Extraction of Informative Genes from Microarray Data

Topon Kumar Paul
Department of Frontier Informatics
The University of Tokyo
Chiba 277-8561, Japan
topon@iba.k.u-tokyo.ac.jp

Hitoshi Iba
Department of Frontier Informatics
The University of Tokyo
Chiba 277-8561, Japan
iba@iba.k.u-tokyo.ac.jp

ABSTRACT

Identification of those genes that might anticipate the clinical behavior of different types of cancers is challenging due to availability of a smaller number of patient samples compared to huge number of genes, and the noisy nature of microarray data. After selection of some good genes based on signal-to-noise ratio, unsupervised learning like clustering and supervised learning like k-nearest neighbor (kNN) classifier are widely used in cancer researches to correlate the pathological behavior of cancers with the gene expression levels' differences in cancerous and normal tissues. By applying adaptive searches like Probabilistic Model Building Genetic Algorithm (PMBGA), it may be possible to get a smaller size gene subset that would classify patient samples more accurately than the above methods. In this paper, we propose a new PMBGA based method to extract informative genes from microarray data using Support Vector Machine (SVM) as a classifier. We apply our method to three microarray data sets and present the experimental results. Our method with SVM obtains encouraging results on those data sets as compared with the rank based method using kNN as a classifier.

Categories and Subject Descriptors

I.2.6 [**ARTIFICIAL INTELLIGENCE**]: Learning—*Knowledge acquisition, Parameter learning*; J.3 [**LIFE AND MEDICAL SCIENCES**]: *Biology and genetics, Health*

General Terms

Algorithms, Performance

Keywords

Gene subset selection, probabilistic model building genetic algorithm, support vector machine, classification of cancer data, informative genes, gene expression, weighted fitness, k-nearest neighbor classifier

GECCO'05, June 25–29, 2005, Washington, DC, USA.

1. INTRODUCTION

In modern clinical neuro-oncology, the right and accurate treatment of patients with cancer depends on accurate diagnoses by using a complex combination of clinical and histopathological data. In some cases, this task is difficult or impossible because of atypical clinical presentation or histopathology [22]. Now many researchers are investigating whether gene expression profiling, coupled with class prediction methodology, could be used to classify different types of tumor samples in a manner more objective, explicit and consistent than standard pathology. The hypothesis behind this research is that gene expression levels are affected by a large number of environmental factors, including temperature, stress, light, and other signals, that lead to change in the level of hormones and other signaling substances, and many or all human diseases may be accompanied by specific changes in the expression levels of some genes [23].

Gene expression is the process by which mRNA and eventually protein is synthesized from the DNA template of each gene. mRNA is a single-stranded molecule consisting of four DNA bases tethered to a sugar-phosphate backbone. The portion of each gene that is represented as mRNA is known as *coding sequence* for that gene. Since mRNA is an exact copy of DNA coding regions, genomic analysis at the mRNA level is used as a measure of gene expression. In other words, gene expression level indicates the amount of mRNA produced in a cell during protein synthesis; and is thought to be correlated with the amount of corresponding protein made.

Recent advances in DNA microarray technology allow scientists to measure expression levels of thousands of genes simultaneously and determine whether those genes are active, hyperactive or silent in normal or cancerous tissues. Since these microarray devices generate huge amount of raw data, new analytical methods must be developed to identify those genes that have distinct signatures of expression levels in cancerous tissues over normal or other types of tissues. Widely used technique in cancer research to extract some informative genes from a microarray data set is the ranking of genes based on signal-to-noise (S2N) statistics [9]. The classification of the samples is done by first building a temporary data set by taking the gene expressions of the selected genes and then applying a suitable classifier. In this context, k-nearest neighbors [4, 16, 24, 7], clustering [3, 8], support vector machine [7, 10, 22], etc. have been used. *The problem of this approach of gene selection is that it totally ignores the effects of the selected genes on the performance of the classifier, whereas an optimal selec-*

tion of genes may not be independent of the algorithm to be used to construct the classifier. Moreover, it may be possible to find a smaller size gene subset which will classify samples more accurately. Recently, there have been used evolutionary computation methods to identify informative genes where Golub's weighted voting classifier [9] has been used as a part of evaluation function of a gene subset [19, 20, 6, 1, 11]. Evolutionary computation methods have advantage over ranking based gene selection method because different combinations of genes are evaluated in evolutionary computations through generation of different individuals of a population.

In this paper, we propose a new adaptive search method to extract informative genes from microarray data. Our method belongs to the category Probabilistic Model Building Genetic Algorithm [21], a variant of Genetic Algorithm. We use Support Vector Machine(SVM) [25] as a classifier. SVM is well suited to the analysis of broad patterns of gene expressions from DNA microarray data. It can easily deal with a large number of genes with a smaller number of training patterns. Our objective in this paper is to select a smaller size gene subset that will produce higher classification accuracy. We test our proposed method on three microarray data sets of binary and multi-class classification problems. Our method outperforms rank based gene selection method in respect of classification accuracy. Our contribution in this paper is to introduce a new gene selection method and to present our findings on the microarray data which may help the biologists/medical scientists to select a set of genes which would be biologically more relevant to cancer diagnosis. Since our paper is on the biological applications of genetic and evolutionary computations, we will emphasize more on application details rather than on theoretical analysis.

