Classification and Clustering Using Intelligent Techniques: Application to Microarray Cancer Data

Anita Bai

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India

Classification and Clustering Using Intelligent Techniques: Application to Microarray Cancer Data

Thesis submitted in partial fulfillment of the requirements for the degree of

Master of Technology

in

Computer Science and Engineering

(Specialization: Software Engineering)

by

Anita Bai

(Roll No: 211CS3291)

under the supervision of

Prof. S. K. Rath

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela, Odisha, 769 008, India

May 2013

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

Certificate

This is to certify that the work in the thesis entitled ***"Classification and Clustering Using Intelligent Techniques: Application to Microarray Cancer Data"*** by ***Anita Bai*** is a record of an original research work carried out by her under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Master of Technology with the specialization of Software Engineering in the department of Computer Science and Engineering, National Institute of Technology Rourkela. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Place: NIT Rourkela
Date: 3 June 2013

(Prof. S. K. Rath)
Professor, CSE Department
NIT Rourkela, Odisha

Acknowledgment

I am grateful to numerous local and global peers who have contributed towards shaping this thesis. At the outset, I would like to express my sincere thanks to Prof. Santanu Kumar Rath for his advice during my thesis work. As my supervisor, he has constantly encouraged me to remain focused on achieving my goal. His observations and comments helped me to establish the overall direction of the research and to move forward with investigation in depth. He has helped me greatly and been a source of knowledge.

I am very much indebted to Prof. Ashok Kumar Turuk, Head-CSE, for his continuous encouragement and support. He is always ready to help with a smile. I am also thankful to all the professors of the department for their support.

I am really thankful to my all friends, especially my 'puzzle' group of best friends. I would also like to thank all the Ph.D. scholars and especially to Ph.D. scholars Swati Vipsita and Yeresime Suresh for helping me and giving advise. My sincere thanks to everyone who has provided me with kind words, a welcome ear, new ideas, useful criticism, or their invaluable time, I am truly indebted.

I must acknowledge the academic resources that I have got from NIT Rourkela. I would like to thank administrative and technical staff members of the Department who have been kind enough to advise and help in their respective roles.

Last, but not the least, I would like to dedicate this thesis to my family, for their love, patience, and understanding.

Anita Bai

Email : anitaahirwarnitr@gmail.com

Abstract

Analysis and interpretation of DNA Microarray data is a fundamental task in bioinformatics. Feature Extraction plays a critical role in better performance of the classifier.

We address the dimension reduction of DNA features in which relevant features are extracted among thousands of irrelevant ones through dimensionality reduction. This enhances the speed and accuracy of the classifiers. Principal Component Analysis (PCA) is a technique used for feature extraction which helps to retrieve intrinsic information from high dimensional data in eigen spaces to solve the curse of dimensionality problem. The curse of dimensionality means $n >> m$, where n is a large number of features and m is a small number of samples (may be too less). Neural Networks (NN) and Support Vector Machine (SVM) are implemented and their performances are measured in terms of predictive accuracy, specificity, and sensitivity. First, we implement PCA for significant feature extraction and then FFNN trained using Backpropagation (BP) and SVM are implemented on the reduced feature set.

Next, we propose a Multiobjective Genetic Algorithm-based fuzzy clustering technique using real coded encoding of cluster centers for clustering and classification. This technique is implemented on microarray cancer data to select training data using multiobjective genetic algorithm with non-dominated sorting (MOGA-NSGA-II). The two objective functions for this multiobjective techniques are optimization of cluster compactness as well as separation simultaneously. This approach identifies the solution i.e. the individual chromosome which gives the optimal value of the compactness and separation. Then we find high confidence points for these non-dominated set using a fuzzy voting technique. Support Vector Machine (SVM) classifier is further trained by the selected training points which have high confidence value. Then remaining points are classified by trained SVM classifier. Finally, the four clustering label vectors through majority voting ensemble are combined, i.e., each point is assigned a class label that obtains

the maximum number of votes among the four clustering solutions. The performance of the proposed MOGA-SVM, classification and clustering method has been compared to MOGA-BP, SVM, BP. The performance are measured in terms of Silhoutte Index, ARI Index respectively. The experiment were carried on three public domain cancer data sets, viz., Ovarian, Colon and Leukemia cancer data to establish its superiority.

Keywords: *Cancer Classification; Feature Reduction; Multiobjective genetic algorithm; Neural Network; Pareto-optimality; Principal components; Support Vector Machine(SVM)*

Contents

Certificate i

Acknowledgement ii

Abstract iii

List of Figures viii

List of Tables ix

1 Introduction 2

1.1 Introduction 2

1.1.1 DNA 3

1.1.2 Cancer Classification 3

1.1.3 Cancer Clustering 5

1.2 Basic Concepts of Artificial Neural Network 6

1.2.1 Activation Function 7

1.2.2 NN Architecture 9

1.2.3 Learning Paradigm 11

1.3 Neural Network for Cancer Classification 12

1.4 Motivation 12

1.5 Objective: 13

1.6 Thesis Organization 14

2 Related Work 16

3 Feature Extraction and Cancer Classification 19

3.1 Dimension Reduction 19

3.1.1 Objectives of Feature Extraction 20

3.1.2 Dimension Reduction using Principal Component Analysis: . 21
3.2 Classifiers 22
3.2.1 Back Propagation Neural Networks Classifier 22
3.2.2 BP Neural Network Classifier Hybrid with PCA Algorithm . 23
3.2.3 Why SVM for cancer classification 24
3.2.4 The SVM Classifier and Kernel Selection 24
3.3 Proposed Work I: 27
3.3.1 Data Preprocessing and Cleaning 27
3.3.2 Data Normalization 27
3.4 Implementation 31
3.4.1 Data Sets 31
3.4.2 Input Parameters 32
3.4.3 Performance Measures 33
3.5 Numerical Simulation, Results and Discussion 33
3.6 Conclusion 36

4 Multiobjective Genetic Algorithm-Based Fuzzy Clustering combining with Support Vector Machine for Clustering and Classification 38

4.1 Evolutionary Algorithms 39
4.1.1 Brief Overview of GA 39
4.1.2 Single Objective Optimization Problem (SOOP) 39
4.1.3 Multiobjective Optimization 40
4.1.4 Brief Overview of MOGA 41
4.1.5 Fast Non-Dominated Sorting 42
4.1.6 NSGA-II 44
4.2 Proposed Work II: MOGA-SVM 47
4.3 IMPLEMENTATION 51
4.3.1 Data Sets 51
4.3.2 Parameters for MOGA-SVM 52
4.3.3 Performance metrics 53
4.3.4 Result 55

4.4 Conclusion 57

5 Conclusion and Future Work **59**

Bibliography **60**

Dissemination of Work **64**

List of Figures

1.1 The Mathematical Model of Artificial Neuron 6
1.2 Multilayer Feedforward Network 10
1.3 Recurrent Network 10
1.4 Brief overview of the entire process 12

3.1 Multilayered Backpropagation Neural Network 23
3.2 SVM Classifier 25
3.3 PCA-SVM or PCA-BPNN classifiers for cancer data 28
3.4 Schematic illustration of the proposed method for Leukemia cancer data 29
3.5 3D and 2D Schematic representation of data across first three PC's and two PC's (Leukemia Cancer data set) 34
3.6 3D and 2D Schematic representation of data across first three PC's and two PC's (Ovarian Cancer data set) 34
3.7 Plot showing Accuracy vs. No. of PC's using PCA 35
3.8 2D Schematic representation of data across first two features (Leukemia data set) 35

4.1 Schematic representation of Pareto-optimal solutions 44
4.2 Process of NSGA-II 45
4.3 Schematic representation of chromosome 48
4.4 Schematic representation of Pareto-optimal fronts produced by MOGA-NSGA-II for cancer data 55

List of Tables

3.1 Accuracy vs. No. of PC's using PCA-SVM (Leukemia Cancer data set) 35
3.2 Classification Results: SVM Kernels 36
3.3 Classification Results: Traditional BP, SVM, PCA-BP, and PCA-SVM 36

4.1 PARAMETER VALUES FOR MOGA-SVM 53
4.2 Performance of Fuzzy Compactness (π) and Fuzzy Seperation (sep) 55
4.3 Classification Results: SVM Kernels with MOGA 56
4.4 Classification Results: Traditional BP, SVM, MOGA-BP, and MOGA-SVM 56
4.5 Comparison of different algorithms in terms of silhouette score and ARI Index for cancer data sets 57

Chapter 1

Introduction

Chapter 1

Introduction

1.1 Introduction

Much research is being done in the academics as well as the industries towards the application of bioinformatics that uses computational approaches to solve biological problems. The goal of this field is to retrieve, analyze and interpret the vast and complex genomic data sets that are uncovered in large volumes of genes in molecular biology. Biological data mining posses various challenges like gene discovery, drug discovery, gene finding, revealing unknown relationship with respect to structure and function of genes to understand biological systems. This field faces demands for immediate prediction and classification due to the availability of DNA cancer data, structure information of proteins and microarray technology to provide dynamic information about thousand of genes in data. The aims of Bioinformatics are:

1. To organize data in a way that allows researcher and practitioners to access existing information and to submit new entries as they are produced.

2. To develop tools, softwares and resources that aid in the analysis and management of data.

3. To use this data to analyze and interpret the results in a biologically meaningful manner.

4. To help practitioners in the pharmaceutical industry in understanding the microarray cancer data structures which helps the disease prediction easy.

1.1.1 DNA

DNA (also known as **deoxyribonucleic acid**) is the hereditary material in humans and almost all other organisms. DNA is the material present in our cells that makes up our genes. Genes carry and follow the instructions our bodies use to grow and function. Nearly every cell in a persons body has the same DNA. Most DNA is presented in the cell nucleus (where it is called nuclear DNA), and a less amount of DNA can also be found in the mitochondria (where it is called mitochondrial DNA or mtDNA). The information under DNA is stored as a code made up of four chemical bases: adenine (A), guanine (G), cytosine (C), and thymine (T). Sequence of nucleotides in a DNA is defined by the sequence of a gene, which is encoded in the genetic code i.e. 4 standard of nucleotides. One strand is a polynucleotide (a sequence of nucleotides of 4 types); the second strand has their complementary base pairs (A = T , C = G). DNA consists of 2 strands of nucleotides forming a double helix structure. DNA nucleotides vary depending on 4 possible nitrogenous bases A,C,T,G. In the basis of accurance of pattern ,find the gene values and partition in classes.

1.1.2 Cancer Classification

Cancer classification is a challenging task of bioinformatics. For cancer classification we concentrate on behaviour of DNA microarray. The Microarray technology (DNA microarray) [1] allows us to measure the expression levels of thousands of genes simultaneously, providing great chance for cancer diagnosis and prognosis. A microarray cancer data having 'p' genes and 'n' samples (observations) are typically organized in a 2D matrix $X = [a_{ij}]$ of size $p \times n$. Each element a_{ij} gives the expression level of the i^{th} gene at the j^{th} sample.

$$X_{n\times p} = \begin{array}{c} \\ s_1 \\ s_2 \\ \vdots \\ s_n \end{array} \begin{array}{c} \begin{array}{cccc} f_1 & f_2 & \cdots & f_p \end{array} \\ \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{np} \end{pmatrix}_{samples\times features} \end{array}$$

x_{np} = sample s_1 have expression measures for genes from f_1 to f_p .
$s_i = (a_{i1}, a_{i2}, ..., a_{ip})-$ gene expression profile/feature vector for sample i
f_j = set of samples a_{ij} for feature j.
where, i = 1, . . . , n. and j= 1,....., p.

When microarray datasets are organized as samples versus gene fashion, then they are very helpful for classification of different types of tissues and identification of those genes whose expression levels are good diagnostic indicators. The microarray datasets, where the tissue samples represent the samples from cancerous (malignant) and non-cancerous (benign) cells, the classification of them will result in binary cancer classification. On the other hand, if the samples are from different subtypes of cancer, then it becomes the problem of multi-class cancer classification.

The task for cancer classification are of two aspects: identifying new cancer classes and assigning genes to known classes, which are called class discovery and class prediction [2]. DNA microarray technology is a promising tool for cancer diagnosis. It generates large-scale gene expression profiles that include valuable information on organization as well as cancer [3].

Classification, or supervised learning is one of the major data mining processes. Classification is concerned with assigning memberships to samples based on supervised expression patterns. The classification of data has two stages. In the first stage, a model is determined from a set of data called training data (the classes of which have been established beforehand). This model is shown as rules or mathematical formulae. In the second stage the correctness of the evolved model is estimated. This is done by studying the results of the evolved models function on a set of data (test data). The classes of the test data also are determined beforehand. DNA microarray cancer data classification, which determines a new cancer data belongs to which class it helps to find types of cancer. The aim of classification is to predict target class for the given unknown input. There are many approaches available for classification tasks such as statistical techniques, decision trees, fuzzy logic, neural networks etc.

1.1.3 Cancer Clustering

Clustering an unsupervised classification technique, is the process of grouping or organizing a set of objects into distinct group based on some similarity or dissimilarity measure among the individual objects, such that the objects in the same group are more similar to each other than those in other groups. Clustering, an important microarray analysis tool, is used to identify the sets of genes with similar expression profiles. Clustering methods partition a set of objects into groups based on some similarity/dissimilarity metric where the value of may or may not be known a priori. Clustering can be mainly divided into two types Hard Clustering and Soft Clustering.

