CANCER PREDICTION
USING
PATTERN CLASSIFICATION OF MICROARRAY DATA

By: Sudhir Madhav Rao &Vinod Jayakumar
Instructor: Dr. Michael Nechyba

1. Abstract

The objective of this project is to apply well known statistical methods to reliably classify tumors based on the gene expression values obtained from cDNA micro arrays. In this paper we compare the performance of different discriminant methods ranging from a simple Bayesian Classification to more sophisticated methods like Neural Networks, SOM and Classification Trees. The performance obtained was comparable and for some models even better than already published work in literature.

Keywords: cDNA Micro array, Tumors, Leukemia, Discriminant Analysis.

2. Introduction

The reliable classification of tumor is essential for apt treatment. Current clinical and morphological methods for classifying human tumor are highly subjective and unreliable. cDNA micro arrays and high – density oligonucleotide chips are proving to be powerful tools in the treatment of cancer [3,5]. By allowing the monitoring of expression levels of thousands of genes simultaneously, such techniques may lead to more complete understanding of the molecular variations among tumors and hence to a finer and more reliable classification.

The enormous gene database available poses numerous challenging questions ranging from analysis of images produced by micro array data to feature extraction, prediction and discovery of tumor. The inherent problem in this data set is the relatively larger size of variables (genes) compared to the samples (observations).

The dataset used in this project is the classic Golub ALL-AML cancer dataset obtained from the URL: http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi. The objective is to classify the samples into two types of leukemia namely Acute Lymphoblastic leukemia (ALL) and acute Myeloid Leukemia (AML). The data set consists of 72 samples each containing 7129 genes. Out of the 72 samples 38 training samples were collected from adult bone marrow. The remaining 34 test samples are more heterogeneous as it comprises of 24 bone marrow samples and also 10 blood samples. Further the test samples were procured from both adult and children with a totally different sample preparation protocols.

A broad range of parametric as well as non-parametric statistical methods were covered. The classification started off with simple methods like ML based classification and Fisher Linear Discriminant Analysis (FLDA). Further we considered more sophisticated non parametric methods like K-Nearest Neighbor (KNN), Neural Networks and Classification Trees. The performance of all these methods for this problem is analyzed and compared.

3. Preprocessing and Feature Extraction (gene selection)

The preprocessing involves [2],

- Thresholding: floor of 100 and ceiling of 16000
- Filtering: exclusion of genes with max / min <=5 and (max – min)<=500.

- Base 10 logarithmic transformation.
- Normalize data to mean 0 and variance 1

After preprocessing there were 3051 genes remaining. Each gene is assigned a weight equal to **3.1**. It was found that genes top 40 weights are sufficient for classification [2].

$$\frac{\sum_i \sum_k I(y_i = k)(\bar{x}_{kj} - \bar{x}_{.j})^2}{\sum_i \sum_k I(y_i = k)(x_{ij} - \bar{x}_{kj})^2}, \tag{3.1}$$

For algorithms like Expectation-Maximization (EM) and FLDA the dimension of the data should be lesser than the number of observations. Instead of selecting the best 10 genes out of the top 40, a simple 1 Gaussian EM algorithm was executed on each of the 3051 genes. The 10 genes which gave the lowest misclassification error were selected. Though these 10 genes occur in top 40 they are not necessarily in the same order of importance. Thus it would be naïve to select best 10 genes among the top 40.

