A Global Rule Pruning for ACO in Classification Task of Data Mining

Major project Submitted in partial fulfillment of the requirements

For the award of degree of

Master of Technology
In
Information Technology

Submitted By
MAHESHWAR
(Roll No. 2K11/ISY/14)

Under the guidance of
Mr. RAHUL KATARYA
(Assistant Professor)
Department of Information Technology

Department of Information Technology

Delhi Technological University

Delhi

Session 2011-2013

CERTIFICATE

This is to certify that **Mr. Maheshwar (2k11/ISY/14)** has carried out the major project titled **"A Global Rule Pruning for ACO in Classification Task of Data Mining"** as a partial requirement for the award of Master of Technology degree in Information Systems by Delhi Technological University.

The major project is a bonafide piece of work carried out and completed under my supervision and guidance during the academic session **2012-2013.** The matter contained in this report has not been submitted elsewhere for the award of any other degree.

(Project Guide)

Mr. Rahul Katarya
(Assistant Professor)
Department of Information Technology
Delhi Technological University

Bawana Road, Delhi-110042

ACKNOWLEDGEMENTS

I take this opportunity to express my sincere gratitude towards **Assistant Professor Rahul Katarya** for his constant support and encouragement. His excellent guidance has been instrumental in making this project work a success. I would like to thank faculty members of the Department of Information Technology at Delhi Technological University for their valuable time and suggestions when required. I would like to thank my family and friends also, who have been a source of encouragement and inspiration throughout the duration of the project. I would like to thank the entire DTU family for making my stay at DTU a memorable one.

Maheshwar
Roll No: 2k11/ISY/14
M.Tech (Information Systems)
E-mail: maheshwar1524@gmail.com

Abstract

Ant colony optimization algorithms have been used to extract classification rules in data mining. Different variants of ACO have been proposed and successfully applied to discover a list of classification rules. But most of these algorithms follow sequential covering of the algorithm where a rule is pruned without taking into consideration its effects on the whole list of the rules i.e. each rule is pruned individually. Moreover, the relation between the different terms in the rule generated is not taken into consideration. We have proposed a new rule pruning procedure that takes into account the association between the terms in a generated rule during pruning.

Keywords: Ant colony optimization (ACO), Data mining, Classification, rule pruning.

Table of Contents

LIST OF FIGURES

CHAPTER 1

DATA MINING AND KDD

1.1 Overview

As the days goes on, the amount of data is also increasing. Data in every field whether it is of private sector, government sector, school, colleges etc. is growing day by day. So, it becomes tedious task to pick up right and accurate amount of data in order to make certain decision and calculation. This is the point where data mining comes into play. This chapter discussed the data mining process briefly. Section 1.2 gives overview of data mining and then discussed the tools and task of data mining process. Section 1.3 details knowledge discovery process and section 1.4 gives thesis overview.

1.2 Data Mining

Data mining analogous to Knowledge Discovery in Databases (KDD) is the process of extracting data patterns and useful information from a large data set and analyzing it from different perspective. Data mining encompasses different tasks and techniques that help in extracting knowledge that is accurate and more comprehensible to users [1], [2]. In this section we will take a look at data mining system, various data mining tasks and tools.

1.2.1 Data Mining System

A typical data mining system has following major components [3].

- **Database, data warehouse, World Wide Web, or other information repository:** This component involves databases, spreadsheets, and data from World Wide Web and from data warehouses. Important techniques at the component are cleaning the data and integration
- **Database or data warehouse server:** This component fetches the relevant data on the basis of request submitted by the user.
- **Knowledge base:** This component stores information that helps in searching and evaluating the interesting pattern in the large data. Concept hierarchies are used in order to organize attributes and organize the data at different abstraction levels.

- **Data mining engine:** This component is core to the data mining system and involves tasks like classification, clustering characterization, prediction, cluster analysis, outlier analysis, association and correlation analysis and evolution analysis.

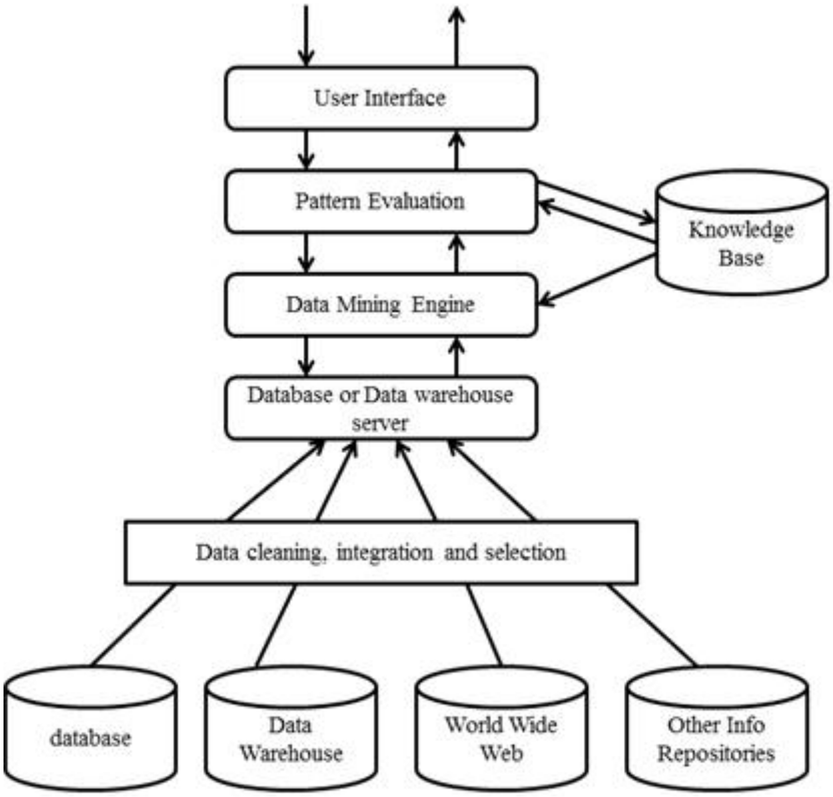

Figure 1.1: Data mining system

- **Pattern evaluation modules:** This component has measures for interestingness. It interacts with other data mining modules to provide a direction to search. Thresholds are fixed to filter out the discovered patterns.
- **User interface:** This is the component from where user actually accesses the data mining system and performs different tasks on data. This component gives all information to user so as to get better search results and patterns and check their interestingness. Figure 1 shows the typical architecture of a data mining system [3].

1.2.2 Data Mining Tasks

In this part of Data Mining, goals are decided dependent on the anticipated results from the overall process of KDD. Several methods/techniques with different goals can be applied simultaneously or sequentially to achieve a desired outcome. Agrawal et al. in [4], [5] introduces different classes of data mining problem. Most Data Mining tasks come under the following categories:

- ***Data Processing***: KDD may process, analyze select, filter, aggregate, sample, clean, and/or transform data depending on the goals and necessities.
- ***Classification:*** Classification analyzes a set of data whose class label is known and constructs a model for each class based on the features in the data. For example, classification can be done on diseases and help to predict the kind of diseases based on the symptoms of patients. In the classification task, each tuple belongs to a class in a predefined set of classes.
- ***Association Rules:*** Association rules helps in discovering frequent pattern in dataset so as to analyze the data from different perspective and to make certain decisions. Association rule determines how closely the terms in the rule are.
- ***Prediction:*** Prediction helps in determining the certain values of a data or attribute value that is missing from the dataset. In this method, relevancy among the attribute is calculated and on the basis of that prediction is made about those missing values. Different techniques like regression analysis, correlation analysis and decision trees have been used for this task. Evolutionary algorithms like genetic algorithms have also been used for prediction.
- ***Clustering:*** Clustering involves the task of grouping data that is similar. Similarity can be measured using different approaches. The most common used is distance based. Clustering is said to be good one if the inter group similarity is low and intra group similarity is high among the data.
- ***Regression:*** It is a function that helps in mapping a data item to a real-valued variable. Regression analyzes the dependency among the different attributes values i.e. values of one attribute can be dependent on values of another attribute in the same item set.
- ***Model Visualization:*** It is the visual interpretation of complex relationships in multidimensional data Model. Visualization makes the discovered knowledge understandable and interpretable by humans. Visualization model can range from scatter plots, histograms, 3-dimensional plots, and even animated graphics.
- ***Exploratory Data Analysis:*** Exploratory data analysis is the interactive exploration of a data set without dependence on previously conceived assumptions and models, thus trying to

identify interesting patterns often with visual methods. Statistical model can be used to analyze the dataset.

- **Bayesian Belief Networks (BBN):** Bayesian belief networks use the concept of Bayes probability theorem. These networks are acyclic graphs where attributes are represented in form of nodes and the dependency between two nodes i.e. attributes is represented by the edge. Concept of conditional probability is used to find relation between the child node and parent node.
- **Case Based Learning:** An approach to the design of learning algorithms that is inspired by the fact that when people encounter new situations they often explain them by reference to familiar experiences, adapting the explanations to fit the new situation. It is closely related to explanation-based learning.
- **Decision Trees:** Decision tree based algorithms are most useful approaches in classification problems and comes under eager learner category i.e. when a training case is given, it will construct a generalization model before receiving new test case to classify. In this method of classification, a tree is constructed to model the classification process. Once the tree is built, it is applied to each test case that is to be classified.
- **Rule Induction:** Rule based algorithms are straightforward way to perform classification. Rules are the best way to represent the information and are easily understandable by human i.e. have high comprehensibility [3]. Rule based algorithms have a set of IF-THEN rules to classify the data.
- **Neural Networks:** Neural networks are also used for making systems that can help in various data mining tasks like clustering, classification etc. Neural networks are inspired by the structure of neurons in brain. It involves various input and output connected to each other with different weights. These weights are updated after every iteration of the algorithm and process is repeated till the stopping criterion is met.
- **Genetic Algorithms:** Genetic Algorithms [17] are evolutionary algorithms It is a global optimization method that manipulates a string of numbers in a manner similar to how chromosomes are changed in biological evolution.. Genetic algorithm has been discussed in chapter 2 briefly.

Figure 1.2 summarizes all the data mining tasks and techniques. It also shows that data mining is an important part of knowledge discovery in database process discussed in next section.

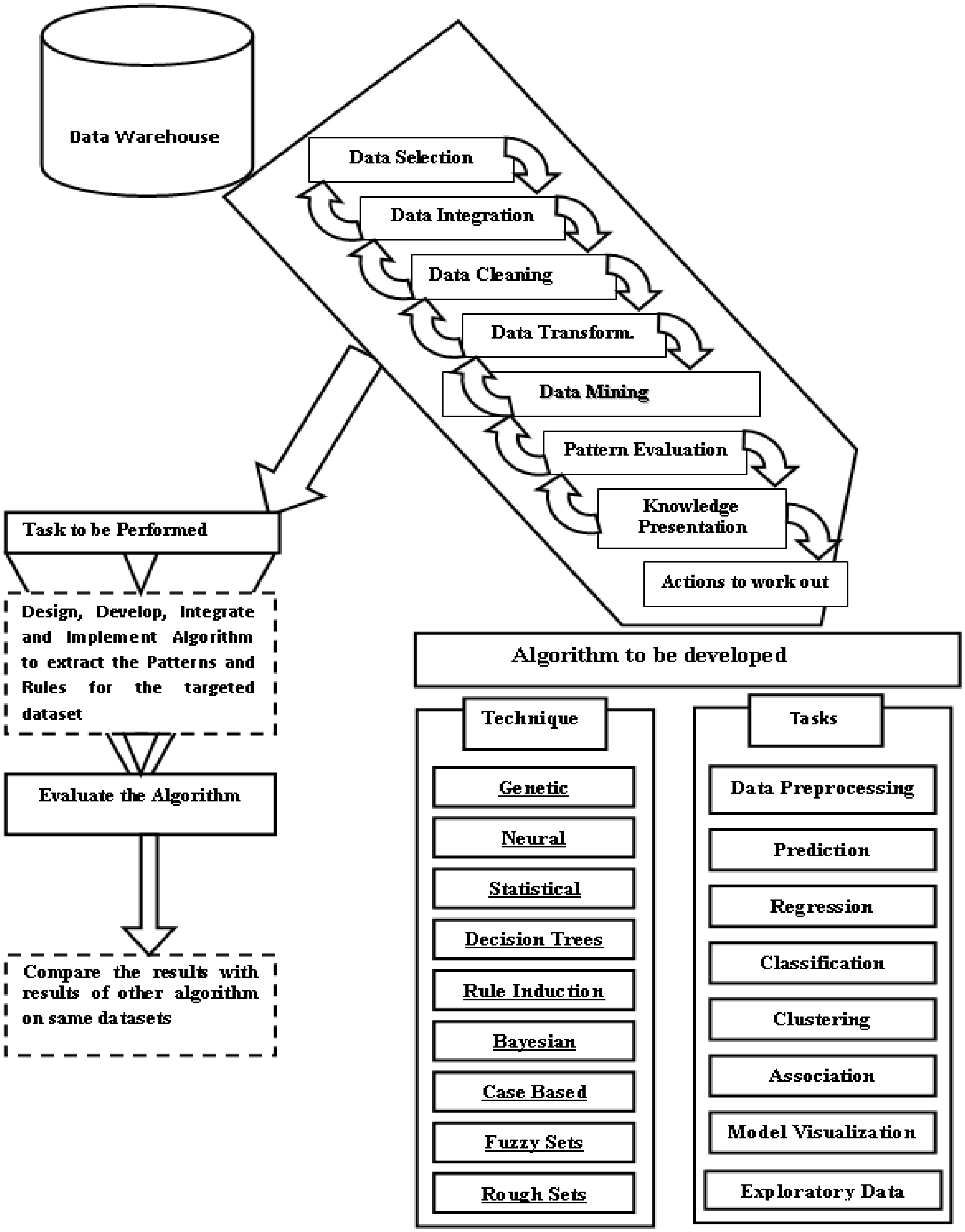

Figure 1.2: Different data mining tasks and techniques

1.2.3 Data Mining Tools

At present time, commercial data mining tools use several common techniques to extract knowledge. These include association rules, clustering, neural networks and statistical analysis. We have discussed all these in the previous sub-section. So data mining tools also use advanced techniques such as genetic algorithm, Bayesian networks, regression, pattern matching etc.

