Chapter 15

MICROARRAY DATA ANALYSIS USING NEURAL NETWORK CLASSIFIERS AND GENE SELECTION METHODS

Gaolin Zheng[1], E. Olusegun George[2], Giri Narasimhan[1,3]
[1]School of Computer Science, Florida International University, Miami, FL 33199.
[2]Mathematical Sciences Department, University of Memphis, Memphis, TN 38152.
[3]Corresponding Author.

Abstract: Different research groups have conducted independent gene expression studies on tissue samples from human lung adenocarcinomas [Bhattacharjee et al. 2001; Beer et al. 2002]. In this paper we (a) investigate methods to integrate data obtained from independent studies, (b) experiment with different gene selection methods to find genes that have significantly differential expression among different tumor stages, (c) study the performance of neural network classifiers with correlated weights, and (d) compare the performance of classifiers based on neural networks and its many variants on gene expression data. Raw cell intensity data were preprocessed for our analyses. Affymetrix array comparison spreadsheets were used to extract the overlapping probe sets for the data integration study. We considered neural network classifiers with random weights selected from a univariate normal distribution and optimized using Bayesian methods. The performance of the neural network was further enhanced using ensemble techniques such as bagging and boosting. The performance of all the resulting classifiers was compared using the Michigan and Harvard data sets from the CAMDA website. Three gene selection methods were used to find significant genes that could discriminate between the various stages of lung cancer. Significant genes, which were mined from the Gene Ontology (GO) database using the GoMiner and AmiGO packages, were found to be involved in apoptosis, angiogenesis, and cell growth and differentiation. Neural networks enhanced with bagging exhibited the best performance among all the classifiers we tested.

Key words: Microarray, lung adenocarcinoma, robust multiarray averaging, gene selection. neural network classifiers, gene ontology

1. INTRODUCTION

Human lung cancer is a major public health problem. More recently, different research groups have conducted independent and systematic microarray-based gene expression studies on a large number of human lung cancer tissue samples [Bhattacharjee et al., 2001; Beer et al., 2002]. The objectives of this paper are (a) to investigate methods to integrate data obtained from independent studies, (b) to experiment with different gene selection methods to find genes that have significantly differential expression among different tumor stages, (c) to study the performance of neural network classifiers with correlated weights when applied to human lung adenocarcinoma gene expression data, and (d) to compare the performance of classifiers based on neural networks and its many variants on the same data.

Data integration is necessary because often, different laboratories, possibly using different microarray technologies and different probe designs, carry out independent investigations. The experiments are expensive and tumor tissues are a precious research resource. It is possible to gain more insight by integrating all the information carefully.

Gene selection methods are important in order to identify critical genes that deserve further biological investigations. They also are useful to reduce the size of the computational problem that is faced when handling enormous microarray data sets.

Classifiers for microarray data for lung cancer tissue samples, if efficacious, can be as a clinical tool (a) to decide whether a new lung tissue sample is cancerous or not, (b) to identify the type of lung cancer, (c) to identify the stage and progress of the disease, and (d) to predict prognosis and survival information about the patient. Classifiers also help to model the data and to identify hidden correlations in them.

Once a list of differentially expressed genes is generated from the microarray data, it is important to understand the relationships among the genes in question. The Gene Ontology (GO) Consortium [Ashburner et al., 2000] maintains databases that help to obtain biological and functional annotations of these genes. GO organizes genes into hierarchical categories based on biological process, molecular function and subcellular localization. Two mining tools AmiGO [www.godatabase.org] and GoMiner [Zeeberg et al. 2003] were used in this study to obtain functional annotations of the significant genes. All the experiments were performed with implementations using the R statistical package [www.cran.r-project.org].

2. DATA ANALYSIS

2.1 Preprocessing

For our analysis, we started with Affymetrix raw cell intensity data. Bioconductor Affy package [www.bioconductor.org] was used to read cell intensity files. All the image files were obtained and the chips with remarkable spatial artifacts were removed from the study.

The popular methods to obtain expression values from Affymetrix cell intensity files are MAS 4.0 AvDiff [www.affymetrix.com], MAS 5.0 Signal [www.affymetrix.com], Li and Wong's Model-Based Expression Index (MBEI) [Li et al. 2001], and robust multiarray averaging (RMA) [Irizarry et al. 2003]. RMA uses only background-corrected perfect match (PM) values, followed by probe level normalization and robust multiarray averaging. RMA was the method chosen for this study because it gives the best summary of bias, variance, and model fit [Irizarry et al. 2003].

