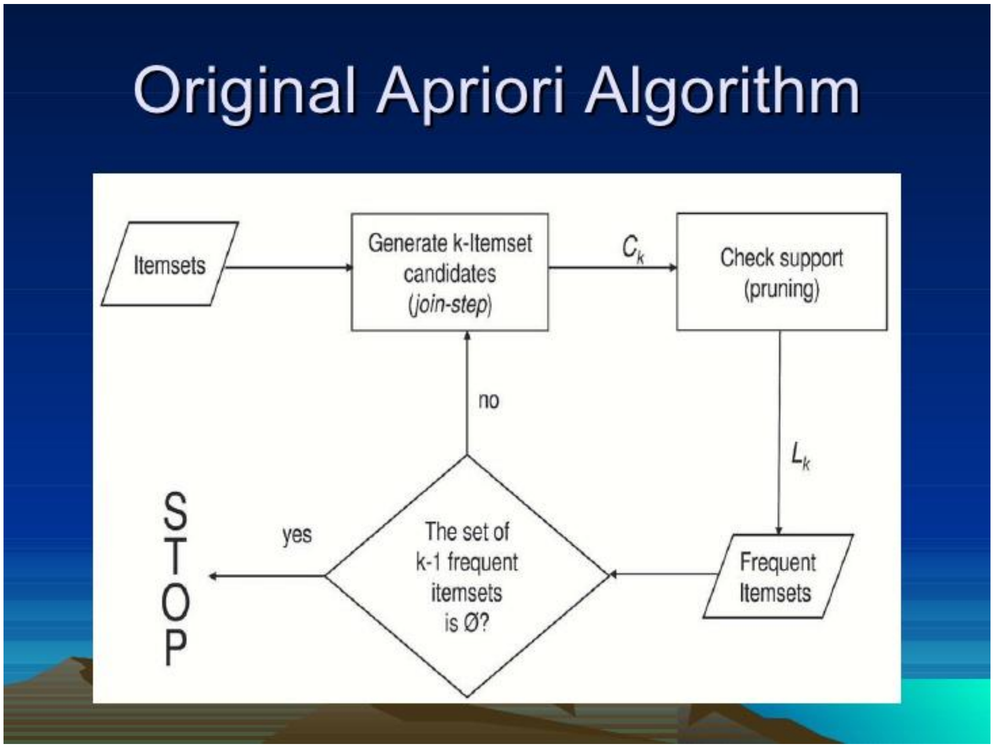
Conclusion:

1. Entropy to measure discriminatory power of an attribute for classification task. It defines the amount of randomness in attribute for classification task. Entropy is minimal means the attribute appears close to one class and have a good discriminatory power for classification
2. Information Gain to rank attribute for filtering at given node in the tree. The ranking is based on high information gain entropy in decreasing order.
3. The recursive ID3 algorithm that creates a decision tree.

Association Rule

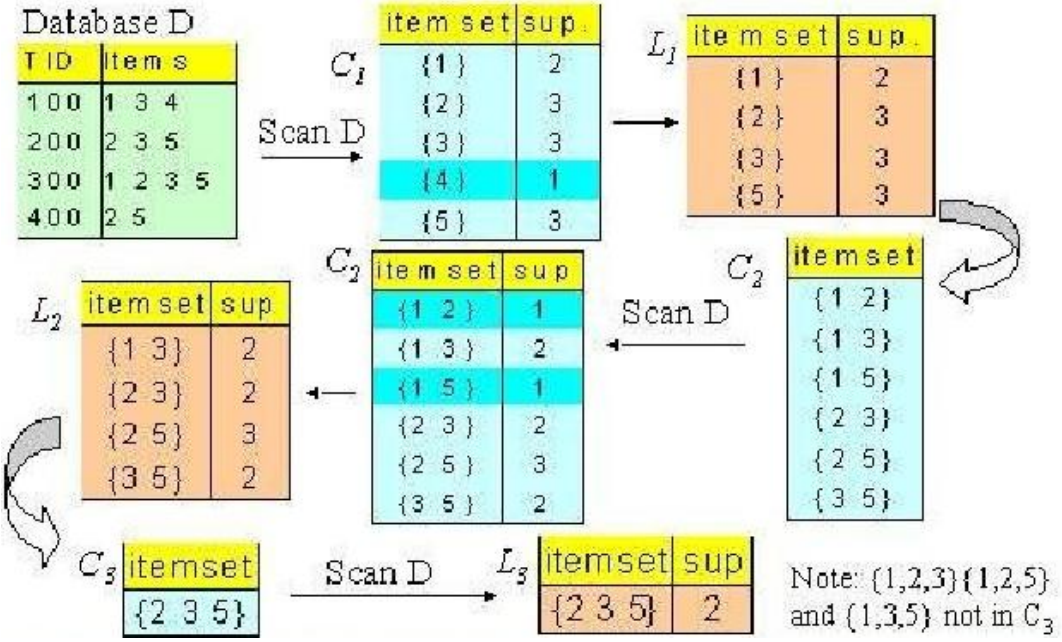
APRIORI ALGORITHM

The Apriori algorithm is an algorithm that attempts to operate on database records, particularly transactional records, or records including certain numbers of fields or items. It is one of a number of algorithms using a "bottom-up approach" to incrementally contrast complex records, and it is useful in today's complex machine learning and artificial intelligence projects.



Original Apriori Algorithm

Itemsets
Generate k-Itemset candidates (*join-step*)
C_k
Check support (pruning)
L_k
Frequent Itemsets
The set of k-1 frequent itemsets is Ø?
no
yes
STOP



Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

itemset
{2 3 5}

Scan D

L_3

itemset	sup
{2 3 5}	2

Note: {1,2,3}{1,2,5} and {1,3,5} not in C_3

GENETIC ALGORITHM

Genetic Algorithm (GA) is a search-based optimization technique based on the principles of **Genetics and Natural Selection**. It is frequently used to find optimal or near-optimal solutions to difficult problems which otherwise would take a lifetime to solve. It is frequently used to solve optimization problems, in research, and in machine learning.

OPTIMIZATION

Optimization is the process of **making something better**. In any process, we have a set of inputs and a set of outputs

Optimization refers to finding the values of inputs in such a way that we get the "best" output values. The definition of "best" varies from problem to problem, but in mathematical terms, it refers to maximizing or minimizing one or more objective functions, by varying the input parameters.

The set of all possible solutions or values which the inputs can take make up the search space. In this search space, lies a point or a set of points which gives the optimal solution. The aim of optimization is to find that point or set of points in the search space.

Genetic Algorithm

Nature has always been a great source of inspiration to all mankind. Genetic Algorithms (GAs) are search based algorithms based on the concepts of natural selection and genetics. GAs are a subset of a much larger branch of computation known as **Evolutionary Computation**.

