Iqra University IUIC
Islamabad Campus

COMPREHENSIVE RULES DISCOVERY USING PARTICLE SWARM OPTIMIZATION FOR BACK PAIN DIAGNOSIS

By

Hasan Kamal

(Registration No. 051-10-13010)

Supervised by

Dr. Hajira Jabeen

Assistant Professor, Iqra University, Islamabad Campus

Thesis Submitted as partial fulfillment of requirement for the Degree of Master of Science in Computer Science (MSCS)

Department of Computing and Technology

Iqra University Islamabad Campus

January, 2013

ABSTRACT

Particle swarm optimization (PSO) algorithm has been found efficient for many optimization tasks. PSO has an ability to search for the optimal solution using less particles and iterations as compared to other evolutionary algorithms. A new approach is proposed in this thesis which uses PSO to discover classification rules for real valued data for back pain diagnosis. The resulting rules are comprehensible and offer better generalization abilities. The performance of the proposed rules based classification algorithm have been tested on a data set which is downloaded from a well know repository i.e. UCI Machine Learning Repository considering six attributes derived from the shape and orientation of the pelvis and lumbar spine to classify orthopedic patients into normal and abnormal category. Where, the abnormal patients suffer from lower back pain, which is a common concern that affects most of the people at some point in their life. The proposed algorithm offers transparent-understandable rules that are discovered from the data in its original form. The algorithm is compared with neural network, support vector machine, decision tree, k-nearest neighbor. The classification results are found encouraging.

ACKNOWLEDGEMENTS

First, I am grateful to ALLAH by whose grace I have been able to complete this thesis successfully. I would love to express my deepest appreciation and sincerest gratitude to my supervisor Dr. Hajira Jabeen for her inspiring guidance, remarkable suggestions, keen interest, constant encouragement and friendly discussions which enabled me to complete this thesis. I am grateful to her for the co-operation and support she generously gave me when I needed it most. I owe success to my parents, sisters and brother who are the real source of my encouragement and invaluable guidance that made me capable of all what I am today.

I would also like to thank my friends for encouraging and supporting me to embark on my research.

DECLARATION

I hereby declare that this research, neither as whole nor as part has been copied out from any source. It is further declare that I have prepared this report entirely on the basis of my personal efforts made under the sincere guidance of teachers especially my supervisor Dr. Hajira Jabeen. If any part of this thesis is proved to be copied out from any source or found to be reproduction of some other, I will stand by the consequences. No potion of the work presented has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Hasan Kamal
(MS (CS), 13010)

DEDICATION

I dedicate this thesis to my parents who always inspired and encouraged me.

THESIS APPROVAL SHEET

It is certify that Hasan Kamal of MS (CS) Department of Computer & Technology, Student ID (13010) of IQRA University Islamabad, has submitted the final Thesis report on "**Comprehensive Rules Discovery Using Particle Swarm Optimization for Back Pain Diagnosis**". We have read the report and it fulfills the partial of Master of Science in Computer Science degree.

INTERNAL EXAMINER:

Name:

Designation:

Organization:

Signature: ______________________________

EXTERNAL EXAMINER:

Name:

Designation:

Organization:

Signature: ______________________________

SUPERVISOR:

Name: Dr. Hajira Jabeen

Designation: Assistant Professor

Organization: Iqra University, Islamabad Campus

Signature: ______________________________

Table of Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 Introduction

Lower back pain is a common problem. Low back pain is not a specific disease, but a symptom that occur because of various activities in our everyday life. A research conducted on 1221 peoples concluded that 368 patients (30.1%) had never experienced low back pain, 565 (46.3%) had moderate or low back pain, and 288 (23.6%) suffered from chronic low back pain [1]. Patients with chronic low back pain complaints significantly higher disorders in the lower spine (lower vertebral column). They require more medical care and treatment resulting in the loss of time and work. Mostly the victims of the lower back pain are associated with the jobs requiring repeated heavy lifting, use of heavy machine tools or jackhammers and motor vehicle operations [2].

Back pain may have many causes, but most probably is the impairment in the vertebral column. The two major diseases of the vertebral column are [3]:

- Disk Hernia or Spinal Disk herniation.
- Spondylolisthesis.

1.1.1 Disk hernia or spinal disk herniation:

Normally the cause of lower back and leg pain is disk herniation [3]. Almost 60% to 80% of people having the lower back and leg pain are because of disk herniation. The resulted pain is of high severity however after a few months of treatment, most of the people feel extremely comfortable and relieved.

1.1.2 Spondylolisthesis

In Spondylolisthesis the bones or vertebrates in the vertebral column get dislocated. It resulted in only two conditions; the upper vertebral move forward over the sacrum or it moves backwards. In both condition the spinal nerves are pinched by the vertebral. This causes severe lower back pain.

The population health of a country, which is mainly based on the access and availability of medical facilities, has a strong impact on its accumulative growth. Disease diagnosis is the major and critical task in medical domain. Accurate and timely diagnosis of the disease can certainly increase the chances of recovery. It is a difficult task and demands certain level of expertise. Carrying out proper and correct diagnosis and interpretation of data so obtained is considered as the most important aspect of medical care. However, due to the time constraint it is very challenging for the physician to sift through the patient history or data in a compressed time environment. Here the intelligent tools play a vital role by helping or assisting the physician to take smart and quick decisions in time constrained environment. In this scenario, expert systems for disease diagnosis could perform a major role in medical diagnosis, prediction and classification.

1.2 Data Classification

Data classification is a two-step process which categorizes it using the available data for its most effective and efficient use.

In the first step, a model is built describing a pre-determined set of data classes. The model is constructed by analyzing the set of examples belonging to a predefined class. In case of classification, the samples or examples are also referred to as data tuples. The set of examples constructs a training data set which analyzed to build a model. From the data set a set of examples are randomly selected to make training set and such individuals examples are called training examples. Typically, the learned model is represented in the form of classification rules, decision trees, neural networks etc.

In the second step the model is used for classification. To determine the authenticity of the classification, the predictive accuracy of the model is estimated on a separate test data set. It is due to the reason that if the accuracy of the model is estimated based on the training data set, the estimate could be positive since the learned model tends to over fit the data. If the accuracy of the model is considered acceptable, the model can be used to classify future data tuples for which the class level is not known.

Categorized data can be used in different domains like medicine, business, engineering etc. Interest in data classification is fast growing in order to deal with the prohibitive amount of information we encounter in our daily life [2].

1.2.1 Traditional classification techniques

Traditional classifiers refer to the two classification techniques namely unsupervised and supervised classification.

1.2.1.1 Unsupervised learning

Unsupervised learning of a machine is trying to find hidden structure of the problem in unlabeled data. However learning of machine using unlabeled examples, means no reward or slip signal to determine a possible solution. It is a prominent difference between unsupervised and supervised learning.

1.2.1.2 Supervised learning

Supervised classification is defined by the application of prior information of classes to determine the identity of unknown data. The training data are composed of a set of training examples. In supervised learning, each instance is used as an input value (attributes) and the desired output value pair itself to the class i.e. make a subset of the class (belongs to class). A supervised learning algorithm analyzes the training data and produces a classifier. The classifier for any legal input value should be able to predict the correct class. The learning algorithm needs to be extended so that it can be generalize for un-seen scenarios in a reasonable manner.

1.2.2 Traditional classifiers

Some of the most commonly used data classification techniques are:

1.2.2.1 Artificial neural networks

Artificial neural networks (ANN) work on the principle of the human brain neurophysiology. ANN has ability to learn with the help of examples for data classification or in order to find samples in the data. Once trained on the training data, they have ability to forecast for un-see data. They perform global search on data, but have deficiency of clear representation (black box representation) used to classify the data. It also lacks in the understanding of the basic principles of user data which is used for categorizing. Error back propagation is the most commonly used method in neural network in learning process.

1.2.2.2 Decision trees

Decision trees are the tree-shaped structures representing decision rules for classifying data set problems. “Divide and conquer” approach is used for construction of decision trees. Gradually smaller and smaller sub-set of training data is created, so the subset contains the maximum extent possible elements of same class. In each phase of the algorithm for a tree node an attribute is selected. It is intended to select the attribute subsets that will best highlight the underlay classified class.

1.2.2.3 Support vector machine (SVM)

Support vector machines work in the space of possible inputs by finding a hyper-surface between them. This hyper-surface class will attempt to distribute a Class A by Class B. The largest class of distributed hyper-surface level will be chosen for the examples of the Class A and Class B for closest distance [4]. Intuitively, this classification is close to the test data which is correct, but not identical to training data.

1.3 Data Classification Using PSO

In this thesis we light up a new horizon for use of well know and regimented optimization method, known as Particle Swarm Optimization (PSO), in the domain of data classification. PSO generally helps to solve optimization problems by simulating the fish schooling, bird flocking, and insect swarming.

1.4 Motivation

This thesis discusses the opportunities for extracting rules from the given data (or databases) for the better decision making in medical domain. The goal is to propose a transparent rule based classifier with acceptable accuracy. The simplicity of the rules, uniformity and transparency makes it an appropriate tool of real-world representation of medical information. Traditional methods of rule mining produce a large number of rules with many conditions which makes the system unusable over medical data [5].

Besides rule extraction another objective is feature extraction to facilitate the physician to have the knowledge about the most contributing attribute. For the simplicity of rules we use SQL queries as rule set. The other advantages of using SQL queries include their readability for non-experts and ease of integration with existing databases [6]. By introducing SQL queries in the system one makes it compatible with other systems having the capability to couple with SQL databases. This increases it’s portability.

For finding the classifier with discussed characteristics there are many evolutionary algorithms (EA) techniques like Differential Evolution (DE), Genetic Algorithm (GA), or Particle Swarm Optimization (PSO). We have used PSO due to its convergence speed, robustness and good performance of global search on a large and noisy data. Regarding convergence speed, PSO is always the fastest [7].

1.5 Objectives

The primary objectives of this thesis are:

- To develop a new technique of rule extraction for data classification using PSO.
- To propose new transparent classifier.
- To determine classifier which also helpful in feature extraction.
- No pre/post processing of the data.
- To compare our classification results with different evolutionary algorithms and ANN.

1.6 Contribution

The major contribution of this thesis is the new methodology for rule extraction using Particle swarm optimization. Generating a transparent classifier with pronounced ability of feature extraction.

1.7 Overview of the Thesis

The thesis is organized as follows:

Chapter 2 gives the background of the disease. Furthermore it includes literature survey of the different traditional classification techniques used for data classification.

Chapter 3 contains detailed discussion on PSO. Furthermore it includes literature survey of PSO used for classification task.

Chapter 4 explains the methodology used for the diagnosis and classification of disease. Approach used in implementation of algorithm; SQL queries; structure of the data set.

Chapter 5 provides the discussion over the experimental results. It also contains the comparison between the results of proposed technique with the other well-known existing techniques.

Finally, Chapter 6 concludes the thesis. Limitations and future work is discussed. Possible extensions of the framework are also proposed.

CHAPTER 2

LITERATURE REVIEW

A disease is a condition of the body in which its parts loses their normal functions. More commonly it may be caused by external factors, such as infectious disease. Every living organism can be victimized by diseased. There are hundreds of different diseases [8]. Disease is a medical disorder with specific symptoms and signs [9]. Each of the symptoms and signs, assist physician in diagnostics of the disease. A symptom such as fever, bleeding, or pain can be complained by the patient. A sign of inflammation of the blood vessel or something like an extension of the internal organs of the body can be detect by the physician. Every disease has a cause, although some factors may be hidden.

Lower back pain (LBP) is one of the major symptom of disorder of vertebral column [10]. LBP presents a huge challenge to the healthcare system despite of the improvement in scientific technology and medical sciences. Greater insight has been gained on what not to do with patients with LBP rather than what to do for them. When LBP lasts longer than 3 months it is classified as chronic low back pain [11].

2.1 Diagnosis of Vertebral Column Pathologies

The vertebral column or some time refers as spinal column is a system comprising a set of vertebrae, intervertebral discs, nerves, muscles, bone and ligaments. According to Guanabara [10], the main functions of the vertebral column are as follows:

- The support shaft of the human body.
- Protective bone of the spinal cord and nerve roots.
- Shaft movement of the body, allowing movement in three planes: frontal, sagittal and transverse.

This complex system sometimes is subjected to inefficiencies that cause lower back pain, with most varied intensities [12]. Hernia and Spondylolisthesis disk are all examples of vertebral pathologies causing severe lower back pain.

2.1.1 Disk hernia or spinal disk herniation

About 60 to 80% cause of lower back pain and leg pain in people is due to disc herniation. It results a high degree of pain, however after a few months of treatment, most people feel very relieved [13].

A herniated disc arises as a result of various minor disturbances to vertebral column that will occur over time, damaging the structures of intervertebral disc, or can occur as a result of severe disturbance in the spine. Actually our spinal cord is made of 33 vertebras and there is a cushion between them to sustain external jerks and pressure. Also spinal nerve passes through one side of spinal cord or vertebral column (spinal cord is made of vertebrates). When the cushion is moved from its original position there are good chances that it exerts pressure on the spinal nerves which causes pain; in fact there are not enough fringes between spinal cord and spinal nerves to accommodate this dislocation of the cushion (or some time named as herniated disk) [12]. Figure 2.1 and 2.2 shows the vertebral column with disorder of disk herniation.

Figure 2.1 Disk herniation in vertebral column [14]

Spinal Disk herniation in the core intervertebral is either migration of herniated disc to the center of the vertebral column or the margin toward the spinal canal or in the spaces where the nerve roots exists, leading to compression of nerve roots. Some of the symptoms are:

- Lower back pain.
- Leg/foot pain.
- Numbness in the leg/foot.
- Weakness in the leg/foot.

