Introduction to Data Mining functionalities, Mining different kind of data, Pattern/Context based Data Mining, Bayesian Classification: Bayes theorem, Bayesian belief networks Naive Bayesian classification, Introduction to classification by Back propagation and its algorithm, Other classification methods: k-Nearest Neighbor, case based reasoning, Genetic algorithms, rough set approach, Fuzzy set approach

Data mining refers to extracting or mining knowledge from large amountsof data. The term is actually a misnomer. Thus, data mining should have been more appropriately named as knowledge mining which emphasis on mining from large amounts of data. It is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. 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. The key properties of data mining are
Automatic discovery of patterns

Prediction of likely outcomes

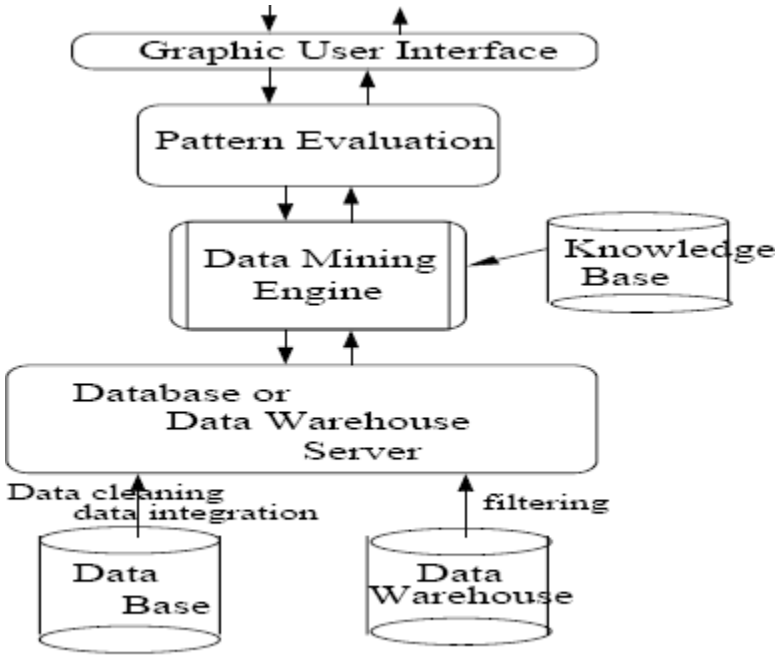
Creation of actionable information

Focus on large datasets and databases

Architecture of a typical data mining system/Major Components

Data mining is the process of discovering interesting knowledge from large amounts of data stored either in databases, data warehouses, or other information repositories. Based on this view, the architecture of a typical data mining system may have the following major components:

1. A database, data warehouse, or other information repository, which consists of the set of databases, data warehouses, spreadsheets, or other kinds of information repositories containing the student and course information.
2. A database or data warehouse server which fetches the relevant data based on users‟ data mining requests.
3. A knowledge base that contains the domain knowledge used to guide the search or to evaluate the interestingness of resulting patterns. For example, the knowledge base may contain metadata which describes data from multiple heterogeneous sources.
4. A data mining engine, which consists of a set of functional modules for tasks such as classification, association, classification, cluster analysis, and evolution and deviation analysis.
5. A pattern evaluation module that works in tandem with the data mining modules by employing interestingness measures to help focus the search towards interestingness patterns.
6. A graphical user interface that allows the user an interactive approach to the data mining system.



