CHAPTER II

REVIEW OF LITERATURE

2.1 Introduction

Data mining addresses two basic tasks: Verification and Discovery. The verification task seeks to verify user’s hypotheses [1]. While the discovery task searches for unknown knowledge hidden in the data. In general, discovery task can be further divided into two categories. The objective of data mining is both prediction and description. That is, to predict unknown or future values of the attributes of interest using other attributes in the databases, while describing the data in a manner understandable and interpretable to humans.

2.2 Data Mining

Data mining is frequently described as "the process of extracting valid, authentic, and actionable information from large databases." In other words, data mining derives patterns and trends that exist in data. These patterns and trends can be collected together and defined as a mining model. Building a mining model is part of a larger process that includes everything from asking questions about the data and creating a model to answer those questions, to deploying the model into a working environment [4]. This process can be defined by using the following six basic steps:

- Defining the Problem
- Preparing Data
- Exploring Data
- Building Models
- Exploring and Validating Models
- Deploying and Updating Models
Figure 2.1 illustrates the relationships between each step in the process, and the technologies designed by data mining that can be used to complete a task in each step.

![Diagram of Data Mining Concepts]

**Figure 2.1: Data Mining Concepts**

**Defining the Problem**

The first step in the data mining process, as highlighted, is to clearly define the business problem. This step includes analyzing business requirements, defining the scope of the problem, defining the metrics by which the model will be evaluated, and defining specific objectives for the data mining project.

**Preparing Data**

The second step in the data mining process is to consolidate and clean the data that is identified in Defining the Problem. Data can be scattered across a company and stored in different formats, or may contain inconsistencies such as flawed or missing entries. For example, the data might show that a customer bought a product before customer was actually even born, or that the customer shops regularly at a store located 2,000 miles from her home. Before it starts to build models, it must fix these problems.
Typically, it is working with a very large dataset and cannot look through every transaction. Therefore, to use some form of automation, such as in Integration Services, to explore the data and find the inconsistencies. Therefore, before starting to build mining models, one should identify these problems and determine how to fix them.

**Exploring Data**

The third step in the data mining process is to explore the prepared data. One must understand the data in order to make appropriate decisions when creating the mining models. Exploration techniques include calculating the minimum and maximum values, calculating mean and standard deviations, and looking at the distribution of the data. Standard deviations and other distribution values can provide useful information about the stability and accuracy of the results. By exploring the data in the light of our own understanding of the business problem, we can decide if the dataset contains flawed data, and then we can devise a strategy for fixing the problems or gain a deeper understanding of the behaviors that are typical of our business.

**Building Models**

The fourth step in the data mining process is to build the mining models. One will use the knowledge that he/she gained in the Exploring Data step to help define and create the models. Before building a model, it must randomly separate the prepared data into separate training and testing datasets. It uses the training dataset to build the model, and the testing dataset to test the accuracy of the model by creating prediction queries. It can use the Percentage Sampling Transformation in Integration Services to split the dataset.
Exploring and Validating Models

The fifth step in the data mining process is to explore the models that are built and test their effectiveness. We do not want to deploy a model into a production environment without first testing how well the model performs; we may have created several models and will have to decide which model will perform the best. If none of the models created in the Building Models step perform well, we may have to return to a previous step in the process, either by redefining the problem or by reinvestigating the data in the original dataset.

Deploying and Updating Models

The last step in the data mining process is to deploy the models that performed the best in a production environment. After the mining models exist in a production environment, one can perform many tasks, depending on their needs.

2.2.1 Data Mining Task

Data Mining is a crucial and important part of the KDD process, most researchers use both terms interchangeably. Figure 2.2 represents the iterative nature of the KDD process.

- Developing an understanding of the application domain, the goals of the system and its users, and the relevant prior background and prior knowledge [14]. This is the initial preparatory step. It prepares the scene for understanding what should be
done with the many decisions (about transformation, algorithms, representation, etc.). The people who are in charge of a KDD project need to understand and define the goals of the end-user and the environment in which the knowledge discovery process will take place (including relevant prior knowledge). As the KDD process proceeds, there may be even a revision and tuning of this step.

- Selecting and creating the data set or focusing on a subset of variables or data samples, on which the goal of data that will be used for the knowledge discovery should be determined. This includes finding out what data is available, obtaining additional necessary data, and then integrating all the data for the knowledge discovery into one data set, including the attributes that will be considered for the process. This process is very important because the Data Mining learns and discovers from the available data. This is the evidence base for constructing the models. If some important attributes are missing, then the entire study may fail. From success of the process it is good to consider as many as possible attribute at this stage. On the other hand, to collect, organize and operate complex data repositories is expensive, and there is a tradeoff with the opportunity for best understanding the phenomena. This tradeoff represents an aspect where the interactive and iterative aspect of the KDD is taking place. It starts with the best available data set and later expands and observes the effect in terms of knowledge discovery and modeling.

- Pre-processing and data cleansing: In this stage, data reliability is enhanced which includes removing the noise, collecting the necessary information for modeling, selecting methods for handling missing data fields, accounting for time sequence information and changes.
• Data transformation: In this stage, the generation of better data for the data mining is prepared and developed. Methods here include dimension reduction (such as feature selection and extraction, and record sampling), and attribute transformation (such as discretization of numerical attributes and functional transformation). This step is often crucial for the success of the entire KDD project, but it is usually very project-specific and projection, finding appropriate features to represent data, using dimensionality reduction or transformation methods to reduce the number of variables to find invariant representations for data.

• Choosing the data mining task depending on the goal of KDD: clustering, classification, regression and so forth. This mostly depends on the KDD goals, and also on the previous steps.

• There are two major goals in Data Mining: Prediction and Description. Prediction is often referred to as supervised Data mining, while descriptive Data mining includes the unsupervised and visualization aspects of Data mining. Most data mining techniques are based on inductive learning, where a model is constructed explicitly or implicitly by generalizing from a sufficient number of training examples. The underlying assumption of the inductive approach is that the trained model is applicable to future cases. The strategy also takes into account the level of meta-learning for the particular set of available data.

  ➢ Selecting methods and algorithms to be used for searching for the patterns in the data.
  ➢ Mining the knowledge: searching for patterns of interest.
  ➢ Evaluating or interpreting the mined patterns, with a possible return to any previous steps.
Using this knowledge for promoting the performance of the system and resolving any potential conflicts with previously held beliefs or extracted knowledge.

These are the steps that all KDD and data mining tasks progress through.

The relative importance of description and prediction can vary between different applications. These two goals can be fulfilled by any of a number data mining tasks including: classification, regression, clustering, summarization, and Dependency modelling and deviation detection.

A. Predictive Tasks

The following are general tasks that serve predictive data mining goals:

- Classification – to segregate items into several predefined classes. Given a collection of training samples, this type of task can be designed to find a model for class attributes as a function of the values of other attributes.
- Regression – to predict a value of a given continuously valued variable based on the values of other variables, assuming either a linear or nonlinear model of dependency. These tasks are studied in statistics and neural network fields.
- Deviation Detection – to discover the most significant changes in data from previously measured or normative values.

Explicit information outside the data, like integrity constraints or predefined patterns, is used for deviation detection [35]. An approached the problem from the inside of the data, using the implicit redundancy.

B. Descriptive Tasks

- Clustering – to identify a set of categories, or clusters, that describe the data.
• Summarization – to find a concise description for a subset of data. Tabulating the mean and standard deviations for all fields is a simple example of summarization. There are more sophisticated techniques for summarization and they are usually applied to facilitate automated report generation and interactive data analysis.

• Dependency modeling – to find a model that describes significant dependencies between variables. For example, probabilistic dependency networks use conditional independence to specify the structural level of the model and probabilities or correlation to specify the strengths (quantitative level) of dependencies.

C. Mixed Tasks

There are some tasks in data mining that have both descriptive and predictive aspects. Using these tasks, movement from basic descriptive tasks toward higher-order predictive tasks is enabled [4]. Here, task are given below:

• Association Rule Discovery – Given a set of records each of which contain some number of items from a given collection, produce dependency rules which will predict the occurrence of an item based on patterns found in the data. In association, a pattern is discovered based on a relationship of a particular item on other items in the same transaction.

• Sequential Pattern Discovery – Given a set of objects, where each object is associated with its own timeline of events, find rules that predict strong sequential dependencies among different events. Rules are formed by first discovering patterns followed by event occurrences which are governed by timing constraints found within those patterns.
A brief description of the main concepts of data mining has been given. This Chapter focuses on methods and algorithms of data mining in the context of descriptive and predictive tasks. The research background of both the association rule and sequential pattern mining – newer techniques in data mining, that deserve a separate discussion – will be discussed in chapter five. Data mining does not take place in a vacuum. In other words, any application of this method of analysis is dependent upon the context in which it takes place. Therefore, it is necessary to know the environment in which data mining methods are to be used.

2.2.2 Taxonomy of Data Mining

Taxonomy is called for to help in understanding the variety of methods, their interrelation and grouping. Figure 2.3 represents the taxonomy of data mining which is useful to distinguish between two main types of Data Mining [37]: verification-oriented (the system verifies the user’s hypothesis) and discovery oriented (the system finds new rules and patterns autonomously).

Discovery methods are those that automatically identify patterns in the data. The discovery method branch consists of prediction methods versus description methods. It also develops patterns, which form the discovered knowledge in a way which is understandable and easy to operate upon. Some Prediction-oriented methods can also help provide understanding of the data. Most of the discovery-oriented Data Mining techniques (quantitative in particular) are based on inductive learning, where a model is constructed, explicitly or implicitly.

The underlying assumption of the inductive approach is that the trained model is applicable to future unseen examples. Verification methods, on the other hand, deal with the evaluation of a hypothesis proposed by an external source. These methods
include the most common methods of traditional statistics, like goodness of fit test, tests of hypotheses (e.g., t-test of means), and analysis of variance (ANOVA). These methods are less associated with Data Mining than their discovery-oriented counterparts, because most Data mining problems are concerned with discovering an hypothesis (out of a very large set of hypotheses), rather than testing a known one.

Figure 2.3: Data Mining Taxonomy

A traditional statistical method is model estimation as opposed to one of the main objectives of Data mining: model identification and construction, which is evidence based (though overlap occurs). Another common terminology, used by the machine-learning community, refers to the prediction methods as supervised learning, as opposed to unsupervised learning.

Unsupervised learning refers to modeling the distribution of instances in a typical, high-dimensional input space. Unsupervised learning refers mostly to techniques that group instances without a prespecified, dependent attribute. For instance, it covers clustering methods but not visualization methods. Supervised methods are methods that attempt to discover the relationship between input attributes (sometimes called
independent variables) and a target attribute sometimes referred to as a dependent variable).

The relationship discovered is represented in a structure referred to as a model. Usually models describe and explain phenomena, which are hidden in the data set and can be used for predicting the value of the target attribute knowing the values of the input attributes. The supervised methods can be implemented on a variety of domains, such as marketing, finance and manufacturing. It is useful to distinguish between two main supervised models: classification models and regression models. The latter map the input space into a real-value domain. For instance, a regressor can predict the demand for a certain product given its characteristics. On the other hand, classifiers map the input space into predefined classes.

### 2.2.3 Supervised Learning

Supervised methods are methods that attempt to discover the relationship between input attributes (sometimes called independent variables) and a target attribute (sometimes referred to as a dependent variable) [39]. The relationship discovered is represented in a structure referred to as a model, which are hidden in the dataset and can be used for predicting the value of the target attribute knowing the values of the input attributes. The supervised methods can be implemented in a variety of domains such as marketing, finance and manufacturing. It is useful to distinguish between two main supervised models: classification models (classifiers) and Regression Models. Regression models map the input space into a real-value domain. For instance, a regressor can predict the demand for a certain product given its characteristics. On the other hand, classifiers map the input space into pre-defined classes. There are many alternatives for representing classifiers, for example, support vector machines,
decision trees, probabilistic summaries, algebraic function, etc. Along with regression and probability estimation, classification is one of the most studied models, possibly one with the greatest practical relevance. The potential benefits of progress in classification are immense since the technique has great impact on other areas, both within Data Mining and in its applications.

2.2.4 Unsupervised Learning

Classification is used mostly as a supervised learning method, clustering for unsupervised learning. The goal of clustering is descriptive, that of classification is predictive. Since the goal of clustering is to discover a new set of categories, the new groups are of interest in themselves, and their assessment is intrinsic. In classification tasks, however, an important part of the assessment is extrinsic, since the groups must reflect some reference set of classes.

Clustering groups data instances into subsets in such a manner that similar instances are grouped together, while different instances belong to different groups. The instances are thereby organized into an efficient representation that characterizes the population being sampled. Formally, the clustering structure is represented as a set of subsets \( C = C_1, \ldots, C_k \) of \( S \), such that: \( S = \bigcup_{i=1}^{k} C_i \) and \( C_i \cap C_j = \emptyset \) for \( i \neq j \).