Hard Clustering

Hard Clustering is based on classical set theory, and in this method of clustering the object either does or does not belong to a cluster [4]. If each data point is assigned to a single cluster, then the clustering is called crisp (hard) clustering. In Hard clustering data is partitioned into specified number of mutually exclusive subsets. Using hard partitioning for algorithms based on analytic functional causes difficulties because hard partitioning is discrete in nature and also since this functional are not differentiable.

Soft (Fuzzy) Clustering

If a data point has certain degrees of belongingness to each cluster, the partitioning is called fuzzy. In Soft Clustering [4], unlike hard clustering the object doesn't belong to a particular cluster rather an object belongs to more than one cluster simultaneously with different degree of membership and with every object there is an associated set of membership levels. The membership level indicates the strength of the association between that object and a particular cluster. Objects on the boundaries between several classes are assigned a membership value between 0 and 1 indicating partial membership rather than they are not forced them to fully belong to a single cluster.

1.2 Basic Concepts of Artificial Neural Network

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. Advantages of ANN include:

1. Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.

2. Self-Organization: An ANN can create its own organization or representation of the information it receives during learning time.

3. Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.

4. Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

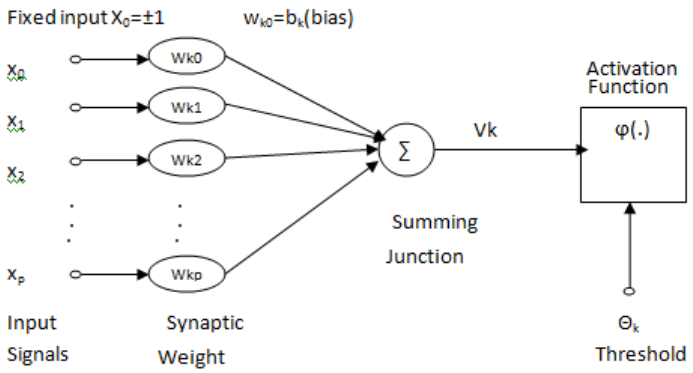


Figure 1.1: The Mathematical Model of Artificial Neuron

An ANN is typically defined by three types of parameters:

1. The interconnection pattern i.e. the synaptic weight between different layers of neurons.

2. The learning process for updating the weights of the interconnections.

3. The activation function that converts a neuron's weighted input to its output activation.

In mathematical form for a given artificial neuron, let there be m + 1 inputs with signals x_0 to x_m with associated synaptic weights w_0 through x_m. Usually, the x_0 input is assigned the value +1, which makes it a bias input with $w_{k0} = b_k$=0. Hence only m actual inputs to the neuron, from x_1 to x_m. The output of k^{th} neuron is defined by Y which is computed using an activation function φ(x) :

$$y_k = \varphi(\sum_{j=0}^{m} w_j x_j) \tag{1.1}$$

Where $v_k = \sum_{j=0}^{m} w_j x_j$ is the summing function in which each input is multiplied by the associated synaptic weight and then added. The mathematical model is illustrated in Fig 1.3.

1.2.1 Activation Function

In computational networks, the activation function defined as the function which generates the output for a given set of inputs. It acts as a non-linear filter, which is one of the important parameters of ANN that characterizes the architecture. The behavior of neural networks depend upon the choice of activation function [5], i.e., how they map input data to output data considering weights of those interconnections. Activation functions with a bounded range are often called squashing functions.

Activation function can be of following types:

1. **Linear function:** Here the input units use the identity function. A linear combination formed where the weighted sum input of the neuron with a linearly dependant bias becomes the system output. Specifically:

$$g(y) = y \qquad where\ y = \sum w_i x_i + b$$

2. **Threshold function:** This function is also known as Step function or Heaviside function. Here sum is compared with a threshold value θ.

$$\begin{cases} g(x) = 1 & if(x \geq \theta) \\ g(x) = 0 & if(x < \theta) \end{cases}$$

This kind of function is often used in single layer networks and called as binary step function.

3. **Signum function:** This function is also known as Quantizer function. It is especially advantageous for use in neural networks because it is easy to differentiate and can dramatically reduce the computation burden for training. It applies to applications whose desired output values are between -1 and 1.

$$\begin{cases} g(x) = 1 & if(x \geq \theta) \\ g(x) = -1 & if(x < \theta) \end{cases}$$

4. **Sigmoidal function:** This function is a continuous function that varies between asymptotic values 0 and 1 or 1 and -1. The sigmoidal function can be described as

$$g(x) = \frac{1}{1 + e^{(-\alpha x)}}$$

where α is the slope parameter, which adjust the abruptness of the function due to the change in asymptotic values.

5. **Ramp function:** The ramp function combines the step function with a linear output function. As long as the activation is smaller than the threshold value θ_1, the neuron shows the output $y_i = 0$; if the activation exceeds the threshold value θ_2, the output is y_i=1. The neuron's output for activations in the interval between the two threshold values $\theta_1 \leq z_i$ (x ,$w_i \leq \theta_2$) is determined by a linear interpolation of the activation.

$$y_i = \mathrm{f}(z_i) = \begin{cases} 0 & if(z_i \leq \theta_1) \\ (z_i - \theta_1) \cdot \frac{1}{\theta_2 - \theta_1} & if(\theta_1 \leq z_i \leq \theta_2) \\ 1 & if(z_i \geq \theta_2) \end{cases}$$

Activation functions for the hidden layers are needed to deploy non-linearity into the networks which makes multi-layer networks so powerful. The sigmoid function

always been the common choice, either in symmetric [-1, 1] or asymmetric [0, 1] form. The sigmoid function is global in nature i.e. it divides the feature space into two halves, one for the response is approaching 1 and another for (0/-1). Hence it is very efficient for making indiscriminate target value distribution in the feature space.

1.2.2 NN Architecture

- **Single layer Feed Forward Network:** This type of network comprise of two layers, namely the input layer and output layer. Feed-forward ANNs (figure) allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect the previous layer neurons. Feed-forward ANNs tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. The input layer neuron receives the input signals and the output layer neuron receives the output signals. The interconnected links carry the synaptic weights. Despite the two layers, the network is termed single layer since it is the output layer alone which performs computation. The input layer merely transmits signals to the output layer hence, the name single layer feed forward network. Such a network is said to be feed forward or acyclic in nature.

- **Multilayer Feed Forward Network:** This class of feed forward neural network distinguishes itself by the presence multiple layers. The architecture of this network posses input layer, output layer and one or more intermediary layers called hidden layers, whose computation nodes are correspondingly called hidden neurons or hidden units. The function of hidden neuron is to intervene between the external input and network output in some useful manner. By adding one or more hidden layers, the network is enabled to extract higher order statistics i.e. valuable when size of input layer is large. A multilayer feed forward network with m input neurons, h1 neurons in the first hidden layer, h2 neurons in the second hidden layer and n output

neurons in the output layer is referred as m-h1-h2-n network. The neural network is said to be fully connected if every node in each layer is connected to every other node in adjacent forward layer. If some of the communication links are missed then the network is a partially connected network. Fig illustrates a multilayer feed forward network:

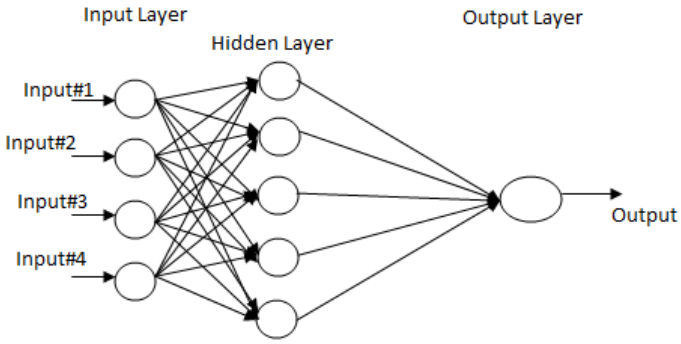


Figure 1.2: Multilayer Feedforward Network

- **Recurring Network:** These networks differ from feed forward network architectures in the sense that there is at least one feedback loop. Thus, in these networks there could exist one layer with feedback connections .There could also be neurons with self-feedback links i.e. the output of a neuron is feedback into itself as input. The presence of feedback loops in the recurrent structure has a profound impact on the learning capability of the network and its performance.

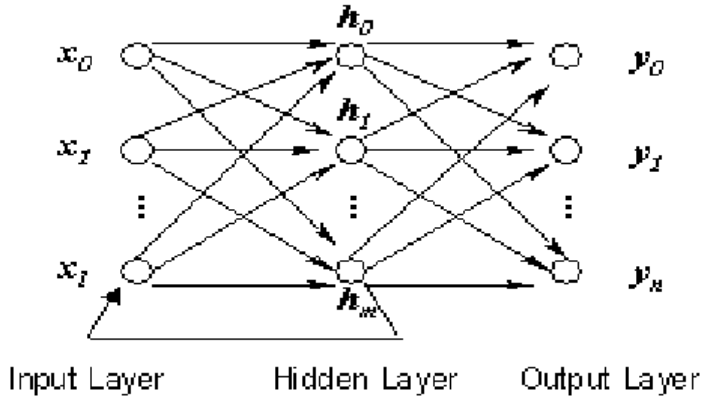


Figure 1.3: Recurrent Network

1.2.3 Learning Paradigm

Learning is a process by which the free parameters of neural network are adapted through a process of stimulation by the environment in which the network is embedded. It can be broadly classified into 3 categories: supervised learning, unsupervised learning, reinforcement learning.

- **Supervised Learning:** It is also referred as learning with a teacher. For every input pattern a desired output pattern (targeted output) is provided. The difference between the computed output and desired output generates the error in the network, which used to change network parameters for the improvement in networks performance. Error-correction learning and stochastic learning are generally used in supervised learning process. Its task often include function approximation problem (regression) i.e. a given training data consisting of pairs of input patterns x, and corresponding target t, the goal is to find a function f(x):x $\rightarrow$ y that matches the desired response (y) for each training input.

- **Unsupervised Learning:** In this paradigm no teacher is to oversee the learning process i.e. no target output is given for the network. Hence the network learns by its own through discovering and adapting the features, regulations, correlations or categories in the input pattern automatically. It usually performs the same task as an auto-associative network does, compressing the information from inputs. It is also referred to as self-organization, that it self-organizes data presented to the network and detects their emergent collective properties. Paradigms of unsupervised learning are Hebbian learning and Competitive learning. The identification of new tumor classes using gene expression profiles

- **Reinforcement Learning:** In this learning paradigm, a teacher does not present the expected answer but only indicates if the computed output is correct or incorrect. The learning of an input-output mapping is performed through continued interaction with the environment that will minimize a

scalar index of performance. The information provided helps the network in its learning process. A reward is given for a correct answer computed and a penalty for a wrong answer.

1.3 Neural Network for Cancer Classification

NN are an effective tool in the field of cancer Classification. In the training stage (Approximation), neural networks extract the features of the input data. In the recognizing stage (Generalization), the network distinguishes the pattern of the input data by the features, and the result of recognition is greatly influenced by the hidden layer. Every instance in any dataset used by machine learning algorithms is represented using the same set of features. The features may be continuous, real coded, categorical or binary.

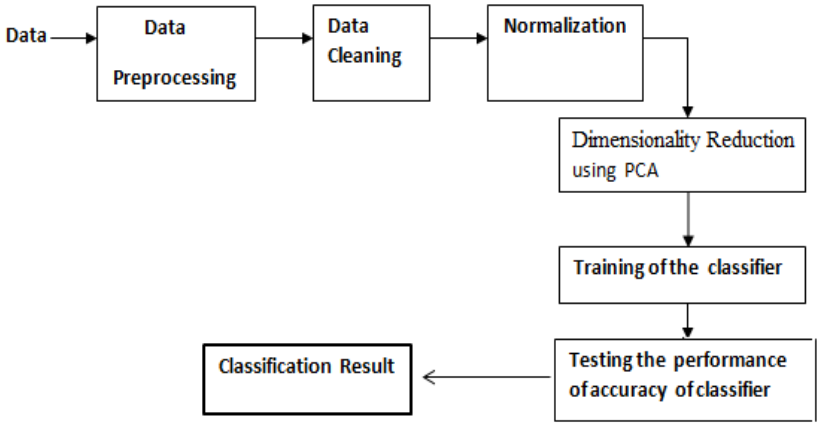


Figure 1.4: Brief overview of the entire process

If instances are given with known labels (the corresponding correct outputs) then the learning is called supervised, in contrast to unsupervised learning, where instances are unlabeled. Learning is usually accomplished by adjustment and modification of the connected synaptic weights.