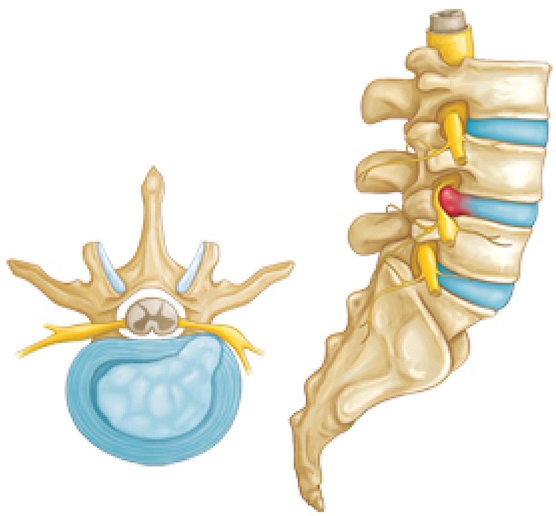

Figure 2.2 Disk herniation in vertebral column [14]

2.1.2 Spondylolisthesis

Spondylolisthesis occurs when one of the 33 vertebrae spinal slides relative to the other. In Spondylolisthesis bones or vertebrae becomes inapt in the spine. There are only two possibilities, top vertebrae dislocated either to ahead or behind of the sacrum. In both the cases spinal nerves are pinched by vertebrae. This will cause serious pain [3]. Figure 2.3 shows the X-Ray of the patient having low back pain because of Spondylolisthesis.

It is generally observed that when slip occurs towards the base of the spine in the lumbar region causes pain or symptoms of nerve root irritation [12]. The mechanism which causes this type of injury is not well known, however there are theories that suggest some possible reasons which are:

- Fracture fatigue conjugated to an inherited defect or predisposition
- Fracture occurred during childbirth.
- Displacement of a vertebra of the other secondary lumbar lordosis.
- Weak ligaments and structures fasceais the region involved.
- Malformation of the articular facets.

Some of the symptoms are:

- Lower back pain.
- Tightness of muscles.
- Buttocks and thighs pain.

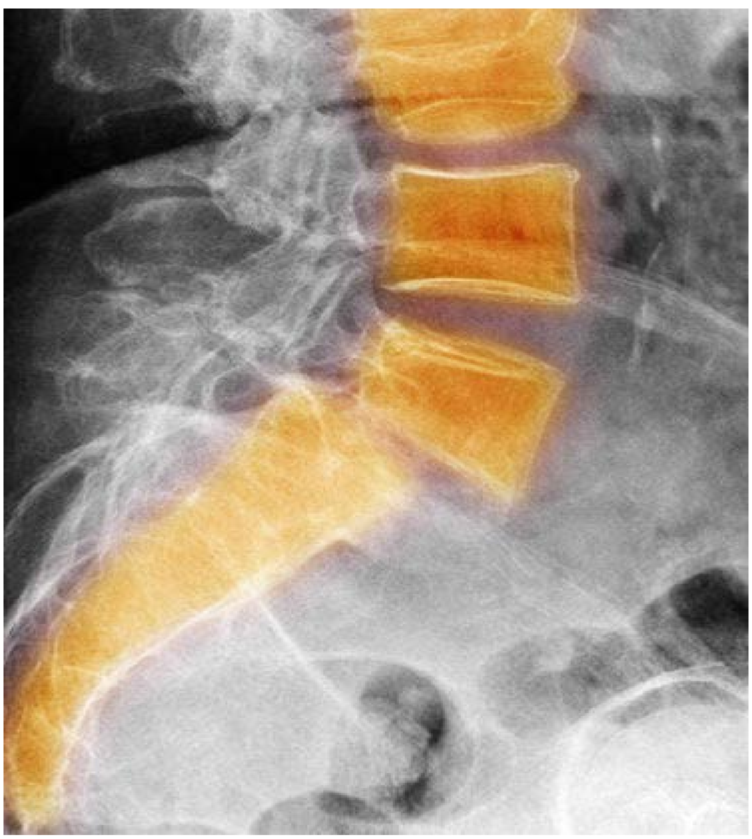

Figure 2.3 Vertebral Column having disorder of Spondylolisthesis [15]

2.2 Traditional Classification Techniques

2.2.1 Artificial neural networks

Artificial neural network (ANN) classifiers are one of the oldest techniques for classification [16]. They are generally used to find unknown patterns in the data and to model the complex relationship between the inputs and outputs. ANNs have been developed by generalizing mathematical models which copy the human brain neurophysiology [17]. ANNs model performs the particular task in the way in which brain performs a particular task. The neural networks employ massive interconnection of simple computing cells called neurons. These neurons are arranged in a number of layers. Each connection link has a weight. Each neuron sums up the inputs along with its weights and applies an activation function (used to control the amplitude of the output) to determine its output. The neural network performs useful computation through the learning or training process. The weights are adjusted when data are presented to the network during a "training" process. A good training program can ensure that an ANN performs multi-dimensional tasks like predicting outputs, object classification, approximating functions, recognizing a pattern from a wide range of data input, or completing a known pattern [17,18,19]. A simple multilayer ANN is shown in figure 2.4.

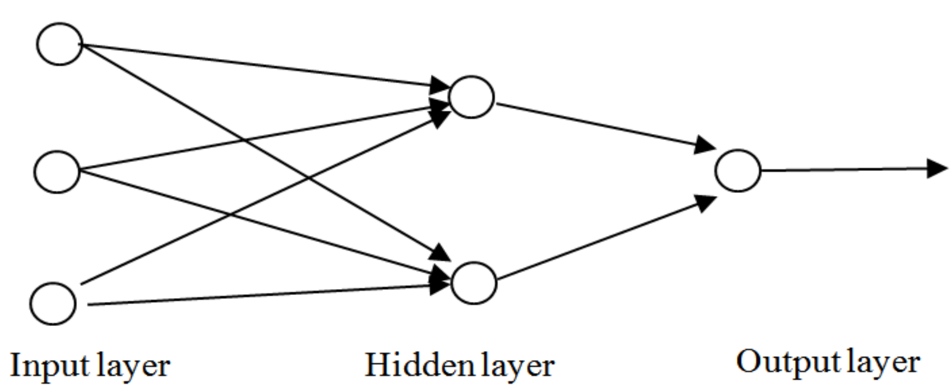

Figure 2.4 Simple multilayer ANN

One of the initial purposes of the development of ANN classifiers was to overcome the limitations of linear program; as the parallel nature of ANNs enables it to perform tasks that a linear program is unable to address. Secondly, ANNs learns continuously and do not require reprogramming. They are easy to implement.

Bakpo, F.S and Kabari, L. G [20] used feed-forward neural network for diagnosing Skin diseases. This means that the artificial neurons are arranged in layers and are capable of forwarding the signal, (i.e., from input to output) whereas the errors are propagated back. From input layer neurons network receives input symptoms, and output is given by neuron of output layer. A network may contain multiple intermediate hidden layers. Back propagation is used as learning or training algorithm. It uses supervised learning technique, in which network computes its weight with the help of provided input and output examples, and then calculates the error. This error is the difference between actual and expected results. A success rate of 90% is achieved.

Reddy N. P. [21] used artificial neural networks for diagnosis of human cardiovascular system. Neural network was trained over the normal and affected patient's data. It was then tested and validated using the testing data. These networks can be used to classify the real time physiological measurements taken from the patient. Accuracy of the network lies between 84% to 87% in most of the cases.

Wu et al. [22] used a multi-layer perceptron neural network to find out damage of a three-story building. Damage was assessed by reducing structural stiffness from 75% to 50%. The input to the neural network was the Fourier transform of the columns on particular floors, while the output was the level of damage in each associated structure. They achieved only 25% accuracy for damage diagnosis.

However, the development and use of an ANN does contain a significant disadvantage [23]. The main concern is that the ANN is design specific because the network architecture is determined mostly through trial and error. Each application is case-specific, and thus the literature does not provide any guideline on various aspects. For example, how many hidden layers and, how many neurons in each layer are required, which neuron connections are necessary or otherwise, what is the minimum return threshold, how many training patterns should be used, and what should be the weights of neuron connections [24]. Skidmore in [23] demonstrated that these factors can significantly affect the classification result, leading him to conclude that while the ANN performs as expected, "...the adjustments and fine tuning required for the input parameters would deter many users."

2.2.2 Decision tree

A decision tree is tree like structure, where each node other than leaf nodes represents the predicting attributes of the data set and the branches coming out of those nodes represents the values of the attributes in that node. Finally the leave nodes of tree represent the class [25] [26]. For increasing human readability we can represent the learned or trained tree as a set of if-then rules. Decision tree used top-down approach for classifying new unknown-class. A basic decision tree structure is shown in figure 2.5.

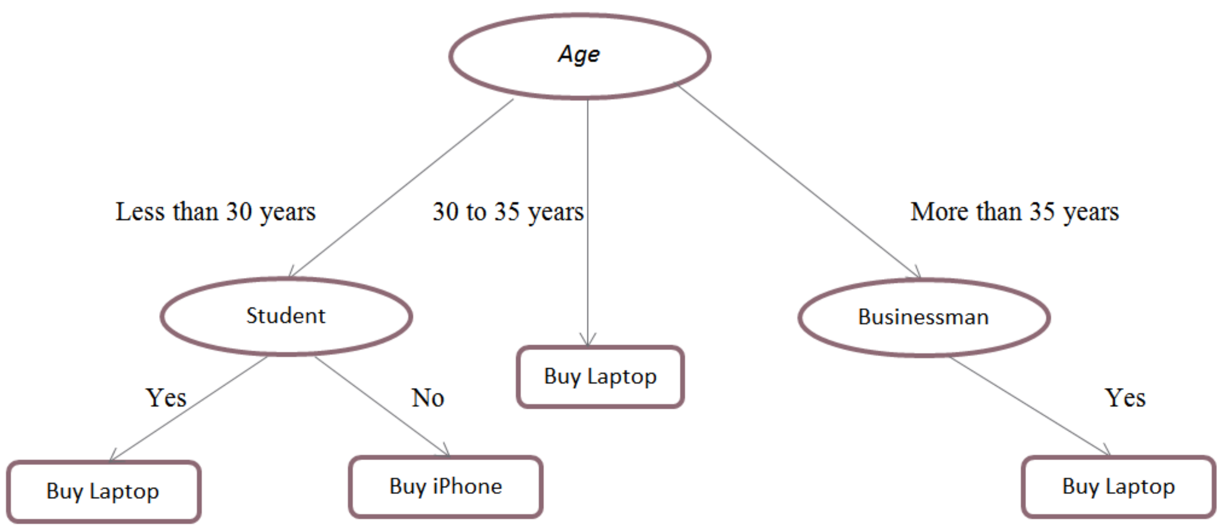

Figure 2.5 A decision tree

"Divide and conquer" approach is used for the construction of decision trees. Gradually smaller and smaller sub-set of training data is created, so that the subset may contain the maximum number of possible elements of same class. In each phase of the algorithm for a tree node an attribute is selected. It is intended to select the attribute subsets that will best highlight the underlay classified class.

The most common used learning algorithms for decision tree includes the ID3 [26] [27]and its successor C4.5 [28]. Decision tree can also be used for descriptive mining as it is very easy to generate a set of rules from a decision tree.

2.2.2.1 Tree pruning

After building the tree it is made pruned to check the over fitting and noise. In most of the cases, we prune decision tree until all the leaves contain data of a single class. Most commonly there are two types of pruning methods:

- Stopping or Pre-pruning
- Post pruning

Pre-pruning tries to look the best way of splitting the subset and assess the spited subset in terms of information gain or gain ratio. If this assessment falls below some threshold, the division is rejected. In this way the tree building and pruning process works simultaneously at each node of the tree.

But such stopping rules based on threshold would not give a best-pruned tree, because finding appropriate threshold is itself a difficult job [29]. Too high threshold can terminate division before the benefits of subsequent splits become evident, while too low results in little simplification.

On other hand, post pruning first grows the over fitted tree and then prune it. Though growing and then pruning is a time consuming process, it gives more reliable results. Post pruning calculates error at each node and then discards sub tree, which gives maximum error. This is also known as error-based pruning.

One of the early works using decision tree algorithms was used in diagnosis of lymphatic cancers [30]. Its success gave rise to the development of Assistant system [31] [32]. The

Assistant system was used in various fields including oncology, urology and prognosis of survival in hepatitis [31].

Many versions and extensions of the decision tree algorithms have been developed and are in use. One of such work has been done for ECG diagnosis [33]. Another method was used in the diagnosis of stroke.

In one of the studies, a variety of machine learning algorithms including C4-5 method, an extension of decision tree algorithm were applied to a database of Alzheimer's patient and controls. The attributes for construction of the decision tree were responded to Bless Orientation Memory and Concentration Exam (BOMC) and the Functional Activities Questionnaire (FAQ) in addition to demographic data. BOMC and FAQ are recommended tests by Agency for Health Care Policy Research (AHCPR), which assess the functional cognitive ability. Use of post pruning experiments have shown that dementia could be detected by an increase of 15-20% accuracy compared to just using BOMC, FAQ, or both tests [34].

However, decision tree still has some significant drawbacks. It can be unstable after the minor variation in the data; that results in the need for building a completely new tree [34]. Secondly, in case of some dominating class in the data set, we have a biased tree in result [26]. Finally, it is not very easy to implement the XOR and multiplexer problems with decision trees because of its over-complexity. Output attribute must be categorical; decision trees extracted from numerical data can be complex [30].

2.2.3 Support vector machine

Classifying using Support Vector Machine (SVM) is a supervised learning technique that uses a labeled data set for training and tries to find a decision function that classifies the training data. More often this technique is used for two class classifying [35]. It is based on statistical learning theory [36] [37].

SVM algorithm tries to find a hyper plane that splits training set optimally in pre-defined classes (a practical example can be found in [38]). Actually the algorithm has the ability to determine the parameters which in return determine the general equation of the plane. Let us consider a two dimensional problem, the solution of this problem would require to find out a line that "best" separates points in the positive class (points that are in the class A) from points in the negative class (points that are in the class B). The main objective is producing a classifier that works well on unseen data and has good generalization capabilities.