Consequently, any instance in \( S \) belongs to exactly one and only one subset.

- **Distance Measures**

Many clustering methods use distance measures to determine the similarity or dissimilarity between any pair of objects. It is useful to denote the distance between two instances \( x_i \) and \( x_j \) as: \( d(x_i, x_j) \) [14]. A valid distance measure should be symmetric and obtains its minimum value (usually zero) in case of identical vectors.
The distance measure is called a metric distance measure if it also satisfies the following properties:

1. Triangle inequality \( d(x_i, x_k) \leq d(x_i, x_j) + d(x_j, x_k) \) \( \forall x_i, x_j, x_k \in S \).

2. \( d(x_i, x_j) = 0 \Rightarrow x_i = x_j \) \( x_i, x_j \in S \).

**Similarity Functions**

Similarity measure is defined as the distance between various data points. The performance of many algorithms depends upon selecting a good distance function over input data set [4]. While, similarity is an amount that reflects the strength of relationship between two data items, dissimilarity deals with the measurement of divergence between two data items.

An alternative concept to that of the distance is the similarity function \( s(x_i, x_j) \) that compares the two vectors \( x_i \) and \( x_j \). A similarity function where the target range is \([0, 1]\) is called a dichotomous similarity function. In fact, the methods described in the previous sections for calculating the “distances” in the case of binary and nominal attributes may be considered as similarity functions, rather than distances. This function should be symmetrical (namely \( s(x_i, x_j) = s(x_j, x_i) \)) and have a large value when \( x_i \) and \( x_j \) are somehow “similar” and constitute the largest value for identical vectors.

Here, a brief overview of similarity measure functions used in this thesis is presented:

1. **Euclidean distance:** Euclidean distance determines the root of square differences between the coordinates of a pair of objects. For vectors \( x \) and \( y \) distance \( d(x, y) \) is given by:
\[ \text{Sim} (x, y) = d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \]  

Where \( x \) and \( y \) are \( n \)-dimensional vectors.

2. **Cosine distance**: Cosine distance measure for text clustering determines the cosine of the angle between two vectors given by the following formula \[2\]:

\[ \text{Sim} (x_i, x_j) = \cos \theta = \frac{(x_i \cdot x_j)}{||x_i|| \cdot ||x_j||} \]

Where, \( \theta \) refers to the angle between two vectors and \( x_i, x_j \) are \( n \)-dimensional vectors.

3. **Jaccard distance**: The Jaccard distance, involves the measurement of similarity as the intersection divided by the union of the data items \[3\]. The formulae could be stated as:

\[ \text{Sim} (x_i, x_j) = \frac{(x_i \cdot x_j)}{||x_i||^2 + ||x_j||^2 - (x_i \cdot x_j)} \]

4. **Pearson Correlation distance**: Pearson’s correlation distance is another measure of the extent to which two vectors are related \[3\]. The distance measure could be mathematically stated as:

\[ \text{Sim}(x, y) = \frac{\sum xy - \frac{\sum x \sum y}{n}}{\sqrt{\left(\sum x^2 - \frac{(\sum x)^2}{n}\right) \left(\sum y^2 - \frac{(\sum y)^2}{n}\right)}} \]

**2.2.5 Pattern Discovery**

The kinds of patterns that can be discovered depend upon the data mining tasks employed. There are two types of data mining tasks such as descriptive data mining
tasks that describe the general properties of the existing data, and predictive data mining tasks that attempt to do predictions based on inference on available data. The data mining functionalities and the variety of knowledge they discover are briefly presented in the following list:

**Characterization:**

Data characterization is a summarization of general features of objects in a target class, and produces what is called characteristic rules. The data relevant to a user-specified class are normally retrieved by a database query and run through a summarization module to extract the essence of the data at different levels of abstractions.

**Discrimination:**

Data discrimination produces what is called discriminant rules and is basically the comparison of the general features of objects between two classes referred to as the target class and the contrasting class.

**Association analysis:**

Association analysis is the discovery of what is commonly called association rules. It studies the frequency of items occurring together in transactional databases, and based on a threshold called support, identifies the frequent item sets [22]. Another threshold, confidence, which is the conditional probability than an item appears in a transaction when another item appears, is used to pinpoint association rules.

**Classification:**

Classification analysis is the organization of data in given classes. Also known as supervised classification, the classification uses given class labels to order the objects in the data collection [36]. Classification approaches normally use a training set where
all objects are already associated with known class labels. The classification algorithm learns from the training set and builds a model. The model is used to classify new objects.

**Prediction**:  
Prediction has attracted considerable attention given the potential implications of successful forecasting in a business context. There are two major types of predictions: one can either try to predict some unavailable data values or pending trends, or predict a class label for some data. The latter is tied to classification [38]. Once a classification model is built based on a training set, the class label of an object can be foreseen based on the attribute values of the object and the attribute values of the classes. Prediction is however more often referred to the forecast of missing numerical values, or increase/ decrease trends in time related data. The major idea is to use a large number of past values to consider probable future values.

**Clustering**:  
Similar to classification, clustering is the organization of data in classes. However, unlike classification, in clustering, class labels are unknown and it is up to the clustering algorithm to discover acceptable classes [41]. Clustering is also called unsupervised classification, because the classification is not dictated by given class labels. There are many clustering approaches all based on the principle of maximizing the similarity between objects in a same class (intra-class similarity) and minimizing the similarity between objects of different classes (inter-class similarity).

**Outlier analysis**:  
Outliers are data elements that cannot be grouped in a given class or cluster. Also known as exceptions or surprises, they are often very important to identify. While
outliers can be considered noise and discarded in some applications [40] [41], they can reveal important knowledge in other domains, and thus can be very significant and their analysis valuable.

**Evolution and deviation analysis:**

Evolution and deviation analysis pertain to the study of time related data that changes in time. Evolution analysis models evolutionary trends in data, which consent to characterizing, comparing, classifying or clustering of time related data. Deviation analysis, on the other hand, considers differences between measured values and expected values, and attempts to find the cause of the deviations from the anticipated values. It is common that users do not have a clear idea of the kind of patterns they can discover or need to discover from the data at hand. It is therefore important to have a versatile and inclusive data mining system that allows the discovery of different kinds of knowledge and at different levels of abstraction. This also makes interactivity an important attribute of a data mining system.

### 2.3 Clustering Techniques

Cluster analysis organizes data by abstracting underlying structure either as a grouping of individuals or as a hierarchy of groups, no predefined classification is required. The task is to learn a classification from the data [41]. The representation can then be investigated to see if the data group according to preconceived ideas or to suggest new experiments. Clustering is a method of unsupervised learning, and a common technique for statistical data analysis used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics [20].

Clustering can be used for:
- Exploratory data analysis: visualize the data at hand and get a feel of what the data looks like, what its properties are.

- Generalization: discover instances that are similar to each other, and hence can be handled in the same way.

Figure 2.4: Summary of Clustering Techniques

Clustering algorithms can have different properties as shown in figure 2.4:

- Hierarchical or flat: hierarchical algorithms induce a hierarchy of clusters of decreasing generality, for flat algorithms, all clusters are the same.

- Iterative: the algorithm starts with initial set of clusters and improves them by reassigning instances to clusters.

- Hard and soft: hard clustering assigns each instance to exactly one cluster. Soft clustering assigns each instance a probability of belonging to a cluster.

- Disjunctive: instances can be part of more than one cluster.
Types of Clustering

- Hierarchical algorithms find successive clusters using previously established clusters. These algorithms usually are either agglomerative ("bottom-up") or divisive ("top-down"). Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters. Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters [37].

- Partition algorithms typically determine all clusters at once, but can also be used as divisive algorithms in the hierarchical clustering.

- Density-based clustering algorithms are devised to discover arbitrary-shaped clusters. In this approach, a cluster is regarded as a region in which the density of data objects exceeds a threshold. DBSCAN and OPTICS are two typical algorithms of this kind [41].

- Subspace clustering methods look for clusters that can only be seen in a particular projection (subspace, manifold) of the data. These methods thus can ignore irrelevant attributes. The general problem is also known as Correlation clustering while the special case of axis-parallel subspaces is also known as Two-way clustering, co-clustering or bi-clustering: in these methods not only the objects are clustered but also the features of the objects, i.e., if the data is represented in a data matrix, the rows and columns are clustered simultaneously. They usually do not however work with arbitrary feature combinations as in general subspace methods. But this special case deserves attention due to its applications in bioinformatics.
Many clustering algorithms require the specification of the number of clusters to produce in the input data set, prior to execution of the algorithm. Barring knowledge of the proper value beforehand, the appropriate value must be determined, a problem on its own for which a number of techniques have been developed.

Density-based methods: Most partitioning methods cluster objects based on the distance between objects. Such methods can find only spherical-shaped clusters and encounter difficulty at discovering clusters of arbitrary shapes. Other clustering methods have been developed based on the notion of density. Their general idea is to continue growing the given cluster as long as the density (number of objects or data points) in the “neighborhood” exceeds some threshold; that is, for each data point within a given cluster, the neighborhood of a given radius has to contain at least a minimum number of points. Such a method can be used to filter out noise (outliers) and discover clusters of arbitrary shape.

Grid-based methods: Grid-based methods quantize the object space into a finite number of cells that form a grid structure. All of the clustering operations are performed on the grid structure (i.e., on the quantized space). The main advantage of this approach is its fast processing time, which is typically independent of the number of data objects and dependent only on the number of cells in each dimension in the quantized space.

Model-based methods: Model-based methods hypothesize a model for each of the clusters and find the best fit of the data to the given model. A model-based algorithm may locate clusters by constructing a density function that reflects the spatial distribution of the data points. It also leads to a way of
automatically determining the number of clusters based on standard statistics, taking “noise” or outliers into account and thus yielding robust clustering methods.

**Partitioning Method**

Suppose we are given a database of n objects, the partitioning method construct k partition of data. Each partition will represent a cluster and k≤n. It means that it will classify the data into k groups, which satisfy the following requirements:

- Each group contains at least one object.
- Each object must belong to exactly one group.

These algorithms, which tend to work well with isolated and compact clusters, are the most intuitive and frequently used methods [20]. The basic idea is to find a clustering structure that minimizes a certain error criterion which measures the “distance” of each instance to its representative value. The most well-known criterion is the Sum of Squared Error (SSE), which measures the total squared Euclidian distance of instances to their representative values.

The simplest and most commonly used algorithm, employing a squared error criterion is the K-means algorithm. This algorithm partitions the data into K clusters (C1,C2…CK) represented by their centers or means. The center of each cluster is calculated as the mean of all the instances belonging to that cluster.

The algorithm starts with an initial set of cluster centers, chosen at random or according to some heuristic procedure. In each iteration, each instance is assigned to its nearest cluster center according to the Euclidean distance between the two. Then
the cluster centers are re-calculated. The center of each cluster is calculated as the mean of all the instances belonging to that cluster:

Input: S (instance set), K (number of cluster)

Output: clusters

1: Initialize K cluster centers.

2: while termination condition is not satisfied do

3: Assign instances to the closest cluster center.

4: Update cluster centers based on the assignment.

5: End while

Another partitioning algorithm is K-medoids or PAM (Partition Around Medoids). This algorithm is very similar to the K-means algorithm. It differs from the latter mainly in its representation of the different clusters. Each cluster is represented by the most centric object in the cluster, rather than by the implicit mean that may not belong to the cluster. The K-medoids method is more robust than the K-means algorithm in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean. Both methods require the user to specify K, the number of clusters. Starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering works effectively for small data sets, but does not scale well for large data sets.
A. Hierarchical Clustering

Hierarchical clustering is a method of cluster analysis which seeks to build a hierarchy of clusters as referred in figure 2.4. Strategies for hierarchical clustering generally fall into two types:

- **Agglomerative**: This is a "bottom up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.

- **Divisive**: This is a "top down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

➤ **Agglomerative Hierarchical clustering**

Hierarchical clustering algorithms are either top-down or bottom-up. Bottom-up algorithms treat each document as a singleton cluster at the outset and then successively merge (or agglomerate) pairs of clusters until all clusters have been merged into a single cluster. Bottom-up hierarchical clustering is therefore called hierarchical agglomerative clustering. Top-down clustering requires a method for splitting a cluster.

Some of the available methods are

1) single-nearest distance or single linkage.
2) complete-farthest distance or complete linkage.
3) average-average distance or average linkage.
4) Centroid distance.
5) Ward's method - sum of squared Euclidean distance is minimized.
In this manner, grouping the data is done, until one cluster is formed. Now on the basis of dendrogram graph it can calculate the numbers of clusters that should be actually present.

