

A STUDY ON PATTERN CLASSIFICATION OF BIOINFORMATICS DATASETS

Yogesh Kumar Jain



Department of Computer Science and Engineering
National Institute Of Technology
Rourkela-769008, Orissa, India

A STUDY ON PATTERN CLASSIFICATION OF BIOINFORMATICS DATASETS

Thesis Submitted

By

Yogesh Kumar Jain (108CS058)

In partial fulfillment for the award of
the degree of

Bachelors of Technology

In

Computer Science and Engineering

Under the Guidance of

Prof. S. K. Rath



Department of Computer Science and Engineering
National Institute Of Technology
Rourkela-769008, Orissa, India

CERTIFICATE



NATIONAL INSTITUTE OF TECHNOLOGY
ROURKELA

This is to certify that the thesis entitled “**A study on Pattern Classification of Bioinformatics Datasets**” is submitted by Yogesh Kumar Jain, (Roll No: 108CS058) to this Institute in partial fulfillment of the requirement for the award of the degree of Bachelors of Technology in Department of Computer Science And Engineering, is a bonafide record of the work carried out under my supervision and guidance. It is further certified that no part of this thesis is submitted for the award of any degree.

Prof. S. K. Rath

Department of Computer Science and Engineering
National Institute Of Technology, Rourkela

ACKNOWLEDGEMENT

I would like to express my hearty thanks to my guide, Prof. S. K. Rath, Department of Computer Science and Engineering, NIT Rourkela for his guidance, support, and encouragement throughout the project work. Without his knowledge, endless efforts, patience, and answers to my numerous questions, this project would have never been possible. The experimental methods and results presented in this thesis have been influenced by him in one way or the other. It has been a great privilege for me to do my project under his supervision.

I am also thankful to Ms. Swati Vipsita, Department of Computer Science and Engineering for guiding and motivating me throughout the project.

Finally, I would like to thank my parents and colleagues for their support and motivation to complete this project.

Yogesh Kumar Jain

Roll no.: 108CS058

Computer Science and Engineering

National Institute of Technology, Rourkela

ABSTRACT

Pattern Classification is a supervised technique in which the patterns are organized into groups of pattern sharing the same set of properties. Classification involves the use of techniques including applied mathematics, informatics, statistics, computer science and artificial intelligence to solve the classification problem at the attribute level and return to an output space of two or more than two classes. Probabilistic Neural Networks (PNN) is an effective neural network in the field of pattern classification. It uses training and testing data samples to build a model. However, the network becomes very complex and difficult to handle when there are large numbers of training data samples. Many other approaches like K-Nearest Neighbour (KNN) algorithms have been implemented so far to improve the performance accuracy and the convergence rate. KNN is a supervised classification scheme in which we select a subset from our whole dataset and that is used to classify the samples. Then we select a classified dataset subset and that is used to classify the training dataset. The Computation cost becomes too expensive when we have a larger dataset. Then we use genetic algorithm to design a classifier. Here we use genetic algorithm to divide the samples into different class boundaries by the help of different lines. After each generation we get the accuracy of our algorithm then we continue till we get our desired accuracy or our desired number of generation. In this project, a comparative study of Probabilistic Neural Network, K-Nearest Neighbour and Genetic Algorithm as a Classifier is done. We have tested these different algorithms using instances from lung cancer dataset, Libra Movement dataset, Parkinson dataset and Iris dataset (taken from the UCI repository and then normalized). The efficiency of the three techniques are compared on the basis of the performance accuracy on the test data, convergence time and on the implementation complexity.

Keywords: Pattern Classification, Principal Component Analysis (PCA), Probabilistic Neural Network (PNN), K-Nearest Neighbour (KNN), Genetic Algorithm (GA), Classifier and Sigma value

Table of Contents

Chapter 1	8
Introduction	8
Chapter 2	10
Literature Survey	10
CHAPTER 3	11
Basic Concepts of Classification.....	11
3.1 Data Mining	11
3.2 Pattern Classification using Soft Computing Techniques.....	11
3.3 Principal Component Analysis.....	12
CHAPTER 4	15
Techniques used for classification	15
4.1 Probabilistic Neural Network.....	15
4.2 K-Nearest Neighbour	17
4.3 Genetic Algorithm:	19
CHAPTER 5	21
Experimental Details.....	21
5.1 Datasets used:.....	21
5.2 Probabilistic Neural Network:	22
5.3 K-Nearest Neighbour:	24
5.4 Genetic Algorithm:	25
CHAPTER 6	29
Results and Discussions.....	29
6.1 Probabilistic Neural Network:	29
6.2 K-Nearest Neighbour:	31
6.3 Genetic Algorithm:	33
CHAPTER 7	35
Conclusion	35
Bibliography	36

Table of Figures

1. The score of the different components after applying PCA	13
2. A general PNN structure	16
3. The general structure followed by a GA.....	20
4. PNN Structure used along with the Gaussian formula and average function.....	23
5. 2-dimensional space with data points and lines dividing them.....	26
6. Graph of accuracy v/s principal component for different sigma values.....	29
7. Graph of accuracy v/s principal component for different sigma values.....	30
8. Graph of accuracy v/s number of training samples for KNN using lung cancer data.....	31
9. Graph of accuracy v/s number of training samples for KNN using libra movement data.....	32
10. Graph of generations v/s accuracy.....	33
11. Graph of accuracy v/s generation number with different values of H	34

Chapter 1

Introduction

Data Mining is a field in which we divide a huge dataset into smaller classes depending upon their properties. Data mining can be called the process that helps in the discovery of new correlations and patterns in a large dataset. It involves the use of artificial intelligence and many different algorithms to analyze data. Here we try to model the data in such a way that it fits into any of the algorithms. We examine the characteristics of each data sample of the dataset and try to figure out a model that is closest to the characteristics of the data sample. The overall goal is to use the present dataset to extract knowledge which can be used as a human-understandable structure in the future.

Before Data mining process begin we need to get our data preprocessed. The real world data is dirty full of noise, incomplete and inconsistent. The data to be used must be noise free and error free. Different steps involved in data preprocessing are data cleaning, data integration, data transformation, data reduction. Data mining involves various steps like classification, clustering, summarization, regression, association rule learning and anomaly detection.

First we use a PCA (Principal Component Analysis) for data reduction. Initially the data is normalized then PCA is applied to reduce then data dimension for classification.

Classification is the process in which we use some of the data to categorize the present dataset into different classes for future use. Classification is done in such a way that the properties of each data samples in a particular class are similar to each other. All approaches known use the knowledge of the already present datasets. Usually a subset of the present dataset is used as the training sample for the classification technique. Different data mining approaches can be used for classification.

