

# Neural Network Classifiers and Gene Selection Methods for Microarray Data on Human Lung Adenocarcinoma

Gaolin Zheng  
School of Computer Science  
Florida International University  
Miami FL 33199, USA  
(+1) 305.348.1749  
gzhen001@cs.fiu.edu

E. Olusegun George  
Mathematical Sciences Department  
University of Memphis  
Memphis TN 38152, USA  
(+1) 901.678.5088  
eogeorge@memphis.edu

Giri Narasimhan  
School of Computer Science  
Florida International University  
Miami FL 33199, USA  
(+1) 305.348.3748  
giri@cs.fiu.edu

## ABSTRACT

In an attempt to account for correlations in gene expression data, we considered neural network classifiers with random weights selected from a normal distribution. The optimal parameters of the distribution were determined using Bayesian methods. The performance of such a Bayesian neural network was compared with that of a standard feed-forward hierarchical neural network. The performance of the neural network was further enhanced using ensemble techniques called bagging and boosting. A version of the neural network that used a combination of bagging and boosting was also designed. The performance of all the resulting classifiers was compared using gene expression data from the processed Michigan and Boston data sets available from the CAMDA website. They were also tested on standard benchmarking data available from the UCI machine learning repository. In order to keep the number of input variables and weight parameters of the neural network small, gene selection tools were used to select the most significant genes for the analysis. Five different gene selection methods were implemented and compared. All the neural network versions were implemented using the **R** statistical software environment. We conclude that bagging significantly improved the performance of both feed-forward neural networks and Bayesian neural networks. Boosting improved the accuracy only in a limited number of tests. The robust method for gene selection (GS-Robust) is novel and helped to achieve the lowest error rates among all the methods tried.

were trained and tested with lung cancer gene expression data sets available from the CAMDA website.

## 1. INTRODUCTION

In an attempt to model correlations in gene expression data, we considered several variants of neural network classifiers with random weights sampled from a normal distribution. The effect of using such random weights is to induce correlations among the gene expression measurements. The neural network classifiers

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*Conference '00*, Month 1-2, 2000, City, State.

Many different classification schemes and diagnostic prediction methods have been employed on gene expression data from cancers and the resulting classifications have been correlated to various factors [1-4]. In particular, researchers have made sub-classifications of adenocarcinoma into subgroups that correlated with the degree of tumor differentiation (referred to as “stage” of tumor) [5, 6]. The Boston and Michigan data sets from CAMDA have made available gene expression values for a large number of samples from three different stages of tumors (as well as samples from non-disease patients).

Analysis of gene expression data is challenging because the data are very sparse, redundant, correlated, noisy, and contain high experimental and biological variations. Neural networks have been used to analyze gene expression data [7, 8]. In this paper, we present several new methods for classifying gene expression data from lung cancer patients. Our approach uses Bayesian regularized feed-forward neural networks (as developed by MacKay [9]) and their many variants.

Ensemble neural network methods provide an improved learning paradigm. These methods involve the design of an “ensemble” of neural networks such that collectively their individual abilities. However, methods of classification of gene expression data using ensemble neural network appear to be lacking. The challenge in the use of an ensemble of base classifiers is to decide which classifiers to rely on, or how to combine classifications produced by the classifiers [10]. A necessary and sufficient condition for an ensemble of classifiers to be accurate is that its individual members should be reasonably accurate and diverse [11]. Traditional ensemble methods include *bagging* [12] and *boosting* [13]. Bagging involves bootstrapping and simple aggregation of the classification results. Boosting works by sequentially applying a classification algorithm to reweighted versions of the training data, and then taking a weighted majority vote of the sequence of classifiers thus generated.

In Section 2, we introduce some of the key concepts necessary to describe our methods and results. In Section 3, we briefly describe the implementation, the data sets and the experiments that were performed. In Section 4, we conclude with some discussions.