GAs were developed by John Holland and his students and colleagues at the University of Michigan, most notably David E. Goldberg and has since been tried on various optimization problems with a high degree of success.

In GAs, we have a **pool or a population of possible solutions** to the given problem. These solutions then undergo recombination and mutation (like in natural genetics), producing new children, and the process is repeated over various generations. Each individual (or candidate solution) is assigned a fitness value (based on its objective function value) and the fitter individuals are given a higher chance to mate and yield more "fitter" individuals. This is in line with the Darwinian Theory of "Survival of the Fittest".

In this way we keep "evolving" better individuals or solutions over generations, till we reach a stopping criterion.

Genetic Algorithms are sufficiently randomized in nature, but they perform much better than random local search (in which we just try various random solutions, keeping track of the best so far), as they exploit historical information as well.

Advantages of GAs

GAs have various advantages which have made them immensely popular. These include

- Does not require any derivative information (which may not be available for many real-world problems).
- Is faster and more efficient as compared to the traditional methods.
- Has very good parallel capabilities.
- Optimizes both continuous and discrete functions and also multi-objective problems.
- Provides a list of “good” solutions and not just a single solution.
- Always gets an answer to the problem, which gets better over the time.
- Useful when the search space is very large and there are a large number of parameters involved.

Limitations of GAs

Like any technique, GAs also suffer from a few limitations. These include –

- GAs are not suited for all problems, especially problems which are simple and for which derivative information is available.
- Fitness value is calculated repeatedly which might be computationally expensive for some problems.
- Being stochastic, there are no guarantees on the optimality or the quality of the solution.
- If not implemented properly, the GA may not converge to the optimal solution.

Before beginning a discussion on Genetic Algorithms, it is essential to be familiar with some basic terminology which will be used throughout this tutorial.

• Population – It is a subset of all the possible solutions to the given problem. The population for a GA is analogous to the population for human beings except that instead of human beings, we have Candidate Solutions representing human beings.

• Chromosomes – A chromosome is one such solution to the given problem.

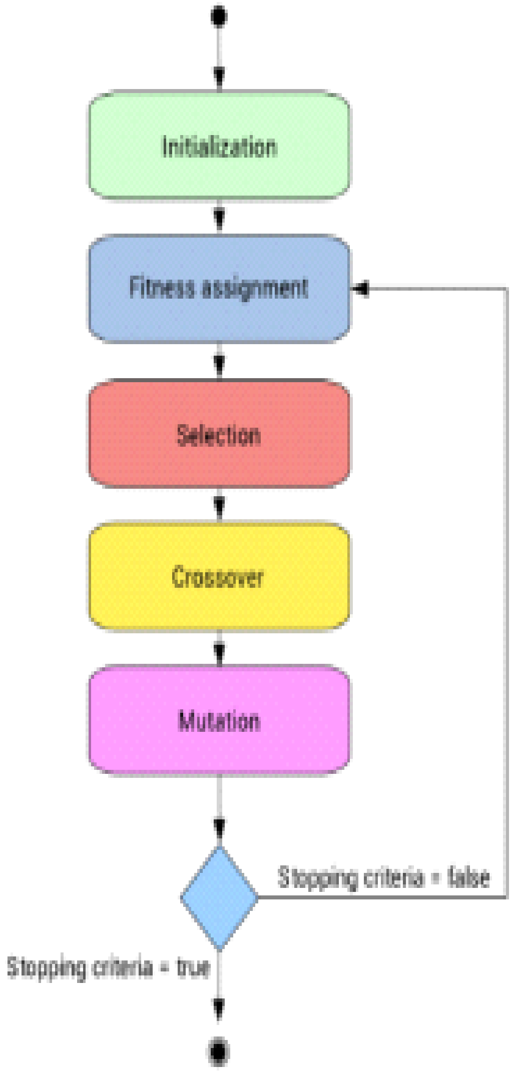
• Gene – A gene is one element position of a chromosome.

• Allele – It is the value a gene takes for a particular chromosome.

Population is a subset of solutions in the current generation. It can also be defined as a set of chromosomes. There are several things to be kept in mind when dealing with GA population –

- The diversity of the population should be maintained otherwise it might lead to premature convergence.
- The population size should not be kept very large as it can cause a GA to slow down, while a smaller population might not be enough for a good mating pool. Therefore, an optimal population size needs to be decided by trial and error.

The population is usually defined as a two dimensional array of – **size population, size x, chromosome size**.



Population Initialization

There are two primary methods to initialize a population in a GA. They are –

- **Random Initialization** – Populate the initial population with completely random solutions.
- **Heuristic initialization** – Populate the initial population using a known heuristic for the problem.

It has been observed that the entire population should not be initialized using a heuristic, as it can result in the population having similar solutions and very little diversity. It has been experimentally observed that the random solutions are the ones to drive the population to optimality. Therefore, with heuristic initialization, we just seed the population with a couple of good solutions, filling up the rest with random solutions rather than filling the entire population with heuristic based solutions.

It has also been observed that heuristic initialization in some cases, only effects the initial fitness of the population, but in the end, it is the diversity of the solutions which lead to optimality.

Fitness

The fitness function simply defined is a function which takes a **candidate solution to the problem as input and produces as output** how "fit" our how "good" the solution is with respect to the problem in consideration.

Calculation of fitness value is done repeatedly in a GA and therefore it should be sufficiently fast. A slow computation of the fitness value can adversely affect a GA and make it exceptionally slow.

In most cases the fitness function and the objective function are the same as the objective is to either maximize or minimize the given objective function. However, for more complex problems with multiple objectives and constraints, an **Algorithm Designer** might choose to have a different fitness function.

A fitness function should possess the following characteristics –

- The fitness function should be sufficiently fast to compute.
- It must quantitatively measure how fit a given solution is or how fit individuals can be produced from the given solution.

In some cases, calculating the fitness function directly might not be possible due to the inherent complexities of the problem at hand. In such cases, we do fitness approximation to suit our needs.

Selection

Parent Selection is the process of selecting parents which mate and recombine to create off-springs for the next generation. Parent selection is very crucial to the convergence rate of the GA as good parents drive individuals to a better and fitter solutions.

However, care should be taken to prevent one extremely fit solution from taking over the entire population in a few generations, as this leads to the solutions being close to one another in the solution space thereby leading to a loss of diversity. **Maintaining good diversity** in the population is extremely crucial for the success of a GA. This taking up of the entire population by one extremely fit solution is known as **premature convergence** and is an undesirable condition in a GA.

Crossover

The crossover operator is analogous to reproduction and biological crossover. In this more than one parent is selected and one or more off-springs are produced using the genetic material of the parents. Crossover is usually applied in a GA with a high probability – $\boldsymbol{p_c}$.

Mutation

In simple terms, mutation may be defined as a small random tweak in the chromosome, to get a new solution. It is used to maintain and introduce diversity in the genetic population and is usually applied with a low probability – $\boldsymbol{p_m}$. If the probability is very high, the GA gets reduced to a random search.