2.2 Data integration

We used two data sets described by Bhattacharjee et al. [2001] and Beer et al. [2002]. We refer to the two data sets as the Harvard data sets and the Michigan data sets, respectively. The two studies used different types of Affymetrix chips for their experiments. The Michigan study used the HuGeneFL type chips, while the Harvard study used the HG_U95Av2 type chip. Array Comparison Spreadsheet HuGeneFL to Human Genome U95A [www.affymetrix.com/support] was used to obtain a list of probe sets with 5 or more overlaps for the two Affymetrix chip types. Cell intensity files were read into an AffyBatch object. Invariant set normalization was then performed at the probe level for the AffyBatch object followed by RMA to obtain the expression values. Expression values of the selected probe sets were extracted from both Michigan and Harvard data sets and combined after matching their IDs using the Array Comparison Spreadsheet mentioned above.

2.3 Gene selection

We were interested in identifying genes that could discriminate advanced tumor stages from early tumor stages. Analysis of variance (ANOVA), significance analysis of microarrays (SAM) and a robust gene selection method referred to as GS-Robust, proposed by us, were the three gene selection methods employed in this study.

For the ANOVA model on the data from the individual studies, stage, gender and smoking information were used as fixed factors. For the model on the integrated data, stage, gender and smoking information were used as fixed factors, while the study (i.e., Harvard vs. Michigan) was used as a random factor. Genes were ranked based on their P-values.

Significance analysis of microarrays (SAM), developed by Tusher et al. [2001], was also used to identify significant genes from microarray data. It is more accurate (lower false discovery rates) than conventional methods [Singhal et al., 2003].

GS-Robust was proposed by us as a robust variant of the F-ratio used in ANOVA. Like F-ratio, it too is a measure of the ratio of between groups and within group variations. Larger GS-Robust values indicate higher discrimination power. For the ith gene, the GS-Robust statistic is defined by

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

where $\underline{g}_{ij}$ is the vector of gene expression values for the ith gene in the jth class, and k is the total number of classes. Unlike F-ratio, GS-Robust uses median absolute deviation, and substitutes mean with median measures. GS-Robust is, therefore, less sensitive to outliers. A disadvantage of the GS-Robust statistic is that it does not have a standard null distribution. As such statistical significance (p-values) may be evaluated by using a bootstrap or permutation resampling procedure. Another disadvantage of GS-Robust (and also SAM) is that there is no obvious approach to extend it to models with multiple factors. However the degrees of freedom for the statistic are the same for all the genes, we can use this measure to rank the discriminative power of the genes. In this paper, a comparative study was performed on the three gene selection methods mentioned above.

Principal component analysis (PCA), a data reduction method, was also used in this study to select the desirable input features for classification. PCA was performed on the correlation matrix. As is customary, in measurements that have different scales, we used the correlation matrix because of the intrinsic heteroscedastic nature of gene expression. Moreover, although principal components are not scale invariant, the principal components generated from correlation matrices are more tractable and allow for more meaningful comparisons of genes. The principal components contributing to at least 75% of the variation were used for classification.

2.4 Neural network classifiers

A neural network implements a non-linear function *y*(*x*, *w*), where *y* is the output function for input *x* and network parameters (or weights) *w*. Given a training set, i.e., set of pairs of the form $\langle x_i, y_i \rangle, i = 1,\ldots, N$, the neural network can be trained to model the given data as closely as possible, and thereby determine the weight vector *w* that best describes the given data. The training procedure 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. For the sake of comparison, support vector machines (SVM), K nearest neighbor (KNN), and random forest classifiers were also implemented and tested.

Neural networks have been used to model gene expression data, where the output function may represent a medical condition or some clinical or biological event such as the recurrence of a disease or prognosis of certain cancers [Khan et al., 2001; Ando et al., 2002; Mateos et al., 2002; Grey et al., 2003]. However in these papers, the network parameters *w* are assumed fixed deterministic constants.

In such models where the weights are not random, the correlations that exist between outputs are artificially induced through the iterative process of the neural network itself. However, these correlations need to be explicitly incorporated into the model. One way to do this is through weight vector (network parameters). Using random weight components induces correlation among genes, since the posterior weights become correlated and account for the fact that genes act in concert with a collection of other genes forming gene networks. In this paper we assume a simple correlation model, i.e., that components of the weight vector are random under a univariate model.