1.4 Motivation

It is required to develop an intelligent system to classify the microarray cancer data with high accuracy. Many machine learning techniques have been success-

fully implemented for classification and clustering. In the first stage, training of the classifier is done with known labeled samples. After the classifier is successfully built, unknown test samples are used to measure the effectiveness of the classifier. Among the soft computing paradigms, Neural Network (NN) approach efficiently handles linear and non-linear data classification tasks. Dimension reduction used for remove Noisy or irrelevant features which gives negative effect on accuracy of any classifier. Neural networks have been chosen as technical tools for the microarray classification task because extracted features of the DNA data are distributed in a high dimensional space with complex characteristics. SVM are used for cancer classification because correctly separate entities into appropriate classes. The main motivation for dimension reduction are reduce curse of dimensionality problem. Different clustering algorithms usually attempt to cluster the gene expression data but in this report multiobjective optimization of cluster validity measures such as compactness and separation among clusters in microarray cancer data has been proposed and computation of cluster modes is costlier than that of cluster means, the algorithm needs to find the membership matrices that takes a reasonable amount of time. However, as fuzzy clustering is better equipped with to handle overlapping clusters.
Two most desirable features of an multiobjective Genetic algorithm:

- Convergence to Pareto optimal front
- Maintenance of Diversity.

1.5 Objective:

In the experiment on genes we can find the gene which are affected by cancer are identified by classification and clustering. Correct prediction of unknown genes or newly discovered mainly concerns the biologists or researchers for prediction of cancers in cell, molecular function, drug discovery, medical diagnosis etc.

- An efficient classification technique needs to be implemented or develop an efficient classifier to correctly classify the unknown genes so that the cancer

patient are diagonised correctly and this treatment can be done as per the diagnosis.

- To develop an efficient classifier which can classify and cluster the new microarray genes correctly using intelligent techniques and optimizes the result.
- To cluster the unknown genes and optimize cluster compactness and separation simultaneously for each chromosome.

1.6 Thesis Organization

The rest of the thesis is organized as follows: In Chapter 2, all the efforts in the literature have been focused which describe the various approaches for microarray cancer classification using Backpropagation neural networks, SVM and role of MOGA and NSGA-II in clustering. In Chapter 3, feature extraction and dimension reduction technique and architecture of BPNN and SVM classifier are discussed with their experimental results. In Chapter 4 basic concept of GA, MOGA and NSGA-II is discussed. Implementation of MOGA-SVM for classification and clustering with microarray cancer data and performance measures using Silhout index S(C) and Adjusted Rand Index (ARI) has also been discussed with their simulation results. Finally Chapter 5 concludes the report and gives a platform for further research.

Chapter 2

Chapter 2

Related Work

In previous work, due to the presence of large number of genes and high complexity of biological networks, there is a great need to develop analytical methodology to analyze and to exploit the information captured by gene expression data. In the pattern layer of Backpropagation Neural Network(BPNN) model, due to the presence of redundant nodes the computational complexity of the network increases and so does the computational cost. The performance of Back-propagation training algorithm applied to a feedforward multilayer neural network and its performance depend on the activation function and error-correction rule [6]. Feature extraction of microarray genes has a greater impact on its classification and clustering as it is taken as input to any network. The use of gene expression data in discriminating two types of very similar cancers acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) presented in [7]. Classification results are reported in [2] using methods other than neural networks. Here, we explore the role of the feature vector in classification. To achieve the best performance with a learning algorithm on a particular training set, a feature subset selection method should be applied. PCA is an orthogonal transformation of the coordinate system in which the data are represented. The new transformed coordinate values by which data are represented are called principal components [8].

Principal component analysis has been applied to analyze gene expression data and to improve cluster quality are studied in [9]. The diagnosis of multiple common adult malignancies could be achieved purely by molecular classification, this is done by using Support vector machine algorithm [10]. Support Vector Ma-

chines (SVMs) are a popular machine learning method for classification, regression, and other learning tasks are presented in [11]. A class discovery procedure automatically discovered the distinction between acutemyeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) without previous knowledge of these classes. An automatically derived class predictor was able to determine the class of new leukemia cases are presented in [2]. One particular machine learning algorithm, Support Vector Machines (SVMs), has shown promise in a variety of biological classification tasks, including gene expression microarrays are presented in [10], [12]. SVM method and one of its improved version CSVM as the classifier gave a better result using gene expression data [13]. The selection of a small subset of genes out of the thousands of genes in microarray data is important for accurate classification of phenotypes are presented in [14]. Multiobjective genetic algorithms gives fast nondominated sorting approach NSGA-II. In this paper we investigate the Goldberg's notion of non dominated sorting in GA's along with niche and speciation method to find multiple pareto optimal points simultaneously [15].

K. Deb et al. presented much better spread of solutions and better convergence near the true Pareto-optimal front compared to Pareto-archived evolution strategy and strength-Pareto EA two other elitist MOEAs [28]. Ramaswamy et al. presented tumor gene expression for Multiclass cancer diagnosis [10].

From the related works it has been concluded that Feature extraction for the microarray cancer data is important for classification and clustering. To reduce features from data is important to increase the efficiency of the network, hence a principal component analysis is used for feature reduction. In various paper it has shown that SVM has A greater efficiency in performance of classification as it has various parameter to regularize. Multiobjective genetic algorithms is used to obtain non- dominated solutions .

Chapter 3

Chapter 3

Feature Extraction and Cancer Classification

3.1 Dimension Reduction

Data analysis causes problem where the data objects have a large number of features which is more prevalent in areas such as multimedia data analysis and bioinformatics. It is often beneficial to reduce the dimension of the data in order to improve the efficiency and accuracy of data analysis [16]. One of the problems with high-dimensional datasets is that, in many cases, not all the measured variables are important for understanding the underlying phenomena of interest. While certain computationally expensive novel methods can construct predictive models with high accuracy from high-dimensional data, it is still of interest in many applications to reduce the dimension of the original data prior to any modelling of the data. To deal with this issue, dimension reduction techniques are often applied as a data pre-processing step or as part of the data analysis to simplify the data model. By working with this reduced representation, tasks such as classification or clustering can often yield more accurate and readily interpretable results, while computational costs may also be significantly reduced.

The motivation for dimension reduction can be summarized as follows:

- The identification of a reduced set of features that are predictive of outcomes can be very useful from a knowledge discovery perspective.
- For many learning algorithms, the training, clustering or classification time

increases directly with large number of features.

- Noisy or irrelevant features can have the same influence on classification and clustering as predictive features so they will impact negatively on accuracy.
- Things look more similar on average the more features used to describe them. It shows that the resolution of a similarity measure can be worse in 20D than in a 5D space.

In mathematical terms, the problem can be stated as: we have n observations ,each being a realization of p-dimensional random variable X = (x_1, x_2, . , x_p) where to find a lower dimensional representation of it, s = (s_1,s_2, , s_k) with k $\leq$ p, that captures the content in the original data according to some criterion. Here X_{np} is transformed to X_{nk} . Typically this is a linear transformation W_{kp} that will transform each object $x_i$to x_1' in k dimensions with mean μ = (μ_1, . , μ_p) and covariance matrix $\sum$ = (X-μ) (X-μ)^T = $\sum$pxp such matrix can be represented by X= {x_{ij}:1≤i≤n, 1≤j≤p}.

$$X_i' = WX_i \tag{3.1}$$

3.1.1 Objectives of Feature Extraction

The objectives of feature extraction are many, the major ones are:

- To avoid over fitting and improve model performance.
- To provide faster and more cost-effective models, and
- To gain a deeper insight into the underlying processes that generated the data.

As the data sets have various attributes and due to the huge amount of data we cannot consider all the dataset. So we apply different analysis algorithm to reduce our data sets. So that only few data sets can contribute to the final result. Here we have used a dimension reduction methods i.e Principal Component Analysis (PCA) to reduce the size of data.

3.1.2 Dimension Reduction using Principal Component Analysis:

PCA was invented in 1901 by Karl Pearson. Principal component analysis (PCA) is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of uncorrelated variables called principal components [17]. Hence the central idea is to reduce the dimensionality of the data set while retaining as much as possible the variation in data set. Principal components (PC's) are linear transformation of the original set of variables. PC's are uncorrelated and ordered in such a way that the first few PC's contain most of the variables in the original data set. The number of PCs are less than or equal to the number of original variables. PCA provides an efficient way to find these components which contribute to the data variation and thus reduce the input dimensions. If the data are concentrated over a particular linear subspace, PCA provides a way to compress data and simplify the representation without losing much information. But if the data are concentrated over a non-linear subspace the PCA will fail to work well.

Principal Component Analysis (PCA), is a very powerful statistical technique, to represent the d-dimensional data in a lower-dimensional space without any significant loss of information. The aim is to project the original I-dimensional space into an I_0 dimensional linear subspace, where $I > I_0$ such that the variance in the data is maximally explained within the smaller I_0 dimensional space . PCA rigidly rotates the axes of the n-dimensional space to new positions (principal component) such that principal component 1 has the highest variance, PC 2 has the next highest variance and so on. The covariance among each pair of the principal component is zero so the PC's are uncorrelated and ordered in such a way that the first few PC's contain most of the variables in the original data set. If the data are concentrated over a particular linear subspace, PCA provides a way to compress data and simplify the representation without losing much information.

3.2 Classifiers

3.2.1 Back Propagation Neural Networks Classifier

In multilayered feedforward network, neurons are organized into layers. The input layer is composed not of full neurons, but rather consists simply of the values in a data record, that constitutes inputs to the next layer of neurons. The next layer is called a hidden layer; there may be many hidden nodes. The final layer is the output layer, where there is one node for each class. A single forward pass through the network results in the assignment of a value to each output node, and the record is assigned to whichever classifications node had the highest value. Multilayer feedforward networks are trained using the Backpropagation (BP) learning algorithm. Backpropagation training algorithm when applied to a feed-forward multilayer neural network is known as Backpropagation neural network. Functional signals flows in forward direction and error signals propagate in backward direction. That's why it is Error Backpropagation or shortly backpropagation network. The activation function that can be differentiated (such as sigmoidal activation function) is chosen for hidden and output layer computational neurons. The algorithm is based on an error - correction rule. Learning is based upon mean squared error and generalized delta rule. The rule applied for weight updation is generalized delta rule [18], [6].

The algorithm defines:

1. Initialization of weights (w) and biases (b) to random small values and target (t) is fixed.

2. Forward computation: Output of each layer is $y = \Phi\left(wx \times b\right)$. where w = synaptic weight, x = input and b = bias value. Output of input layer is the input of hidden layer. In this way actual output is calculated.

3. Error is calculated by the difference of target and the actual output at output layer of neuron. Error $e = t - y$.

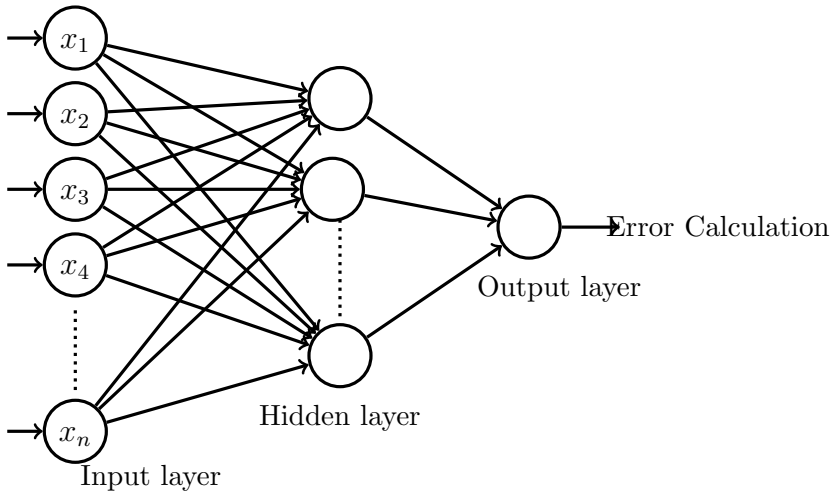


Figure 3.1: Multilayered Backpropagation Neural Network

4. Backward computation: Error at each layer is calculated by partial differentiation. For output layer error, $e_0 = 0.5 \times (d\Phi\,(hidden)\,/dy\,(hidden)) \times e$ and For hidden layer error, $e_h = (d\Phi\,(Y_{input})\,/dY_{input}) \times w_0ut \times e_0$.

5. Weights and biases in each layer are updated according to the computed errors. Updated weight, $w_{new} = w_{old} - lr \times e_{layer} \times x_{layer}$ layer . Updated bias, $b_{new} = b_{old} - lr \times e_{layer}$ layer where e_{layer} is the error of the particular layer and x_{layer} is the input that is fed to the layer and lr is the learning rate.

6. Step 2 to 5 is repeated until the acceptable minimized error.

3.2.2 BP Neural Network Classifier Hybrid with PCA Algorithm

Although back propagation is the most popular learning method in the neural network community, the drawbacks of it are often pointed out:

1. Very slow computing speed

2. The possibility of getting trapped in local minima.

3. More hidden nodes leads to overfitting and greater capacity of assimilating data.

4. The convergence obtained from backpropagation learning is very slow.

5. The convergence in backpropagation learning is not guaranteed.

3.2.3 Why SVM for cancer classification

SVMs are used for cancer classification mainly due to following two reasons:

1. SVMs have demonstrated the ability to not only correctly separate entities into appropriate classes, but also to identify instances whose established classification is not supports by the data.

2. SVM have many mathematical features that make them attractive for gene expression analysis, including their flexibility in choosing a similarity function, sparseness of solution when dealing with large data sets, the ability to handle large feature spaces, and the ability to identify outliers.