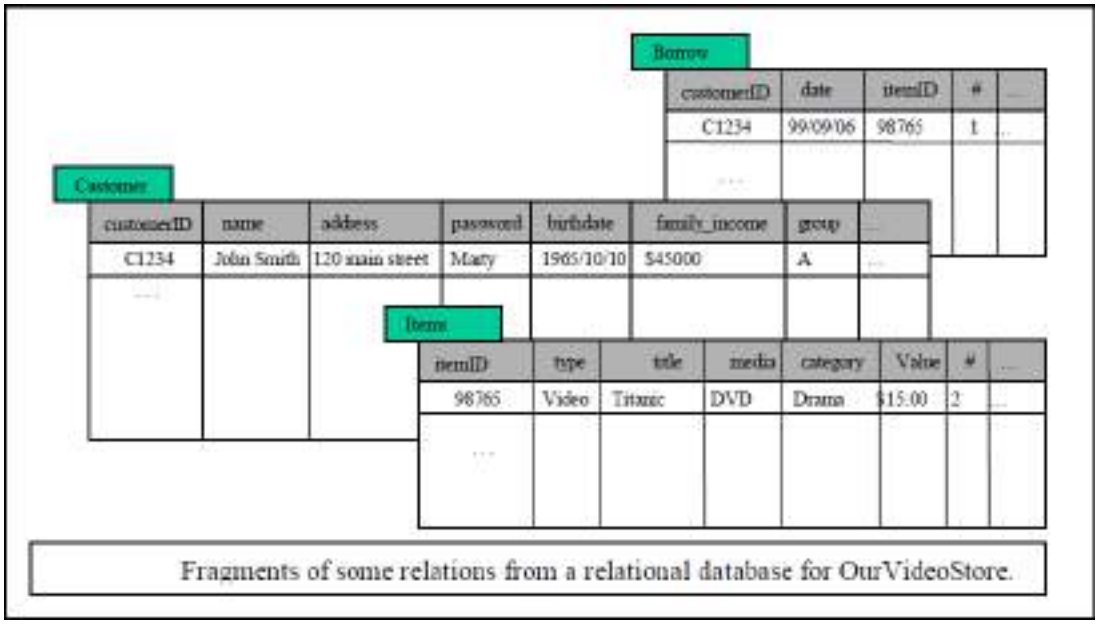
Architecture of a typical data mining system.

In principle, data mining should be applicable to any kind of information repository. This includes relational databases, data warehouses, transactional databases, advanced database systems,

flat files, and the World-Wide Web. Advanced database systems include object-oriented and object- relational databases, and special c application-oriented databases, such as spatial databases, time-series databases, text databases, and multimedia databases.

Flat files: Flat files are actually the most common data source for data mining algorithms, especially at the research level. Flat files are simple data files in text or binary format with a structure known by the data mining algorithm to be applied. The data in these files can be transactions, time-series data, scientific measurements, etc.

Relational Databases: a relational database consists of a set of tables containing either values of entity attributes, or values of attributes from entity relationships. Tables have columns and rows, where columns represent attributes and rows represent tuples. A tuple in a relational table corresponds to either an object or a relationship between objects and is identified by a set of attribute values representing a unique key. In following figure it presents some relations Customer, Items, and Borrow representing business activity in a video store. These relations are just a subset of what could be a database for the video store and is given as an example.



The most commonly used query language for relational database is SQL, which allows retrieval and manipulation of the data stored in the tables, as well as the calculation of aggregate functions such as average, sum, min, max and count. For instance, an SQL query to select the videos grouped by category would be:

SELECT count(*) FROM Items WHERE type=video GROUP BY category.

Data mining algorithms using relational databases can be more versatile than data mining algorithms specifically written for flat files, since they can take advantage of the structure inherent to relational databases. While data mining can benefit from SQL for data selection, transformation and consolidation, it goes beyond what SQL could provide, such as predicting, comparing, detecting deviations, etc.

Contrast pattern based data mining is concerned with the mining of patterns and models that contrast two or more datasets. Contrast patterns can describe similarities or differences between the datasets. They represent strong contrast knowledge and have been shown to be very successful for constructing accurate and robust clusters and classifiers. The increasing use of contrast pattern data mining has initiated a great deal of research and development attempts in the field of data mining. A comprehensive revision on the existing contrast pattern based data mining research is given in this paper. They are generally categorized into background and representation, definitions and mining algorithms, contrast pattern based classification, clustering, and other applications, the research trends in future

Bayesian classification is based on Bayes' Theorem. Bayesian classifiers are the statistical classifiers. Bayesian classifiers can predict class membership probabilities such as the probability that a given tuple belongs to a particular class.

Baye's Theorem

Bayes' Theorem is named after Thomas Bayes. There are two types of probabilities −

- Posterior Probability [P(H/X)]
- Prior Probability [P(H)]

where X is data tuple and H is some hypothesis.