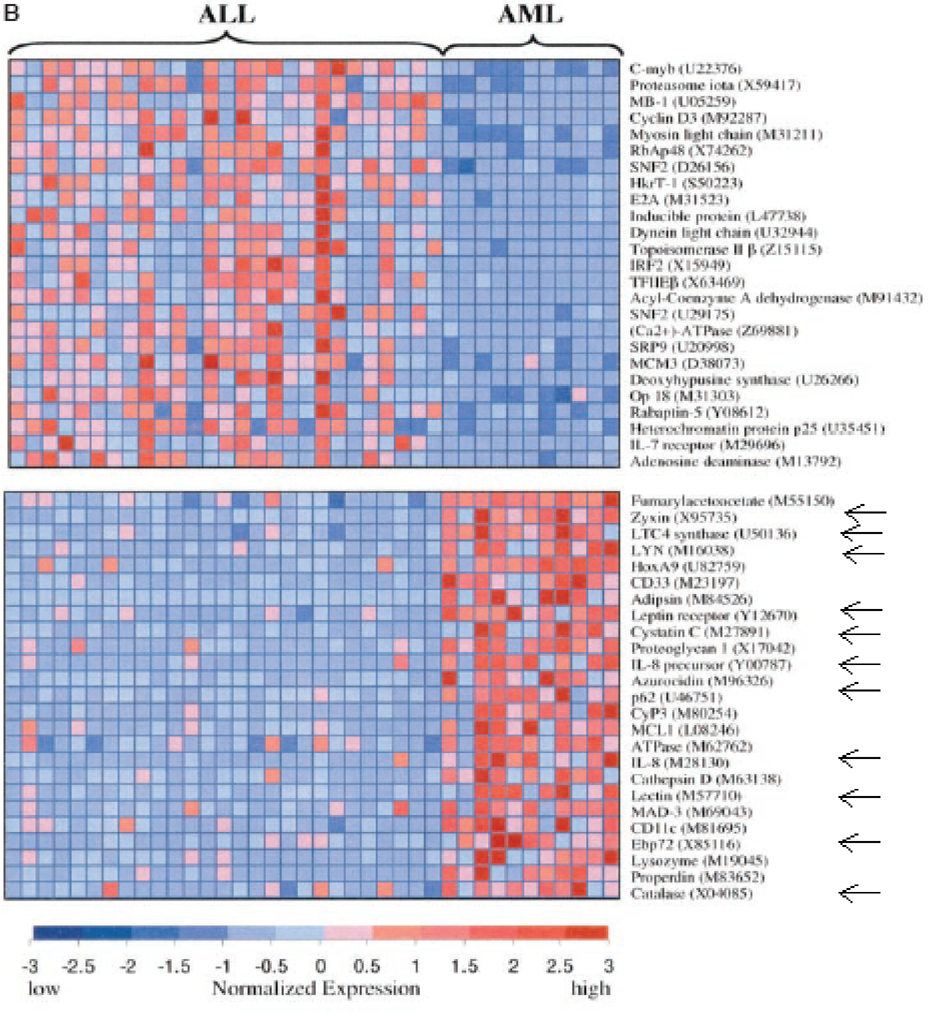

Figure 3.1.Prominent genes got by Golub [3]. The arrows show the top 10 genes selected by 1 Gaussian EM algorithm.

From the above figure we note that all the 10 genes appear among the genes which are highly correlated with AML. This is intuitively appealing because for these genes (depicted by rows) the ratio of expression values for AML and ALL samples (depicted by columns) is very high.

4. Methods of Classification

The following methods are addressed in this paper

1. Weighted Voting Method
2. ML based classification
3. Fisher Linear Discriminant Analysis (FLDA)
4. Expectation-Maximization (EM)
5. K Nearest Neighbor (KNN)
6. Neural Networks (NN)
7. Support Vector Machines (SVM)
8. Self Organizing Map
9. Classification Trees

5. Weighted Voting Method (WVM)

This is a simple Bayes classifier implemented by Golub [3] and implemented here for comparison purposes only. A different standardization of the data was followed. Prior to logarithmic transformation and after thresholding, a subset of 50 genes was selected based on parameter in equation **5.1**. 25 genes with the largest P ratio (highly correlated with ALL) and 25 genes with the smallest P ratio (highly correlated with AML) constituted the 50 set. The learning data was then log transformed and normalized to mean 0 and variance 1.

$$P(g)= (\mu_{ALL} - \mu_{AML}) / (\sigma_{ALL} + \sigma_{AML}) \qquad \textbf{(5.1)}$$

The classification was based on simple voting scheme. For each sample a particular gene g casts a vote given by the formula **5.2.** Depending on whether the vote is positive or negative, the particular gene casts vote in favor of one of the two classes. The total such votes is summed for each class and the sample is categorized as the class which has the highest magnitude of votes.

$$V = P(g)*(x-b) \text{ where } b = (\mu_{ALL} + \mu_{AML})/2 \qquad \textbf{(5.2)}$$

In Bayesian binary detection the discriminant function is 5.3 assuming that the within class variance σ^2 of classes ALL and AML are identical. The WVM gives 1 misclassification in train as well as in test data.

$$V = ((\mu_{ALL} - \mu_{AML}))/ \sigma^2)*(x-b) \qquad \textbf{(5.3)}$$

6. ML (Maximum-Likelihood) based classification:

The ML based decision is given by,

$$\sum_{j=1}^{p} \frac{(x_j - \bar{x}_{2j})^2}{\hat{\sigma}_j^2} \geq \sum_{j=1}^{p} \frac{(x_j - \bar{x}_{1j})^2}{\hat{\sigma}_j^2}, \qquad \textbf{(6.1)}$$

Then classify 1, if not classify 2.
Where,
p – Number of genes selected for classification (or dimension)
x_{2j} - mean vector of class 2
x_{1j} - mean vector of class 1
σ - Std deviation of the respective classes.

Misclassification as a function of dimension of the data is shown in the figure below,

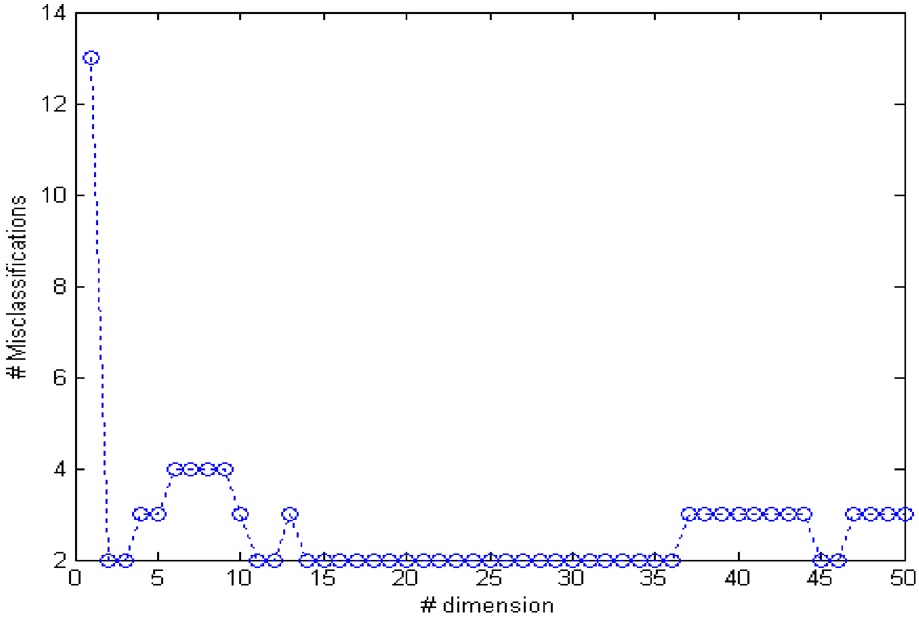

Figure 8.1: Misclassification Vs dimension

The least number of misclassifications obtained is 2. Hence, ML proves to be an effective and simple method for classification.