➢ **Divisive Hierarchical clustering**

It is just the reverse of Agglomerative Hierarchical approach.

**Advantages**

1) No Apriori information about the number of clusters required.

2) Easy to implement and gives the best result in some cases.

**Disadvantages**

1) Algorithm can never undo what was done previously.

2) Time complexity of at least $O(n^2 \log n)$ is required, where ‘$n$’ is the number of data points.

3) Based on the type of distance matrix chosen for merging different algorithms, they can suffer with one or more of the following:

   i) Sensitivity to noise and outliers

   ii) Breaking large clusters

   iii) Difficulty handling different sized clusters and convex shapes

4) No objective function is directly minimized

5) Sometimes it is difficult for the dendrogram to identify the correct number of clusters.
The result of the hierarchical methods is a dendrogram, representing the nested grouping of objects and similarity levels at which groupings change. A clustering of the data objects is obtained by cutting the dendrogram at the desired similarity level.

The merging or division of clusters is performed according to some similarity measure, chosen so as to optimize some criterion (such as a sum of squares). The hierarchical clustering methods could be further divided according to the manner that the similarity measure is calculated.

- **Single-link clustering** (also called the connectedness, the minimum method or the nearest neighbor method) — methods that consider the distance between two clusters to be equal to the shortest distance from any member of one cluster to any member of the other cluster [39]. If the data consist of similarities, the similarity between a pair of clusters is considered to be equal to the greatest similarity from any member of one cluster to any member of the other cluster.

- **Complete-link clustering** (also called the diameter, the maximum method or the furthest neighbor method) - methods that consider the distance between two clusters to be equal to the longest distance from any member of one cluster to any member of the other cluster.

- **Average-link clustering** (also called minimum variance method) - methods that consider the distance between two clusters to be equal to the average distance from any member of one cluster to any member of the other cluster.
The disadvantages of the single-link clustering and the average-link clustering can be summarized as follows: Single-link clustering has a drawback known as the “chaining effect”:

A few points that form a bridge between two clusters cause the single-link clustering to unify the two into one.

- Average-link clustering may cause elongated clusters to split and portions of neighboring elongated clusters to merge.

- The complete-link clustering methods usually produce more compact clusters and more useful hierarchies than the single-link clustering methods, yet the single-link methods are more versatile.

Generally, hierarchical methods are characterized by the following strengths:

- Versatility—The single-link methods, for example, maintain good performance on data sets containing non-isotropic clusters, including well-separated, chainlike and concentric clusters.

- Multiple partitions — hierarchical methods produce not one partition, but multiple nested partitions, which allow different users to choose different partitions, according to the desired similarity level. The hierarchical partition is presented using the dendrogram.

**B Non Hierarchical Clustering**

In Non-Hierarchical clustering encompass several methods to build a cluster as refer Figure6. Such as,
A single-pass method is one in which the partition is created by a single pass through the data set or, if randomly accessed, in which each compound is examined only once to decide which cluster it should be assigned to.

A relocation method is one in which compounds are moved from one cluster to another to try and improve on the initial estimation of the clusters. The relocating is typically accomplished based on improving a cost function, describing the ‘goodness’ of each resultant cluster.

The nearest-neighbor approach is more compounds-centered than the other nonhierarchical methods. In it, the environment around each compound is examined in terms of its most similar neighboring compounds, with commonality between nearest neighbors being used as a criterion for cluster formation.

The nearest-neighbor lists and creates the clusters in this stage of nonhierarchical clustering; there are three steps to be carried out:

1. Tag each compound with a sequential cluster label so that each is a singleton.

2. For each pair of compounds, i and j (i < j), compare the nearest-neighbor lists on the basis of the three neighborhood conditions. If the three conditions are passed, replace the cluster label for compound j with the cluster label for compound i. Then, scan all previously processed compounds and replace any occurrences of the cluster label for compound j by the cluster label for compound i.
3. Scan to extract clusters by retrieving all compounds assigned the same cluster label. The Jarvis–Patrick method requires \( O(N^2) \) time and \( O(N) \) space.

- In mixture model clustering the data are assumed to exist as a mixture of densities that are usually assumed to be Gaussian (normal) distributions, since their densities are not known in advance. Solutions to the mixture model are derived iteratively in a manner similar to the relocation methods.

The well-known relocation method is the k-means method, for which there exist many variants and different algorithms for its implementation [41]. The k-means algorithm minimizes the sum of the squared Euclidean distances between each item in a cluster and the cluster centroid. The basic method used most frequently in chemical applications proceeds as follows:

1. Choose an initial set of \( k \) seed compounds to act as initial cluster centers.
2. Assign each compound to its nearest cluster centroid (classification step).
3. Recalculate each cluster centroid (minimization step).
4. Repeat steps 2 and 3 for a given number of iterations or until no compounds are moved from one cluster to another.

- Density-based, or mode-seeking, methods regard the distribution of descriptors across the data set as generating patterns of high and low density that, when identified, can be used to separate the compounds into clusters.

C Unsupervised Clustering

Unsupervised prototype-based clustering aims at determining the correct number of clusters, without any prior knowledge about it, using one of four approaches. The first
approach is to proceed by repeating the clustering for several values at a high computational cost, and using a validity measure to choose the best partition. The second approach is to perform several passes through the data set, seeking one cluster at a time and then removing from the data set of the next pass the points belonging to a found cluster if it passes a validity test as in the GMVE [40]. The problem with these approaches lies in the difficulty in designing validity measures that can truly evaluate the goodness of fit of a given cluster or partition because they usually require setting thresholds that can widely vary in practice. Also, most validity measures either assume a known underlying inliers or noise distribution or are very sensitive to noise, and hence are not appropriate for general robust clustering. The third approach consists of starting the clustering process with an over specified number of clusters, and then merging similar clusters and eliminating spurious clusters until the correct number of clusters is left as in Compatible Cluster Merging. The fourth and most recent approach is based on Competitive Agglomeration, which starts by partitioning the data set into an over specified number of clusters. Then, as the clustering progresses, adjacent clusters compete against each other for data points, and clusters that lose in the competition gradually become depleted and vanish.

2.4 Association Rule Mining

Association rules mining are one of the major techniques of data mining and it is perhaps the most common form of local-pattern discovery in unsupervised learning systems. Association is one of the best known data mining technique. Data mining refers to discover knowledge in huge amounts of data. It is a scientific discipline that is concerned with analyzing observational data sets with the objective of finding unsuspected relationships and produces a summary of the data in novel ways that the owner can understand and use.
The technique is likely to be very practical in applications which use the similarity in customer buying behaviour in order to make peer recommendations. Association rule mining finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories. There are various techniques proposed for generating frequent itemset so that association rules are mined efficiently. The approaches of generating frequent itemset are divided into basic three techniques [22].

- **Horizontal layout based data mining techniques**
  - Apriori algorithm
  - DHP algorithm
  - Partition
  - Sample

- **Vertical layout based data mining techniques**
  - Éclat algorithm

- **Projected database based data mining techniques**
  - FP-tree algorithm
  - H-mine algorithm

Data patterns and models can be mined from many different kinds of databases, such as Relational Databases, Data Warehouses, Transactional Databases, and Advanced Database Systems (Object-Oriented, Relational, Spatial and Temporal, Time-Series, Multimedia, Text, Heterogeneous, Legacy, Distributed, and WWW).

Association rule mining can be defined formally as follows:

\[ I = \{i_1, i_2, i_3 \ldots \text{in} \} \] is a set of items, such as products like (computer, CD, printer, papers, and so on). Let DB be a set of database transactions where each
transaction T is a set of items such that T⊆I. Each transaction is associated with unique identifier, transaction identifier (TID). Let X, Y be a set of items, an association rule has the form X→Y, where

\[ X \cap Y = \emptyset. \] X is called the antecedent (conditional part of the rule, and can be found in the data) and Y is called the consequent of the rule (the item found in combination with the antecedent) where X, Y is a set of items is called as an itemset or a pattern. Let \( \text{freq}(X) \) be the number of rows (transactions) containing X itemset in the given database.

**Measures of Association Rules**

The task of mining association rules is essentially to discover strong association rules in large databases. After the large item sets are identified, the corresponding association rules can be derived in a straightforward manner. Much of the research work has been focused on the first sub problem and as the database is accessed in this part of the computation, we concentrate on this sub problem.

Essentially association mining is about discovering a set of rules that is shared among a large percentage of data. There are two ways of measuring usefulness, being objectively and subjectively. Objective measures involve statistical analysis of the data, such as Support and Confidence.

**Support**

The rule X→Y holds with support s if s% of transactions in D contain X →Y. Rules that have a ‘s’ greater than a user-specified support is said to have minimum support.
Confidence

The rule $X \rightarrow Y$ holds with confidence $c$ if $c\%$ of transactions in $D$ that contain $X$ also contain $Y$. Rules that have a ‘$c$’ greater than a user-specified confidence is said to have a minimum confidence [37].

An itemset (or a pattern) is frequent if its support is equal to or more than a user specified minimum support (a statement of generality of the discovered association rules). Association rule mining is to identify all rules meeting user-specified constraints such as minimum support and minimum confidence (a statement of predictive ability of the discovered rules). One key step of association mining is frequent itemset (pattern) mining, which is to mine all itemsets satisfying user specified minimum support.

However a large number of these rules will be pruned after applying the support and confidence thresholds. Therefore the previous computations will be wasted. To avoid this problem and to improve the performance of the rule discovery algorithm, mining association rules may be decomposed into two phases:

1. Discover the large itemsets, i.e., the sets of items that have transaction support above a predetermined minimum threshold known as frequent Itemsets.

2. Use the large itemsets to generate the association rules for the database that have confidence above a predetermined minimum threshold.

The overall performance of mining association rules is determined primarily by the first step. The second step is easy. After the large item set are identified, the corresponding association rules can be derived in straightforward manner. Our main consideration of the thesis is First step i.e. to find the extraction of frequent item set.
2.5 Classification Techniques

Classification is a data mining technique used to classify each item in a set of data into one of predefined set of classes or groups. Data classification is a two step process.

- In the first step, a model is built by analyzing the data tuples from training data having a set of attributes. For each tuple in the training data, the value of class label attribute is known. Classification algorithm is applied on training data to create the model.

- In the second step of classification, test data is used to check the accuracy of the model. If the accuracy of the model is acceptable then the model can be used to classify the unknown data tuples.

Modeling step contributes in applying the classification techniques to build classifier models. Description (Classification) and prediction are two forms of data analysis that can be used to extract models describing important data classes or to predict future data trends, while classification predicts categorical labels (classes), prediction models continuous valued functions. Model evaluation step demonstrate the evaluation result of classifiers in graphical form.

There are two main supervised models: classification models (classifiers) and Regression models. Regression models map the input space into a real-value domain. Classifiers map the input space into pre-defined classes. There are many alternatives for representing classifiers like support vector machines, decision trees, probabilistic summaries, algebraic function, etc.
2.5.1 Classification and Prediction

Classification is a task in Data mining. Data mining, as indicated before, is a machine learning discipline, and is inspired by pattern recognitions, which is a branch of science, of which one of its goals is to classify objects into a number of categories referred to as classes.

Objects refer to compact data units specific to a particular problem, which is in general, known as patterns. Classification prediction encompasses two levels: classifier construction and the usage of the classifier constructed. The former is concerned with the building of a classification model by describing a set of predetermined classes from a training set as a result of learning from that dataset. Each sample in the training set is assumed to belong to a predefined class, as determined by the class attribute label. The model is represented as classification rules, decision trees, or mathematical formula.

The later involves the use of a classifier built to predict or classify unknown objects based on the patterns observed in the training set. The entire process begins with collection of evidence acquired from various data sources or warehouses. In the ideal situation, the data should be of low dimensionality, independent and discriminative so that its values are very similar to characteristics in the same class but very different in features from different classes.

Raw data hardly satisfies these conditions and therefore a set of procedures called feature generation, extraction and selection is required to provide a relevant input for Classification system [7]. Prediction is a wide topic and runs from predicting the failure of components or machinery, to identifying fraud and even the prediction of...
company profits. Used in combination with the other data mining techniques, prediction involves analyzing trends, classification, pattern matching, and relation.

Analyzing past events or instances helps make a prediction about an event. Using the credit card authorization, for example, to combine decision tree analysis of individual past transactions with classification and historical pattern matches to identify whether a transaction is fraudulent. Making a match between the purchase of flights to the US and transactions in the US, it is likely that the transaction is valid.

- Classification algorithms
  - Decision tree
  - Rule-based induction
  - Neural networks
  - Memory(Case) based reasoning
  - Genetic algorithms
  - Bayesian networks

- Prediction Techniques

Prediction is achieved with the help of regression. Regression analysis can be used to model the relationship between one or more independent or predictor variables and a dependent or response variable (continuous value).