2. NOTATION AND TERMS

Before we describe our method, we give notations that will be used later. We will use the term individual to mean a gene subset which may be a possible solution of the problem at hand. If a microarray data set of l samples conatins expression levels of n genes, each individual will consist of n random binary variables $\{X_1, X_2, \ldots, X_n\}$. Let $\mathbf{X}$ denote the set of these random variables and $\mathbf{x}$ be the vector of values of $\mathbf{X}$. If a gene i is selected, the corresponding variable X_i will be 1, otherwise it is set to 0. Let $p(x_i, t)$ be the probability of X_i being 1 in a population of individuals at generation t and $q(x_i, t)$ is the marginal distribution of X_i. $p(\mathbf{x}, t)$ and $q(\mathbf{x}, t)$ are the probability and marginal distribution vectors of $\mathbf{X}$ at generation t, N is the number of individuals in a population, M is the number of individuals selected from the population for calculation of marginal probabilities of the variables, and Q is the number of offspring to produce in a generation. We will use the notation x_i^j to denote the value of the variable X_i in j^{th} individual. Other notations and terms (if any) will be described at the places of their use.

3. GENE SELECTION ALGORITHM

Our proposed method of gene selection is based on Probabilistic Model Building Genetic Algorithm (PMBGA). Instead of applying crossover and mutation operators, a PMBGA generates new possible solutions (individuals) by sampling the probability distribution which is calculated from the selected solutions of previous generations. Different PMBGAs assume different structures of variables and calculate probability distribution accordingly. A good review on PMBGAs (also known as Estimation of Distribution Algorithms [15]) can be found in [14, 17, 18].

The success of a traditional Genetic Algorithm (GA) depends on the appropriate choice of crossover and mutation operators; similarly, the success of a PMBGA depends on its capability of learning a structure of the variables from the selected individuals. The structure of genes of a microarray data set can be described by a Bayesian network. But learning of a Bayesian network from data is NP-hard problem and it would be virtually impossible to build a network structure containing thousands of genes. On the other hand, if the recombination operators (especially mutation operator) of a GA are not carefully designed, it would be very difficult to generate compact size gene subsets and take much time to calculate classification accuracy (training and test) of bigger size gene subsets. These have motivated us to design a method that would successively reduce the number of genes of different individuals but keep the diversity of the population in different generations. We will call our gene selection method Random Probabilistic Model Building Genetic Algorithm (RPMBGA).

In our algorithm, whether a gene would be selected or not depends on its probability $p(x_i, t)$. Initial population of different gene subsets is generated by setting the probability $p(x_i, t)$ of each gene being selected to 0.5 and applying the following decision rule:

$$x_i^j = \begin{cases} 1 & \text{if } r < p(x_i, t); \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $r \in [0, 1]$ is a random number usually generated by calling the $rand()$ function of programming languages. Let us give an example of generating initial population of four genes in details. Given the initial probability vector $p(\mathbf{x}, 0) =< 0.5, 0.5, 0.5, 0.5 >$, N random vectors are generated. Suppose two of them are $R_1 =< 0.002, 0.69, 0.045, 0.85 >$ and $R_2 =< 0.73, 0.032, 0.45, 0.21 >$. By using decision rule (1), we get two gene subsets as $\{1, 0, 1, 0\}$ and $\{0, 1, 1, 1\}$.

Next, we need to update the probability vector to generate new individuals. In PBIL [2], a member of the group PMBGA, the probability is updated by the weighted average of $p(x_i, t)$ and the marginal distribution of that variable $q(x_i, t)$:

$$p(x_i, t+1) = \alpha p(x_i, t) + (1 - \alpha) q(x_i, t) \quad (2)$$

where $\alpha \in [0, 1]$ is called learning rate which is usually fixed at a value during initialization. In a data set containing smaller number of genes, PBIL may produce good results but in microarray data sets containing huge number of genes, it may not return compact size gene subsets for a fixed value of α. We have performed experiments on different microarray data sets with different values of α but in each run, it terminates with many genes selected. In the research on microarray data, it is assumed that only a few genes anticipate the pathological behavior of cancers. Smaller number of genes will be selected if we can somehow reduce the probability of a gene being selected and can keep the search adaptive. Though theoretical analysis of our method would not be provided in this paper, we achieve this end by incorporating a random variable in (2). So, we update probability

as follows:

$$p(x_i, t+1) = \alpha\beta_i p(x_i, t) + (1-\alpha)(1-\beta_i)q(x_i, t) \quad (3)$$

where $\beta_i \in [0, 1]$ is a random number. For a fixed value of α, $p(x_i, t+1)_{PBIL} \geq p(x_i, t+1)_{RPMBGA}$. Therefore, our method will select smaller number of genes as compared to PBIL. The marginal distribution of X_i is calculated as follows:

$$q(x_i, t) = \frac{\sum_{j=1}^{M} x_i^j}{M} \quad (4)$$

where x_i^j is the value of the variable X_i in j^{th} individual. Our overall algorithm is as follows:

