Selection of the Most Useful Subset of Genes for Gene Expression-Based Classification

Topon K. Paul
Graduate School of Frontier Sciences
The University of Tokyo
Kashiwanoha 5-1-5, Kashiwa, Chiba 277-8561
Email: topon@iba.k.u-tokyo.ac.jp

Hitoshi Iba
Graduate School of Frontier Sciences
The University of Tokyo
Kashiwanoha 5-1-5, Kashiwa, Chiba 277-8561
Email: iba@iba.k.u-tokyo.ac.jp

***Abstract*— Recently, there has been a growing interest in classification of patient samples based on gene expressions. Here the classification task is made more difficult by the noisy nature of the data, and by the overwhelming number of genes relative to the number of available training samples in the data set. Moreover, many of these genes are irrelevant for classification and have negative effect on the accuracy and on the required learning time for the classifier. In this paper, we propose a new evolutionary computation method to select the most useful subset of genes for molecular classification. We apply this method to three benchmark data sets and present our unbiased experimental results.**

I. Introduction

DNA microarray offers the ability to measure the levels of expressions of thousands of genes simultaneously. These microarrays consist of specific oligonucleotides or cDNA sequences, each corresponding to a different gene, affixed to a solid surface at very precise location. When an array chip is hybridized to labeled cDNA derived from a particular tissue of interest, it yields simultaneous measurements of mRNA levels in the sample for each gene represented on the chip. Since mRNA levels are thought to correlate roughly with the levels of their translation products, the active molecules of interest, the DNA microarray results can be used as a crude approximation of the protein contents and the state of the sample. Gene expression levels are affected by a number of environmental factors, including temperature, stress, light, and other signals, that lead to change in the level of hormones and other signaling substances. A systemic and computational analysis of this vast amount of data provides information about dynamical changes in functional state of living beings. The hypothesis that many or all human diseases may be accompanied by specific changes in gene expressions has generated much interest among the Bioinformatics community in classification of patient samples based on gene expressions for disease diagnosis and treatment.

Classification based on microarray data faces with many challenges. The main challenge is the overwhelming number of genes compared to the number of available training samples, and many of these genes are not relevant to the distinction of samples. These irrelevant genes have negative effect on the accuracy of the classifier, and increase data acquisition cost as well as learning time. Moreover, different combination of genes may provide similar classification accuracy. Another challenge is that DNA array data contain technical and biological noises. So, development of a reliable classifier based on gene expression levels is getting more attention.

The main target of gene identification task is to maximize the classification accuracy and minimize the number of selected genes. 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. So, it is impractical to search whole space exhaustively, unless n is small. There are two approaches: filter and wrapper approaches [10] for gene subset selection. In filter approach, the data are preprocessed and some top rank genes are selected using a quality metric, independently of the classifier. Though the filter approach is computationally more efficient than wrapper approach, it ignores the effects of selected genes on the performance of the classifier but the selection of optimal gene subset is always dependent on the classifier.

In wrapper approach, the gene subset selection algorithm conducts the search for a good subset by using the classifier itself as a part of evaluation function. The classification algorithm is run on the training set, partitioned into internal training and holdout sets, with different gene subsets. The internal training set is used to estimate the parameters of a classifier, and the holdout set is used to estimate the fitness of a gene subset with that classifier. The gene subset with the highest estimated fitness is chosen as the final set on which the classifier is run. Usually in the final step, the classifier is built using the whole training set and the final gene subset, and then accuracy is estimated on the test set. When number of samples in training data set is smaller, cross-validation technique is used. In k-fold cross-validation, the data D is randomly partitioned into k mutually exclusive subsets, $D_1, D_2, \ldots, D_k$ of approximately equal size. The classifier is trained and tested k times; each time $i(i = 1, 2, \ldots, k)$, it is trained with $D \backslash D_i$ and tested on D_i. When k is equal to the number of samples in the data set, it is called Leave-One-Out-Cross-Validation (LOOCV) [9]. The cross-validation accuracy is the overall number of correctly classified samples, divided by the number of samples in the data. When a classifier is stable for a given data set under k-fold cross-validation, the

variance of the estimated accuracy would be approximately equal to $\frac{a(1-a)}{N}$ [9], where a is the accuracy and N is the number of samples in the data set. A major disadvantage of the wrapper approach is that it requires much computation time.