According to Bayes' Theorem,

$$P(H/X)= P(X/H)P(H) / P(X)$$

Bayesian Belief Network

Bayesian Belief Networks specify joint conditional probability distributions. They are also known as Belief Networks, Bayesian Networks, or Probabilistic Networks.

- A Belief Network allows class conditional independencies to be defined between subsets of variables.
- It provides a graphical model of causal relationship on which learning can be performed.
- We can use a trained Bayesian Network for classification.

There are two components that define a Bayesian Belief Network −

- Directed acyclic graph
- A set of conditional probability tables

Directed Acyclic Graph

- Each node in a directed acyclic graph represents a random variable.
- These variable may be discrete or continuous valued.
- These variables may correspond to the actual attribute given in the data.

Naive Bayes classifiers are a collection of classification algorithms based on **Bayes' Theorem**. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.To start with, let us consider a dataset.

Consider a fictional dataset that describes the weather conditions for playing a game of golf. Given the weather conditions, each tuple classifies the conditions as fit("Yes") or unfit("No") for plaing golf.

The dataset is divided into two parts, namely, **feature matrix** and the **response vector**.

- Feature matrix contains all the vectors(rows) of dataset in which each vector consists of the value of **dependent features**. In above dataset, features are 'Outlook', 'Temperature', 'Humidity' and 'Windy'.
- Response vector contains the value of **class variable**(prediction or output) for each row of feature matrix. In above dataset, the class variable name is 'Play golf'.

Assumption:

The fundamental Naive Bayes assumption is that each feature makes an:

- independent
- equal

contribution to the outcome.

With relation to our dataset, this concept can be understood as:

- We assume that no pair of features are dependent. For example, the temperature being 'Hot' has nothing to do with the humidity or the outlook being 'Rainy' has no effect on the winds. Hence, the features are assumed to be **independent**.
- Secondly, each feature is given the same weight(or importance). For example, knowing only temperature and humidity alone can't predict the outcome accuratey. None of the attributes is irrelevant and assumed to be contributing **equally** to the outcome.

Backpropagation (backward propagation) is an important mathematical tool for improving the accuracy of predictions in data mining and machine learning. Essentially, backpropagation is an algorithm used to calculate derivatives quickly. Artificial neural networks use backpropagation as a learning algorithm to compute a gradient descent with respect to weights. Desired outputs are compared to achieved system outputs, and then the systems are tuned by adjusting connection weights to narrow the difference between the two as much as possible. The algorithm gets its name because the weights are updated backwards, from output towards input.

The difficulty of understanding exactly how changing weights and biases affects the overall behavior of an artificial neural network was one factor that held back wider application of neural network applications, arguably until the early 2000s when computers provided the necessary insight. Today, backpropagation algorithms have practical applications in many areas

of artificial intelligence (AI), including optical character recognition (OCR), natural language processing (NLP) and image processing.

Because backpropagation requires a known, desired output for each input value in order to calculate the loss function gradient, it is usually classified as a type of supervised machine learning. Along with classifiers such as Naïve Bayesian filters and decision trees, the backpropagation algorithm has emerged as an important part of machine learning applications that involve predictive analytics.

Properties of the Backpropagation Algo.

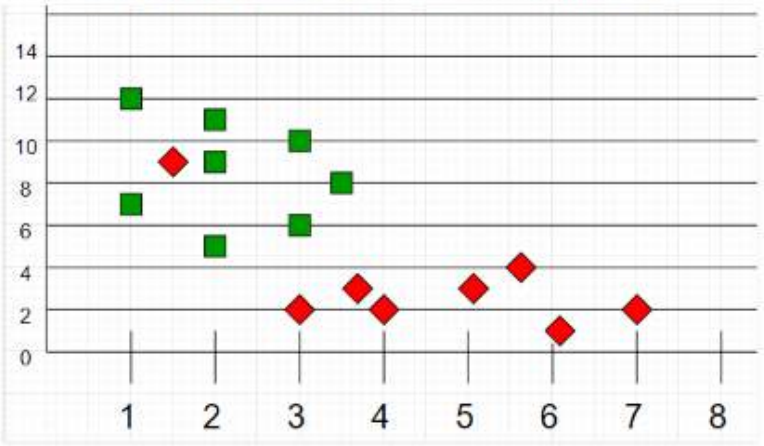
- Learns weights for a multilayer network, given a fixed set of units and interconnections
- It uses gradient descent to minimize the squashed error between the network outputs and the target values for these outputs.
-
- is the set of output units in the network, and are the target and output values associated with the th output unit and the training example
- In multilayer networks the error surface can have multiple minima, but in practice Backpropagation has produced excellent results in many real-world applications
- the algorithm is for two layers of sigmoid units and does stochastic gradient descent