Mutation is the part of the GA which is related to the “exploration” of the search space. It has been observed that mutation is essential to the convergence of the GA while crossover is not.

The termination condition of a Genetic Algorithm is important in determining when a GA run will end. It has been observed that initially, the GA progresses very fast with better solutions coming in every few iterations, but this tends to saturate in the later stages where the improvements are very small. We usually want a termination condition such that our solution is close to the optimal, at the end of the run.

Usually, we keep one of the following termination conditions −

- When there has been no improvement in the population for X iterations.
- When we reach an absolute number of generations.
- When the objective function value has reached a certain pre-defined value.

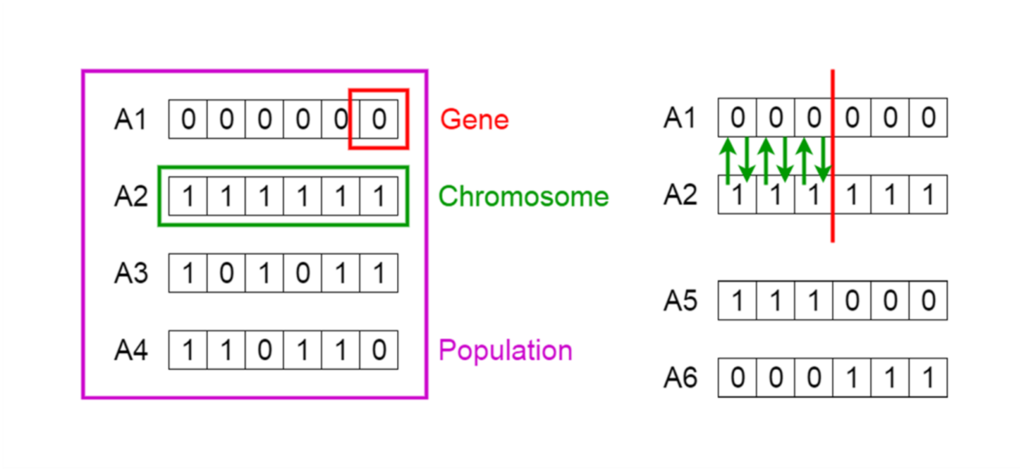
Termination

In a genetic algorithm we keep a counter which keeps track of the generations for which there has been no improvement in the population. Initially, we set this counter to zero. Each time we don’t generate off-springs which are better than the individuals in the population, we increment the counter.

However, if the fitness any of the off-springs is better, then we reset the counter to zero. The algorithm terminates when the counter reaches a predetermined value.

Like other parameters of a GA, the termination condition is also highly problem specific and the GA designer should try out various options to see what suits his particular problem the best.

A **genetic algorithm** is a search heuristic that is inspired by Charles Darwin’s theory of natural evolution. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation.



Notion of Natural Selection

The process of natural selection starts with the selection of fittest individuals from a population. They produce offspring which inherit the characteristics of the parents and will be added to the next generation. If parents have better fitness, their offspring will be better than parents and have a better chance at surviving. This process keeps on iterating and at the end, a generation with the fittest individuals will be found.

This notion can be applied for a search problem. We consider a set of solutions for a problem and select the set of best ones out of them.

Five phases are considered in a genetic algorithm.

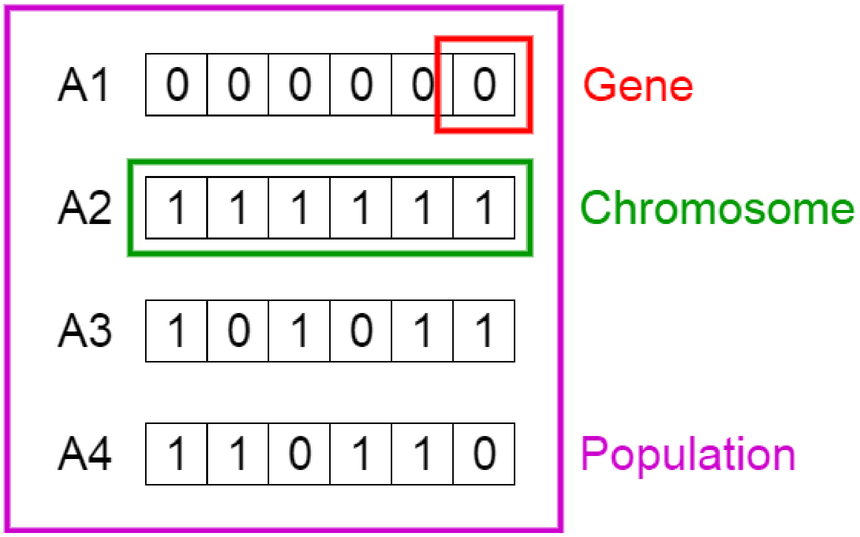
1. Initial population
2. Fitness function
3. Selection
4. Crossover
5. Mutation

Initial Population

The process begins with a set of individuals which is called a **Population**. Each individual is a solution to the problem you want to solve.

An individual is characterized by a set of parameters (variables) known as **Genes**. Genes are joined into a string to form a **Chromosome** (solution).

In a genetic algorithm, the set of genes of an individual is represented using a string, in terms of an alphabet. Usually, binary values are used (string of 1s and 0s). We say that we encode the genes in a chromosome.



Population, Chromosomes and Genes

Fitness Function

The **fitness function** determines how fit an individual is (the ability of an individual to compete with other individuals). It gives a **fitness score** to each individual. The probability that an individual will be selected for reproduction is based on its fitness score.

Selection

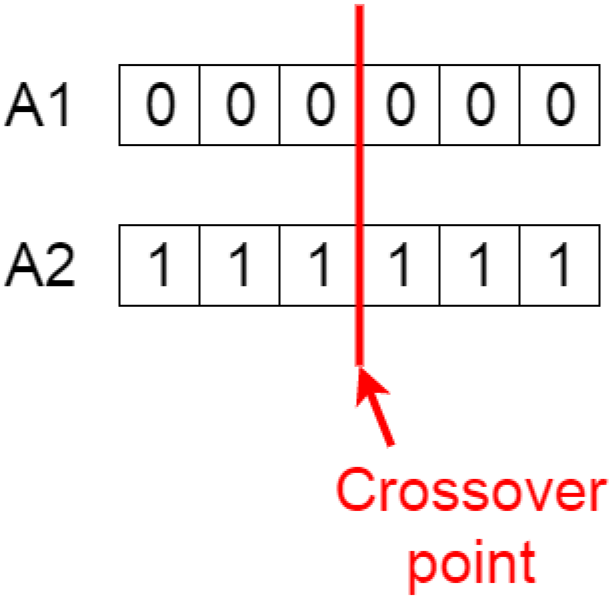
The idea of **selection** phase is to select the fittest individuals and let them pass their genes to the next generation.