Numerous search algorithms have been used to find an optimal gene subset. In this paper, we use one Probabilistic Model Building Genetic Algorithm (PMBGA), which generates offspring by sampling the probability distribution calculated from the selected individuals under an assumption about the structure of the problem, as a gene selection algorithm. For classification, we use both Naive-Bayes classifier [4] and the classifier proposed in [6], [21]. The experiments have been done with three well-known data sets. The experimental results show that our proposed algorithm is able to provide better accuracy with selection of smaller number of informative genes as compared to both Multiobjective Evolutionary Algorithm (MOEA) [12] and Population Based Incremental Learning (PBIL)[3].

II. Classifiers and Prediction Strength

A. Naive-Bayes Classifier

Naive-Bayes classifier uses probabilistic approach to assign the class to a sample. That is, it computes the conditional probabilities of different classes given the values of the genes and predicts the class with highest conditional probability. During calculation of conditional probability, it assumes the conditional independence of genes.

Let C denote a class from the set of m classes, $\{c_1, c_2, \ldots, c_m\}$, $\mathbf{X}$ is a sample described by a vector of n genes, i.e., $\mathbf{X} =< X_1, X_2, \ldots, X_n >$; the values of the genes are denoted by the vector $\mathbf{x} =< x_1, x_2, \ldots, x_n >$. Naive-Bayes classifier tries to compute the conditional probability $P(C = c_i|\mathbf{X} = \mathbf{x})$ (or in short $P(c_i|\mathbf{x})$) for all c_i and predicts the class for which this probability is the highest. Using Bayes' rule, we get

$$P(c_i|\mathbf{x}) = \frac{P(\mathbf{x}|c_i)P(c_i)}{P(\mathbf{x})} \ . \tag{1}$$

Since NB classifier assumes the conditional independence of genes, the equation (1) can be rewritten as

$$P(c_i|\mathbf{x}) = \frac{P(x_1|c_i)P(x_2|c_i)\cdots P(x_n|c_i)P(c_i)}{P(x_1, x_2, \ldots, x_n)} \ . \tag{2}$$

The denominator in (2) can be neglected, since for a given sample, it is fixed and has no influence on the ranking of classes. Thus, the final conditional probability takes the following form:

$$P(c_i|\mathbf{x}) \propto P(x_1|c_i)P(x_2|c_i)\cdots P(x_n|c_i)P(c_i) \ . \tag{3}$$

Taking logarithm we get,

$$\ln P(c_i|\mathbf{x}) \propto \ln P(x_1|c_i) + \cdots + \ln P(x_n|c_i) + \ln P(c_i) \ . \tag{4}$$

For a symbolic (nominal) gene,

$$P(x_j|c_i) = \frac{\#(X_j = x_j, C = c_i)}{\#(C = c_i)} \tag{5}$$

where $\#(X_j = x_j, C = c_i)$ is the number of samples that belong to class c_i and gene X_j has the value of x_j, and $\#(C = c_i)$ is the number of samples that belong to class c_i. If a gene value does not occur given some classes, its conditional probability is set to $\frac{1}{2N}$, where N is the number of samples. For a continuous gene, the conditional density is defined as

$$P(x_j|c_i) = \frac{1}{\sqrt{2\pi}\sigma_{ji}} e^{-\frac{(x_j-\mu_{ji})^2}{2\sigma_{ji}^2}} \tag{6}$$

where μ_{ji} and σ_{ji} are the expected value and standard deviation of gene X_j in class c_i. Taking logarithm of equation (6) we get,

$$\ln P(x_j|c_i) = -\frac{1}{2}\ln(2\pi) - \ln\sigma_{ji} - \frac{1}{2}\left(\frac{x_j - \mu_{ji}}{\sigma_{ji}}\right)^2 \tag{7}$$

Since the first term in equation (7) is constant, it can be neglected during calculation of $\ln P(c_i|\mathbf{x})$.

The advantage of the NB classifier is that it is simple and can be applied to multi-class classification problems.