7. Fisher Linear Discriminant Analysis:

Classification can be erroneous when the dimension of the data is large compared to the sample size. FLD finds the optimum vector in space, projection of data on which minimizes misclassifications. This vector is given by

$\mathbf{w} = \mathbf{S_W}^{-1} (\mathbf{m}_1\text{-}\mathbf{m}_2)$ **(7.1)**

Where,

$\mathbf{S_W} = \mathbf{S}_1\text{+}\mathbf{S}_2$ **(7.2)**

$\mathbf{S}_i = \Sigma\ (\mathbf{x}\text{-}\mathbf{m_1})\ (\mathbf{x}\text{-}\mathbf{m_2})^t$ **(7.3)**

$\mathbf{m}_1$ **and** $\mathbf{m}_2$ are mean vectors of class ALL and AML.

The projected data is given by,

$y = \mathbf{w}^t\ \mathbf{x}$ **(7.4)**

Note that y is a scalar. Hence, we have a one-dimensional data to work with, relaxing the classification part. The projection of 38, 10-dimensional training samples is shown below.

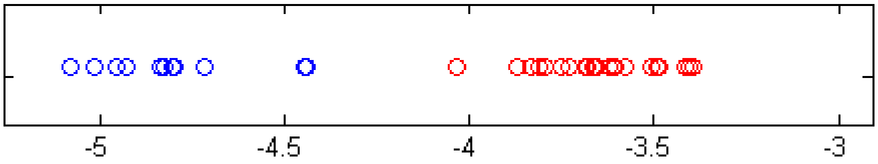

Figure 7.1: distribution of ALL and AML sample points after projection onto the vector w

7.1Classification:

Since the data is one-dimensional, classification can be performed using simple classifiers like distance method, KNN. The performance of EM algorithm is also tested on the 1-D data.

7.1.1 KNN:

The number of misclassifications is obtained as a function of dimension of the training set before projection and k. the optimum performance (# of misclassification =1) is obtained for d= 10,12,13,15 and k=3, respectively. An interesting point to note is that optimum k=3 for FLD as compared k=1 when working on a d-dimensional data. Loss of information when projecting from a d-dimensional space to 1-D might be the reason for this. Misclassification as a function of‘d’ and ‘k’ is show below.

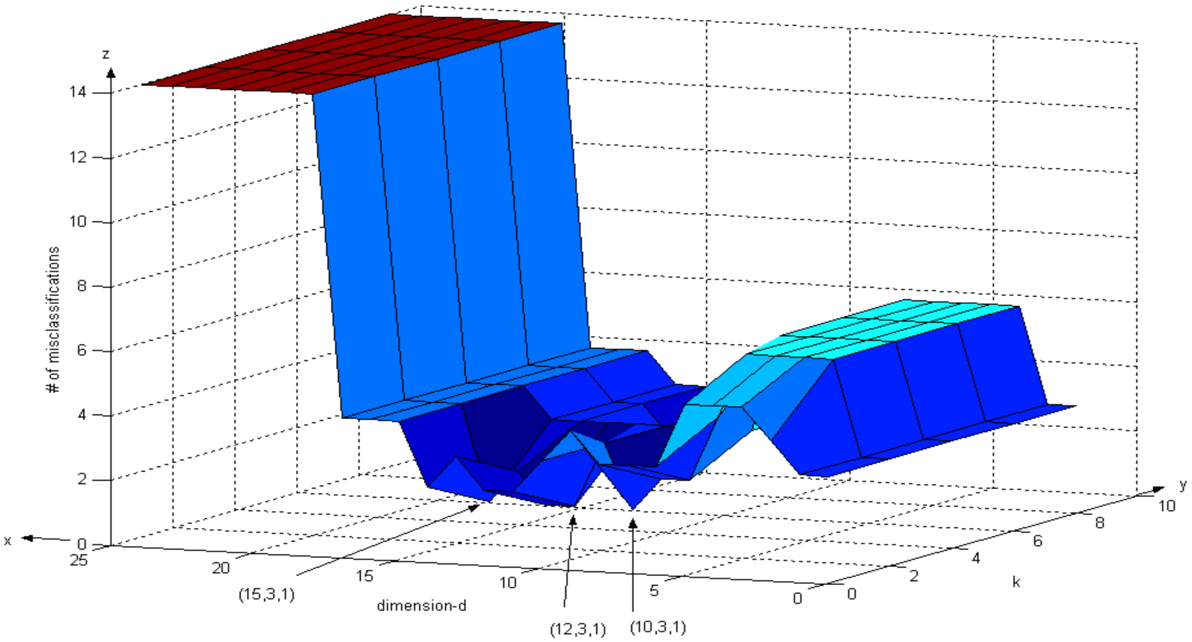

Figure7.2. Misclassifications as a function of k and dimension's'd'.

Red portion of the graph shows highest number of misclassifications. This is because the covariance matrix **S** is singular for higher dimensions. (n>>d is a requirement for FLD).

7.1.2EM:

Misclassification error is obtained as a function of dimension of the original data for a 1-Gaussian mixture EM algorithm.