PROCEDURE GeneSelection;
Generate initial population of different gene subsets
Evaluate initial population
WHILE (termination_criteria NOT Satisfied) DO
Select M promising individuals
Calculate marginal distribution using (4)
Update probability vector according to (3)
for i=1 to Q do //Q=number of offspring
for j=1 to n do
$r = rand()$
Generate x_j^i *using decision rule (1)*
Evaluate the newly generated gene subsets
Create new population by combining old and new gene subsets

An example of generating new offspring containing 5 genes is given below:

1. Initial probability vector:

   $$p(\mathbf{x}, 0) =< 0.5, 0.5, 0.5, 0.5, 0.5 > .$$

   Suppose, the initial population contains the following individuals (fitness follows colon):
   (a) 10011:0.59 (b) 11010:0.60 (c) 10001:0.85 (d) 01110:0.75 (e) 00111:0.54

2. Select some individuals based on fitness(b,c,d): 11010:0.60, 10001:0.85 and 01110:0.75.

3. Calculate marginal distribution of each X_i:

   $$q(\mathbf{x}, 1) =< \frac{2}{3}, \frac{2}{3}, \frac{1}{3}, \frac{2}{3}, \frac{1}{3} >$$

4. Generate a random vector:

   $$\beta =< 0.10, 0.25, 0.43, 0.67, 0.90 >$$

5. Update the probability vector using (3)($\alpha = 0.9$):

   $$p(\mathbf{x}, 1) =< 0.1050, 0.1625, 0.2125, 0.3235, 0.4083 >$$

6. Generate a set of random vectors:

   (a) $R_1 =< 0.10, 0.054, 0.7, 0.8, 0.77 >$
   (b) $R_2 =< 0.23, 0.56, 0.20, 0.15, 0.95 >$
   (c) $R_3 =< 0.45, 0.054, 0.17, 0.53, 0.57 >$

7. Generate new offspring by comparing $p(\mathbf{x}, 1)$ and each random vector $R_i (i = 1, 2, 3)$ and applying decision rule (1):
   (a)11000 (b)00110 (c)01100

8. Evaluate new offspring and generate new population by combining old population and new offspring.

3.1 Evaluation of a Gene Subset

A gene subset (individual) is evaluated by its accuracy on the training data and the number of genes selected in it. Usually, the value of the fitness function is used as an evaluation measure. In our method, we calculate the fitness of an individual as follows:

$$fitness(\mathbf{x}) = w * A(\mathbf{x}) + (1-w) * (1 - NGS(\mathbf{x})/n) \quad (5)$$

where $A(\mathbf{x})$ is the accuracy on training data using only the expression values of the selected genes in $\mathbf{x}$, $NGS(\mathbf{x})$ is the number of genes selected in $\mathbf{x}$ and $w \in [0, 1]$ is the assigned weight of accuracy. By (5), we have scalarized the two objectives of gene identification task into one. In our experiments, we give more emphasis on accuracy rather than on number of selected genes. Hence in our experiments, $w > (1-w)$.

4. ACCURACY ESTIMATION BY SUPPORT VECTOR MACHINE

Since the number of available training samples is smaller, we calculate the accuracy on training data through Leave-One-Out-Cross-Validation (LOOCV)[12]. In LOOCV, one sample from the training set is excluded, and rest of the training samples are used to build the classifier. Then the classifier is used to predict the class of the left out one, and this is repeated for each sample in the training set. The LOOCV estimate of accuracy is the overall number of correct classifications, divided by the number of samples in the training set. After completion of one generation, a classifier is built taking the gene expression values of the selected genes from training data, and then the classes of the test samples are predicted one by one by taking gene expressions of the selected genes from test data.

We use Support Vector Machine (SVM) as a classifier. SVM is a supervised learning technique first discussed by Vladimir Vapnik [25]. An SVM is a maximum-margin hyperplane that lies in some space. Given training examples labeled as either "+1" or "-1", a maximum-margin hyperplane splits the "+1" and "-1" training examples such that the distance from the closest examples (the margin) to the hyperplane is maximized. The use of the maximum-margin hyperplane is motivated by statistical learning theory, which provides a probabilistic test error bound which is minimized when the margin is maximized.

Suppose, the class of each training vector $\mathbf{x}_i \in R^n, i = 1, 2, \ldots, l$ is labeled as $y_i \in \{+1, -1\}$. (Readers should not confuse this $\mathbf{x}_i$ with the x_i in gene selection which takes a binary value. Here $\mathbf{x}_i$ is a vector of gene expressions.) The SVM separates the training vectors in a ϕ-mapped space (possibly of infinite dimension) with an error cost $C > 0$:

$$\min_{\mathbf{w}, b, \xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l} \xi_i$$

subject to

$$y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad (6)$$

$$\xi_i \geq 0, i = 1, \ldots, l \ .$$

Due to high dimensionality of the vector variable $\mathbf{w}$, usually

(6) is solved through its Lagrangian dual problem:

$$\min_{\alpha} F(\alpha) = \frac{1}{2}\alpha^T \mathbf{Q}\alpha - \mathbf{e}^T\alpha$$