B. Classifier Based on Weighted Voting

Classifier based on weighted voting has been proposed in [6], [21]. We will use the term *Weighted Voting Classifier (WVC)* to mean this classifier. To determine the class of a sample, weighted voting scheme has been used. The vote of each gene is weighted by the correlation of that gene with a particular class. The weight of a gene g is the correlation metric defined as

$$W(g) = \frac{\mu_1^g - \mu_2^g}{\sigma_1^g + \sigma_2^g} \tag{8}$$

where μ_1^g, σ_1^g and μ_2^g, σ_2^g are the mean and standard deviation of the values of gene g in class 1 and 2, respectively. The weighted vote of a gene g for an unknown sample $\mathbf{x}$ is

$$V(g) = W(g)\left(x_g - \frac{\mu_1^g + \mu_2^g}{2}\right) \tag{9}$$

where x_g is the value of gene g in that unknown sample. Then, the class of the sample $\mathbf{x}$ is

$$class(\mathbf{x}) = sign\left\{\sum_{g\in G} V(g)\right\} \tag{10}$$

where G is the set of selected genes. If the computed value is positive, the sample $\mathbf{x}$ belongs to class 1; negative value means $\mathbf{x}$ belongs to class 2.

This classifier is applicable to two-class classification tasks.

C. Prediction Strength

It is always preferable for a classifier to give a confidence measure (prediction strength) of a decision about the class of a test sample. One can define a metric for decision confidence and determine empirically the probability that a decision of any particular confidence value according to that metric is true. By defining a minimum confidence level to classification, one can decrease the number of false positive and false negatives at

the expense of increasing the number of unclassified samples. The combination of a good confidence metric and a good threshold value will result in a low false positive and/or low false negative rate without a concomitant high unclassified samples. The choice of appropriate decision confidence metric depends on the particular classifier and how the classifier is employed.

For Naive-Bayes classifier, the prediction strength metric for two class problems can be defined as the relative log likelihood difference of the winner class [8]. That is, the prediction strength of the classifier for an unknown sample $\mathbf{x}$ is

$$ps = \frac{\ln P(c_{winner}|\mathbf{x}) - \ln P(c_{loser}|\mathbf{x})}{\ln P(c_{winner}|\mathbf{x}) + \ln P(c_{loser}|\mathbf{x})} \ . \qquad (11)$$

In our experiment, we have refrained from employing decision confidence metric for Naive-Bayes classifier due to unavailability of a suitable threshold value.

Golub et al. [6] and Slonim et al. [21] defined the prediction strength for weighted voting classifier as follows:

$$ps = \left| \frac{V_+ - V_-}{V_+ + V_-} \right| \qquad (12)$$

where V_+ and V_- are respectively the absolute values of sum of all positive $V(g)$ and negative $V(g)$ calculated using equation (9).

The classification of an unknown sample is accepted if $ps > \theta$(θ is the prefixed prediction strength threshold), else the sample is classified as undetermined. In our experiment, we consider undetermined samples as misclassified samples.

III. Accuracy Estimation

We use LOOCV procedure during the gene selection phase to estimate the accuracy of the classifier for a given gene subset and a training set. 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. Thereafter, a classifier is built using all the training samples, and it is used to predict the class of all test samples one by one. Final accuracy on the test set is the number of test samples correctly classified by the classifier, divided by the number of test samples. Overall accuracy is estimated by first building the classifier with all training data and the final gene subset, and then predicting the class of all samples (in both training and test sets) one by one. Overall accuracy is the number of samples correctly classified, divided by total number of samples. This kind of accuracy estimation on test set and overall data is unbiased because we have excluded test set during the search for the best gene subset.

IV. Gene Selection Method

The Probabilistic Model Building Genetic Algorithm (PMBGA) [18] has been used as a gene selection method. PMBGA replaces the crossover and mutation operators of traditional evolutionary computations; instead, it uses probabilistic model building and sampling techniques to generate offspring. It explicitly takes into account the problem specific interactions among the variables. In evolutionary computations, the interactions are kept implicitly in mind; whereas in a PMBGA, the interrelations are expressed explicitly through the joint probability distribution associated with the individuals of variables, selected at each generation. The probability distribution is calculated from a database of selected candidate solutions of previous generation. Then, sampling this probability distribution offspring are generated. The flow chart of a PMBGA is shown in figure 1. Since a PMBGA tries to capture the structure of the problem, it is thought to be more efficient than the traditional genetic algorithm. The other name of PMBGA is Estimation of Distribution Algorithm (EDA), which was first introduced in the field of evolutionary computations by Mühlenbein in 1996 [14].

A PMBGA has the follow components: encoding of candidate solutions, objective function, selection of parents, building of a structure, generation of offspring, selection mechanism, and algorithm parameters like population size, number of parents to be selected, etc.