## 2. KEY CONCEPTS

**Neural Networks:** A neural network implements a non-linear function  $y(x, w)$  where  $y$  is the output of the function for input  $x$  and network parameters (or weights)  $w$ . Given a training set, i.e., set of pairs of the form  $\langle x_i, y_i \rangle, i = 1, \dots, N$ , the neural network can be trained to model the given data as closely as possible. In other words, it is possible to determine the weight vector  $w$  that will best describe the given training data. The training procedure is an optimization procedure that involves minimizing an appropriate error function. Once the optimal weight vector is determined, the neural network acts as a classification or

regression tool, depending on whether the output is from a discrete or continuous set of values.

Neural networks have been used to model gene expression data, where the output function may represent some medical or biological event such as the recurrence of a disease or the production of an enzyme. The main drawback of such a method is that no correlation is assumed between the components of the weight vector  $w$ . In particular, for modeling gene expression data it is necessary to model the correlations between weight vector components because genes act in concert with a collection of other genes forming gene networks. We overcome this drawback by assuming the simplest correlation model, i.e., that the components of the weight vector are random under a univariate model. Such a mixed model induces correlation amongst gene expressions.

**Bayesian Neural Networks:** In our Bayesian model for neural networks, we treat the weights as random variables. This approach has the effect of inducing correlations among the gene expressions. The optimal weights are obtained as the modes of the posterior densities  $P(w | \langle x_i, y_i \rangle)$ . As such, they are computed by maximizing  $P(w | \langle x_i, y_i \rangle)$ . In this paper, we report on experiments comparing the performance of neural networks to that of its Bayesian counterpart.

**Ensemble Neural Network Techniques:** More recently, it has been shown that the performance of classifiers can be enhanced by using ensemble techniques such as bagging or boosting. Both these techniques are termed as “ensemble” techniques because they effectively correspond to designing multiple classifiers in such a way that their collective performance is better than their individual performance.

**Bagging:** Bagging is an acronym for “bootstrap aggregating” [12]. The idea is to take  $k$  repeated bootstrap samples  $D_1, D_2, \dots, D_k$ , from the data and to design  $k$  classifiers using them as the training sets. For any given test data, all the  $k$  classifiers vote to give a resulting classification. Breiman has noted that neural network classifiers tend to be unstable, and that bagging tends to improve unstable classification methods more than stable ones. In this paper, we report on experiments comparing the performance of neural networks and their Bayesian counterparts when enhanced with bagging.

**Boosting:** Boosting was designed to boost the performance of weak classifiers [13]. As in bagging,  $k$  classifiers are designed. However, the training samples are weighted, with higher weights assigned to misclassified samples. The weights are iteratively modified to minimize expected error over different input distributions. After the classifiers are designed, they are assigned weights based on their performance on the training data. A weighted voting scheme is then used to determine the resulting classification for a given test sample. In this paper, we report on

experiments comparing the performance of neural networks and their Bayesian counterparts when enhanced with boosting.

**Gene Selection Methods:** Due to the limitations of the statistical package, it was necessary to limit the size of the neural network and the number of inputs to it. This was achieved by using gene selection methods. An efficient gene selection method was therefore very essential to apply our methods. Several methods have been described in the literature [15-21], all of which rank the genes in the order of significance. We used the following five methods for our analysis: GS-ANOVA, GS-SAM, GS, GS-Robust, and GS-PCA. In GS-ANOVA, the genes are ranked based on the F-statistic computed using ANOVA. In GS-SAM, the SAM program from the Stanford Genomics Group was used to obtain a ranked list of significant genes. GS-PCA involves doing a principal component analysis on the genes and using the components that contribute a large fraction to the variation.