Two pairs of individuals (**parents**) are selected based on their fitness scores. Individuals with high fitness have more chance to be selected for reproduction.

Crossover

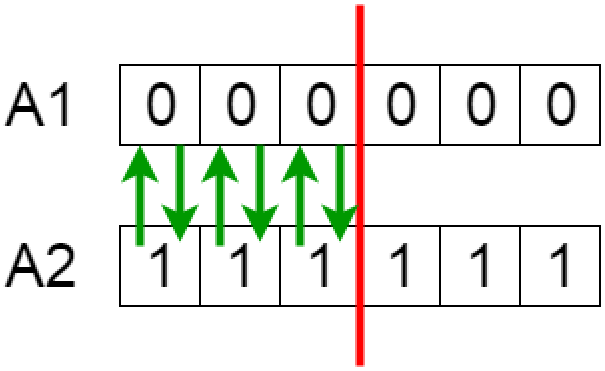
Crossover is the most significant phase in a genetic algorithm. For each pair of parents to be mated, a **crossover point** is chosen at random from within the genes.

For example, consider the crossover point to be 3 as shown below.



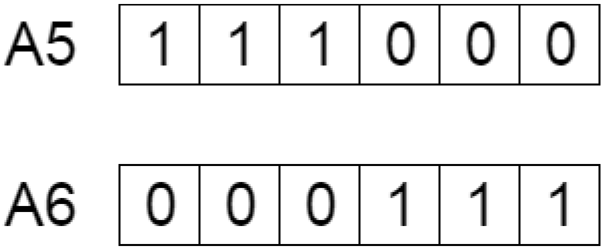
Crossover point

Offspring are created by exchanging the genes of parents among themselves until the crossover point is reached.



Exchanging genes among parents

The new offspring are added to the population.

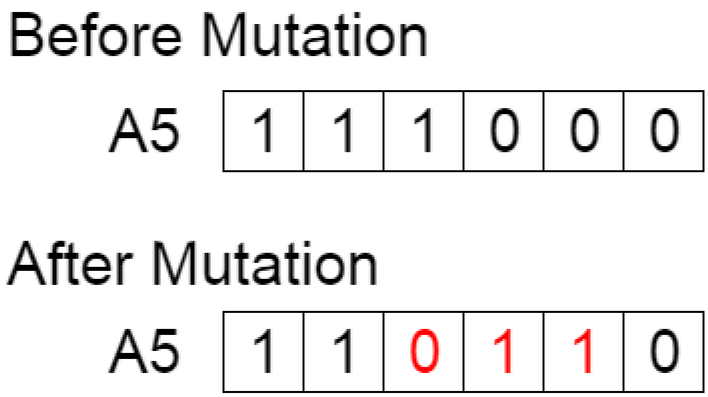


A5	1	1	1	0	0	0
A6	0	0	0	1	1	1

New offspring

Mutation

In certain new offspring formed, some of their genes can be subjected to a **mutation** with a low random probability. This implies that some of the bits in the bit string can be flipped.



Before Mutation

A5	1	1	1	0	0	0

After Mutation

A5	1	1	0	1	1	0

Mutation: Before and After

Mutation occurs to maintain diversity within the population and prevent premature convergence.

Termination

The algorithm terminates if the population has converged (does not produce offspring which are significantly different from the previous generation). Then it is said that the genetic algorithm has provided a set of solutions to our problem.

UNIT-IV

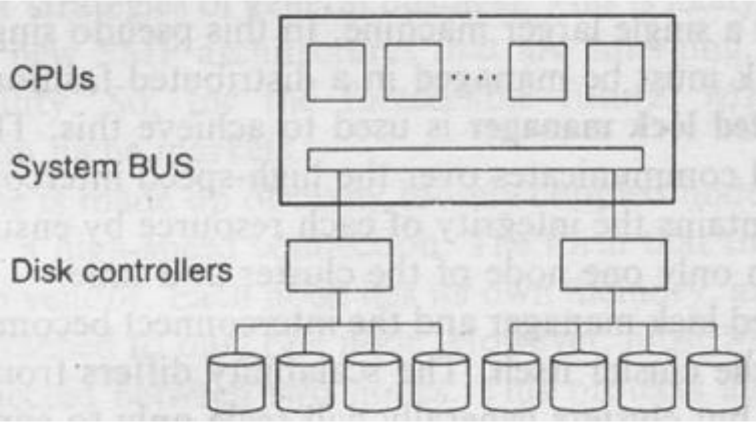
Hardware and Operational design of Data Warehouse

Server Hardware

- Two main hardware architectures
    - Symmetric Multi-Processing (SMP)
    - Massively Parallel Processing (MPP)
- An SMP machine is a set of tightly coupled CPUs that share memory and disk
- An MPP machine is a set of loosely coupled CPUs, each of which has its own memory and disk

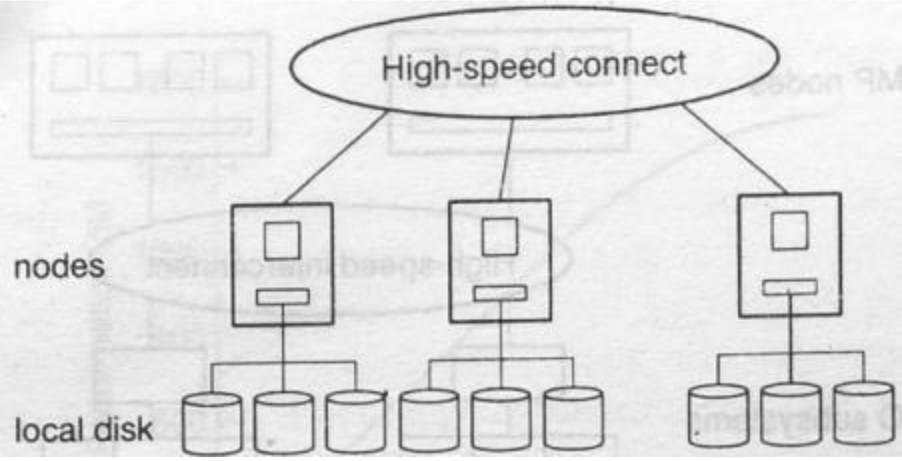
Symmetric Multi-Processing (SMP)

- An SMP machine is a set of CPU s that share memory and disk.
- This is sometimes called a **shared-everything** environment
- the CPUs in an SMP machine are all equal
- a process can run on any CPU in the machine, run on different CPUs at different time



Massively Parallel Processing (MPP)

- made up of many loosely coupled nodes linked together ·by a high-speed connection
- Each node has its own memory, and the disks are not shared
- most MPP systems allow a disk to be dual connected between two nodes
- protects against an individual node failure causing disks to be unavailable