2.4.1 Bayesian regularization of network weights

In regular neural networks, after initializing the network parameters by choosing randomly from a univariate model, the training set is used to optimize the network parameters. The method can be further improved by determining the parameters of the univariate model using standard Bayesian techniques. This is achieved by choosing the optimal weights as the modes of the posterior probability density functions $P(w \mid \langle x_i, y_i \rangle)$, i.e., by maximizing $P(w \mid \langle x_i, y_i \rangle)$. Here $P(w \mid \langle x_i, y_i \rangle)$ is the posterior probability of network weights given the input data. In this paper, we report on experiments comparing the performance of regular neural networks to that of its Bayesian counterpart using lung cancer gene expression data.

2.4.2 Ensemble techniques

More recently, it has been shown that using ensemble techniques such as bagging and/or boosting can enhance the performance of classifiers. Both these techniques are termed as "ensemble" techniques because they correspond to designing a "committee" of classifiers such that their collective performance surpasses their individual performance.

Bagging: Bagging is an acronym for "bootstrap aggregating" [Breiman 1996]. The idea is to design k data sets (denoted by $D_1, D_2, ..., D_k$) by a process of repeated bootstrap sampling from the original data set, and to design k independent 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 [Breiman 1996], and that bagging tends to improve unstable classification methods more than stable ones. In this paper, we report on experiments comparing the performance of regular neural networks and their Bayesian counterparts with and without bagging.

Boosting: Boosting was designed to boost the performance of weak classifiers [Schapire 1990]. As in bagging, k classifiers are successively designed. Unlike with bagging, the training samples are weighted with all samples having equal weights initially. In successive classifiers, weights are iteratively modified so that higher weights are assigned to samples misclassified in previous classifiers and the expected error over different input distributions is minimized. 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 regular neural networks and their Bayesian counterparts with and without the enhancement of boosting.

2.4.3 K-fold cross-validation

In order to compare the performance of the various classifiers mentioned above, we used the standard statistical method of K-fold cross-validation. According to this method, the data was divided into K groups and K separate tests were run. When testing samples from each of the groups, the classifier was trained with the K-1 remaining groups. The error rate was reported after averaging over all the groups.

2.4.4 Practical issues

When designing classifiers for data sets with two or more categories, the training data set may not be balanced in the sense that the number of samples in each category may not be the same. This may cause a bias in the classifiers that are designed. To address this problem, one could create bootstrap copies of samples from the underrepresented classes until a balance is achieved [Japkowicz 2000], or one could randomly remove samples from the overrepresented classes. The first approach suffers from oversampling. The second approach tends to lose potentially significant information. Choosing the lesser of the two evils, we adopted the first approach to adjust the classifiers. It was not used in our gene selection study.

3. RESULTS AND DISCUSSIONS

3.1 Preprocessing

Five of the chips from the Michigan data set, namely L01, L54, L88, L89, and L90, had remarkable spatial artifacts (Figure 1), and were removed from the study. Data on 81 patients were used in the study, of which 64 patients had stage 1 adenocarcinoma, and 17 had stage 3 adenocarcinoma. Of the 81 individuals, 48 were women and 33 were men. Only eight of the 81 were non-smokers while the rest were smokers. Gene expression values for the 7129 probe sets were generated using RMA.

In the Harvard data set, four chips, namely CL2001032701AA, CL2001032709AA, CL2001032634AA, and CL2001032623AA had remarkable spatial artifacts (Figure 1) and were removed from the study. Expression values from the replicates were averaged. After this step, 76 stage 1 adenocarcinoma tumor samples, 24 stage 2 adenocarcinoma tumor samples, and 10 stage 3 adenocarcinoma tumor samples were used for our analyses. Sixty-five of the patients were women, and 45 samples were men. Only 12 of the 110 were nonsmokers. Expression data for the 12625 probe sets were generated using RMA.