  - Linear regression
  - Non Linear Regression

Data patterns and models can be mined from many different kinds of databases, such as Relational Databases, Data Warehouses, Transactional Databases, and Advanced
Classification and Prediction can be defined as follows:

- Assumption: after data preparation, we have a data set where each record has attributes $X_1, \ldots, X_n$, and $Y$.

- Goal: learn a function $f : (X_1, \ldots, X_n) \rightarrow Y$, then use this function to predict $y$ for a given input record $(x_1, \ldots, x_n)$. – Classification: $Y$ is a discrete attribute, called the class label.

- Usually a categorical attribute with small domain – Prediction: $Y$ is a continuous attribute called supervised learning, because true labels are known for the initially provided data.

- Typical applications: credit approval, target marketing, medical diagnosis, fraud detection.

**Building a Classification and Prediction System**

The building of a classification process model can be broken down into four major components. Those are technique choice, data pre-processing, training, and testing or evaluation. This ends the classification part of the process. The next phase is the decision making stage. This may involve risk or reliability analysis; it could be model tuning aimed at minimizing the cost or further context-based optimization.

**Technique selection**

As the “no free lunch” theorem suggests, there is no technique that has been proven to offer the best solution to all classification or prediction of problems [40]. The decision regarding the selection of the classification model is critical as well as difficult,
especially if there is little prior knowledge about the nature of the problem. Another problem stems from the fact that the classification process is to a large extent unpredictable and quite often nondeterministic, which means that the appropriateness of the choice made cannot be immediately justified [7].

This can be achieved after going through the entire classification process and a feedback on the performance is established. A new iteration begins again when the feedback indicates poor or unsatisfactory performance until an acceptable level of confidence is established in the performance feedback.

**Data Pre-processing and Transformation**

Data pre-processing and transformation is the process of converting information or data from one format to another format. While the strategy is often thought of in terms of converting documents from one format to another, data transformations may also involve converting programs from one type of computer language to a different format in order to allow the program to run on a specific platform. The actual transformation may involve converting multiple data streams into a common format, or converting a single format into multiple different forms for use across a wide spectrum of platforms.

In actual use, data transformation involves the use of an executable program that is capable of reading the base or original language of the data, and identifying the language or languages that the data must translate into in order to be used by other programs. Once the mapping for the transformation is accomplished, the program then converts the data into the single or multiple formats desired, and distributes the converted data accordingly. With many applications, this takes place in a matter of seconds.
Learning

Learning is the term used to describe the actual process of training the classification model. The process could be categorized into three, depending on the availability and reliability of the evidence [7]. One can distinguish three learning strategies: Supervised, Unsupervised. Supervised and unsupervised learning differ only in the causal structure of the model. In supervised learning, the model defines the effect one set of observations, called inputs, has on another set of observations, called outputs. In other words, the inputs are assumed to be at the beginning and outputs at the end of the causal chain. The models can include mediating variables between the inputs and outputs. In unsupervised learning, all the observations are assumed to be caused by latent variables, that is, the observations are assumed to be at the end of the causal chain. In practice, models for supervised learning often leave the probability for inputs undefined. This model is not needed as long as the inputs are available, but if some of the input values are missing, it is not possible to infer anything about the outputs. If the inputs are also modeled, then missing inputs cause no problem since they can be considered latent variables as in unsupervised learning.

Testing

The testing stage is arguably the most critical phase of the classification model development process. This stems from the fact that it offers the model developer the most informative measure of classifier performance which then could justify its use, leading to possible optimization, redesign or elimination of other models showing bad performance. This in effect will form the basis of model selection [4]. Model testing has some caveats, which must be noted. It argues that the common notion that a more elaborate classifier that produces complex non-linear class boundaries is better than
simple linear models may not be always true. The complex models tend to over fit the training data so that although their performance on the training set is usually much better than simple linear models, they could show very weak performance for new dataset.

2.5.2 Classification Algorithms

A. Decision Trees

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 [35]. 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 subspaces according to a certain discrete function of the input attributes values.

The decision tree can be constructed from the given set of attributes. While some of the trees are more accurate than others, finding the optimal tree is computationally infeasible because of the exponential size of the search space. These algorithms usually employ a greedy strategy that grows a decision tree by making a series of locally optimum decision about which attributes to use for partitioning the data. Such algorithm called Hunt algorithm which is basis of many existing decision tree induction algorithm including ID3, C4.5, and CART.

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 values.
HUNT Algorithm

Hunt’s algorithm works if every combination of attribute values is present in the training data and each combination has a unique class label. Let \( D_t \) be the set of training record that are associated with node t and \( y = \{y_1, y_2, y_3...y_n\} \). These assumptions are too stringent for use in most practical situation [35].

Step 1: If all the records in \( D_t \) belong to the same class \( y_t \) then t is the leaf node labeled as \( y_t \).

Step 2: If \( D_t \) contains the records that belong to more than one class, an attribute test condition is selected to partition the records into smaller subsets. A child node is created for each outcomes of the test condition and the records in \( D_t \) are distributed to the children based on the outcomes. The algorithm is then recursively applied to each child node.

It works if every combination of attribute values is present in the training data and each combination has a unique class label.

Top-Down Algorithmic Framework for Decision Trees Induction.

Tree_growing (S, A, y, SplitCriterion, StoppingCriterion)

where:

S - Training Set

A - Input Feature Set

y - Target Feature

SplitCriterion - the method for evaluating a certain split

StoppingCriterion - the criteria to stop the growing process
Create a new tree T with a single root node.

IF StoppingCriterion(S) THEN

Mark T as a leaf with the most common value of y in S as a label.

ELSE

Find attribute that obtains the best SplitCriterion (ai, S).

Label t with a

FOR each outcome vi of a:

Set Subtreei = TreeGrowing (ai=vi, S, y).

Connect the root node of tT to Subtreei with an edge that is labelled as vi

END FOR

END IF

RETURN TreePruning (S, T, y)

TreePruning (S, T, y) Where: S - Training Set

y - Target Feature T - The tree to be pruned

DO

Select a node t in T such that pruning it maximally improve some evaluation criteria

IF t ≠Ø THEN T = pruned(T, t)

UNTIL t = Ø

RETURN T
B. Univariate Splitting

Univariate means that an internal node is split according [37] to the value of a single attribute. These criteria can be characterized in different ways, such as:

- According to the origin of the measure: information theory, dependence, and distance.
- According to the measure structure: impurity based criteria, normalized impurity based criteria and Binary criteria.

Impurity-based Criteria

Given a random variable \( x \) with \( k \) discrete values, distributed according to \( P = (p_1, p_2, \ldots, p_k) \), an impurity measure is a function \( \varphi : [0, 1]^k \to R \) that satisfies the following conditions:

- \( \varphi (P) \geq 0 \)
- \( \varphi (P) \) is minimum if \( \exists i \) such that component \( p_i = 1 \).
- \( \varphi (P) \) is maximum if \( \forall i, 1 \leq i \leq k, p_i = 1/k \).
- \( \varphi (P) \) is symmetric with respect to components of \( P \).
- \( \varphi (P) \) is smooth (differentiable everywhere) in its range.

Note that if the probability vector has a component of 1 (the variable \( x \) gets only one value), then the variable is defined as pure. On the other hand, if all components are equal, the level of impurity reaches maximum.

Given a training set \( S \), the probability vector of the target attribute \( y \) is defined as:

\[
P_y(S) = \left( \frac{\sum_{y=c_1} S}{|S|}, \ldots, \frac{\sum_{y=c_{\text{dom}(y)}} S}{|S|} \right)
\]
The goodness-of-split due to discrete attribute $a_i$ is defined as reduction in impurity of the target attribute after partitioning $S$ according to the values $v_{i,j} \in \text{dom}(a_i)$:

$$\Delta \Phi(a_i, S) = \varphi (P_y(S)) - \sum_{j=1}^{\text{dom}(a_i)} \frac{|a_i = v_{i,j}|}{|S|} \varphi (P_y(\sigma_{a_i = v_{i,j}} S))$$

**Information Gain**

Information gain is an impurity-based criterion that uses the entropy measure (origin from information theory) as the impurity measure (Quinlan, 1987).

$$\text{Entropy}(y, S) = -\sum_{v_{i,j} \in \text{dom}(a_i)} \frac{\sigma_{a_i = v_{i,j}} S}{|S|} \cdot \text{Entropy}(y, \sigma_{a_i = v_{i,j}} S)$$

where:

$$\text{Entropy}(y, S) = \sum_{v_{c,j} \in \text{dom}(y)} \frac{\sigma_{y = c,j} S}{|S|} \cdot \log_2 \frac{\sigma_{y = c,j} S}{|S|}$$

**Gini Index**

Gini index is an impurity-based criterion that measures the divergences between the probability distributions of the target attribute’s values. The Gini index has been used in various works and it is defined as:

$$\text{Gini}(y, S) = 1 - \sum_{c_j \in \text{dom}(y)} \left( \frac{\sigma_{y = c,j} S}{|S|} \right)^2$$

Consequently the evaluation criterion for selecting the attribute $a_i$ is defined as:

$$\text{GiniGain}(a_i, S) = \text{Gini}(y, S) - \sum_{v_{i,j} \in \text{dom}(a_i)} \frac{\sigma_{a_i = v_{i,j}} S}{|S|} \cdot \text{Gini}(y, \sigma_{a_i = v_{i,j}} S)$$
Likelihood-Ratio Chi–Squared Statistics

The likelihood–ratio is defined as

\[ G^2(a_i, S) = 2 \cdot \ln(2) \cdot |S| \cdot InformationGain(a_i, S) \]

This ratio is useful for measuring the statistical significance of the information gain criterion. The zero hypothesis (H0) is that the input attribute and the target attribute are conditionally independent. If H0 holds, the test statistic is distributed as \( \chi^2 \) with degrees of freedom equal to: \((\text{dom}(a_i) - 1) \cdot (\text{dom}(y) - 1)\).

DKM Criterion

The DKM criterion is an impurity-based splitting criterion designed for binary class attributes (Dietterich \textit{et al.}, 1996) and (Kearns and Mansour, 1999). The impurity-based function is defined as:

\[ DKM(y, S) = 2 \vee \left( \frac{\left| \sigma_{y=c_1} \right|}{|S|} \right) \left( \frac{\left| \sigma_{y=c_2} \right|}{|S|} \right) \]

It has been theoretically proved (Kearns and Mansour, 1999) that this criterion requires smaller trees for obtaining a certain error than other impurity based criteria (information gain and Gini index).

Normalized Impurity Based Criteria

The impurity-based criterion described above is biased towards attributes with larger domain values. Namely, it prefers input attributes with many values over attributes with less values (Quinlan, 1986). For instance, an input attribute that represents the national security number will probably get the highest information gain. However,
adding this attribute to a decision tree will result in a poor generalized accuracy. For that reason, it is useful to “normalize” the impurity based measures, as described in the following sections.

**Gain Ratio**

The gain ratio “normalizes” the information gain as follows

\[
\text{GainRatio}(a_i, S) = \frac{\text{InformationGain}(a_i, S)}{\text{Entropy}(a_i, S)}
\]

Note that this ratio is not defined when the denominator is zero. Also the ratio may tend to favor attributes for which the denominator is very small. Consequently, it is suggested in two stages. First the information gain is calculated for all attributes. As a consequence, taking into consideration only attributes that have performed at least as good as the average information gain, the attribute that has obtained the best ratio gain is selected. It has been shown that the gain ratio tends to outperform simple information gain criteria, both from the accuracy aspect, as well as from classifier complexity aspects.

**Distance Measure**

The distance measure, like the gain ratio, normalizes the impurity measure. However, it suggests normalizing it in a different way (Lopez de Mantras, 1991):

\[
\Delta \phi(a_i, S) = \sum_{v_i, j \in \text{dom}(a_i)} \sum_{k \in \text{dom}(y)} \sum_{\sigma_{a_i=v_i, j} \land \sigma_y=c_k} \frac{|S|}{|S|} \cdot \log_2 \frac{|S|}{|S|}
\]
Binary Criteria

The binary criteria are used for creating binary decision trees. These measures are based on division of the input attribute domain into two sub-domains. Let \( \beta (ai, \text{dom1} (ai), \text{dom2} (ai), S) \) denote the binary criterion value for attribute \( ai \) over sample \( S \) when \( \text{dom1} (ai) \) and \( \text{dom2} (ai) \) are its corresponding sub domains. The value obtained for the optimal division of the attribute domain into two mutually exclusive and exhaustive sub-domains is used for comparing attributes.