3.2.4 The SVM Classifier and Kernel Selection

A support vector machine (SVM) [19] is a computer techniques used for the supervised learning process to analyze and recognize patterns, derived from statistical learning theory developed by Vladimir N. Vapnik and Corinna Cortes in 1995. The goal of SVM is to produce a model (based on the training set) which predicts the target values of the test set making it as a non-probabilistic linear classifier. Viewing the input data as two sets of vectors in a d-dimensional space, an SVM constructs a separating hyperplane in that space, which maximizes the margin between the two classes of points. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the neighbouring data points of both classes. Larger margin or distance between these parallel hyperplanes indicates better generalization error of the classifier [19]. Implies that only support vectors matter; other training examples are ignorable.

The SVM is designed for binary-classification problems, assuming the data are linearly separable. Given the training data $(x_i, y_i), i = 1, 2....m, x_i \epsilon R^n, y_i \epsilon \{+1, -1\}^t$ where,

R^n : is the input space,

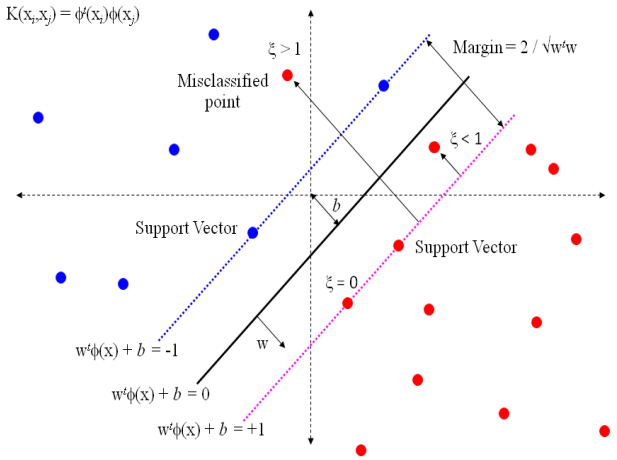


Figure 3.2: SVM Classifier

x_i : is the sample vector and

y_i : is the class label of x_i,

the separating hyperplane (w, b) is a linear discriminating function that solves the optimization problem:

$$\min_{\text{w,b}} \frac{1}{2} W \cdot W^T \qquad subject\ to Y_i(< W \cdot X_i > +b) - 1 \geq 0 \tag{3.2}$$

$i = 1, 2.....m.$

The minimal distance between the samples and the separating hyperplane, i.e. the margin, is $\frac{1}{\|W\|}$. Data points closest to the hyperplane are called Support Vectors. In order to relax the margin constraints for the nonlinearly separable data, the slack variables ξ_i are introduced into the optimization problem:

$$\min_{\text{w,b,}\xi} \frac{1}{2} W \cdot W^T + C \sum_{i=1}^{m} \xi_i \qquad subject\ to Y_i(< W \cdot X_i > +b) \geq 1 - \xi_i..... \tag{3.3}$$

$\xi_i \geq 0$ In terms of these slack variables, the problem of finding the hyperplane that provides the minimum number of training errors, i.e. to keep the constraint violation as small as possible. C > 0 is the penalty parameter of the error term. The problem of finding the weight vector w can be formulated as minimizing the

following function:

$$L(w) = \frac{1}{2}\|w\|^2 \tag{3.4}$$

subject to

$$y_i(w \cdot \phi(x_i) + b) \geq 1 \tag{3.5}$$

i= 1, 2....., n. Here, b is the bias and the function, $\phi(x)$ maps the input vector to the feature vector.

Quadratic optimization algorithms can identify which training points x_i are support vectors with non-zero Lagrangian multipliers α_i which used to solve the optimization problem.

The dual formulation is given by maximizing the following:

Find α_1, ..., α_N such that

$$Q(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j K(x_i, x_j) \tag{3.6}$$

subject to (1) $y_i \sum \alpha_i = 0$, and

(2) $0 \leq \alpha_i \leq$ C $\forall\ \alpha_i$, i=1, 2,...., n

where f(x) = $\sum \alpha_i \mathrm{Y}_i \mathrm{X}_i^T \mathrm{X}$ + b

Only a small fraction of the α_i coefficients are nonzero. The corresponding pairs of x_i entries are known as support vectors and they fully define the decision function.

SVM maps the training samples from the input space into a higher-dimensional feature space via a mapping function, called kernel function.

Furthermore, $K(X_i, X_j) = X_i^T \cdot X_j$ is called the kernel function. There are following four basic kernels: linear, Gaussian, RBF, Sigmoid kernels as below.

Linear Kernel:

$$K(x_i, x_j) = x_i^T x_j$$

Polynomial kernel:

$$K(x_i, x_j) = (1 + x_i \cdot x_j)^d$$

Radial basis function kernel:

$$K(x_i, x_j) = exp(-\gamma \left\| x_i - x_j \right\|)^2$$

Sigmoid kernel:

$$K(x_i, x_j) = tanh(kx_i \cdot x_j - \delta)$$

Now BPNN and SVM are applied to the reduced data set to find the networks accuracy.

Many computation techniques are proposed for solving the first problem, but gives very less attention to solve other problems. The number of reduced features retrieved will be determined by cumulative energy threshold value.

3.3 Proposed Work I:

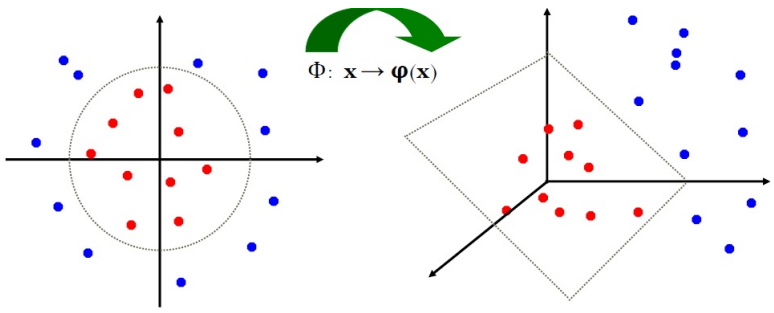
After the data set is normalizes using the following euquation, PCA is then implemented for reducing the high dimensional DNA microarray data. On the reduced data set feed forward neural network and SVM are implemented and their performance accuracies are compared.

3.3.1 Data Preprocessing and Cleaning

Filling in missing values, smoothing noisy data, identifying and removing outliers and resolving inconsistencies.

3.3.2 Data Normalization

Data normalization is followed after data preprocessing and cleaning. Data normalization is essential to the performance of classifiers. We use Z-min-max



normalization method. It transforms the data into the desired range [0, 1].

$$X_{norm} = (X_{m\times n} - min) \,/\, (max - min) \tag{3.7}$$

X_{norm} is the result of the normalization, x_{mn} is the feature(gene) to be normalized,max is upper bound of the gene expression value, and min is lower bound of the gene expression value.

SVM and BPNN often does not gives better accuracy for high dimension, to improve the efficiency, we proposed to apply Principal component analysis on the original data set, to obtain a reduced dataset containing possibly uncorrelated variables without any loss [5], [18]. Then the reduced data set will be applied to SVM and BPNN classifier to improve performance of the classifier.

Our first contribution is to prove that PCA is able to reduce dimension of features and to provide classification competitive performance than traditional classifiers in terms of speed and predictive accuracy, and precision of convergence [20].

Hybrid approach is being proposed for reduction of features and structure modeling of classifiers using PCA [16], [17]. After the implementation of PCA, two classifiers such as Feed Forward Neural Network (FFNN) trained using BP algorithm and SVM [19] are implemented. The general procedure of the algorithm explianed in the Figure 3.3:

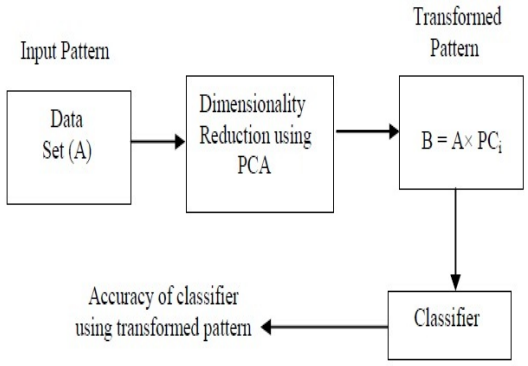


Figure 3.3: PCA-SVM or PCA-BPNN classifiers for cancer data

The brief overview of our entire proposed process is shown below in Figure 3.4:

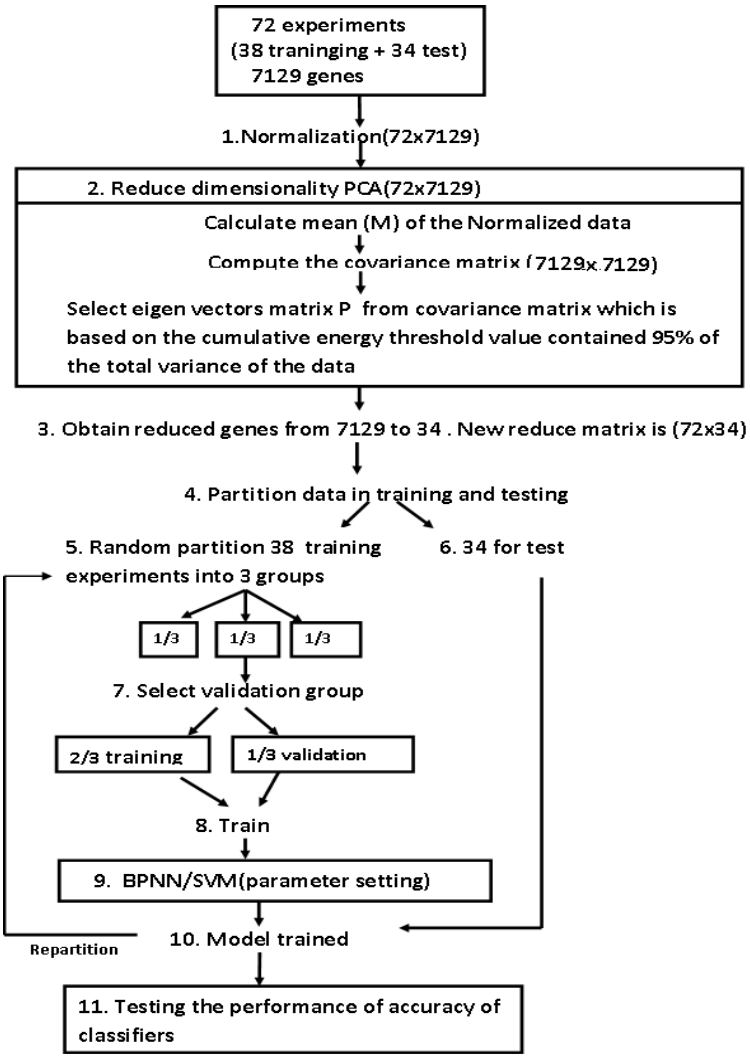


Figure 3.4: Schematic illustration of the proposed method for Leukemia cancer data

The entire data set of all 72 experiments was first Normalized (step 1) and then the dimensionality was further reduced by principal component analysis (PCA) to 34 PCA projections, (2) from the original 7129 expression values. Next, the 34 test experiments were set aside (6) and the 38 training experiments were randomly partitioned into 3 groups from reduced matrix (5). One of these groups was reserved for validation and the remaining 2 groups for traning (7). BPNN/SVM models were then trained using for each sample the 34 PCA values as input and the cancer category as output (9). The samples were again randomly partitioned and the entire training process repeated (10). The 34 test experiments

were subsequently classified using all the trained models. The entire process (5-10) was repeated.

Algorithm:

The goal of PCA is to derive another matrix P matrix which will describe a linear transformation of every column in X(every training gene) in the eigenfaces subspace, in the form: W=PX, where W are the projections of the training genes on the subspace described by the eigenfaces. The rows of P matrix represent the principal components and they are orthogonal.

The steps involved in proposed algorithm are as follows:

Data: DNA Microarray input matrix X

Result: Reduced data set

Phase-1: Apply PCA to reduce the dimension of the data set (input matrix X)

Step 1: Let's suppose we have X data matrix with m rows of samples and n columns of genes (features). $a_{ij}s$ represent the gene values.

Step 2: To reduce redundant or missing values in matrix X we apply data normalization on matrix X as follows equation:

$$X_{norm} = (X_{m\times n} - min) \,/\, (max - min) \tag{3.8}$$

Step 3: Calculate the mean M from the data set where Xi (i = 1, 2,....., m) represents the i^{th} column of X and M represent the mean of genes.

$$M = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{3.9}$$

Step 4: The genes are mean M centered by subtracting the mean value of gene from each gene vector and let K_i be defined as mean centered genes.

$$K_i = X_i - M \tag{3.10}$$

Where i=1, 2, ..., n.

Step 5: Compute the covariance matrix S_A, where A = [K1, K2... Kn]:

$$S_A = \sum_{i=1}^{n} AA^T\,(n-1) \tag{3.11}$$

Step 6: Calculate the Eigen Values $\lambda_1, \lambda_2 ....., \lambda_n$ of the Covariance matrix S_A, as sorted in a decreasing order. Let the corresponding eigen vectors be denoted as $a_1, a_2 ....., a_n$.