Chapter 1. Introduction

Probabilistic Neural Network is a type of neural network which can classify any number of input patterns to any number of classifications. A PNN learns more quickly than any other Neural Network. PNN is as a supervised neural network that can be used in system classification and pattern recognition.

K Nearest Neighbour can be considered as statistical learning algorithms and it is extremely simple to implement and leaves itself open to a wide variety of variations. Here the training sample stores the data points which are given to it and then when the sample dataset is given to it, it finds the closest training data sample according to some distance formula(sometimes it is Euclidean distance) and gives the category of the closest data sample as the class for the sample dataset.

Genetic Algorithm is a model inspired by evolution. Here an initial population is selected which can be considered as a solution to a problem and then different process of crossover and mutation is applied which keeps most of the next population having characteristics of their previous generation and improvised. Genetic algorithm as a classifier is used to obtain the class boundaries and once the training samples are divided in different class boundaries then the sample dataset can be tested and depending upon the region it belong its class can be found.

In this project, a comparative study and of these three algorithms is done and the results are compared using different datasets. The datasets used are taken from the UCI repository. The datasets used are Lung cancer dataset, Iris dataset and Libra Movement dataset. The datasets taken were not normalized so they were first normalized (that is value from 0-1) and then were applied to the algorithms.

Chapter 2

Literature Survey

Probabilistic Neural network is a very efficient classifier. It can be used in the establishment of intelligent systems for classifying marine vessels by the acoustic radiated noise. The evaluation results of proposed method in [7] with real data show this method is successful in classifying ships to separate categories (heavy and medium ships). In [10], ship classification based on covariance of discrete wavelet using PNN is implemented. Five different types of ships were chosen to be modeled: namely, an aircraft carrier, a destroyer, a frigate, a research ship, and a merchant ship. In this paper the covariance matrix of discrete wavelet transform of ship image has been used for Ship Classification using Probabilistic neural networks. Pattern recognition applications, especially handwritten character recognition [8], are one of the most widely used applications of Probabilistic Neural Networks (PNN). PNNs application covers: classification brain tissues in multiple sclerosis, classification image textures, classification of liver tumors, EEG pattern classification, cloud classification, power disturbance recognition and classification and allocation of the load profiles to consumers etc. [9].

Genetic Algorithm is also widely used in different areas. Genetic Algorithm as a classifier is used in [1] for classifying 2-dimensional data. Further improvements were done and Genetic Algorithm as a classifier was implemented in n-dimensional data in [1].

CHAPTER 3

Basic Concepts of Classification

3.1 Data Mining

Data Mining is a field in which we divide a huge dataset into smaller classes depending upon their properties. Data mining can be called the process that helps in the discovery of new correlations and patterns in a large dataset.

Data mining techniques differ from the traditional techniques in many ways:

- **Input:** In data mining techniques we need to provide it with a preprocessed data which is free of inconsistencies, errors and incomplete.
- **Output:** In data mining we may not get an output which is a subset of the data we provide but we may even get a different output which is obtained by thorough analysis of the data samples in the data set.
- **Query:** It is the desired result we want from the data mining technique. But in data mining it may be that the data miner doesn't know what the desired output is to be.

3.2 Pattern Classification using Soft Computing Techniques

Patterns are all around us. Patterns can be found in nature, e.g., bubbles and waves, buildings, etc. Patterns can also be found in software. Each pattern describes a problem which occurs over and over again in our environment, and then it helps us to describe the solution to that problem, such that it can be used any time in future use, without having to do it again.

Patterns can be found in many datasets. So classification of the present dataset makes it easy for the classification of any new data sample. Performance of any algorithm depends on the accuracy and the convergence rate but keeping in mind the computation cost and the speed. Despite showing high results a technique may not be considered as the best because it may be taking very high computational cost and very slow in performance. Speed inversely depends on the size of the database which is given as an input. For a real time system, the size of the database is usually very large. While handling such large databases the computational cost increases and the main drawback is the data abundance. Thus to avoid it we need an efficient and effective classifier which reduces the search space and classifies effectively.

Patterns need to be normalized first in order to be applied to any algorithm to classify them. So we apply Principal Component analysis.

3.3 Principal Component Analysis

Principal Component Analysis is a method by which we can find the attributes which contribute the most to the final classification of the data set. Principal component analysis (PCA) is a mathematical procedure that converts a set of correlated variables into a set of values of uncorrelated variables called principal components. The number of original variables is always more than or equal to the principal components generated by PCA [12]. This transformation can be defined as the first principal components has the highest contribution to the final characteristics of the data sample and each succeeding component is high next and it is in no way related to the previous component. This linear transformation has been widely used in pattern classification data analysis and compression.

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.

Working of a PCA:

The different steps involved in a PCA algorithm are:

- Organize the data set
- Calculate the empirical mean and then the deviations from the mean
- Find the covariance matrix
- Find the eigenvectors and eigenvalues of the covariance matrix
- Rearrange the eigenvectors and eigenvalues
- Compute the cumulative energy content for each eigenvector
- Select a subset of the eigenvectors as basis vectors
- Convert the source data to z-scores
- Project the z-scores of the data onto the new basis

PCA is a mathematical process that does a linear transformation of the dataset to a new system such that the element which influences to the class greatest is set as the first coordinate, that is the first principal component followed by the next great and so on.

But when the data is non-linear in nature, PCA is unable to find the principal components. In that case we have to use Kernel Principal Component Analysis (K-PCA) which can work on non-linear data.

After getting the principal components (that is after the data reduction step) we can now apply our data to the algorithms.

Chapter 3. Basic Concepts of Classification

The scores of the components are after applying PCA to lung cancer data[14] is shown:

```
latent =  
  
66.9840  
11.4834  
7.7441  
5.0409  
2.6631  
2.1012  
1.5302  
0.5986  
0.4646  
0.2906  
0.2500  
0.1767  
0.1461  
0.1387  
0.0855  
0.0718  
0.0480  
0.0346  
0.0305  
0.0162  
0.0135  
0.0112
```

Figure 1. The score of the different components after applying PCA.

CHAPTER 4

Techniques used for classification

4.1 Probabilistic Neural Network

It is a supervised classifier that can map any number of input to any number of output. We give a subset of our data set to it as an input and the data is then processed through various layers. It was derived from the Bayesian network and a statistical algorithm Kernel Fisher discriminant analysis. Similar to other neural networks it also has different layers.

Probabilistic neural network (PNN) is closely related to Parzen [14] window probability distribution function estimator. A PNN consists of several smaller sub-networks; all of them are a Parzen window probability distribution function estimator for each of the classes.

