Computational Intelligent Techniques for Tumor Classification (Using Microarray Gene Expression Data)

Srinivas Mukkamala, Qingzhong Liu, Rajeev Veeraghattam, Andrew H. Sung
Department of Computer Science
New Mexico Tech, Socorro, NM 87801, USA
{srinivas|liu|rajeev|sung}@cs.nmt.edu

Abstract

Computational intelligent techniques can be useful at the diagnosis stage to assist the Oncologist in identifying the malignancy of a tumor. In this paper we perform a t-test for significant gene expression analysis in different dimensions based on molecular profiles from microarray data, and compare several computational intelligent techniques for classification accuracy on selected datasets. Classification accuracy is evaluated with Linear genetic Programs, Multivariate Regression Splines (MARS), Classification and Regression Tress (CART) and Random Forests. We analyze both type of errors false positives and false negatives on four datasets. Linear Genetic Programs and Random forests perform the best for detecting malignancy of different tumors. Our results demonstrate the potential of using learning machines in diagnosis of the malignancy of a tumor. The classifiers used perform the best using the most significant features expect for Prostate cancer dataset.

1 Introduction

Though most cells in our bodies contain the same genes, not all of the genes are used in each cell. Some genes are turned on, or "expressed" when needed. Many genes are used to specify features unique to each type of cell. Microarray technology looks at many genes at once and determines which are expressed in a particular cell type. Using DNA microarray analysis thousands of individual genes can be spotted on a single square inch slide. DNA targets are arrayed onto glass slides (or membranes) and explored with fluorescent or radioactively labeled probes [1]. Obtaining genome-wide expression data from cancerous tissues gives insight into the gene expression variation of various tumor types, thus providing clues for cancer classification of individual samples. One of the key challenges of microarray studies is to derive biological insights from the unprecedented quantities of data on gene expression patterns. Partitioning genes into closely related groups has become an element of practically all analyses of microarray data [2]. But identification of genes faces with many challenges. The main challenge is the overwhelming number of genes compared to the smaller number of available training samples. In machine learning terminology, these data sets have high dimension and small sample size. And many of these genes are irrelevant to the distinction of samples. These irrelevant genes have negative effect on the accuracies of the classifier. Another challenge is that DNA array data contain technical and biological noise. Thus, it is critical to identify a subset of informative genes from a large data that will give higher classification accuracy.

Many methods have been proposed in the past to reduce the dimensionality of gene expression data [3]. Several machine learning techniques have been successfully applied to cancer classification using microarray data [4]. One of the early methods is a hierarchical algorithm developed by Eisen et al. [5]. Other popular algorithms, such as neural networks, K-Nearest Neighbor (KNN), support vector machines, kernel based classifiers, genetic algorithms and Self-Organizing Maps (SOM) are widely applied for tumor classification [3, 6].

In this paper, we extract different dimensional gene data based on t-test and apply Regression Splines (MARS), Classification and Regression Tress (CART) Random Forests and Linear Genetic Programs (LGP) to extracted datasets, and compare the classification accuracy on microarray data.

This paper is organized as follows: section 2 presents gene expression data and t-test analysis to extract key features; section 3 introduces Multivariate Regression Splines (MARS) and section 4 Classification and Regression Tress (CART). Random forests are described in section 5. A brief introduction to Linear Genetic Programs (LGP) is given in section 6. Section 7 describes the datasets and classifier performance. Summary and conclusions are given in section 8.

2 Gene Expression Data Selection

For a given classifier and a training set, the optimality of a gene identification algorithm can be ensured by an exhaustive search over all possible gene subsets. For a data set with n genes, there are 2^n gene subsets. Due to the high dimension of microarrays data, it is impractical to search whole space exhaustively. In our experiments, we choose the significant data based on Student's *t*-test.

2.1 Student’s *t*-test

Student’s *t*-test deals with the problems associated with inference based on “small” samples. The unpaired t method tests the null hypothesis that the population means related to two independent, random samples from an approximately normal distribution are equal [7].