Two new methods, called GS and GS-Robust, were also used for this purpose. In GS, a statistic based on the sum of square error between classes and the sum of square error within classes was computed. In GS-Robust, a similar statistic was computed based not on the sum of square error, but on the median absolute deviation (*MAD*). More precisely, for a  $k$ -class classification problem, if  $\underline{g}_{ij}$  is the vector of gene expression values for the  $i^{\text{th}}$  gene in the  $j^{\text{th}}$  class, then the GS and the GS-Robust methods gives the following scores for the  $i^{\text{th}}$  gene:

$$GS_i = \frac{\sum_{j=1}^k (\overline{\underline{g}_{ij}} - \overline{\underline{g}_{i..}})^2 / (k-1)}{\sum_{j=1}^k \sum_{l=1}^{n_{ij}} (g_{ijl} - \overline{\underline{g}_{ij}})^2 / (n_{ij} - 1)}, \text{ where}$$

$$\overline{\underline{g}_{ij}} = \text{mean}(g_{ij}), \text{ and}$$

$$\overline{\underline{g}_{i..}} = \text{mean}\{\text{mean}(g_{ij}), j = 1, \dots, k\}, \text{ and}$$

$$GSRobust_i = \frac{MAD[\text{median}(\underline{g}_{i1}), \dots, \text{median}(\underline{g}_{ik})]}{\sum_{j=1}^k MAD(\underline{g}_{ij})}$$

The GS criterion is somewhat similar to, but not equivalent to, the F-ratio selection criterion. GS-Robust is a robust version of the GS criterion using the  $L_1$ -norm to measure deviation.

**k-fold Cross-Validation Method:** This is a standard statistical method to validate the accuracy of the classification results. The data is divided into  $k$  groups and  $k$  separate tests are run. When testing samples from each of the groups, the classifier is trained with the  $k-1$  remaining groups. The error rate is reported after averaging over all the groups. All classifier experiments with one labeled data set were tested using this method.

### 3. EXPERIMENTAL RESULTS

All the neural network classifiers and gene selection methods described in this paper were implemented using the **R** statistical package. We tested with 6 neural network classifiers in all. The first one was a standard feed-forward neural network (nnet); the next two were derived from the first after enhancing with bagging (nnet.bag) and boosting (nnet.boost). The last three were similar to the first three with the difference that they used Bayesian neural networks.

A number of benchmark data sets from the UCI repository were downloaded and tested to evaluate and compare the quality of the neural network classifiers. The Iris, BreastCancer, and the HouseVotes84 data sets were used for this purpose [22]. Table 1 shows the error rates from experiments with the classifiers for the three benchmark data sets. In this table and in all the others that follow, results on error rates are shown as mean  $\pm$  SD of the 5-fold cross-validation error from 10 independent runs.

**Table 1. Error rates from Experiments with Benchmark Data**

NN Type	Benchmark Data Sets		
	Iris	BreastCancer	HouseVotes84
nnet	0.105 $\pm$ 0.048	0.065 $\pm$ 0.009	0.053 $\pm$ 0.009
nnet.bag	0.033 $\pm$ 0.009	0.045 $\pm$ 0.004	0.043 $\pm$ 0.004
nnet.boost	0.030 $\pm$ 0.005	0.064 $\pm$ 0.005	0.045 $\pm$ 0.004
bayesian	0.036 $\pm$ 0.013	0.063 $\pm$ 0.007	0.055 $\pm$ 0.007
bayes.bag	0.027 $\pm$ 0.010	0.047 $\pm$ 0.003	0.045 $\pm$ 0.006
bayes.boost	0.028 $\pm$ 0.003	0.075 $\pm$ 0.006	0.044 $\pm$ 0.005

From the CAMDA data sets we tested the neural network classifiers on the Michigan and Boston processed data sets. These were chosen because the type of microarrays used for both data sets were the same. The classifications were performed using the stage level of the tumor as the prediction. The Michigan and Boston data sets use Stages I, II and III as the labels on samples for their samples from diseased patients. Besides these, the Michigan data set also had some normal samples, corresponding to patients with no tumors. This provided us with two data sets each divided into 4 classes of samples. It may be noted that, in fact, one of the classes was missing in both data sets (Stage II from the Michigan data set, and normal samples from Boston data set).