The different layers of Probabilistic Neural Network are:

- **Input Layer:** It consists of the input to the neural network the training sample whose class has to be found out.
- **Pattern Layer:** It consists of the Gaussian function with the training samples are the center to it. We then find the Gaussian distance for all of them using the formula.
- **Output Layer:** Here all the outputs from the summation layers are taken and a selection of the largest value is done and that particular class of the selection determines the class of the sample.
- **Summation Layer:** Here all the output of the pattern layer is received and then they are added and send to the output layer for class determination. It is also the hidden layer in the Neural Network.

PNN can take any number of input and can map them to any number of output so for that reason PNN works more efficiently than other Neural Networks. Neural Networks in common have a back propagation which this neural network doesn't have making it faster than others.

A general structure of a probabilistic neural network is given below:

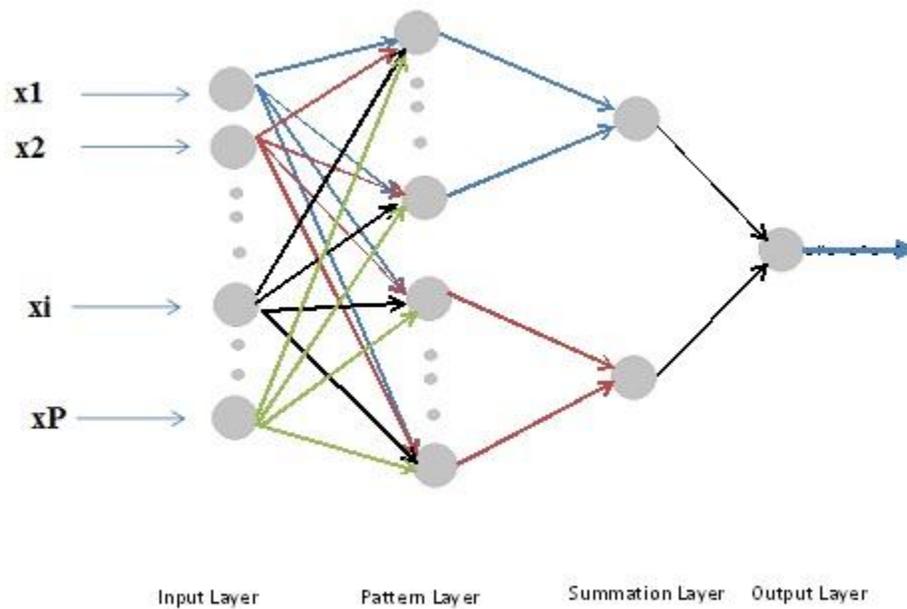


Figure 2. A general PNN structure

Some of the advantages of PNN are:

- It can map any number of input to any number of classification.
- Unlike other Neural Network it doesn't have a back propagation which makes it a fast learner.
- Once trained removing or adding a data sample doesn't require retraining of the sample.
- It saves a lot of computational space.

Some of the disadvantages of PNN are:

- It becomes highly expensive in terms of computational cost for the network when we have a very large database.
- It requires a large memory to save the network.
- For large data set the neural network slows down.

4.2 K-Nearest Neighbour

It is one of the most trivial classification algorithms which use the neighbor characteristics to determine the class of the sample data. It is also a supervised classification technique where we need to give a particular subset of the dataset to examine the rest of the data set.

KNN is a non-parametric lazy learning algorithm. By non-parametric, we mean that it does not make any assumptions on the given data distribution. This characteristic of KNN is very useful in real world because in real world most of the data doesn't obey any theoretical assumptions made (Gaussian mixtures, linearly separable, etc.).

It works on the principal where we find the Euclidean distance from all the neighboring elements and the K^{th} nearest neighbor is taken as the class for the given sample data.

Properties of KNN method:

- All the sample data represent to points in an n-Dimensional space.
- Nearest neighbor for each dataset is found using the Euclidean distance.
- The target function can be either real valued or discrete in nature.
- For discrete-valued, the k-NN returns the most common value among the k training examples nearest to the given sample.
- Voronoi diagram: the decision surface induced by 1-NN for a typical set of training examples.

Chapter 4. Techniques used for classification

For KNN with continuous data we need to find the mean of the Euclidean distances from all the training points.

After getting the classes for the sample set they are matched with the class we have and the accuracy of the KNN is found. We can vary the value of k and for different value of k we can get different results.

KNN Classification:

- Calculate distances of all training vectors to test vector
- Pick k closest vectors
- Calculate average/majority

Advantages of KNN:

- It is very simple to understand and code.
- If there is no majority then the test sample can be rejected.
- It can handle datasets which are incomplete and inconsistent.

Disadvantages of KNN:

- Network is highly sensitive to noise and irrelevant data.
- If the distance weights are not considered then the most frequently used classes dominate the classification.
- It requires a large memory to store all the training and testing data samples.
- For large data samples the KNN becomes very slow.

4.3 Genetic Algorithm:

Genetic Algorithms are inspired by the biological evolution method. They are a randomized search algorithm. They perform searching in manner similar to the natural selection in nature. They follow a single point “survival of the fitter” and try to evolve a solution to a search problem. A population of possible solution is first taken randomly and then the based on their fitness to the fitness function, it is decided whether they will undergo crossover or whether they will be rejected. As the generation passes, the members of the solution get fitter and fitter.

Genetic algorithm produces a near to the optimal solution. They also have a large amount of implicit parallelism. GA’s have application in various fields including from neural networks to VLSI design, Pattern Recognition to Image processing, etc. GA’s perform multimodal search in complex landscapes and provide near optimal solutions for objective or fitness function of an optimization problem.

In genetic algorithm the search space parameters are first encoded to a string of chromosomes. A group of chromosomes who act as an initial solution to the optimization problem are called a population.

Genetic algorithm can be used because they are more robust then the general AI techniques. They can take a large amount of input data as their input and can handle a large amount of data. Unlike other techniques Genetic Algorithm has a tendency to even adjust with a data which is noisy and have inconsistency. Genetic Algorithm although random uses the information from previous generations to improve the search.

Basic Concept of Genetic Algorithm:

- **Selection**
- **Crossover**
- **Mutation**
- **Acceptance**

Chapter 4. Techniques used for classification

This scheme is followed by the Genetic Algorithm until we get our required fitness. As soon as we get the chromosomes in the last population are treated as a solution to our optimization problem.

The general scheme followed by GA is given below.