Under the assumption of equal underlying population means, if t < 0, “P(T <= t) one-tail” gives the probability that a value of the t-Statistic would be observed that is more negative than t. If t >=0, “P(T <= t) one-tail” gives the probability that a value of the t-Statistic would be observed that is more positive than t. “t Critical one-tail” gives the cutoff value so that the probability of observing a value of the t-Statistic greater than or equal to “t Critical one-tail” is Alpha.

“P(T <= t) two-tail” gives the probability that a value of the t-Statistic would be observed that is larger in absolute value than t. “P Critical two-tail” gives the cutoff value so that the probability of an observed t-Statistic larger in absolute value than “P Critical two-tail” is Alpha.

Assuming unequal variances, the following formula is used to determine the statistic value *t*:

$$d = \frac{\overline{X}_1 - \overline{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \qquad (1)$$

The following formula is used to calculate the degrees of freedom, df:

$$df = \frac{\left[\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right]^2}{\frac{\left(s_1^2/n_1\right)^2}{n_1 - 1} + \frac{\left(s_2^2/n_2\right)^2}{n_2 - 1}} \qquad (2)$$

Where $\overline{X}_1$ and $\overline{X}_2$ are the sample means, n_1 and n_2 are the sample size, *d* is the Behrens-Welch test statistic evaluated as a Student quantile with *df* freedom using Satterthwaite’s approximation.

$$s_1^2 = \frac{\sum_{j=1}^{n_1} (x_i - \overline{X}_1)^2}{n_1 - 1} \qquad (3)$$

$$s_2^2 = \frac{\sum_{j=1}^{n_2} (x_i - \overline{X}_2)^2}{n_2 - 1} \qquad (4)$$

Accumulated measure of the spread of data about the mean is derived from the following formula:

$$S^2 = \frac{n_1 S_1^2 + n_2 S_2^2}{n_1 + n_2 - 2} \qquad (5)$$

3 MARS

Multivariate Adaptive Regression Splines (MARS) is a nonparametric regression procedure that makes no assumption about the underlying functional relationship between the dependent and independent variables. Instead, MARS constructs this relation from a set of coefficients and basis functions that are entirely “driven” from the data [8].

The method is based on the “divide and conquer” strategy, which partitions the input space into regions, each with its own regression equation. This makes MARS particularly suitable for problems with higher input dimensions, where the curse of dimensionality would likely create problems for other techniques [8,9].

Basis functions: MARS uses two-sided truncated functions of the form as basis functions for linear or nonlinear expansion, which approximates the relationships between the response and predictor variables. A simple example of two basis functions (t-x)+ and (x-t)+[9,11]. Parameter *t* is the knot of the basis functions (defining the "pieces" of the piecewise linear regression); these knots (parameters) are also determined from the data. The "+" signs next to the terms *(t-x)* and *(x-t)* simply denote that only positive results of the respective equations are considered; otherwise the respective functions evaluate to zero.

The MARS Model

The basis functions together with the model parameters (estimated via least squares estimation) are combined to produce the predictions given the inputs. The general MARS

$$y = f(x) = \beta_o + \sum_{m-1}^{M} \beta_m h_m(X) \qquad (6)$$

Where the summation is over the M nonconstant terms in the model, y is predicted as a function of the predictor variables X (and their interactions); this function consists of an intercept parameter (β_o) and the weighted by (βm) sum of one or more basis functions $h_m(X)$ [9].

Model Selection

After implementing the forward stepwise selection of basis functions, a backward procedure is applied in which the model is pruned by removing those basis functions that are associated with the smallest increase in the (least squares) goodness-of-fit. A least squares error function (inverse of goodness-of-fit) is computed. The so-called Generalized Cross Validation error is a measure of the goodness of fit that takes into account not only the residual error but also the model complexity as well. It is given by

$$GCV = \sum_{i=1}^{N} (y_i - f(x_i))^2 /(1 - c/n)^2 \qquad (7)$$

with $C = 1 + cd$

Where N is the number of cases in the data set, d is the effective degrees of freedom, which is equal to the number of independent basis functions. The quantity c is the penalty for adding a basis function. Experiments have shown that the best value for C can be found somewhere in the range 2 < d < 3 [9].

4 CART

CART builds classification and regression trees for predicting continuous dependent variables (regression) and categorical predictor variables (classification) [10].