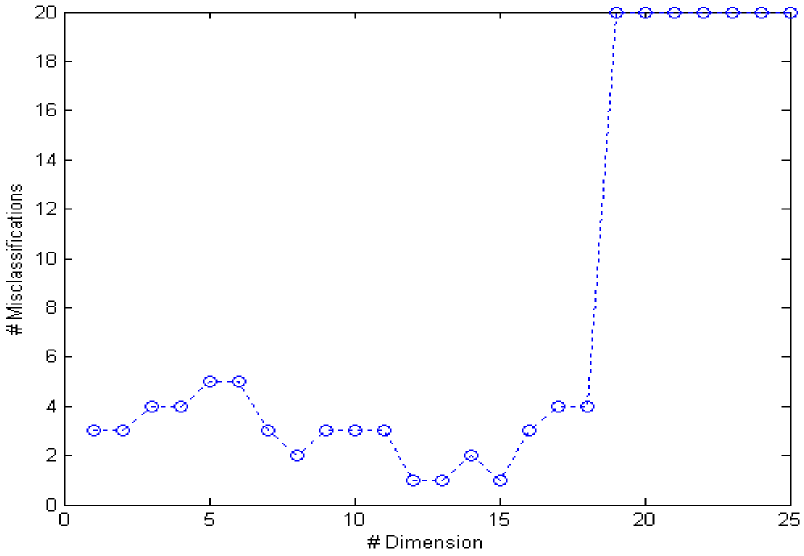

Figure7.3.Misclassification Vs Dimension.

The optimum performance is obtained when a 12, 13 or 15 dimensional data is linearly converted to a scalar quantity.

8. Expectation-Maximization Algorithm:

The EM algorithm is an effective tool to model the distribution and clustering of data within a class. A wide variety of distributions can be efficiently modeled using mixture of gaussian distributions. The dimension of the data has to be reduced as the numbers of training sample points are limited. The performance of the k-mixture EM algorithm is tested for different values of 'k' and dimension and is summarized in table 1. Here, the dimension of the data implies the number of genes selected for the discrimination of samples. The maximum dimension chosen is 10.

Table 8.1 Performance of EM algorithm

ALL		AML		# Misclassifications
k	d (# genes)	k	d	
1	10	1	1,2,3,4,5	4
2	6	2	3	2
2	6	1	3,4	2
3	1	1	2	7
3	1	3	2	7

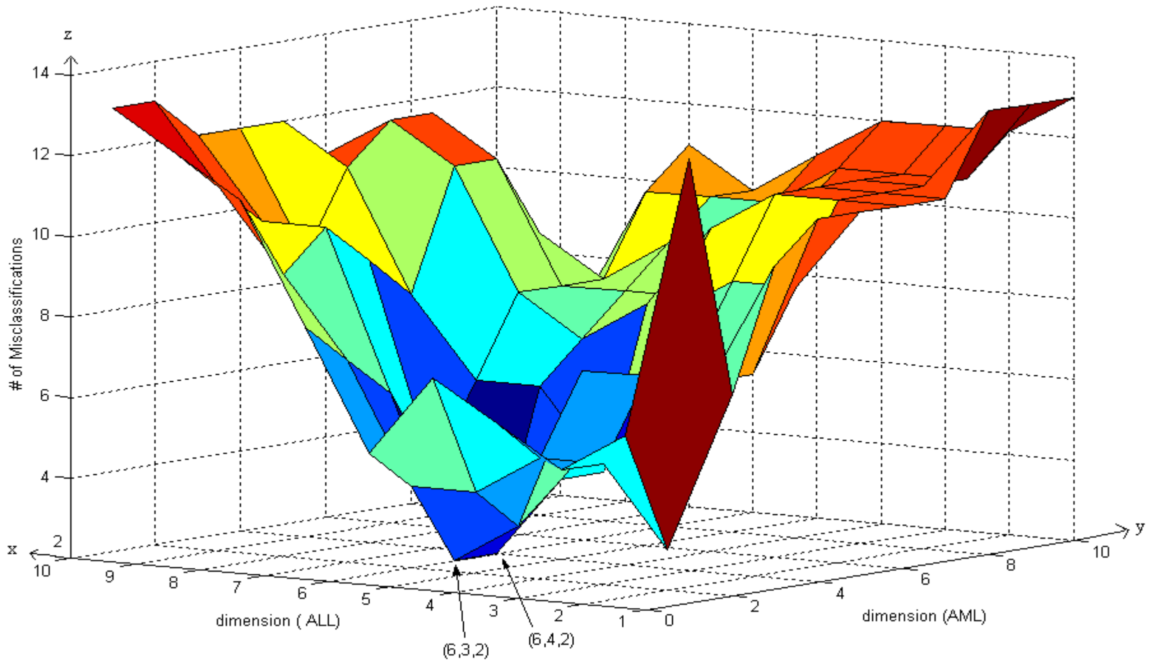

Figure 8.1: # of misclassifications as a function of dimension of ALL and dimension of AML. The least misclassifications occur at points (6, 3) and (6, 4), i.e. dimension of ALL distribution is 6 and dimension of AML distribution is 3 or 4.

As we can see from figure 1, the # of misclassifications increases as the dimension (i.e., the number of genes selected for classification) increases. This might be due to the limited sample size of the training data. Another interesting feature of the EM model is that the dimension of the ALL distribution has to be greater than the AML distribution to obtain the least error. The reason could be that there is more number of ALL training samples compared to ALL. The other reason could be that the ALL sample vectors appear predominantly in three clusters, while the AML appear in a single cluster. This is illustrated in the figure below.