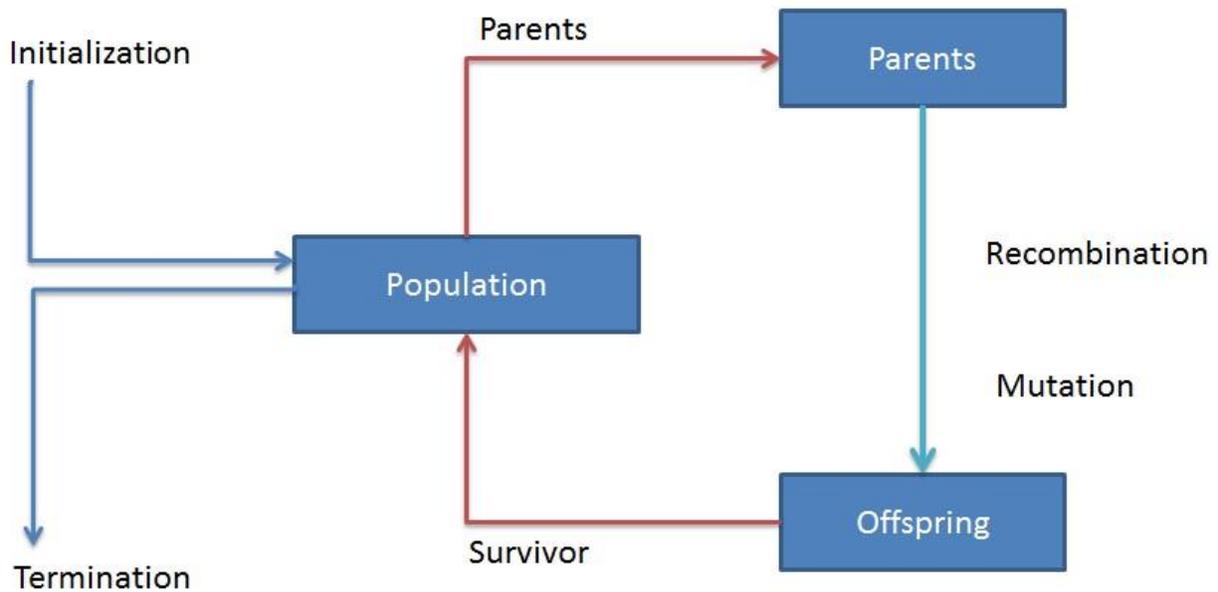


Figure 3. The general structure followed by a GA

The algorithm which is followed by the GA is:

1. Begin
2. Initialize the initial population using any random function
3. Calculate the fitness value for each element in the GA
4. Repeat step 5-8 until we get our required fitness value
5. Select the chromosomes from the population that will go under crossover
6. Crossover of the selected chromosomes is done and a new set of chromosomes are found.
The parents along with children under go mutation.
7. Mutation of the children and the parents.
8. Select the new population based on the fitness function.
9. End

CHAPTER 5

Experimental Details

5.1 Datasets used:

All the datasets were taken from the UCI repository[14]. As they were raw data, we needed to clean the data and then we had to normalise the data to be used in our algorithm.

Different datasets used are:

- Lung cancer data set
 - It has 57 Attributes and 32 data samples which are divided into three class.
 - Class I : 9 samples
 - Class II : 10 samples
 - Class III : 13 Samples
- Parkinson Data set
 - It has 23 attributes and 197 data samples.
- Iris Data set
 - It has 5 attributes and 150 data samples.

Different other types of data sets were also considered like Libra Movement dataset.

5.2 Probabilistic Neural Network:

The PNN we use also follow the same structure as the general Probabilistic Neural Network. In our network we take the input as the test dataset which is obtained as the output after the application of PCA on our normalized data. Then in our pattern layer we have the Gaussian function. Then the sum of all the Gaussian distance is added in the summation layer and an average is taken.

The Gaussian function formula we are using to evaluate the accuracy is

$$\frac{\sum_{k=1}^{ni} e^{-\frac{|X-X(i,k)|^2}{\sigma^2}}}{n_i}$$

Where $X(i, k)$ is the attribute k of i^{th} classifier,

And the n_i is the test case i ,

And σ is the smoothing factor.

Here sigma is the smoothing factor whose value can be varied from 0.1 to 0.9 and after applying the test sample and getting the result we can determine which sigma value is appropriate for our PNN.

Then in the output layer we take the best class that is the one with the maximum value from the summation layer.

Here is the network along with all the formulas we are using in our Neural Network.

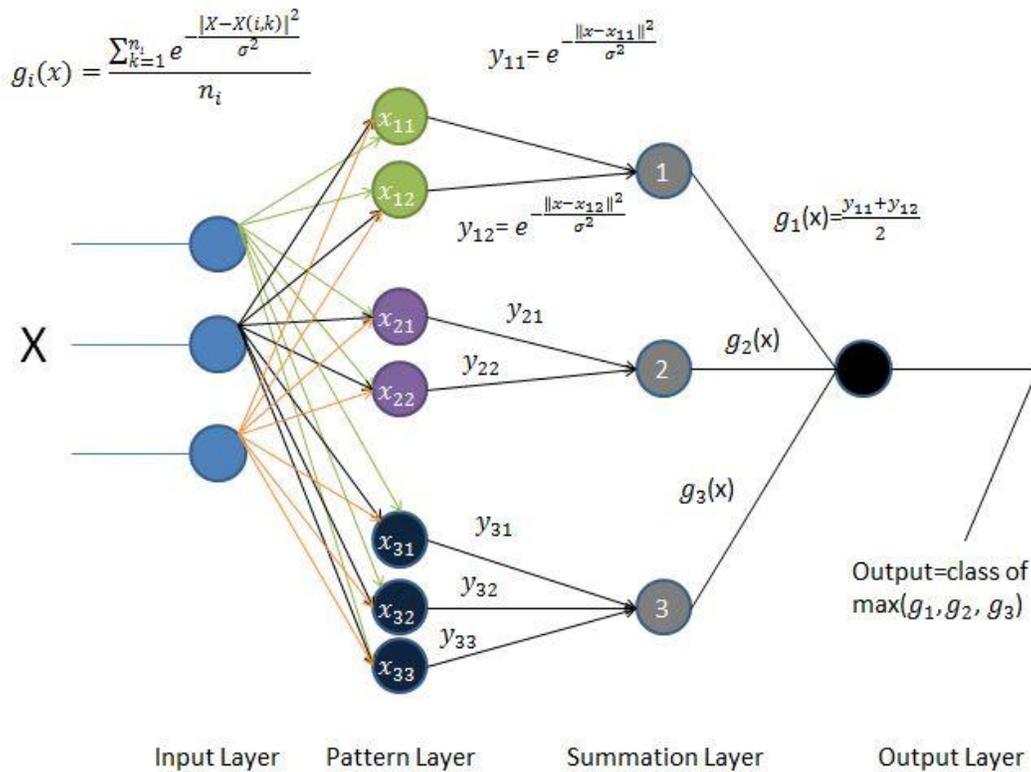


Figure 4. Probabilistic Neural Network Structure used in the algorithm along with the Gaussian formula and averaging function.