subject to

$$0 \le \alpha_i \le C, i = 1, \ldots, l \ , \tag{7}$$

$$\mathbf{y}^T\alpha = 0 \ ,$$

where $Q_{ij} \equiv y_i y_j \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ and $\mathbf{e}$ is the vector of all ones. Here,

$$K(\mathbf{x}_i, \mathbf{x}_j) \equiv \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) \tag{8}$$

is called the kernel function. Some most widely used kernel functions are: the polynomial kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (a\mathbf{x}_i^T\mathbf{x}_j + r)^d$, the RBF (Radial Basis Function) kernel $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma\|\mathbf{x}_i - \mathbf{x_j}\|^2}$, the linear kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T\mathbf{x}_j$. By definition of (8), the matrix $\mathbf{Q}$ is symmetric and positive definite (PSD). After (7) is solved, $\mathbf{w} = \sum_{i=1}^{l} y_i\alpha_i\phi(\mathbf{x}_i)$; so, the decision function for any test vector $\mathbf{x}$ is

$$sign(\sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b) \tag{9}$$

where b is calculated through primal-dual relationship. After successful training of SVM, most α_i is zero, and the training patterns with non-zero weights are called support vector, and those with strict inequality $0 < \alpha_i < C$ are marginal support vectors. Many resources on SVM, including computer implementations can be found at `http://www.kernel-machines.org`.

Though the original SVM was intended for binary classification, it has been extended to multiclass problem using 'one-against-all' and 'one-against-one' methods. We will describe only the second method ('one-against-one') of SVM for multiclass classification due its usage in our experiments. In 'one-against-one' method [13], $k(k-1)/2$ classifiers (k= number of classes) are constructed where each classifier is trained on data from two classes (i, j) and the class of a test sample $\mathbf{x}$ is predicted by 'winner-takes-all' voting strategy. If the decision function says that $\mathbf{x}$ is in class class i, the vote for the ith class is increased by one, else vote for jth class is increased by one. Then $\mathbf{x}$ is predicted to be in the class which has the highest votes. In the case that two classes have identical votes, the one with lower index is selected. In our experiments, we have used LIBSVM [5] implementation of SVM.

5. EXPERIMENTS

5.1 Preprocessing of Microarray Data

Usually, microarray data files contain Affymetrix's GeneChip software generated gene expression values in scaled average difference units. There is a P, M, or A label associated with each average difference expression value which indicates whether RNA for the gene is present, marginal, or absent, respectively (as determined by the GeneChip software). Files are organized such that each column contains expression levels of different genes in a single sample and each row contains expression levels of a single gene in different samples. These files may have many negative values which are replaced by using a threshold of θ_l and a ceiling of θ_h. If a value is less than θ_l , it is replaced by θ_l; similarly, if a value is greater than θ_h, it is replaced by θ_h. Missing values, if any, are determined by applying kNN method. Then variation filters are applied to exclude those genes which violate $max(g) - min(g) > \Delta$ and $max(g)/min(g) > \Omega$. Different researchers have applied different values of θ_l, θ_h,Δ and Ω for their microarray data. Thereafter, normalization method is applied on the values. In our experiments, we linearly scale all expression values in the range [0,1] due to the requirement of LIBSVM which we have used as SVM source code. If y is a gene expression value of a gene g, the scaled value would be: $\frac{y - min(g)}{max(g) - min(g)}$ where $min(g)$ and $max(g)$ are the minimum and maximum values of gene expressions of g among different samples.

Table 1: Microarray data sets used in experiments

Data Set	#Genes	Classes	#Samples
Lung Carcinoma	3312	5	203
Brain Cancer	4434	2	50
Prostate Cancer	5966	2	102

5.2 Data Sets

For our experiments, we have chosen three microarray data sets of cancer research. These include Lung Carcinoma [4], Brain Cancer [16] and Prostate Cancer [24] data sets. Summary of the data sets are shown in table 1 and the details are given in next subsections. In the table, #Genes denotes the number of genes that are left after preprocessing.

5.2.1 Lung Carcinoma Data Set

The Lung Carcinoma data set [4] contains mRNA expression levels corresponding to 12,600 transcript sequences in 203 lung tumor and normal tumor samples. The 203 samples consist of 139 lung adenocarcinomas (AD), 21 squamous (SQ) cell carcinoma cases, 20 pulmonary carcinoid (COID) tumors and 6 small cell lung cancers (SCLC), as well as 17 normal lung (NL) samples. Negative gene expressions have been replaced by setting $\theta_l = 0$. Using a standard deviation threshold of 50 expression units, only 3312 genes were selected out of 12600. The original data sets are available at `http://research.dfci.harvard.edu/meyersonlab/lungca.html`. We then rescale the data and divide it randomly into mutually exclusive training set consisting of 102 samples and test set of 101 samples. We treat this data set as a 5-class(AD,SQ,COID,SCLC and NL) classification problem. When we calculate overall accuracy, we treat all the 203 samples as training data.