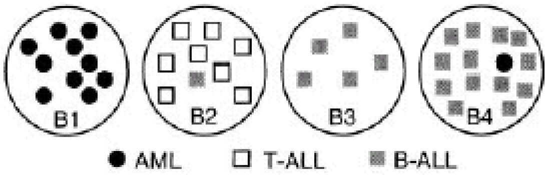

Figure 8.2: Clusters in ALL and AML

9. KNN (k-nearest neighbor):

In this method each test sample is projected into the d-dimensional space. In this space, k nearest neighbor samples from the training set is chosen. Each test vector is assigned ALL or AML by the majority voting among the k training samples. Figure 9.1 shows the variation of misclassifications as a function of k and d.

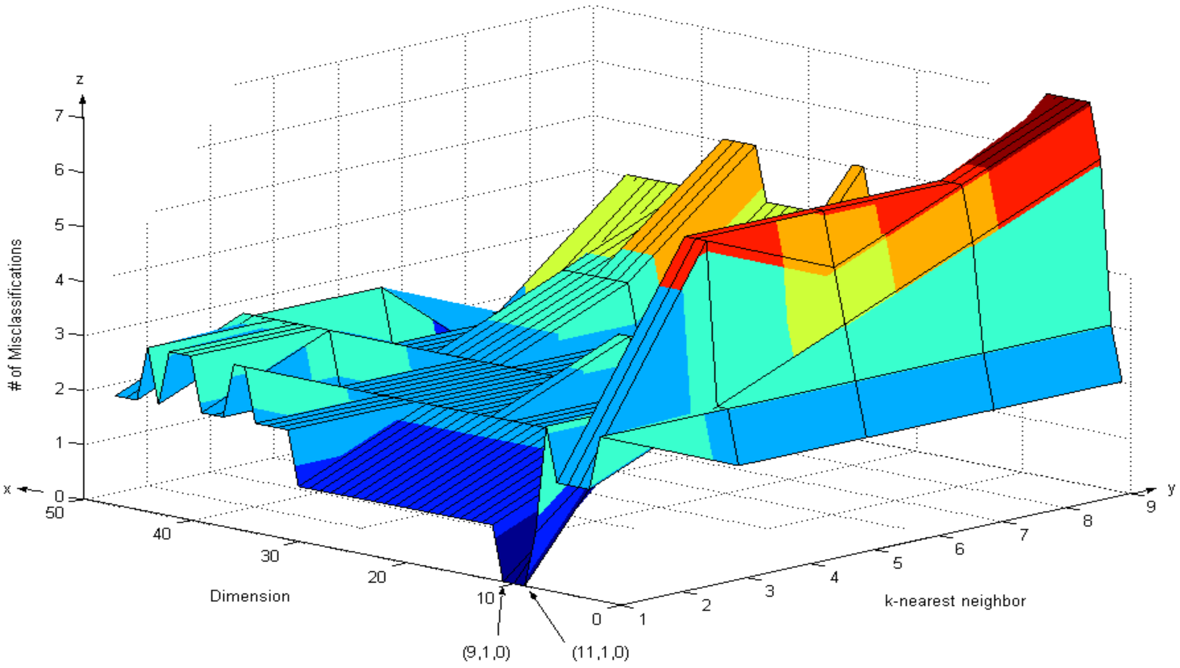

Figure 9.1: # of misclassifications as a function of k and dimension.

From the above figure we can conclude that we get the optimum performance when the dimension (# of genes) is 9, 10 or 11 and k =1(i.e., classify based only on the nearest neighbor). Sandrine Dudiot, et.al, also obtained this result.

10. Neural Networks

Neural Network (NN) is one of the most common non-parametric methods used for pattern classification. By not assuming any prior model for the data and using the correlations between various dimensions of the data, NN gives new insight into pattern classification.

For this project, online back propagation algorithm (BPN) with momentum learning was implemented. The code was written for a general one hidden layer MLP which takes in the input data, desired data, model size, number of epochs, learning rate and alpha and then gives the average error, final weight vectors and the output when the input data is passed through the final weight vectors. The basic architecture of the network is summarized below

Table 10.1 Basic Network Architecture

Parameters	Values
Number of Hidden Layers	1
Activation Function	Hyperbolic Tangent function
Number of Input nodes	50
Number of output nodes	2 (one for each class)

There were two methods which could be used to extract the significant genes out of 7129 genes: one was the 50 genes extracted by Golub as reported in [3] and the other was 40 genes extracted by Speed as described in [2] and both sets had some common genes. The BPN performance was tested with both input datasets. It was observed that the 50 dataset performed better as well as consistent compared to the 40 dataset. One possible explanation is that the 50 genes may be better correlated with each other which improved the performance of neural networks. Thus for remaining experiment we use the 50 genes as input data to the NN.

The same preprocessing steps as explained in Section were done on the input data so that the input data lies in the linear region of the activation function. This range of the output data was 1 and -1. The following step by step procedure was followed to construct the neural networks and find its optimal parameters.

a) Early Stopping: One of the main challenges faced in functional genomics is that the data set has very few samples. This may lead to over fitting of the model which is very true in NN. To prevent this early stopping procedure in which the training data is divided into estimation set and validation set. A typical value of r is 5 to 10% which means 5 to 10% of the training data is kept as validation set and remaining as estimation set. Here in this problem, we keep 3 out of 38 samples in validation set which mean r ~ 8%. A stable neural network with the following parameters was taken to conduct the experiment.