Step 7: Choosing components and form a feature vector by taking the eigenvectors that you want to keep from the list of eigenvectors, and forming a matrix $P[m \times a]$ with these eigenvectors in the columns, Where a is determined based on some threshold on the eigenvalues. The cumulative energy threshold value can be calculated as:

$$\frac{\sum_{i=1}^{g} x_i}{\sum_{i=1}^{p} x_i} > \theta \tag{3.12}$$

Step 8: Derive the new data set with principal components (PC's) by the following formula

Final Data ($PC's$) = Original matrix (X) x eigenvector matrix

Phase-2: Apply SVM [19] or BPNN [18]

Step 9: Partition reduced Final data in training and testing data set.

Step 10: Train BPNN and SVM model with training data and

Step 11: Test BPNN and SVM model with testing data and calculate the accuracy.

Now BPNN and SVM applied to the reduced data set to find the networks accuracy.

3.4 Implementation

The simulation process is carried on a machine having Intel(R) core (TM) 2 Duo processor 3.0 GHz and 3 GB of RAM. The MATLAB version used is R2012(a). The simulation was carried out with 3 microarray cancer data sets.

3.4.1 Data Sets

Data Set 1: Leukemia cancer

Number of Instances: 72 (consist of 2 classes for distinguishing: Acute Myeloid

Leukemia(AML) and Acute Lymphoblastic Leukemia (ALL). The complete dataset contains 25 AML and 47 ALL samples. 38 samples for training set and 34 samples for test set are chosen for simulation).

Number of Attributes: 7129

Resultant data set (after PCA): 72x34.

The data sets taken from public Kent Ridge Biomedical Data Repository with URL: http://sdmc.lit.org.sg/GEDatasets/Datasets.html. or following

URL: http://www.inf.ed.ac.uk/teaching/courses/dme/html/datasets0405.html.

Data Set 2: Ovarian cancer

Number of Instances: 216 (consist of 2 classes for distinguishing: Cancer and Normal. The complete dataset contains 121 ovarian cancer and 95 normal cancer samples. 119 samples for training set and 97 samples for test set are chosen for simulation).

Number of Attributes: 4000.

Resultant data set (after PCA): 216x28.

The data set taken from public Kent Ridge Biomedical Data Repository with url http:// sdmc.lit.org.sg/GEDatasets/Datasets.html.

Data Set 3: Colon cancer

Number of Instances: 62 (consist of 2 classes for distinguishing: tumor biopsies and normal biopsies . The samples consist of 36 tumor biopsies collected from tumors, and 27 normal biopsies collected from healthy part of the colons of the same patient.)

Number of Attributes: 2000.

Resultant data set (after PCA): 62x12.

The data sets taken from http://microarray.princeton.edu/oncology.

3.4.2 Input Parameters

We have design BPNN architecture as 72x3x1 for Leukemia cancer, 216x3x1 for Ovarian and 62x3x1 for colon cancer data set.

BPNN Paramerters: Number of nodes in hidden layer=3, learning rate=0.2, Number of iterations=1000.

SVM Parameters: C = 2, γ = 8, d = 3.

The parameters that should be optimized include penalty parameter C and the kernel function parameters such as the $\gamma(gamma)$ and d for the radial basis function (RBF) kernel. Generally d is set to be 2. Thus the kernel value is related to the Euclidean distance between the two samples. γ is related to the kernel width. Proper parameters setting can improve the SVM classification accuracy.

3.4.3 Performance Measures

The measure used to evaluate the performance of classifiers:

Accuracy = (correctly classified instances) / (Total no. of instances) *100%

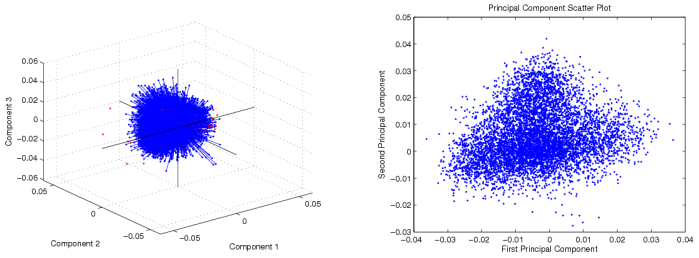
1. Accuracy =(TP+TN) / (TP+FP+TN+FN)
2. Sensitivity = (TP/TP+FN)*100%
3. Specificity = (TN/ TN+FP) * 100%
   Where, TP = true positive, TN = true negative
   FP = false positive, FN = false negative.

3.5 Numerical Simulation, Results and Discussion

Initially simulation was carried out considering the original features and BPNN and SVM classifiers. This classification aaproach is validated by considering three other data sets i.e. Leukemia cancer, Ovarian cancer and colon cancer data. The accuracy obtained with traditional BPNN and SVM were 91% and 93.1% taking Leukemia cancer and 87.1% and 96.2% taking Ovarian cancer and 56.7% and 90.03% taking Colon cancer data respectively showing in Table 3.2 and Table 3.3.

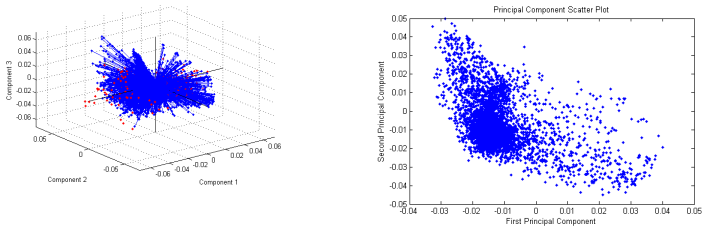
After the implementation of PCA, the data distribution across the first three principal components (PC's) and first two principal components (PC's) are shown

below in Figure 3.5 for Leukemia cancer data set, Figure 3.6 for Ovarian cancer data set. The classification accuracy varying with number of principal components (PC's) showing in Table 3.1. The data distribution across the first two features showing in Figure 3.8. The Accuracy Vs. graphs were plotted for the principal component which has shown the maximum accuracy showing in Figure 3.7.
The accuracy obtained with traditional BPNN and SVM were showing in Table 3.3.



(a) 3D representation (b) 2D representation

Figure 3.5: 3D and 2D Schematic representation of data across first three PC's and two PC's (Leukemia Cancer data set)



(a) 3D representation (b) 2D representation

Figure 3.6: 3D and 2D Schematic representation of data across first three PC's and two PC's (Ovarian Cancer data set)

Using PCA-based approach, the original number of features in Leukemia cancer got reduced from 7129 to 34 Latents (PC's) (i.e. reduced by 99.03%). It covers 95% of the total variance of the data. Therefore, there is hardly any loss of information along a dimension reduction. If the first 34 PC's are chosen, it gives best classification results. In Ovarian cancer from 4000 to 28 (i.e. reduced by 82%) and Colon cancer from 2000 to 12 Latents (i.e. reduced by 86.05%). Considering the reduced features, the accuracy obtained with PCA-BPNN and PCA-SVM were

Table 3.1: Accuracy vs. No. of PC's using PCA-SVM (Leukemia Cancer data set)

No of PC's	Accuracy (%)
10	86.03
20	89.04
30	98.03
40	98.08
50	97.12
60	97.23
70	88.23
80	90.03
90	94.08
100	98.04

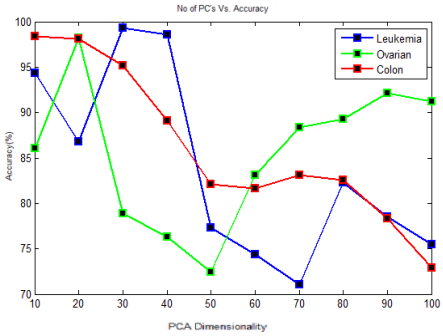


Figure 3.7: Plot showing Accuracy vs. No. of PC's using PCA

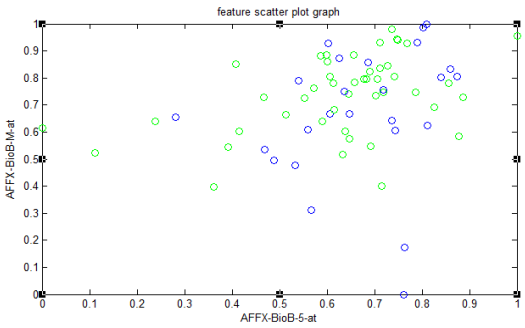


Figure 3.8: 2D Schematic representation of data across first two features (Leukemia data set)

97.3% and 98.08% for leukemia cancer and 96.2% and 98.09% for Ovarian data set and 95.02% and 97.04% for Colon cancer data set respectively.

Table 3.2: Classification Results: SVM Kernels

Data Set	Classifiers	Time (in sec)	Sensitivity (%)	Specificity (%)	Accuracy (%)
Leukemia (ALL vs. AML)	Linear	0.1548	100	93.33	96.08
	Polynomial	0.0696	100	83.33	87.08
	RBF	0.1548	100	93.3	98.08
	Sigmoid	0.0580	58.9	76.2	58.82
Ovarian (Cancer Vs. Normal)	Linear	0.1976	98.3	100	84.02
	Polynomial	0.1793	98.3	100	98.04
	RBF	0.0976	80	64.1	74.02
	Sigmoid	0.2818	34.4	76.9	59
Colon (Tumor biopsies Vs. Normal biopsies)	Linear	0.0956	98.3	100	84.02
	Polynomial	0.0451	97.03	98	99.02
	RBF	0.1146	85.2	94.4	84.8
	Sigmoid	0.2318	34.4	66.9	69

Table 3.3: Classification Results: Traditional BP, SVM, PCA-BP, and PCA-SVM

Data Set	Classifiers	Time (in sec)	Sensitivity (%)	Specificity (%)	Accuracy (%)
Leukemia (ALL vs. AML)	BP	6.17	97	86	91
	SVM	0.23	93	67.3	93.1
	PCA-BP	23.74	96	97	97.3
	PCA-SVM	0.1548	100	93.3	98.08
Ovarian (Cancer Vs. Normal)	BP	20.02	98	88.2	87.1
	SVM	9.45	68	81	96.2
	PCA-BP	20.02	98	98.2	96.2
	PCA-SVM	0.0976	98.3	100	98.09
Colon (Tumor biopsies Vs. Normal biopsies)	BP	20.02	48	58.2	56.7
	SVM	9.45	88	81	90.03
	PCA-BP	20.02	92.2	88.2	95.02
	PCA-SVM	0.0451	97.3	98	97.04

3.6 Conclusion

PCA-BP learning algorithm is designed to reduce network error between the actual output and the desired output of the network in a gradient descent manner. Experimental results illustrate PCA-SVM method showing better results than PCA-BPNN, traditional BPNN and SVM, in terms of speed, accuracy and complexity. The two stage approach of classification has shown promising results as they have outperformed traditional approaches. In this work the problem of cancer classification is solved successfully using PCA-SVM. If the data are concentrated over a particular linear subspace, PCA provides a way to compress data and simplify the representation without losing much information. But if the data are concentrated over a non-linear subspace, PCA fails to work well. Work can be further extended by implementing singular value decomposition or independent component analysis for dimension reduction. Accuracy can be checked by considering some more number of objectives (such as discarded features, weight value association with accuracy etc.) which can be efficiently solved using Genetic algorithm(GA) [21] and MultiObjective Genetic algorithm (MOGA) [15].

Chapter 4

Chapter 4

Multiobjective Genetic Algorithm-Based Fuzzy Clustering combining with Support Vector Machine for Clustering and Classification

We propose a novel method for selecting the final clustering solution from the set of Pareto-optimal solution based on majority voting among the Pareto front solutions. Non-dominated Sorting Genetic Algorithm-II (NSGA-II) based multiobjective clustering algorithm has been adopted that optimizes the cluster compactness and cluster separation simultaneously. A challenging issue in MOO is obtaining a final solution from the set of Pareto-optimal solutions. It combines the multiobjective clustering technique with support vector machine (SVM) based classifier to obtain the good performance of classifier in terms of accuracy, specificity and sensitivity for classification and in terms of Silhoutte Index and ARI Index for clustering [19].

The main contributions of this article are embodied in the following five aspects:

1. This article presents two fitness functions (fuzzy compactness and fuzzy separation) for an individual chromosome, being optimized simultaneously.

2. This approach identifies the solution i.e. the individual chromosome which gives the optimal value of the compactness and separation.

3. The multi-objective technique is first used to produce a set of non-dominated solution. The non-dominated set is then used to find some high confidence points using a fuzzy voting technique.

4. Points which have higher degree are taken as a training data in SVM classifier and remaining points as testing data.

5. An experiment is designed and is being applied this approach to three microarray cancer data set. The experimental results confirm that MOGA-SVM approach gives more effective result for classification and clustering.

6. Performance of MOGA-SVM compared with other classifiers and methods in terms of accuracy, sensitivity, specificity, Silhoutte index and ARI index.

4.1 Evolutionary Algorithms

Two most desirable features of an Evolutionary Algorithm:

- Convergence to Pareto optimal front- To improve the convergence on the Pareto fronts Multiobjective Evolutionary Algorithm (MOEA) uses non-dominated sorting algorithms.

- Maintenance of Diversity (Representation of the entire Pareto optimal front)- Increase the diversity of solutions.

4.1.1 Brief Overview of GA

Genetic algorithms (GAs) [21] [22] are popular search and optimization strategies guided by the principle of Darwinian evolution. Although genetic algorithms have been previously used in data clustering problems [23] [24], as earlier, most of them use a single objective to be optimized, which is hardly equally applicable to all kinds of datasets.