K-Nearest Neighbours

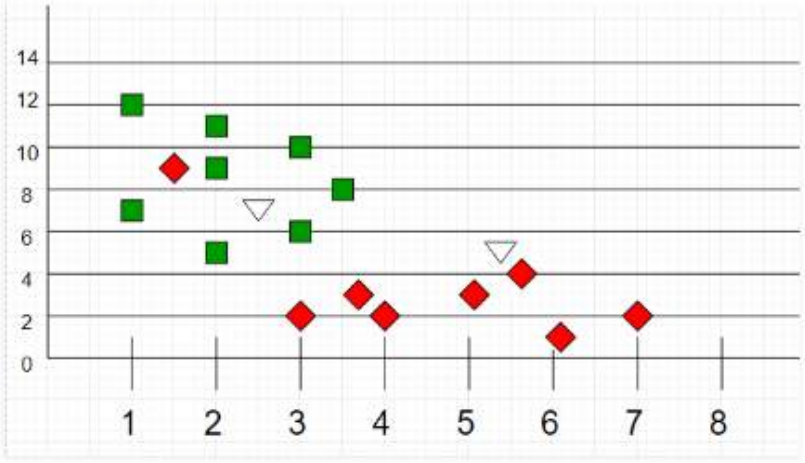
K-Nearest Neighbors is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection.

It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data).
We are given some prior data (also called training data), which classifies coordinates into groups identified by an attribute.

As an example, consider the following table of data points containing two features:



Now, given another set of data points (also called testing data), allocate these points a group by analyzing the training set. Note that the unclassified points are marked as 'White'.



Intuition

If we plot these points on a graph, we may be able to locate some clusters or groups. Now, given an unclassified point, we can assign it to a group by observing what group its nearest neighbors belong to. This means a point close to a cluster of points classified as 'Red' has a higher probability of getting classified as 'Red'.

Intuitively, we can see that the first point (2.5, 7) should be classified as 'Green' and the second point (5.5, 4.5) should be classified as 'Red'.

Algorithm

Let m be the number of training data samples. Let p be an unknown point.

1. Store the training samples in an array of data points arr[]. This means each element of this array represents a tuple (x, y).
2. for i=0 to m:
3. Calculate Euclidean distance d(arr[i], p).
4. Make set S of K smallest distances obtained. Each of these distances corresponds to an already classified data point.
5. Return the majority label among S.

Case-based reasoning (**CBR**), broadly construed, is the process of solving new problems based on the solutions of similar past problems. An auto mechanic who fixes an engine by recalling another car that exhibited similar symptoms is using case-based reasoning. A lawyer who advocates a particular outcome in a trial based on legal precedents or a judge who creates case law is using case-based reasoning. So, too, an engineer copying working elements of nature (practicing biomimicry), is treating nature as a database of solutions to problems. Case-based reasoning is a prominent type of analogy solution making.

Case-based reasoning has been formalized for purposes of computer reasoning as a four-step process:[1]

1. **Retrieve:** Given a target problem, retrieve from memory cases relevant to solving it. A case consists of a problem, its solution, and, typically, annotations about how the solution was derived. For example, suppose Fred wants to prepare blueberry pancakes. Being a novice cook, the most relevant experience he can recall is one in which he successfully made plain pancakes. The procedure he followed for making the plain pancakes, together with justifications for decisions made along the way, constitutes Fred's retrieved case.
2. **Reuse:** Map the solution from the previous case to the target problem. This may involve adapting the solution as needed to fit the new situation. In the pancake example, Fred must adapt his retrieved solution to include the addition of blueberries.
3. **Revise:** Having mapped the previous solution to the target situation, test the new solution in the real world (or a simulation) and, if necessary, revise. Suppose Fred adapted his pancake solution by adding blueberries to the batter. After mixing, he discovers that the batter has turned blue – an undesired effect. This suggests the following revision: delay the addition of blueberries until after the batter has been ladled into the pan.
4. **Retain:** After the solution has been successfully adapted to the target problem, store the resulting experience as a new case in memory. Fred, accordingly, records his new-found procedure for making blueberry pancakes, thereby enriching his set of stored experiences, and better preparing him for future pancake-making demands