CART analysis consists of four basic steps:

- The first step consists of tree building, during which a tree is built using recursive splitting of nodes. Each resulting node is assigned a predicted class, based on the distribution of classes in the learning dataset which would occur in that node and the decision cost matrix.
- The second step consists of stopping the tree building process. At this point a "maximal" tree has been produced, which probably greatly overfits the information contained within the learning dataset.
- The third step consists of tree "pruning," which results in the creation of a sequence of simpler and simpler trees, through the cutting off of increasingly important nodes.
- The fourth step consists of optimal tree selection, during which the tree which fits the information in the learning dataset, but does not overfit the information, is selected from among the sequence of pruned trees. Each of these

The decision tree begins with a root node t derived from whichever variable in the feature space minimizes a measure of the impurity of the two sibling nodes. The measure of the impurity at node t, denoted by i(t), is as shown in the following equation:

$$i(t) = -\sum_{f=1}^{k} p(w_j / t) \log p(w_j / t) \qquad (8)$$

Where p(wj | t) is the proportion of patterns xi allocated to class wj at node t. Each non-terminal node is then divided into two further nodes, tL and tR, such that pL , pR are the proportions of entities passed to the new nodes tL, tR respectively. The best division is that which maximizes the difference given in:

$$\Delta i(s,t) = i(t) - pi_L(t_L) - pi_R(t_R) \qquad (9)$$

The decision tree grows by means of the successive sub-divisions until a stage is reached in which there is no significant decrease in the measure of impurity when a further additional division s is implemented. When this stage is reached, the node t is not sub-divided further, and automatically becomes a terminal node. The class wj associated with the terminal node t is that which maximizes the conditional probability p(wj | t). Each of the terminal node describes a data value; each record is classifies into one of the terminal node through the decisions made at the non-terminal node that lead from the root to that leaf [8,10].

5 Random Forests

A random forest is a classifier consisting of a collection of tree structured classifiers **{h(x,Θ_k), k=1, …}** where **{Θ_k}** are independent identically distributed random vectors and each tree casts a unit vote for the most popular class of input **X** .

The common element in random trees is that for the **K**$^{\mathbf{th}}$ tree, a random vector **Θ_k** is generated, independent of the past random vectors **Θ_1,… Θ_{k-1}** but with the same distribution; and a tree is grown using the training set and **Θ_k**, resulting in a classifier **h(x,Θ_k)** where **x** is an input vector. For instance, in bagging the random vector **Θ** is generated as the counts in **N** boxes resulting from **N** darts thrown at random at the boxes, where N is number of examples in the training set. In random split selection **Θ** consists of a number of independent random integers between 1 and K. The nature and dimensionality of **Θ** depends on its use in tree construction. After a large number of trees are generated, they vote for the most popular class [11].

The random forest error rate depends on two things:

- ✓ The ***correlation*** between any two trees in the forest. Increasing the correlation increases the forest error rate.
- ✓ The ***strength*** of each individual tree in the forest. A tree with a low error rate is a strong classifier.

Increasing the strength of the individual trees decreases the forest error rate.

6 Linear Genetic Programming

Linear Genetic Programming (LGP) is a variant of the genetic programming technique that acts on linear genomes **Error! Reference source not found.**. The linear genetic programming technique used for our current experiment is based on machine code level manipulation and evaluation of programs. Its main characteristic, in comparison to tree-based GP, is that the evolvable units are not the expressions of a functional programming language (like LISP); instead, programs of an imperative language (like C) are evolved [12,13,14].

In the automatic induction of machine code by GP, individuals are manipulated directly as binary code in memory and executed directly without passing through an interpreter during fitness calculation. The LGP tournament selection procedure puts the lowest selection pressure on the individuals by allowing only two individuals to participate in a tournament. A copy of the winner replaces the loser of each tournament. The crossover points only occur between instructions. Inside instructions the mutation operation randomly replaces the instruction identifier.