The important steps of the PMBGA are the estimation of probability distribution, and generation of offspring by sampling that distribution. Different kinds of algorithms have been proposed on PMBGA. Some assume the variables in a problem are independent of one another, some consider bivariate dependency, and some multivariate. If the assumption is that variables are independent, the estimation of probability distribution as well as generation of offspring becomes easier. A good review on PMBGA can be found in [11], [15], [16], [17], [19]. For our experiments, we propose another one which is described in the next subsection.

A. Proposed Method

Before the description of our proposed algorithm, let us give some notations. Let $X = \{X_1, X_2, \ldots, X_n\}$ is the set of n binary variables corresponding to n genes in the data set, and $x = \{x_1, x_2, \ldots, x_n\}$ is the set of values of those variables with $x_i (i = 1, \ldots, n)$ being the value of the variable X_i [readers should not confuse this X with that in the classifier, the X in the classifier is a vector of values of genes while that here is a vector of binary variables]. Q is the number of individuals selected from a population for the purpose of reproduction. $p(x_i, t)$ is the probability of variable X_i being 1 in generation t and $M(x_i, t)$ is the marginal distribution of that variable. The joint probability distribution is defined as

$$p(x, t) = \prod_{i=1}^{n} p(x_i, t|pa_i) \qquad (13)$$

where $p(x_i, t|pa_i)$ is the conditional probability of X_i in generation t given the values of the set of parents pa_i. If the variables are independent of one another, the joint probability distribution becomes the product of the probability of each variable $p(x_i, t)$. To select informative genes for molecular

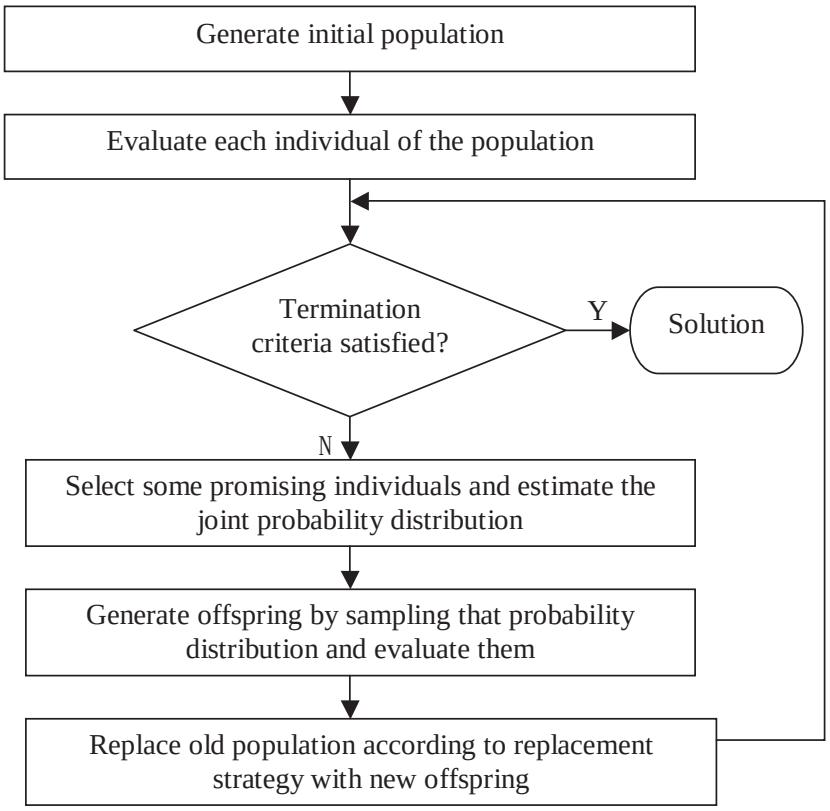

Fig. 1. Flowchart of a PMBGA

classification, we consider that variables are independent. We use binary encoding and probabilistic approach to generate the value of each variable corresponding to a gene in the data set. The initial probability of each variable is set to zero assuming that we don't need any gene for classification. Then, that probability is updated by the weighted average of marginal distribution and the probability of previous generation. That is, the probability of X_i has been updated as

$$p(x_i, t+1) = \alpha p(x_i, t) + (1-\alpha) M(x_i, t) \overline{w}(g_i) \qquad (14)$$