Table 10.2 Network parameters used for early stopping

Parameters	**Values**
Learning rate	0.01
Momentum constant	0
Number of Epochs	100
Number of iterations	20

The network was simulated 20 times and the average learning curve for validation set and estimation set was plotted as shown below. The figure clearly shows the validation set has minimum average output error at around 50 epochs (a complete pass of the whole training set makes one epoch). Thus the optimal number of training time for the NN is **50 epochs** (The red line is U shaped with minimum at around 50)**.**

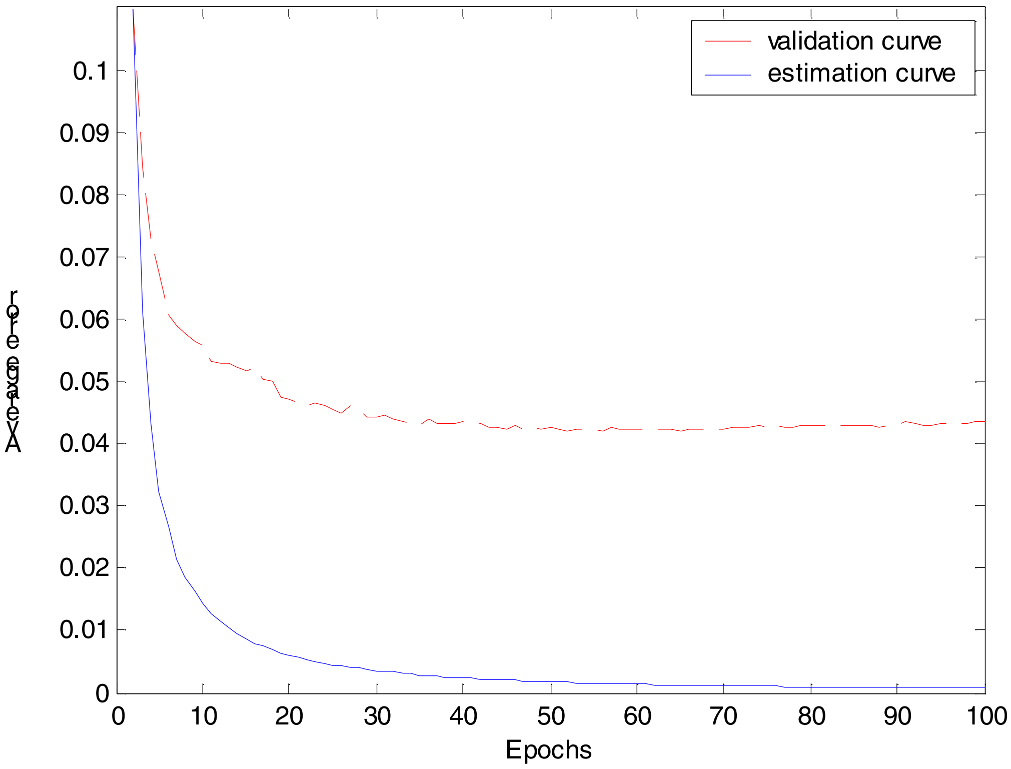

Figure 10.1 Early Stopping Criteria

b) <u>Hidden Nodes</u>: Having selected the optimal number of epochs, we next select optimal number of hidden nodes in the hidden layer. The learning rate and momentum constant were kept the same to 0.01 and 0 respectively and the number of epochs to 50. For a particular number of hidden nodes the NN was simulated 30 times and average percent correct classification and standard deviation were tabulated as shown below. The table clearly shows that N=5 is the optimal number of hidden nodes for which the standard deviation is also low.

Table10.3. Statistics of Percentage Correct Classification for Testing Data

Number of hidden nodes	Mean %	Standard Deviation %
2	93.3333	3.9334
3	93.2353	3.7163
4	93.3333	3.6989
5	94.4118	2.8221
6	94.1176	2.1847
9	93.4314	3.1548
12	93.6275	3.1924
15	93.4314	3.1548
18	92.2549	3.6624
21	92.5490	3.2526

c) Learning rate and Momentum constant (μ and α): With the optimal number of epochs and number of hidden nodes fixed at 50 and 5 respectively, we finally conduct the experiment to find optimal μ and α. For a particular combination of optimal learning rate and momentum constant the NN is run for 30 iterations and the average percentage correct classification was tabulated as shown below.

Table10.4. Statistics of Percentage Correct Classification for Testing Data

	α = 0.001	α = 0.005	α = 0.01	α = 0.05	α = 0.1	α = 0.15	α = 0.2
μ = 0	90.5882	93.5294	94.5098	95.3922	94.7059	94.2157	93.7255
μ = 0.2	93.8235	93.7255	93.3333	94.7059	94.6078	93.9216	91.7647
μ = 0.5	92.6471	93.4314	94.3137	94.4118	94.1176	94.0196	92.0588
μ = 0.8	92.9412	94.3137	93.9216	95.0000	93.0392	92.7451	91.8627

The results indicate that a learning rate of 0.05 and momentum constant of 0 performs the best. Thus the complete optimal network parameters for the **50-5-2** network are as shown below.

Table10.5 Final optimal Network parameters

Parameters	Values
Learning rate	0.05
Momentum constant	0
Number of Epochs	50
Number of Hidden nodes	5

The best result obtained for this configuration is 100% on train data and 97.06% on test data and with only one misclassification. The confusion matrix shows that one AML is misclassified. The learning curve shows smooth convergence.