COMPANY	PRODUCT	TECHNIQUES	PLATEFORM	INTERFACE
Acknosoft	Kate	Decision trees, Cased based reasoning	Win NT	Microsoft Access
Business Object	Business Miner	Neural nets, Machine learning	Win NT	ODBC
Data Distilleries	Data Surveyor	Comprehensive, Can mix DM	UNIX	ODBC, ODBC compliant
DBMiner Technology Inc.	DBMiner	OLAP analysis, Association, Classification, Clustering algorithm	Win NT	Microsoft 7.0 OLAP MGr
IBM	Intelligent Miner	Classification, Association rules, Predictive models	UNIX (AIX)	IBM DB2
SAS	Enterprise Miner	Decision trees, Association rules, Regression, Neural nets, Clustering	UNIX (Solaris), Win NT, Macintosh	ODBC, Oracle, AS/400
Silicon Graphics	MineSet	Decision trees, Association rules	UNIX (Irix)	Oralce, Sybase, Informix

Table 1.1: Data Mining Tools

Most of the data mining tools use the ODBC (Open Database Connectivity) interface. Most of the tools work in the Microsoft Windows environment and a few works in the UNIX operating system. Table 1.1 represents the various data mining tools along with their vendors, technique used, platform and user interface.

1.3 The KDD Process

Knowledge Discovery in Databases (KDD) is process of extracting useful knowledge from volumes of data [6]. A collaborative design of KDD process in distributed environment is proposed to provide effective support to users [7]. Knowledge Discovery in Databases (KDD) and Data Mining are rapidly emerging interdisciplinary field. Data Mining as discussed above is the process of extracting meaningful information and patterns to make decisions and currently the hot area to work and. Data mining itself is a step in the knowledge discovery process as shown in figure 2 [3. A data warehouse is a repository where the data is stored. This data is non-volatile, of historic nature and subject-oriented.

Knowledge Discovery in Databases is “the non-trivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data”. Knowledge Discovery in Databases merges together database management, statistics, data visualization, clustering, machine learning, expert systems and artificial intelligence. The Knowledge Discovery in Databases process comprises of following iterative steps [3].

i. **Data Selection:** The data irrelevant data is removed at this step and data which is of our use is kept for analysis.

ii. **Data Cleaning:** Data cleaning process removes the noisy and irrelevant data from the selected data.

iii. **Data Integration:** Data from different data sources that are usually heterogeneous are grouped and integrated to from a single dataset.

iv. **Data Transformation:** At this step, data is transformed into form that is suitable for the process of mining so that appropriate technique could be applied.

v. **Data Mining:** This step as we told earlier mines information and patterns that are of user interests.

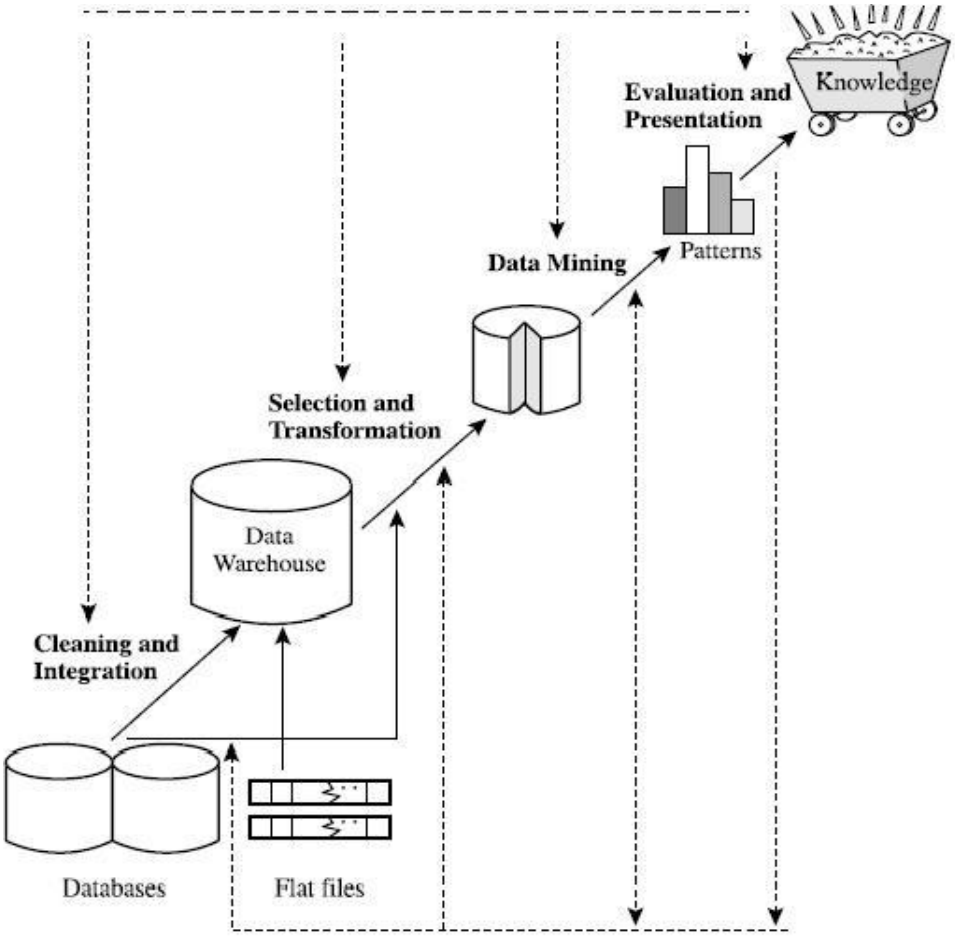

Figure 1.3: Data mining as a step in the KDD process

vi. **Pattern Evaluation:** Patterns discovered at data mining step are evaluated and certain decisions are made.

vii. **Knowledge Representation:** At this step, the knowledge discovered is represented in suitable form that can user easily analyzed. Most common way is representing knowledge in form of visual ways.

viii. **Actions:** In this step discovered knowledge is applied in real life operations and tasks. Implementation of knowledge is the ultimate goal of KDD process.

It is important to say here that Data Mining is a step in KDD process, which takes only 15% to 25% of the overall effort. KDD process has application in different areas like hospital information system (HIS) [8] and others.

1.4 Thesis Overview

After introducing the data mining, the tools and techniques, KDD process the rest of work is organized in the following six chapters.

- **Chapter2** It provides an overview to classification task of data mining and discusses the basic procedure of ant colony optimization (ACO) algorithm. After giving a brief overview of the ACO algorithm, it gives a direction to how ACO algorithm can be implemented in classification task of data mining.
- **Chapter3** provides the details of related work in this field i.e. classification using ACO in data mining. It discusses the various approaches proposed by the various authors and also reflects light on the future direction that can be followed for rules classification using ant colony optimization algorithm.
- **Chapter4** discusses the proposed approach and provides details of the new rule generating procedure that has been proposed.
- **Chapter 5** computes and analyses the results obtained while comparing the proposed approach with the other standard algorithm in this field. The result is compared in terms of average predictive accuracy, average number of rules generated and average number of terms per rules.
- **Chapter6** contains conclusion and scope for future work. Further modification and improvement that can be done in this direction have been suggested.

CHAPTER 2

CLASSIFICATION USING ACO

2.1 OVERVIEW

The process of classification is among several data mining tasks whose aim is to find a model that can represent a relationship between the predictor attributes i.e. attributes whose class is to be predicted and the class attributes' values. For classification task, different strategies have been applied which are comprehensible to users. Over the years, many biologically inspired algorithms like PSO, ACO have been used in classification. These evolutionary algorithms have provided good results and able to predict class more accurately than previous algorithms. This chapter is dedicated to those evolutionary algorithm and there procedure i.e. the way they are implemented for classification task of data mining. The whole chapter is divided into three sections. Section 1 details the classification process. Section 2 discussed the evolutionary algorithms which have been implemented in data mining for classification purpose. Section 3 enlightens the procedure how ACO has been used for generating rules which are more comprehensible to users.

2.2 CLASSIFICATION IN DATA MINING

Classification is a data mining task in which the aim is to discover, from labeled case, a model that can be used to predict the class of unlabeled cases [3]. Classification is the most popular data mining technique. It is the most frequent occurring task of human decision making and assigns an object to a predefined class according to its characteristics [9], [10]. Many problems in different areas such as image and pattern recognition, medical science, engineering, credit risk evaluation can be considered as classification problems.

2.2.1 TYPES OF CLASSIFICATION TECHNIQUE

Various classification techniques like linear and logistic regression, decision trees and rules, k-nearest neighbor classifier, support vector machine (SVM), neural network has been proposed [11], [12]. Some of these techniques provide high accuracy but there user comprehensibility is low i.e. model presented by these techniques is not easily understandable by human [12], [13]. Some techniques provide both accuracy and comprehensibility [14], [15]. So, classification techniques can be divided into basic five categories:

- Statistical-Based Algorithms
- Distance-Based Algorithms
- Decision Tree Based Algorithms
- Neural Network Based Algorithms
- Rule Based Algorithms

These techniques have been discussed below in details.

2.2.1.1 Statistical Based Algorithms

Statistical based algorithms are supervised learning methods that generalize a set of training data by creating a mathematical equation one or more input attributes to a single output attribute. Many statistical based methods have been developed. Bayesian classification and regression analysis are two most widely used statistical based methods.

Figure 1 shows simple linear regression with one input value. This model is an estimate of what an input-output relationship is. This linear generated model can be used to predict an output value given an input value using the formula

$$Y = c_0 + c_1x_1 + \ldots\ldots\ldots\ldots\ldots + c_nx_n \qquad (2.1)$$

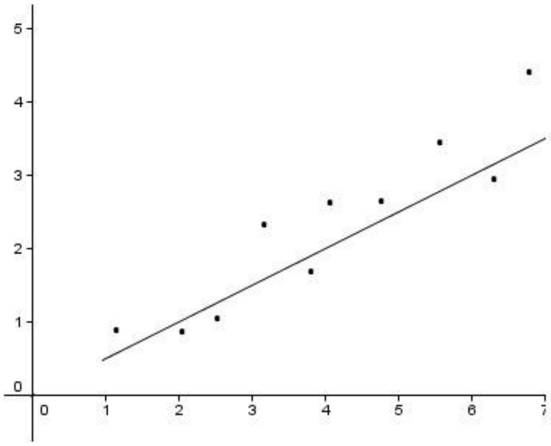

Figure 2.1: Simple Linear Regression

There are many reasons why linear regression may not be used to estimate output data. Linear regression model is quite sensitive to noisy and outlier data. Due to presence of these, the equation of line may change to

$$Y = c_0 + c_1x_1 + \ldots\ldots\ldots\ldots.. + c_nx_n + €\text{, where € is random error.} \qquad (2.2)$$

Regression can be used to perform classification using two different approaches:

1. **Division:** The data are divided into regions based on class.
2. **Prediction:** Formulas are generated to predict the output class value.

In the first case, the data is plotted into n-dimensional space without any explicit class value known. Through regression, the space is divided into regions-one per class. While in second approach, a value for each class is included in the graph. Using regression, the formula for a line to predict class values is generated.

Bayesian classification is based on the Bayes rule of conditional probability which states that

$$P(H \mid E) = P(E \mid H) \text{ x } P(H) / P(E) \qquad (2.3)$$

Where H is the hypothesis to be tested and E is the evidence associated with the hypothesis. From classification point of view, the hypothesis is the dependent variable and represents the predicted class. The evidence is determined by the values of the input attribute. P(E | H) is the conditional probability that H is true given evidence E. P(H) is a prior probability, which denotes the probability of the hypothesis before presentation of any evidence. The naïve Bayesian classifier or simple Bayesian classifier performs following steps for classification [3].

1. Let d be the training set of tuples and their associated class labels. Each tuple is represented by an n-dimensional attribute vector, $X = (x_1, x_2, \ldots\ldots., x_n)$.
2. Let there are m classes. Classifier predicts that X belongs to the class having highest posterior probability, condition on X. A tuple X belong to class C_i if and only if

$$P(C_i \mid X) > P(C_j \mid X) \quad \text{for } 1 \leq j \leq m, j \neq i. \qquad (2.4)$$

$$\text{Where, } P(C_i \mid X) = P(X \mid C_i) \text{ x } P(C_i) / P(X)$$

Here P(X) is constant for all classes.

3. Assuming that the class attributes' values are conditionally independent of one another, $P(X \mid C_i)$ is calculated as :

$$P(X \mid C_i) = P(x_1 \mid C_i) \times P(x_2 \mid C_i) \times \ldots\ldots \times P(x_n \mid C_i) \qquad (2.5)$$

So to predict the class label of X, $P(X \mid C_i)P(C_i)$ is calculated fro each class C_i. The classifier predicts that the class label of the tuple X is class C_i if and only if

$$P(X \mid C_i)P(C_i) > P(X \mid C_j)P(C_j) \quad \text{for } 1 \leq j \leq m, j \neq i. \qquad (2.6)$$

Bayesian classifiers are best among all classifier in the sense that they produces minimum error rate. Bayesian classifiers have been used for pattern recognition [28], speech recognition [29], feature extraction [30], and feature based sentimental classification [31]. In [32], a comparison is made between the neural network and Bayesian classifiers.

2.2.1.2 Distance Based Algorithms

Distance based algorithms are based on the concept that objects or items belonging to same class have more similarity to each other than to the objects and items of different class. The similarity measures here can be the distance between the item and the class attribute. Most common distance based algorithm used for classification is K-Nearest Neighbor.

K-Nearest Neighbor classifiers comes under the category of lazy learners [3] i.e. when a training tuple is given, a lazy learner simply stores it, and waits until it is given a test tuple. In this method, a training tuple with n attributes is represented as a point in n dimensional space. So, all training tuples are stored in n dimensional space. When a test tuple is give, K-Nearest Neighbor searches for K tuples in the n dimensional space i.e. Pattern space for K tuples that are closely related to the given test tuple.

"Similarity" or "closeness" is measured in term of Euclidean distance. The Euclidean distance between two tuples, say, $T_1 = (t_{11}, t_{12}, \ldots\ldots, t_{1n})$ and $T_2 = (t_{21}, t_{22}, \ldots\ldots, t_{2n})$ is defined as:

$$\mathrm{Dis}(T_1, T_2) = \mathbf{SQRT}\left(\sum\nolimits^{n}_{i=1}(x_{1i}-x_{2i})^2\right) \qquad (2.7)$$

This formula works well when the attributes values are numeric. If the values are categorical then, we just compare the corresponding values of the attribute in tuple T_1 with that in tuple $T_{2.}$ The continuous values of attribute A can be normalized to convert them in a range [0, 1] by

$$v' = v - \min_A / \max_A - \min_A \qquad (2.8)$$

Problem with Nearest Neighbor classifiers is that they are slow. If number of training tuples is D and K = 1, then O(D) comparisons are required to classify a given test tuple. Many techniques can be used to speed up the classification process like partial distance, pruning, editing. In partial distance method, the distance is calculated based on the subset of the n attributes. If this distance is greater than a specified threshold, then further calculation for that tuple is halted and move next tuple. In pruning method, the tuple that proves to be useless is removed and thus number of tuples to compare becomes less than the original set of tuples.