Tables 2 and 3 show the error rates for the classifiers on the two data sets. As discussed before, we used 5 gene selection methods for selecting genes that were used to train the classifiers. This explains the five columns in the tables. Table 4 below shows results from the cross-validation experiments. In the first set of experiments, the neural network classifiers were trained with the

Michigan data set and tested with the Boston data set. In the second set of experiments, their roles were reversed.

Next, we present our experiments with the gene selection methods. The first four gene selection methods produced explicit list of significant genes, while PCA only provides components that are linear combinations of genes. We inspected the 200 most significant genes reported by the first four methods and counted the amount of overlap in their lists. The sizes of their pairwise overlaps are given in Table 5.

**Table 5. Shown are the sizes of the intersection between pairs of lists of the 200 most significant genes picked by the four gene selection methods.**

	GS-SAM	GS	GS-ANOVA	GS-Robust
GS-SAM	200			
GS	167	200		
GS-ANOVA	179	164	200	
GS-Robust	23	28	20	200

As Table 5 shows, the list of genes selected by GS-Robust turned out to be the most different. In fact, 167 of the genes selected by GS-Robust were not on any of the other three lists. Only eight genes were on all the four lists. The gene names of these genes are as follows: GAPD, MGP, RTVP1, DDXBP1, FGR, FGFR2, TNNC1, and KIAA0140. Table 6 shows the contribution to the total variation of the first 10, 20, 30, and 40 components. We picked the first 40 components because they collectively cover at least 80% of the variation.

**Table 6. Contribution of the PCA components to the total variation.**

PCA Components		First 10	First 20	First 30	First 40
Contribution to the total variation	Boston	0.605	0.699	0.763	0.812
	Michigan	0.443	0.620	0.735	0.815

#### 4. DISCUSSION AND CONCLUSIONS

Neural network classifiers enhanced with bagging exhibited consistently good performance, and were clearly better than the ones without any enhancements. Erratic performance was exhibited by neural network classifiers enhanced with boosting. Also, the bagging variants were much faster than their boosting counterparts. The Bayesian neural networks performed marginally below the corresponding feed-forward variants. It is, however,

significant to note that when the GS-Robust classifier was used, the Bayesian variants outperformed their feed-forward counterparts.

GS-Robust appeared to have excellent ability to select significant genes for neural network classifiers. This method consistently outperformed the other gene selection methods. Roughly speaking, GS-Robust is the non-parametric version of GS-ANOVA and GS.

Even though it is interesting that the set of genes selected by GS-Robust had very small overlap with the other methods (Table 5), the most striking results appear in Tables 2-4. The classifiers with GS-Robust had k-fold cross validation error rates of approximately 24% when tested and trained with the Michigan data set. The corresponding error rates were approximately 15% with the Boston data set. It is significant to note that when we performed cross validation experiments where we trained with one entire data set and tested with the other, the results were unexpected. When the training was done with the Boston data set and tested with the Michigan data set, the error rates went up from 23% to 28% for the Bayesian neural networks enhanced with bagging. In sharp contrast, when we trained the neural network with the Michigan data set and tested with the Boston data set, the error rates were down from 15% to about 3% (which is close to the error rates achieved for the tests with the benchmark data sets).

Do these results say something about the relative quality of the two microarray data sets, or about the different sources of variations in the samples from which the two data sets were obtained? Why was the model generated by the combination of the GS-Robust gene selection method and the neural network classifier particularly effective with one data set and not the other? Further analyses are being pursued to address these questions.

#### 5. ACKNOWLEDGEMENTS

Research was supported in part by NIH Grant P01 DA15027-01.

#### 6. REFERENCES

- [1] Fuller, G., et al., Molecular classification of human diffuse gliomas by multidimensional scaling analysis of gene expression profiles parallels morphology-based classification, correlates with survival, and reveals clinically-relevant novel glioma subsets. *Brain Pathol.*, 2002. 12(1): p. 108-16.
- [2] Nutt, C., et al., Gene expression-based classification of malignant gliomas correlates better with survival than histological classification. *Cancer Res.*, 2003. 63(7): p. 1602-7.