In Figure 2.6 we can have many of the possible linear classifiers that separate the data points in two distinct classes. But we have only single optimal classifier that separates the data points and helps to maximize the margins or the distance between the nearest data points of each of the classes. This optimal linear classifier is called the optimal separating hyper plane.

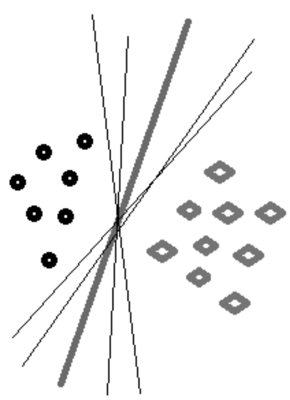

Figure 2.6 Optimal separating hyper plane

Let x is a vector and xi are the components of vector x. xi represents the ith component of the vector in a data set (xi; yi), whereas yi are the labels linked to xi. Usually the liner classifier is defined as dot product or the inner product or some time may refers as scalar product between two of the vectors, defined as wx + b = $\sum$i wixi $+ b$. A linear classifier is based on a linear function of the form:

f(x) = wx + b ………………………………………… (1)

Where vector w represents the weight vector and b is a constant value of bias. The hyper plane separates the space into two halves: function f(x) returns either positive (+) or negative (-) value. The sign of the returning value determines that the data point belongs to either one of the class. Decision boundary of the classifier is the boundary that separates positives data points from negative ones. The decision boundary defined by a hyper plane is said to be linear because of the equation (1). A linear classifier is the one in which decision boundary depends on the liner equation. Conversely, when the decision boundary of a classifier depends on the non-linear equation then the classier is said to be non-linear.

Bernhard used SVM [39], for identification of the patients with breast cancer, whom chemotherapy helps to increase survival time. SVM was used to classify patients in two classes; one with the initial stage of cancer and the second one beyond the initial stage. 82.7% accuracy was achieved after testing the data of 253 patients.

SVM used by Vojislav [40] in a system used for cancer diagnosis. DNA micro-array data is used as a classification training data set in SVM. Diagnosis error obtained by this system was lesser than the other known techniques used for cancer diagnosis using the same training data set. 36% of error reduction was achieved by using SVM.

S. Mukherjee et al in [41] performed classification on leukemia cancer data [42] using SVMs. The study analyzed classification ability of SVM on high dimensional microarray data. They used the feature selection method proposed by Golub et al in [43]. They ranked the features and picked top 49, 99 and 999 genes for classification. Classification using all the 7129 genes in data set was also performed. The study proposed two methods;

- SVM classification without rejections.
- SVM classification with rejections.

The former method was classifying the data set using linear SVM classifier with top 49, 99 and 999 genes and also using the complete set of 7129 genes. The SVM classifier achieved better accuracy compared to the method proposed by Golub et al [43]. The non-linear polynomial SVM classifier did not improve accuracy for the data set. The second method used a threshold value to reject test samples if they lie closer to the boundary plane. The overall accuracy obtained was 100% with a few samples rejected in each category of the filtered genes. Of the top 49 genes, 4 were rejected. Likewise, 2 genes were rejected from the top 99 genes, none rejected from the top 999 genes and 3 genes were rejected from the total 7129 genes. They concluded that linear SVM classifier with rejections based on confidence values performed well on the leukemia cancer data set [41].

However SVM's also has some drawbacks. Perhaps the most serious problem with SVM is the requirement of extensive memory and high algorithmic complexity for a large scale quadratic programming tasks [44]. Another challenge for SVM's is the choice of the hyper parameter and the choice of kernel functions [45]. Finally, the SVM classifier is fundamentally a two-class binary classifier which limits its applicability. Modifications are needed for addressing multi-class classification problems.

2.2.4 K-nearest neighbour

K-nearest neighbour (KNN) is different from other common classification techniques. It does not construct the classifier in advance. KNN compares each new data sample with old training data and tries to find out k nearest neighbours to the new sample on the basis of some suitable similarity or distance [46]. KNN classification technique is a simple classification method. The main idea involved in KNN classification is that similar observations belong to similar classes.

The algorithm of KNN is as follows:

- For each data point of the testing data set, the k nearest neighbours is located.
- Euclidean Distance formula is used to measure the distances between the testing data set points and training data set points.
- Now k nearest neighbours is examined to determine that most of the data points belong to which classification (category). This category is then assigned to the data point being examined.
- Above steps are repeated for rest of the data points in the testing data set.

Advantages of KNN:

- Easy to implement.
- Only single parameter required i-e number of K nearest neighbours.
- Does not require the prior knowledge about the distribution of the training data [46].

Sebastiano and Roli in [47] marked out the covered land patches using multi sensor images. KNN along with other classifiers is used to classify covered land. They achieved different efficiencies, with different number of neighbours (K = 3 - 50). An overall accuracy of 74% (k=15) was achieved with SAR (Synthetic Aperture Radar) image, whereas 80.5% (k=3) with ATM (Airborne Thematic Mapper) scanner. They achieved 89.8% (k=25) when both types of images are used together.

However, it still has some drawbacks. One of the major drawbacks of KNN classifiers is that the classifier needs all available data. This leads to significant overhead, if the size of training data set is big. Some of the other drawbacks are as follows;

- The performance of KNN depends highly on the choice of K.
- The computation complexity and memory limitation [48].
- It uses all the attributes of the data set regardless of its importance [48].
- The equal treatment of the K neighbours is conceptually unreasonable. Since all nearest neighbours are given equal voting weights to determine the belongingness of the input pattern. The information provided by distances is not utilized fully in the classification process. This can reduce the classification accuracy [49].

CHAPTER 3

PARTICLE SWARM OPTIMIZATION

A particle swarm optimization (PSO) is population based optimization technique proposed by Kennedy and Eberhart in 1995 [50]. The algorithm mimics the behavior of insect swarming or bird flock flying together in the search of optimal space in multi-dimensional environment. And try to adjust their distance and velocity for the sake of better optima. PSO is very similar to evolutionary algorithms (EAs) in a way that both algorithms are population based and additionally have fitness function. The population is randomly initialized and then search is made for optima.

Kennedy and Eberhart claims [50] that PSO is very easy to implement and has ability to solve wide variety of optimization problems. For example continuous non-linear and discrete optimization problems. PSO is based upon two models.

- Cognitive model
- Social model

In cognitive model, each particle is influenced by itself and updates its position or velocities on particle's personal history. In contrast the particles in social model update themselves by keeping in view the global or neighborhood history.

In PSO system a particle is treated as an individual with respect to population of particles; sometimes population is also refers as swarm. Each particle is a candidate solution of the optimization problem. It flies through the solution space for search of global optimum or the optimal solution. During its flight it continuously changes its position and velocity as well as keeps track of its previous position and velocities. It also keeps track of local best (individual) and global best (best of best in the entire population) positions and velocities. The new position of the particle can be influenced by its best position ever (i-e. its personal experience throughout the flight) as well as by its neighborhood particles (i-e. the experience of neighborhood particles throughout the flight). A local best position of the particle is the position, in which it so far achieves maximum fitness (best solution). In PSO literature and algorithm it refers as *pbest* or *lbest*. A global best value is the best value obtained so far by any particle in the neighborhood of that particle. This value is called *gbest*. Fitness function is used to calculate the performance of each particle in the population that how much it is close to global optimum. Fitness function can be different for problem to problem.

A particle in the population has following attributes:

- *Xi*: It's current position.
- *Vi*: It's current velocity.
- *Pbest*: Personal best position.
- *Gbest*: Global best position.

Velocity and position updating equation for the particles are as follow:

$$V_i\ (t + 1) = V_i\, t + c1 * r1\ (pbest_i - X_i\, t) + c2\ * r2\ (gbest_i - X_i\, t) \quad (2.1)$$

$$X_i\ (t + 1) = X_i\, t + V_i\ (t + 1) \quad (2.2)$$

Where *r1* and *r2* are two random numbers chosen between [0 , 1]. *c1* and *c2* are two learning constants.

3.1 The Algorithm

The implementation steps of original PSO process are as follows:

1. Initialize the n dimensional population with random velocities and position in the solution space.
2. Calculate the fitness of each particle in the solution space.
3. Compare the fitness of the particle with its pbest. If it is better than pbest, update the pbest velocity with current velocity and pbest position with current position.

4. Now compare the whole population's fitness with the gbest; if find any better than gbest update gbest velocity and position with current particle's velocity and position respectively.
5. Update the velocity and position of all the particles using equations 2.1 and 2.2, respectively.
6. Now start looping until the exit criteria meet. (More often we have max number of iterations or optimal fitness value as exit criteria)

3.2 PSO Flowchart

Figure 3.1 shows the detailed flowchart of PSO.

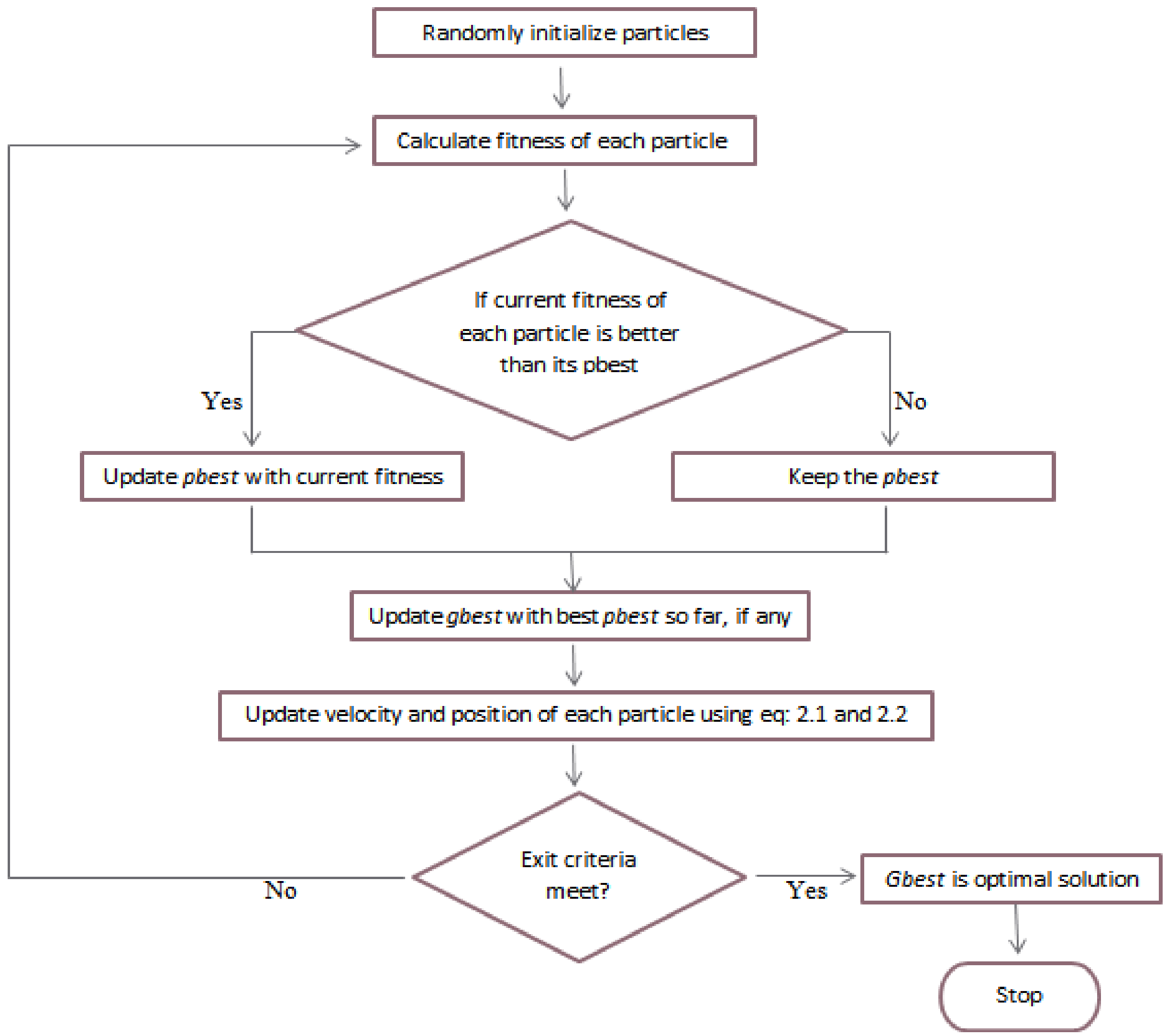

Figure3.1 PSO Flowchart

3.3 Modified PSO

Shi and Eberhart [51] introduced a new parameter to PSO named as *inertia weight w*. Inertia weight multiplies with the previous velocity of the particle and controls its impact. It introduces the ability of global and local search abilities in swarm. A large inertia weight can lead to the global search (exploration); whereas a small inertia weight helps to increase local search abilities (exploitation).

Updated version of equation 2.1 after introducing inertia weight is:

$V_i\ (t + 1) = wV_i\, t + c1 * r1\ (pbest_i - X_i\, t) + c2\ * r2\ (gbest_i - X_i\, t)$ . . . . (2.3)

Many researches used constant inertia weight to achieve optimal results [52,53,54]. On the other hand however some uses decreasing inertia weight [55,56]. A swarm with decreasing inertia weight has more capabilities of local search than global search [57].

3.4 Advantages of PSO

Following are the major advantages of PSO,

1. Easy to understand.
2. Its implementation is very easy and straight forward that is why it can be applied to any scientific research or engineering problem.
3. It has very few adjusting parameters as compare to other optimization techniques.
4. It has fast convergence rate than other optimization techniques.
5. It is derivative-free algorithm [58].
6. Calculations in the algorithm are very simple [58].
7. PSO is less sensitive to initial population initialization [58].