where $\alpha \in [0, 1]$ is called the learning rate, and $\overline{w}(g_i) \in [0, 1]$ is the normalized weight of gene g_i corresponding to X_i in the data set. This weight is the correlation of gene g_i with the classes. This is calculated as follows:

$$\overline{w}(g_i) = \frac{|W(g_i)|}{MAX\{|W(g_1)|, |W(g_2)|, \ldots, |W(g_n)|\}} \qquad (15)$$

where each $W(g_i)$ is calculated according to (8). The marginal distribution of X_i is calculated as follows:

$$M(x_i, t) = \frac{\sum_{j=1}^{Q} \delta_j^i}{Q} \qquad (16)$$

where $\delta_j^i \in \{0, 1\}$ is value of variable X_i in the selected j^{th} individual. By sampling $p(x_i, t+1)$, the value of X_i is generated for the next generation. The steps of our proposed algorithm are as follows:

1) Divide the data into training and test sets, and calculate weight of each gene.
2) Generate initial population, evaluate it, and initialize probability vector.
3) While termination criteria is not satisfied do the following:
   a) Select some promising individuals.
   b) Calculate marginal distribution of each variable and update probability vector according to equation (14).
   c) Generate offspring by sampling that probability vector and evaluate them.
   d) Replace old population with offspring.

Let us give an example of generating an offspring using our method. Suppose, there are 5 genes in a data set with normalized weight vector $\overline{w}(g) = (0.05, 0.1, 0.01, 1.0, 0.2)$, probability vector at generation t is $p(x, t) = (0.1, 0.05, 0.2, 0.5, 0.3)$ and the marginal probability vector calculated from the selected individuals is $M(x, t) = (0.5, 0.1, 0.3, 0.9, 0.5)$. If we set $\alpha = 0.1$, the updated probability vector using equation (14) would be $p(x, t+1) = (0.0325, 0.014, 0.0227, 0.86, 0.12)$. Now generate a vector of random numbers from uniform distribution. Suppose the vector of random numbers is $R = (0.1, 0.2, 0.01, 0.75, 0.3)$. Now comparing each $p(x_i, t)$ with R_i, we get the offspring $(0, 0, 1, 1, 0)$ (output is 1 if $p(x_i, t) \geq R_i$) .

B. Our Proposed Method and PBIL

Population Based Incremental Learning(PBIL), proposed by Baluja [3], was motivated by the idea of combining Genetic Algorithm with Competitive Learning which is often used in training of Artificial Neural Networks. Like our proposed method, PBIL considers binary representation of individuals, start with the initialization of the probability vector and update it at each generation. But the probability of $X_i(i = 1, \ldots, n)$ has been updated as

$$p(x_i, t+1) = \alpha p(x_i, t) + (1-\alpha)M(x_i, t) \ . \qquad (17)$$

The difference between our method and PBIL is that we have combined the weight of each gene during update of the probability vector. The difference between performance of these two algorithms can be verified empirically.

C. Encoding and Fitness Calculation

In our experiments, the individuals in a population are binary-encoded with each bit for each gene. If a bit is '1', it means that the gene is selected in the gene subset; '0' means its absence.

The fitness of an individual has been assigned as the weighted sum of the accuracy and dimensionality of the gene subset corresponding to that individual. It is

$$fitness(X) = w_1 * a(X) + w_2 * (1 - d(X)/n) \qquad (18)$$

where w_1 and w_2 are weights from $[0, 1]$, $a(X)$ is the accuracy of X, $d(X)$ the number of genes selected in X, and n is the total number of genes. This kind of fitness calculation was used in [13].

D. Population Diversity

In gene subset selection, different combinations of genes may produce same classification accuracy. In this sense, we can say that the problem is a multimodal optimization problem. For multimodal optimization, maintaining population diversity is very important. One technique widely used for this purpose is *Sharing*, first introduced by Holland [7]. The premise behind this technique is to reduce the fitness of individuals that have highly similar members within the population. This reward discourages redundant individuals in a domain from reproduction.