4.1.2 Single Objective Optimization Problem (SOOP)

An optimization problem that involves optimization of single objective is known as Single Objective Optimization Problem.

In general a single objective function can be defined as minimizing or maximizing a function f(x) subject to inequality constraints $g(x) \geq 0$, for all i = 1, 2.....,m and equality constraints h(x) = 0, for all j = 1, 2.... p, $x \epsilon \Omega$. So, the solution minimizes or maximizes the function f(x), where x is a n-dimensional decision vector variable $x = (x_1, x_2.....x_n)$ from some universe Ω. The inequality and equality constraints must be fulfilled while optimizing (minimizing or maximizing) the objective function f(x). In SOOP, only a single optimal solution is obtained. And either the maximum or the minimum fitness value is selected as the optimal (best) solution depending upon the problem.

4.1.3 Multiobjective Optimization

Multiobjective Optimization (MOO) is a very powerful technique, used to find solutions to many real-world search and optimization problems. Many of these problems have multiple objectives, which lead to the need of obtaining a set of optimal solutions, known as effective solutions. It has been found that Multiobjective Optimization algorithm is a highly effective one for finding multiple effective solutions in a single simulation run. It means that can be optimized simultaneously.

The multiobjective optimization criterion [25] formally, can be stated as follows: Find the vector $\overline{x}^* = [x_1^*, x_2^*....., x_n^*]^T$ of the decision variables that will satisfy the 'm' inequality constraints as,

$$g_i(\overline{x}) \geq 0, i = 1, 2, ..., m \tag{4.1}$$

and the p equality constraints as,

$$h_i(\overline{x}) = 0, i = 1, 2, ..., p \tag{4.2}$$

and optimizes the vector function

$$\overline{f}(\overline{x}) = [f_1(\overline{x}), f_2(\overline{x}).....f_k(\overline{x})]^T \tag{4.3}$$

The constraints given in (4.1) and (4.2) define the feasible region F which contains all the admissible solutions. Any solution outside this region is inadmissible since

it violates one or more constraints. The vector $\overline{x}^*$ denotes an optimal solution in F. In the context of multiobjective optimization, the difficulty lies in the definition of optimality, since we find a situation where a single vector $\overline{x}^*$ represents the optimum solution with respect to all the objective functions.

In a precise manner, MOPs are those problems where the goal is to optimize k objective functions simultaneously. The set of k objective functions can be either all maximize or all minimize or combination of both. The objective functions can be linear or non-linear and continuous or discrete in nature. There are a number of popular multiobjective optimization techniques. Among them, the GA based techniques such as NSGA-II, and SPEA2 [26] are very popular.

4.1.4 Brief Overview of MOGA

Multiobjecyive optimization is different from single objective optimization. In single objective optimization one attempt to obtain the best design or decision. Which is usually global maximun or global minimum depending on the optimization problem is that of minimization or maximization. In the case of multiple objectives there may not exist one solution which is best (global maximum or global minimum) with respect to all objectives. In a typical multiobjective optimization problem, there exists a set of solution which are superior to the rest of solutions in the search space when all objectives are considered but are inferior to other solutions in the space in one or more objectives. These solutions are known as pareto optimal solutions or non-dominated solutions [27]. To solve many real-world problems, it is necessary to optimize more than one objective simultaneously. MOGA has been introduced to optimize multiobjective problems [15]. Recently, a lot of multiobjective GAs (MOGAs) have been used by researchers for microarray cancer data. Basically, MOGA is characterized by its fitness assignment and diversity maintenance strategy. Multiobjective genetic algorithms (MOGAs) are used in this regard in order to determine the appropriate cluster centers (modes) and the corresponding partition matrix. Non-dominated sorting GA-II (NSGA-II), which is a popular elitist MOGA, is used as the underlying optimization method. The two objective functions, i.e., the global fuzzy compact-

ness of the clusters and fuzzy separation, are optimized simultaneously. Unlike single objective optimization, which yields a single best solution, in MOO the final solution set contains a number of Pareto-optimal solutions, none of which can be dominated or further improved on any one objective without degrading another.

The single objective formulation is extended to reflect the nature of multi-objective optimization problem where there is more than one objectives function which needs to be optimizing [25]. Thus there is set of solutions instead of a single solution i.e. a set of optimal solution and they are found using Pareto-optimality theory. The set of solutions obtained is based on dominance and non-dominance.

Issues in MOGA

- Fitness assignment- In each generation the non-dominated set is maintained, fitness is adjusted according to the domination of each individual.
- Density estimation- In order to encourage diversity in the population fitness is reduced for similar solutions.

4.1.5 Fast Non-Dominated Sorting

Generally non-dominated sorting is one of the main time consuming parts of multiobjective evolutionary algorithm (MOEA). So, design of fast non-dominated sorting algorithm is very necessary to improve the performance of a MOEA.

In fast non-dominated sorting approach, the population is sorted based on non-domination. After initializing the population, it is sorted based on non-domination in each front. The first front being completely dominant in the current population, the individuals in the second front is only dominated by the individuals of first front and the front goes on. The individuals are assigned rank (fitness) values or based on front to which they belong. Individuals of first front are assigned rank 1 and individuals in second front are assigned a value of 2 and so on. In addition to rank also a second parameter called crowding distance is calculated for every individual. Crowding distance measures how close an individual is to its neighbours. Large crowding distance will maintain a better diversity in the population. NSGA - II has been designed in such a way that the time complexity

is small, hence the non-domination process is fast.

For population size of P and number of objective function O, fast non-dominated can defined as follow
For each individual p, two entities are calculated

- Domination Count, n_p the number of individuals (solutions) which dominates the individual p, and
- S_p, a set of solutions which the individual p dominates.

All solutions in the first non-domination front will have $n_p = 0$. Then for every individual q in S_p, reduce the domination count by one and in doing so, if for any individual the domination count becomes zero then we put it into separate list Q, and the second front is identified. The process is continued until all fronts are identified. The total complexity of the fast non-domination procedure is OP^2, whereas the complexity of normal non-domination sorting is OP^3.

NSGA

The Non-dominated sorting Genetic Algorithm is a popular non-domination based genetic algorithm for multiobjective optimization. Actually NSGA is an extension of Genetic Algorithm for solving multiple objective function optimizations.

Drawbacks of NSGA:

- Computational complexity,
- Lack of elitism and
- Choosing the optimal parameter value for sharing parameter σ_{share}.

Pareto Terminology

The concept of Pareto optimality comes handy in the domain of multiobjective optimization.

A formal definition of Pareto optimality from the viewpoint of minimization problem may be given as follows: A decision vector $\overline{x}^*$ is called Pareto optimal if

and only if there is no $\overline{x}$ that dominates $\overline{x}^*$, i.e., there is no $\overline{x}$ such that

$\forall_i \in \{1, 2, ..., k\}$, $f_i(\overline{x}) \leq f_i(\overline{x}^*)$ and

$\exists_i \in \{1, 2, ..., k\}$, $f_i(\overline{x}) \leq f_i(\overline{x}^*)$

Where, $\overline{x}^*$ is Pareto optimal if there exists no feasible vector $\overline{x}$ that causes a reduction of some condition without a simultaneous increase in at least another. In general, Pareto optimum usually admits a set of solutions called non-dominated solutions.

1. **Dominance-** A solution is said to dominate other if it is better in all objectives than the other solution. Mathematically, Solution vector $x = (x_1, x_2, ..., x_k)$ is said to dominate solution vector $y = (y_1, y_2, ..., yk)$ if and only if xi dominates yi for all i = 1, 2,...., k.

2. **Non-dominance-** A solution is said to be non-dominated if it is better than the other solutions in atleast one objective. When Pareto points are plotted in objective space, the non-dominated solutions generates the pareto fronts. The set of all non-dominated solutions is called the "Pareto front" or pareto otimal solutions.

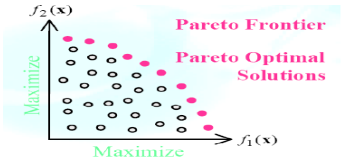


Figure 4.1: Schematic representation of Pareto-optimal solutions

4.1.6 NSGA-II

A modified and updated version of NSGA is called NSGA - II was developed, it has better sorting, incorporates elitism and the sharing parameter need not to be chosen a priori. The elitism feature favours the elites of a population i.e. the non-dominated solution among the parent and child populations are directly propagated to the next generation. In this way a good solution found early will never be lost unless a better solution is discovered. The near-Pareto-Optimal

string of the last generation provides different solutions to the clustering problem. Main goals of Nondominated sorting genetic algorithm (NSGA-II) are as follows:

- High computational complexity of nondominated sorting- The currently-used nondominated sorting algorithm has a computational complexity.
- Lack of elitism- Show that elitism can speed up the performance of the GA significantly, which also can help preventing the loss of good solutions once they are found.

The most characteristic part of NSGA-II is its elitism operation, where the parent and child populations are combined and the non-dominated solutions from the combined population are propagated to the next generation. The near-Pareto optimal strings of the last generation provide the different solutions to the clustering problem [28].

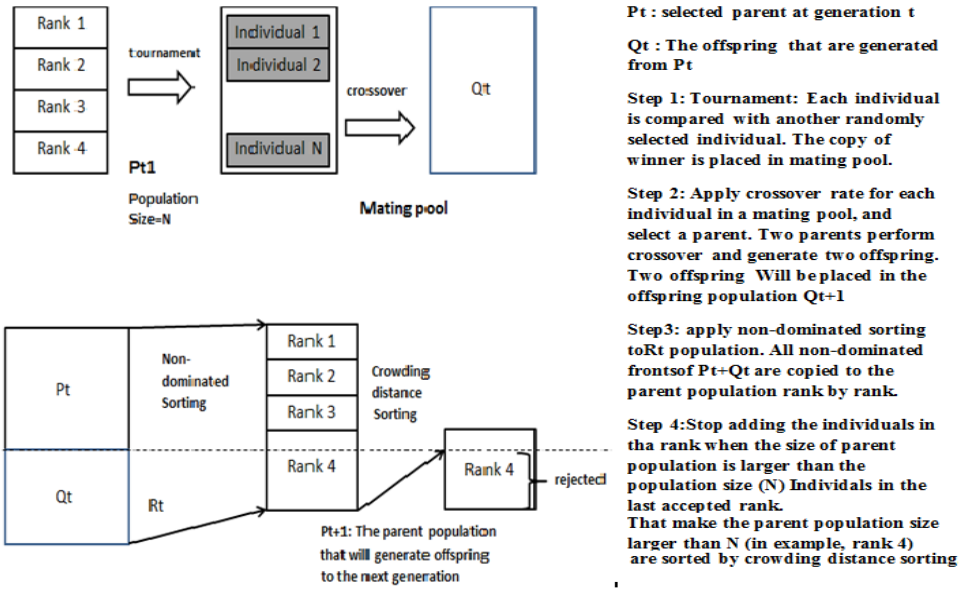


Figure 4.2: Process of NSGA-II

Fitness Assignment Ranking Based on Non-Domination Sorting

Each individual of the population is assigned a rank (fitness) value based on the non-domination sorting procedure. After calculating the rank, for the individuals of same front crowding distance is also calculated.

Diversity Mechanism

The non-domination sorting algorithm converge the solution to the Pareto optimal front. But along with the convergence one more desirable feature of MOGA needs to be maintain, the diversity of the front i.e. a good spread of the solutions along the Pareto optimal front. The original NSGA uses a well-known sharing parameter which sets the desired extent of diversity. But this method makes the computation complex and also increased the dependence of the method on value of sharing parameter chosen. But In NSGA - II, the use of crowded comparison approach eliminated the above difficulties to some extent.

Density Estimation - Crowding Distance Assignment

The basic idea behind the crowding distance is finding the Euclidean distance between individual in a front based on their objectives in the m dimensional hyper space. The individuals in the boundary are always selected since they have infinite distance assignment. Crowding distance approaches aim to obtain a uniform spread of solutions along the best-known Pareto front without using a fitness sharing parameter.

Crowded Operator based sorting

Crowded comparison operator (I) is used to guide the process of selection at the various stages of the algorithm toward a uniformly spread-out Pareto optimal front. Assume that every individual i in the population has two attributes:

- Non-domination rank (i_rank)
- Crowding Distance ($i_distance$)

Now, between two individuals i and j , the individual with lower rank will be selected(i.e. $i_rank < jrank$) or if both individual belongs to the same front then their crowding distance is compared, and individual with greater crowding distance i.e. an individual located in a lesser crowded region is selected.

Crowding Distance Assignment (I)

l=I

for each i, set $I(i)_{distance} = 0$

for each objective m

I=sort(I,m)

$I(1)_{distance} = I(l)_{distance} = \infty$

for i=2 to (l-1)

$I(i)_{distance} = I(i)_{distance} + \frac{(I(i+1)m - I(i-1)m)}{(f_m^{max} - f_m^{min})}$

Elitism

The most characteristic part of NSGA - II is its elitism operation, where the non-dominated solutions among the parent and the child populations are propagated to the next generation.