The basic steps of the algorithm in PNN are:

1. Begin
2. Select a subset form the dataset which will be the test dataset and select another which will be the training dataset.
3. Pass the test dataset to the input layer of the neural network.
4. From the input layer it goes to the pattern layer where it goes to each and every training sample of the training dataset class wise.
5. For each class the accuracy is found using the Gaussian function.

6. Then in the summation layer accuracy of each training sample of a particular class is added.
7. In the output layer the maximum of all the values from the summation layer is found and the class corresponding is given as the output.
8. End

The architecture (attributes, total number of nodes in pattern layer, number of nodes in summation layer, final output of the PNN) for different datasets of the PNN are:

- **Lung Cancer Dataset:** The architecture of the PNN is (57x21x3x1)
- **Libra Movement Dataset:** The architecture of the PNN is (91x105x15x1)

It is tested on Lung cancer data and Libra Movement dataset and iris dataset.

5.3 K-Nearest Neighbour:

In our project we use the Euclidean distance to find the nearest neighbor in the KNN algorithm in pattern classification. As KNN being a supervised classification technique we give it an input subset from our dataset for training sample. Then using that training sample it will classify the test sample.

Steps involved in KNN:

1. Begin
2. Select a subset which is already classified such that it can act as a training sample.
3. Select another subset which will act as a test dataset.
4. Repeat step 5-7 until all the test sample is classified
5. For a particular sample calculate the Euclidean distance of it from all the training samples.
6. Arrange the distance in terms of ascending order.

7. Find the k^{th} closest neighbour and depending upon the class of that particular neighbour, the class of the test is found.
8. End

It is tested using lung cancer data and iris dataset.

5.4 Genetic Algorithm:

In our project we have tried to use Genetic algorithm for pattern classification in a 2 dimensional data space. Here we use Genetic Algorithm to define the class boundaries which can distinguish between two samples of different class easily. We have the samples as point in our 2-dimensional space and then we try to generate lines at random which are then given to GA as input and optimized.

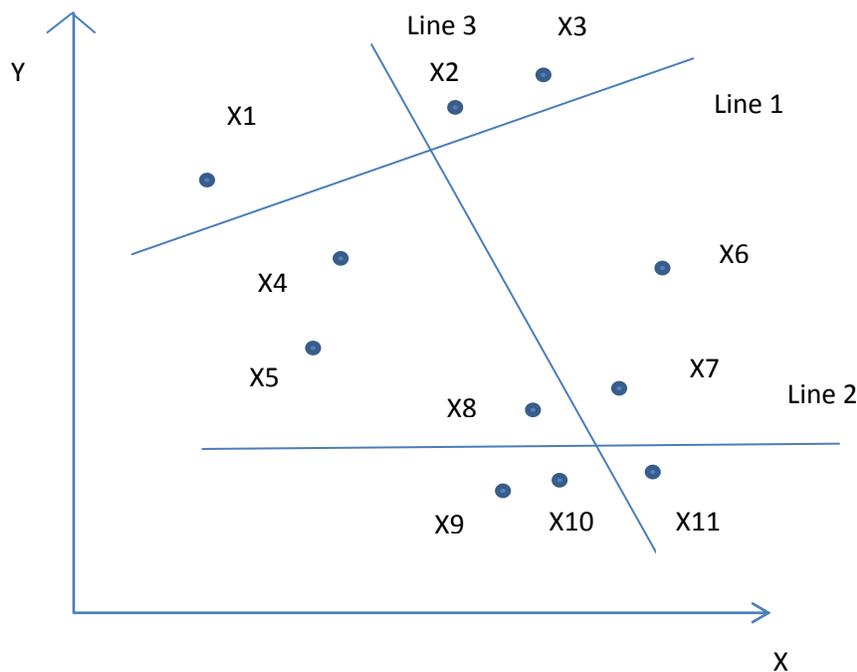


Figure 5. 2-dimensional space with data points and lines dividing them

Chapter 5. Experimental Details

Initially we collect all the points and make a rectangle bounding all the points that gives us our search space for the lines. A line can be represented as

$$x\cos\Theta + y\sin\Theta = d$$

where,

Θ is the angle with the y-axis,

d is the distance from the origin

Then in the search space we calculate the maximum number of lines possible in the same. So we calculate the minimum distance between any two points in the space. And then we calculate the diagonal distance of the search space. Considering the distance between any two line minimum equal to half of the minimum distance, we can find the maximum number of lines. It can be found by diagonal distance divided by the minimum distance multiplied by 2 (greatest integer).

Now that we have got the \max_lines we can now go to the slope of the line which is determined by the help of Θ . Θ can range from 0 to π . So we take the angle as $0, \alpha\pi, 2\alpha\pi, 3\alpha\pi, \dots, n\alpha\pi$ where n is $1/\alpha$. In that way we can have our \max_lines as an integer and \max_angle as also an integer.

We take an initial slope and draw two lines passing through the two base points and find the line which ever has a minimum distance from the origin. That is the base line and any line below that line is discarded. So the maximum a line can go can be with

$$d = d_{\min} + ((dist/2) * \max_lines)$$

so this can be thought as $d = d + (p*(dist/2))$ where p can be from $[0, \max_lines]$.

In this way we got our range of the population that can be made. Now we can randomly select any two p and n and we get our H number of lines. And these h number of lines are represented in GA with the help of p and n .

Chapter 5. Experimental Details

After that we go for the classification. For each point we calculate whether it is in positive side or the negative side of the lines and store them as a string. For every point we get a h bit string of 0,1 which indicate that it lies in that region. In a h lines case there can be maximum 2^h regions. If in a single region there is only one class point then that is assigned to it but if there are multiple classes we try to find the class which has the majority and in case of a tie they are broken randomly. After this we can calculate the total number of miscalculation, that serves as our fitness function we have to reduce the total misclassification.

From the figure 5 we can have a sample of classification

- Region 000---> class 1(0 samples) class 2(2 samples)
- Region 001---> class 1(1 samples) class 2(0 samples)
- Region 010---> class 1(2 samples) class 2(1 samples)
- Region 011---> class 1(1 samples) class 2(1 samples)
- Region 100---> class 1(0 samples) class 2(0 samples)
- Region 101---> class 1(0 samples) class 2(0 samples)
- Region 110---> class 1(1 samples) class 2(0 samples)
- Region 111---> class 1(1 samples) class 2(1 samples)

As we can see that there are misclassifications, in region 010 we have to take it as class 1 as it has more number of class 1 than class 2 and similarly we classify all the elements and calculate the fitness value and if it is upto our requirement then we can accept the lines. Otherwise we can take the current population and generate a new population by the help of GA. We have to take p and n and apply GA to get new set of n and p. We continue our process until we get a minimum misclassification or upto a particular number of generation.