5.2.2 Brain Cancer Data Set

The Brain Cancer data set [16] contains expression levels of 12625 genes of 50 gliomas samples: 28 glioblastomas and 22 anaplastic oligodendrogliomas divided into two subsets of classic and non-classic gliomas. The classic subset contains 14 glioblastomas and 7 anaplastic oligodendrogliomas with classic histology and it is used as a training set to predict the classes of clinically common, histologically non-classic samples consisting of 14 glioblastomas and 15 anaplastic oligodendrogliomas samples. The complete set of data is available at `http://www-genome.wi.mit.edu/cancer/pub/glioma`. After preprocessing of the data with $\theta_l = 20$,$\theta_h = 16000$, $\Delta = 100$ and $\Omega = 3$, only 4434 genes

were left. Then we scale the values as stated above. During calculation of overall LOOCV accuracy, we treat all the 50 samples as training data.

5.2.3 Prostate Cancer Data Set

The initial data set of prostate cancer [24] contains gene expressions profiles which were derived from 52 prostate tumors and 50 non-tumor prostate (normal) samples using oligonucleotide microarrays containing probes for approximately 12,600 genes and ESTs. The independent data set contains 8 normal and 27 tumor prostate samples. Raw data of initial set are avaiable at `http://www-genome.wi.mit.edu/MPR/prostate`. Raw expression values are preprocessed with $\theta_l = 10$,$\theta_h = 16000$, $\Delta = 50$ and $\Omega = 5$. After preprocessing, only 5966 genes were left which are then normalized. Due to unavailability of the independent data set, we divide the initial set into mutually exclusive training and test sets, each containing 50% of the total samples; but during overall accuracy estimation, we treat all 102 samples as a training set.

5.3 Experimental Setup

We generate initial population randomly with the probability of each gene being selected 0.5 (equal probability of being selected or not). The parameters of gene selection algorithm are: population size=100, offspring size=100, maximum number of generation=100, total run=20, $w = 0.75$ and $\alpha = 0.1$. The value of w is chosen to give more emphasize on accuracy rather than on number of selected genes because the ultimate objective of this research is the accurate classification of patient samples. We could increase the population size, but it would take more time to get experimental results. For smaller data sets, we have found that increasing population size does not affect the acquired classification accuracy adversely. Our replacement strategy is CHC in which we combine all the old population and the newly generated offspring and then select the best 100 individuals for the next generation. For calculation of marginal probability, we select the best half of the population. For SVM, we use RBF kernel with values of C=32, $\gamma = 0.0078125$; these values are obtained by applying grid search on the training data as recommended in [5]. Our gene selection algorithm terminates when there is no improvement of the fitness value of the best individual in 10 consecutive generations or maximum number of generations has passed. In each run, after termination of the algorithm, instead of taking the best one that has the highest fitness value, we take all the gene subsets from the population that have the best training accuracy (fitness may be different) and calculate test accuracy of each gene subset by SVM classifier. That is why, the number of gene subsets selected by our method are greater than or equal to 20.

5.4 Experimental Results

In this section, we present the experimental results on the data sets. Some experimental results are available online at `http://www.iba.k.u-tokyo.ac.jp/english/ Supplement/ BioSupplement.html`. Due to space limitations, we will use probe set# (feature#) instead of gene name to indicate a gene. Before the implementation of our gene selection method on the data sets, we run SVM on the data containing single gene expression values to determine whether a single gene exists which can classify all the (training+test) samples without any error. Our findings are summarized in table 2. The results are of one run. We have not found any gene that can classify all the samples (train+test) of each data set without any error. For Lung Carcinoma data set, we find three genes: *32254_at, 34847_s_at and 37588_s_at*; each of them obtains maximum 76.35% classification accuracy while genes *32076_at, 33328_at, 35221_at, 36851_g_at, 37160_at, 38475_at, 39271_at* and *39401_at* each produces the lowest 67.98% accuracy on the same data set. For Brain Cancer, we find the top three genes *1113_at, 35169_at and 40367_at* that obtain respectively 66%, 64% and 64% overall accuracy while two genes *39079_at and 40090_at* each produces the lowest 22% overall accuracy on the data set. Three genes *37720_at, 37639_at and 33674_at* produce 83.33%, 78.43% and 77.45% overall accuracy respectively on Prostate Cancer data set. For this data set, the lowest accuracy (=33.33%) is obtained by the expression values of the gene *39068_at*.

Table 2: Single gene LOOCV accuracy

Data Set	Feature#	Accuracy
Lung Carcinoma	32254_at	76.35
	34847_s_at	76.35
	37588_s_at	76.35
Brain Cancer	1113_at	66.0
	35169_at	64.0
	40367_at	64.0
Prostate Cancer	37720_at	83.33
	37639_at	78.43
	33674_at	77.45