Owing to these advantages PSO is extremely popular in different fields like classification, pattern recognition, function optimization, feature selection, system design, disease diagnostics, multi-objective optimization, evolutionary computing and signal processing [59]. It has already been successfully applied on training of neural networks as well [60].

Like other stochastic search algorithms PSO also have two major drawbacks [61]. Premature convergence of the swarm is one of them. More often PSO prematurely converges when tries to optimize a multi-modal problem. The reason for the premature convergence of PSO is following. For the *gbest* point, PSO particles try to converge on the single point which lies in the area between the *pbest* (personal best) and *gbest* (global best). Even this point is not guaranteed as local optimum. Detailed study can be found in Van den Bergh [62] research thesis. Another cause for this premature convergence is the fast rate of information passing between the particles which may produce similar kind of particles and loses the diversity of swarm. This may lead to increase in the chances of being stuck within local optima [63]. Up till now different modifications are suggested by researchers to overcome this problem. But the most effective one is inertia weight [62], that has already been discussed, rest of the suggestions are not in the scope of our current discussion.

The other drawback of PSO is regarding its performance. PSO performs very efficient in some cases where as it performs worse in other search algorithm. Its performance is problem-dependent. The main cause of this problem-dependent nature of PSO lies in its different parameter settings. By optimizing parameters for the same problem, the performance of the PSO can be enhanced. In fact no standard parameter setting can be found in literature which works well in all proportions. This problem is very common in PSO where adjustment of its parameter have a huge effect on its performance [64]. For instance, if one increase the value of the inertia weight “*w*”, it will result in the increase in the speed of particles which causes more global search also known as exploration. It will also reduce the local search abilities known as exploitation. Similarly in case of reduction in the inertia weight “*w*” will decrease the speed of particles which causes less exploration and more exploitation. Therefore finding the optimal value for inertia weight “*w*” is a trivial job and varies from problem to problem. Hence the performance of the PSO is totally problem dependent. Self-adaptive PSO have the capability to overcome this problem dependent nature of PSO [65]. In this variant, PSO continuously updates its parameters from the feedback coming from the search process [64]. Self-adaptation has already been successfully applied by Claeric [55], Shui and Elberhart [66] and Heu and Elberhart [65]. Another way to overcome problem dependent nature of PSO is the use of hybridization technique. In Hybridization, one can combine different techniques and take the advantages of all the applied techniques [64]. Hybridization has already been successfully applied by Anegealine [67], Loverberg [64], Kricnk and Loverberg [68] and Et al veram-a-chaneani [69].

3.5 Classification Using PSO

Tewolde and Hanna used PSO [70] for classification of breast cancer. They use two techniques for classification of tumor i.e. single and multiple surfaces based data classification. In single surface based data classification, they use PSO to create a hyper plane that separates the two classes from each other with acceptable classification accuracy. The particles of the swarm encode the equation of hyper plane by assigning their coordinate values to the coefficients of equation for hyper plane. In multiple surfaces based data classification PSO creates pair of parallel hyper planes that separate two classes. Figure 3.2 can explain it better.

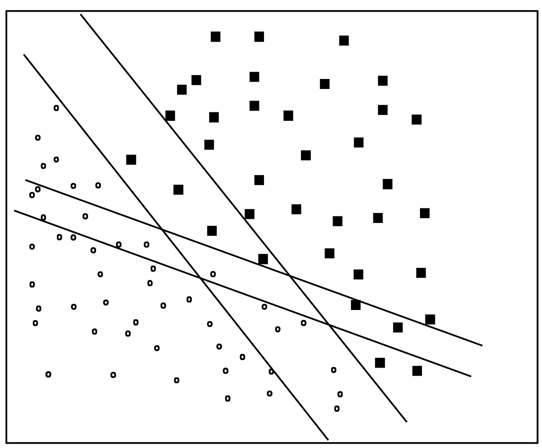

Figure3.2 PSO Hyper Plane

For generating these two parallel planes PSO particle encode four equations using their coordinate values. Author claimed that the results achieved using single surface based classification is 97.4% on breast cancer data set; whereas the accuracy increased to 100% when multiple surfaces based data classification is used.

Rajiv and Kannan used PSO [71] to construct the classification rules for breast cancer data set. They used three variants of PSO to generate if-then rules from the data. Their goal was to generate the rules which not only can classify the data correctly, but also comprehendible to user. The resulted rules are comprehensible and the overall efficiency is kept on increasing with the number of iterations. They concluded that by maximizing the number of iterations one can achieve better success rates. The drawback in their technique was that they could be able to produce good results only in the case of categorical data.

Wang and Zhang [72] evolved rules for classification using PSO. They proposed an approach in which particle's dimension are equal to the number of attributes contains by the data set. Fitness function used in this technique is:

Fitness = (N / N + FP) * (TP / TP + FN) * (TN / TN + FP);

Where N = The number of records in the data set.

Accuracy achieved is about 73.59%.

Andre and Taylor used PSO [73] for rule extraction of rules. They used Michigan approach and the corresponding algorithm is MOPSO (Multi-objective Particle Swarm Optimization). In Michigan approach each particle represents the single rule. MOPSO algorithm runs over the discrete and continuous data and generates the corresponding rules from the data. The maximum efficiency achieved was 95.18% when MOPSO is used on discrete data.

Liu Qin Shi [74] also worked on rule based classification with Particle Swarm Optimization. They introduce a new approach to evolve if-then rules from PSO. They divided single particle into three parts, first parts indicates the existence of attribute, second part have information about the operator and the third part represents the value of the attribute. They used following fitness function:

Fitness = w1 * (Accuracy * Coverage) + w2 * Succinctness

where, Accuracy = TP / (TP + FP),

Coverage = TP / (TP + FN) and

Succinctness = 1 – (countAnt – 1) / AttributeCount

Where countAnt is number of conditions in the rule, and AttributeCount is number of attributes present in the data set.
The best accuracy they achieved on categorical and continuous data is 95.48% and 65.59% respectively.

Wei Chang [75], used improved simplified swarm optimization SSO to create rule-based classifier for thyroid gland classification. He used SSO because of its simplicity, efficiency and flexibility [76]. The major drawback in using SSO is that, it is only valid for discrete data [75]. Like other techniques (ACO, GA, PSO) SSO also operates in similar way i.e. population initializes randomly with in the solution space and then it search for the optimal solution till the maximum number of generations are met. Each particle acts like a rule which comprises of two dimensional array with first element of the array is lower bound and the second element is the upper bound of the attribute. Number of rows in the matrix is decided by the number of the attributes of the data set. Thus the size of matrix is double the size of the total number of attributes in the data set. Classification accuracy is the fitness of the rule. The maximum efficiency they achieved is 95.3% using SSO after taking average of 1-fold, 2-fold and 3-fold cross validation.

Sousa, Silva and Neves used PSO [77] to discover rules for data classification. They used CPSO (Constricted Particle Swarm Optimization) to evolve rules from data set. Breast cancer diagnosis and animal classification data sets were used; both of them are standard benchmark data set. So they were able to compare the results with other standard techniques.

Saleem and Xin [78] extract SQL queries based rules from data set using evolutionary algorithm. The goal was to generate SQL query from genotype which runs over the database and extract the desired data. To represent a genotype as SQL query they need to divide gene into three parts, first part represents Attribute name or attribute, second represents the logical operator and the third one describes the value of the attribute. They use the following fitness function:
Fitness = 100 - FP - (2 * FN)

The aim of fitness function is to give higher probabilities to the attributes with high fitness values. The rule generated from the genotype is like:

Select *
From [tablename]
Where (legs = 4) (predator = true) (feathers = false) (venomous = false)

This shows clearly that this approach only works on categorical data. Another drawback of this approach is the high memory demand used for storing attribute names.

The author claimed that overall accuracy achieved on categorical data was 100%, however on continuous data accuracy was only 16.9%.

All the approaches discussed above have good performance only over the categorical data. The main reason behind is, that every researcher used the equality for the attribute value. The equality can only work in the case of categorical data. For achieving good results for continuous data we should replace equality to some collection or range or series of values for particular attribute. Additionally the fitness function used by most of researchers is only by considering classification accuracy of the classifier, which is not an adequate measure to validate the performance of classifier. In next chapter we will discuss the proposed technique for discovering SQL based rules using PSO.

CHAPTER 4

THE PROPOSED TECHNIQUE

In this chapter we will discuss the proposed technique for discovering comprehensive rules using PSO for back pain diagnosis. The section 4.1 explains the methodology used for the diagnosis and classification. Section 4.2 presents the data set used for training and testing of our algorithm. In section 4.3, we will discuss the rule representation techniques used in the study. Section 4.4 and section 4.5 covers the initial initialization criteria. Section 4.6 explains the position and velocity update functions that are used. Section 4.7 explains the feature selection technique. Section 4.8 discusses the technique used to encode rule into SQL query and also fitness function used for calculating fitness of our rule.

4.1 Methodology

In this thesis, we have proposed a new variable encoding scheme which is different from several conventional encoding schemes that are discussed in the literature. We have used particle swarm optimization (PSO) a well know optimization algorithm for discovering comprehensive rules [79]. Comprehensive classification rules with the help of PSO for medical data set for the diagnosis of lower back pain are discovered. The rules discovered using our algorithm is transparent and easy to understand. These transparent rules remove a big hurdle in the adoption of such systems in medical domain. The conventional rule mining systems are too complex and difficult to understand by physician and make the system useless for them [5]. Back pain may have many causes, but most of the time impairment in the vertebral column is the main cause. Impairment in the vertebral column results in two major diseases i.e. Spinal Disk herniation and Spondylolisthesis. The proposed rule based classification system not only classifies the infected patients but also presents the rules, which are transparent and portable to any database management systems in the form of SQL queries. Our rule also helps in feature selection. Thus, the main attributes of our rule are transparent, portable and helps in feature selection.

4.1.1 Calculating dimension and ranges

Algorithm 4.1 states how to calculate dimension and range of the proposed technique:

```
• Begin
• Input database
• Calculate the number of attributes in database
• For (all the attributes of database - 1)
    o Count ++
    o Range [ai, bi] = min(attr[i]), max(attr[i])
• End
• Return number of dimensions = Count * 2
• Return ranges [a, b]
• End
```

Algorithm 4.1 ALGO I Calculating dimension and ranges

4.1.2 Initializing velocity and positions

Algorithm 4.2 states the velocity and position initialization of PSO particle in the proposed technique:

```
•  Begin
•  For (Number of Particles in swarm - 1)
    o Velocity[i] = Rand()
    o Position[i] = Rand(min(range[i]), max(range[i]))
•  End
•  Return Velocity and Position
•  End
```

Algorithm 4.2 ALGO II Initializing velocity and positions

4.1.3 Discovering rules from particle's data

Algorithm 4.3 states how to discover rules from particle's data in the proposed technique:

```
• Begin
• For (Number of Particles in swarm - 1)
    o Rule = SELECT * FROM [table] WHERE
    o For (Number of attributes of particle - 1)
          ▪ IF (Position[a] <= Position[b])
            • Rule  += Attr[i]  €  [Position[a] to Position[b]]

          ▪ End
• End
• Return Rules
• End
```

Algorithm 4.3 ALGO III Discovering rules from particle's data

4.2 Diagnosis Data

The data set which is used in this study is downloaded from a well know repository i.e. UCI Machine Learning Repository. The data set belongs to medical domain and having real valued attributes. It contains the data of 310 orthopedic patients. Out of these 310 patients 100 patients belong to normal class and rest of them (210) to abnormal/infected class. The data set is divided into two sets. First set is training set; it is used to train the algorithm for the production of a comprehensive rule. The second set i.e. test set is used for the testing purposes. We used 10 fold cross validation technique to assess our results. This means that each time there are 279 samples in the training set and rest of 31 samples are used test set.

The data set contains the values of six biomechanical features of orthopedic patient's vertebral column which helps us to classify them in one of the classes. These six attributes defines the orientation and the shape of pelvis and lumbar spine in the vertebral column. Their names are pelvic incidence, pelvic tilt, lumbar lordosis angle, sacral slope, pelvic radius and grade of spondylolisthesis.

4.2.1 Attributes ranges

In the table below the min/max ranges of the attributes in the data set along with the variable names used in the algorithm are presented.

Vertebral Column Data Set	
Names	**Possible Range**
Pelvic Incidence	26 – 130
Pelvic Tilt	-7 – 50
Lumbar Lordosis Angle	14 – 126
Sacral Slope	13 – 122
Pelvic Radius	70 – 164
Grade Of Spondylolisthesis	-12 – 419

Table 4.1 Data set attributes ranges

4.3 Rule Representation

In the proposed approach, the rules are encoded using two different approaches; which are AND operator and the OR operator.

AND operator returns true if and only if both of the conditions on which the operator is applied returns true. On the other hand OR operator returns true, if at least one of the conditions on which the operator applied returns true. It will only be false if both of the conditions on which the operator applied returns false.

Using AND operator we encoded a rule as below:

'if (attribute 1 € [value 'a' to value 'b']) AND (attribute 2 € [value 'c' to value 'd']) AND (attribute 3 € [value 'e' to value 'f']) AND (attribute 4 € [value 'g' to value 'h']). . . .' Then we have class = 'yes'

Thus to satisfy the above rule the values for all attribute must be within the designated ranges otherwise it will not chose for nominated class. This rule will have good outcome in a scenario in which we have data set with dependent attributes. But for a data set in which the values for the attributes are independent to each other may not yield good results.

Using OR operator we encoded a rule as below:

'if (attribute 1 € [value 'a' to value 'b']) OR (attribute 2 € [value 'c' to value 'd']) OR (attribute 3 € [value 'e' to value 'f']) OR (attribute 4 € [value 'g' to value 'h']). . . .' Then we have class = 'yes'

For satisfying the above rule if the value of any selected attribute lies within the designated ranges it will be chosen for nominated class. This rule will have good outcome in the both scenarios; either a data set with dependent attribute values or a data set with independent attribute values.