Genetic Algorithms

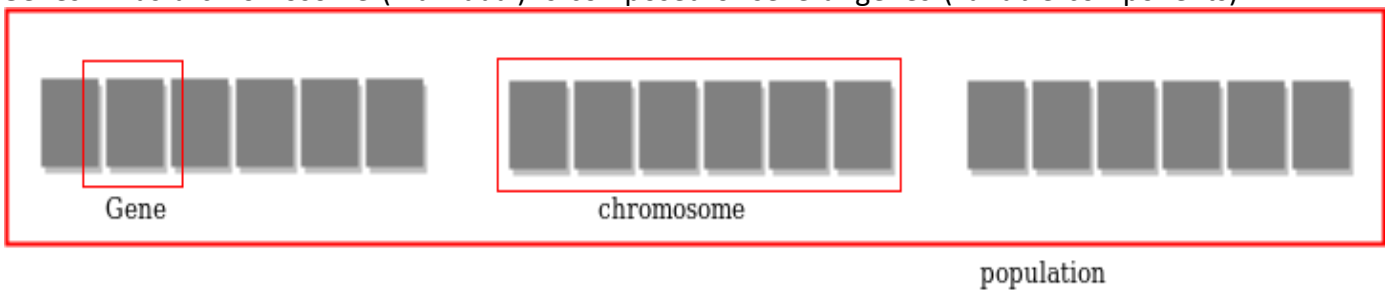
Genetic Algorithms(GAs) are adaptive heuristic search algorithms that belong to the larger part of evolutionary algorithms. Genetic algorithms are based on the ideas of natural selection and genetics. These are intelligent exploitation of random search provided with historical data to direct the search into the region of better performance in solution space. **They are commonly used to generate high-quality solutions for optimization problems and search problems.**
Genetic algorithms simulate the process of natural selection which means those species who can adapt to changes in their environment are able to survive and reproduce and go to next generation. In simple words, they simulate “survival of the fittest” among individual of consecutive generation for solving a problem. **Each generation consist of a population of individuals** and each individual represents a point in search space and possible solution. Each individual is represented as a string of character/integer/float/bits. This string is analogous to the Chromosome.

Genetic algorithms are based on an analogy with genetic structure and behavior of chromosome of the population. Following is the foundation of GAs based on this analogy –

1. Individual in population compete for resources and mate
2. Those individuals who are successful (fittest) then mate to create more offspring than others
3. Genes from "fittest" parent propagate throughout the generation, that is sometimes parents create offspring which is better than either parent.
4. Thus each successive generation is more suited for their environment.

Search space

The population of individuals are maintained within search space. Each individual represent a solution in search space for given problem. Each individual is coded as a finite length vector (analogous to chromosome) of components. These variable components are analogous to Genes. Thus a chromosome (individual) is composed of several genes (variable components).



Fitness Score

A Fitness Score is given to each individual which **shows the ability of an individual to "compete"**. The individual having optimal fitness score (or near optimal) are sought.

The GAs maintains the population of n individuals (chromosome/solutions) along with their fitness scores.The individuals having better fitness scores are given more chance to reproduce than others. The individuals with better fitness scores are selected who mate and produce **better offspring** by combining chromosomes of parents. The population size is static so the room has to be created for new arrivals. So, some individuals die and get replaced by new arrivals eventually creating new generation when all the mating opportunity of the old population is exhausted. It is hoped that over successive generations better solutions will arrive while least fit die.

Each new generation has on average more "better genes" than the individual (solution) of previous generations. Thus each new generations have better **"partial solutions"** than previous generations. Once the offsprings produced having no significant difference than offspring produced by previous populations, the population is converged. The algorithm is said to be converged to a set of solutions for the problem.