In GP an intron is defined as part of a program that has no influence on the fitness calculation of outputs for all possible inputs. Fitness F of an individual program p is calculated as

$$\begin{aligned} F(p) &= \frac{1}{nm}\sum_{j=1}^{n}\left(o_{ij}^{pred} - o_{ij}^{des}\right)^2 + \frac{w}{n}CE \qquad (10) \\ &= MSE + wMCE \end{aligned}$$

i.e., the mean square error (MSE) between the predicted output (o_{ij}^{pred}) and the desired output (o_{ij}^{des}) for all n training samples and m outputs. The classification error (CE) is defined as the number of misclassifications. Mean classification error (MCE) is added to the fitness function while its contribution is determined by the absolute value of weight (w) [12].

7 Experimental Results an Analysis

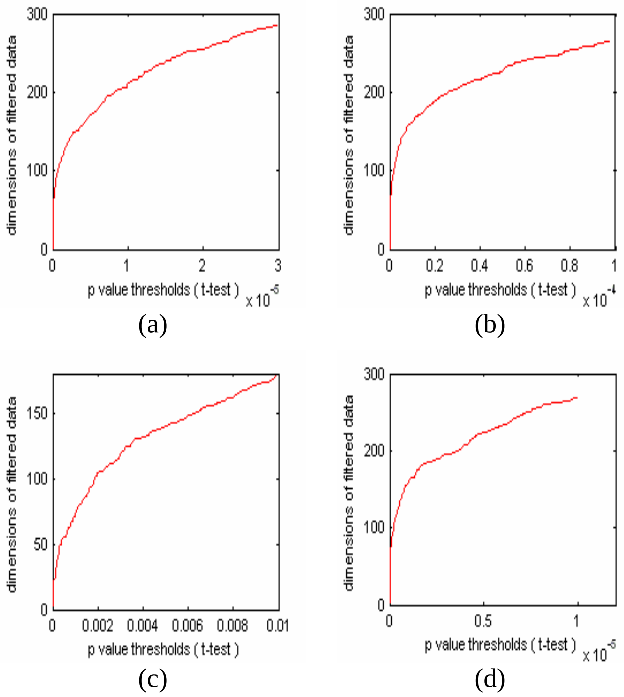
Leukemia, Lymphoma and Prostate cancer data sets are obtained from broad institute [19]. Colon cancer data set is obtained from Princeton gene expression project [20]. Significant gene data obtained from t-test is used for measuring the performance of the classifiers. Fifty percent of the data is used for training and the rest is used for testing. Leukemia data set has (37 training samples and 38 testing samples). Lymphoma data set has (40 training samples and 39 testing samples). Prostate data set has (52 training samples and 52 testing samples). Colon data set has (32 training samples and 32 testing samples).

Data sets used in our experiments.

- Leukemia data set comes from a study of gene expression in two types of acute Leukemia: 48 acute lymphoblastic Leukemia (ALL) samples and 25 acute myeloblastic Leukemia (AML) samples. It was studied in [15].
- Lymphoma data set consists of 58 diffuse large B-cell lymphoma (DLBCL) samples and 19 follicular lymphoma (FL) samples. It was studied in [16]. The data file, lymphoma_8_lbc_fscc2_rn.res, and the class label file, lymphoma_8_lbc_fscc2.cls are used in our experiments for identifying DLBCL and FL.
- Prostate data set in [17] contains 52 prostate tumor samples and 50 non-tumor prostate samples.
- The Colon data set in [18] consists of 40 tumor and 22 normal colon tissues.

7.1 Gene Data Selection Based on t-test

Different thresholds are set in our experiments and different dimension of the most significant gene data are extracted as feature space. Figure 1 (a, b, c, d) shows the dimensions of the filtered significant data according to different p-value thresholds for Leukemia, Lymphoma, Colon, and Prostate data sets.



(a)

(b)

(c)

(d)

Figure 1. The dimensions of filtered significant data for Prostate (a), Leukemia **(b),** Colon **(c),** and

Lymphoma **(d)** data sets, respectively. The p-values of filtered data are smaller than the corresponding thresholds in x-label. Figure 1 (a, b, c, and d) indicates that the significance levels of Prostate**,** Lymphoma and Leukemia data sets are higher than Colon data set.