This rule encoding is more flexible then the several conventional rule encoding schemes, which have already been discussed in literature. The flexibility of this rule encoding scheme lies in the consideration of a range of values for a single attribute. Moreover this scheme has the flexibility to the extent that is can even use single value for the attribute, if needed.

4.4 Particle Initialization

All the particles are initialized randomly with random velocities and positions. But the data set which is under consideration have numerical values for the entire set of attributes. Therefore, while initializing random positions we need to restrict its range from minimum range to maximum range for the particular attribute. We must need to assure that the random positions of all the particles are within the designated lower and upper bound range of data set attributes.

4.5 Particle Dimension

The number of dimensions of each particle is twice the number of attributes present in the data, excluding the class attribute. For accommodating the upper and lower bound of the attributes range we need twice the number of dimension than the number of attributes. First value represents the lower bound and seconds the upper bound of an attribute. Let us suppose one have *i* number of attributes in the data set, then the number of dimensions of each particle are *2i*. The lower bound and upper bound for the particle are as follows:

Min (a_i) = min (value of attribute i) . . . . (4.1)

Max (a_i) = max (value of attribute i) . . . . (4.2)

Where, *Min (a_i)* is the lower bound range and *Max (a_i)* is the upper bound range of the particle a_i.

4.6 Position and Velocity Update

In PSO, each particle possesses its own position and velocity. In the search of global optima both position and velocity of a particle are updated in every iteration. Every particle moves in the direction that is influenced by global best particle, by its own previous experience and by some random factor.

The velocity of each particle is updated according to the following equation:

$$V_{i+1} = \omega V_i + C_1 rand(0,1)(X_{pbest} - X_i) + C_2 rand(0,1)(X_{gbest} - X_i) \qquad (4.3)$$

Where w is the inertia weight, C_1 and C_2 are the constriction coefficients, “*rand*” is any random value between 0 to 1, X_{pbest} and X_{gbest} are the personal and global bests of the particle. The constriction coefficients control the particle convergence and prevent explosion by eliminating the need for V_{max} parameter [80]. We have used the Clerc's constriction method, which recommends following values:

$$C_1=1.49618,\ C_2=1.49618 \text{ and } w=0.7298. \quad (4.4)$$

X_{pbest} is the best position achieved by particle up till now, whereas X_{gbest} is the best of best position achieved by any particle in the swarm or the best position of the whole swarm. After the velocity update, each particle updated its position using following equation:

$$X_{i+1} = X_i + V_{i+1} \quad (4.5)$$

In equations 4.3 and 4.5 *i* represents the iteration number. The position and velocity are updated for the each dimension of particle. While updating the position of particle make sure, that it must be in the designated range for that dimension.

The overall working of PSO algorithm is stated in the Algorithm 4.4:

4.7 Fitness

To evaluate the fitness of a particle, the rule is encoded into an SQL query. The SQL query runs over the database and returns result count. The main advantage of SQL query is easy to understand, enhanced readability for non-experts and ease of integration with any exiting database system, which increases the portability of system.

The genotype produced by particle's parameter is translated into phenotype i.e. SQL query, initialize the string with the value "SELECT Count (*) FROM <Table Name> WHERE", and then appending the values of particle parameters on the end of string. For example the genotype returns by particle parameters are:

[27, 97, -2, 10.9, 19.7, 112.7, 13, 114]

```
• Begin
• Initialize particles
• While termination criteria not met
    o Calculate fitness of particles
    o For each particle
        ▪ Update particle gbest
        ▪ Update particle velocity
        ▪ Update Particle Position
    o End particles
• End while
• Return best particle
• End
```

Algorithm 4.4 PSO algorithm

Will be translated into phenotype:

With AND operator:

SELECT *Count (*)*

FROM *<Table Name>*

WHERE *Pelvic Incidence between (27 and 97) AND Pelvic Tilt between (-2 and 10.9) AND Lumbar Lordosis Angle between (19.7 and 112.7) AND Sacral Slope between (13 and 114)*

With OR operator:

SELECT *Count (*)*

FROM *<Table Name>*

WHERE *Pelvic Incidence between (27 and 97) OR Pelvic Tilt between (-2 and 10.9) OR Lumbar Lordosis Angle between (19.7 and 112.7) OR Sacral Slope between (13 and 114)*

Now the above SQL query sent to database, and then returned results are analyzed.

4.7.1 Fitness function

For analyzing the results returned by SQL query one needs a fitness function. The fitness function used in our approach is Area under the Curve (AUC), AUC has been defined as a measure that correctly evaluates the discriminating power of a classifier. It acts as a better measure for classifier efficiency than the traditional classification accuracy, especially for imbalanced data-sets.

$$AUC = \frac{1}{2} * [(tp/(tp+fn))+(tn/(tn+fp))] \quad (4.5)$$

4.8 Feature Selection

Our proposed technique is also helpful in feature selection. Feature selection is a very common term used in data mining to describe the techniques available for reducing inputs to a manageable size for processing and analysis. Feature selection implies not only imposing an arbitrary or predefined cutoff on the number of attributes that can be considered when building a model, but also the choice of attributes, meaning that either the analyst or the modeling tool actively selects or discards attributes based on their usefulness for analysis.

As discussed earlier that we have consider lower and the upper bound ranges for each particle. During the rule evolution or updating particle, it may happen that the value of lower bound grows greater than the value of upper bound of any particular attribute. In such case, we simply neglect that attribute and it is not considered in the classification decision. This encoding enables an automatic feature selection mechanism embedded in classification process.

Figure 4.1 presents the flow chart of proposed technique.

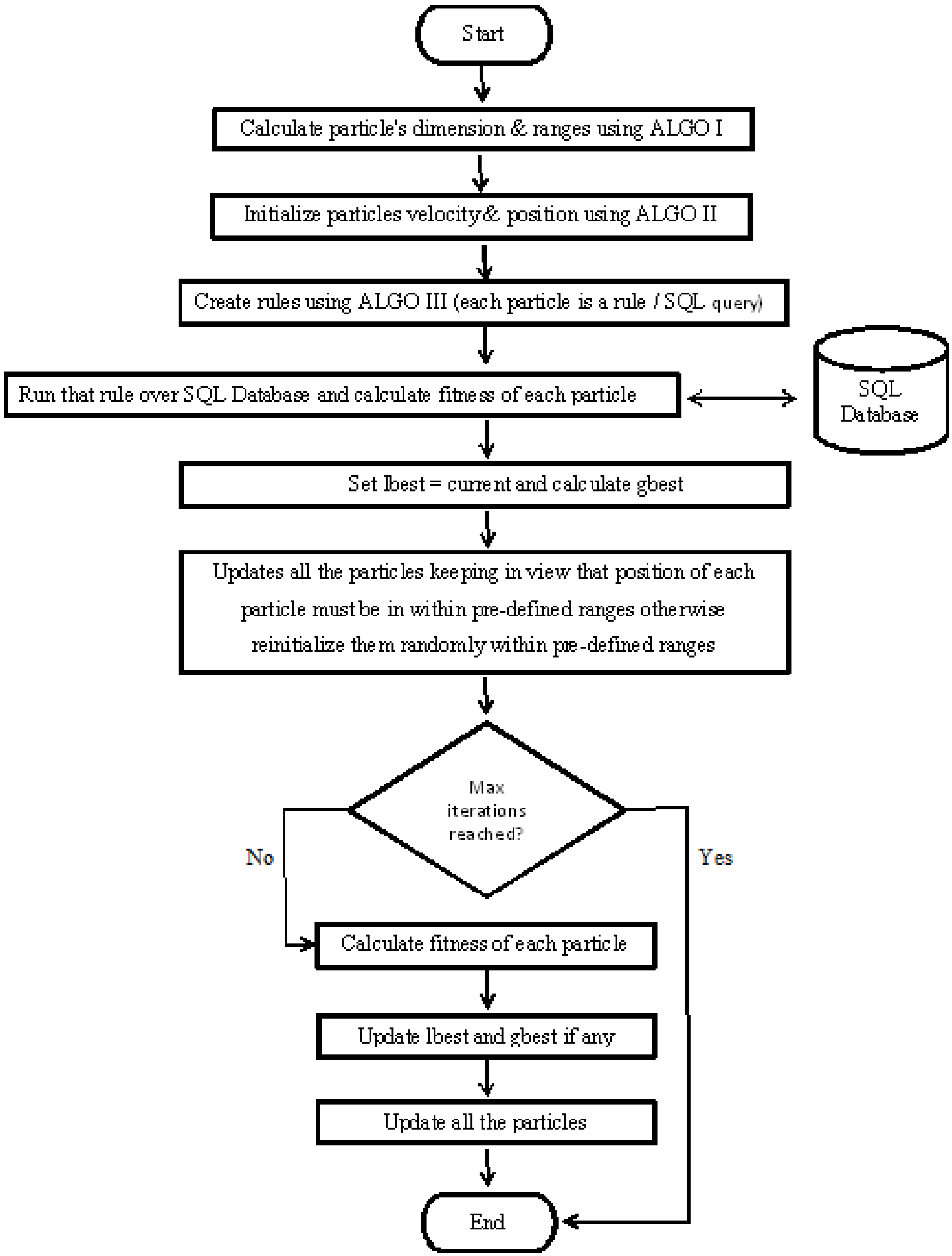

Figure 4.1: Flow Chart of Proposed Technique

CHAPTER 5

EXPERIMENTAL RESULTS

In this chapter we will discuss the results of the proposed technique that are presented in last chapter to evaluate its accuracy. It covers the outcome and the accuracy of diagnosing and classifying the normal and abnormal patients by proposed technique. This chapter also provides a comparison study between the proposed technique and three most popular classification techniques ANN, Decision Trees and Support Vector Machine on the same data set.

The section 5.1 presents the quick over view of data set being used. Section 5.2 explains the experimental setup. Two types of comparison are made. The section 5.3 exhibits the results and comparison of the two variants of the proposed technique. Section 5.4 presents the comparison between the proposed technique with some of well know classification techniques already existing in the literature and finally the section 5.5 presents the summary of the chapter.

5.1 The Data Set

The data set belongs to medical domain and downloaded from a publically free library i-e: UCI machine learning repository *http://www.ics.uci.edu/.* The main attributes of the data set are listed in Table 5.1.

No. of Attributes	6
No. of Instances	310
No. of Classes	2
Missing values	No
Data Type of Attributes	Decimal

Table 5.1 : Description of vertebral column data set

Table 5.2 gives a quick over view of the maximum/minimum ranges of each attribute present in the data set.

Attribute Names	Possible Maximum - Minimum Range
Pelvic Incidence	26 – 130
Pelvic Tilt	-7 – 50
Lumbar Lordosis Angle	14 – 126
Sacral Slope	13 – 122
Pelvic Radius	70 – 164
Grade Of Spondylolisthesis	-12 – 419

Table 5.2 Attribute max/min ranges

5.2 Experimental Setup

The proposed algorithm is written in C# using .Net framework version 4.0. C# has been chosen as development language because of its ease to integrate with any existing data base engine, specifically Microsoft SQL Server. Microsoft SQL Server 2008 is used as data base engine. The particulars of machine and software version used for conducting all the experiments are tabulate in Table 5.3.

Table 5.4 present the PSO parameters setting used in the proposed methodology. These parameters have been taken from [80] [81] after careful empirical analysis the problem.

All of the results have been conducted using 10 fold cross-validation technique. In 10 fold cross-validation, the data set is randomly divided into 10 equal parts. Nine parts of the data are used to train the classifier and the last 10th part is used as test set. This technique runs 10 times iteratively over the whole data set. This will provide a significantly measure of the algorithm's performance.

Processor	Intel(R) Core(TM) i5 CPU @ 2.50GHz
Installed Physical Memory (RAM)	8.00 GB
OS Name	Microsoft Windows 7 Professional
Development Language	C#
.Net Framework version	4.0
Development IDE	Microsoft Visual Studio 2010 Premium
Data base Engine	SQL Server 2008

Table 5.3 Hardware/software details

PSO PARAMETERS FOR EXPERIMENTATION	
Population Size	100
Particle Dimensions	12
No. of Iterations	100
Inertia weight	0.7
R1, R2	Randomly between 0.0 to 1.0
C1	1.49
C2	1.49

Table 5.4 PSO parameters

5.3 Outcome of the Proposed Technique

In the proposed technique, we have encoded our rules using two different approaches. First by using AND operator and in the second approach we used OR operator.

As mentioned earlier, all of the results are obtained using 10 fold cross-validation technique. The data of 310 instances are available, 10 fold cross-validation technique is used. Every time randomly we chose 279 samples in the training set and rest of 31 samples are selected for test set. We run algorithm 100 times for each fold and then chose one of them with best AUC.

5.3.1 Using AND operator

Table “5.5” and “5.6” present the detailed classification results using AND operator for the training and testing data set respectively using 10 fold cross validation. The columns true positive and true negative represent the result of true positive and true negative values. Similarly the column false negative and false positive represent the false positive and false negative values, where true/false positive and negative are defined as;

True positive = the result is positive and is classified as positive
True negative = the result is negative and correctly classified as negative
False positive = the result is negative and misclassified as positive
False negative = the result is positive and misclassified as negative
AUC column represents the result of area under the convex hull for one point value.

5.3.2 Using OR operator

Table “5.7” and “5.8” present the detailed classification results using OR operator for the training and testing data set respectively using 10 fold cross validation. The columns true positive and true negative represent the result of true positive and true negative values. Similarly the column false negative and false positive represent the false positive and false negative values, where true/false positive and negative are defined as;

True positive = the result is positive and is classified as positive
True negative = the result is negative and correctly classified as negative
False positive = the result is negative and misclassified as positive
False negative = the result is positive and misclassified as negative. AUC column represents the result of area under the convex hull for one point value.