In table 3, the best classification accuracy on training and test data and the number of genes selected are shown. The highest training accuracy on Lung Carcinoma data is 99.02% which is obtained with a gene subset having 80 genes, and the corresponding test accuracy is 93.14%. The maximum test accuracy on this data set is 94.12%, and the corresponding training accuracy and the number of genes selected are 96.08% and 107, respectively. Out of 31 gene subsets selected by our algorithm in 20 runs, the lowest number of genes in a subset is 40 which produces 96.08% and 85.29% training and test accuracy, respectively. For this data set, we also find four more gene subsets having 52, 61,70 and 51 genes which obtain the highest training accuracy(=99.02%) but lower test accuracy (92.16%, 91.18%, 90.20% and 89.22%, respectively). On Brain Cancer data set, we get 981 gene subsets in 20 runs each having 100% training accuracy on 21 training samples but varying test accuracy. Out of these, only 19 gene subsets get the highest 90.48% test accuracy. The top five gene subsets are shown in figure 1. We get a gene subset having minimum three genes {34679_at, 35622_at and 630_at} which produces 80.95% test accuracy and 100% training accuracy. On Prostate Cancer data, we get 177 different gene subsets after 20 runs. Of these, we get 104 gene subsets with 100% training accuracy; out of them, 11 gene subsets produce test accuracy$\geq$ 90.20% and the best gene subset has 24 genes that returns the highest 100% and 94.12% training and test accuracy, respectively. The smallest number of genes selected by our algorithm on this data set is 6 (*31509_at, 34678_at, 34738_at, 37639_at, 38681_at and 40508_at*) that returns 100% training and 82.35% test accuracy. The average experimental results are shown in table 5. In the table, a value of the form $x \pm y$ indicates average value x with standard devia-

Table 3: Best results obtained by our gene selection method RPMBGA

Data set	Best training accuracy	Best test accuracy	Minimum number of selected genes
Lung Carcinoma	99.02 (Test accuracy=93.14) (#Genes=80)	94.12 (Training accuracy=96.08) (#Genes=107)	40 (Training Accuracy=96.08) (Test accuracy=85.29)
Brain Cancer	100.0 (Test accuracy=90.48) (#Genes=6)	90.48 (Training accuracy=100.0) (#Genes=6)	3 (Training accuracy=100.0) (Test accuracy=80.95)
Prostate Cancer	100.0 (Test accuracy=94.12) (#Genes=24)	96.08 (Training accuracy=98.04) (#Genes=13)	6 (Training accuracy=100.0) (Test accuracy=82.35)

Table 4: Gene subsets that produce the best overall classification accuracy

Data Set	Best gene subset with feature#
Lung Carcinoma (67)	1085_s_at, 1252_at, 1295_at, 1313_at, 1368_at, 1488_at, 1822_at, 1823_g_at, 1975_s_at, 2089_s_at, 31559_at, 319_g_at, 32169_at, 32843_s_at, 33027_at, 33272_at, 33274_f_at, 33282_at, 33341_at, 33505_at, 33678_i_at, 33699_at, 33826_at, 33859_at, 34699_at, 35027_at, 35125_at, 35774_r_at, 36096_at, 36162_at, 36703_at, 36889_at, 37000_at, 37021_at, 375_at, 37505_at, 37669_s_at, 37697_s_at, 37722_s_at, 37826_at, 37976_at, 38048_at, 38054_at, 38118_at, 38166_r_at, 38417_at, 38708_at, 39018_at, 39058_at, 39089_at, 39163_at, 39448_r_at, 39561_at, 39581_at, 39649_at, 39660_at, 39720_g_at, 39790_at, 40324_r_at, 41146_at, 41375_at, 41449_at, 41634_at, 41834_g_at,552_at, 903_at, 977_s_at
Brain Cancer (4)	1937_at, 36164_at, 38791_at, 392_g_at
Prostate Cancer (17)	1295_at,1794_at,1818_at, 31389_at, 31444_s_at, 35726_at, 37437_at, 38026_at, 38158_at, 38196_at, 38630_at, 39750_at, 40436_g_at, 40491_at, 41106_at, 41300_s_at, 41867_at

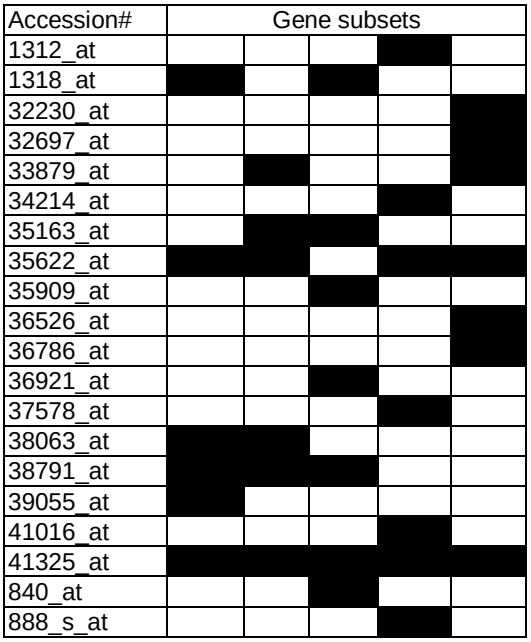

Figure 1: Top five gene subsets that produce 100% training and 90.48% test accuracy on Brain Cancer Data

tion y. The overall results of 20 runs on each data set is shown in table 6. Here we run our algorithm with the settings of the parameters as described before, treat whole data (training+test) as training set and calculate accuracy through leave-one-out-cross-validation. In the table, the number of genes in the subset that produces the best overall accuracy, and the overall accuracy returned by the subset of minimum number of genes are reported in parentheses. The best overall accuracy returned by our method on Lung Carcinoma, Brain Cancer and Prostate Cancer data sets are 98.03%, 96.0% and 98.04% with subsets of 67, 4 and 17 genes, respectively. The gene subsets are shown in table 4. The smallest numbers of genes selected by our method from these data sets are 41, 4 and 12 which return 97.54%, 96.0% and 96.08% overall accuracy, respectively on the data sets. The average overall accuracy and the number of genes selected are also shown in the table. Interestingly, we observe that the gene subset that produces the best overall accuracy on each data set does not include any top ranked gene of table 2, and either overall average or the best accuracy