Twoing Criterion

The gini index may encounter problems when the domain of the target attribute is relatively wide. In this case it is possible to employ binary criterion called twoing criterion. This criterion is defined as:

\[
\text{twoing}(a_i, \text{dom}_1(a_i), \text{dom}_2(a_i), S) = 0.25 \left( \frac{\sum_{y \in \text{dom}_1} \sum_{y \notin \text{dom}_2} P_y}{\sum_{y \in \text{dom}_1} \sum_{y \notin \text{dom}_2} P_y} \right) \left( \frac{\sum_{y \notin \text{dom}_1} \sum_{y \in \text{dom}_2} P_y}{\sum_{y \notin \text{dom}_1} \sum_{y \in \text{dom}_2} P_y} \right)
\]

When the target attribute is binary, the gini and twoing criteria are equivalent. For multi-class problems, the twoing criteria prefer attributes with evenly divided splits.

Orthogonal (ORT) Criterion

This binary criterion is defined as:

\[
\text{ORT} (a_i, \text{dom1}(a_i), \text{dom2}(a_i), S) = 1 - \cos \theta (P_{y1}, P_{y2})
\]

where \( \theta (P_{y1}, P_{y2}) \) is the angle between two vectors \( P_{y1} \) and \( P_{y2} \). These vectors represent the probability distribution of the target attribute in the partitions \( x_i \in \text{dom}_1(a_i)S \) and \( x_i \in \text{dom}_2(a_i)S \) respectively. It has been shown that this criterion performs better than the information gain and the Gini index for specific problem constellations.
Kolmogorov–Smirnov Criterion

A binary criterion that uses Kolmogorov–Smirnov distance has been proposed in Friedman (1977) and Rounds (1980). Assuming a binary target attribute, namely \( \text{dom}(y) = \{c_1, c_2\} \), the criterion is defined as:

\[
\text{KS}(a_1, \text{dom}_1(a_1), \text{dom}_2(a_1), S) = \frac{\sum_{a_i \in \text{dom}_1(a_1) \text{AND} y = c_1} S}{\sigma_{y = c_1} S} - \frac{\sum_{a_i \in \text{dom}_1(a_1) \text{AND} y = c_2} S}{\sigma_{y = c_2} S}
\]

o handle target attributes with multiple classes and missing data values. Their results indicate that the suggested method outperforms the gain ratio criteria.

AUC–Splitting Criteria

The idea of using the AUC metric as a splitting criterion, is an attribute that obtains the maximal area under the convex hull of the ROC curve. It has been shown that the AUC–based splitting criterion outperforms other splitting criteria both with respect to classification accuracy and area under the ROC curve. It is important to note that unlike impurity criteria, this criterion does not perform a comparison between the impurities of the parent node with the weighted impurity of the children after splitting.

Stopping Criteria

The growing phase continues until a stopping criterion is triggered. The following conditions are common stopping rules:

1. All instances in the training set belong to a single value of \( y \).
2. The maximum tree depth has been reached.
3. The number of cases in the terminal node is less than the minimum number of cases for parent nodes.

4. If the node were split, the number of cases in one or more child nodes would be less than the minimum number of cases for child nodes.

5. The best splitting criteria is not greater than a certain threshold.

C. Pruning Methods

Overview

Employing tightly stopping criteria tends to create small and under–fitted decision trees. On the other hand, using loosely stopping criteria tends to generate large decision trees that are over–fitted to the training set. Pruning methods originally suggested in (Breiman et al., 1984) were developed for solving this dilemma. According to this methodology, a loosely stopping criterion is used, letting the decision tree to over fit the training set. Then the over-fitted tree is cut back into a smaller tree by removing sub–branches that are not contributing to the generalization accuracy. It has been shown in various studies that employing pruning methods can improve the generalization performance of a decision tree, especially in noisy domains. When the goal is to produce a sufficiently accurate compact concept description, pruning is highly useful. Within this process, the initial decision tree is seen as a completely accurate one. Thus the accuracy of a pruned decision tree indicates how close it is to the initial tree. There are various techniques for pruning decision trees. Most of them perform top-down or bottom-up traversal of the nodes. A node is pruned if this operation improves a certain criteria. The following subsections describe the most popular techniques.
Cost–Complexity Pruning

Cost-complexity pruning (also known as weakest link pruning or error-complexity pruning) proceeds in two stages. The first stage is a sequence of trees $T_0$, $T_1$, $T_k$ is built on the training data where $T_0$ is the original tree before pruning and $T_k$ is the root tree. In the second stage, one of these trees is chosen as the pruned tree, based on its generalization error estimation.

The tree $T_{i+1}$ is obtained by replacing one or more of the sub–trees in the predecessor tree $T_i$ with suitable leaves. The sub–trees that are pruned are those that obtain the lowest increase in apparent error rate per pruned leaf:

$$\alpha = \frac{\varepsilon(pruned(T,t),S) - \varepsilon(T,S)}{|leaves(T)| - |leaves(pruned(T,t))|}$$

where $\varepsilon(T, S)$ indicates the error rate of the tree $T$ over the sample $S$ and $|leaves(T)|$ denotes the number of leaves in $T$. $pruned(T,t)$ denotes the tree obtained by replacing the node $t$ in $T$ with a suitable leaf.

In the second phase the generalization error of each pruned tree $T_0$, $T_1$. . . , $T_k$ is estimated. The best pruned tree is then selected. If the given dataset is large enough, the authors suggest breaking it into a training set and a pruning set. The trees are constructed using the training set and evaluated on the pruning set. On the other hand, if the given dataset is not large enough, they propose to use cross–validation methodology, despite the computational complexity implications.

Reduced Error Pruning

A simple procedure for pruning decision trees, known as reduced error pruning, has been suggested by Quinlan (1987). While traversing over the internal nodes from the bottom to the top, the procedure checks for each internal node, whether replacing it
with the most frequent class does not reduce the tree’s accuracy. In this case, the node is pruned. The procedure continues until any further pruning would decrease the accuracy.

In order to estimate the accuracy, Quinlan (1987) proposes to use a pruning set. It can be shown that this procedure ends with the smallest accurate sub-tree with respect to a given pruning set.

**Minimum Error Pruning (MEP)**

The minimum error pruning performs bottom-up traversal of the internal nodes. In each node it compares the l-probability error rate estimation with and without pruning. The l-probability error rate estimation is a correction to the simple probability estimation using frequencies [22]. If \( St \) denotes the instances that have reached a leaf \( t \), then the expected error rate in this leaf is:

\[

\varepsilon'(t) = -\frac{1}{n} \max_{c_i \in dom(y)} \left( \frac{\sigma_{y=c_i} S_t + I \cdot pr(y = c_i)}{|S_t| + I} \right)

\]

where \( pr(y = c_i) \) is the \( \text{a-priori} \) probability of \( y \) getting the value \( c_i \), and \( I \) denotes the weight given to the \( \text{a-priori} \) probability. The error rate of an internal node is the weighted average of the error rate of its branches. The weight is determined according to the proportion of instances along each branch. The calculation is performed recursively up to the leaves. If an internal node is pruned, then it becomes a leaf and its error rate is calculated directly using the last equation. Consequently, we can compare the error rate before and after pruning a certain internal node. If pruning this node does not increase the error rate, the pruning should be accepted.
Pessimistic Pruning

Pessimistic pruning avoids the need of pruning set or cross validation and uses the pessimistic statistical correlation test.

The basic idea is that the error ratio estimated using the training set is not reliable enough. Instead, a more realistic measure, known as the continuity correction for binomial distribution, should be used:

\[ \varepsilon'(T, S) = \varepsilon(T, S) + \frac{|leaves(T)|}{2|S|} \]

However, this correction still produces an optimistic error rate. Consequently, one should consider pruning an internal node \( t \) if its error rate is within one standard error from a reference tree, namely (Quinlan, 1993):

\[ \varepsilon'(\text{pruned}(T, t), S) \leq \varepsilon'(T, S) + \sqrt{\frac{\varepsilon'(T, S)(1-\varepsilon'(T, S))}{|S|}} \]

The last condition is based on statistical confidence interval for proportions. Usually the last condition is used such that \( T \) refers to a sub-tree whose root is the internal node \( t \) and \( S \) denotes the portion of the training set that refers to the node \( t \).

The pessimistic pruning procedure performs top-down traversing over the internal nodes. If an internal node is pruned, then all its descendants are removed from the pruning process, resulting in a relatively fast pruning.

Error-based Pruning (EBP)

Error-based pruning is an evolution of pessimistic pruning. It is implemented in the well-known C4.5 algorithm.
As in pessimistic pruning, the error rate is estimated using the upper bound of the statistical confidence interval for proportions.

\[ \varepsilon \cup B(T, S) = \varepsilon(T, S) + Z_{\alpha} \frac{\varepsilon(T, S)(1 - \varepsilon(T, S))}{|S|} \]

where \( \varepsilon \cup B(T, S) \) denotes the misclassification rate of the tree \( T \) on the training set \( S \). \( Z \) is the inverse of the standard normal cumulative distribution and \( \alpha \) is the desired significance level.

Let \( \text{subtree} (T, t) \) denote the subtree rooted by the node \( t \). Let \( \text{maxchild} (T, t) \) denote the most frequent child node of \( t \) (namely most of the instances in \( S \) reach this particular child) and let \( St \) denote all instances in \( S \) that reach the node \( t \). The procedure performs bottom-up traversal over all nodes and compares the following values:

1. \( \varepsilon UB(\text{subtree} (T, t), St) \)
2. \( \varepsilon UB(\text{pruned} (\text{subtree} (T, t), t), St) \)
3. \( \varepsilon UB(\text{subtree} (T, \text{maxchild} (T, t)), S_{\text{maxchild}} (T, t)) \)

According to the lowest value the procedure either leaves the tree as is, prune the node \( t \), or replaces the node \( t \) with the subtree rooted by \( \text{maxchild} (T, t) \).

**Optimal Pruning**

The issue of finding optimal pruning introduced an algorithm which guarantees optimality, known as OPT. This algorithm finds the optimal pruning based on dynamic programming, with the complexity of \( \Theta(|\text{leaves} (T)|^2) \), where \( T \) is the initial decision tree. The second research introduced an improvement of OPT called OPT–2, which also performs optimal pruning using dynamic programming. However, the time
and space complexities of OPT–2 are both \( \Theta (|\text{leaves} (T^*)| \cdot |\text{internal} (T)|) \), where \( T^* \) is the target (pruned) decision tree and \( T \) is the initial decision tree.

Since the pruned tree is habitually much smaller than the initial tree and the number of internal nodes is smaller than the number of leaves, OPT–2 is usually more efficient than OPT in terms of complexity.

**Minimum Description Length (MDL) Pruning**

The minimum description length can be used for evaluating the generalized accuracy of a node (Rissanen, 1989, Quinlan and Rivest, 1989, Mehta et al., 1995). This method measures the size of a decision tree by means of the number of bits required to encode the tree. The MDL method prefers decision trees that can be encoded with fewer bits. The cost of a split at a leaf \( t \) can be estimated as (Mehta et al., 1995):

\[
\text{Cost}(t) = \sum_{c_i \in \text{dom}(y)} |\sigma_{y=c_i} S_t| \cdot \ln \left( \frac{|S_t|}{|\sigma_{y=c_i} S_t|} \right) + \frac{|\text{dom}(y)| - 1}{2} \ln \frac{|S_t|}{2} + \ln \frac{\pi}{r} \left( \frac{|\text{dom}(y)|}{2} \right)
\]

where \( S_t \) denotes the instances that have reached node \( t \). The splitting cost of an internal node is calculated based on the cost aggregation of its children.

**D. Measures of Regression**

Regression models involve the following variables:

- The unknown parameters, denoted as \( \beta \), which may represent a scalar or a vector.
- The independent variables \( X \).
- The dependent variable, \( Y \).
In various fields of application, different terminologies are used in place of dependent and independent variables [4]. A regression model relates $Y$ to a function of $X$ and $\beta$. The approximation is usually formalized as $E(Y \mid X) = f(X, \beta)$. To carry out regression analysis, the form of the function $f$ must be specified. Sometimes the form of this function is based on knowledge about the relationship between $Y$ and $X$ that does not rely on the data. If no such knowledge is available, a flexible or convenient form for $f$ is chosen.

Assume now that the vector of unknown parameters $\beta$ is of length $k$. In order to perform a regression analysis the user must provide information about the dependent variable $Y$:

- If $N$ data points of the form $(Y, X)$ are observed, where $N < k$, most classical approaches to regression analysis cannot be performed: since the system of equations defining the regression model is underdetermined, there are not enough data to recover $\beta$.
- If exactly $N = k$ data points are observed, and the function $f$ is linear, the equations $Y = f(X, \beta)$ can be solved exactly rather than approximately. This reduces to solving a set of $N$ equations with $N$ unknowns (the elements of $\beta$), which has a unique solution as long as the $X$ are linearly independent. If $f$ is nonlinear, a solution may not exist, or many solutions may exist.
- The most common situation is where $N > k$ data points are observed. In this case, there is enough information in the data to estimate a unique value for $\beta$ that best fits the data in some sense, and the regression model when applied to the data can be viewed as an over determined system in $\beta$.