Service Level agreement

- A service level agreement (SLA) is essential to the design process of the data warehouse
- essential to the design of the backup strategy
- design decisions as partitioning of the fact data
- Some more important topics that the SLA must cover are
  - user online access - hours of work
  - user batch access
  - user expected response times

SLA requirements by the data warehouse

- SLAs with the organization's operations, systems and network groups like
  - priority of network access
  - network availability guarantees
  - network bandwidth guarantees
  - priority of access to the backup hardware
- The SLA(s) must cover all the dependencies that the data warehouse has on the outside world

Categories of SLA

□ **User requirements**: elements that directly affect the users, such as hours of access and response times

□ **System requirements**: needs imposed on the system by the business, such as system availability

User Requirements

Detailed information is required on the following:

□ user online access -·hours of work

□ user batch access

□ user expected response times

  o average response times

  o maximum acceptable response times

Online User Access Requirements

□ business working hours, online working hours

□ The overall online requirement will be a combination of all the user group requirements

□ design the data warehouse for the definite requirements you have, but take these assumptions about the future into consideration

□ user batch requirements

□ queries running overnight - ability to submit large jobs to run outside the online window

□ expected response times

□ ascertain both the average response times that are required and the worst response they will accept

□ Ask users what they need, not what they want

System Requirements

☐ key system requirement is the maximum acceptable downtime for the system

☐ The SLA needs to stipulate fully any measures related to downtime

Availability

☐ some measure of the required availability of the server is needed

☐ measured as a required percentage of uptime

- o D = 100 –A where D =acceptable downtime, and A is the percentage required availability

☐ This .can be further broken down into

- o acceptable online downtime (Dn = 100-An),
- o acceptable offline downtime (Df = 100 - Af)
- o where An is the percentage of N for which the system is required to be available;
- o Af is the percentage of (24-N) hours for which the system is required to be available;

☐ In a data warehouse system this would be Dn > Df

☐ These availability figures will drive the resilience requirements, and may affect the architecture.

Example:

☐ if high availability is required and an SMP architecture is being used, you may want to consider a cluster solution, even if the data warehouse does not require the extra node for CPU bandwidth.

☐ The extra node can be used for automatic failover, thereby giving greater availability

Permissible Planned Downtime

☐ maximum amount of planned time in any period that the data warehouse can be down

SLA requirements by the data warehouse

☐ SLAs with the organization's operations, systems and network groups like

- o priority of network access
- o network availability guarantees
- o network bandwidth guarantees
- o priority of access to the backup hardware

☐ The SLA(s) must cover all the dependencies that the data warehouse has on the outside world

SECURITY

The objective of a data warehouse is to make large amounts of data easily accessible to the users, hence allowing the users to extract information about the business as a whole. But we know that there could be some security restrictions applied on the data that can be an obstacle for accessing the information. If the analyst has a restricted view of data, then it is impossible to capture a complete picture of the trends within the business.

The data from each analyst can be summarized and passed on to management where the different summaries can be aggregated. As the aggregations of summaries cannot be the same as that of the aggregation as a whole, it is possible to miss some information trends in the data unless someone is analyzing the data as a whole.

Security Requirements

Adding security features affect the performance of the data warehouse, therefore it is important to determine the security requirements as early as possible. It is difficult to add security features after the data warehouse has gone live.

During the design phase of the data warehouse, we should keep in mind what data sources may be added later and what would be the impact of adding those data sources. We should consider the following possibilities during the design phase.

- Whether the new data sources will require new security and/or audit restrictions to be implemented?
- Whether the new users added who have restricted access to data that is already generally available?

This situation arises when the future users and the data sources are not well known. In such a situation, we need to use the knowledge of business and the objective of data warehouse to know likely requirements.

The following activities get affected by security measures –

- User access
- Data load
- Data movement
- Query generation

User Access

We need to first classify the data and then classify the users on the basis of the data they can access. In other words, the users are classified according to the data they can access.

Data Classification

The following two approaches can be used to classify the data –

- Data can be classified according to its sensitivity. Highly-sensitive data is classified as highly restricted and less-sensitive data is classified as less restrictive.

- Data can also be classified according to the job function. This restriction allows only specific users to view particular data. Here we restrict the users to view only that part of the data in which they are interested and are responsible for.

There are some issues in the second approach. To understand, let's have an example. Suppose you are building the data warehouse for a bank. Consider that the data being stored in the data warehouse is the transaction data for all the accounts. The question here is, who is allowed to see the transaction data. The solution lies in classifying the data according to the function.