Table3.2. Confusion Matrix for test data

	ALL	AML
ALL	20	0
AML	1	13

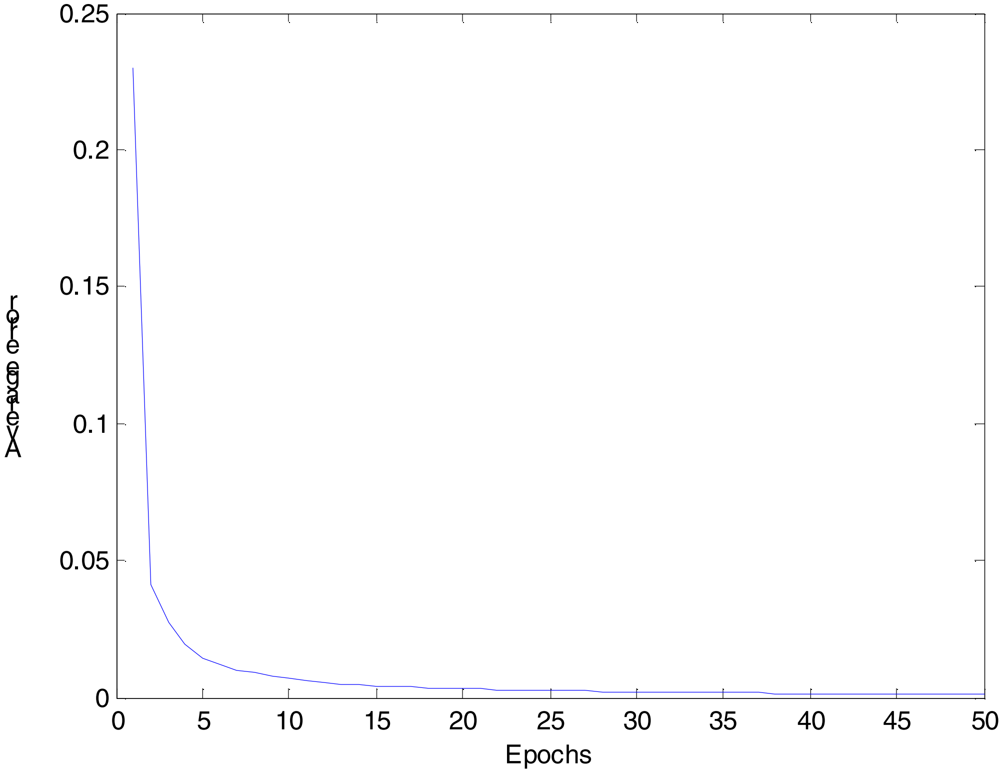

Figure 10.2 Learning Curve for the best neural network

Thus to conclude the neural network performs very well with only one misclassification on the test data and thus can be applied to problems of pattern classification in functional genomics.

11. Self Organizing Maps

Self Organizing Maps also nick named SOM is an unsupervised learning algorithm which finds clusters and groups in the original data. Being unsupervised method, SOM are not only applied for class prediction but also for class discovery in which new types of tumor are discovered by the clusters formed in the output space.

The Golub data set was analyzed using the SOM. A 2D matrix of neurons was used to directly map the 50D data to 2D. The 2D neurons were located uniformly in N*N grid in the 2D plane. The SOM algorithm was implemented for training the SOM with exponential decay of learning rate and variance parameters. As before 2 groups namely ALL and AML were selected to evaluate the performance of this unsupervised learning. The table below shows the average percentage of correct classification and standard deviation for 20 iterations as a function of no: of neurons.

Table11.1. Average Performance of Unsupervised learning on 50D test data

Size of grid (N)	No: of neurons(N*N)	Average % Correct	Standard deviation
2	4	95.0000	2.1550
3	9	95.2941	1.7595
4	16	96.1765	1.9322
5	25	97.7941	1.6180
6	36	98.2353	1.7595

Since the total available data points are only 38 in train data we cannot exceed more than N=6 as size of grid. Nevertheless we see that the percentage accuracy increases steadily on the training data. Obviously fitting N=6 is an over fitting model. N=4 is a good model and to see its clustering performance we plot the clustering in output 2D space for both train and test data.

The figure clearly shows that N=4 SOM achieves good clustering result One of the best results for this N=4 is given below. The confusion matrix shows that one AML is misclassified.

Table11.2. Best Performance of Unsupervised learning on 50D data

N	4
Percentage Correct Classification	97.06%

Table11.3. Confusion Matrix for test data

	ALL	AML
ALL	20	0
AML	1	13

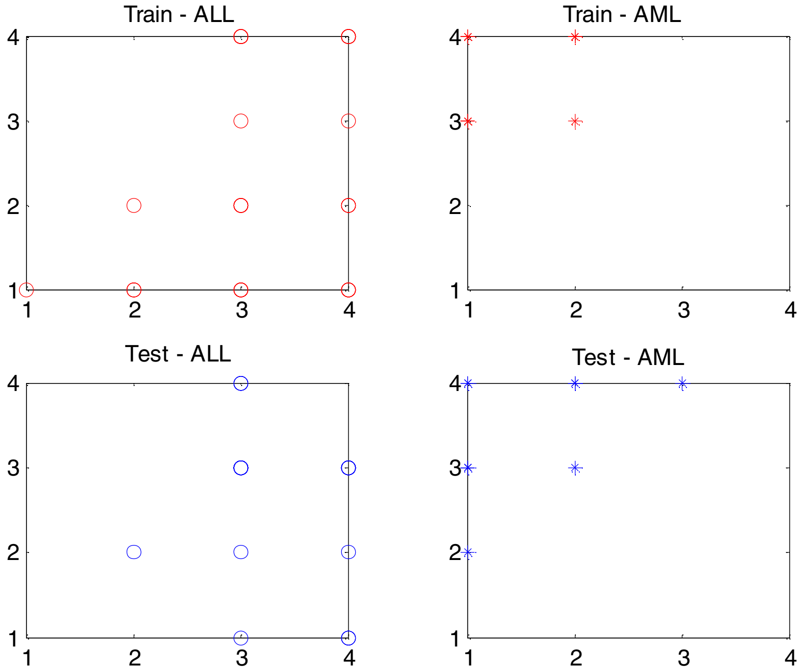

Figure 11.1 Clustering in output 2D space for N=4 grid size SOM

The successful application of unsupervised learning method like SOM with results comparable to supervised learning NN is a significant achievement in tumor classification.