An improvement in k-nearest neighbor classifier is done where k nearest neighbor for each class of training set is calculated separately and mean is calculated for these k neighbors [33], [34]. K-nearest neighbor is used for many biometric applications like real-time finger and wrist control [35], brain image retrieval [36].These have been used to check quality of drinking water [37], for intrusion detection system [38].

2.2.1.3 Decision Tree Based Algorithms

Decision tree based algorithms are most useful approaches in classification problems and comes under eager learner category i.e. when a training case is given, it will construct a generalization model before receiving new test case to classify. In this method of classification, a tree is constructed to model the classification process. Once the tree is built, it is applied to each test case that is to be classified. A decision tree has following properties:

- Each internal node is labeled with an attribute, A_i.
- Each arc is labeled with a predicate that can be applied to the attribute associated with the parent.
- Each leaf node is labeled with the class, C_j.

Fig. shows a decision tree for the problem of *buys_computer* [3]. Here each internal node is represented with the attribute (age, student, credit_rating) and the leaf nodes represent the class. In this case, there are two classes i.e. no and yes.

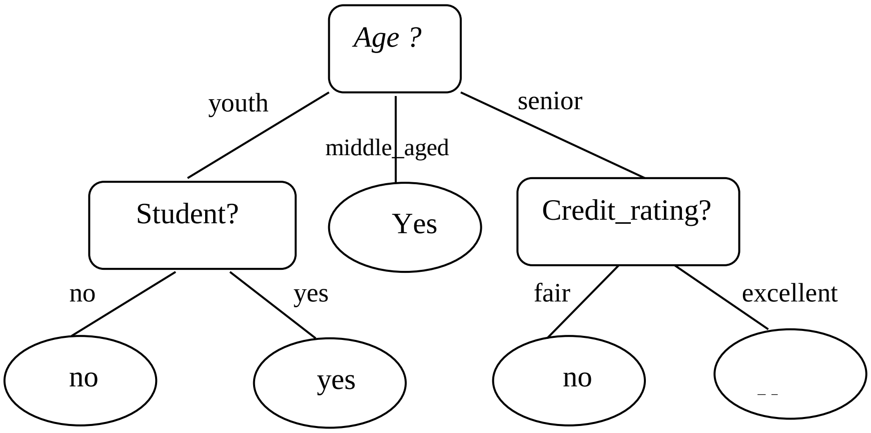

Figure 2.2: Decision Tree for Classification

Classification using decision tree is a two steps process:

1. Create the decision tree using Decision Tree Induction.
2. For each test case, use the decision tree to predict its class.

The major factors that affect the performance of decision tree algorithms are:

- Choosing the splitting attributes i.e. which attribute is selected to partition the test cases into individual classes.
- Order of splitting attributes i.e. order in which splitting attribute is selected. .
- Tree structure i.e. whether tree is balanced or unbalanced. There should be fewest levels if possible.
- Stopping criteria i.e. creation of tree should be stop when training data is classified perfectly.

Many decision tree based techniques like ID3, CART, C4.5 have been developed and used for classification task of data mining. Decision tree algorithms have been used for classifying emotions [39]. An improved decision tree ID3 algorithm is used to classify the imbalanced, noisy and dispersive data in forensic [40]. Decision trees have been used for diagnostic purposes [41] also.

2.2.1.4 Neural Network Based Algorithms

In these techniques, a model to classify a given set of training case developed. These techniques provide good results in terms of predictive accuracy. But the model created using these techniques are not easily interpretable by human. To solve a classification problem using neural network involves following steps:

1. Determine the number of output nodes and what attributes would be used as input. The number of hidden layers must be decided.
2. Determine the weights and function to be used.
3. For each case in the training set, propagate it through the network and evaluate the output prediction to the actual result. If prediction is accurate, adjust labels to ensure that this prediction has a higher output weight the next time. Otherwise, adjust the weights to provide a lower output value for this class.
4. For each case, C_i in dataset, propagate it through the network and make the appropriate classification.

Figure shows the basic structure of a neural network i.e. it has three layers namely, Input layer, Hidden layer, Output layer. These layers are connected to each other via weights along edge connecting from each node in previous layer and next layer.

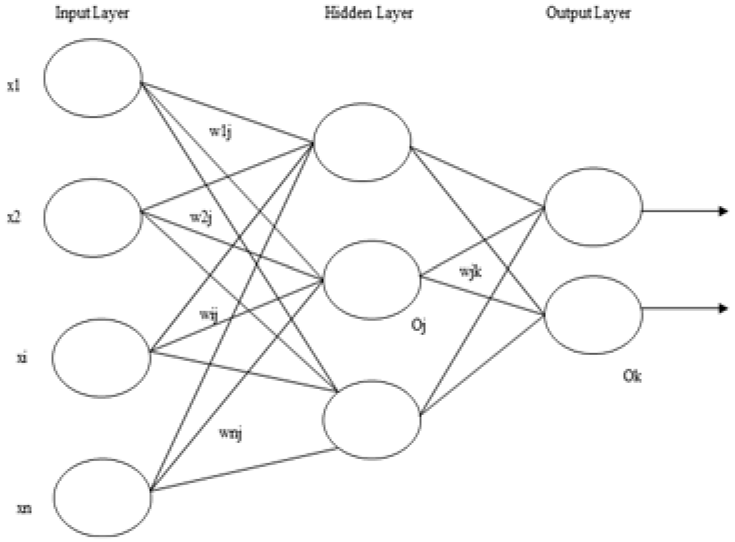

Figure 2.3: Neural Network

Many models like multilayer feed-forward network, radial basis function network, etc have been developed. Neural network algorithms have many advantages for classification:

- Neural networks are more robust than decision trees because of weights.
- There is low error rate and thus have high degree of accuracy.
- They have better performance in noisy environment.
- They improve its performance by learning.

Though neural networks have many advantages, there is some short coming also.

- Neural networks are difficult to understand i.e. user comprehensibility is low.
- Neural networks are quite expensive to use.
- Input values must be numeric.

Neural networks are used in many biometric applications like for face recognition systems [42], identification of hand movements [43]. In other applications, neural networks are used to personalized TV programs [44], star identification [45]. Radial Basis Function (RBF) model of neural network are used for vegetable price prediction [46].

2.2.1.5 Rule Based Algorithms

Rule based algorithms are straightforward way to perform classification. Rules are the best way to represent the information and are easily understandable by human i.e. have high comprehensibility [3]. Rule based algorithms have a set of IF-THEN rules to classify the data. An IF-THEN rule is of form [3]

IF condition THEN conclusion.

For example, consider a rule R1,

R1: IF age = *youth* AND student = *yes* THEN buys_computer = *yes*.

The condition part is also known as antecedent part and is a conjunction of attributes. The conclusion part is called as rule's consequent part and contains the class predicted when all attributes in antecedent part satisfy the conditions.

Rule based classification algorithms are best way of classification as rules can be easily interpreted by human. Rule based classification algorithms have been used for linguistic feature selection [47], ocean satellite image classification based on ontology [48], texture analysis [49]. We also have discussed how rule based classification is implemented using evolutionary technique i.e. ant colony optimization (ACO).

2.2.2 EVALUATING ACCURACY OF A CLASSIFIER

There are a lots of techniques used to measure the accuracy of a classifier. Most of these techniques are based on random partitioning of given dataset. These techniques may increase overall computation time. Some of these techniques are discussed under [3]:

- **Holdout Method and Random Sub sampling:** In this method, the data set is partitioned into two independent sets i.e. training set and test set. The training set is used to derive the model while the test set is used to measure accuracy of the model. Random sub sampling is a variant of holdout method in which the holdout method is repeated *k*- times.
- **Cross-validation:** In this method, the initial data is partitioned into k mutually exclusive subsets. Training and testing is performed k times. In i^{th} iteration, partition D_i acts as the test set while other partitions act as training set. We have also used cross-validation technique to measure the predictive accuracy of our model.
- **Bootstrap:** This method samples the given training tuples uniformly with replacement i.e. each time a tuple is selected, it is equally likely to be selected again and re-added to the training set.

2.3 EVOLUTIONARY ALGORITHMS

Evolutionary algorithms, subset of evolutionary computation in artificial intelligence, are generic population based meta-heuristic optimization algorithms [16]. These algorithms are biological inspired where the heuristic function which is problem dependent determines the environment in which the solution lives. Now a day, a number of biologically inspired algorithms e.g. genetic algorithm, ant colony optimization has been developed. These algorithms have been used in different domains in human life like image processing, data mining, medical sciences etc. The reason of using these algorithms over traditional algorithms is their accuracy in determining the solution.

2.3.1 Genetic Algorithm

Genetic Algorithms [17] are a family of computational models inspired by evolution. It is a global optimization method that manipulates a string of numbers in a manner similar to how chromosomes are changed in biological evolution. It solves a minimization problem by converting it into a maximization problem and finds global maxima. An initial population made up of strings of numbers is chosen at random or is specified by the user. Each string of numbers is called a "chromosome" or an "individual," and each number slot is called a "gene." A set of chromosomes forms a population. Each chromosome represents a given number of traits which are the actual parameters that are being varied to optimize the "fitness function". The fitness function is a performance index that we seek to maximize.

Genetic algorithm basically works in iterative manner and generates the new population from the old population. Each string in the population is represented in binary form. Genetic algorithm involves three genetic operators named *selection, crossover, and mutation* and applies these operators on the initial population strings to produce a new generation of strings. This iterative process improves the quality of the solution successively. The process ends when an optimal solution is found. Figure 1 shows how genetic algorithm iteratively finds the solution to a problem. The three basic operators perform the following functionalities:

1) *Selection:* Selection operator selects a proportion of the existing population to produce a new population. Selection criteria is based on the value of the fitness function i.e. population with good fitness value is selected for crossover at next step. Fitness function is the problem dependent heuristic function and measures the quality of the solution.
2) *Crossover:* This operator is used to generate the new population when the parent population is crossover. In crossover, a random point along the length of chromosomes is selected and genes of one chromosome after this point are swapped with the genes of another chromosome. Different methods are used for selection of chromosomes before applying crossover operator like roulette wheel selection, rank selection and tournament selection etc. Many crossover techniques are used depending on the selection of point along the length of chromosome e.g. single point crossover, two point crossover, cut and slice etc [15].
3) *Mutation:* Mutation operator is used to provide stochasticity to the solution and maintain genetic diversity between the two generations. In mutation, an arbitrary bit is changed from

its initial state. Different mutation types involve bit string mutation, gaussian mutation, non-uniform mutation etc [16]. Mutation operator prevents the local minima by avoiding the chromosome population from being similar.

These three operators are applied iteratively until the optimized solution is found, starting from the initial population which is selected at random.

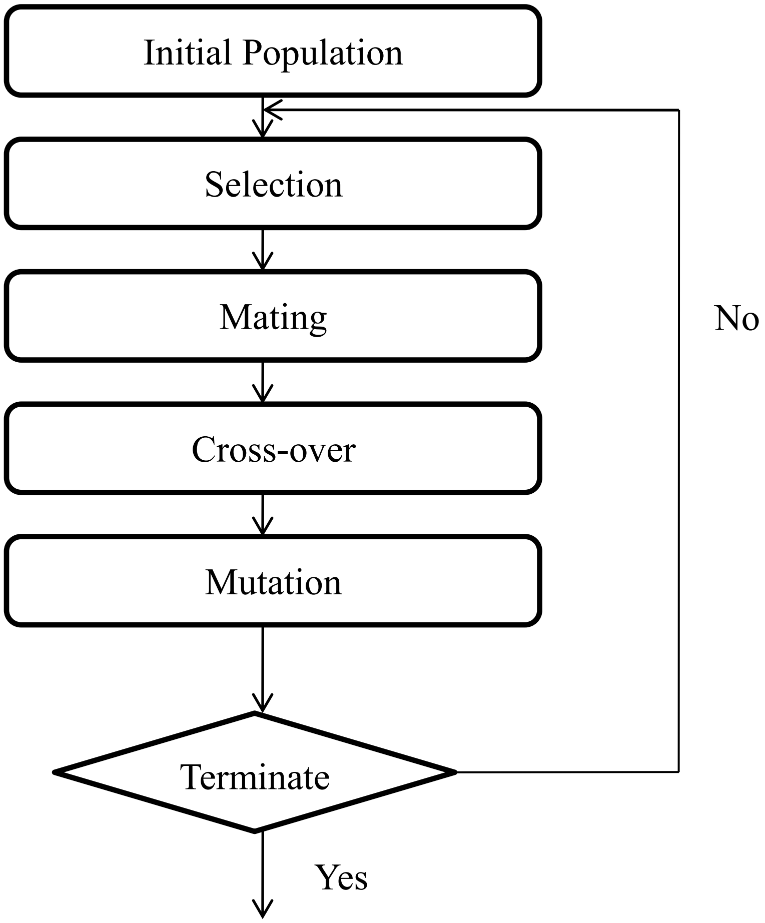

Figure 2.4: Genetic Algorithm Flow Chart

Genetic algorithm based approach has been used for classification task in data mining. It has been used to extract features [18], [19]. It is also used to discover knowledge from database by combining with neural network [20]. Fuzzy classification based models have been proposed in paper [21], [22] which are based on genetic algorithms. Genetic algorithms are used to develop face recognition systems and discovering rules from biological data [23], [24], and [25]. Genetic algorithms are also used for clustering task of data mining [26].

2.3.2 Particle Swarm Optimization

The technique [27] generates a set of relevant solutions called population and then finds an optimal solution through searching and updating the past history of the particles of the population. It is influenced by cognitive and social behavior of the swarms. Each particle has some velocity according to which it moves in the multi-dimensional space. Each particle also has memory to keep information of its previously visited space.