The control parameters for our genetic algorithm are:

- Probability of crossover (P_c)=0.6
- Probability of mutation (P_m)=0.05
- Chromosome length is $\max_lines + \max_angles$.
- Population length can be $(\max_lines + \max_angles) * H$.

Chapter 5. Experimental Details

Steps involved in GA:

1. Begin
2. All the points are taken into a 2-d space.
3. We draw a rectangle bounding all the points.
4. The min line as the lower boundary was found out using the distance from the origin
5. Calculate the max_lines and maximum angles possible.
6. Generate the first population using any random function.
7. Repeat step 9-11 until total generation < 500 or misclassification < 10%
8. Using the first population find the total misclassification.
9. If misclassification > 10% then go for crossover and mutation using for the lines which give high value of classification.
10. Increment generation and update new population.
11. End

It is tested using a randomly generated dataset and the results are shown in the results.

CHAPTER 6

Results and Discussions

6.1 Probabilistic Neural Network:

It was tested using the lung cancer data and Libra Movement data.

The graph obtained for various sigma values for the lung cancer data is below.

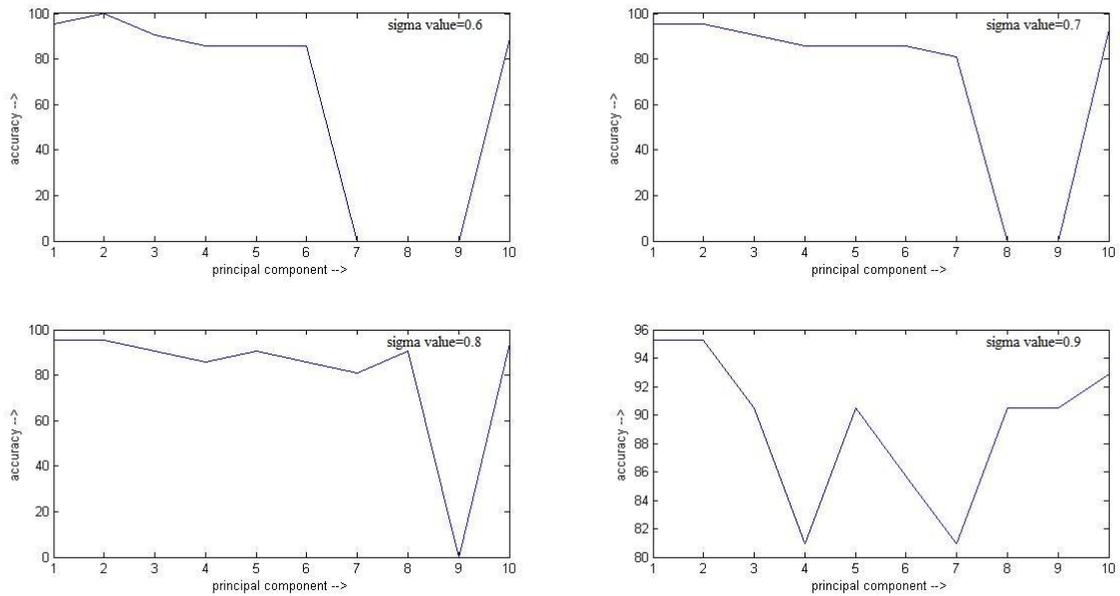


Figure 6. Graph of accuracy v/s principal component for different sigma values

Here we can see that for sigma value we get an accuracy of an average of 87.4%.

Chapter 6. Results and Discussion

The graph obtained for various sigma values for the lung cancer data is below

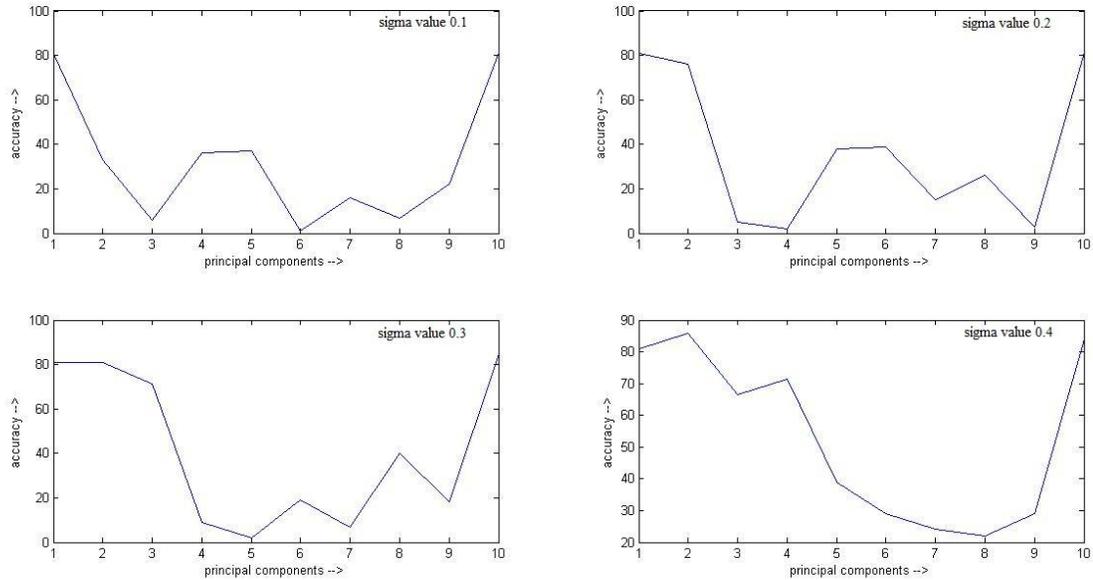


Figure 7. Graph of accuracy v/s principal component for different sigma values

As we can see for lower values of sigma the accuracy decreases.

6.2 K-Nearest Neighbour:

It was tested using the lung cancer data and Libra Movement data.

The graph obtained for the lung cancer data is below.