The shared fitness of an individual i is given by $f_i^{shared} = \frac{f_i}{m_i}$, where f_i is the raw fitness of that individual, and m_i is the niche count, which defines the amount of overlap of the individual i with the rest of the population. The niche count is calculated by summing up a sharing function over all members of the population: $m_i = \sum_{j=1}^{N} sh(d_{ij})$. The distance d_{ij} represents the distance between individual i and individual j in the population, determined by a similarity metric. In our experiment when two individuals have the same fitness, there can be two possibilities: either they are same in genotype or different. We use genotype similarity to calculate shared fitness, and define

$$sh(d_{ij}) = \begin{cases} 1 & \text{if individuals } i \text{ and } j \text{ have same genotype;} \\ 0 & \text{otherwise.} \end{cases} \qquad (19)$$

We select some top ranks individuals (the number is fixed during experiment) that have higher shared fitness for calculation of marginal probability distribution. During the regeneration steps, we combine old population and offspring to generate new population. We take the best individuals from the combined population. During selection of individual, if both individuals have same shared fitness but different genotype, we take that one which has higher average gene weight of the selected genes.

V. Related Works in Molecular Classification Using Evolutionary Algorithms

Previously, Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [5], Multi-objective Evolutionary Algorithm(MOEA) [12] and Parallel Genetic Algorithm [13] with weighted voting classifier have been used for the selection of informative genes responsible for the classification of the DNA microarray data.

In the optimization using NSGA-II, three objectives have been identified. One objective is to minimize the size of gene subset; the other two are the minimization of mismatches in the training and test samples, respectively. The number of mismatches in the training set is calculated using LOOCV procedure, and that in the test set is calculated by first building a classifier with the training data and the gene subset and then predicting the class of the test samples using that classifier. Due to inclusion of the third objective, the test set is, in reality, has been used as a part of training process and is not independent. Thus the reported 100% classification accuracy for the three cancer data sets is not generalized accuracy, rather a biased accuracy on available data. In supervised learning, the final classifier should be evaluated on an independent test set that has not been used in any way in training or in model selection [10], [20].

In the work using MOEA, also three objectives have been used; the first and the second objectives are the same as above, the third object is the difference in error rate among classes, and it has been used to avoid bias due to unbalanced test patterns in different classes. For decision making, these three objectives have been aggregated. The final accuracy presented is the accuracy on the training set (probably on the whole data) using LOOCV procedure. It is not clear how the available samples are partitioned into training and test sets, and why no accuracy on the test set has been reported.

In the gene subset selection using parallel genetic algorithm, the first two objectives are used and combined into a single one by weighted sum, and the accuracy on the training and test sets (if available) have been reported. In our work, we follow this kind of fitness calculation.

VI. Experiments

A. Data Sets

We evaluate our method on three cancer data sets: Leukemia, Lymphoma and Colon. The data sets are described in table I. The first and the second data sets

TABLE I

DATA SETS USED IN THE EXPERIMENTS

Data Set	Total Genes	Classes	Total Samples
Leukemia	7129	ALL	47
		AML	25
Lymphoma	4026	DLBCL	42
		Others	54
Colon	2000	Normal	22
		Cancer	40

need some preprocessing; we have downloaded the preprocessed data (Leukemia and Lymphoma data sets) from http://www.iitk.ac.in/kangal/bioinformatics.

a) Leukemia Data Set: This is a collection of gene expressions of 7129 genes of 72 leukemia samples reported by Golub et al. [6]. The data set is divided into an initial training set of 27 samples of Acute Lymphoblastic Leukemia (ALL) and 11 samples of Acute Myeloblastic Leukemia (AML), and an independent test set of 20 ALL and 14 AML samples. The data sets can be downloaded from http://www.genome.wi.mit.edu/MPR. These data sets contain many negative values which are meaningless for gene expressions, and need to be preprocessed. The negative values have been replaced by setting the threshold and maximum value of gene expression to 20 and 16000, respectively. Then genes that have $max(g) - min(g) > 500$ and $max(g)/min(g) > 5$ are excluded, leaving a total of 3859 genes. This type of preprocessing has been used in [5]. Then the data have been normalized after taking logarithm of the values.

b) Lymphoma Data Set: The Diffused Large B-Cell Lymphoma (DLBCL) data set [1] contains gene expression measurements of 96 normal and malignant lymphocyte samples, each measured using a specialized cDNA microarray, containing 4026 genes that are either preferentially expressed in lymphoid cells or of known immunological or oncological importance. The expression data in raw format are available at http://llmpp.nih.gov/lymphoma/data/figure1/figure1.cdt. It contains 42 samples of DLBCL and 54 samples of other types. There are some missing gene expression values which have been replaced by applying k-nearest neighbor algorithm in [5]. Then the expression values have been normalized, and the data set is randomly divided into mutually exclusive training and test sets of equal size.

c) Colon Data Set: This data set, a collection of expression values of 62 colon biopsy samples measured using high density oligonucleotide microarrays containing 2000 genes, is reported by Alon et al. [2]. It contains 22 normal and 40 colon cancer samples. It is available at http://microarray.princeton.edu/oncology. These gene expression values have been log transformed, and then normalized. We divide the data randomly into mutually exclusive training and test sets of equal size.