Table 5: Average accuracy returned and number of genes selected by RPMBGA

Data Set	Training Accuracy	Test Accuracy	#Genes Selected
Lung Carcinoma	97.03 ± 1.25	89.15 ± 3.20	69.03 ± 20.10
Brain Cancer	100 ± 0	72.50 ± 5.90	8.48 ± 4.94
Prostate Cancer	99.03 ± 1.27	84.29 ± 4.57	17.14 ± 7.40

Table 6: LOOCV overall accuracy on data sets

Data	Accuracy		#Genes	
	Average	Best	Average	Best
Lung Carcinoma	97.30± 0.44	98.03 (67)	73.09± 21.83	41 (97.54)
Brain Cancer	93.17± 2.02	96.0 (4)	20.55± 7.51	4 (96.0)
Prostate Cancer	96.62± 0.62	98.04 (17)	48.52± 47.07	12 (96.08)

on each data set is superior to the best accuracy produced by single gene. *From this, we can infer that there may exist some kinds of correlations among the selected genes of the best subset; when we take a single gene from the subset, the correlation breaks down and it does not produce good accuracy on the data set.*

In table 7, we compare our results with those by applying signal-to-noise statistics and k-Nearest Neighbor classifier. In the table, our results are reported in parentheses, and 'NA' indicates 'NOT AVAILABLE'. To make consistent comparison, we have redone signal-to-noise statistics to select the top rank genes and applied kNN with $k = \frac{\#samples}{\#classes}$. Following original literature, we have selected the top 100, 20 and 16 genes from the data sets of Lung Carcinoma, Brain Cancer and Prostate Cancer, respectively. The top genes of Brain Cancer have been selected from the training data only while others have been selected from whole data. The overall accuracy obtained on Lung Carcinoma data containing 203 samples divided into 5 classes is 71.43%. Our method RPMBGA returns 98.03% overall accuracy on the same data using a 67-gene subset. (Note that in the original literature, Bhattacharjee et al. [4] classified a subset of 156 samples of adenocarcinoma and normal lung tissues into 8 types. They used signal-to-noise statistics to select the top 100 genes from the 675-gene set that they had used for hierarchical clustering for input into a LOOCV kNN classifier. They obtained 87% LOOCV accuracy on the data set.) In the case of Brain Cancer data, the accuracy on the 21 training samples of classical gliomas and on 29 test samples of non-classical gliomas by the top 20 genes selected by S2N statistics are 95.24% and 58.62%, respectively. We find 100% classification accuracy on training data and 90.48% accuracy on test data with a 6-gene subset. Finally, comparative results on Prostate Cancer Data are provided. The 16-gene model of S2N produces 93.14% training accuracy whereas our 17-gene subset (the genes of the subset are shown in table 4), obtained by RPMBGA, gets 98.04% training accuracy. In the original literature, the reported test accuracy is 86%. *Due to unavailability of the data of independent test samples, we are unable to present our test accuracy.* In each case, our method gets better accuracy than rank based method. It is our gene selection algorithm, not SVM, that produces the better accuracy on the data sets because SVM can not distinguish between relevant and non-relevant genes; it is used as a classifier to classify data. When SVM is applied on the data without any gene selection, it produces accuracy much lower than the accuracy (training/test/overall) we have reported in this paper. It is very natural that the probability of finding a good gene subset from a collection of many (hundreds) possible solutions is higher than the probability of finding that one from a limited number of solutions. However, subsets with higher number of genes do not produce the best accuracy because there are many irrelevant genes in microarray data which act negatively on the classification accuracy obtained by other genes. Starting from the initial population, our gene selection method successively reduces many of those irrelevant genes and finally terminates with a population having very small number of genes selected in each individual.

Table 7: Experimental results of S2N statistics with kNN classifer (our results are in parentheses)

Data Set	Accuracy(%)			#Genes selected
	Training	Test	Overall	
Lung [4] Carcinoma	NA	NA	71.43 (98.03)	100 (67)
Brain Cancer[16]	95.24 (100.0)	58.62 (90.48)	NA	20 (6)
Prostate Cancer[24]	93.14 (98.04)	86.0 NA	NA	16 (17)

6. SUMMARY AND CONCLUSION

In this paper, we propose and apply our gene selection method RPMBGA to the classification of samples of three microarray data sets. We assume that very few genes are needed to classify cancer samples and smaller size gene subset may provide more insight into the data. To this end, starting from an initial population of different gene subsets having each subset on the average half of genes selected, we reduce the number of selected genes in successive generations by reducing the probability of a gene being selected. As a classifier, we have used SVM which has the capability of handling thousands of genes with a smaller number of training samples. Applying our method to three microarray data sets, we have found that it is possible to get subsets with smaller number of genes that produce better classification accuracy (which may or may not be the best one) as compared to rank based gene selection method with KNN classifier.