4.2 Proposed Work II: MOGA-SVM

We use Z-min-max normalization method shown in equation (3.7). It transforms the data to scale with in the range of [0, 1]. For data reduction Principal Component Analysis (PCA) is used [17]. From the large genes, the 100 genes with the largest variation across samples are selected. In this study, the multiobjective clustering technique uses NSGA-II, a popular multiobjective genetic algorithm, as the underlying multiobjective framework. Coordinates of the cluster-centers are encoded in the chromosomes of the genetic algorithm and two objective functions are developed so as to simultaneously optimize cluster compactness 'π' and cluster separation 'Sep' as shown in equation (4.7) and (4.9) respectively.

1. **Chromosome Representation:** The number of genes in the chromosome represents the sequence of attributes or feature values representing the K cluster modes. If each object has p features $\{f_1, f_2, ..., f_p\}$ the length of a chromosome will be $k \times p$, where the first p positions (or genes) represent the p-dimensions of the first cluster mode, the next p positions represent that of the second cluster mode, and so on.
   Every gene of the chromosome is real coded as shown in Figure 4.3.

2. **Population initialization:** The initial K cluster modes encoded in each chromosome are randomly chosen as K random objects of the cancer dataset.

Figure 4.3: Schematic representation of chromosome

This process is repeated for each of the P chromosomes in the population, where P is the population size.

3. **Computation of fitness function:** In this paper, the global compactness 'π' of the clusters and the fuzzy separation 'Sep' are considered as the two fitness functions, which need to be optimized simultaneously [29].
For computing the measures, the modes encoded in a chromosome are first extracted. Let these be denoted as $S = [s_1, s_2, ..., s_K]$. The membership values u_{ik},i = 1, 2 ,..., K and k = 1, 2,..., n are computed as follows [30]:

$$u_{ik} = \frac{1}{\sum_{j=1}^{k} \left(\frac{D(s_i,x_k)}{D(s_j,x_k)} \right)^{\frac{1}{g-1}}}, \quad for \;\; 1 \leq i \leq K, 1 \leq k \leq n \tag{4.4}$$

where $D(s_i, x_k)$ is the distance between point x_k and cluster s_i. If $D(s_j, x_k)$ is equal to zero for some value of j, then u_{ik} is set to zero for all i=1, 2,....K, $i \neq j$, while u_{ik} is set equal to one. 'g' is the weighting coefficient. The variation 'σ_i' and fuzzy cardinality 'n_i' of the 'i^{th}' cluster i =1, 2,..., K are calculated using the following equations [29]:

$$\sigma_i = \sum_{k=1}^{n} u_{ik}^m D(s_i, x_k), 1 \leq i \leq K \tag{4.5}$$

and

$$n_i = \sum_{k=1}^{n} u_{ik}, 1 \leq i \leq K \tag{4.6}$$

The global compactness 'π' of the solution represented by the chromosome is then computed as [29]:

$$\pi = \sum_{i=1}^{K} \frac{\sigma_i}{n_i} = \sum_{i=1}^{K} \frac{\sum_{k=1}^{n} u_{ik}^m D(s_i, x_k)}{\sum_{k=1}^{n} u_{ik}} \tag{4.7}$$

To compute other fitness function i.e., fuzzy separation 'Sep', the mode s_i of the i^{th} cluster is assumed to be the center of a fuzzy set $\{s_j | 1 \leq j \leq k, j \neq i\}$. Hence, the membership degree of each s_j to s_i, $j \neq i$ is computed as [29]:

$$\mu_{ij} = \frac{1}{\sum_{l=1,l\neq j}^{K} \left(\frac{D(s_i,x_k)}{D(s_j,x_k)}\right)^{\frac{1}{g-1}}}, i \neq j \tag{4.8}$$

Subsequently, the fuzzy separation is defined as [29]:

$$sep = \sum_{i=1}^{k} \sum_{j=1,j\neq i}^{k} \mu_{ij}^{g} D\left(s_i, s_j\right) \tag{4.9}$$

Objective is to minimize the fuzzy compactness (π) and maximize the fuzzy separation (Sep). But in this paper, the multi-objective problems are minimized i.e., the objective function π and $\frac{1}{sep}$ are minimized simultaneously.

4. **Selection:** The selection operation used here is the crowded binary tournament selection, used in NSGA-II to improve both quality and diversity of Pareto solutions. After selection, the selected chromosomes are put in the mating pool.

5. **Crossover:** Real coded GAs use Simulated Binary Crossover (SBX) [31] [32] operator for crossover.

   Simulated Binary Crossover: The crossover operation used here is the Simulated Binary Crossover depending on crossover probability μ_c. It is used to generate the new offspring solutions from the chromosomes selected in the mating pool in every iteration. Simulated Binary Crossover simulates the binary crossover observed in nature and is given as below

$$c_{1,k} = \frac{1}{2}\left[\left(1-\beta_k\right) p_{1,k} + \left(1+\beta\right) p_{2,k}\right] \tag{4.10}$$

$$c_{2,k} = \frac{1}{2}\left[\left(1-\beta_k\right) p_{1,k} + \left(1+\beta\right) p_{2,k}\right] \tag{4.11}$$

where $c_{i,k}$ is the i^{th} child with k^{th} component, $p_{i,k}$ is the selected parent and $\beta_k\,(\leq 0)$ is a sample from a random number generated having the density.

$$p\left(\beta\right) = \frac{1}{2}\left(\mu_c + 1\right)\beta^{\mu_c}, if 0 \leq \beta \leq 1 \tag{4.12}$$

$$p(\beta) = \frac{1}{2}(\mu_c + 1)\frac{1}{\beta^{\mu_c}}, if \beta > 1 \tag{4.13}$$

This distribution can be obtained from a uniformly sampled random number u between $(0, 1)^{\mu_c}$ is the distribution index for crossover. That is

$$\beta(u) = (2u)^{\frac{1}{\mu_c+1}} \tag{4.14}$$

$$\beta(u) = \frac{1}{(2(1-u))^{(}\mu_c + 1)} \tag{4.15}$$

6. **Polynomial Mutation:** For performing the mutation [32], a mutation probability μ_m has been used to create a offspring population N. If a chromosome is selected to be mutated, the gene position that will undergo mutation is selected randomly. After that, the gene value of that position is replaced by another random value chosen from the corresponding gene domain. Then elitism operation to choose a particular solution has been applied among the set of non-dominated solutions N based on the best fitness value.

$$c_k = p_k + \left(p_k^u - p_k^l\right)\delta_k \tag{4.16}$$

where c_k is the child and p_k is the parent with p_k^u being the upper bound on the parent component, p_k^l is the lower bound and δ_k is small variation which is calculated from a polynomial distribution by using

$$\delta_k = (2r_k)^{\frac{1}{\mu_m+1}} - 1, if r_k < 0.5 \tag{4.17}$$

$$\delta_k = 1 - (2 - (1 - r_k))^{\frac{1}{\mu_m+1}}, if r_k > 0.5 \tag{4.18}$$

where, r_k is an uniformly sampled random number between (0, 1) and μ_m is mutation distribution index.

7. **Computation of fuzzy membership matrix U_{ik}:** For each of the non-dominated solutions $y_i, 1 \leq i \leq N$, u_{ik} may be computed as per equation (4.4). This matrix is reorganized to make them consistent with each other i.e., cluster j in the first solution should be equivalent to cluster j in all other solutions. For example, the solution string (x, y, z), (m, n, p) is equivalent to (m, n, p), (x, y, z).

8. **Fuzzy majority voting technique:** The motivation is that the points that are assigned to a cluster with high membership degree by most of the non-dominated solutions can be considered as they are clustered properly. Points having maximum membership degree (to some cluster j) have been selected. These points are referred to as training points. They also should be greater than membership threshold $\alpha(0 < \alpha < 1)$ and fuzzy majority voting threshold value $\beta(0 < \beta < 1)$, for at least N solutions.

9. **SVM classifier**: For each points

   - Selected points that can be used to train the classifier.
   - Four different Kernel functions are used for training such as Linear, Polynomial, Sigmoidal and Radial Basis Function.
   - The remaining low-confidence points (test points) can be thereafter classified using four trained SVM classifiers.
   - The label vectors of the training and test points are combined to yield the label vector λ of the complete dataset for each classifier.
   - Combine the four clustering label vectors through majority voting ensemble, i.e., each point is assigned a class label that obtains the maximum number of votes among the four clustering solutions. Ties are broken randomly.

4.3 IMPLEMENTATION

The simulation process is carried on a machine having Intel(R) core (TM) 2 Duo processor 3.0 GHz and 3 GB of RAM. The MATLAB version used is R2012(a). The simulation was carried out with 3 data sets. First 100 genes with largest variation across samples are selected out of large genes using PCA.

4.3.1 Data Sets

Data Set 1: Leukemia cancer

Number of Instances: 72 (consist of 2 classes for distinguishing: Acute Myeloid

Leukemia(AML) and Acute Lymphoblastic Leukemia (ALL). The complete dataset contains 25 AML and 47 ALL samples.)

Number of Attributes: 7129

Resultant data set (after PCA): 72x100.

The data sets taken from public Kent Ridge Biomedical Data Repository with URL: http://sdmc.lit.org.sg/GEDatasets/Datasets.html. or following

URL: http://www.inf.ed.ac.uk/teaching/courses/dme/html/datasets0405.html.

Data Set 2: Ovarian cancer

Number of Instances: 216 (consist of 2 classes for distinguishing: Cancer and Normal. The complete dataset contains 121 ovarian cancer and 95 normal cancer samples.)

Number of Attributes: 4000.

Resultant data set (after PCA): 216x100.

The data set taken from public Kent Ridge Biomedical Data Repository with url http:// sdmc.lit.org.sg/GEDatasets/Datasets.html.

Data Set 3: Colon cancer

Number of Instances: 62 (consist of 2 classes for distinguishing: tumor biopsies and normal biopsies . The samples consist of 36 tumor biopsies collected from tumors, and 27 normal biopsies collected from healthy part of the colons of the same patient.)

Number of Attributes: 2000.

Resultant data set (after PCA): 62x100.

The data sets taken from http://microarray.princeton.edu/oncology.

4.3.2 Parameters for MOGA-SVM

Chromosome: In our experiment every chromosome represents 200 attributes or features. The population size (N_p) for every generation is fixed at 50. Then the initial population matrix size is 50x200. MOGA-SVM scheme are shown in Table 4.1. While the stopping criteria is met (i.e., Max 1000 generations) the execution

Table 4.1: PARAMETER VALUES FOR MOGA-SVM

Parameters	Value
Population size P	50
Maximum Number of generation G	1000
Crossover probability (μ_c)	10
Mutation probabilityp(μ_m)	1/chromosome length
Fuzzy exponent g	2
Membership threshold α	0.65
Majority voting threshold β	0.65
kernel function parameters γ	8
Penalty parameter C	2
kernel function parameters d	3

of NSGA-II algorithm stops. The sizes of the training and testing sets depend on the two parameters α and β . Here, α is the membership threshold, i.e., it is the maximum membership degree above which a point can be considered as a training point. The parameter β (majority voting threshold) determines that a minimum number of non-dominated solutions agree with each other in the fuzzy voting context. If α and β are increased, the size of the training set will decrease, but it implies that more number of nondominated solutions agree with each other and confidence of the training set is high. However, if α and β are decreased, the size of the training set increases, but it indicates that less number of non-dominated solutions have agreement among themselves and the training set has less confidence. To achieve a trade off between the size and confidence of the training set, after several experiments, we have set both the parameters to a value of 0.65.

4.3.3 Performance metrics

The performance of MOGA-SVM evaluated in classification and clustering techniques results.

Classification

The measure used to evaluate the performance of MOGA-SVM classifiers are accuracy, sensitivity and specificity:

Clustering

For evaluating the performance of the clustering algorithms on the three cancer data sets, an external validity measure namely Silhouette Index (S(C)) [33] and

an internal validity measure namely Adjusted Rand Index (ARI) [34] are used.

Silhouette index: Silhouette index is a cluster validity index that is used to judge the quality of any clustering solution C. The silhouette can be used to: (i) select the number of clusters and (ii) assess how well individual observations (samples) are clustered. Suppose 'a' represents the average distance of a point from the other points of the cluster to which the point is assigned, and 'b' represents the minimum of the average distances of the point from the points of the other clusters. Therfore, the silhouette width s of the point is defined as:

$$s = \frac{b - a}{max\,(a, b)} \tag{4.19}$$

For a given number of clusters K, the overall average silhouette width for the clustering is simply the average of s over all observations n,

$$s(C) = \frac{\sum_1^n s}{n} \tag{4.20}$$

Silhouette index s(C)is the average silhouette width of all the data points (genes) and it reflects the compactness and separation of clusters. The value of silhouette index varies from -1 to 1 and higher value indicates better clustering result. Hence this index is a good indicator for selecting the number of clusters.