TRAINING RESULTS						
Fold #	**True**	**False**	**True**	**False**	**AUC**	**Accuracy**
	Positive		**Negative**			
1	155	2	45	77	0.87	71.68%
2	148	0	77	54	0.86	80.65%
3	129	2	98	50	0.85	81.36%
4	127	2	98	52	0.84	80.65%
5	126	2	98	53	0.84	80.29%
6	128	2	98	51	0.85	81%
7	69	0	100	110	0.85	60.57%
8	139	1	87	52	0.88	81%
9	150	1	81	47	0.87	82.8%
10	108	0	81	90	0.87	67.74%
Average	127.9	1.2	86.3	63.6	**0.86**	**76.77**%
Min	69	0	45	47	0.84	60.57%
Max	155	2	100	110	0.88	82.8%
Stdev	23.71	0.87	20.30	16.08	0.01	7.11%

Table 5.5 AND training results

TESTING RESULTS						
Fold #	**True**	**False**	**True**	**False**	**AUC**	**Accuracy**
	Positive		**Negative**			
1	2	0	21	7	0.61	76.67%
2	1	0	23	6	0.57	80%
3	28	0	0	2	0.93	93.33%
4	24	0	0	6	0.8	80%
5	25	0	0	5	0.83	83.33%
6	21	0	0	9	0.7	70%
7	26	0	0	4	0.87	86.67%
8	2	0	12	16	0.56	46.67%
9	7	0	18	5	0.79	83.33%
10	0	0	19	11	0.5	63.33%
Average	13.6	0	9.3	7.1	**0.72**	**76.33**%
Min	0	0	0	2	0.5	46.67%
Max	28	0	23	16	0.93	93.33%
Stdev	11.45	0	9.66	3.80	0.14	12.69%

Table 5.6 AND testing results

TRAINING RESULTS						
Fold #	**True**	**False**	**True**	**False**	**AUC**	**Accuracy**
	Positive		**Negative**			
1	171	0	79	29	0.93	89.61%
2	169	0	77	33	0.92	88.17%
3	157	7	93	22	0.9	89.61%
4	151	3	97	28	0.9	88.89%
5	149	2	98	30	0.9	88.53%
6	150	4	96	59	0.9	79.61%
7	122	0	100	57	0.92	79.57%
8	160	1	87	31	0.91	88.53%
9	172	1	81	25	0.93	90.68%
10	169	5	76	29	0.89	87.81%
Average	157	2.3	88.4	34.3	**0.91**	**87.1**%
Min	122	0	76	22	0.89	79.57%
Max	172	7	100	59	0.93	90.68%
Stdev	14.46	2.28	8.99	12.21	0.01	3.84%

Table 5.7 OR training results

TESTING RESULTS						
Fold #	**True**	**False**	**True**	**False**	**AUC**	**Accuracy**
	Positive		**Negative**			
1	8	6	15	1	0.8	76.67%
2	5	0	23	2	0.86	93.33%
3	28	0	0	2	0.93	93.33%
4	27	0	0	3	0.9	90%
5	30	0	0	0	1	100%
6	30	0	0	0	1	100%
7	26	0	0	4	0.87	86.67%
8	17	0	12	1	0.97	96.67%
9	10	2	16	2	0.86	86.67%
10	7	0	19	4	0.82	86.67%
Average	18.8	0.8	8.5	1.9	**0.9**	**91**%
Min	5	0	0	0	0.8	76.67%
Max	30	6	23	4	1	100%
Stdev	9.91	1.83	8.90	1.37	0.07	6.84%

Table 5.8 OR testing Results

5.3.3 Comparison between AND operator and OR operator outcomes

Figure 5.1 and Table 5.9 show the details of comparison between the classification accuracy of both variants used in the thesis. The OR variant shows better performance over AND variant on both training and testing data sets. Both of variants were run with identical variables in terms of algorithm parameters and data set categorization. The reason behind better performance of OR than AND is its non-linear nature of grouping things together. Thus if a data set is targeted with non-linear classifier then OR variant do have better performance in terms of classification accuracy than AND variant. However if the data set can linearly be classifiable than AND should give us acceptable accuracy while comparing with OR. Therefore the classification accuracy of our algorithm is dependent upon the nature of data set.

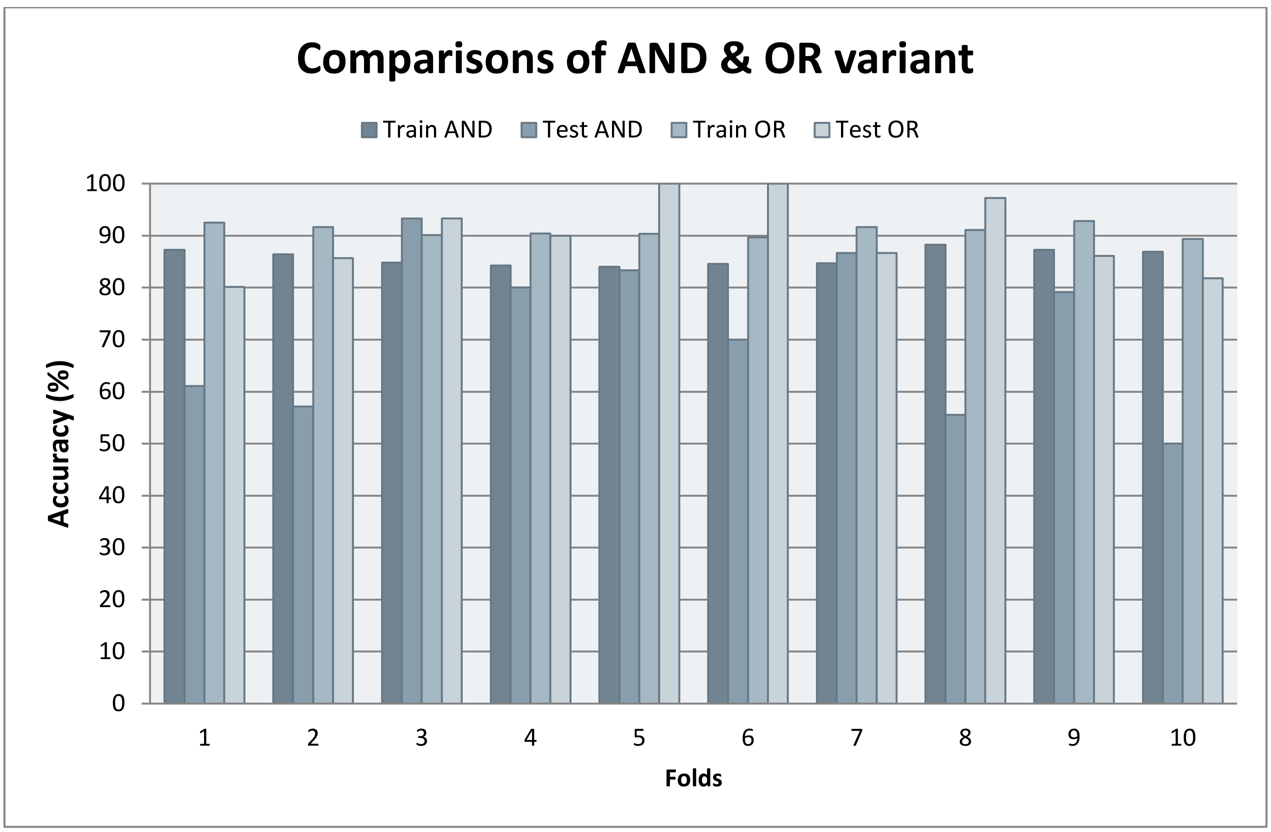

Figure 5.1 Comparisons of AND & OR variant

Fold #	AND		OR	
	Train	**Test**	**Train**	**Test**
1	71.68%	76.67%	89.61%	76.67%
2	80.65%	80%	88.17%	93.33%
3	81.36%	93.33%	89.61%	93.33%
4	80.65%	80%	88.89%	90%
5	80.29%	83.33%	88.53%	100%
6	81%	70%	79.61%	100%
7	60.57%	86.67%	79.57%	86.67%
8	81%	46.67%	88.53%	96.67%
9	82.8%	83.33%	90.68%	86.67%
10	67.74%	63.33%	87.81%	86.67%
Average	**76.77**%	**76.33**%	**87.1**%	**91**%
Min	60.57%	46.67%	79.57%	76.67%
Max	82.8%	93.33%	90.68%	100%
Stdev	7.11%	12.69%	3.84%	6.84%

Table 5.9 Comparison between AND operator and OR operator accuracies

Table “5.10” and “5.11” present ten best rules generated from proposed technique using variant AND and OR respectively. One can come to the conclusion; if carefully observe the rules. Most of the rules contain Grade of Spondylolisthesis ranges. So the attribute which has more weightage for classification than others attribute is Grade of Spondylolisthesis. Grade of Spondylolisthesis provide physicians with a standardized, medically accepted scale to measure the severity of a vertebral slip [82].

10 BEST RULES (AND Variant)
Select count(*) from Table Where (Pelvic Tilt between 1 and 44) AND (Lumbar Lordosis Angle between 51 and 80)
Select count(*) from Table Where (Pelvic Incidence between 37 and 76) AND (Grade Of Spondylolisthesis between 52 and 111)
Select count(*) from Table Where (Grade Of Spondylolisthesis between 15 and 303)
Select count(*) from Table Where (Grade Of Spondylolisthesis between 50 and 107)
Select count(*) from Table Where (Grade Of Spondylolisthesis between 50 and 122)
Select count(*) from Table Where (Lumbar Lordosis Angle between 50 and 124) AND (Sacral Slope between 28 and 116)
Select count(*) from Table Where (Grade Of Spondylolisthesis between 50 and 119)
Select count(*) from Table Where (Lumbar Lordosis Angle between 43 and 117) AND (Grade Of Spondylolisthesis between 10 and 202)
Select count(*) from Table Where (Lumbar Lordosis Angle between 51 and 103)
Select count(*) from Table Where (Pelvic Tilt between 7 and 43) AND (Sacral Slope between 41 and 102) AND (Grade Of Spondylolisthesis between 14 and 233)

Table 5. 10 best rules (AND variant)

10 BEST RULES (OR Variant)
Select count(*) from Table Where (Pelvic Incidence between 27 and 40) OR (Lumbar Lordosis Angle between 47 and 76) OR (Sacral Slope between 66 and 66) OR (Pelvic Radius between 95 and 103) OR (Grade Of Spondylolisthesis between 20 and 310)
Select count(*) from Table Where (Pelvic Incidence between 28 and 38) OR (Pelvic Radius between 133 and 154) OR (Grade Of Spondylolisthesis between 18 and 283)
Select count(*) from Table Where (Lumbar Lordosis Angle between 20 and 21) OR (Pelvic Radius between 102 and 107) OR (Grade Of Spondylolisthesis between 19 and 389)
Select count(*) from Table Where (Pelvic Tilt between -7 and 4) OR (Lumbar Lordosis Angle between 52 and 84) OR (Sacral Slope between 80 and 83) OR (Grade Of Spondylolisthesis between 18 and 211)
Select count(*) from Table Where (Pelvic Incidence between 84 and 117) OR (Lumbar Lordosis Angle between 51 and 104) OR (Grade Of Spondylolisthesis between 12 and 286)
Select count(*) from Table Where (Lumbar Lordosis Angle between 51 and 98) OR (Grade Of Spondylolisthesis between 10 and 152)
Select count(*) from Table Where (Lumbar Lordosis Angle between 52 and 89) OR (Sacral Slope between 54 and 82) OR (Grade Of Spondylolisthesis between 85 and 185)
Select count(*) from Table Where (Pelvic Incidence between 27 and 40) OR (Pelvic Tilt between 28 and 30) OR (Lumbar Lordosis Angle between 49 and 85)
Select count(*) from Table Where (Pelvic Incidence between 31 and 37) OR (Lumbar Lordosis Angle between 50 and 82) OR (Sacral Slope between 49 and 93)
Select count(*) from Table Where (Pelvic Incidence between 77 and 106) OR (Lumbar Lordosis Angle between 50 and 59) OR (Sacral Slope between 102 and 106) OR (Grade Of Spondylolisthesis between 11 and 310)

Table 5. 11 Best rules (OR variant)

5.4 Comparison with Other Well-Known Classification Techniques

As the accuracy is very important measure in the domain of medical field, so we compare classification accuracy of our proposed technique with some of the famous well-known classification techniques named: Artificial Neural Network (ANN), Support Vector Machine (SVM), Decision Trees and k-Nearest Neighbor. Open source software named Waikato Environment for Knowledge Analysis (WEKA) [83] is used to conduct the results from the above listed algorithms. All of the results are collected using default settings provided by WEKA and 10 fold cross-validation.

5.4.1 Weka experimental settings

This sub heading briefly states the experiment settings of different algorithm implementations from WEKA.

5.4.1.1 Artificial neural network (ANN)

The ANN implemented in WEKA was multilayer perceptron. It contains single hidden layer, input layer and output layer. Numbers of neurons in input are six, hidden layer contains four neurons and output layer has two neurons. The multilayer perceptron used sigmoid as transfer function. The test mod used was 10-fold cross-validation. Figure 5.2 shows the network architecture of ANN implemented in WEKA.

5.4.1.2 Decision tree

The decision tree implemented in WEKA is of size 19. The number of leaves of the tree is 10. J48 pruned tree learning algorithm is used. The test mod used was 10-fold. Figure 5.3 shows the decision tree implemented in WEKA.