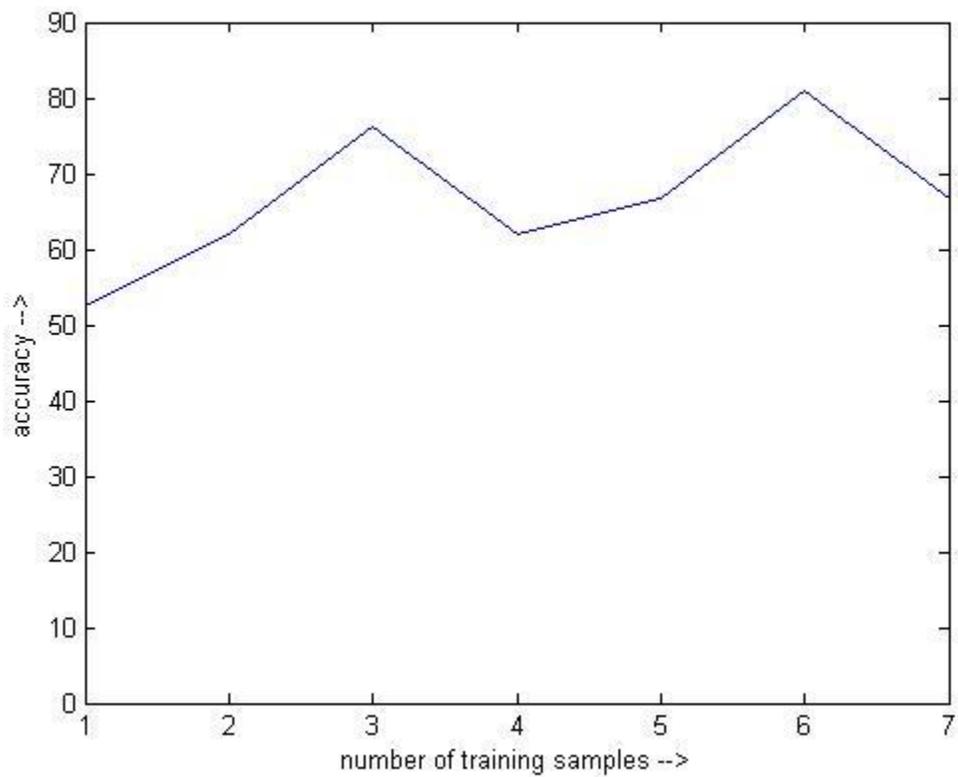


Figure 8. Graph of accuracy v/s number of training samples for KNN using lung cancer data

The accuracy of 80.95% highest is achieved at using 6 training samples. Depending upon the training sample and number of training sample accuracy varies.

The graph obtained for the Libra Movement data is below.

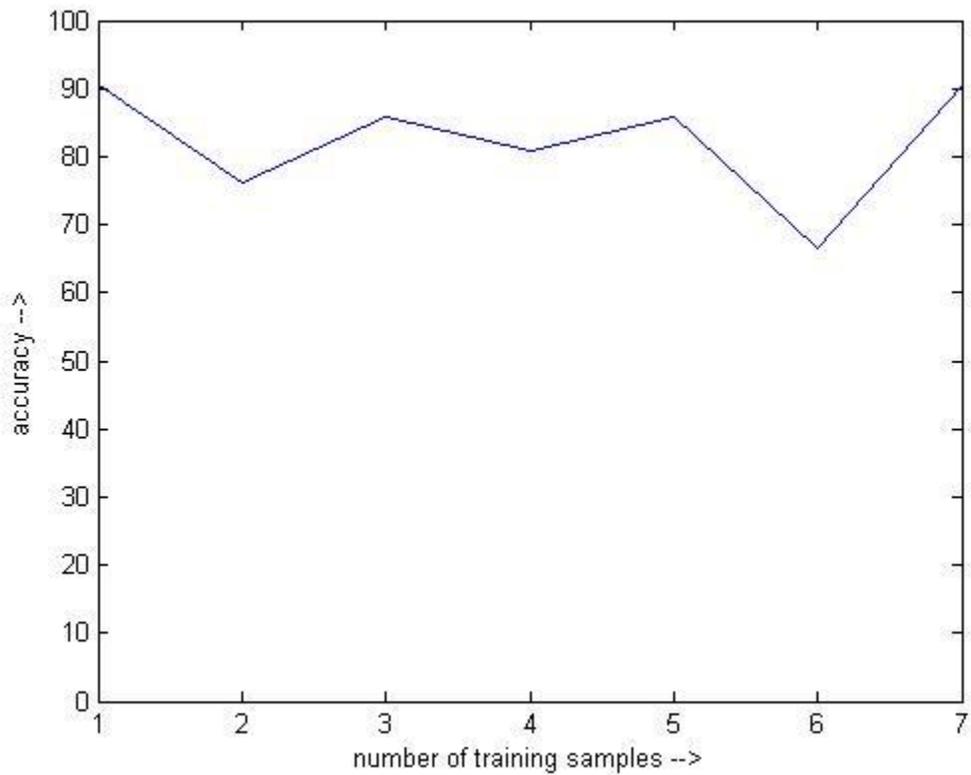


Figure 9. Graph of accuracy v/s number of training samples for KNN using Libra Movement data

The accuracy of 90.48% highest is achieved at using 1, 7 training samples. Depending upon the training sample and number of training sample accuracy varies.

6.3 Genetic Algorithm:

It was tested using random data.