In the last case, the regression analysis provides the tools for:
• Finding a solution for unknown parameters $\beta$ that will, for example, minimize the distance between the measured and predicted values of the dependent variable $Y$ (also known as method of least squares).

• Under certain statistical assumptions, the regression analysis uses the surplus of information to provide statistical information about the unknown parameters $\beta$ and predicted values of the dependent variable $Y$.

Consider a regression model which has three unknown parameters, $\beta_0$, $\beta_1$, and $\beta_2$. Suppose to perform in experiment, 10 measurements all at exactly the same value of independent variable vector $X$ (which contains the independent variables $X_1$, $X_2$, and $X_3$). In this case, regression analysis fails to give a unique set of estimated values for the three unknown parameters; the experiment did not provide enough information. The best one can do is to estimate the average value and the standard deviation of the dependent variable $Y$. Similarly, measuring at two different values of $X$ would give enough data for a regression with two unknowns, but not for three or more unknowns.

• If the experimenter had performed measurements at three different values of the independent variable vector $X$, then regression analysis would provide a unique set of estimates for the three unknown parameters in $\beta$.

• In the case of general linear regression, the above statement is equivalent to the requirement that the matrix $X^TX$ is invertible.

E. Bayesian Network

Bayesian network (BNs) are graphical models for reasoning under uncertainty, where the nodes represent variables (discrete or continuous) and arcs represent direct connections between them. These direct connections are often causal connections. In addition, BNs model the quantitative strength of the connections between variables,
allowing probabilistic beliefs about them to be updated automatically as new information becomes available.

BNs correspond to another GM (Graphical Model) structure known as a Directed Acyclic Graph (DAG) that is popular in the statistics, the machine learning, and the artificial intelligence societies. The structure of a DAG is defined by two sets: the set of nodes (vertices) and the set of directed edges. The nodes represent random variables and are drawn as circles labeled by the variable names. The edges represent direct dependence among the variables and are drawn by arrows between nodes. In particular, an edge from node Xi to node Xj represents a statistical dependence between the corresponding variables. Thus, the arrow indicates that a value taken by variable Xj depends on the value taken by variable Xi, or roughly speaking that variable Xi “influences” Xj. Node Xi is then referred to as a parent of Xj and, similarly, Xj is referred to as the child of Xi. An extension of these genealogical terms is often used to define the sets of “descendants” – the set of nodes that can be reached on a direct path from the node, or “ancestor” nodes – the set of nodes from which the node can be reached on a direct path [4]. The structure of the acyclic graph guarantees that there is no node that can be its own ancestor or its own descendent. Such a condition is of vital importance to the factorization of the joint probability of a collection of nodes.

A Bayesian network B is an annotated acyclic graph that represents a JPD over a set of random variables V. The network is defined by a pair $B = \{G, \Theta\}$, where G is the DAG whose nodes X1, X2... Xn represents random variables, and whose edges represent the direct dependencies between these variables. The graph G encodes independence assumptions, by which each variable Xi is independent of its non descendents given its parents in G. The second component denotes the set of
parameters of the network. This set contains the parameter \( \theta_{x_i|\Omega_i} = P_B(x_i|\Omega_i) \) for each realization \( x_i \) of \( X_i \) conditioned on \( \pi_i \), the set of parents of \( X_i \) in \( G \). Accordingly, \( B \) defines a unique JPD over \( V \), namely:

\[
P_B(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P_B(X_i|\Omega_i) = \prod_{i=1}^{n} \theta_{x_i|\Omega_i}
\]

For simplicity of representation we omit the subscript \( B \) henceforth. If \( X_i \) has no parents, its local probability distribution is said to be unconditional, otherwise it is conditional. If the variable represented by a node is observed, then the node is said to be an evidence node, otherwise the node is said to be hidden or latent.

F. Rule induction

Rule induction is one of the most important techniques of machine learning. Since regularities hidden in data are frequently expressed in terms of rules, rule induction is one of the fundamental tools of Data Mining at the same time. Usually rules are expressions of the form

\[
\text{if (attribute}_1, \text{value}_1) \text{ and (attribute}_2, \text{value}_2) \text{ and } \cdots \text{ and (attribute}_n, \text{value}_n) \text{ then (decision, value)}.
\]

Some rule induction systems induce more complex rules, in which values of attributes may be expressed by negation of some values or by a value subset of the attribute domain.

Data from which rules are induced are usually presented in a form similar to a table in which cases (or examples) are labels (or names) for rows and variables are labeled as attributes and a decision. We will restrict our attention to rule induction which belongs to supervised learning: all cases are preclassified by an expert. In different
words, the decision value is assigned by an expert to each case. Attributes are independent variables and the decision is a dependent variable. Obviously, numerical attributes must be converted into symbolic attributes before or during rule induction. The process of converting numerical attributes into symbolic attributes is called discretization (or quantization).

G. Neural Network

Neural networks [1] and [35] are computing models for information processing and are particularly useful for identifying the fundamental relationship among a set of variables or patterns in the data. It’s able to solve problems that have imprecise patterns or data containing incomplete and noisy information with a large number of variables. This fault tolerance feature is appealing to data mining problems because real data are usually dirty and do not follow clear probability structures that typically required by statistical models.

Back propagation learns by iteratively processing a data set of training tuples, comparing the network’s prediction for each tuple with the actual known target value. The target value may be the known class label of the training tuple (for classification problems) or a continuous value (for prediction). For each training tuple, the weights are modified so as to minimize the mean squared error between the network’s prediction and the actual target value. These modifications are made in the “backwards” direction, that is, from the output layer, through each hidden layer down to the first hidden layer (hence the name Back propagation).

Initialize the weights: The weights in the network are initialized to small random numbers. Each unit has a bias associated with it, as explained below. The biases are similarly initialized to small random numbers.
Each training tuple, $X$, is processed by the following steps. Propagate the inputs forward: First, the training tuple is fed to the input layer of the network. The inputs pass through the input units, unchanged. That is, for an input unit, $j$, its output, $O_j$, is equal to its input value, $I_j$. Next, the net input and output of each unit in the hidden and output layers are computed. The net input to a unit in the hidden or output layers is computed as a linear combination of its inputs. To help illustrate this point, a hidden layer or output layer unit is shown in Figure 2.5. Each such unit has a number of inputs to it that are, in fact, the outputs of the units connected to it in the previous layer. Each connection has a weight. To compute the net input to the unit, each input connected to the unit is multiplied by its corresponding weight, and this is summed. Given a unit $j$ in a hidden or output layer, the net input, $I_j$, to unit $j$ is

$$I_j = \sum_i w_{ij} O_i + \theta_j$$

where $w_{ij}$ is the weight of the connection from unit $i$ in the previous layer to unit $j$; $O_i$ is the output of unit $i$ from the previous layer; and $\theta_j$ is the bias of the unit.

![Figure 2.5. Neural Network](image)

The bias acts as a threshold in that it serves to vary the activity of the unit. Each unit in the hidden and output layers takes its net input and then applies an activation function to it, as illustrated in Figure 2.5. The function symbolizes the activation of
the neuron represented by the unit. The logistic, or sigmoid, function is used. Given the net input $I_j$ to unit $j$, then $O_j$, the output of unit $j$, is computed as

$$O_j = \frac{1}{1 + e^{-I_j}}$$

This function is also referred to as a *squashing function*, because it maps a large input domain onto the smaller range of 0 to 1. The logistic function is nonlinear and differentiable, allowing the back propagation algorithm to model classification problems that are linearly inseparable.

The output values, $O_j$, for each hidden layer, up to and including the output layer, which gives the network’s prediction. In practice, it is a good idea to cache (i.e., save) the intermediate output values at each unit as they are required again later, when back propagating the error. This trick can substantially reduce the amount of computation required.

2.6 Other Classification Approach

2.6.1 Genetic Algorithms

A more recent technique applied to classification problems is GA which are heuristic search techniques based on the theory of natural selection and evolution [13]. Most data mining-related GA are used in the task of rule extraction in propositional and first-order logic. GA are efficient search methods based on the principles of natural selection and population genetics in which random operators on a population of candidate solutions are employed to generate new points in the search space [12].

- GA processes:

A genetic algorithm is an iterative procedure until a pre-determined stopping condition (usually the number of generation). Genetic algorithm involves a population
of individuals, each one represented by a finite string of symbols, known as the genome, encoding a possible solution in a given problem space.

A **genetic operator** is an operator used in genetic algorithms to maintain genetic diversity, known as Mutation (genetic algorithm) and to combine existing solutions into others, Crossover (genetic algorithm). The main difference between them is that the mutation operators operate on one chromosome, that is, they are unary, while the crossover operators are binary operators.

➢ **Selection**

While there are many different types of selection, the most common type is roulette wheel selection. In roulette wheel selection, individuals are given a probability of being selected that is directly proportionate to their fitness. Two individuals are then chosen randomly based on these probabilities to produce offspring.

The simple genetic algorithm as a pseudo code is:

```pseudo
begin
initialize g to 0
initialize members of P0 to random values
evaluate fitness of members of P0
while (termination condition not reached) loop
increment g
select members of Pg-1 to compose Pg
randomly change members of Pg
evaluate fitness of members of Pg
end
```
end loop

close

While this code is very general and will obviously not compile, it illustrates the basic structure of a selection algorithm.

➢ Crossover

The crossover operator represents the mixing of genetic material from two selected parent chromosomes to produce one or two child chromosomes. Once the two parent chromosomes are selected, a random number between 0-1 is produced. If the random number is greater than a certain crossover rate, the exchange between two chromosomes is required. In the present study, the crossover rate is set as 0.5. Generally the single-point crossover is used, i.e. only one specific chromosome is selected for the exchange at a time.

In this technique, the child is derived from three parents. They are randomly chosen. Each bit of first parent is checked with bit of second parent whether they are same. If same then the bit is taken for the offspring otherwise the bit from the third parent is taken for the offspring. For example, the following three parents:

parent1 1 1 0 1 0 0 0 1 0

parent2 0 1 1 0 0 1 0 0 1

parent3 1 1 0 1 1 0 1 0 1

Produces the following offspring:

offspring 1 1 0 1 0 0 0 0 1

The multi-point crossover is used in the present study where the random numbers are generated first when the exchange is required and the chromosomes are exchanged.
based on the actual situation. The most common solution is called crossover, and while there are many different kinds of crossover, the most common type is single point crossover. In single point crossover, it chooses a locus at which you swap the remaining alleles from one parent to the other. This is complex and is best understood visually.

Generally, the children take one section of the chromosome from each parent. The point at which the chromosome is broken depends on the randomly selected crossover point. This particular method is called single point crossover because only one crossover point exists. Sometimes only child 1 or child 2 is created, but oftentimes both offspring are created and put into the new population.

The most important one is the crossover operation. In the crossover operation, two solutions are combined to form two new solutions or offspring. The parents are chosen from the population by a function of the fitness of the solutions.

▶ Mutation

After selection and crossover, we have a new population full of individuals. Some are directly copied, and others are produced by crossover. In order to ensure that the individuals are not all exactly the same, it allow for a small chance of mutation. Loop through all the alleles of all the individuals, and if that allele is selected for mutation, it can be either changed by a small amount or replaced with a new value. The probability of mutation is usually between 1 and 2 tenths of a percent.

This mutation operator takes the chosen genome and inverts the bits. (i.e. if the genome bit is 1, it is changed to 0 and vice versa)

The mutation of bit strings ensue through bit flips at random positions.
Example:

\[
\begin{array}{cccccccc}
 & & & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
\downarrow & & & & & & & & & \\
\end{array}
\]

\[
\begin{array}{cccccccc}
 & & & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\
\end{array}
\]

**Advantages of Genetic Algorithm**

- It can solve every optimization problem which can be described with the chromosome encoding.
- It solves problems with multiple solutions. Since the genetic algorithm execution technique is not dependent on the error surface, it can solve multi-dimensional, non-differential, non-continuous and even non-parametrical problems.
- Structural genetic algorithm gives us the possibility to solve the solution structure and solution parameter problems at the same time by means of genetic algorithm.
- Genetic algorithm is a method which is very easy to understand and it practically does not demand the knowledge of mathematics. Genetic algorithms are easily transferred to existing simulations and models.