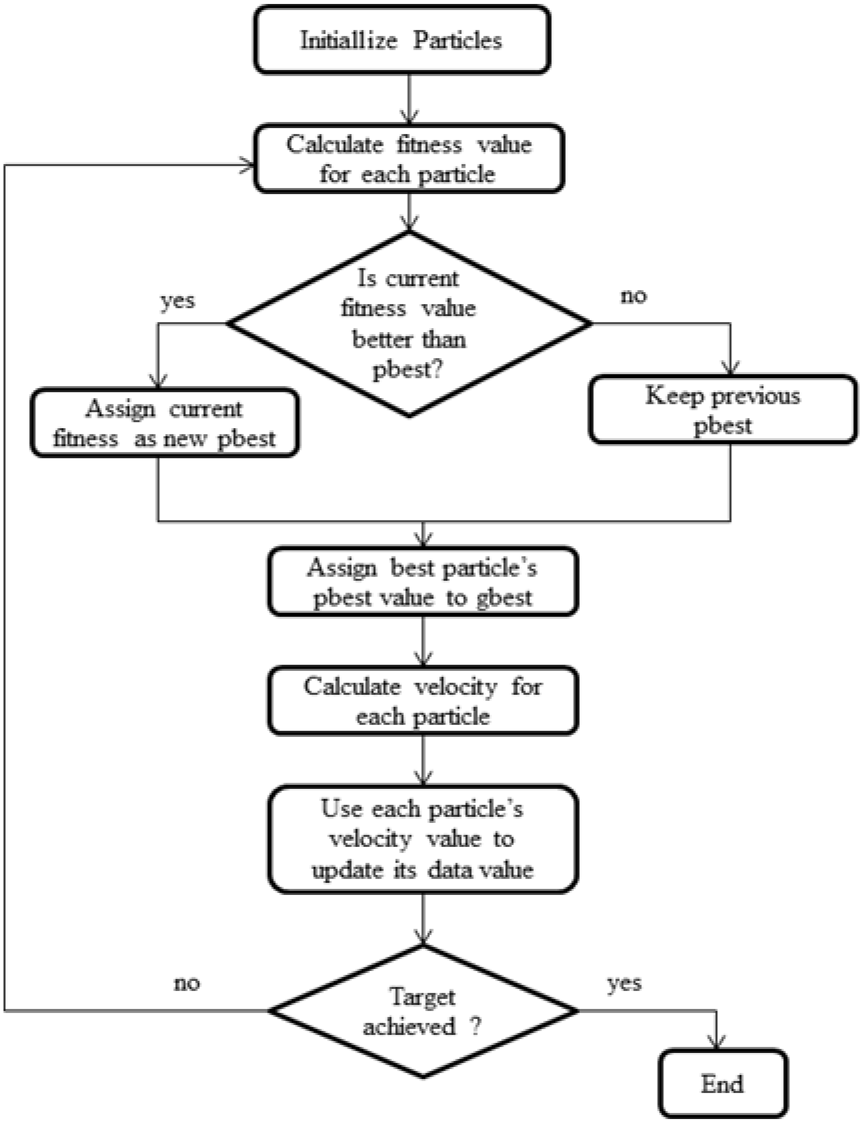
Initiallize Particles
Calculate fitness value for each particle
Is current fitness value better than pbest?
yes
no
Assign current fitness as new pbest
Keep previous pbest
Assign best particle's pbest value to gbest
Calculate velocity for each particle
Use each particle's velocity value to update its data value
no
yes
Target achieved ?
End

Figure 2.5: Particle Swarm Optimization(PSO) flow chart

The PSO algorithm is guided by two factors:

a) Movement of the particle in local neighborhood.

b) Movement of the particle in global neighborhood.

Local Best Solutions: - are the best solutions due to particle itself searching for the best solution in the restricted swarm.

Global Best Solution: - is the best solution due to all the particles participating in the solution space. For each iteration, Local and global best positions are updated only if better solution is found.

Notations

i) Position of the ith particle:-

$X_i = (X_{i1}, X_{i2}, \ldots.., X_{iN})$ is the ith particle of the swarm. Here, the first subscript
denotes the particle number and the second subscript denotes the dimension.

ii) Velocity of the ith particle:-

$Vi = (V_{i1}, V_{i2}, \ldots.., V_{iN})$

iii) Local best position X_{best} of the swarm:-

$X_{best} = (P_{i1}, P_{i2}, \ldots.., P_{iN})$

iv) Velocity Update:-

$V_{id}^{(k+1)} = (\omega * v_{id}^{k} + \gamma_1 * \alpha(X_{bestid} - X_{id}^{k}) + \gamma_2 * \beta(g_{best} - X_{id}^{k}))$

Where, i = (1, 2,…, m) is the number of swarms.

d= (1, 2,…, N) is the dimension of the objective function to be optimized

g_{best} is the global best solution of the swarm.

k is the iteration number.

w is the inertia weight to control the previous velocity vector of the swarm on the new one. It is a tradeoff between global and local exploration and helps in reducing the number of iterations for searching an optimal solution.

γ_1 and γ_2 are the random numbers between 0 and 1.

α is called the cognitive parameter.

β is called the social parameter.

Generally, $\alpha + \beta <=4$ and by default, $\alpha = \beta =2$.

v) Position Update:-

$X_{id}^{(k+1)} = x_{id}^{k} + (V_{id}^{(k+1)} / q)$

Where, q is the correction factor (optional) to speed up the convergence process.

Particle swarm optimization (PSO) based algorithm are used for various data mining tasks i.e. for clustering [50]-[53], for classification [54]-[56]. PSO is also used for association rule mining [57].

2.3.3 Ant Colony Optimization (ACO)

Ant colony optimization is an optimization algorithm based on swarm intelligence and was initially proposed by Marco Dorigo in 1992. Since then, ACO has been formalized into meta-heuristic for solving optimization problem by many authors [58], [59]. A meta-heuristic is a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide set of different problems [59]. A meta-heuristic is general-purpose algorithmic framework that can be applied to different optimization problems with relatively few modifications. Examples of meta-heuristics include simulated annealing [60], [61], tabu search [62]-[64], iterated local search [65], evolutionary computation [66]-[69], and ant colony optimization [70], [58], [59], [71].

Ants which are completely blind or have a limited visual capability are able to locate the shortest path between the food and the nest. Ants use released pheromones as communication medium. When an ant walks from the food source to nest, it drops pheromone on the ground, thereby making a trail of pheromone. The path with greater amount of pheromones deposited on it is more likely to be followed by the ants. In this way, ants build candidate solution to optimization problem based on their pheromone and heuristic function. Heuristic function is the problem dependent function. After iterative process of building candidate solutions, ants are able to converge to optimal solution.

Figure 2.6 shows the movement of ant between food and nest.

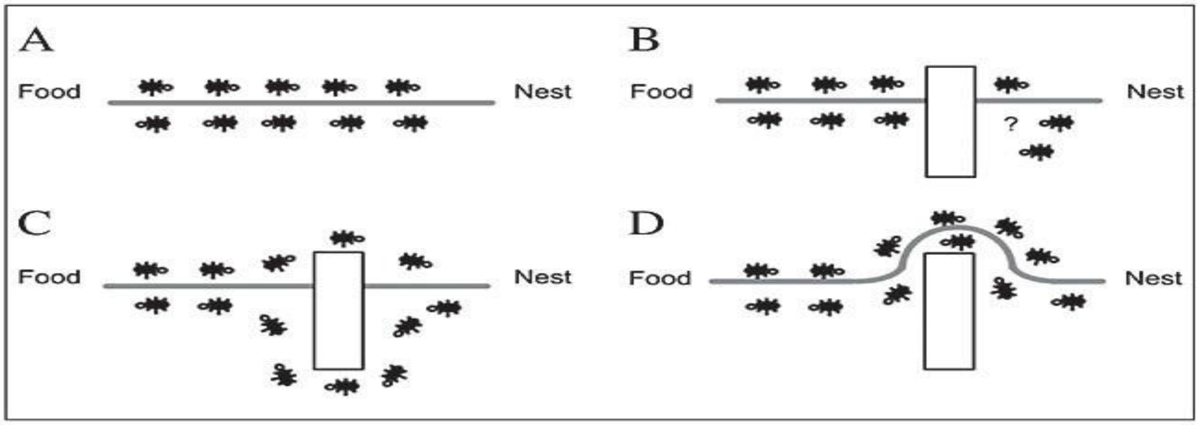

Figure 2.6: A. Ants in a pheromone trail between nest and food; **B:** an obstacle interrupts the trail; **C:** ants find two paths to go around the obstacle; **D:** a new pheromone trail is formed along the shorter path.

So, while designing ant colony optimization algorithm, following points should be considered [72]:

1. Problem representation for which ants create/modify a solution using probabilistic transition rule, based on the amount of pheromone and a local problem dependant heuristic.
2. Construction of valid solution by a proper procedure i.e. solutions that are legal in real world.
3. Problem dependant heuristic function that measures the quality of items that can be added to current partial solution.
4. Tactics for updating the pheromone
5. Probabilistic transition rule based on pheromone trail and heuristic function.

ACO has been used to find a solution of different problems [70], like scheduling [73], [74], travelling salesman problem [71], [75], [76], timetabling [77], vehicle routing [78], [79], [80], and routing in packet switched networks [81]. Recently, ant colony optimization (ACO) algorithms have been used in data mining domain both in clustering [82], [83] and classification [84], [85], [86].

2.4 USING ACO FOR CLASSIFICATION IN DATA MINING

The first use of ant colony optimization (ACO) for classification task of data mining was reported in [84] by Parpinelli et al. and named as AntMiner. AntMiner algorithm follows a sequential strategy by creating one rule at a time until all the training cases are covered by the rules created or the training cases left are few. The high level pseudo code of the Ant miner algorithm is given in Pseudo code 1 [87].

```
Training Sets = {all training examples};
DiscoveredRuleList = {} /* initialized with empty list */
repeat
        each trail is initialized with same amount of
        Pheromone;
repeat
        An ant incrementally constructs a candidate classification rule;
        Prune the just-constructed rule;
        Update the pheromone of all trails;
until (stopping criteria)
        Choose the best rule out of all candidate rules constructed by all ants;
  Best rule is added to DiscoveredRuleList;
Training Sets = Training Sets – {examples correctly covered by best rule};
until (stopping criteria)
```

Pseudo code 1: A high-level description of the original Ant-Miner

The inner repeat-until loop incrementally constructs a candidate rule. The whole process starts with an empty rule and incrementally adds one term to the current constructed rule. This process of constructing rule ends when either of the two criterion is met: adding the term to the current rule makes it to cover number of training cases less than user specified threshold, or all the attributes have been added to the current rule i.e., there are no more attributes to be added to the rule. After the rule construction process is finished, the rule created is pruned to remove the irrelevant terms from the rule and to increase the quality of the rule. Rule pruning strategy used in [87] is same as used in [88]. The pheromone amount associated with the terms present in the current generated rule is increased based on the rule quality. The quality of the rule is expressed by formula [89] given below:

$$Q= \text{sensitivity x Specificity} \qquad (2.9)$$

Rules creation process is repeated until the maximum number of iterations of process has reached or the rule covers the minimum number of training cases. After rule is pruned, the amount of pheromone associated to terms included in the rule is updated using equation:

$$T_{ij}(t+1)=T_{ij}(t) + t_{ij}(t)*Q, \qquad \text{for all } i, j \varepsilon R \qquad (2.10)$$

For the terms that do not occur in rule, a pheromone evaporation strategy should be carried out. For this purpose the pheromone amount associated with the term T_{ij} is decreased by dividing the current value by summation of all T_{ij}, for all i, j.

The repeat-until loop is ended when either of the two stopping criteria is met: number of constructed rules is equal or greater than maximum number of ants specified by user or the rule constructed by the current ant is same as created by any other previous ant. After the repeat-until loop is completed, the best rule generated is added to the list of discovered rule. It may be noted that, each iteration of while loop corresponds to a single rule. Training cases satisfied by generated rules are removed from the training set. This process is repeated until all the test cases are classified or the numbers of test cases left are few.

The probability of that a term_{ij} is chosen to be added to the current partial rule is

$$P_{ij} = \eta_{ij}.\tau_{ij}(t) / \sum^{a}_{i=1} x_i.\sum^{bi}_{j=1}(\eta_{ij}.\tau_{ij}(t)) \qquad (2.11)$$

Where, η_{ij} is problem dependent heuristic function for $\text{term}_{ij.}$ $\tau_{ij}(t)$ is amount of pheromone associated with term_{ij} at iteration t. a is total number of attributes. x_i is set to one if attribute A_i was not yet used by the current ant or to zero, otherwise. B_i is number of values in the domain of the i^{th} attribute.

2.5 CONCLUSION

In this chapter, we have discussed all the various types of classification, evolutionary algorithms that have been used in data mining. We also see how ant colony optimization (ACO) has been proposed and used for classification task of data mining. In the next chapter, we have discussed the related work in the classification task of data mining using ACO and also discussed the direction where work can be done to make AntMiner more robust and fast.

CHAPTER 3

RELATED WORK AND PROPOSED APPROACHES

3.1 OVERVIEW

This chapter throws lights on the related work that has been done by different authors and researchers in the field of generating classification rules using ant colony optimization. It includes the various versions of ant-miner starting with the original Ant-Miner which was the very first algorithm for classification task of data mining proposed in 2002. The chapter is divided two sections. Section 3.1 discusses the related work and different versions of original Ant-Miner proposed by different researchers. Section 3.2 discusses the problem statement and a direction to proposed approach.

3.2 RELATED WORK

Since Parpinelli et al. [84], [87] first applied the ACO algorithm in data mining known as Ant-Miner algorithm, a number of different variants have been proposed and successfully implemented for rules classification. The major areas for future research in Ant-Miner [87] proposed by Parpinelli were:

- the types of attributes it handles,
- heuristic function used to select the next term to be added to current rule,
- and pheromone updating strategy.

After Parpinelli providing a direction to future scope in applying ant colony optimization (ACO) for classification task of data mining, numerous authors and researchers focused on different future directions and proposed different versions of original Ant-Miner.

3.2.1 Versions of Ant-Miner

Various versions of Ant-Miner are proposed that try to overcome the various limitations in original Ant-Miner and enhance it with different strategies. This sub-section discussed the various versions of original Ant-Miner.

3.2.1.1 cAnt-Miner

Basically a dataset can contain different types of attributes. Most commonly used are continuous attributes and categorical attributes. Paper [87] discussed the way of handling categorical attributes only. The continuous attributes are discretized first in the preprocessing steps. F. Otero [90], [91] discussed the way of handling continuous attributes. The new version, he proposed of Ant-Miner i.e. cAnt-Miner sets thresholds on values in continuous attributes' domain dynamically while the rule is under construction. cAnt-Miner can handle continuous attributes directly, there was no need of discretization in preprocessing step.