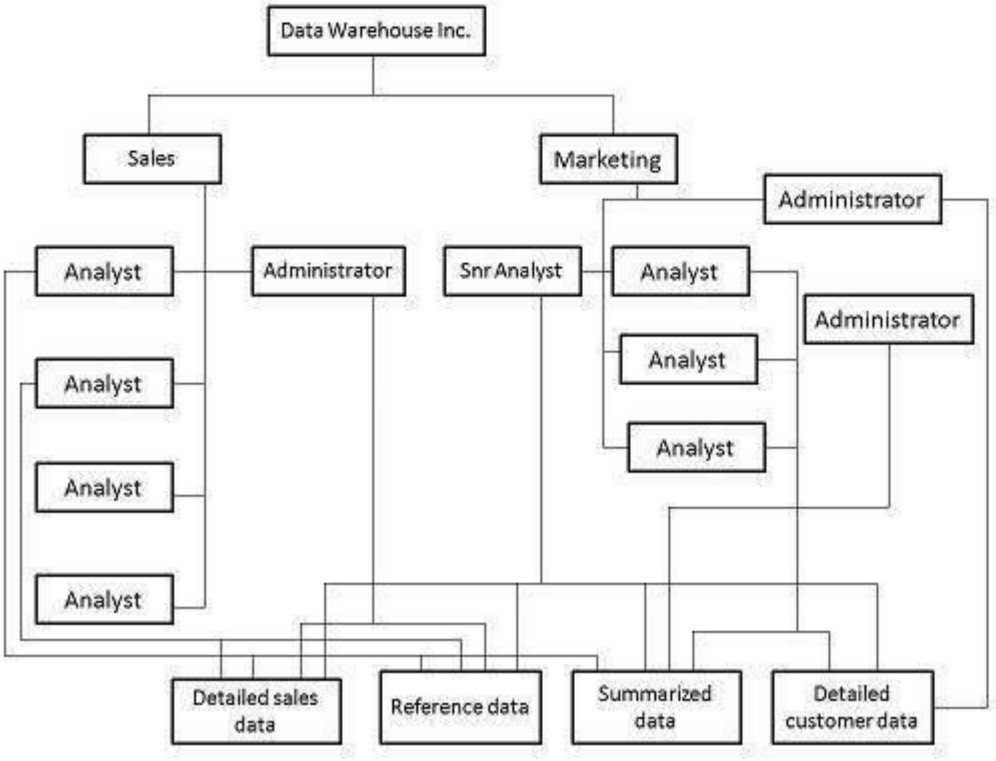
User classification

The following approaches can be used to classify the users −

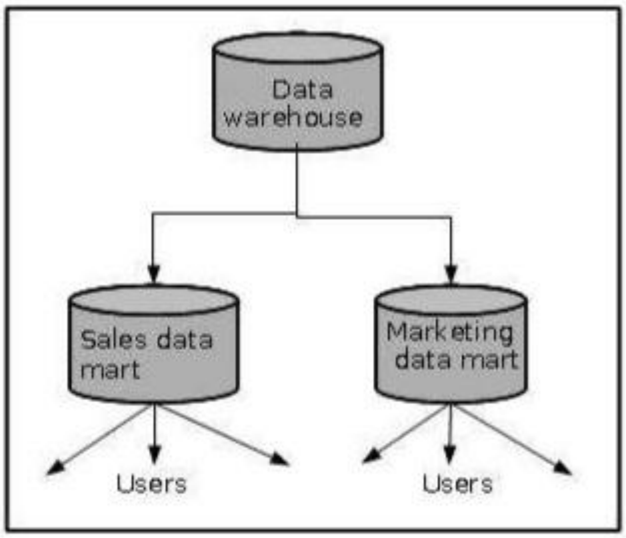
- Users can be classified as per the hierarchy of users in an organization, i.e., users can be classified by departments, sections, groups, and so on.
- Users can also be classified according to their role, with people grouped across departments based on their role.

Classification on basis of Department

Let's have an example of a data warehouse where the users are from sales and marketing department. We can have security by top-to-down company view, with access centered on the different departments. But there could be some restrictions on users at different levels. This structure is shown in the following diagram.

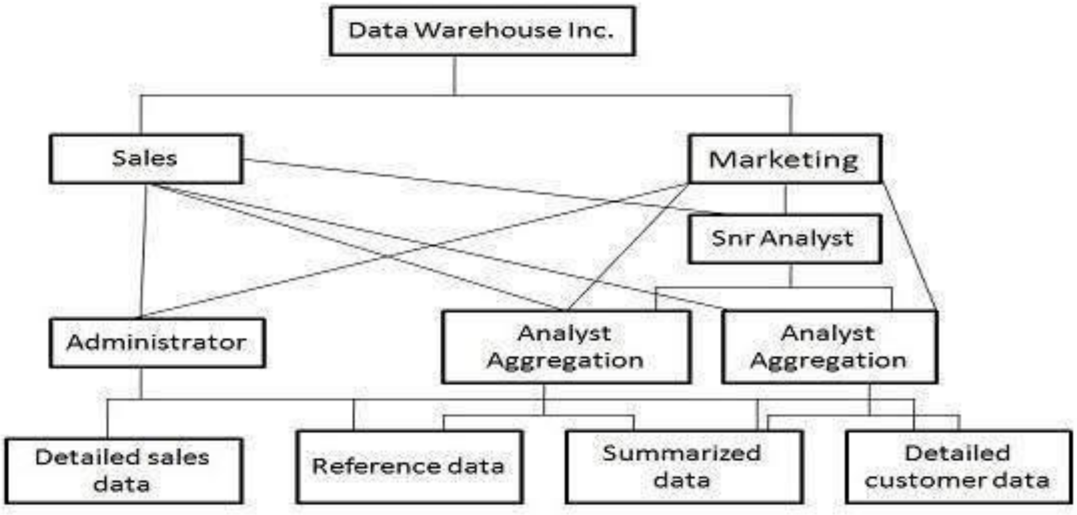


But if each department accesses different data, then we should design the security access for each department separately. This can be achieved by departmental data marts. Since these data marts are separated from the data warehouse, we can enforce separate security restrictions on each data mart. This approach is shown in the following figure.



Classification Based on Role

If the data is generally available to all the departments, then it is useful to follow the role access hierarchy. In other words, if the data is generally accessed by all the departments, then apply security restrictions as per the role of the user. The role access hierarchy is shown in the following figure.



Audit Requirements

Auditing is a subset of security, a costly activity. Auditing can cause heavy overheads on the system. To complete an audit in time, we require more hardware and therefore, it is recommended that wherever possible, auditing should be switched off. Audit requirements can be categorized as follows −

- Connections
- Disconnections
- Data access
- Data change

Note − For each of the above-mentioned categories, it is necessary to audit success, failure, or both. From the perspective of security reasons, the auditing of failures are very important. Auditing of failure is important because they can highlight unauthorized or fraudulent access.

Network Requirements

Network security is as important as other securities. We cannot ignore the network security requirement. We need to consider the following issues −

- Is it necessary to encrypt data before transferring it to the data warehouse?
- Are there restrictions on which network routes the data can take?

These restrictions need to be considered carefully. Following are the points to remember −

- The process of encryption and decryption will increase overheads. It would require more processing power and processing time.
- The cost of encryption can be high if the system is already a loaded system because the encryption is borne by the source system.

Data Movement

There exist potential security implications while moving the data. Suppose we need to transfer some restricted data as a flat file to be loaded. When the data is loaded into the data warehouse, the following questions are raised −

- Where is the flat file stored?
- Who has access to that disk space?

If we talk about the backup of these flat files, the following questions are raised −

- Do you backup encrypted or decrypted versions?
- Do these backups need to be made to special tapes that are stored separately?
- Who has access to these tapes?

Some other forms of data movement like query result sets also need to be considered. The questions raised while creating the temporary table are as follows −

- Where is that temporary table to be held?
- How do you make such table visible?

We should avoid the accidental flouting of security restrictions. If a user with access to the restricted data can generate accessible temporary tables, data can be visible to non-authorized users. We can overcome this problem by having a separate temporary area for users with access to restricted data.

Documentation

The audit and security requirements need to be properly documented. This will be treated as a part of justification. This document can contain all the information gathered from −

- Data classification
- User classification
- Network requirements
- Data movement and storage requirements
- All auditable actions

Impact of Security on Design

Security affects the application code and the development timescales. Security affects the following area −

- Application development
- Database design
- Testing

Application Development

Security affects the overall application development and it also affects the design of the important components of the data warehouse such as load manager, warehouse manager, and query manager. The load manager may require checking code to filter record and place them in different locations. More transformation rules may also be required to hide certain data. Also there may be requirements of extra metadata to handle any extra objects.

To create and maintain extra views, the warehouse manager may require extra codes to enforce security. Extra checks may have to be coded into the data warehouse to prevent it from being fooled into moving data into a location where it should not be available. The query manager requires the changes to handle any access restrictions. The query manager will need to be aware of all extra views and aggregations.

Database design

The database layout is also affected because when security measures are implemented, there is an increase in the number of views and tables. Adding security increases the size of the database and hence increases the complexity of the database design and management. It will also add complexity to the backup management and recovery plan.