7.2 Classifiers Performance

We applied MARS, CART, Random forests and LGPs to Leukemia (6,27,53), Lymphoma (7,28,55), Colon (7,15,27,54) and Prostate (6,26,52) cancer data sets, for detecting malignancy of a tumor with different data dimensionalities given in the parenthesis. Classification accuracies are summarized in tables 1 to 4. Table 1 summarizes Leukemia classification accuracies of MARS, CART, LGP and Random forests on 6, 27 and 53. Table 2 summarizes Prostate cancer classification accuracies. Table 3 summarizes Colon cancer classification accuracies. Table 4 summarizes Lymphoma cancer classification accuracies.

Table 1: Leukemia Classification Accuracies

	No of Features					
	6		**27**		**53**	
	Class 1	**Class2**	**Class 1**	**Class2**	**Class 1**	**Class2**
MARS	75	100	83.33	76.92	100	84.62
CART	95.83	92.3	91.66	92.3	91.66	92.3
LGP	**100**	**100**	**100**	**100**	**100**	**100**
Random Forests	91.66	100	95.83	100	95.83	100

Table 2: Prostate Cancer Classification Accuracies

	No of Features					
	6		**26**		**52**	
	Class 1	**Class2**	**Class 1**	**Class2**	**Class 1**	**Class2**
MARS	44	**92.31**	60	**96.15**	88	92.31
CART	64	92.3	60	**96.15**	60	**96.15**
LGP	**92**	**92.31**	**96**	**96.15**	**100**	**96.15**
Random Forests	68	92.3	80	88.46	80	88.46

Table 3: Colon Cancer Classification Accuracies

	No of Features					
	7		**27**		**54**	
	Class 1	**Class2**	**Class 1**	**Class 2**	**Class 1**	**Class2**
MARS	63.64	80	**81.82**	85	81.82	80
CART	36.36	**95**	36.36	**95**	36.36	**95**
LGP	**81.82**	90	**81.82**	90	**81.82**	90
Random Forests	63.63	90	81.81	80	72.72	85

Table 4: Lymphoma Cancer Classification Accuracies

	No of Features					
	7		**28**		**54**	
	Class 1	**Class2**	**Class 1**	**Class2**	**Class 1**	**Class2**
MARS	**100**	44.44	79.31	77.78	96.55	33.33
CART	86.2	88.88	**96.55**	55.55	96.55	55.55
LGP	**100**	**100**	**96.55**	**100**	**100**	**100**
Random Forests	89.65	**100**	**96.55**	88.88	96.55	88.88

Detection rates and false alarms are evaluated for the cancer data sets, and the obtained results are used to form the ROC curves. The point (0,1) is the perfect classifier, since it classifies all positive cases and negative cases correctly. Thus an ideal system will initiate by identifying all the positive examples and so the curve will rise to (0,1) immediately, having a zero rate of false positives, and then continue along to (1,1).

In each of these ROC plots, the x-axis is the false positive rate, calculated as the percentage of normal considered as malignancy; the y-axis is the detection rate, calculated as the percentage of malignancy. A data point in the upper left corner corresponds to optimal high performance, i.e, high detection rate with low false alarm rate. Classification accuracies of the best feature set for different cancer classifications are given in Figures 2, 3, 4, and 5. Figure 2 summarizes the classification performance of classifiers for Leukemia cancer dataset using 6 features. Figure 3 summarizes the classification performance of classifiers for Prostate cancer dataset using 52 features. Figure 4 summarizes the classification performance of classifiers for Colon cancer dataset using 27 features. Figure 5 summarizes the classification performance of classifiers for Lymphoma cancer dataset using 54 features. LGP performed the best for all the datasets with different feature dimensionalities.