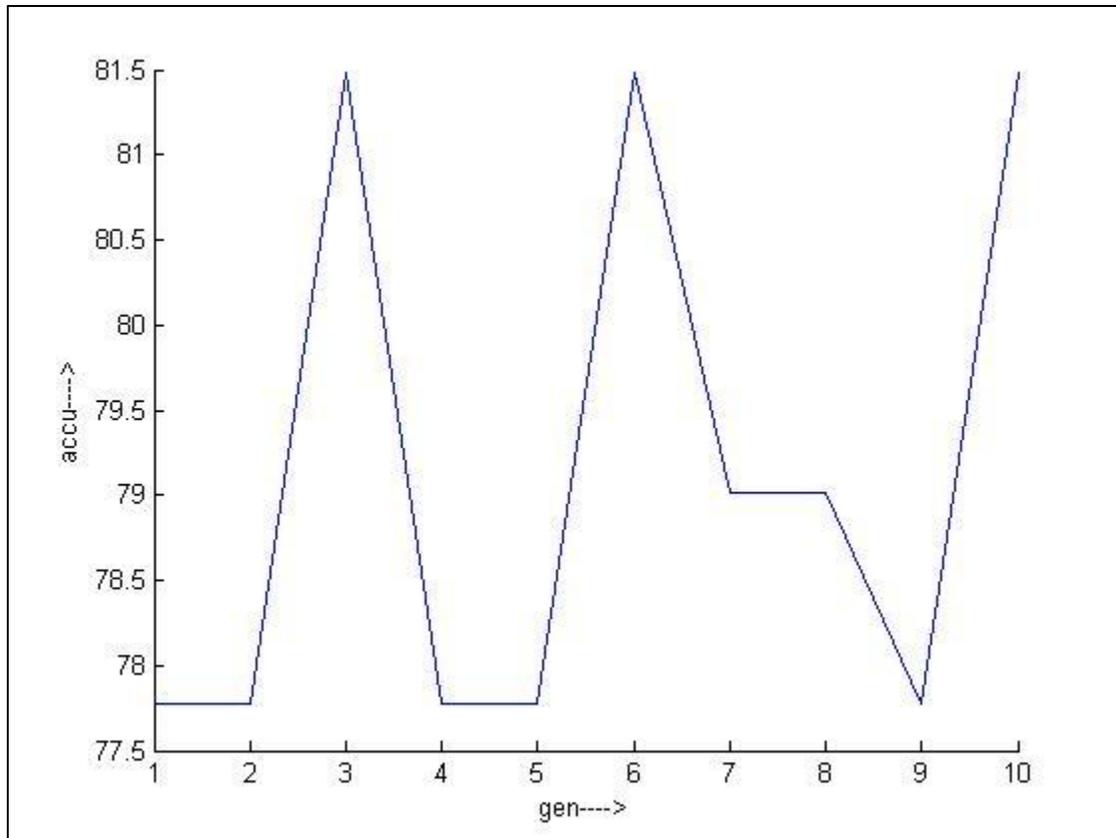


Figure 10. Graph of generations v/s accuracy

We can get drop in accuracy because of the mutation step.

Chapter 6. Results and Discussion

For different values of H (number of lines) we get different accuracy.

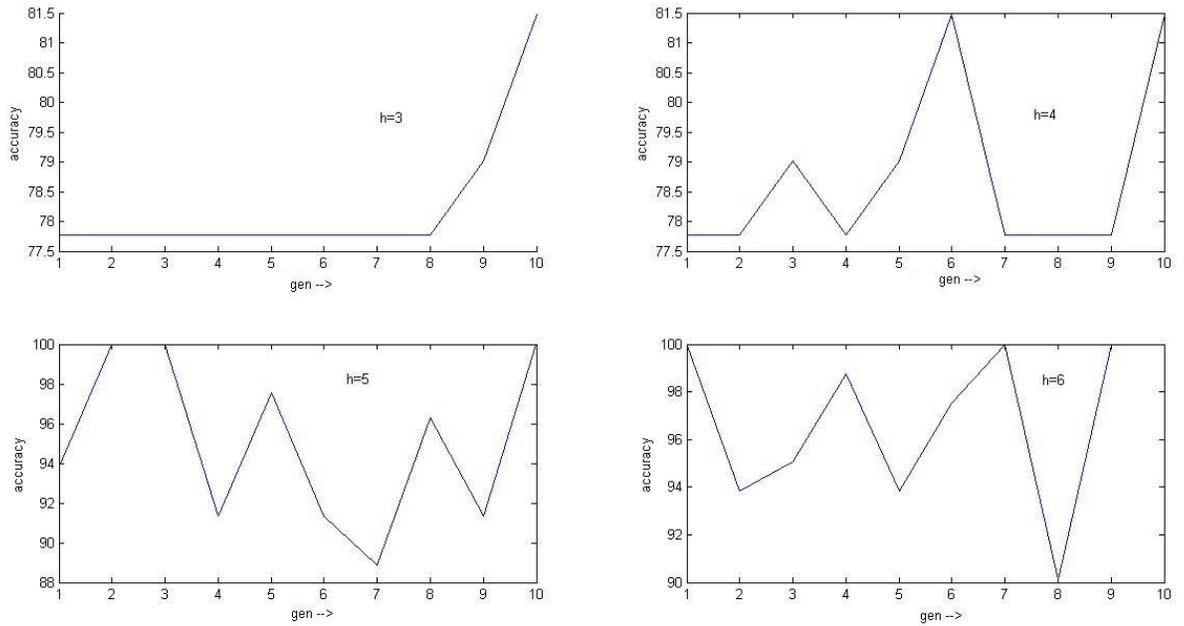


Figure 11. Graph of accuracy v/s generation number with different values of H

As we can see for h=6 we get accuracy as 100% for some generation.

CHAPTER 7

Conclusion

For a smaller data sample PNN works efficiently as compared to the other two techniques. But as the size of the training dataset increases the network complexity of PNN grows as a result computation takes too much time. The same problem also arises when KNN as a classifier is used for large training and test dataset. But from the above implementations Genetic Algorithm is efficient when used as a classifier as it divides the n-dimensional plane into various parts thereby classifying the dataset into respective classes. Overall GA classification technique is effective in classification of data samples. It is observed that for a large number of H (lines) the accuracy of the GA classifier further increases.

Bibliography

1. Sankar K. Pal, fellow, IEEE, and Sanghamitra Bandyopadhyay, and C. A. Murthy
“Genetic algorithms for generation of class boundaries” VOL. 28, no. 6, December 1998
2. S Bandyopadhyay, C A Murthy and S K Pal Pattern “Classification with Genetic Algorithms” 16th march 1995
3. “Fundamentals of Genetic Algorithm”, AI Course lectures, pg:39-40, R. C. Chakraborty
4. Dongqing Liu, Ting Shi, Joseph A. DiDonato, John D. Carpten, Jianping Zhu and Zhong-Hui Duan “Application of Genetic Algorithm/K-Nearest Neighbor Method to the Classification of Renal Cell Carcinoma”, 2004 IEEE Computational Systems Bioinformatics Conference (CSB 2004)
5. Ying Lu and Jiawei Han, “Cancer Classification Using Gene Expression Data”
6. Lindsay I Smith, “A tutorial on Principal Components Analysis”, February 26, 2002
7. M Farrokhrooz, H.Keivani, H.Rastegar, “Marine Vessels Classification using Probabilistic Neural Networks”, 2005 IEEE.
8. Jonathon Shlens, “A Tutorial on Principal Component Analysis”, Dated: April 22, 2009, Version 3.01)
9. Ibrahiem M.M.E.I Emary and S. Ramakrishnan, “On the application of various probabilistic neural networks in solving different pattern classification problems”, World Applied Sciences Journal, 4(6):772 780, 2008.
10. Leila Fallah Araghi, Hamid Khaloozade , Mohammad Reza Arvan , “Ship Identification using probabilistic neural networks”, Proceedings of the International Multi Conference of Engineers and Computer Scientists, 2009 Vol II , IMECS 2009, March 18 - 20, 2009, Hong Kong.
11. Khalaf khatatneh, Ibrahiem M.M El Emary and Basem Al- Rifai, “Probabilistic Artificial Neural Network for Recognizing the Arabic Hand Written Characters”, Journal of Computer Science, 2 (12): 879-884, 2006 ISSN 1549-3636.
12. Lindsay I Smith, “A tutorial on Principal Components Analysis”, February 26, 2002
13. Donald F Specht, “Probabilistic Neural Network”, Neural Networks, Vol.3 pp. 109-118, 1990