Testing

Testing the data warehouse is a complex and lengthy process. Adding security to the data warehouse also affects the testing time complexity. It affects the testing in the following two ways −

- It will increase the time required for integration and system testing.
- There is added functionality to be tested which will increase the size of the testing suite.

BACKUP & RECOVERY

A data warehouse is a complex system and it contains a huge volume of data. Therefore it is important to back up all the data so that it becomes available for recovery in future as per requirement. In this chapter, we will discuss the issues in designing the backup strategy.

Backup Terminologies

Before proceeding further, you should know some of the backup terminologies discussed below.

- **Complete backup** − It backs up the entire database at the same time. This backup includes all the database files, control files, and journal files.
- **Partial backup** − As the name suggests, it does not create a complete backup of the database. Partial backup is very useful in large databases because they allow a strategy whereby various parts of the database are backed up in a round-robin fashion on a day-to-day basis, so that the whole database is backed up effectively once a week.
- **Cold backup** − Cold backup is taken while the database is completely shut down. In multi-instance environment, all the instances should be shut down.
- **Hot backup** − Hot backup is taken when the database engine is up and running. The requirements of hot backup varies from RDBMS to RDBMS.
- **Online backup** − It is quite similar to hot backup.

Hardware Backup

It is important to decide which hardware to use for the backup. The speed of processing the backup and restore depends on the hardware being used, how the hardware is connected, bandwidth of the network, backup software, and the speed of server's I/O system. Here we will discuss some of the hardware choices that are available and their pros and cons. These choices are as follows −

- Tape Technology
- Disk Backups

Tape Technology

The tape choice can be categorized as follows −

- Tape media
- Standalone tape drives
- Tape stackers
- Tape silos

Tape Media

There exists several varieties of tape media. Some tape media standards are listed in the table below −

Tape Media	Capacity	I/O rates
DLT	40 GB	3 MB/s
3490e	1.6 GB	3 MB/s
8 mm	14 GB	1 MB/s

Other factors that need to be considered are as follows −

- Reliability of the tape medium
- Cost of tape medium per unit
- Scalability
- Cost of upgrades to tape system
- Cost of tape medium per unit
- Shelf life of tape medium

Standalone Tape Drives

The tape drives can be connected in the following ways −

- Direct to the server
- As network available devices
- Remotely to other machine

There could be issues in connecting the tape drives to a data warehouse.

- Consider the server is a 48node MPP machine. We do not know the node to connect the tape drive and we do not know how to spread them over the server nodes to get the optimal performance with least disruption of the server and least internal I/O latency.
- Connecting the tape drive as a network available device requires the network to be up to the job of the huge data transfer rates. Make sure that sufficient bandwidth is available during the time you require it.
- Connecting the tape drives remotely also require high bandwidth.

Tape Stackers

The method of loading multiple tapes into a single tape drive is known as tape stackers. The stacker dismounts the current tape when it has finished with it and loads the next tape, hence only one tape is available at a time to be accessed. The price and the capabilities may vary, but the common ability is that they can perform unattended backups.

Tape Silos

Tape silos provide large store capacities. Tape silos can store and manage thousands of tapes. They can integrate multiple tape drives. They have the software and hardware to label and store the tapes they store. It is very common for the silo to be connected remotely over a network or a dedicated link. We should ensure that the bandwidth of the connection is up to the job.

Disk Backups

Methods of disk backups are −

- Disk-to-disk backups
- Mirror breaking

These methods are used in the OLTP system. These methods minimize the database downtime and maximize the availability.

Disk-to-Disk Backups

Here backup is taken on the disk rather on the tape. Disk-to-disk backups are done for the following reasons −

- Speed of initial backups
- Speed of restore

Backing up the data from disk to disk is much faster than to the tape. However it is the intermediate step of backup. Later the data is backed up on the tape. The other advantage of disk-to-disk backups is that it gives you an online copy of the latest backup.

Mirror Breaking

The idea is to have disks mirrored for resilience during the working day. When backup is required, one of the mirror sets can be broken out. This technique is a variant of disk-to-disk backups.

Note − The database may need to be shutdown to guarantee consistency of the backup.

Optical Jukeboxes

Optical jukeboxes allow the data to be stored near line. This technique allows a large number of optical disks to be managed in the same way as a tape stacker or a tape silo. The drawback of this technique is that it has slow write speed than disks. But the optical media provides long-life and reliability that makes them a good choice of medium for archiving.

Software Backups

There are software tools available that help in the backup process. These software tools come as a package. These tools not only take backup, they can effectively manage and control the backup strategies. There are many software packages available in the market. Some of them are listed in the following table −

Package Name	**Vendor**
Networker	Legato
ADSM	IBM
Epoch	Epoch Systems
Omniback II	HP
Alexandria	Sequent

Criteria for Choosing Software Packages

The criteria for choosing the best software package are listed below −

- How scalable is the product as tape drives are added?
- Does the package have client-server option, or must it run on the database server itself?
- Will it work in cluster and MPP environments?
- What degree of parallelism is required?
- What platforms are supported by the package?
- Does the package support easy access to information about tape contents?
- Is the package database aware?

What tape drive and tape media are supported by the package?

UNIT-V

TUNING

A data warehouse keeps evolving and it is unpredictable what query the user is going to post in the future. Therefore it becomes more difficult to tune a data warehouse system. In this chapter, we will discuss how to tune the different aspects of a data warehouse such as performance, data load, queries, etc.

Difficulties in Data Warehouse Tuning

Tuning a data warehouse is a difficult procedure due to following reasons −

- Data warehouse is dynamic; it never remains constant.
- It is very difficult to predict what query the user is going to post in the future.
- Business requirements change with time.
- Users and their profiles keep changing.
- The user can switch from one group to another.
- The data load on the warehouse also changes with time.

Note − It is very important to have a complete knowledge of data warehouse.

Performance Assessment

Here is a list of objective measures of performance −

- Average query response time
- Scan rates
- Time used per day query
- Memory usage per process
- I/O throughput rates

Following are the points to remember.

- It is necessary to specify the measures in service level agreement (SLA).

- It is of no use trying to tune response time, if they are already better than those required.
- It is essential to have realistic expectations while making performance assessment.
- It is also essential that the users have feasible expectations.
- To hide the complexity of the system from the user, aggregations and views should be used.
- It is also possible that the user can write a query you had not tuned for.

Data Load Tuning

Data load is a critical part of overnight processing. Nothing else can run until data load is complete. This is the entry point into the system.

Note – If there is a delay in transferring the data, or in arrival of data then the entire system is affected badly. Therefore it is very important to tune the data load first.