To produce an integrated data set from the two data sets, we chose 3742 probe sets that had five or more overlaps in the two data sets. The overlap information was obtained using the Array Comparison Spreadsheet available on the Affymetrix website [www.affymetrix.com/support]. Corresponding subsets of data (corresponding to the 3742 chosen probe sets) from the Michigan and Harvard studies were also generated for our experiments on individual data sets. The perfect match and mismatch intensities from the subsets were normalized using invariant set separately. Robust multiarray averaging method was applied to the AffyBatch object resulting from invariant set normalization to generate the expression values for the 3742 probe sets.

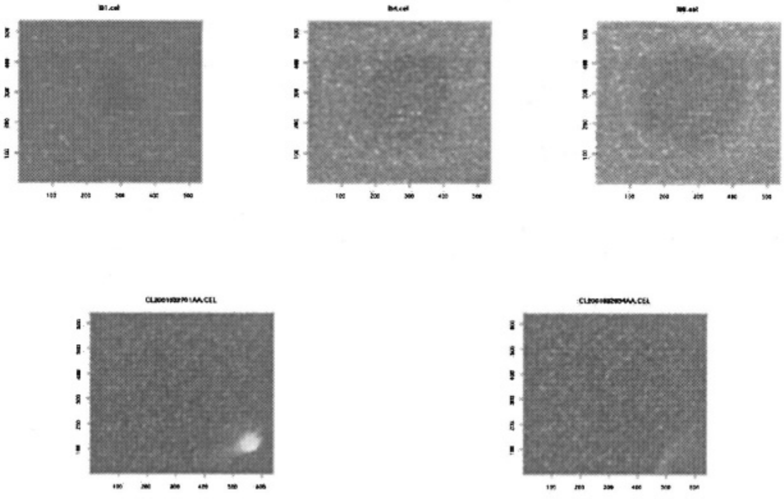

Figure 1. Images of the chips from the Michigan (top row) and Harvard (bottom row) data sets with remarkable spatial artifacts.

3.2 Identifying genes discriminating the tumor stages

Three lists of top 500 genes were generated using multifactor ANOVA, GS-Robust and SAM. Figure 2 shows the intersections of the three groups.

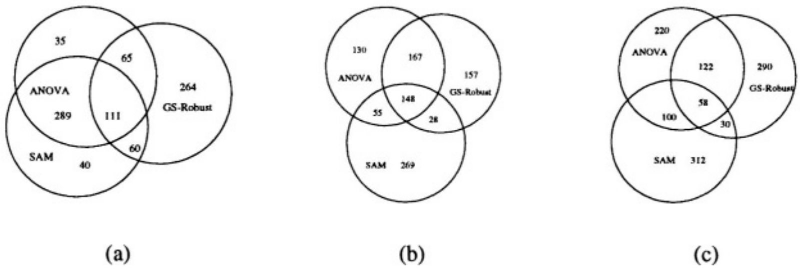

Figure 2. The intersection of the top 500 genes obtained using the three gene selection methods on the (a) Harvard, (b) Michigan, and (c) integrated data sets.

GS-Robust (for the Michigan data set) and the SAM method (for the Harvard data set) selected a list of significant genes that were considerably different from the ones picked by the other methods. For the integrated data set, the overlap in the top 500 lists generated by the three methods was greatly reduced.

3.2.1 Querying significant genes against GO Database

The Gene Ontology (GO) database was queried using GoMiner [Zeeberg et al., 2003]. Significant genes selected using ANOVA were fed into GoMiner and p-values were computed for each GO term based on Fisher's exact tests [Zeeberg et al., 2003] as follows: Let p_1 be the probability that a gene will be flagged under the GO term and p_2 be the probability that it will not. The null hypothesis H_0: $p_1 = p_2$, will be true if genes are flagged under the GO term purely by chance, and there is no significant difference in the two categories. We use the Fisher's exact test to test this hypothesis. This is a conditional test given the sufficient statistics $(n_f/n,(N_f-n_f)/(N-n))$ where n_f is the number of flagged genes under the GO term, n is total number of genes under the GO term, N_f is number of flagged genes on the microarray, and N is the total number of genes on the microarray.

Identifying significant genes: With the help of GoMiner and the Unigene Ids, some of the significant genes (for each of the three sets) and the biological process they are involved in are given below in Table 1.

The analysis of the Michigan data set resulted in five molecular function (MF) GO terms (and their relationships) with p-value less than 0.01 (see Figure 3). A similar analysis of the Harvard data set resulted in 12 MF GO terms (and their relationships), as shown in Figure 4.