3.2.1.2 Multi-Pheromone Ant System

Multi-Pheromone Ant System was proposed by Khalid M. Salama and Ashraf M. Abdelbar in [92]. The original Ant-Miner [87] was could handle only the categorical attributes, so the operator used for attribute and its value was "=". In multi-pheromone ant system, four different extensions have been proposed:

1) Allowing logical negation operator in antecedent part of the rule.

2) Using stubborn ants i.e. ants that uses own personal history.

3) Different types of pheromones for different rule class permitted.

4) Allowing each ant to have its own value of α and β stating that each ant has its own personality.

In this Ant-Miner version, the idea is something like that every class deposits different pheromone value on terms. In rule construction, ant is affected by the pheromone value in pheromone array corresponding to the respective rule class. Updating is also done on the same concept and ant deposits pheromone on the terms that belongs to that trial. Each class value was also shown in form of node. For more details about the pheromone update equation and heuristic function, refer to [92].

3.2.1.3 Ant-Miner3

Leu [85] proposed a new Ant-Miner3 that lies on density based heuristic function. The original Ant-Miner used the concept of entropy to find the heuristic value. However, ACO algorithm may not require the right information in this heuristic value as the small errors can be compensated by using pheromone. So we can define a simpler heuristic that may do the work as being performed by the complex one. As a result, Leu [85] proposed in this paper the following easily computable density estimation equation:

$$\eta_{ij} = \mathit{majority\_classT_{ij}} / \mathrm{mod}(T_{ij}) \text{ where,}$$

majority_classT$_{ij}$ is partition T_{ij} majority class. This heuristic showed results equivalent to the result obtained using entropy function. Though this idea of using density function as heutistic function was not new, it had been widely used in Bayesian classification and Lazy Bayes Rule. But using it in Ant-Miner domain was new. For further reading, refer to [85].

3.2.1.4 Ant-Miner+

Another version AntMiner+ was proposed which was typically based on the MAX-MIN ant system. In this an environment was defined for ants to walk. Parameters were also set automatically. Directed graph that acted as the environment for ants to move was defined. Ants travel from one vertex to another forming a rule. Ant creating best rule was allowed to update pheromone of terms or node on the path it covered. You can get further details about the working of Ant-Miner+, refer to [93].

3.2.1.5 cAnt-Miner$_{PB}$

The latest version of cAnt_Miner is discussed in [94] named cAnt-Miner$_{PB.}$ This proposed algorithm uses the sequential covering approach to generate rules while taking into consideration rule interaction. This paper deals with the idea that a rule's outcome can affect the rules that can be generated after that rule because every time a rule is generated the search space is changed as the training data covered by the rule is removed.This new ant-miner algorithm was compared with other state-of-art algorithm i.e. cAnt-Miner, PSO/ACO2, CN2, PART, JRip. The experiment results and comparison showed that cAnt-Miner$_{PB}$ is better than other algorithm when predictive accuracy, average number of rules generated and average number of terms per rule are measured. For more about cAnt-Miner$_{PB}$, refer to [94].

3.2.2 Other related work

Paper [95], [96] proposed different types of pheromones for each rule class permitted i.e. the rule class is selected by the ant and then corresponding pheromone is deposited and pheromone updating strategy respectively.

A. Chan [97] has proposed a new way to prune rule generated using ant colony algorithm. This work combines the original Ant-Miner method of rule pruning with a rule pruning on the basis of information gain discussed as in [98]. For more on information gain, refer to [99]. Rule pruning procedure explained in [97] is a new hybrid rule pruner i.e. it combines original rule pruning method to rule pruning based on information gain. In this method, first calculate the information gain for each term using all cases in training set. A user defined variable, r, to define the number to terms given to original ant pruning method.

In this hybrid rule pruning method, the parameter, r, is user defined. Value of r is to be chosen carefully. They have also given a future direction to choose the value of r automatically and developed hybrid rule pruning method that is adaptive. Paper [94] also guides future direction to evaluate a global method to prune rule. In global method, a rule is pruned while taking into consideration its effects on other rules in the list.

3.3 CONLUSION

So, from all the literature survey and related work in the direction of rule classification in data mining using ACO algorithm, it can be concluded that the work on the rule pruning procedure is an area still to be explored. In our work, we have proposed a new global rule pruning procedure that will count the effect of removing a term from rule while pruning on other rules in the list. This new global pruner will also estimate the association among the terms in the rule generated and based on the estimation concludes whether the current term should be removed from the rule or not.

CHAPTER 4

PROBLEM DEFINITION AND PROPOSED APPROACH

4.1 OVERVIEW

Literature survey done in chapter 3 shows that there are numerous directions and areas where work can be done to improve the application of ant colony optimization (ACO) in classification task of data mining. Chapter 3 also discussed the various versions of original ant-miner proposed by various authors also enlighten the future directions given by different authors. In this chapter, we have discussed our proposed approach and try to improve the original ant miner in respect of numbers of rules generated, average number of terms per rule that define rules' comprehensibility. The whole chapter is divided into subsections. Section 2 defines rule pruning. Section 3 states the problem. Section 4 discussed the proposed approach.

4.2 RULE PRUNING

Rule pruning is a process where irrelevant terms are removed from the rules that are present in it. Rule pruning is done in order to remove the over fitting of training data. It not only avoids over fitting but makes the rules simple also as the smaller the rule the better it can be understand by user. A rule, R, is pruned if the quality of the pruned version of R is greater than that of original R. Various rule pruning strategies like FOIL are there that can be used.

4.3 PROBLEM DEFINTION

In the original method of rule pruning in Ant Miner [87], a term is removed from rule by the ant until the removal of a term increase the rule quality. It is measured using the formula [89] given in equation (4.1).

$$Q = (TP/(TP+FN) * TN/(TN+FP)) \qquad (4.1)$$

Where TP = cases covered by the rule and class is same as predicted by the rule

FN = cases covered by the rule and class is different from the class that is predicted by the rule.

TN = training cases that rule does not cover and has class is different from that is predicted by the rule

FP = training cases that rule does not cover and has class is same to the one that is predicted by the rule

The pseudo code of rule pruning procedure in original Ant-Miner algorithm [87] is as:

```
while (number of terms in the rule antecedent > 1)
    for each (Term$_i$ in current rule to be pruned)
        remove temporarily term$_i$ and assign the rule the most occurrence class among the training cases covered by the rule;
        Evaluate rule quality;
        if (rule quality is improved with respect to original rule quality) then
            remove that term from rule permanently;
    end for each
end while
```

PseudoCode2. Ant-Miner original rule pruning Procedure.

In the algorithm discussed in [94], the interaction between the rules is taken into account but the pruning procedure used prune the rules locally i.e. one rule is created and pruned without considering its effect on other rules in the list. The term selected to be removed from the rule's antecedent part of rule should be chosen carefully. There may be association between the different terms e.g. in association rules there is a correlation between the terms. This correlation or association between the terms should be carefully measured before removing a term from the current rule generated.

4.4 PROPOSED APPROACH

Before generating the rules and applying rule pruning, first define the environment in which ants move. The environment is defined as a directed graph. Directed graph are graph in which there is direction from one node to another. The nodes in our case represent the variables or terms. So, if variables V_1, V_2,….., V_n are given with variable V_i having values $Value_{i1}$, $Value_{i2}$,…..$Value_{ip}$ where p is the maximum possible values for variable V_i, then the graph is defined as:

Vertices: Vertex is defined for each variable V_i for each of its values $Value_{ij}$. The complete set of vertices is defined as:

$$\sum^{n}_{i=1} P_i \qquad (4.2)$$

The vertex set for one variable is called as vertex group.

Correspondence: Correspondence defines the path between the vertices in vertex sets. The path or correspondence is defined as:

$$\Gamma(\text{Start}) = v_{i,j} \quad i = 1, 2, \ldots\ldots, n$$

$$j = 1, 2, \ldots\ldots., p_i$$

$$\Gamma(v_{i,j}) = v_{k,l} \quad k=1, 2\ldots\ldots\ldots, i\text{-}1,i+1,\ldots.n$$

$$l = 1, 2, \ldots\ldots.,p_k$$

The graph as shown in figure 4.1 shows the symmetric graph which defines the environment for ants to move for classification and generating rules.

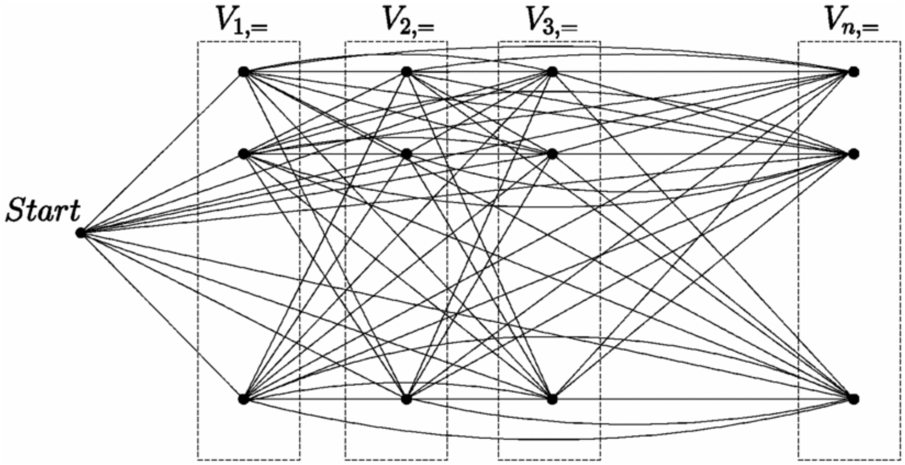

Figure 4.1: Symmetric graph of environment for ants to move

In our problem of classification ants discover the classification rules including the terms coming in their path to the rule. The high level pseudo code for the proposed approach is shown under:

Input: Dataset

Output: best discovered rules list.

Algorithm:

```
initialize all $term_{ij}$ with same amount of pheromone
$T_{ij}(t=0) = 1/\sum^{a}_{i=1}bi$,
Where a =number of attributes and $b_i$ = possible values attribute $A_i$ .
DiscoveredRuleList = []; /*initially empty discovered rule list.
n = 0;
while (n < maximum iterations  )
    $list_n$={};
    for  n=1 to no_of_ants
    do
        TestCases  = TrainngSet;
        while( TestCases > max_uncovered_cases )
        do
            HeuristicFunction(TrainingCases);
            rule = CreateRule(TrainingCases);
            TestCases= TrainingCases – Covered(TrainingCases);
            $list_n$ = $list_n$ + rule;
        end while
    end for
    n= n + 1;
end while
Prune($list_n$);
DiscoveredRuleList = $list_n$
return DiscoveredRuleList;
```

Pseudocode 3: Proposed algorithm.

The algorithm discussed above, first generate all the rules and store them in the list in the inner do-while loop. After the rules are generated, pruning method is called The probability of selecting a term$_{ij}$ to add it to current partially formed rule is

$$P_{ij} = \frac{\eta_{ij}\,.\tau_{ij}(t)}{\sum^{a}_{i=1} x_i \,.\, \sum^{bi}_{j=1}(\eta_{ij}\,.\tau_{ij}(t))} \qquad (4.3)$$

The heuristic function, $\eta_{ij,}$ in equation (4.3) used to determine which term should be added to the current partial rule is same as used in [87]. This heuristic function is based on the information theory [100]. It calculates the value of η_{ij} for the term$_{ij}$ on the basis of its entropy. $\tau_{ij}(t)$ is updated according to the equation (4.1).Before the algorithm begins, every node have equal amount of pheromone. The DiscoveredRuleLIst is empty initially. The training cases are equal to training set. Ants generate the rules in the inner while loop and the rules are added to the list$_n$. After all the rules are generated, the list is pruned.

Prune function used above for rule pruning prunes rules globally. It first finds the association among the terms in the rules using their support and confidence values. This association helps in deciding which terms are important in the rule and should not be discarded. After this step, all the irrelevant terms i.e. whose support and confidence is less than a specified threshold are removed from rules. Find the confidence value for each rule and select those rules only with confidence value greater than the set threshold. At last step, rules are sorted on the basis of their confidence. The high level pseudo code for pruning method is as:

```
Prune($list_n$){
Step 1: find the association of terms present in the rule to other terms in the rules present in rule list.
Step 2: Select those terms that have higher support and confidence than threshold.
Step 3: Discard all terms having lower values for support and confidence than specified threshold.
Step 4: Find the confidence of each rule and select rules having confidence value greater than the threshold.
Step 5: Sort the rules on the basis of their confidence.
}
```

Pseudo Code 4: Proposed rule pruning method.

After $list_n$ is pruned, DiscoveredRuleList is updated by adding all rules in $list_n$ to DiscoveredruleList. After the algorithm reaches the maximum iteration, the algorithm stops and the DiscoveredRuleList is returned.

4.5 CONCLUSION

Throughout this chapter, we discussed the problem and then discussed the proposed algorithm. We also discussed how the new rule pruning procedure is applied for rule pruning. The computational results and analysis of the above algorithm are discussed in next chapter.

CHAPTER 5

COMPUTATIONAL RESULTS AND ANALYSIS

In order to analyze the new rule pruning method, we have used publically available dataset from UCI machine repository [101]. We have carried out result on one dataset named dermatology. Table 1 shown below summarized the dataset details.

Table 1: Dataset details

Dataset	Nominal Attributes	Continuous Attributes	Classes	Size
Dermatology	33	1	6	366

The first column gives name of the dataset used. Other columns indicate nominal attributes, continuous attributes, number of classes in the dataset and number of test cases in the dataset respectively. We chose this dataset because it contains maximum number of attributes than other datasets used in [87]. As the number of attributes increases, the rule pruning process takes more time and become complex. So, our aim was to choose the dataset with maximum attributes so as to compare the predictive accuracy with respect to result obtained in [87] for that dataset.

We have compared the result with previously proposed ACO based rules classification algorithms. Some of the known such algorithms are as follows:

1) *c*Ant-Miner2MDL [90], [91]: This algorithm uses minimum description length (MDL) principle [102] to create a threshold for continuous attributes domain values during rule generation process.

2) PSO/ACO2 [54]: This hybrid algorithm deals with both continuous and nominal attributes and constructs rules in two steps: nominal attributes are used firstly to create rule’s antecedent part and then continuous attributes are added to extend the rule’s antecedent.