B. Experimental Setup

We generate initial population with each individual having 10 to 60 random bit positions set to '1'. This has been done to reduce the run time. For calculation of marginal distribution, we select best half of the population (truncation selection, $\tau = 0.5$). The setting of other parameters are: Population Size=500, Maximum Generation=50, Offspring Size=450, α=0.1, w_1=0.75 and w_2=0.25. We use both Naive-Bayes and weighted voting classifiers separately to predict the class of a sample. The algorithm terminates when there is no improvement of the fitness value of the best individual in 5 consecutive generations or maximum number of generations has passed.

C. Experimental Results

Here we present the experimental results of our algorithm and PBIL on the three data sets. All the results are the average of 50 independent runs. For PBIL, we set $\alpha = 0.9$ instead of 0.1. We tried PBIL with $\alpha = 0.1$ but it returned neither minimum size of gene subset nor encouraging accuracy in reasonable time. With the new value PBIL produced satisfactory results. For comparison, we also provide the experimental results of MOEA by Liu and Iba [12]. Though it is stated in the paper that the accuracy presented is on training set, it is actually the accuracy of all data (since all data have been used as training set) with prediction strength threshold 0. In the presented results, each value of the form $x \pm y$ indicates the average value x with the standard deviation y. The experimental results are shown in tables II–VI. The values inside parentheses are the experimental results of PBIL.

TABLE II

AVERAGE ACCURACY RETURNED BY OUR ALGORITHM USING WEIGHTED VOTING CLASSIFIER WITH PREDICTION STRENGTH THRESHOLD 0. THE RESULTS OF PBIL ARE SHOWN IN PARENTHESES

Data Set	Training Set	Test Set	Overall
Leukemia	1.0 ± 0.0	0.90 ± 0.06	0.96 ± 0.03
	(1.0 ± 0.0)	(0.86 ± 0.06)	(0.93 ± 0.03)
Lymphoma	0.99 ± 0.01	0.93 ± 0.04	0.96 ± 0.02
	(0.98 ± 0.02)	(0.91 ± 0.05)	(0.94 ± 0.03)
Colon	0.95 ± 0.03	0.81 ± 0.08	0.88 ± 0.04
	(0.91 ± 0.04)	(0.77 ± 0.01)	(0.84 ± 0.05)

From the experimental results, we see that our algorithm outperforms both PBIL and MOEA in both respects of experimental results: number of genes selected and the accuracy returned. Although all the methods may produce almost the same results on training data, they return different accuracies on test and overall data.

In the case of Leukemia and Lymphoma data sets, both our method and PBIL produce almost 100% accuracy on training data using Naive-Bayes classifier and weighted voting classifier with prediction strength threshold=0, and in the case of Colon data, our algorithm finds 95% accuracy while PBIL returns 91% accuracy. Big differences among two methods and

TABLE III

AVERAGE ACCURACY RETURNED BY OUR ALGORITHM USING WEIGHTED VOTING CLASSIFIER WITH PREDICTION STRENGTH THRESHOLD 0.30. THE RESULTS OF PBIL ARE SHOWN IN PARENTHESES

Data Set	Training Set	Test Set	Overall
Leukemia	0.99 ± 0.01	0.87 ± 0.06	0.94 ± 0.03
	(0.95 ± 0.02)	(0.80 ± 0.08)	(0.88 ± 0.04)
Lymphoma	0.97 ± 0.02	0.91 ± 0.05	0.94 ± 0.02
	(0.95 ± 0.03)	(0.88 ± 0.05)	(0.91 ± 0.03)
Colon	0.90 ± 0.04	0.74 ± 0.07	0.83 ± 0.04
	(0.83 ± 0.05)	(0.70 ± 0.09)	(0.77 ± 0.05)