Finally, an analysis of the integrated data set gave 6 MF GO terms (and their relationships), as shown below in Figure 5.

Table 1. Significant genes identified from the three data sets.

Study	***Biological Process***	***Induced***	***Repressed***
Michigan	Apoptosis	BIRC2	BBC3, MUC2, PLG
	Angiogenesis	FGF2, POFUT1, VEGF	EPAS
	Cell growth		TGFB
	Cell Cycle	CDC27, CDC7, CDK7, CKS2	
Harvard	Apoptosis	PRKAA1, GSK3B	CASP3, PLG
	Angiogenesis	VEGF	
	DNA Replication	DNTT, SSBP1	
Integrated	Apoptosis		CASP3
	Angiogenesis	VEGFC	
	Cell differentiation	MYF5, PAX6	

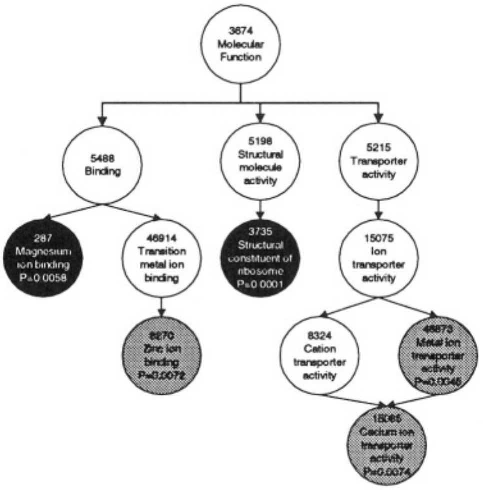

Figure 3. Relationships among significant GO terms identified from the Michigan data set. Note that the significant terms with more overexpressed genes (dark circles), more underexpressed genes (gray circles), and with insignificant changes (white circles) are marked appropriately.

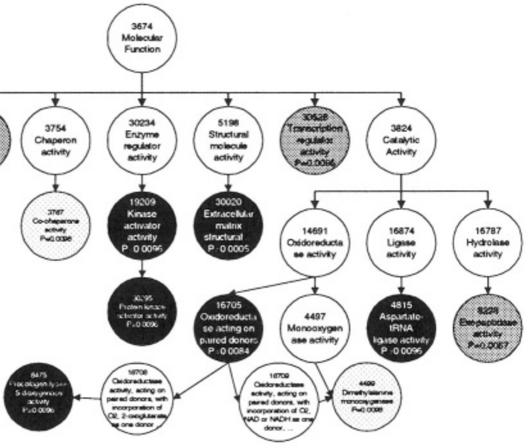
3674 Molecular Function
5488 Binding
19214 Surfactant activity P=0.0096
3754 Chaperon activity
30234 Enzyme regulator activity
5198 Structural molecule activity
30528 Transcription regulator activity P=0.0096
3824 Catalytic Activity
3786 Actin binding P=0.0096
3767 Co-chaperone activity P=0.0096
19209 Kinase activator activity P 0.0096
30020 Extracellular matrix structural P=0.0005
14691 Oxidoreductase activity
16874 Ligase activity
16787 Hydrolase activity
30295 Protein kinase activator activity P=0.0096
16705 Oxidoreductase acting on paired donors P=0.0084
4497 Monooxygenase activity
4815 Aspartate-tRNA ligase activity P=0.0095
8238 Exopeptidase activity P=0.0067
8475 Procollagen-lysine 5-dioxygenase activity P=0.0096
16706 Oxidoreductase activity, acting on paired donors, with incorporation of O2, 2-oxoglutarate as one donor
16709 Oxidoreductase activity, acting on paired donors, with incorporation of O2, NAD or NADH as one donor, ...
4499 Dimethylaniline monooxygenase P=0.0096

Figure 4. The relationships among the significant MF GO terms identified from the Harvard data set. Note that the significant terms with more overexpressed genes (dark circles), more underexpressed genes (gray circles), equal number of overexpressed and underexpressed genes (dotted circles), and with insignificant changes (white circles) are marked appropriately.