3) CN2 [103]: CN2 is also rule classification algorithm that follows the sequential covering approach to construct rules similar to ACO procedure to construct rules.

4) cAnt-MinerPB [87]: This algorithm also construct rules while taking rule interaction into account i.e. result of one rule can affect other rules subsequently generated. This algorithm also deals with both types of attributes i.e. nominal as well as continuous.

The whole work has been carried out on a system having 2.0 GHz Intel Core Duo processor, 1GB RAM, and 80 GB hard disk. The software used for implementing the proposed approach is MATLAB. Effect of the system parameters is significant. The other approaches with which we compare the result is carried out on Intel Xeon 3.33 GHz processor with 8 GB RAM. So, we don't discuss the time complexity parameter and compare it with other algorithm.

5.1 PARAMETERS SETTING

The parameters used in the above algorithm are: maximum number of iterations, the colony sizes i.e. the number of ants, maximum uncovered test cases, minimum number of test cases covered by the ant. These parameters are set as:

1) no_of_ants = 2000;
2) min_covered_rules = 10;
3) max_uncovered_cases = 10;
4) no_of_iteration = 100;

There parameters are chosen without any optimization. However, these non-optimized parameters also produce good result.

5.2EXPERIMENT RESULTS

Rule pruning is the major concern of our work. So, we first have evaluated the result without applying the proposed rule pruning procedure.

Table 2 shows the experimental results which details the number of rules generated and average number of rules generated without using rule pruning.

Table 2: Experimental results without rule pruning

Dataset	Number of rules produced	Average number of terms per rules
Dermatology	36	28

From table, it can be analyzed that the number of rules and terms are considerably high i.e. rules are not more comprehensible to user. So, we also evaluated the results after applying the proposed global rule pruning procedure. Table 3 details the effect of the proposed global rule pruning procedure and summarizes the results as shown below:

Table 3: Experimental results with proposed global rule pruning

Dataset	Number of rules produced	Average number of terms per rules
Dermatology	29	15.43

From results shown in table 3, we can say that the proposed global rule pruning is quite affective as the numbers of rules generated have been decreased considerably and the number of terms per rule is also less than the original rule.

5.3 COMPARISON WITH OTHER ALGORITHMS

Table 4 compares the predictive accuracy of the proposed approach with different above said algorithm. From comparison it is clear that ACO algorithm with new proposed rule pruning procedure shows predictive accuracy comparative to other algorithms. The predictive accuracy of the rules to classify the new training data is 85.38 % according to our approach. We have used 10-fold cross validation technique to measure the predictive accuracy.

Table 4 summarizes the comparison.

Table 4: Predictive Accuracy

Algorithm	Predictive Accuracy in %
cAnt-Miner2MDL	89.52
PSO/ACO2	91.80
CN2	91.79
cAnt-MinerPB	92.46
Proposed Approach	85.38

Table 5 and Table 6 represent the average number of rules generated and average number of terms in the discovered rules respectively. Average number of rules discovered is a measure for end users' comprehensibility [104].

Table 5: Number of Rules

Algorithm	Average number of rules
cAnt-Miner2MDL	9.21
PSO/ACO2	10.27
CN2	18.80
cAnt-MinerPB	19.21
Proposed Approach	29

Table 6: Average number of terms per Rule

Algorithm	Average number of terms per rule
CAnt-Miner2MDL	20.47
PSO/ACO2	30.61
CN2	48.41
CAnt-MinerPB	44.39
Proposed Approach	15.43

Comparison shows that though the number of rules discovered is little bit high yet the proposed approach is competitive to other algorithms in terms of average number of terms per rule. The number of terms per rule i.e. size of the classification modal is also lowest according to our approach so the rules are more comprehensible to users. Figure 5.1-5.3 shows the results in bar chart form for proper visualization.

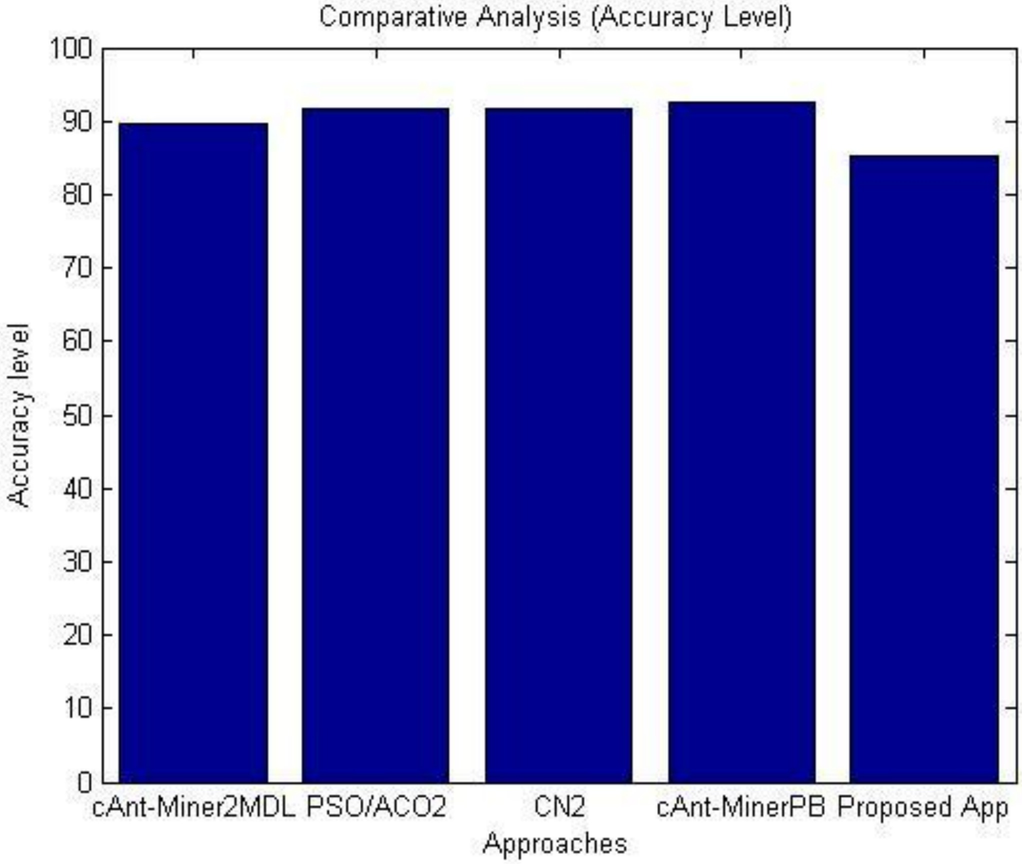

Figure 5.1: Comparative Analysis (Average Accuracy)

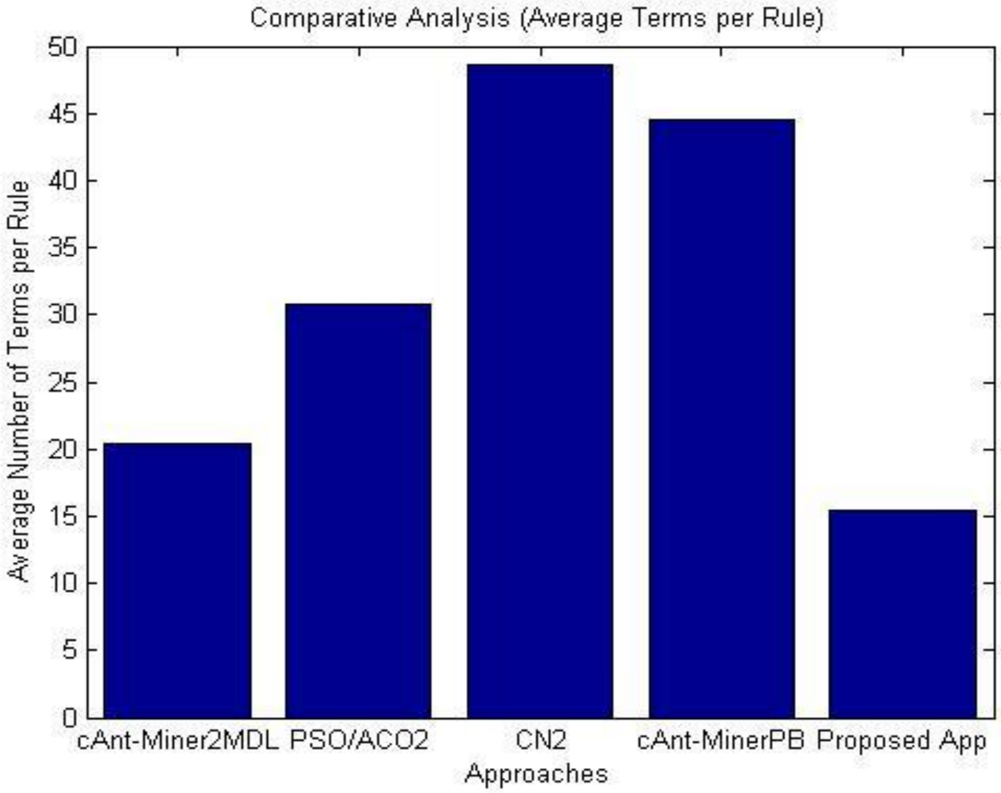

Figure 5.2: Comparative Analysis (Average Terms per Rule)

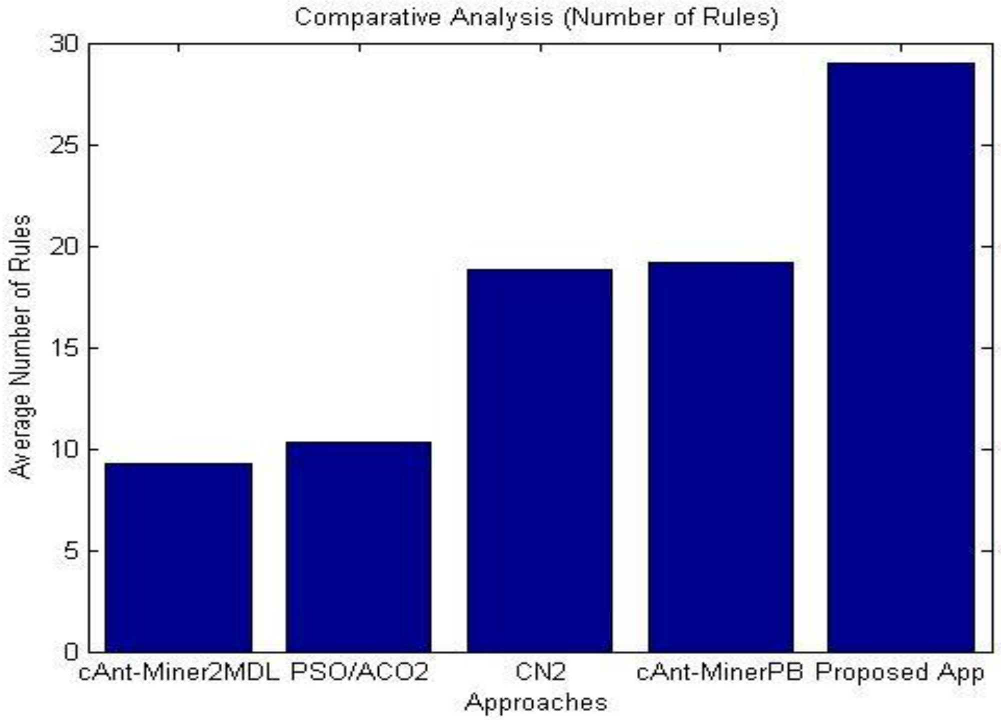

Figure 5.3: Comparative Analysis (Number of Rules)

Figure 5.4-5.6 shows the effect of setting confidence threshold on the number of rules generated, average number of terms per rule, average accuracy in bar graph form.

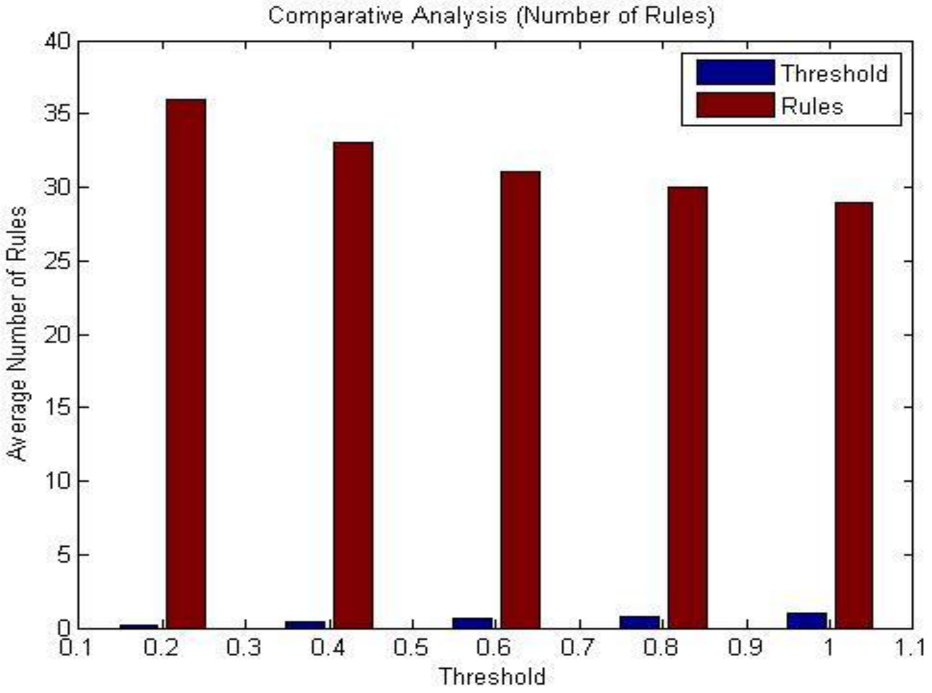

Figure 5.4: Comparative Analysis (Number of Rules v/s Threshold)