Adjusted Rand Index: Suppose T is the true clustering of the samples of a cancer data set based on domain knowledge and C a clustering result given by some clustering algorithm. Let a, b, c and d respectively denote the number of sample pairs belonging to the same cluster in both T and C, the number of pairs belonging to the same cluster in T but to different clusters in C, the number of pairs belonging to different clusters in T but to the same cluster in C,and the number of pairs belonging to different clusters in both T and C. ARI (T,C)is then defined as follows:

$$ARI\,(T, C) = \frac{2\,(ad - bc)}{(a + b)\,(b + d) + (a + c)\,(c + d)} \tag{4.21}$$

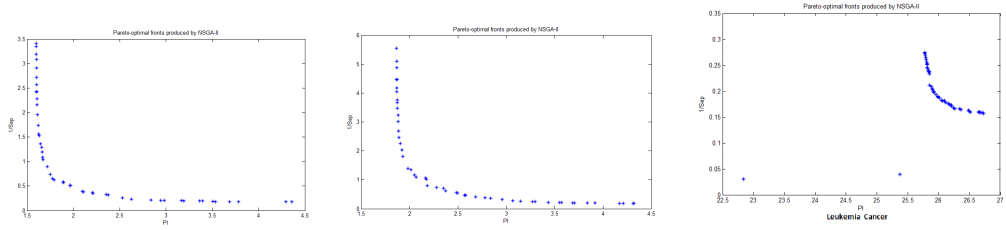
The value of ARI(T,C) lies between 0 and 1, and the higher value indicates that C is more similar to T.Also, ARI (T,T)=1.

4.3.4 Result

Initially simulation was carried out considering the computation of individual chromosome which gives the optimal value of minimum Fuzzy Compactness 'π' and maximum Fuzzy Seperation 'sep' after the implementation of NSGA-II for Ovarian, Colon and Leukemia cancer data showing in Table 4.2. Figure 4.4. shows pareto optimal fronts obtained after the implementation of NSGA-II for Ovarian, Colon and Leukemia cancer data. The Figure 4.4. plots the pareto optimal fronts produced by one of the runs of the multiobjective algorithm along with the best solutions. In Figure 4.4. the X-axis represents the first fitness (π) value and Y-axis represents the second fitness (Sep). Figure 4.4. also marks the selected solution from the non-dominated Pareto-optimal set. It appears that these selected solutions tend to fall at the knee regions of the Pareto fronts. The count of dot is less than or equal to 50, as population size equals 50.

Table 4.2: Performance of Fuzzy Compactness (π) and Fuzzy Seperation (sep)

	Data Set	Fuzzy Compactness (π)	Fuzzy Seperation(sep)	Chromosome Number
MOGA	Ovarian Cancer	3.7821	5.6179	1st
	Colon Cancer	3.7050	4.882	$2^n d$
	Leukemia Cancer	25.8552	4.29	$2^n d$



(a) Ovarian Cancer (b) Colon Cancer (c) leukemia Cancer

Figure 4.4: Schematic representation of Pareto-optimal fronts produced by MOGA-NSGA-II for cancer data

Classification Results:

The performance of MOGA-SVM kernels in terms of classification are shown in Table 4.3 and Table 4.4. RBF (Radial Basis Function) kernel with MOGA has given better result among all the SVM kernels for Ovarian cancer shown in Table

4.3. Comparision of Classification result of MOGA-SVM with MOGA-BP, SVM and BP classifiers is shown in Table 4.4. Table 4.4 shows that MOGA-SVM gives higher accuracy in minimum time; it means it classifies the cancer data in accurately. The label vectors of the training and test points are combined to obtain label vector λ for the complete data set.

Table 4.3: Classification Results: SVM Kernels with MOGA

Data Set	Classifiers	Time (in sec)	Sensitivity (%)	Specificity (%)	Accuracy (%)
Ovarian (Cancer Vs. Normal)	Linear	0.1021	94.3	100	92.08
	Polynomial	0.0976	100	96.6	99.08
	RBF	0.8261	83.6	65	76.03
	Sigmoid	0.2112	58	73	62.03
Colon (Tumor biopsies Vs. Normal biopsies)	Linear	0.0651	96.2	86.3	80.06
	Polynomial	0.0461	98.3	95.3	99.06
	RBF	0.1342	88.3	78.4	96.02
	Sigmoid	0.1621	82	59	54.08
Leukemia (ALL vs. AML)	Linear	0.2612	100	90.6	98.01
	Polynomial	0.0433	100	88.6	76.06
	RBF	0.1261	96	98.3	99.03
	Sigmoid	0.2318	35.2	67	66.02

Table 4.4: Classification Results: Traditional BP, SVM, MOGA-BP, and MOGA-SVM

Data Set	Classifiers	Time (in sec)	Sensitivity (%)	Specificity (%)	Accuracy (%)
Ovarian (Cancer Vs. Normal)	BP	6.18	86	97	87.1
	SVM	0.43	97	89.4	96.2
	MOGA-BP	1.8231	96	86.3	98.4
	MOGA-SVM	0.0976	100	96.6	99.08
Colon (Tumor biopsies Vs. Normal biopsies)	BP	10.42	48	59	56.7
	SVM	0.23	97	86.8	90.03
	MOGA-BP	12.41	86	100	84.08
	MOGA-SVM	0.0461	98.3	95.3	99.06
Leukemia (ALL vs. AML)	BP	4.12	58	86	91
	SVM	9.45	68	92	93.1
	MOGA-BP	0.1821	100	98.2	98.02
	MOGA-SVM	0.1261	98	93.3	99.03

Clustering Results:

For evaluating the performance of the clustering algorithms on the three cancer data sets, an external validity measure namely Silhouette Index (S(C)) [33] and an internal validity measure namely Adjusted Rand Index (ARI) [34] are used. Table 4.5 reports the S(C) and ARI index values for MOGA-SVM clustering algorithm. The values reported in the tables indicate that for the three cancer data sets, MOGA-SVM provides the best silhouette index (S(C)) and Adjusted Rand Index (ARI) scores. It is also evident that the results get improved with the application of SVM clustering on MOGA.

Table 4.5: Comparison of different algorithms in terms of silhouette score and ARI Index for cancer data sets

Methods	S (C)			ARI		
	Ovarian Cancer	Colon Cancer	Leukemia Cancer	Ovarian Cancer	Colon Cancer	Leukemia Cancer
MOGA-SVM	0.5676	0.4213	0.3432	0.6233	0.2411	0.2861

4.4 Conclusion

This article proposes a novel method for obtaining a final solution from the set of non-dominated solutions produced by NSGA-II based real-coded multi-objective fuzzy clustering scheme, that optimizes two fitness functions i.e., fuzzy compactness 'π' and fuzzy seperation 'sep' simultaneously. Results on microarray cancer datasets have been demonstrated and statistical superiority has been established through statistical significance test for clustering in terms of Silhouette Index and ARI Index and for classification in terms of accuracy, specificity, and sensitivity . As a scope of further research, performance of other popular classifiers combined with different MOGA technique, such as AMOSA [35] has to be tested.

Chapter 5

Chapter 5

Conclusion and Future Work

Classification and Clustering of Bioinformatics data play a vital role in detection of cause of diseases. In this report BPNN, SVM, PCA-SVM and PCA-BP techniques are implemented for clasification and BPNN, SVM, MOGA-SVM and MOGA-BP are implemented for classification and clustering both. PCA-BP learning algorithm is designed to reduce network error between the actual output and the desired output of the network in a gradient descent manner for classification. It was observed that PCA-SVM gives maximum accuracy. If the data are concentrated over a particular linear subspace, PCA provides a technique to compress data and simplify the representation without losing much information. But if the data are concentrated over a non-linear subspace, PCA fails to work well. We propose a novel method for obtaining a final solution from the set of non-dominated solutions produced by NSGA-II based real-coded multiobjective fuzzy clustering scheme, that optimizes two fitness functions i.e., fuzzy compactness 'π' and fuzzy seperation 'sep' simultaneously successfully. Results on microarray cancer datasets have been demonstrated and statistical superiority has been established through statistical significance test in terms of accuracy, specificity, sensitivity for classification and Silhouette Index and ARI Index for clustering. The experimental results show that the MOGA-SVM approach is more effective by comparing it to MOGA-BP, PCA-SVM, PCA-BP, SVM, and BP methods for clustering and classification. As a scope of further research, performance of other popular classifiers combined with different MOGA techniques, have to be tested and different parameters, various operators may be considered for higher efficiency.

Bibliography

[1] M.J. Heller. Dna microarray technology: devices, systems, and applications. *Annual review of biomedical engineering*, 4(1):129–153, 2002.

[2] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, 286(5439):531–537, 1999.

[3] IN Sarkar, PJ Planet, TE Bael, SE Stanley, M Siddall, R DeSalle, and DH Figurski. Characteristic attributes in cancer microarrays. *Journal of biomedical informatics*, 35(2):111–122, 2002.

[4] J Valente De Oliveira, Witold Pedrycz, et al. *Advances in fuzzy clustering and its applications*. Wiley Online Library, 2007.

[5] Guoqiang Peter Zhang. Neural networks for classification: a survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 30(4):451–462, 2000.

[6] Andries P Engelbrecht. *Computational intelligence: an introduction*. wiley, 2007.

[7] A. Toure and M. Basu. Application of neural network to gene expression data for cancer classification. In *Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on*, volume 1, pages 583–587. IEEE, 2001.

[8] S. Vipsita, B.K. Shee, and S.K. Rath. Protein superfamily classification using kernel principal component analysis and probabilistic neural networks. In *India Conference (INDICON), 2011 Annual IEEE*, pages 1–6. IEEE, 2011.

[9] K.Y. Yeung and W.L. Ruzzo. An empirical study on principal component analysis for clustering gene expression data. Technical report, Technical report, Department of Computer Science and Engineering, University of Washington, 2000.

[10] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C.H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J.P. Mesirov, et al. Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences*, 98(26):15149–15154, 2001.

[11] C.C. Chang and C.J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[12] Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[13] X. Zhang and H. Ke. All/aml cancer classification by gene expression data using svm and csvm approach. *GENOME INFORMATICS SERIES*, pages 237–239, 2000.

[14] C. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3(02):185–205, 2005.

[15] Kalyanmoy Deb. Multi-objective optimization. *Multi-objective optimization using evolutionary algorithms*, pages 13–46, 2001.

[16] Imola K Fodor. A survey of dimension reduction techniques, 2002.

[17] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

[18] D.E. Rumelhart, G.E. Hintont, and R.J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[19] V. Vapnik. *The nature of statistical learning theory*. springer, 1999.

[20] J. Shlens. A tutorial on principal component analysis. *Systems Neurobiology Laboratory, University of California at San Diego*, 2005.

[21] David E Goldberg. Genetic algorithms in search, optimization, and machine learning. 1989.

[22] L Davis. 1991, handbook of genetic algorithms, van nostrand reinhold, new york.

[23] Ujjwal Maulik and Sanghamitra Bandyopadhyay. Genetic algorithm-based clustering technique. *Pattern recognition*, 33(9):1455–1465, 2000.

[24] Ujjwal Maulik and Sanghamitra Bandyopadhyay. Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification. *Geoscience and Remote Sensing, IEEE Transactions on*, 41(5):1075–1081, 2003.

[25] Carlos A. Coello Coello. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information systems*, 1(3):129–156, 1999.

[26] Eckart Zitzler, Marco Laumanns, Lothar Thiele, Eckart Zitzler, Eckart Zitzler, Lothar Thiele, and Lothar Thiele. Spea2: Improving the strength pareto evolutionary algorithm, 2001.

[27] Chankong Vira and Yacov Y Haimes. *Multiobjective decision making: theory and methodology*. Number 8. North-Holland, 1983.

[28] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.

[29] George E Tsekouras, Dimitris Papageorgiou, Sotiris Kotsiantis, Christos Kalloniatis, and Panagiotis Pintelas. Fuzzy clustering of categorical attributes and its use in analyzing cultural data. *International Journal of Computational Intelligence*, 1(2):147–151, 2004.

[30] Zhexue Huang and Michael K Ng. A fuzzy k-modes algorithm for clustering categorical data. *Fuzzy Systems, IEEE Transactions on*, 7(4):446–452, 1999.

[31] Ram Bhusan Agrawal, Kalyanmoy Deb, and Ram Bhushan Agrawal. Simulated binary crossover for continuous search space. 1994.

[32] MM Raghuwanshi and OG Kakde. Survey on multiobjective evolutionary and real coded genetic algorithms. In *Proceedings of the 8th Asia Pacific symposium on intelligent and evolutionary systems*, pages 150–161, 2004.

[33] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[34] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[35] Sanghamitra Bandyopadhyay, Sriparna Saha, Ujjwal Maulik, and Kalyanmoy Deb. A simulated annealing-based multiobjective optimization algorithm: Amosa. *Evolutionary Computation, IEEE Transactions on*, 12(3):269–283, 2008.

Dissemination of Work

1. Anita Bai and Santanu Kumar Rath, "Classification Based on Support Vector Machine using Multiobjective Genetic Algorithm and Fuzzy Clustering : Application to Microarray Cancer Data ", *25th International Conference on Software Engineering and Knowledge Engineering (SEKE) Boston USA*, June 27-29, 2013. (Accepted)

2. Anita Bai and Santanu Kumar Rath, "A Novel Approach for Feature Extraction and Cancer Classification ", *9th International Symposium on Bioinformatics Research and Applications (ISBRA), USA*, May 20-22, 2013. (Accepted)

3. Anita Bai and Santanu Kumar Rath, "Multiobjective Clustering using Support Vector Machine: Application to Microarray Cancer Data ", *9th International Symposium on Bioinformatics Research and Applications (ISBRA), USA*, May 20-22, 2013. (Accepted)