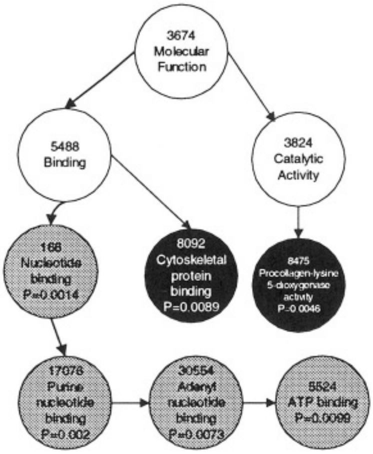
3674 Molecular Function
5488 Binding
3824 Catalytic Activity
166 Nucleotide binding P=0.0014
8092 Cytoskeletal protein binding P=0.0089
8475 Procollagen-lysine 5-dioxygenase activity P=0.0046
17076 Purine nucleotide binding P=0.002
30554 Adenyl nucleotide binding P=0.0073
5524 ATP binding P=0.0099

Figure 5. The relationships among the significant MF GO terms identified from the integrated data set. Note that the significant terms with more overexpressed genes (dark circles), more underexpressed genes (gray circles), and with insignificant changes (white circles) are marked appropriately.

3.3 Classification results

Tables 2, 3, and 4 show the results from our experiments with neural network classifiers using stage information from the Michigan, Harvard, and integrated data sets. Additional classifiers such as SVM, KNN, and random forests were also used for comparisons purposes. In all three sets of experiments, genes were selected using three different ranking schemes and PCA, and the results shown are the mean ± SD of 5-fold cross-validation error from 10 independent runs.

Table 5 shows the results of our cross-validation experiments. When trained with the Michigan data set and tested with the Harvard data set, an accuracy of up to 88% was achieved using bagged neural network classifiers with genes selected using ANOVA. When the roles of the data sets were reversed, an accuracy of only 80% was achieved with most of the gene selection and bagged neural network classifiers. Note that the Michigan data set did not have any data from patients with stage 2 tumors. Only stages 1 and 3 (T1 and T3) were available. Therefore, when we trained with the Michigan data set, all stage 2 data from the Harvard set were left out of the testing. However, when we trained with the Harvard data set, data from all the stages was used (T1, T2 and T3).

Table 2. Experiments on NN classifiers on stage information from the Michigan data set.

	Gene Selection Methods			
	ANOVA	SAM	GS-Robust	GS-PCA
nnet	18.5±3.2%	30.8±6.2%	20.0±2.7%	18.5±2.0%
nnet.bag	16.9±2.7%	23.5±2.6%	18.0±2.1%	14.7±3.1%
nnet.boost	19.7±2.2%	29.2±8.0%	18.8+2.4%	21.2±4.4%
bayesian	15.1± 2.8%	42.3±6.7%	18.3±3.1%	17.2±3.5%
bayes.bag	14.1±2.8%	30.9±2.0%	18.4±2.3%	14.0±2.8%
bayes.boost	17.3±2.4%	38.7±4.1%	19.2±2.4%	17.1±3.0%
SVM	21.4±0.6%	20.8±1.4%	20.5±1.0%	21.5±0.4%
KNN	25.3±0.0%	26.7±0.0%	18.7±0.0%	25.3±0.0%
RandomForest	24.7±0.7%	19.6±0.9%	18.5±1.3%	20.4±1.7%

Table 3. Experiments on NN classifiers on stage information from the Harvard data set.

	Gene Selection Methods			
	ANOVA	SAM	GS-Robust	GS-PCA
nnet	14.6±2.4%	14.0±2.9%	17.7±5.6%	15.1±3.1%
nnet.bag	12.2±1.6%	12.4±1.0%	13.8±2.4%	12.5±3.3%
nnet.boost	14.2±3.0%	15.4±3.1%	18.3±3.3%	19.8±5.7%
bayesian	17.1±2.7%	14.9±2.5%	20.8±3.3%	21.0±4.5%
bayes.bag	12.9±2.2%	13.6±1.8%	17.1±1.8%	18.2±2.1%
bayes.boost	17.1±3.0%	16.1±2.5%	21.3±2.6%	23.3±2.3%
SVM	19.0±0.0%	19.0±0.3%	18.9±0.0%	19.6±0.4%
KNN	21.8±1.3%	22.7±1.0%	13.4±1.3%	29.2±1.5%
RandomForest	17.9±0.7%	17.7±0.1%	18.7±0.1%	20.3±1.1%

Table 4. Experiments on NN classifiers on stage information from the integrated data set.