- [3] Ramaswamy, S., et al., Multiclass cancer diagnosis using tumor gene expression signatures. *Proc Natl Acad Sci*, 2001. 98(26): p. 15149-54.
- [4] Su, A.I., et al., Molecular Classification of Human Carcinomas by Use of Gene Expression Signatures. *Cancer Res*, 2001. 61(20): p. 7388-7393.
- [5] Beer, D.G., et al., Gene-expression profiles predict survival of patients with lung adenocarcinoma. *Nature Medicine*, 2002. 8(8): p. 816-24.
- [6] Bhattacharjee, A., et al., Expression profiling reveals distinct adenocarcinoma subclasses. *PNAS*, 2001. 98(24): p. 13790-13795.
- [7] Futschik, M.E., et al., Evolving connectionist systems for knowledge discovery from gene expression data of cancer tissue. *Artificial Intelligence in Medicine*, 2003. 28(2): p. 165-189.
- [8] Peterson, C. and M. Ringner, Analyzing tumor gene expression profiles. *Artificial Intelligence in Medicine*, 2003. 28(1): p. 59-74.
- [9] MacKay, D., A practical Bayesian framework for backpropagation networks. *Neural Computation*, 1992. 4(3): p. 448-72.
- [10] Deitterich, T., Machine learning research: four current directions. *Artif Intell*, 1997. 18(4): p. 97-136.
- [11] Hansen, L. and P. Salamon. Neural network ensembles. in *IEEE Trans. Pattern Analysis and Machine Intelligence*. 1990.
- [12] Breiman, L., Bagging predictors. *Machine Learning J.*, 1996. 24(2): p. 123-40.
- [13] Schapire, R.E., The strength of weak learnability. *Machine Learning J.*, 1990. 5(2): p. 197-227.
- [14] Horton, N., Multiple imputation in practice: comparison of software packages for regression models with missing variables. *American Statistician*, 2001. 55: p. 244-254.
- [15] Cho, J.-H., et al., New gene selection method for classification of cancer subtypes considering within-class variation. *FEBS Letters*, 2003. 551(1-3): p. 3-7.
- [16] Lee, K.E., et al., Gene selection: a Bayesian variable selection approach. *Bioinformatics*, 2003. 19(1): p. 90-97.
- [17] Stolovitzky, G., Gene selection in microarray data: the elephant, the blind men and our algorithms. *Curr Opin in Struct Biology*, 2003. 13(3): p. 370-376.
- [18] Szabo, A., et al., Variable selection and pattern recognition with gene expression data generated by the microarray technology. *Mathematical Biosciences*, 2002. 176(1): p. 71-98.
- [19] Tabus, I., et al., Classification and feature gene selection using the normalized maximum likelihood model for discrete regression. *Signal Processing*, 2003. 83(4): p. 713-727.
- [20] Tibshirani, R., et al., Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc Natl Acad Sci USA*, 2002. 99(10): p. 6567-6572.
- [21] Xiong, M., et al., Feature (Gene) Selection in Gene Expression-Based Tumor Classification. *Mol Gen Metabol*, 2001. 73(3): p. 239-247.
- [22] Blake, C.L. and C.J. Merz, UCI Repository; <http://www.ics.uci.edu/~mllearn/>. 1998.