Operators of Genetic Algorithms

Once the initial generation is created, the algorithm evolve the generation using following operators –

1) Selection Operator: The idea is to give preference to the individuals with good fitness scores and allow them to pass there genes to the successive generations.

2) Crossover Operator: This represents mating between individuals. Two individuals are selected using selection operator and crossover sites are chosen randomly. Then the genes at these crossover sites are exchanged thus creating a completely new individual (offspring).

2) **Mutation Operator:** The key idea is to insert random genes in offspring to maintain the diversity in population to avoid the premature convergence

Rough Set Theory | An Introduction

The notion of ***Rough sets*** was introduced by Z Pawlak in his seminal paper of 1982 (Pawlak 1982). It is a formal theory derived from fundamental research on logical properties of information systems. Rough set theory has been a methodology of database mining or knowledge discovery in relational databases. In its abstract form, it is a new area of uncertainty mathematics closely related to fuzzy theory. We can use rough set approach to discover structural relationship within imprecise and noisy data.
Rough sets and fuzzy sets are complementary generalizations of classical sets. The approximation spaces of rough set theory are sets with multiple memberships, while fuzzy sets are concerned with partial memberships. The rapid development of these two approaches provides a basis for "soft computing, " initiated by Lotfi A. Zadeh. Soft Computing includes along with rough sets, at least fuzzy logic, neural networks, probabilistic reasoning, belief networks, machine learning, evolutionary computing, and chaos theory.

Basic problems in data analysis solved by Rough Set:

- Characterization of a set of objects in terms of attribute values.
- Finding dependency between the attributes.
- Reduction of superfluous attributes.
- Finding the most significant attributes.
- Decision rule generation.

Goals of Rough Set Theory –

- The main goal of the rough set analysis is the induction of (learning) approximations of concepts. Rough sets constitute a sound basis for KDD. It offers mathematical tools to discover patterns hidden in data.
- It can be used for feature selection, feature extraction, data reduction, decision rule generation, and pattern extraction (templates, association rules) etc.
- Identifies partial or total dependencies in data, eliminates redundant data, gives approach to null values, missing data, dynamic data and others.

Information system –

In Rough Set, data model information is stored in a table. Each row (tuples) represents a fact or an object. Often the facts are not consistent with each other. In Rough Set terminology a data table is called an Information System.
Thus, the information table represents input data, gathered from any domain.

Indiscernibility –

Tables may contain many objects having the same features.A way of reducing table size is to store only one representative object for every set of objects with same features.These objects are called indiscernible objects or tuples.

With any P subset A there is an associated equivalence relation IND(P):

$$\mathrm{IND}(P) = \{\langle x, y\rangle \in \mathbb{U}^2 \mid \forall a \in P, a(x) = a(y)\}$$

pproximations –

It is a formal approximation of a crisp set defined by its two approximations – ***Upper approximation*** and ***Lower approximation***.

- **Upper approximation** is the set of objects which possibly belong to the target set.

  Upper Approximation:

  $$\overline{R}X = \bigcup\{Y \in U/R : Y \cap X \neq \phi\}$$

- **Lower approximation** is the set of objects that positively belong to the target set.

  Lower Approximation:

  $$\underline{R}X = \bigcup\{Y \in U/R : Y \subseteq X\}$$

Genetic Algorithms

The idea of genetic algorithm is derived from natural evolution. In genetic algorithm, first of all, the initial population is created. This initial population consists of randomly generated rules. We can represent each rule by a string of bits.

For example, in a given training set, the samples are described by two Boolean attributes such as A1 and A2. And this given training set contains two classes such as C1 and C2.

We can encode the rule **IF A1 AND NOT A2 THEN C2** into a bit string **100**. In this bit representation, the two leftmost bits represent the attribute A1 and A2, respectively.

Likewise, the rule **IF NOT A1 AND NOT A2 THEN C1** can be encoded as **001**.

Note − If the attribute has K values where K>2, then we can use the K bits to encode the attribute values. The classes are also encoded in the same manner.

Points to remember −

- Based on the notion of the survival of the fittest, a new population is formed that consists of the fittest rules in the current population and offspring values of these rules as well.
- The fitness of a rule is assessed by its classification accuracy on a set of training samples.
- The genetic operators such as crossover and mutation are applied to create offspring.
- In crossover, the substring from pair of rules are swapped to form a new pair of rules.
- In mutation, randomly selected bits in a rule's string are inverted.