**2.6.2 Fuzzy Set Approach**

Rule-based systems for classification have the disadvantage that they involve sharp cutoffs for continuous attributes. Fuzzy set theory is useful for data mining systems performing rule-based classification [35]. It provides operations for combining fuzzy measurements. Suppose that in addition to the fuzzy sets for income, we defined the fuzzy sets junior employee and senior employee for the attribute years employed. Suppose also that we have a rule that, say, tests high income and senior employee in
the rule antecedent (IF part) for a given employee, x. If these two fuzzy measures are AND together, the minimum of their measure is taken as the measure of the rule. In other words, m (high income AND senior employee) (x) = min (m high income(x), m senior employee(x)). This is akin to saying that a chain is as strong as its weakest link.

If the two measures are OR, the maximum of their measure is taken as the measure of the rule. In other words, m (high income OR senior employee) (x) = max (m high income(x), m senior employee(x)): Intuitively, this is like saying that a rope is as strong as its strongest strand.

Given a tuple to classify, more than one fuzzy rule may apply. Each applicable rule attributes a vote for membership in the categories. Typically, the truth values for each predicted category are summed, and these sums are combined. Several procedures exist for translating the resulting fuzzy output into a de-fuzzified or crisp value that is returned by the system.

2.7 Pattern Matching Algorithm

Detection Systems (IDSs) are widely used and recognized as powerful tools for identifying the data in efficient manner. Essentially every intrusion detection system is the ability to search an element through packets and identify an element that matches known attacks.

2.7.1 Knuth-Morris-Pratt Algorithm

The design of the Knuth- Morris- Pratt algorithm [4] and [5] followed by analysis of the Morris and Pratt algorithm which improve the length of the shifts. It compares the pattern with the text from left to right. It decreases the time of searching compared to the Brute Force algorithm. It is a heuristic based algorithm, which requires a verification algorithm following a possible match to verify if a true match occurs. In
the preprocessing phase the space and time complexity is $O(m)$. In the searching phase the time complexity is $O(n+m)$, where $n$ is the length (size) of the file and $m$ is the length of the pattern and that the algorithm never has to move backwards in the text that is being searched. This makes it ideal for processing very large files.

If a mismatch is found, KMP uses the pre-computed number of shifts to determine how far to shift. An algorithm is explained in the following steps:

1. Get the length of the text ‘$n$’, length of the pattern ‘$m$’, prefix function of pattern ‘$p$’ and number of character matched $c$.
2. Initialize the variable $c=0$ for match from beginning.
3. To compare first character of the pattern to the first character of the text. If it is not, match to substitute the value of $p[c]$ to $c$. Otherwise it will be increment as $c=1$.
4. Check whether all the pattern elements are matched with the text elements. If it is not, repeat the process; otherwise the elements are printed by the required pattern.

This makes it ideal for processing very large files. The strongest weakness Knuth-Morris-Pratt has, is that it does not work well as the size of the alphabet increases. This will lead to a higher chance of a mismatch [2].

### 2.7.2 Boyer Moore Algorithm

The design of Boyer Moore algorithm contains three special parts to carry out the task such as Good suffix rule, the bad character rule and the pattern will scan from right to left. If there is a mismatch, then the shift is decided based upon two other rules. Working phases of the task are Matching phase in $O(n/m)$, $O(n)$ time complexity [3].
The bad character rule works: Let us assume that a mismatch is going to occur between the character $p[i] = a$ of the pattern and the character $c[i+j] = b$ of the text during an attempt at position $j$. Then, $p[i+1 .. m] = t[i+j+1 .. j+m] = u$ and $p[i] \neq c[i+j]$. If $b$ is not contained anywhere in $p$, then shift the pattern completely to text $[i+j]$. Otherwise, shift the pattern $p$ until the rightmost occurrence of the character $b$ in $p$ gets aligned with $c[i+j]$.

Good-suffix rule works: The algorithm looks up string $u$ leader character is not $b$ in $P$ from right to left [5]. If such a segment exists, shift right $P$ to get a new attempt window. If there exists no such segment, the shift consists in aligning the longest suffix $v$ of $C[i+j+1 ... j+m-1]$ with a matching prefix of $P$.

If the pattern is never found in the text, then the worst runtime of the algorithm would be $O(n+m)$ time. When the pattern is found in the text, then the worst runtime would be $O(nm)$. If the pattern however is very short or has a low probability to be found, then the algorithm is not so optimal to use.

2.8 Data Mining in the Pharmaceutical Industry

Data mining is the process of extracting information from large data sets through the use of algorithms and techniques drawn from the field of Statistics, Machine Learning and Data Base Management Systems. It uses variety of tools like query and reporting tools, analytical processing tools, and Decision Support System (DSS) tools.

A variety of decision support capabilities will be necessary to increase the productivity of medical personnel, analyze care outcomes, and continually refine care delivery processes to remain profitable while holding the line on costs and maintaining quality of care. There are seven steps in the drug discovery process: disease selection, target hypothesis, lead compound identification (screening), lead
optimization, pre-clinical trial, and clinical trial and pharmacogenomic optimization [34].

According to Cooman, the molecules are clustered into groups according to the chemical properties of the molecules via cluster analysis. This way every time a new molecule is discovered it can be grouped with other chemically similar molecules.

2.8.1 Techniques of Data Mining used in Pharma Sector

The main techniques of data mining used in pharma sector are clustering, classification and neural networks. Table 2.1 provides a clear understanding of how data mining techniques will really add value to Pharma data.

<table>
<thead>
<tr>
<th>Major Data Mining Techniques</th>
<th>Patterns</th>
</tr>
</thead>
</table>
| Clustering                  | • Diseases having similar characteristics  
                              | • Grouping top drugs  
                              | • Analysis of workforce attrition in Pharma sector  
                              | • Groups of drugs most likely to be used, most unlikely to be used  
                              | • Segments related to drugs, diseases |
| Classification and prediction | • Predicting consumer behavior  
                              | • Predicting the likelihood of success in a drug adoption process  
                              | • Predicting the percentage accuracy in  
                              | • performance of a drug  
                              | • Analyze forecast and model information to quantify human capital assets  
                              | • Classifying the historical health records  
                              | • Prediction of what type of drugs most likely to be retained, most likely to be left, most likely to transform their composition |
| Association, Mining Frequent Patterns | • Predicting pharma product behavior and attitude  
• Predicting demand projections by seasonal variations  
• Predicting the performance progress of segments throughout the performance period  
• Identifying the best profile for different drugs  
• Prediction to find what factors will attract new avenues in pharma sector  
• Classify trends of movements through the organization for successful/unsuccessful patient historical records  
• Categorization of drugs, diseases and patients |
| Data mining using other interdisciplinary methods | • Association of training undertaken diseases with drugs  
• Association and analysis of staff movements  
• Application tracking mechanism in physicians adopting drugs with customer’s prescription  
• Standardizing training methods, task performance monitoring, monitor workflow route tasks  
• Provides summary information: various multidimensional summary reports  
• Statistical summary information (data central tendency and variation)  
• Use historical data to build models of fraudulent behavior and use data |
2.8.2 Literature Review in Pharmaceutical Data Analysis

According to Burbidge, the classification task is determined for predicting the inhibition of dihydrofolate reductase by pyrimidines, using data obtained from the UCI machine learning repository [33]. Three artificial neural networks, a radial basis function network, and a C5.0 decision tree are all outperformed by the SVM. The SVM is significantly better when compared with other techniques.

According to Jun Xu and Arnold Hagler, an applications of cheminformatics in drug discovery, such as compound selection, virtual library generation, virtual high throughput screening, HTS data mining, and in silico ADMET are analyzed by structural similarity matrices, and classification algorithms of Data mining [42].

According to Christodoulous A. Nicolaou, the methods used for analyzing HTS data using cluster analysis also found natural groups in the data, thereby revealing families of compounds that exhibit increased activity towards a specific biological target. It is used traditionally a number of clustering algorithms, distance (similarity) measures, and compound representation methods and determined the nature of chemical and biological data and how it adversely impacts the current analysis methodology. It also emphasized the inability of widely used methods to discover the chemical families in a pharmaceutical dataset and point out the occurring of specific problems when one attempts to apply these common clustering and other statistical methods on chemical data [34]. Data-mining algorithm of clustering method accommodates user requests and produce chemically sensible results. It can find the true chemical structural families of compounds in pharmaceutical data, while at the same time accommodate the multi-domain nature of chemical compounds.
2.8.3 Features of Data Mining in Drug Design

Data mining, popularly called as knowledge discovery in large data, enables firms and organizations to make calculated decisions by assembling, accumulating, analyzing and accessing corporate data. It uses variety of tools like query and reporting tools, analytical processing tools, and Decision Support System (DSS) tools.

Many drug discovery technology companies are attempting to address this challenge through developing solutions that will force new drugs to "fail faster and safer." Although this is a noble goal and if realized, would definitely create value for the industry, these solutions often over promise and underestimate the obstacles that stand in the path to lower clinical failure rates [4]. Marketing strategies centered on increasing revenues will be more convincing than those that address reducing expense. Demonstrating that technologies will enable pharmaceutical companies to better target and market to certain customer segments will increase adoption of that technology, and will open the door for projects aimed at reducing costs and increasing clinical trials throughput.

In medical decision making, it shows awareness of need to introduce data analysis techniques in the extraction of knowledge, representative patient cases stored in database. These data enable understanding the safety and efficacy profiles within the patient population. By tackling the question of patient selection within the framework of demonstrating groups that are most responsive, Data mining is sure to penetrate the drug development marketplace. This framework enables specialists to create customized nodes that can be shared throughout the organization, making the application attractive to skilled modelers in a pharmaceutical company’s bioinformatics division.
This research need to use data mining tools and techniques. This can be achieved by clustering the molecules into groups according to the chemical properties of the molecules via cluster analysis (Cooman 2005). This way, every time a new molecule is discovered, it can be grouped with other chemically similar molecules. This would help the researchers in finding out the therapeutic group the new molecule would belong to. Mining can help us to measure the chemical activity of the molecule on a specific disease - say tuberculosis and find out which part of the molecule is causing the action. This way we can combine a vast number of molecules forming a super molecule with only the specific part of the molecule which is responsible for the action and inhibiting the other parts.

This would greatly reduce the adverse effects associated with drug actions. Scientists run experiments to determine activity of potential drugs. They use high speed screening to test tens, hundreds, or thousands of drugs very quickly. The general goal is to find activity on relevant genes or to find drug compounds that have desirable characteristics (whatever those may be). The Data mining techniques that are used in developing new drugs are clustering, classification and neural networks. The basic objective is to determine compounds with similar activity. The reason is, similar activity compounds behave similarly. This is possible only when we have known compound and are looking for something better.

Large amount of data is being collected every day in many business and science areas [1]. This data needs to be analyzed in order to find interesting information, and one of the most important analyzing methods is data clustering.

Clustering is one of the most important data mining tools which help data analyzers to understand the natural grouping of attributes in the data [ ]. Cluster analysis is used in
many field such as data mining [35], pattern recognition and pattern classification [4], data compression [14], machine learning, image analysis, and bioinformatics [8]. Data clustering is methods in which a cluster of objects is made that are somehow similar in characteristics. The criterion for checking the similarity is implementation dependent.

It can be classified into two major classes, as hierarchical and partitioning algorithms. A hierarchical method uses a nested sequence of partitions. This can be done by considering all data as one cluster and then dividing it into smaller ones, this is called Divisive clustering. The other class considers each data point as cluster and then merges them to form a bigger cluster, called Agglomerative clustering.

**Components of a clustering task**

Typical pattern clustering activity involves the following steps [10]:

- Pattern representation.
- Definition of a pattern proximity measure appropriate to the data domain.
- Clustering or grouping
- Data abstraction (if needed)
- Assessment of output (if needed).

**Similarity measure**

Since similarity is fundamental to the definition of a cluster, a measure of the similarity between two patterns drawn from the same feature space is essential to most clustering procedures. Because of the variety of feature types and scales, the
distance measure (or measures) must be chosen carefully. It is most common to calculate the dissimilarity between two patterns using a distance measure defined on the feature space. The most popular metric for continuous features is the Euclidean distance.

Euclidean distance or Euclidean metric is the "ordinary" distance between two points that one can measure with a ruler, and is given by the Pythagorean formula (Eq.1). It is used for determining the closeness of each data point to the cluster centroids. The distance between one vector \( X = (x_1, x_2, ..., x_n) \) and another vector \( Y = (y_1, y_2, ..., y_n) \) is obtained as

\[
\sqrt{d(x, y) = (x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_n - y_n)^2} \tag{1}
\]

**Partition Clustering Algorithms**

A partition clustering algorithm obtains a single partition of the data instead of a clustering structure, such as the dendrogram produced by a hierarchical technique. Partition methods have advantages in applications involving large data sets for which the construction of a dendrogram is computationally prohibitive. A problem accompanying the use of a partition algorithm, is the choice of the number of desired output clusters.