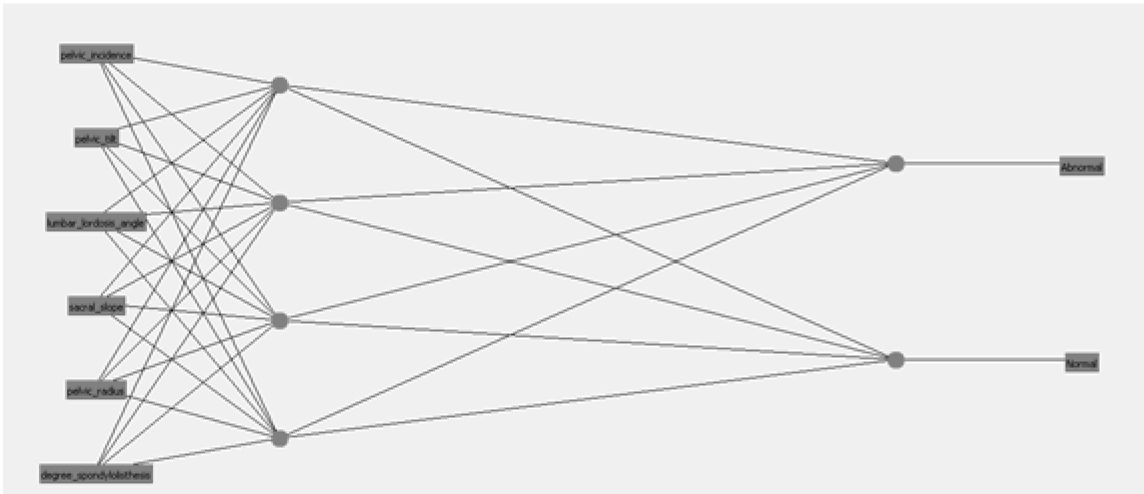

Figure 5.2 ANN Architecture (WEKA)

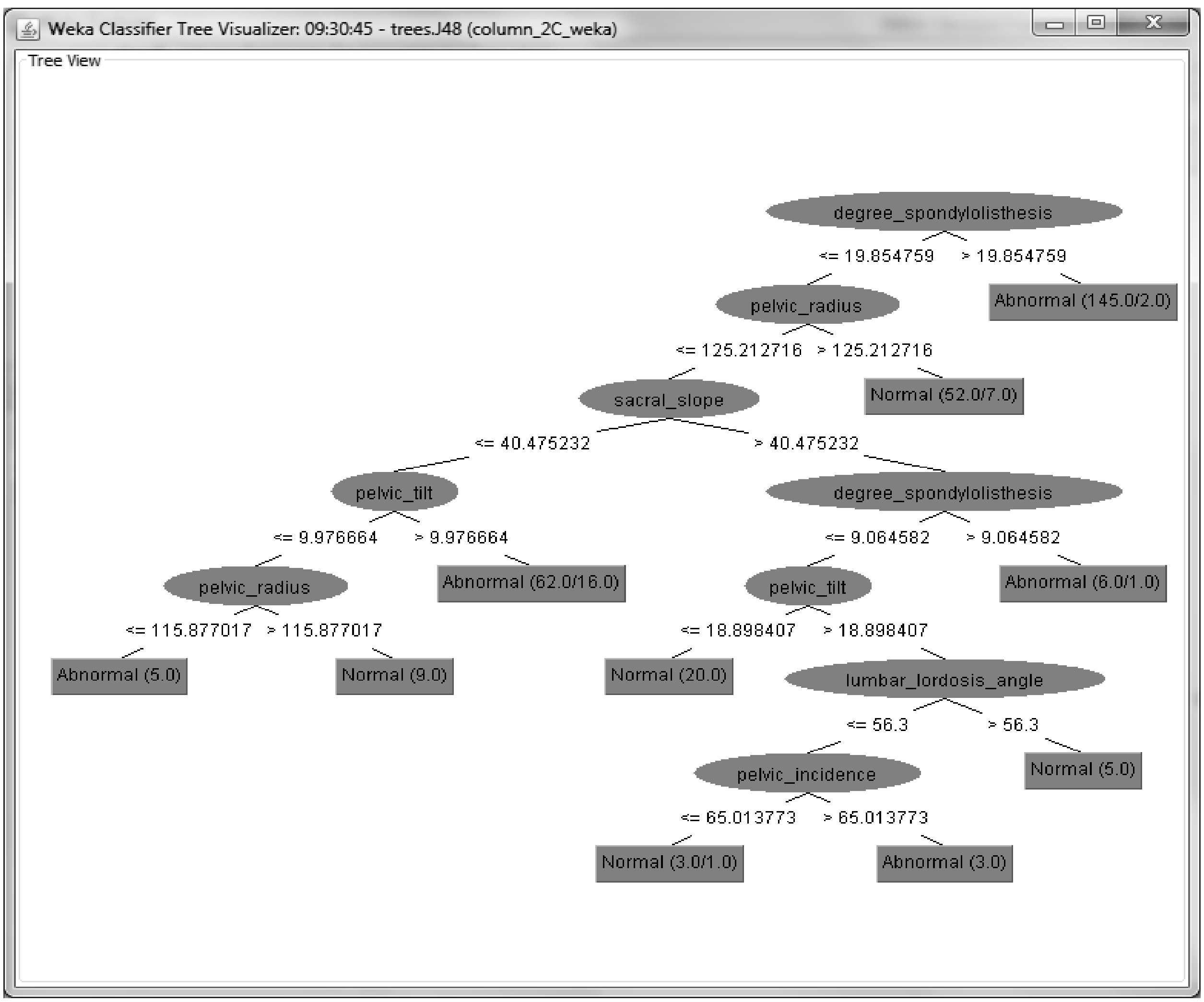

Figure 5.3 Decision Tree (WEKA)

5.4.1.3 K-nearest neighbor (KNN)

The KNN implemented in WEKA is IB1 instance-based classifier. The value chosen for K is 1. The test mod used was 10-fold.

Table 5.12 shows a summary of comparison between proposed technique and ANN, SVM and Decision tree accuracies. Figure 5.4 graphically presents the same comparison for enhanced readability.

Proposed Technique (OR variant)	**91.00%**
Artificial Neural Network (ANN)	**76.41%**
Support Vector Machine (SVM)	**78.70%**
Decision Tree	**81.61%**
k-Nearest Neighbor (KNN)	**81.61%**

Table 5.12 Comparisons between proposed technique and ANN, SVM, decision tree and KNN accuracies

5.5 Summary

The overall performance of the proposed technique is quite good. Proposed technique has resulted in good classification results, however while comparing both of the variants AND & OR, the OR variant reveal improved results than AND variant. Using AND variant, the maximum classification accuracy which is achieved is **76.77**%, whereas **91%** is for OR variant.

Furthermore, in overall comparison when comparing our proposed technique classification accuracy with ANN, SVM and Decision trees classification accuracy, our proposed technique shows better classification accuracy than any of three. Moreover, the proposed technique results are more comprehensible and easily understood.

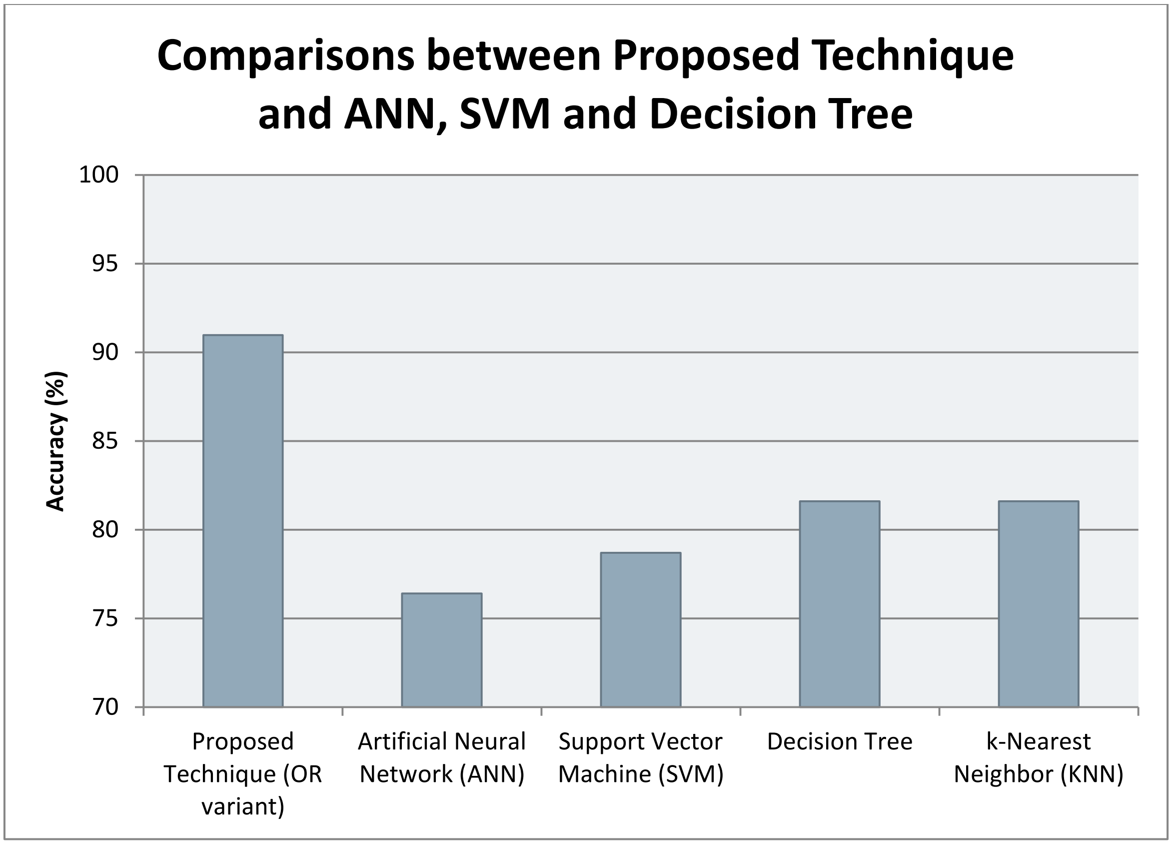

Figure 5.4 Comparisons between proposed technique and ANN, SVM, decision tree and KNN accuracies

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this chapter we conclude the research findings of this thesis and some propositions for the future work that can be done using this thesis as a guideline.

6.1 Conclusion

In this thesis, we have presented an approach that uses rule based classification through Particle swarm optimization technique. The rules are then converted into SQL queries for fast and parallel execution. The proposed method has resulted in good classification results; moreover, the results are comprehensible and easily understood. From the results presented in previous chapters it can be concluded that the classification accuracy of OR variant of proposed technique is better for training and testing than AND variant. We have tested the performance of proposed rule based classification algorithm on data considering six attributes derived from the shape and orientation of the pelvis and lumbar spine to classify orthopedic patients into normal and abnormal category. Where, the abnormal patients suffer from lower back pain, which is a common concern that affects most of the people at some point in their life. The proposed algorithm offers transparent-understandable rules that are discovered from the data in its original form. The algorithm is compared with neural network, support vector machine, decision tree, k-nearest neighbor and the classification results are found encouraging.

6.2 Future Work

This work has opened grounds for future research in the field of rule based classification through Particle swarm optimization technique. It can be expanded for the use of hybrid approach while using the operators (AND/OR) in rule encoding. The proposed technique can also be extended to multi class classification. It can be used for classification task over the variety of data sets.

This work has demonstrated that Particle swarm optimization can be used successfully for discovering classification rules for the diagnosis and classification of one type of vertebral column disorder i.e. back pain. With appropriate data PSOs can also be implemented for the diagnosis of different diseases.

REFERENCES

[1] J W Frymoyer et al., "Risk factors in low-back pain. An epidemiological survey.," *The Journal of bone and joint surgery.*, vol. American volume 65 (2), pp. 213 - 218, 1983.

[2] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, *Advances in Knowledge Discovery and Data Mining*., 1996.

[3] Ajalmar R. da Rocha Neto, Ricardo Sousa, Guilherme de A. Barreto, and Jaime S. Cardoso, "Diagnostic of Pathology on the Vertebral Column with Embedded Reject Option,".

[4] N. Cristianini and J. Shawe-Taylor, "An Introduction to Support Vector Machines," *Cambridge University Press*, 2000.

[5] V. Mangat , "Swarm Intelligence Based Technique for Rule Mining in the Medical Domain," *International Journal of Computer Applications (0975 – 8887),* vol. Volume 4 – No.1, pp. 19-24, July 2010.

[6] M. Salim and X. Yao, "Evolving SQL Queries for Data Mining," in *IDEAL 2002*, Berlin, 2002, pp. 62–67.

[7] J. Vesterstrøm and R. Thomsen, "A Comparative Study of Differential Evolution, Particle Swarm Optimization, and Evolutionary Algorithms on Numerical Benchmark Problems," in *Evolutionary Computation, 2004. CEC2004*, Denmark , 19-23 June 2004, pp. 1980 - 1987 Vol.2.

[8] (2012, Sept.) Publishyourarticles http://www.publishyourarticles.org/knowledge-hub/essay/an-essay-on-the-human-diseases.html

[9] *Dorland's Medical Dictionary*.

[10] Guanabara Koogan, S. J. Hall, and Rio de Janeiro, *Biomecânica Básica*., 2000.

[11] Waddell G, "A New Clinical Model of Low Back Pain and Disability," *The Back Pain Revolution. London, UK; Churchill Livingstone*, 1998.

[12] "On the Application of Ensembles of Classifiers to the Diagnosis of Pathologies of the Vertebral Column: A Comparative Analysis," in *IEEE LATIN AMERICA TRANSACTIONS*, Aug, 2009, pp. 487-496.

[13] C. Shiel William. (2012, June) Medicinenet http://www.medicinenet.com/herniated disc/article.htm

[14] (2012, Dec.) Medicinenet http://www.medicinenet.com/

[15] (2012, Dec.) Sciencephoto. http://www.sciencephoto.com

[16] FG Hepner, T Logan, N Ritter, and N Bryant, "Artificial neural network classification using a," *Photogrammetric Engineering and Remote Sensing*, pp. 469-473, 1990.