Rough Set Approach

We can use the rough set approach to discover structural relationship within imprecise and noisy data.

Note − This approach can only be applied on discrete-valued attributes. Therefore, continuous-valued attributes must be discretized before its use.

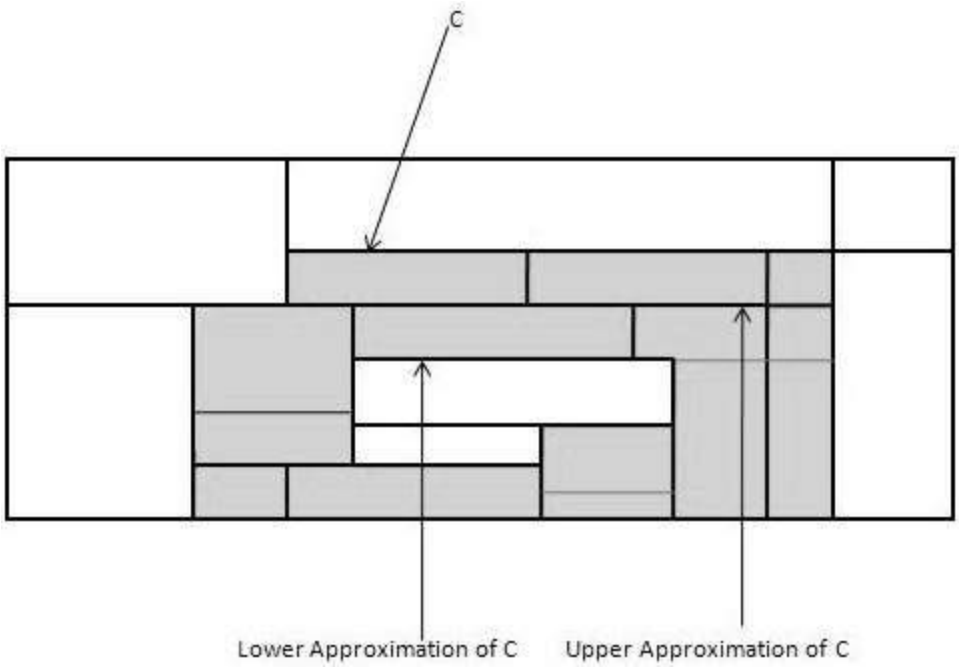
The Rough Set Theory is based on the establishment of equivalence classes within the given training data. The tuples that forms the equivalence class are indiscernible. It means the samples are identical with respect to the attributes describing the data.

There are some classes in the given real world data, which cannot be distinguished in terms of available attributes. We can use the rough sets to **roughly** define such classes.

For a given class C, the rough set definition is approximated by two sets as follows −

- **Lower Approximation of C** − The lower approximation of C consists of all the data tuples, that based on the knowledge of the attribute, are certain to belong to class C.
- **Upper Approximation of C** − The upper approximation of C consists of all the tuples, that based on the knowledge of attributes, cannot be described as not belonging to C.

The following diagram shows the Upper and Lower Approximation of class C −



Fuzzy Set Approaches

Fuzzy Set Theory is also called Possibility Theory. This theory was proposed by Lotfi Zadeh in 1965 as an alternative the **two-value logic** and **probability theory**. This

theory allows us to work at a high level of abstraction. It also provides us the means for dealing with imprecise measurement of data.

The fuzzy set theory also allows us to deal with vague or inexact facts. For example, being a member of a set of high incomes is in exact (e.g. if $50,000 is high then what about $49,000 and $48,000). Unlike the traditional CRISP set where the element either belong to S or its complement but in fuzzy set theory the element can belong to more than one fuzzy set.

For example, the income value $49,000 belongs to both the medium and high fuzzy sets but to differing degrees. Fuzzy set notation for this income value is as follows −

$m_{medium\_income}(\$49k)=0.15$ and $m_{high\_income}(\$49k)=0.96$

where 'm' is the membership function that operates on the fuzzy sets of medium_income and high_income respectively. This notation can be shown diagrammatically as follows −