During fitness calculation of a gene subset, the two objectives of the problem: selection of minimum number of genes and maximization of classification accuracy have been scalarized in our method, and the smaller one of the two subsets having same training accuracy will always be selected due to implicit penalty on the larger one. Our method is also able to obtain multimodal solutions of the problems. We report all the gene subsets that produce the same classification accuracy on the training data after each run of the algorithm rather than the one having highest fitness value because some of these gene subsets may be highly corre-

lated with distinction of different training and test samples but not biologically relevant to cancer diagnosis; only biologists/medical scientists can say which one of these gene subsets is more biologically relevant to cancer diagnosis.

7. REFERENCES

[1] S. Ando and H. Iba. Classification of gene expression profile using combinatory method of evolutionary computation and machine learning. *Genetic Programming and Evolvable Machines*, 5:145–156, 2004.

[2] S. Baluja. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical Report CMU-CS-94-163, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1994.

[3] A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6:281–297, 1999.

[4] A. Bhattacharjee, W. Richards, J. Stauton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Behesti, R. Buneo, M. Gillete, M. Loda, G. Weber, E. Mark, E. Lander, W. Wong, B. Johnson, T. Golub, D. Sugarbaker, and M. Meyerson. Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. In *Proceedings of National Academy of Science*, volume 98, pages 13790–13795, 2001.

[5] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[6] K. Deb and A. R. Reddy. Reliable classification of two-class cancer data using evolutionary algorithms. *BioSystems*, 72:111–129, 2003.

[7] C. Ding. Tumor tissue classification using support vector machines and K-nearest neighbor method. In *Proceedings of the first Conference on Critical Assessment of Microarray Data Analysis*, 2000.

[8] M. B. Eisen, P. T. Spellman, P. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95:14863–14868, 1998.

[9] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(15):531–537, 1999.

[10] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machine. *Machine Learning*, 46(1-3):389–422, 2002.

[11] Y.-H. Kim, S.-Y. Lee, and B.-R. Moon. A genetic approach for gene selection on microarray expression data. In *Proccedings of the Genetic and Evolutionary Computation Conference(GECCO2004)*, pages 346–355, 2004.

[12] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1995.

[13] U. Kreßel. *Pairwise classification and support vector machine*, pages 255–268. MIT Press, Cambridge, MA, 1999.

[14] P. Larrañaga and J. Lozano. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Optimization*. Kluwer Academic Publishers, Boston, USA, 2001.

[15] H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distribution I. Binary parameters. In *Parallel Problem Solving from Nature-PPSN IV*, Lecture Notes in Computer Science (LNCS) 1411, pages 178–187. Springer-Verlag, Berlin, Germany, 1996.

[16] C. L. Nutt, D. R. Mani, R. A. Betensky, P. Tamayo, J. G. Cairncross, C. Ladd, U. Pohl, C. Hartmann, M. E. McLaughlin, T. T. Batchelor, P. M. Black, A. von Deimling, S. L. Pomeroy, T. R. Golub, and D. N. Louis. Gene expression-based classification of malignant gliomas correlates better with survival than histological classification. *Cancer Research*, 63(7):1602–1607, 2003.

[17] T. Paul and H. Iba. Linear and combinatorial optimizations by estimation of distribution algorithms. In *Proceedings of the 9th MPS Symposium on Evolutionary Computation*, pages 99–106. IPSJ, 2003. Article available at `http://www.iba.k.u-tokyo.ac.jp/english/EDA.htm`.

[18] T. Paul and H. Iba. Reinforcement learning estimation of distribution algorithm. In *Proceedings of GECCO2003*, Lecture Notes in Computer Science (LNCS) 2724, pages 1259–1270. Springer-Verlag, 2003.

[19] T. Paul and H. Iba. Identification of informative genes for molecular classification using probabilistic model building genetic algorithm. In *Proceedings of GECCO2004*, Lecture Notes in Computer Science (LNCS) 3102, pages 414–425. Springer-Verlag, 2004.

[20] T. Paul and H. Iba. Selection of the most useful subset of genes for gene expression-based classification. In *Proceedings of the 2004 Congress on Evolutionary Computation (CEC2004)*, pages 2076–2083, Portland, Oregon, USA, 2004.

[21] M. Pelikan, D. Goldberg, and F. Lobo. A survey of optimizations by building and using probabilistic models. Technical Report, Illigal Report 99018, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, USA, 1999.

[22] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C.-H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J. P. Mesirov, T. Poggio, W. Gerald, M. Loda, E. S. Lander, and T. R. Golub. Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Science*, 98(26):15149–15154, 2001.

[23] M. Schena. *DNA Microarrays*. Oxford University Press, New York, USA, 2000.

[24] D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, E. S. Lander, M. Loda, P. W. Kantoff, T. R. Golub, and W. R. Sellers. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, March 2002.

[25] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York,USA, 1995.