One of the most famous partition algorithms is the K-means. K-means is the simplest and most commonly used algorithm employing a squared error criterion [13]. It starts with random initial centroids and keeps reassigning the patterns to clusters, based on the similarity between the pattern and the cluster centroids, until a convergence criterion is met after some number of iterations. The K-means algorithm is popular
because it is easy to implement, and its time complexity is $O(n)$, where $n$ is the number of patterns.

The basic algorithm works as follows

**K-means algorithm**

The k-means algorithm proceeds as follows.

Algorithm: The k-means clustering algorithm

Require: $D = \{d_1, d_2, d_3, \ldots, d_n\}$ // Set of $n$ data points.

$k$ // Number of desired clusters

Ensure: A set of $k$ clusters.

Steps:

1. Arbitrarily choose $k$ data points from $D$ as initial centroids;

2. Repeat

   Assign each point $d_i$ to the cluster which has the closest centroid;

   Calculate the new mean for each cluster;

   Until convergence criteria is met.

**2.9 Data Mining Application**

Data mining is an interdisciplinary field with wide and diverse applications. It is a data analysis approach that has been quickly adapted and used in a large number of domains that were already using statistics. A data mining application, or data mining tool, is typically a software interface which interacts with a large database containing customer or other important data.
Some of these organizations include retail stores, hospitals, banks, and insurance companies. Many of these organizations are combining data mining with such things as statistics, pattern recognition, and other important tools.

Data mining is widely used by companies and public bodies for such uses as marketing, detection of fraudulent activity, and scientific research. Data mining applications are often structured around the specific needs of an industry sector or even tailored and built for a single organization. This is because the patterns within data may be very specific.

Some of the areas where data mining techniques are applied:

- Medicine / Pharmacy.
- Insurance.
- Health Care.
- Banking / Financial data analysis
- Retail / Marketing.
- Telecommunications industry.
- Science and engineering.
- Biomedical research.
- Transportation.
- Business & E-commerce Data.

### 2.10 Challenges in Data Mining

Data mining systems face a lot of problems and pitfalls. Advancements in data mining with various integrations and implications of methods and techniques have shaped the
present data mining applications to handle the various challenges. A system which is quick and correct on some small training sets, could behave completely different when applied to a larger database. A data mining system may work perfectly for consistent data and perform significantly worse when a little noise is added to the training set. Research in data mining has led to advanced knowledge discovery technologies and applications.

The main emerging research issues for advanced technologies and applications in data mining are

- Developing a Unifying Theory of Data mining
- Scaling Up for High Dimensional Data and High Speed Data Streams
- Mining Sequence Data and Time Series Data
- Mining Complex Knowledge from Complex Data
- Data mining in a Network Setting
- Distributed Data mining and Mining Multi-agent Data
- Data mining for Biological and Environmental Problems
- Data-mining-Process Related Problems
- Security, Privacy and Data Integrity
- Dealing with Non-static, Unbalanced and Cost-sensitive Data

The research challenges are arranged into five broad areas:

- improving the scalability of data mining algorithms
- mining non-vector data
- mining distributed data
improving the ease of use of data mining systems and environments, and
privacy and security issues for data mining.

Mining different kinds of knowledge, data mining should cover a wide spectrum of
data analysis and knowledge discovery tasks. It is difficult to know exactly what can
be discovered within a database; the data mining process should be interactive.
Interactive mining users need to focus the search for patterns providing and referring
data mining request based on returned results.

Data mining has become an essential technology for businesses and researchers in
many fields, the number and variety of applications has been growing gradually for
several years and it is predicted that it will carry on to grow. Data mining is a
relatively new technology that has not fully matured. Despite this, there are a number
of industries that are already using it on a regular basis. Some of these organizations
include banking, insurance, retail and telecom, pharmaceutics, health, government and
all sorts of e-businesses.

Many of these organizations are combining data mining with statistics, pattern
recognition, and other important tools.

Data mining can be used to find patterns and connections that would otherwise be
difficult to find. This technology is popular with many businesses because it allows
them to learn more about their customers and make smart marketing decisions.

- In healthcare it’s used for Drug development, Medical diagnostics.
- Insurance – pricing, fraud detection, risk analysis
- Stock Market – market timing, stock selection, risk analysis
- Transportation – performance & network optimization to predict life-cycle
costs of road pavement
Telecommunications – churn reduction
Retail - market basket analysis to help determine marketing strategies
Banks - to detect which customers are using which products so they can offer the right mix of products and services to better meet customer needs i.e., cross sell and up sell.
Credit card companies - to assist in mailing promotional materials to people who are most likely to respond.
Lenders - to determine which applicants are most likely to default on a loan.

Recent Research Achievements
Data mining is a new powerful technology which helps in extracting hidden predictive information from large databases and thus facilitating decision makers to make proactive, knowledge-driven decisions. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified.

The opportunities today in data mining rest solidly on a variety of research achievements, which were interdisciplinary in nature, resting on discoveries made by researchers from disciplines working together collaboratively. Recent achievements are made in areas such as

- Neural Network
- Tree-based Classifiers
- Graphical Models and Hierarchical Probabilistic Representations
- Linear Algebra
2.11 Data Mining Tools

The most important commonly used data mining tools are

- **R**

R is a well supported, open source, command line driven, statistics package. It provides all sorts of data mining, machine learning and statistical techniques. It is particularly used in the areas of bio-informatics and social science.

- **Rapid Miner**

Rapid Miner, formerly YALE (Yet Another Learning Environment), is an environment for machine learning, data mining, text mining, predictive analytics, and business analytics. It is used for research, education, training, rapid prototyping, application development, and industrial applications.

- **WEKA**

Weka (Waikato Environment for Knowledge Analysis) is a popular suite of machine learning software written in Java, developed at the University of Waikato, New Zealand. The Weka workbench contains a collection of visualization tools and algorithms for data analysis and predictive modeling, together with graphical user interfaces for easy access to this functionality. Weka supports several standard data
mining tasks, more specifically, data preprocessing, clustering, classification, regression, visualization, and feature selection.

- **SAS / SAS Enterprise Miner**

SAS (pronounced "sass", originally Statistical Analysis System) is an integrated system of software products provided by SAS Institute Inc., which enables programmers to perform:

- Retrieval, data management, and data mining
- report writing and graphics
- statistical analysis
- business planning, forecasting, and decision support
- operations research and project management
- quality improvement
- applications development
- data warehousing (extract, transform, load)
- platform independent and remote computing

In addition, SAS has many business solutions that enable large-scale software solutions for areas such as IT management, human resource management, financial management, business intelligence, customer relationship management.

- **MATLAB**

The software tool used in proposed work is MATLAB (Matrix Laboratory). It is a numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and
interfacing with programs written in other languages, including C, C++, Java, and Fortran.

An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems. MATLAB users come from various backgrounds of engineering, science, and economics. MATLAB is widely used in academic and research institutions as well as industrial enterprises.

MATLAB provides an intuitive language and a flexible environment for technical computations which integrates mathematical computing and visualization tools for data analysis and development of algorithms and applications. This program system currently features more than 600 mathematical, statistical, and engineering functions. Its open architecture and companion products allow users to explore, develop, share and modify data sets, algorithms and custom tools, thereby achieving a fairly fast updating, improving and expanding of the MATLAB environment.

MATLAB has excellent built-in support for many data analysis and visualization routines: in particular, one of its most useful facilities is efficient exploratory data analysis, which is a natural fit in the context of data mining [8]. MATLAB was first adopted by researchers and practitioners in control engineering. It is now also used in education, in particular the teaching of linear algebra and numerical analysis, and is popular amongst scientists involved in image processing. The MATLAB application is built around the MATLAB language, and the most use of MATLAB involves typing MATLAB code into the Command Window (as an interactive mathematical shell), or executing text files containing MATLAB code and functions.

MATLAB already supports various implementations of different stages of the data mining process, including various toolboxes created by experts in the field. An initial
conclusion of this study is that MATLAB is a powerful and versatile package for fulfilling the requirements of the data mining process [8].

MATLAB Features

- Simple programming environment including control structures like loops and selections.
- Linear algebraic system solvers for vectors, matrices and functionality like computing products and inverses, scaling, summing, multiplications, factorizations, and so on.
- Simple graphic interface for visualization and simulation. Open component system based on toolboxes which can be created, modified, customized and shared by the users.
- High-level language for technical computing Development environment for managing code, files, and data Interactive tools for iterative exploration, design, and problem solving Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, and numerical integration 2-D and 3-D graphics functions for visualizing data Tools for building custom graphical user interfaces Functions for integrating MATLAB based algorithms with external applications and languages, such as C, C++, Fortran, Java, COM, and Microsoft Excel Development Environment.
- Ability to zip and unzip files and folders in the Current Folder browser to simplify sharing of files.
- New visual cues in the Current Folder browser to show directories on the MATLAB path.
- Enhanced tab completion in the MATLAB Editor with support for local variables, sub functions, and nested functions.
• Expanded access in the plot selector to plots from the Curve Fitting, Filter Design, Image Processing, and Signal Processing Toolboxes.

• Enhanced File and Folder Comparison Tool, highlighting changes within lines in file comparisons, and sorting results by name, type, size, or timestamp in folder comparisons Performance and Large Data Set Handling.

• Multithreaded computation support for integer conversion, and integer arithmetic functions.

• Performance improvements for mrdivide, convn, histc, sortrows, and sparse matrix indexed assignment.

Syntax

MATLAB application is built around MATLAB language. The simplest way to execute MATLAB code is to type it in the Command Window, which is one of the elements of MATLAB Desktop. When code is entered in the Command Window, MATLAB can be used as an interactive mathematical shell. Sequences of commands can be saved in a text file, typically using MATLAB Editor, as a script or encapsulated into a function, extending the commands available.

Unlike many other languages, where the semicolon is used to terminate commands, in MATLAB, the semicolon serves to suppress the output of the line that it concludes (it serves a similar purpose in Mathematica).

Advantages of MATLAB

• Entrance time to start using MATLAB is low. After only a few hours of training, a new MATLAB user can start developing simulation tools.
Recent versions of MATLAB compiler can compile to C, C++ and binary code, allowing the use of different optimization options for high-speed executables.

As a consequence, performance of MATLAB linear-algebraic solvers has been drastically improved in the latest versions of the software, which include also additional accelerators for operations with vector/matrix data types.

The open architecture allows for very rapid extension of the range of functionality of MATLAB by developing and sharing new toolboxes.

Using MATLAB tools and toolboxes, it is possible to develop a prototype of an application for a relatively very short time.

 Liability

ARMADA

ARMADA is a MATLAB implemented program with a graphical user interface (GUI). ARMADA continues with processes including noise filtering, spot background correction, data normalization, statistical selection of differentially expressed genes based on parametric or non parametric statistics, cluster or classification analysis based on several widely used clustering methods (Hierarchical, k-means, Fuzzy C-means) or statistical learning algorithms for classification (Discriminant Analysis, k-Nearest Neighbors, Support Vector Machines) and annotation steps, resulting in detailed lists of differentially expressed genes and formed clusters.

In ARMADA tool (Association Rule Mining And Deduction Analysis) the rules are presented in the form of Item1, ItemN => ItemX, The rule is evaluated using two numerical measures.
SUPPORT: The number of occurrences, within the data set, of all components of an association rule occurring within a single transaction (a transaction can be thought of as a row in a conventional database).

CONFIDENCE: This is the probability that, given the antecedent (left hand side of the rule) being true, the consequent (right hand side) is true.

ARMADA contains several visualization techniques for analyzing quantitative and qualitative measures of each rule, as well as several strategies for undertaking data mining which can be used as exclusive or combinatorial methods. The ARMADA output files can be easily imported in a spreadsheet like software such as MS Excel or in a database for further processing and storage and the analysis results can be saved as .mat files for further possible processing with MATLAB’s built-in algorithms.

Depending on the user’s programming experience and analysis preferences, ARMADA can be used to perform analyses step by step through the GUI of the system or as an automated analysis pipeline (by using the batch programming module). For the most experienced user, ARMADA can also be invoked directly from MATLAB’s command window, as the main routines that perform the analysis behind the GUI are designed to run also individually in command line mode with specific arguments (the user should see help inside .m files to perform command line analysis).

ARMADA is a completely open source MATLAB based platform and the user may alter, adjust or extend each of the main functions or create new routines according to specific needs. It should be noted that ARMADA can be used in command line mode only if MATLAB is present on the computer where ARMADA is installed. Otherwise, the program is distributed as a standalone application with MATLAB
Component Runtime (MCR) and MATLAB is not required on the installation machine.

2.12. Summary

In this chapter discusses about the basic concepts of Data mining. It contains the different techniques, tools, methods and application areas in data mining. Also presents the overview of related work and background knowledge used in this thesis to cover specific target and analysis of pharmaceutical compounds.