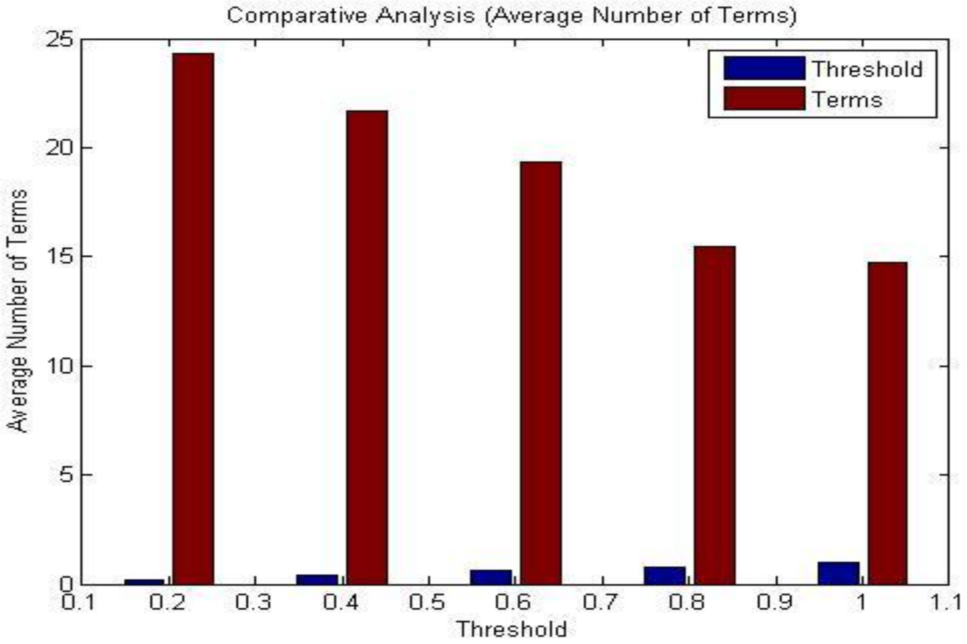

Figure 5.5: Comparative Analysis (Average Terms per Rule v/s Threshold)

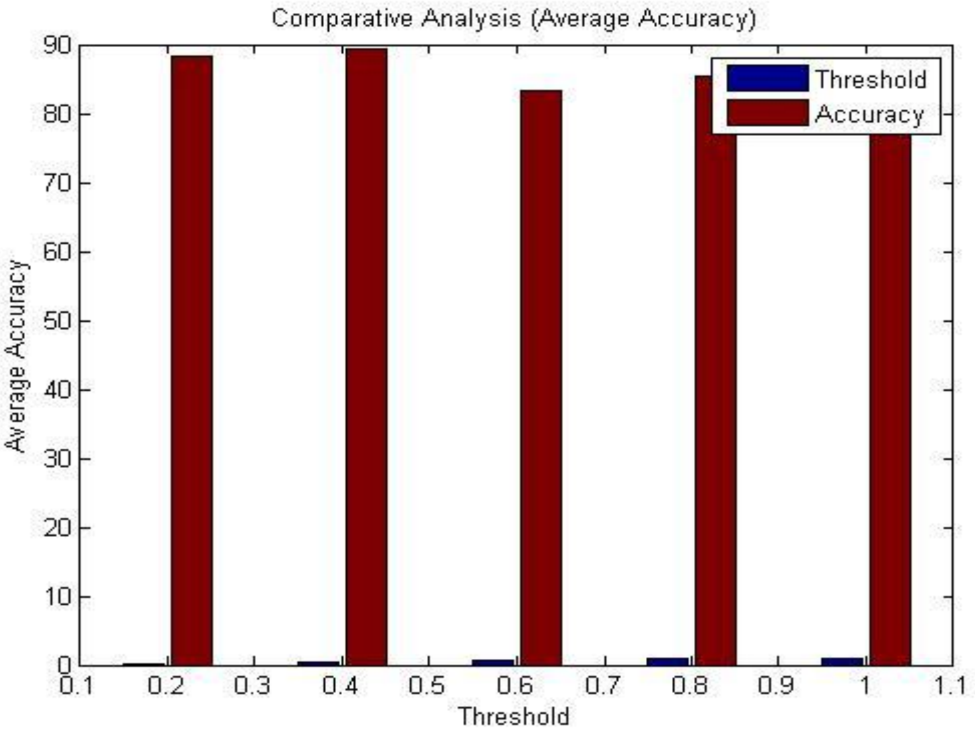

Figure 5.6: Comparative Analysis (Average Accuracy v/s Threshold)

5.4 CONCLUSION

In this chapter, we discussed and analyzed the experimental results. We saw how the proposed global rule pruning procedure is effective. We also compare the results with other popular algorithms and conclude that the results obtained using proposed global rule pruning procedure is quite satisfactory and competitive to these popular algorithms.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 OVERVIEW

This chapter concludes the work carried out to generate rules with global rule pruning procedure and provides direction to future work. Subsection 1 concludes the thesis. Section 2 provides the various directions to future work that can be followed.

6.2 CONCLUSION

Classification is the interesting and most likely task of data mining. Rules based classification is among the various types of classification techniques. Using rules based classification has its own importance in terms of user's comprehensibility i.e. rules can be easily understand and interpret by a man who has no idea of how the process of classification is being carried out over others methods of generating rules e.g. support vector machines(SVM) and neural networks for classification.

Different ancient rules based classification methods have been proposed like C4.5, CART. Over the years, classification was carried out with the help of evolutionary algorithms which were inspired by the biological behavior of swarms. Particle swarm optimization (PSO) and Ant colony optimization (ACO) are two major used evolutionary algorithms in this field. Ant colony optimization (ACO) was first applied in 2002 for generating rules where ant behavior of solving a problem was used. That original ant-miner was named Ant-miner and provided many directions that could be explored.

Ant-miner was modified in terms of different directions like based on the types of parameters it handles, syntax of the IF-THEN rules, environment in which the ant moves, heuristic function that define the problem, way in which pheromone values of different nodes is updated etc. Most of these different methods follow sequential covering approach where a rule is generated and prunes simultaneously without considering its effect on other rules.

We proposed a new global rule pruning strategy that can be applied during pruning step in ACO algorithm for rules classification. We considered the association between the terms in the rules and applied a global rule pruning procedure that takes into account effect of current rule generated on the subsequent rules.

We conducted our experiment on the publically available dataset from UCI learning data repository. Experimental results showed that our proposed rule pruning procedure for pruning step in ACO for rule's classification is competitive to others algorithms in terms of number of rules generated and number of terms per rule.

6.3 FUTURE WORK

There are several key directions for future exploration. These directions are summarized here as under:

1. The proposed approach can be applied to other dataset that have more number of attributes and then analyzing the results.
2. Changes can be made to heuristic function in order to investigate the performance of the proposed approach.
3. Different pheromone updating strategy can be formulated and applied to check the performance of the proposed approach.
4. Meta-learning techniques [27] can be used to find the relationship between the datasets characteristics and is also one of the interesting areas for future research.

REFERENCES

[1] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “From data mining to knowledge discovery: An overview,” in *Advances in Knowledge Discovery & Data Mining*, U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds. Cambridge, MA: MIT Press, 1996, pp. 1–34.

[2] A.Freitas and S. H. Lavington, *Mining Very Large Databases with Parallel Processing*. Norwell, MA: Kluwer, 1998.

[3] Jaiwei, H., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann, San Francisco (2006)

[4] Agarwal C.C. and P.S. Yu., “Data Mining Techniques for Associations, Clustering and Classification”, Proceedings of the Third Pacific-Asia Conference, PAKDD-99, Beijing, China, April, 1999.

[5] Agrawal R., Imielinski and T.Swami,” A: Database mining: A Performance Perspective”, IEEE Trans. Knowledge and Data Engineering, 5(6): 914-925, 1993

[6] Usama Fayyad, Padhraic Smyth, Gregory,” The KDD process for extracting useful knowledge from volumes of data”, ACM,Volume 39 Issue 11, Nov. 1996 Pages 27-34

[7] Diamantini, Claudia; Potena, Domenico; Storti, Emanuele, "A platform for collaborative and distributed KDD process design," *Collaboration Technologies and Systems (CTS), 2012 International Conference on* , vol., no., pp.628,629, 21-25 May 2012

[8] Jun Zhang, Zhijian Wang, Shuren Zhu and Xiaoming Meng,” The Application of KDD in HIS”, Yuan Bo Qin et al., 2012, Applied Mechanics and Materials, 263-266, 1510

[9] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. NewYork: Wiley, 2000.

[10] D. J. Hand and S. D. Jacka, Discrimination and Classification. NewYork: Wiley, 1981

[11] D. J. Hand, “Pattern detection and discovery,” in Pattern Detection and Discovery, ser. Lecture Notes in Computer Science, D. J. Hand, N. Adams, and R. Bolton, Eds. Berlin, Germany: Springer-Verslag, 2002, vol. 2447, pp. 1–12.

[12] B. Baesens, R. Setiono, C. Mues, and J. Vanthienen, “Using neural network rule extraction and decision tables for credit-risk evaluation,” Manage. Sci., vol. 49, no. 3, pp. 312–329, 2003

[13] B. Baesens, “Developing intelligent systems for credit scoring using machine learning techniques,” Ph.D. dissertation, K.U. Leuven, Leuven, Belgium, 2003.

[14] R. S. Parpinelli, H. S. Lopes, and A. A. Freitas, “An ant colony based system for data mining: Applications to medical data,” in Proc. Genetic and Evol. Comput. Conf., 2001, pp. 791–797.

[15] Ziqiang Wang, Xia Sun, and Dexian Zhang,” Classification Rule Mining Based on Particle Swarm Optimization” G. Wang et al. (Eds.): RSKT 2006, LNAI 4062, pp. 436–441, Springer-Verlag, 2006

[16] Wikipedia.org/evolutionary algorithm

[17] D.E. Goldberg, Genetic algorithms in search, optimiazation and machine learning, Addison- Wesley, Reading, MA, 1989.

[18] Kermani, B.G.; White, M.W.; Nagle, H.T., "Feature extraction by genetic algorithms for neural networks in breast cancer classification," *Engineering in Medicine and Biology Society, 1995., IEEE 17th Annual Conference* , vol.1, no., pp.831,832 vol.1, 20-25 Sep 1995

[19] Kannan, A.; Maguire, G.Q.; Sharma, A.; Schoo, P., "Genetic Algorithm Based Feature Selection Algorithm for Effective Intrusion Detection in Cloud Networks," *Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on* , vol., no., pp.416,423, 10-10 Dec. 2012

[20] Zhou Yuanhui; Lu Yuchang; Shi Chunyi, "Combining neural network, genetic algorithm and symbolic learning approach to discover knowledge from databases," *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on* , vol.5, no., pp.4388,4393 vol.5, 12-15 Oct 1997

[21] Kromer, P.; Platos, J.; Snasel, V.; Abraham, A., "Fuzzy classification by evolutionary algorithms," *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on* , vol., no., pp.313,318, 9-12 Oct. 2011

[22] Bozorgnia, Arezoo; Zargar, Samaneh Hajy Mahdizadeh; Yaghmaee, Mohammad H., "Fuzzy improved genetic k-means algorithm," *Electrical Engineering (ICEE), 2011 19th Iranian Conference on* , vol., no., pp.1,6, 17-19 May 2011

[23] Bozorgtabar, B.; Noorian, F.; Rezai Rad, G.A., "A Genetic Programming approach to face recognition," *GCC Conference and Exhibition (GCC), 2011 IEEE* , vol., no., pp.194,197, 19-22Feb.2011doi: 10.1109/IEEEGCC.2011.5752477

[24] Dash, Satya Ranjan; Dehuri, Satchidananda; Rayaguru, Susil, "Discovering interesting rules from biological data using parallel genetic algorithm," Advance Computing Conference (IACC), 2013 IEEE 3rd International , vol., no., pp.631,636, 22-23 Feb. 2013

[25] Al-Maqaleh, B.M.A., "Genetic Algorithm Approach to Automated Discovery of Comprehensible Production Rules," *Advanced Computing & Communication Technologies (ACCT), 2012 Second International Conference on* , vol., no., pp.69,71, 7-8 Jan. 2012

[26] Dutta, D.; Dutta, P.; Sil, J., "Clustering by multi objective genetic algorithm," *Recent Advances in Information Technology (RAIT), 2012 1st International Conference on* , vol., no., pp.548,553, 15-17 March 2012

[27] Kennedy, J. and Eberhart, R. C. “Particle swarm optimization” Proceedings of IEEE International Conference on Neural Networks Vol. IV, IEEE service center, Piscataway, NJ, Nov. 27- Dec. 1, 1995, pp. 1942-1948.

[28] Nadig, A.S.; Potetz, B., "A hierarchical Bayesian model for pattern recognition," *Neural Networks (IJCNN), The 2012 International Joint Conference on* , vol., no., pp.1,8, 10-15 June 2012

[29] Merhav, N.; Ephraim, Y., "A Bayesian classification approach with application to speech recognition," *Signal Processing, IEEE Transactions on* , vol.39, no.10, pp.2157,2166, Oct 1991

[30] Heung-Il Suk; Seong-Whan Lee, "A Novel Bayesian Framework for Discriminative Feature Extraction in Brain-Computer Interfaces," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* , vol.35, no.2, pp.286,299, Feb. 2013

[31] Ghorpade, T.; Ragha, L., "Featured based sentiment classification for hotel reviews using NLP and Bayesian classification," *Communication, Information & Computing Technology (ICCICT), 2012 International Conference on* , vol., no., pp.1,5, 19-20 Oct. 2012

[32] Ben Othman, Ibtissem; Ghorbel, Faouzi, "A new criteria for comparing neural networks and Bayesian classifier," *Computer Applications Technology (ICCAT), 2013 International Conference on* , vol., no., pp.1,5, 20-22 Jan. 2013

[33] Viswanath, P.; Sarma, T.H., "An improvement to k-nearest neighbor classifier," *Recent Advances in Intelligent Computational Systems (RAICS), 2011 IEEE* , vol., no., pp.227,231, 22-24 Sept. 2011

[34] Islam, M.J.; Wu, Q. M J; Ahmadi, M.; Sid-Ahmed, M.A., "Investigating the Performance of Naive- Bayes Classifiers and K- Nearest Neighbor Classifiers," *Convergence Information Technology, 2007. International Conference on* , vol., no., pp.1541,1546, 21-23 Nov. 2007

[35] Bastos-Filho, T.; Tello, R.M.G.; Arjunan, S.; Shimada, H.; Kumar, D., "Real-time control of finger and wrist movements in a virtual hand using traditional features of semg and Bayesian classifier," *Biosignals and Biorobotics Conference (BRC), 2013 ISSNIP* , vol., no., pp.1,5, 18-20 Feb. 2013

[36] Mohanapriya, S.; Vadivel, M., "Automatic retrival of MRI brain image using multiqueries system," *Information Communication and Embedded Systems (ICICES), 2013 International Conference on* , vol., no., pp.1099,1103, 21-22 Feb. 2013