There are various approaches of tuning data load that are discussed below –

- The very common approach is to insert data using the **SQL Layer**. In this approach, normal checks and constraints need to be performed. When the data is inserted into the table, the code will run to check for enough space to insert the data. If sufficient space is not available, then more space may have to be allocated to these tables. These checks take time to perform and are costly to CPU.
- The second approach is to bypass all these checks and constraints and place the data directly into the preformatted blocks. These blocks are later written to the database. It is faster than the first approach, but it can work only with whole blocks of data. This can lead to some space wastage.
- The third approach is that while loading the data into the table that already contains the table, we can maintain indexes.
- The fourth approach says that to load the data in tables that already contain data, **drop the indexes & recreate them** when the data load is

complete. The choice between the third and the fourth approach depends on how much data is already loaded and how many indexes need to be rebuilt.

Integrity Checks

Integrity checking highly affects the performance of the load. Following are the points to remember −

- Integrity checks need to be limited because they require heavy processing power.
- Integrity checks should be applied on the source system to avoid performance degrade of data load.

Tuning Queries

We have two kinds of queries in data warehouse −

- Fixed queries
- Ad hoc queries

Fixed Queries

Fixed queries are well defined. Following are the examples of fixed queries −

- regular reports
- Canned queries
- Common aggregations

Tuning the fixed queries in a data warehouse is same as in a relational database system. The only difference is that the amount of data to be queried may be different. It is good to store the most successful execution plan while testing fixed queries. Storing these executing plan will allow us to spot changing data size and data skew, as it will cause the execution plan to change.

Note − We cannot do more on fact table but while dealing with dimension tables or the aggregations, the usual collection of SQL tweaking, storage mechanism, and access methods can be used to tune these queries.

Ad hoc Queries

To understand ad hoc queries, it is important to know the ad hoc users of the data warehouse. For each user or group of users, you need to know the following −

- The number of users in the group
- Whether they use ad hoc queries at regular intervals of time
- Whether they use ad hoc queries frequently
- Whether they use ad hoc queries occasionally at unknown intervals.
- The maximum size of query they tend to run
- The average size of query they tend to run
- Whether they require drill-down access to the base data
- The elapsed login time per day
- The peak time of daily usage
- The number of queries they run per peak hour

Points to Note

- It is important to track the user's profiles and identify the queries that are run on a regular basis.
- It is also important that the tuning performed does not affect the performance.
- Identify similar and ad hoc queries that are frequently run.
- If these queries are identified, then the database will change and new indexes can be added for those queries.
- If these queries are identified, then new aggregations can be created specifically for those queries that would result in their efficient execution.

TESTING

Testing is very important for data warehouse systems to make them work correctly and efficiently. There are three basic levels of testing performed on a data warehouse −

- Unit testing
- Integration testing
- System testing

Unit Testing

- In unit testing, each component is separately tested.
- Each module, i.e., procedure, program, SQL Script, Unix shell is tested.
- This test is performed by the developer.

Integration Testing

- In integration testing, the various modules of the application are brought together and then tested against the number of inputs.
- It is performed to test whether the various components do well after integration.

System Testing

- In system testing, the whole data warehouse application is tested together.
- The purpose of system testing is to check whether the entire system works correctly together or not.
- System testing is performed by the testing team.
- Since the size of the whole data warehouse is very large, it is usually possible to perform minimal system testing before the test plan can be enacted.

Test Schedule

First of all, the test schedule is created in the process of developing the test plan. In this schedule, we predict the estimated time required for the testing of the entire data warehouse system.

There are different methodologies available to create a test schedule, but none of them are perfect because the data warehouse is very complex and large. Also the data warehouse system is evolving in nature. One may face the following issues while creating a test schedule −

- A simple problem may have a large size of query that can take a day or more to complete, i.e., the query does not complete in a desired time scale.
- There may be hardware failures such as losing a disk or human errors such as accidentally deleting a table or overwriting a large table.

Note − Due to the above-mentioned difficulties, it is recommended to always double the amount of time you would normally allow for testing.

Testing Backup Recovery

Testing the backup recovery strategy is extremely important. Here is the list of scenarios for which this testing is needed −

- Media failure
- Loss or damage of table space or data file
- Loss or damage of redo log file
- Loss or damage of control file
- Instance failure
- Loss or damage of archive file
- Loss or damage of table
- Failure during data failure

Testing Operational Environment

There are a number of aspects that need to be tested. These aspects are listed below.

- **Security** – A separate security document is required for security testing. This document contains a list of disallowed operations and devising tests for each.
- **Scheduler** – Scheduling software is required to control the daily operations of a data warehouse. It needs to be tested during system testing. The scheduling software requires an interface with the data warehouse, which will need the scheduler to control overnight processing and the management of aggregations.
- **Disk Configuration.** – Disk configuration also needs to be tested to identify I/O bottlenecks. The test should be performed with multiple times with different settings.
- **Management Tools.** – It is required to test all the management tools during system testing. Here is the list of tools that need to be tested.
    - Event manager
    - System manager
    - Database manager
    - Configuration manager
    - Backup recovery manager

Testing the Database

The database is tested in the following three ways –

- **Testing the database manager and monitoring tools** – To test the database manager and the monitoring tools, they should be used in the creation, running, and management of test database.
- **Testing database features** – Here is the list of features that we have to test –
    - Querying in parallel
    - Create index in parallel

    - Data load in parallel

- **Testing database performance** − Query execution plays a very important role in data warehouse performance measures. There are sets of fixed queries that need to be run regularly and they should be tested. To test ad hoc queries, one should go through the user requirement document and understand the business completely. Take time to test the most awkward queries that the business is likely to ask against different index and aggregation strategies.

Testing the Application

- All the managers should be integrated correctly and work in order to ensure that the end-to-end load, index, aggregate and queries work as per the expectations.
- Each function of each manager should work correctly
- It is also necessary to test the application over a period of time.
- Week end and month-end tasks should also be tested.

Logistic of the Test

The aim of system test is to test all of the following areas −

- Scheduling software
- Day-to-day operational procedures
- Backup recovery strategy
- Management and scheduling tools
- Overnight processing
- Query performance

Note − The most important point is to test the scalability. Failure to do so will leave us a system design that does not work when the system grows.

FEATURES

Following are the future aspects of data warehousing.

- As we have seen that the size of the open database has grown approximately double its magnitude in the last few years, it shows the significant value that it contains.
- As the size of the databases grow, the estimates of what constitutes a very large database continues to grow.
- The hardware and software that are available today do not allow to keep a large amount of data online. For example, a Telco call record requires 10TB of data to be kept online, which is just a size of one month's record. If it requires to keep records of sales, marketing customer, employees, etc., then the size will be more than 100 TB.
- The record contains textual information and some multimedia data. Multimedia data cannot be easily manipulated as text data. Searching the multimedia data is not an easy task, whereas textual information can be retrieved by the relational software available today.
- Apart from size planning, it is complex to build and run data warehouse systems that are ever increasing in size. As the number of users increases, the size of the data warehouse also increases. These users will also require to access the system.
- With the growth of the Internet, there is a requirement of users to access data online.

Hence the future shape of data warehouse will be very different from what is being created today.