[17] L Fausett, *Fundamentals of Neural Networks Architectures, Algorithms, and Applications*.: India: Pearson Education, Inc., 1994/2008.

[18] J. E. Dayhoff and J. M. DeLeo, "Artificial Neural Networks," in *Conference on Prognostic Factors and Staging in Cancer Management: Contributions of Artificial Neural Networks and Other Statistical Methods, CANCER Supplement, American Cancer Society*, vol. Volume 91, Number 8, April 15, 2001, pp. 1615-1635.

[19] S. Haykin, *Neural Networks A Comprehensive Foundation*.: India: Pearson Education, Inc., 1999/2005.

[20] "Diagnosing Skin Diseases Using an Artificial Neural Network," in *Artificial Neural Networks - Methodological Advances and Biomedical Applications*, pp. 255-270.

[21] N. P Reddy and P Gopinath, "Toward Intelligent web monitoring: Performance of committee neural networks vs. single neural network," in *2000 IEEE EMBS international conference on information technology applications in biomedicine*, 2000.

[22] X Wu, J Ghaboussi, and J. H Garret, "Use of neural networks in detection of structural damage," *Computers and Structures*, vol. 42, pp. 649-659, 1992.

[23] AK Skidmore, BJ Turner, and W Brinkhof, "Performance of a neural network: Mapping forests using GIS and remotely sensed data," *Photogrammetric Engineering and Remote Sensing*, pp. 501-514, 1997.

[24] M Pal and MP Mather, "An assessment of the effectiveness of decision tree methods for land," *Remote Sensing of Environment*, pp. 554-565, 2003.

[25] L Breiman, J Friedman, R Olshen, and C Stone, *Classification and Regression*., 1984.

[26] J Quinlan, "Induction of Decision Trees," *Machine Learning*, pp. 81-106, 1986.

[27] J Quinlan, "Simplifying Decision Trees," *International Journal of Man-Machine Studies*, pp. 221-234, 1987.

[28] J Quinlan, "Programs for Machine Learning," *San Mateo: Morgan Kaufmann*, 1993.

[29] L Breiman, J. H Friedman, R. A Olsen, and C. J Stone, *Classification and Regression Trees: An Introduction. Technical Guide #3.*, 1984.

[30] I BRATKO and P MULEC, "An experiment in autoamtic learning of diagnostic rule," pp. 18-25, 1980.

[31] I KONONENKO, I BRATKO, and E ROSKAR, "Experiments in automatic learning of medical diagnostic rules," in *In International School for the Synthesis of Experts Knowledge Workshop*, 1984.

[32] B CESTNIK, I KONENKO, and I BRATKO, "A knowledge elicitation tool for sophisticated users," *In Progress in Machine Learning*, 1987.

[33] P BOURLAS, E GIAKOUMAKIS, and G PAPAKONSTANTINOU, "A knowledge acquisition and management system for ECG diagnosis," , 1999, p. In Proceedings of workshop on Machine Learning in Medical Applications.

[34] P DATTA, W. R SHANKLE, and M PAZZANI, "Applying machine learning to an Alzheimer's database," in *Artificial Intelligence in Medicine*, 1996, pp. 26-30.

[35] Daniel Morariu, Classification and clustering using SVM, 2005.

[36] Alexander Smola and Bernhard Schoslkopf, "Learning with Kernels," *Support Vector Machines, MIT Press, London*, 2002.

[37] Nello Cristianini and John Shawe- Taylor, "An Introduction to Support Vector Machines and other kernel-based learning methods," *Cambridge University Press*, 2000.

[38] Chih-Wei Hsu, Chih-Chang Chang, and Chih-Jen Lin, "A Practical Guide to Support Vector Classification," Department of Computer Science and Information Engineering National Taiwan University, 2003.

[39] Bernhard Sh¨olkopf, "Survival-time classification of breast cancer patients," National Taiwan University of Science and Technology, Taipei, Taiwan, 2001.

[40] Vojislav Kecman and Ming Huang, "Gene extraction for cancer diagnosis by support vector machines," *Artificial Intelligence in Medicine*, pp. 185–194, 2005.

[41] S Mukherjee et al., "Support Vector Machine Classification of Microarray Data," in *AI Memo 1677, CBCL MIT*, 1998, p. 182.

[42] (2012, Oct.) Broad Institute. http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi

[43] T. R. Golub et al., "Molecular Classification of Cancer: ClassDiscovery and Class Prediction by Gene Expression Monitoring," pp. 531-537, 1999.

[44] J. C Chris and Burges, "Improving the accuracy and speed of support vector machines," *Advances in Neural Information Processing Systems*, no. MIT PRESS, pp. 375-381, 1997.

[45] Carl Gold, Alex Holub, and Peter Sollich, "Bayesian approach to feature selection and paramaeter tuning for support vector machine classifiers," *Neural Networks*, no. 18, pp. 693-701, 2005.

[46] M Khan, Q Ding, and W Perrizo, "K-nearest Neighbor Classification on Spatial Data Stream Using P-trees," in *Proceedings of PAKDD; Springer-Verlag, Lecture Notes in Artificial Intelligence 2336*, May, 2002, pp. 517-528.

[47] B. S Sebastiano and F Roli, "Classification of Multi-sensor Remote-sensing Images by Structured Neural Networks," in *IEEE Transactions on Geoscience and Remote Sensing 33(3)*, pp. 562-578.

[48] Nitin Bhatia and Vandana, "Survey of Nearest Neighbor Techniques," *IJCSIS*, vol. 8, 2010.

[49] T. M. Cover and P. E. Hart, "Nearest Neighbor Pattern Classification," *IEEE Trans. Inf.Theory*, vol. IT-12, pp. 21-27, 1967.

[50] J. Kennedy and R. Eberhart, "Particle Swarm Optimization.," in *In Proceedings of IEEE International Conference on Neural Networks*, vol. 4, Perth, Australia, 1995, pp. 1942-1948.

[51] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence.*, 1998, pp. 69 - 73.

[52] Xuedan Liu, Qiang Wang, Haiyan Liu, and Lili Li, "Particle Swarm Optimization with Dynamic Inertia Weight and Mutation," in *Genetic and Evolutionary Computing, 2009. WGEC '09.*, Guilin, 14-17 Oct. 2009, pp. 620 - 623.

[53] M.G.H. Omran and S. Al-Sharhan, "Using opposition-based learning to improve the performance of particle swarm optimization," in *Swarm Intelligence Symposium, 2008. SIS 2008. IEEE*, 21-23 Sept. 2008, pp. 1 - 6.

[54] Xiaoling Wu and Min Zhong, "Particle swarm optimization based on power mutation," in *Computing, Communication, Control, and Management, 2009.*, 8-9 Aug. 2009, pp. 464-467.

[55] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization," in *Evolutionary Computation, 1999. CEC 99*, 1999, pp. 1951-1957.

[56] Yuhui Shi and R.C. Eberhart, "Fuzzy adaptive particle swarm optimization," in *Evolutionary Computation, 2001. Proceedings of the 2001 Congress*, 2001, pp. 101 - 106.

[57] Hui Wang, Yong Liu, Changhe Li, and Sanyou Zeng, "A Hybrid Particle Swarm Algorithm with Cauchy Mutation," in *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*, Honolulu, HI, 2007, pp. 356 - 360.

[58] Satyobroto Talukder, Mathematical Modelling and Applications of Particle Swarm Optimization, 2011.

[59] Zheng Jianchao, Jie Jing, and Cui Zhihua, "Particle swam optimization Algorithm[M].," in *Science Publishing Company of Beijing.*, Beijing, 2004.

[60] Rui Mendes, Population Topologies and Their Influence in Particle Swarm Performance, 2004.

[61] M. Lovberg, Improving Particle Swarm Optimization by Hybridization of Stochastic Search Heuristics and Self Organized Critically, Master's Thesis., 2002, Department of Computer Science, University of Aarhus, Denmark.

[62] F. Van den Bergh, An Analysis of Particle Swarm Optimizers, PhD Thesis, 2002, Department of Computer Science, University of Pretoria, South Africa.

[63] T. Krink, J. Vesterstrøm, and J. Riget., "Particle Swarm Optimization with Partial Particle Extension," in *Fourth Congress on Evolutionary Computation*, 2002.

[64] M. Lovberg and T. Krink., "Extending Particle Swarm Optimizers with Self-Organized Criticality," in *Fourth Congress on Evolutionary Computation vol 2*, 2002, pp. 1588-1593.

[65] "Adaptive Particle Swarm Optimization: Detection and Response to Dynamic Systems.," in *Proceedings of congress on Evolutionary Computation*, Hawaii, USA, 2002, pp. 1666-1670.

[66] Y. Shi and R. Eberhart, "Fuzzy Adaptive Particle Swarm Optimization," in *Congress on Evolutionary Computation, Seoul, S. Korea*, 2001.

[67] "Using Selection to Improve Particle Swarm Optimization," in *International Conference on Evolutionary Computation*, New Jersey, USA, 1998, pp. 84-89.

[68] T. Krink and M. Løvbjerg., "The LifeCycle model: Combining Particle Swarm Optimisation, Genetic Algorithms and HillClimbers," in *Parallel Problem Solving from Nature VII*, 2002, pp. 621-630.

[69] K. Veeramachaneni, T. Peram, C. Mohan, and L. Osadciw., "Optimization Using Particle Swarm with Near Neighbor Interactions," *Lecture Notes Computer Science vol. 2723, Springer Verlag*, 2003.

[70] Girma S. Tewolde and Darrin M. Hanna, "Particle Swarm Optimization for Classification of Breast Cancer Data using Single and Multisurface Methods of Data Separation," in *IEEE EIT 2007 Proceedings*, 2007.

[71] K.RAJIV GANDHI, MARCUS KARNAN, and S.KANNAN, "Classification Rule Construction Using Particle Swarm Optimization Algorithm for Breast Cancer Data Sets," in *International Conference on Signal Acquisition and Processing*, 2010, pp. 233 - 237.

[72] Z Wang, X Sun, and D Zhang, "Classification Rule Mining Based on Particle Swarm Optimization," in *RSKT 2006*, Berlin, 2006, pp. 436–441.

[73] A. B. De Carvalho, T Savegnago, and A Pozo, "Swarm Intelligence for Rule Discovery in Data Mining," in *ICEIS 2010*, Funchal, Madeira, Portugal, 2010, pp. 314-319.

[74] Y Liu, Z Qin, Z Shi, and J Chen, "Rule Discovery with Particle Swarm Optimization," in *AWCC 2004*, Berlin , 2004, pp. 291-296.

[75] W C Yeh, "Novel swarm optimization for mining classification rules on thyroid gland data," *Information Sciences*, pp. 65-75, 2012.

[76] W.C. Yeh, W.W. Chang, and Y.Y. Chung, "A new hybrid approach for mining breast cancer pattern using discrete particle swarm optimization and statistical method," in *Expert Systems with Applications*, 2009, pp. 8204 – 8211.

[77] T Sousa, A Silva, and A Neves, "A Particle Swarm Data Miner," in *EPIA 2003*, Berlin, 2003, pp. 43–53.

[78] M Salim and X Yao, "Evolving SQL Queries for Data Mining," in *IDEAL 2002*, Berlin, 2002, pp. 62–67.

[79] M Ghannad-Rezaie, H Soltanain-Zadeh, M.-R Siadat, and K.V Elisevich, "Medical Data

Mining using Particle Swarm Optimization for Temporal Lobe Epilepsy," in *Evolutionary Computation, 2006. CEC 2006*, Vancouver, BC, 2006, pp. 761 - 768.

[80] R Poli, J Kennedy, and T Blackwell, "Particle Swarm Optimization : An Overview," *Swarm Intelligence*, pp. 33-57, 2007.

[81] J Kennedy, "Small worlds and mega-minds: Effects of Neighborhood Topology on Particle Swarm Performance," in *In Proceedings of Congress on Evolutionary Computation*, 1931–1938, p. 1999.

[82] (2012, Dec.) Laser Spine Institute. http://www.laserspineinstitute.com/back_problems/spondylolisthesis/spondylolisthesis_resources/grades/

[83] I. H. Witten and E. Frank, "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations:," in *Morgan Kaufmann*, 2000.

[84] De Falco, A. Della Cioppa, and E Tarantino, "Facing classification problems with Particle Swarm Optimization," *Elsevier*, pp. 652–658, 2007.

[85] V D Merwe and A.P. Engelbrecht, "Data clustering using particle swarm optimization," in *Evolutionary Computation, 2003. CEC '03*, 2003, pp. 215 - 220.

[86] S. L. Oliveira, S. A. Britto, and R. Sabourin, "Optimizing Class-Related Thresholds with Particle Swarm Optimization," in *Proceedings of International Joint Conference on Neural Networks*, Montreal, Canada, 2005, pp. 1511-1516.

[87] A Abraham, S Das, and S Roy, "Swarm Intelligence Algorithms for Data Clustering," in *SOFT COMPUTING FOR KNOWLEDGE DISCOVERY AND DATA MINING 2008*, 2008, pp. 279-313.

[88] S. M Uma, K. Rajiv Gandhi, and E Kirubakaran, "A Hybrid PSO with Dynamic Inertia Weight and GA Approach for Discovering Classification Rule in Data Mining," *International Journal of Computer Applications*, pp. 32-37, 2012.

[89] M Gupta, "Application of Weighted Particle Swarm Optimization in Association Rule Mining ," *International Journal of Computer Science and Informatics*, pp. 69-74.

[90] R.J. Kuoa, C.M. Chao, and Y.T. Chiuc, "Application of particle swarm optimization to association rule mining," *Elsevier*, pp. 326-336, 2011.

[91] M Omran, A Salman, and A P Engelbrecht, "Image Classification Using Particle Swarm Optimization," in *International Conference on Simulated Evolution And Learning*.