**Table 2. Error rates from Experiments with the Michigan Data Set**

NN Type	Gene Selection Method				
	GS-ANOVA	GS-SAM	GS	GS-Robust	GS-PCA
nnet	0.2889 ± 0.025	0.2900 ± 0.022	0.2967 ± 0.031	0.2767 ± 0.024	0.288 ± 0.021
nnet.bag	0.2789 ± 0.004	0.2767 ± 0.008	0.2678 ± 0.018	0.2733 ± 0.006	0.278 ± 0.000
nnet.boost	0.2922 ± 0.012	0.2900 ± 0.017	0.2622 ± 0.016	0.2722 ± 0.012	0.282 ± 0.013
bayesian	0.3345 ± 0.048	0.3111 ± 0.046	0.3154 ± 0.036	0.2693 ± 0.030	0.299 ± 0.034
bayes.bag	0.2815 ± 0.008	0.2733 ± 0.014	0.2641 ± 0.021	0.2359 ± 0.017	0.280 ± 0.009
bayes.boost	0.2815 ± 0.012	0.2800 ± 0.015	0.2573 ± 0.019	0.2464 ± 0.015	0.277 ± 0.013

**Table 3. Error rates from Experiments with the Boston Data Set**

NN Type	Gene Selection Method				
	GS-ANOVA	GS-SAM	GS	GS-Robust	GS-PCA
nnet	0.153 ± 0.010	0.150 ± 0.005	0.148 ± 0.006	0.149 ± 0.002	0.150 ± 0.007
nnet.bag	0.148 ± 0.000	0.148 ± 0.000	0.148 ± 0.000	0.148 ± 0.000	0.148 ± 0.000
nnet.boost	0.148 ± 0.000	0.149 ± 0.002	0.148 ± 0.000	0.148 ± 0.000	0.148 ± 0.000
bayesian	0.157 ± 0.016	0.152 ± 0.005	0.145 ± 0.006	0.154 ± 0.014	0.148 ± 0.000
bayes.bag	0.148 ± 0.000	0.149 ± 0.003	0.147 ± 0.002	0.148 ± 0.000	0.148 ± 0.000
bayes.boost	0.147 ± 0.003	0.149 ± 0.005	0.142 ± 0.006	0.149 ± 0.002	0.148 ± 0.000

**Table 4. Error rates from Cross-Validation Experiments.**

Training/ Testing	NN Type	Gene Selection Method				
		GS-ANOVA	GS-SAM	GS	GS-Robust	GS-PCA
<i>Michigan/ Boston</i>	nnet	0.090 ± 0.122	0.055 ± 0.054	0.122 ± 0.257	0.033 ± 0.000	0.142 ± 0.272
	nnet.bag	0.033 ± 0.000	0.033 ± 0.000	0.034 ± 0.003	0.033 ± 0.000	0.035 ± 0.005
	nnet.boost	0.036 ± 0.008	0.055 ± 0.068	0.049 ± 0.037	0.033 ± 0.000	0.054 ± 0.050
	bayesian	0.172 ± 0.309	0.269 ± 0.358	0.405 ± 0.466	0.099 ± 0.126	0.171 ± 0.294
	bayes.bag	0.034 ± 0.003	0.035 ± 0.003	0.057 ± 0.059	0.033 ± 0.003	0.105 ± 0.155
	bayes.boost	0.037 ± 0.007	0.060 ± 0.038	0.138 ± 0.188	0.033 ± 0.000	0.061 ± 0.086
<i>Boston / Michigan</i>	nnet	0.391 ± 0.226	0.250 ± 0.077	0.299 ± 0.154	0.293 ± 0.178	0.221 ± 0.000
	nnet.bag	0.221 ± 0.000	0.221 ± 0.000	0.221 ± 0.000	0.221 ± 0.000	0.221 ± 0.000
	nnet.boost	0.219 ± 0.004	0.222 ± 0.004	0.221 ± 0.000	0.221 ± 0.000	0.221 ± 0.000
	bayesian	0.434 ± 0.245	0.343 ± 0.249	0.380 ± 0.201	0.510 ± 0.336	0.276 ± 0.131
	bayes.bag	0.226 ± 0.015	0.222 ± 0.004	0.307 ± 0.149	0.280 ± 0.167	0.221 ± 0.000
	bayes.boost	0.241 ± 0.042	0.271 ± 0.101	0.399 ± 0.286	0.337 ± 0.206	0.221 ± 0.000