	Gene Selection Methods			
	ANOVA	SAM	GS-Robust	GS-PCA
nnet	13.1±2.0%	17.4+1.9%	12.4±1.7%	13.6±2.5%
nnet.bag	11.3±1.1%	13.3±1.6%	9.3±1.4%	12.1±0.8%
nnet.boost	13.3±2.9%	18.8±4.8%	11.5±2.1%	15.4±3.9%
bayesian	16.7±2.9%	18.8±4.7%	10.9±2.6%	15.1±2.2%
bayes.bag	14.2±2.4%	24.7±2.4%	10.6±2.2%	14.6±2.6%
bayes.boost	16.7±4.8%	19.3±5.1%	13.1±3.2%	17.1±5.5%
SVM	14.8±0.7%	15.2±0.4%	14.2±0.2%	14.5±0.7%
KNN	18.5±0.5%	15.1±0.9%	10.9±0.6%	18.1±0.9%
RandomForest	14.4±0.7%	14.7±0.6%	14.8±0.9%	14.4±1.1%

Table 5. Cross-validation experiments.

Training Set	Testing Set	Classifier Method	Gene Selection Methods		
			ANOVA	SAM	GS-Robust
Michigan (T1 and T3)	Harvard (T1, T3)	nnet	39.5±4.9%	28.7±4.1%	25.9±1.4%
		nnet.bag	11.6±3.3%	20.0±5.6%	13.9±5.7%
		nnet.boost	17.4±4.7%	22.4±4.7%	21.7±8.9%
		Bayesian	18.8±3.0%	25.0±5.3%	26.6±6.7%
		bayes.bag	12.8±0.1%	21.0±0.5%	20.1±0.6%
		bayes.boost	25.5±0.4%	29.8±1.7%	28.9±1.5%
		SVM	14.7±0.3%	15.2±0.4%	14.2±0.2%
		KNN	18.5±0.5%	15.1±0.9%	18.1±0.9%
		RandomForest	14.4±0.7%	14.7±0.6%	14.4±1.1%
Harvard (T1, T2, T3)	Michigan (T1 and T3)	nnet	27.4+17.8%	21.0±0.5%	21.0±0.6%
		nnet.bag	22.3±4.3%	20.9±0.3%	21.1+0.4%
		nnet.boost	42.3±25.5%	21.0±0.5%	26.7±18.0%
		Bayesian	33.7±17.9%	22.2±3.9%	21.0±0.1%
		bayes.bag	32.3±23.0%	20.9±0.7%	21.2±0.5%
		bayes.boost	33.7±14.3%	21.1±0.4%	21.1±0.7%
		SVM	29.0±0.2%	24.4±0.3%	20.3±0.3%
		KNN	29.9±3.1%	23.6±1.5%	21.9±3.2%
		RandomForest	30.7±5.0%	22.3±2.7%	20.4±1.7%

4. CONCLUSIONS

Bagging consistently and significantly improved the performance of feed-forward neural network classifiers in all our experiments. Since bagging incurs only a small amount of computational overhead, it is feasible to apply this ensemble technique to enhance most classifiers. Boosting, on the other hand, showed erratic behavior. Bayesian neural networks did not show any appreciable improvement over the regular neural networks.

The performance of all the gene selection methods was comparable, with two exceptions. It was not clear why SAM performed poorly only on the Michigan data set. GS-Robust performed particularly well on the integrated data set. We conjecture that GS-Robust was better able to cope with the extra noise that must have been introduced during the data integration process. With gene expression data preprocessed using a robust method such as RMA, the performance of ANOVA and GS-Robust were comparable. Without RMA, GS-Robust outperformed ANOVA (data not shown).

Genes significant for carcinoma stage differentiation were identified from the Michigan, Harvard, and the integrated data sets based on our results from analysis of variance at a significance level of 0.05. Among the significant genes identified from the Michigan data set were three apoptosis activators that were repressed significantly, while one apoptosis inhibitor was induced significantly (Table 1). Interestingly, several cell cycle genes (CDC27, CDC7, CDK7, and CKS2) were induced. In contrast, the cell growth gene TGFB, which is related to lung development, was repressed. Three angiogenesis genes were induced significantly, while only one angiogenesis gene was repressed.

In the advanced stage tumors in the Harvard data set, apoptosis activators, PLG and CASP3, were repressed, while apoptosis inhibitors, PRKAA1 and GSK3B, were induced. Genes involved in DNA replication (DNTT and SSBP1), and the angiogenesis-related gene, VEGF, were induced significantly in the advanced stage tumors of the Harvard data set.