[37] Camejo, Jorge; Pacheco, Osvaldo; Guevara, Miguel, "Classifier for drinking water quality in real time," *Computer Applications Technology (ICCAT), 2013 International Conference on* , vol., no., pp.1,5, 20-22 Jan. 2013

[38] Farooqi, A.H.; Munir, A., "Intrusion Detection System for IP Multimedia Subsystem using K-Nearest Neighbor classifier," *Multitopic Conference, 2008. INMIC 2008. IEEE International* , vol., no., pp.423,428, 23-24 Dec. 2008

[39] Sriram, S.; Xiaobu Yuan, "An enhanced approach for classifying emotions using customized decision tree algorithm," *Southeastcon, 2012 Proceedings of IEEE* , vol., no., pp.1,6, 15-18 March 2012

[40] Liu Qin, "Improved Decision Tree Method for Imbalanced Data Sets in Digital Forensics," *Computational and Information Sciences (ICCIS), 2012 Fourth International Conference on* , vol., no., pp.251,254, 17-19 Aug. 2012

[41] Thanasoontornrerk, Rattiya; Vateekul, Peerapon; Kanongchaiyos, Pizzanu; Pongmala, Chatkaew; Suputtitada, Areerat, "Tree induction for diagnosis on movement disorders using gait data," *Knowledge and Smart Technology (KST), 2013 5th International Conference on* , vol., no., pp.47,52, Jan. 31 2013-Feb. 1 2013

[42] Boughrara, H.; Chtourou, M.; Amar, C.B., "MLP neural network based face recognition system using constructive training algorithm," *Multimedia Computing and Systems (ICMCS), 2012 International Conference on* , vol., no., pp.233,238, 10-12 May 2011

[43] Amaral, T.G.; Dias, O.P.; Wolczowski, A.; Pires, V.F., "Neural network based identification of hand movements using biomedical signals," *Intelligent Engineering Systems (INES), 2012 IEEE 16th International Conference on* , vol., no., pp.125,129, 13-15 June 2012

[44] Krstic, M.; Bjelica, M., "Personalized TV program guide based on neural network," *Neural Network Applications in Electrical Engineering (NEUREL), 2012 11th Symposium on* , vol., no., pp.227,230, 20-22 Sept. 2012

[45] Yang Jing; Wang Liang, "An improved star identification method based on neural network," *Industrial Informatics (INDIN), 2012 10th IEEE International Conference on* , vol., no., pp.118,123, 25-27 July 2012

[46] Hemageetha, N.; Nasira, G.M., "Radial basis function model for vegetable price prediction," *Pattern Recognition, Informatics and Medical Engineering (PRIME), 2013 International Conference on* , vol., no., pp.424,428, 21-22 Feb. 2013

[47] Eiamkanitchat, N.; Theera-Umpon, N.; Auephanwiriyakul, S., "A novel neuro-fuzzy method for linguistic feature selection and rule-based classification," *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on* , vol.2, no., pp.247,252, 26-28 Feb. 2010

[48] Almendros-Jimenez, J.M.; Domene, L.; Piedra-Fernandez, J.A., "A Framework for Ocean Satellite Image Classification Based on Ontologies," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of* , vol.6, no.2, pp.1048,1063, April 2013

[49] Yunxiao Luo; Shuhe Zhao; Hongkui Zhou; An Wang; Kexun He; Lei Tan, "A novel classification method based on texture analysis using high-resolution SAR and optical data," *Earth Observation and Remote Sensing Applications (EORSA), 2012 Second International Workshop on* , vol., no., pp.186,190, 8-11 June 2012

[50] Chen, Ching-Yi; Fun Ye, "Particle swarm optimization algorithm and its application to clustering analysis," *Networking, Sensing and Control, 2004 IEEE International Conference on* , vol.2, no., pp.789,794 Vol.2, 2004

[51] Van Der Merwe, D. W.; Engelbrecht, A.P., "Data clustering using particle swarm optimization," *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on* , vol.1, no., pp.215,220 Vol.1, 8-12 Dec. 2003

[52] Xiaohui Cui; Potok, T.E.; Palathingal, P., "Document clustering using particle swarm optimization," *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE* , vol., no., pp.185,191, 8-10 June 2005

[53] M. OMRAN, A. P. ENGELBRECHT, and A. SALMAN*, "*Particle Swarm Optimization Method For Image Clustering*", Int. J. Patt. Recogn. Artif. Intell.* **19**, 297 (2005).

[54] Nicholas Holden, Alex A. Freitas," A hybrid PSO/ACO algorithm for discovering classification rules in data mining", Journal of Artificial Evolution and Application, Volume 2008, Jan 2008.

[55] Tiago Sousa, Arlindo Silva, Ana Neves,"Particle swarm based data mining algorithm for classification task", parallel computing, volume 30, issu 5-6, May-June 2004, Pages 767-783.

[56] Ziqiang Wang, Xia Sun, Dexian Zhang,"Classification rule mining based on particle swarm optimization", vol. 4062, 2006, pp 436-441.

[57] R.J. Kuo, C.M. Chao, Y.T. Chiu, Application of particle swarm optimization to association rule mining, Applied Soft Computing, Volume 11, Issue 1, January 2011, Pages 326-336, ISSN 1568-4946.

[58] M. Dorigo, G. Di Caro, and L.M. Gambardella, "Ant algorithms for discrete optimization," *Artificial Life*, vol. 5, no. 2, pp. 137–172, 1999.

[59] M. Dorigo and G. Di Caro, "The ant colony optimization meta-heuristic," in *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds. New York: McGraw-Hill, 1999, pp. 11–32.

[60] V. Cern´y, “A thermodynamical approach to the traveling salesman problem,” *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41–51, 1985

[61] S. Kirkpatrick, C.D. Gelatt Jr., and M.P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, pp. 671–680, 1983.

[62] F. Glover, “Tabu search—part I,” *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190–206, 1989.

[63] ——, “Tabu search—part II,” *ORSA Journal on Computing*, vol. 2, no. 1, pp. 4–32, 1990.

[64] F. Glover and M. Laguna, *Tabu Search*, Kluwer Academic Publishers, 1997.

[65] H.R. Lourenço, O. Martin, and T. St¨utzle, “Iterated local search,” in *Handbook of Metaheuristics, ser. International Series in Operations Research & Management Science*, F. Glover and G. Kochenberger, Eds., Kluwer Academic Publishers, vol. 57, pp. 321–353, 2002.

[66] L.J. Fogel, A.J. Owens, and M.J. Walsh, *Artificial Intelligence Through Simulated Evolution*, John Wiley & Sons, 1966.

[67] J. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor: University of Michigan Press, 1975.

[68] I. Rechenberg, *Evolutionsstrategie—Optimierung technischer Systeme nach Prinzipien der biologischen Information*, Fromman Verlag, Freiburg, Germany, 1973.

[69] H.-P. Schwefel, *Numerical Optimization of Computer Models*. John Wiley & Sons, 1981.

[70] M. Dorigo and T. St¨utzle, *Ant Colony Optimization*, MIT Press, Cambridge, MA, 2004.

[71] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant System: Optimization by a colony of cooperating agents,” *IEEE Transactions on Systems, Man, and Cybernetics—Part B*, vol. 26, no. 1, pp. 29–41, 1996.

[72] R. Parpinelli, H. Lopes, and A. Freitas, “Data mining with an ant colony optimization algorithm,” IEEE Trans. Evol. Computat., vol. 6, no. 4, pp. 321–332, Aug. 2002.

[73] A. Colorni, M. Dorigo, V. Maniezzo, and M. Trubian, “Ant system for job-shop scheduling,” *J. Oper. Res., Stat., Comput. Sci.*, vol. 34, no. 1,pp. 39–53, 1994.

[74] C. Blum, “Beam-ACO—hybridizing ant colony optimization with beam search: An application to open shop scheduling,” *Comput. Oper. Res.*, vol. 32, no. 6, pp. 1565–1591, 2005.

[75] T. Stützle and H. H. Hoos, “MAX-MIN ant system,” *Future Generation Comput. Syst.*, vol. 16, no. 8, pp. 889–914, 2000.

[76] L. M. Gambardella and M. Dorigo, “Ant-Q: A reinforcement learning approach to the traveling salesman problem,” in *Proc. 12th Int. Conf. Mach. Learn.*, A. Prieditis and S. Russell, Eds., Palo Alto, CA, 1995, pp. 252–260.

[77] K. Socha, J. Knowles, and M. Sampels, “A MAX-MIN ant system for the university timetabling problem,” in *Proc. 3rd Int. Workshop on Ant Algorithms*, M. Dorigo, G. Di Caro, and M. Sampels, Eds., Sep. 2002, vol. 2463, pp. 1–13, Lecture Notes in Computer Science.

[78] A. Wade and S. Salhi, “An ant system algorithm for the mixed vehicle routing problem with backhauls,” in *Metaheuristics: Computer Decision-Making*. Norwell, MA: Kluwer, 2004, pp. 699–719.

[79] B. Bullnheimer, R. Hartl, and C. Strauss, “Applying the ant system to the vehicle routing problem,” in *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, S. Voss, S. Martello, I. Osman, and C. Roucairol, Eds. Norwell, MA: Kluwer, 1999, pp. 285–296.

[80] R. Montemanni, L. M. Gambardella, A. E. Rizzoli, and A. Donati, “Ant colony system for a dynamic vehicle routing problem,” *J. Combinatorial Optim.*, vol. 10, no. 4, pp. 327–343, 2005.

[81] G. Di Caro and M. Dorigo, “Antnet: Distributed stigmergetic control for communications networks,” in *J. Artif. Intell. Res.*, 1998, vol. 9, pp. 317–365.

[82] A. Abraham and V. Ramos, “Web usage mining using artificial ant colony clustering,” in *The Congress on Evolutionary Computation*. Piscataway, NJ: IEEE Press, 2003, pp. 1384–1391.

[83] J. Handl, J. Knowles, and M. Dorigo, “Ant-based clustering and topographic mapping,” *Artif. Life*, vol. 12, no. 1, pp. 35–61, 2006.

[84] R. S. Parpinelli, H. S. Lopes, and A. A. Freitas, “An ant colony based system for data mining: Applications to medical data,” in *Proc. Genetic and Evol. Comput. Conf.*, 2001, pp. 791–797.

[85] B. Liu, H. A. Abbass, and B. McKay, “Density-based heuristic for rule discovery with ant-miner,” in *Proc. 6th Australasia-Japan Joint Workshop on Intell. Evol. Syst.*, Canberra, Australia, 2002, pp. 180–184.

[86] ——, “Classification rule discovery with ant colony optimization,” in *Proc. IEEE/WIC Int. Conf. Intell. Agent Technol.*, 2003, pp. 83–88.

[87] R. S. Parpinelli, H. S. Lopes, and A. A. Freitas, “Data mining with an ant colony optimization algorithm,” *IEEE Trans. Evol. Comput.*, vol.6, no. 4, pp. 321–332, Aug. 2002.

[88] J. R. Quinlan, “Generating production rules from decision trees,” in Proc. Int. Joint Conf. Artificial Intelligence, San Francisco, CA, 1987, pp. 304–307.

[89] H. S. Lopes, M. S. Coutinho, and W. C. Lima, “An evolutionary approach to simulate cognitive feedback learning in medical domain,” in Genetic Algorithms and Fuzzy Logic Systems: Soft Computing Perspectives, E. Sanchez, T. Shibata, and L.Zadeh, Eds, Singapore: World Scientific, 1998, pp. 193–207

[90] F. Otero, A. Freitas, and C. Johnson, “Handling continuous attributes in ant colony classification algorithms,” in Proc. IEEE Symp. CIDM, Mar.–Apr. 2009, pp. 225–231.

[91] F. Otero, A. Freitas, and C. Johnson, “*c*Ant-Miner: An ant colony classification algorithm to cope with continuous attributes,” in *Proc. 6th Int. Conf. Swarm Intell. (ANTS)*, vol. 5217. 2008, pp. 48–59.

[92] K. Salama and A. Abdelbar, “Extensions to the Ant-Miner classification rule discovery algorithm,” in Proc. 7th Int. Conf. Swarm Intell. (ANTS), vol. 6234. 2010, pp. 167–178.

[93] D. Martens, M. D. Backer, R. Haesen, J. Vanthienen, M. Snoeck, and B. Baesens, “Classification with ant colony optimization,” IEEE Trans. Evol. Computat., vol. 11, no. 5, pp. 651–665, Oct. 2007

[94] Fernando E. B. Otero, Alex A. Freitas, and Colin G. Johnson,"A New Sequential Covering Strategy for Inducing Classification Rules With Ant Colony Algorithms", IEEE Trans. Evol. Computat., Vol. 17, no. 1, pp. 64-76, Feb 2013

[95] K. Salama, A. Abdelbar, and A. Freitas, “Multiple pheromone types and other extensions to the ant-miner classification rule discovery algorithm,” Swarm Intell., vol. 5, nos. 3–4, pp. 149–182, 2011.

[96] B. Liu, H. Abbass, and B. McKay, “Classification rule discovery with ant colony optimization,” in *Proc. IEEE/WIC Int. Conf. IAT*, Oct. 2003, pp. 83–88.

[97] A. Chan and A. Freitas, “A new classification-rule pruning procedure for an ant colony algorithm,” in Proc. EA, vol. 3871. 2005, pp. 25–36.

[98] Deborah R. Carvalho and A.A. Freitas, “A hybrid decision tree/genetic algorithm method for data mining” Information Sciences 163(1-3), pp. 13-35. June 2004.

[99] I. H. Witten and E. Frank, Data Mining – Pratical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann 2000.

[100] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. NewYork: Wiley, 1991.

[101] A. Asuncion and D. Newman. (2010). *UCI Machine Learning Repository*, School Inform. Comput. Sci, Univ. California, Irvine [Online]. Available: http://www.ics.uci.edu/$\sim$mlearn/{MLR}epository.html

[102] U. Fayyad and K. Irani, “On the handling of continuous-valued attributes in decision tree generation,” *Mach. Learn.*, vol. 8, no. 1, pp. 87–102, Jan. 1992.

[103] P. Clark and T. Niblett, “The CN2 rule induction algorithm,” *Mach. Learn.*, vol. 3, no. 4, pp. 261–283, 1989.

[104] J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, and B. Baesens, “An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models,” *Decision Support Syst.*, vol. 51, no. 1, pp. 141–154, 2011.