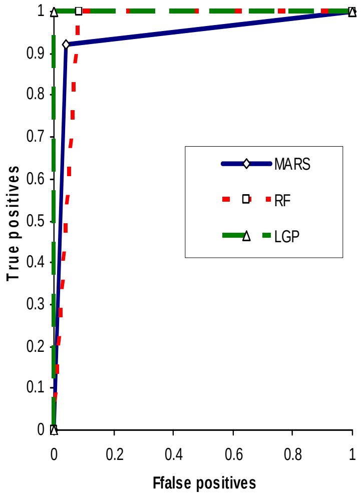


Figure 2. Classifiers Performance on Leukemia Dataset Using 6 Features

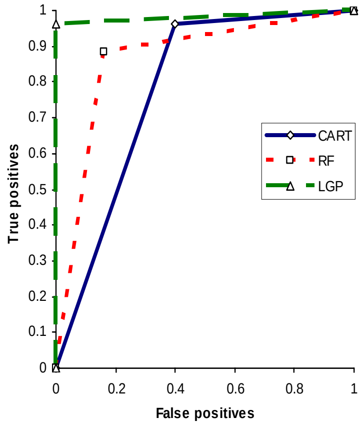


Figure 3. Classifiers Performance on Prostate Dataset Using 52 Features

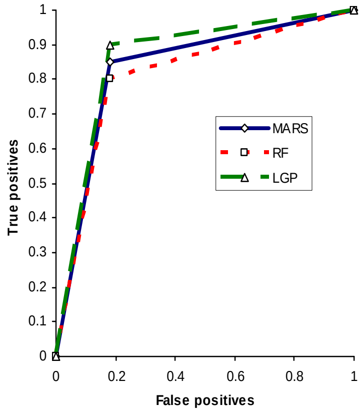


Figure 4. Classifiers Performance on Colon Dataset Using 27 Features

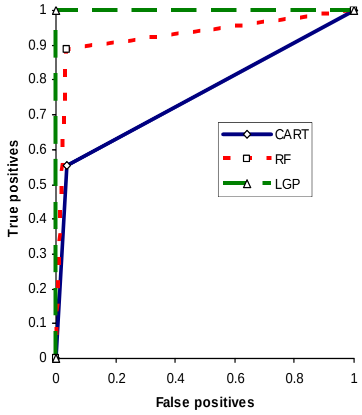


Figure 5. Classifiers Performance on Lymphoma Dataset Using 27 Features

Comparison of classification accuracies of most significant features based on t-test is given in Figures 6-9. Performance results of MARS on most significant features of Leukemia cancer dataset are summarized in Figure 6. Performance results of LGP on most significant features of Prostate cancer dataset are summarized in Figure 7. Performance results of CART on most significant features of Lymphoma cancer dataset are summarized in Figure 8. Performance results of Random forests on most significant features of Colon cancer dataset are summarized in Figure 9.

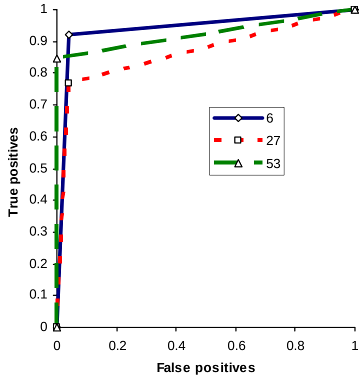


Figure 6. Performance of MARS on Most Significant Features of Leukemia Cancer Dataset

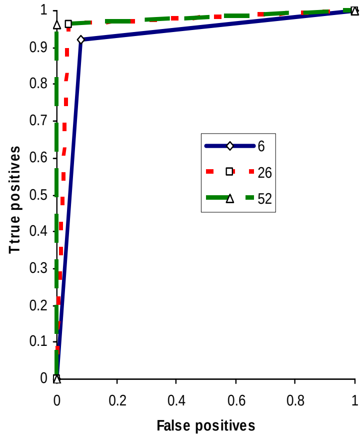


Figure 7. Performance of LGP on Most Significant Features of Prostate Cancer Dataset

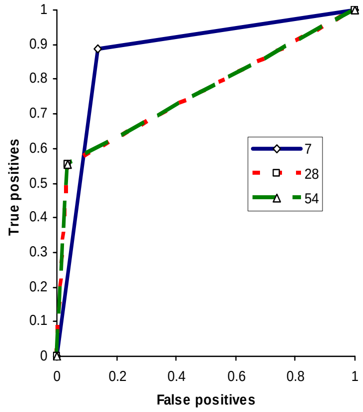


Figure 8. Performance of CART on Most Significant Features of Lymphoma Cancer Dataset