Cell differentiation genes, MYF5 and PAX6, were induced significantly in advanced stage tumors of the integrated data set, as did the angiogenesis-related gene, VEGFC. In contrast, CASP3 (which was also identified from the Harvard data set) was repressed. In summary, genes PLG (apoptosis), CASP3 (apoptosis), and VEGF (angiogenesis) were identified as significant from two independent data sets.

Acknowledgements

Research of E.O.G. & G.N. was supported by NIH Grant P01 DA15027-01.

REFERENCES

Ando, T., M. Suguro, T. Hanai, T. Kobayashi, H. Honda and M. Seto (2002). “Fuzzy neural network applied to gene expression profiling for predicting the prognosis of diffuse large B-cell lymphoma.” Japanese Journal of Cancer Research 93(11): 1207-12.

Ashburner, M., C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin and G. Sherlock (2000). “Gene Ontology: tool for the unification of biology.” Nature Genetics 25: 25 - 29.

Beer, D. G., S. L. R. Kardia, C.-C. Huang, T. J. Giordano, A. M. Levin, D. E. Misek, L. Lin, G. Chen, T. G. Gharib, D. G. Thomas, M. L. Lizyness, R. Kuick, S. H. Hayasaka, J. M. G. Taylor, M. D. Iannettoni, M. B. Orringer and S. Hanash (2002). “Gene-expression profiles predict survival of patients with lung adenocarcinoma.” Nature Medicine 8(8): 816-24.

Bhattacharjee, A., W. G. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. Bueno, M. Gillette, M. Loda, G. Weber, E. J. Mark, E. S. Lander, W. Wong, B. E. Johnson, T. R. Golub, D. J. Sugarbaker and M. Meyerson (2001). “Expression profiling reveals distinct adenocarcinoma subclasses.” PNAS 98(24): 13790–13795.

Breiman, L. (1996). “Bagging predictors.” Machine Learning J. 24(2): 123-40.

Grey, S., S. Dlay, B. Leone, F. Cajone and G. Sherbet (2003). “Prediction of nodal spread of breast cancer by using artificial neural network-based analyses of S100A4, nm23 and steroid receptor expression.” Clin Exp Metastasis 20(6): 507-14.

Irizarry, R., B. Hobbs, F. Collin, Y. Beazer-Barclay, K. Antonellis, U. Scherf and T. Speed (2003). "Exploration, normalization, and summaries of high density oligonucleotide array probe level data." Biostatistics 4(2): 249-264.

Japkowicz, N. (2000). Class imbalance problem: significance and strategies. International Conference on Artificial Intelligence (IC-AI'2000): Special Track on Inductive Learning, Las Vegas.

Khan, J., J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson and P. S. Meltzer (2001). "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks." Nat Med 7(6): 673-9.

Li, C. and W. H. Wong (2001). "Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection." PNAS 98(1): 31-36.

Mateos, A., J. Herrero, J. Tamames and J. Dopazo (2002). Supervised Neural Networks for Clustering Conditions in DNA Array Data after Reducing Noise by Clustering Gene Expression Profiles. Methods of Microarray Data Analysis II. S. M. Lin and K. F. Johnson. Boston, Kluwer Academic Publishers.

Schapire, R. E. (1990). "The strength of weak learnability." Machine Learning J. 5(2): 197-227.

Singhal, S., C. G. Kyvernitis, S. W. Johnson, L. R. Kaiser, M. N. Liebman and S. M. Albelda (2003). "MicroArray Data Simulator For Improved Selection of Differentially Expressed Genes." Cancer Biology & Therapy 2(4): 383-391.

Tusher, V. G., R. Tibshirani and G. Chu (2001). "Significance analysis of microarrays applied to the ionizing radiation response." PNAS 98(9): 5116-5121.

Zeeberg, B. R., W. Feng, G. Wang, M. D. Wang, A. T. Fojo, M. Sunshine, S. Narasimhan, D. W. Kane, W. C. Reinhold, S. Lababidi, K. J. Bussey, J. Riss, J. C. Barrett and J. N. Weinstein (2003). "GoMiner: A Resource for Biological Interpretation of Genomic and Proteomic Data." Genome Biology 4(4): R28.