12. Support Vector Machines

One of the universal feed forward networks like neural networks are Support Vector Machines (SVM) pioneered by Vapnik [4]. The main idea of support vector machines in the context of pattern classification is to construct a hyper plane as the decision surface in such a way that the margin of separation between positive and negative examples is maximized. Further SVM learning algorithm operates only in batch mode.

A main advantage of SVM is that it provides a method for controlling model complexity independent of the dimensionality. Thus no significant genes extraction is required. In our case of genome data, we give all the 7129 genes to the SVM classifier. The practical implementation of SVM is very difficult and so we used the SVM Matlab toolbox (link to the site is http://www.eleceng.ohio-state.edu/~maj/osu_svm/). The SVM classifier did 100% correct classification on train data and 97.06% correct classification on test data. The confusion matrix shows that one ALL is misclassified unlike in other types of classifier where generally AML was getting misclassified.

Table12.1. Confusion Matrix for test data

	ALL	AML
ALL	19	1
AML	0	14

The advantage of SVM is that it gives the same result for a particular dataset whereas in NN we get different results each time we simulate because of different weight initializations. Thus we don't need to search for best result as in NN. Further studies have shown that SVM classifier performs very well in tumor classification and in many cases gives the best result.

13. Classification Trees

The final classifier which was implemented was the Classification Trees which is a complete non parametric method. Binary Tree structured classifiers are constructed by repeated splitting the subsets (nodes) of the measurement space X into two descendant subsets, starting with X itself. Each terminal subset is assigned a class label and the resulting partition of X corresponds to the classifier.

There are three main aspects of tree construction:

(i). The selection of the splits
(ii). The decision to declare a node terminal or to continue splitting
(iii). The assignment of each terminal node to a class

The code was written in S-Plus software which uses the classification and regression tree method (CART). The program gave us a very surprising and interesting result. It classified the training data perfectly based on only one single gene. The gene was 4847th gene named Zyxin (X95735) which is a very important gene and appears both in Golub's 50 subsets and Speed's 40 subsets. For this gene if the expression values are less than 994 then it classifies the sample as ALL or otherwise as AML as shown in the figure below.

When this classification was tested on the test data it gave 3 errors – one misclassification of AML and two misclassifications of ALL. The confusion matrix shows this misclassification thus giving accuracy of only 91.18%

Table13.1. Confusion Matrix for test data

	ALL	AML
ALL	18	2
AML	1	13

This example throws light on the fact that taking the correlation among genes is very important in successful classification of test data which tree does not do. In general thus tree classifiers are not a good method for tumor classification.

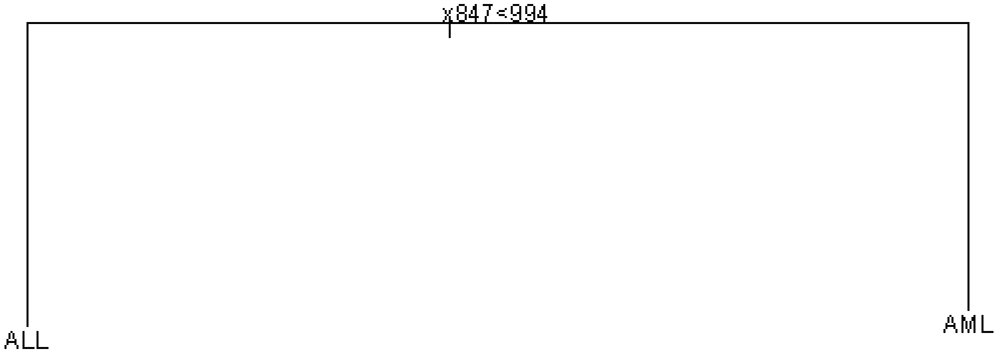

Figure 13.1 Tree Classification for Train data

14. Conclusion

The performance of different methods is summarized below.

Table14.1. Summary of Performance

Statistical Method	**# of Misclassification**
Weighted Voting Method (WVM)	1
ML based classification	2
FLDA	1
EM	2

KNN	0
Neural Networks	1
SOM	1
SVM	1
Classification Trees	3

From this table we conclude that simple algorithms like WVM, FLDA and KNN is sufficient for efficient classification. Zero misclassification was achieved for 1-NN which is not achieved by Sandrine Dudoit [2]. The inherent difference might be the due to different features selection method. The performance of these algorithms has to be evaluated using other datasets to have practical viability.

15. References

1. Chen-hsiang yeang, et, al. Molecular classification of multiple tumor types. Bioinformatics Vol 17, Suppl. 1 2001.
2. Sandrine Dudoit, et, al. Comparison of discrimination methods for classification of tumors using gene expression data. Technical report # 576, June 2000.
3. T.R.Golub, et, al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science, vol. 286 oct.1999.
4. Vapnik, V.N, 1995. The nature of statistical learning theory, New york; Springer-Verlag.
5. Simon Haykin, Neural Networks-A Comprehensive Foundation, Pearson Education Asia.
6. Richard O. Duda, Peter E. Hart and David G. Stork, Pattern Classification, Wiley, 2nd edition.