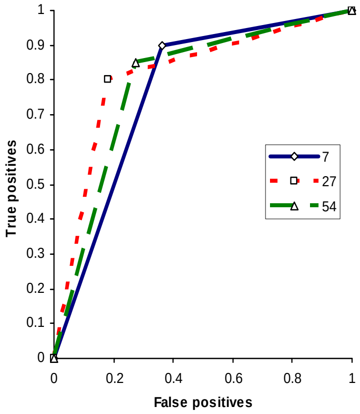


Figure 9. Performance of Random Forests on Most Significant Features of Colon Cancer Dataset

8 Summary and Future Work

Although the performance of the four methods used is comparable in all datasets, we found that linear genetic programs and Random Trees achieved consistently the best results. MARS performs very closely to CART.

LGP performs the best using 6 features for Leukemia dataset and 7 features for Colon and Lymphoma cancer datasets. For Prostate cancer dataset LGP performs the best using 52 features.

The classifiers used in this paper showed comparable or better performance in some cases when compared to the ones reported [artificial neural networks, clustering, support vector machines, etc] in the literature using the same datasets. Our results demonstrate the potential of using learning machines in diagnosis of malignancy of a tumor. As a future work we plan to use large datasets of patients. As more inputs are added, feature selection will have to follow a more stringent scrutiny.

Acknowledgements

Partial support for this research received from ICASA (Institute for Complex Additive Systems Analysis, a division of New Mexico Tech) is gratefully acknowledged.

References

[1] P. Brown, D. Botstein, “Exploring the New World of the Genome with DNA Microarrays”, Nature Genetics Supplement, Vol. 21, pp. 33-37, 1999.

[2] J. Quackenbush, “Computational Analysis of Microarray Data”, Nature Rev. Genteics, Vol. 2, pp. 418-427, 2001.

[3] S. Dudoit, J. Fridlyand, T. Speed, “Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data”, J. Am. Statistical Assoc., Vol. 97, pp. 77-87, 2002.

[4] C. Peterson, M. Ringner, “Analysis Tumor Gene Expression Profiles”, Artificial Intelligence in Medicine, Vol. 28, no. 1, pp. 59-74, 2003.

[5] M. Eisen, P. Spellman, P. Brown, D. Botstein, “Cluster Analysis and Display of Genome-Wide Expression Patterns”, Proc. Nat’l Acad. Sci. USA, Vol. 95, pp. 14863-14868, 1998.

[6] P. Tamyo et al. “Interpreting Patterns of Gene Expression with Self-Organizing Maps: Methods and Application to Hematopoietic Differentiation”, Proc. Nat’l Acad. Sci. USA, Vol. 96, pp. 2907-2912, 1999.

[7] P. Armitage, G. Berry, Statistical Methods in Medical Research, Blackwell 1994.

[8] Salford Systems. TreeNet, CART, MARS, Random Forests Manual.

[9] T. Hastie, R. Tibshirani, and J. H. Friedman, The elements of statistical learning: Data mining, inference, and prediction. Springer, 2001.

[10] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and regression trees. Wadsworth and Brooks/Cole Advanced Books and Software, 1986.

[11] L. Breiman. Random Forests. Journal of Machine Learning, Vol. 45, pp. 5-32, 2001.

[12] J. R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, Cambridge, MA: The MIT Press, 1992.

[13] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Reading, MA: Addison-Wesley, 1989.

[14] AIM Learning Technology, http://www.aimlearning.com.

[15] T. Golub et al., “Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression”, Science, Vol. 286, pp. 531-537, 1999.

[16] M. Shipp et al., “Diffuse Large B-Cell Lymphoma Outcome Prediction by Gene Expression Profiling and Supervised Machine Learning”, Nature Medicine, Vol. 8, no. 1, pp. 68-74, 2002.

[17] D. Singh et al., “Gene Expression Correlates of Clinical Prostate Cancer Behavior”, Cancer Cell, Vol. 1, no. 2, pp. 227-235, 2002.

[18] U. Alon et al., “Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays”, Proc. Nat’l Acad. Sci., Vol. 96, pp. 6745-6750, 1999.

[19] http://www.broad.mit.edu/

[20] http://microarray.princetion.edu/oncology.