TABLE IV

AVERAGE ACCURACY RETURNED BY OUR ALGORITHM USING NAIVE-BAYES CLASSIFIER. THE RESULTS OF PBIL ARE SHOWN IN PARENTHESES

Data Set	Training Set	Test Set	Overall
Leukemia	1.0 ± 0.0	0.90 ± 0.09	0.95 ± 0.05
	(0.99 ± 0.01)	(0.80 ± 0.11)	(0.90 ± 0.06)
Lymphoma	0.99 ± 0.01	0.91 ± 0.04	0.95 ± 0.02
	(0.99 ± 0.01)	(0.90 ± 0.06)	(0.94 ± 0.03)
Colon	0.95 ± 0.03	0.78 ± 0.08	0.87 ± 0.04
	(0.91 ± 0.04)	(0.73 ± 0.09)	(0.83 ± 0.05)

TABLE V

THE AVERAGE NUMBER OF GENES SELECTED BY OUR ALGORITHM USING WEIGHTED VOTING AND NAIVE-BAYES CLASSIFIERS. THE RESULTS OF PBIL ARE SHOWN IN PARENTHESES. WVC=WEIGHTED VOTING CLASSIFIER, PS=PREDICTION STRENGTH THRESHOLD

Data Set	WVC (PS=0)	WVC (PS=0.30)	NB Classifier
Leukemia	3.16 ± 1.0	3.78 ± 1.75	2.92 ± 1.0
	(10.8 ± 7.14)	(6.92 ± 3.94)	(10.2 ± 7.99)
Lymphoma	4.42 ± 2.46	2.42 ± 0.91	5.77 ± 4.10
	(7.76 ± 3.23)	(4.82 ± 2.85)	(14.2 ± 13.16)
Colon	4.44 ± 1.74	3.24 ± 1.34	5.14 ± 2.04
	(5.9 ± 2.98)	(3.44 ± 2.14)	(5.9 ± 3.62)

TABLE VI

THE OVERALL AVERAGE ACCURACY RETURNED AND NUMBER OF GENES SELECTED BY MOEA

Data Set	Overall Accuracy	Number of Genes Selected
Leukemia	0.90 ± 0.07	15.20 ± 4.54
Lymphoma	0.90 ± 0.03	12.90 ± 4.40
Colon	0.80 ± 0.08	11.4 ± 4.27

two classifiers can be observed on test data. PBIL produces better accuracy on test set using weighted voting classifier (PS=0) than those using Naive-Bayes classifier. The same is also true for our method. Both algorithms return lower accuracy using weighted voting classifier with prediction strength threshold 0.30, but the average number of genes selected is smaller (except in the case of Leukemia by our method) than those under zero confidence level. Under all conditions, both algorithms perform badly on Colon data. According to our knowledge, there have been reported no algorithms and no classifiers that return 100% accuracy on this data set. Finally, it is evident from the experimental results that our algorithm with either classifier provides better accuracy and identifies smaller number of informative genes than those by other methods for classification. Moreover, all our reported results are unbiased.

VII. DISCUSSION

Selection of the most useful genes for classification of available samples into two or more classes is a multi-objective optimization problem. There are many challenges for this classification task. Unlike other functional optimizations which use the values of the functions as fitness, this problem needs something beyond these values. It may be the case that you get 100% accuracy on training data but 0% accuracy on test data. So, the selection of proper training and test sets, and design of a reliable search method are very important. This problem has been solved in the past using both supervised and unsupervised methods. In this paper, we propose a new PMBGA for the selection of the gene subsets. Our method outperforms other algorithms by selecting the most useful gene subset for better classification.

In microarray data, overfitting (and sometimes underfitting) is a major problem because the number of training samples given is very small compared to the number of genes. To avoid it, many researchers use all the data available to guide the search and report the accuracies that were used during the gene selection phase as the final accuracies. This kind of estimation is biased towards the available data, and may predict poorly when used to classify unseen samples. But our accuracy estimation is unbiased because we have isolated the test data from training data during gene selection phase. Whenever a training set is given, we have used that one only for the selection of genes, and the accuracy on the independent test set is presented using the final gene subset; whenever the data is not divided, we randomly partition it into two exclusive sets: training and test sets, and provide accuracy as described before.

Our algorithm finds smaller numbers of genes but results in more accurate classification. This is consistent with the hypothesis that for a smaller training set, it may be better to select a smaller number of genes to reduce the algorithm's variance; and when more training samples are available, more genes should be chosen to reduce the algorithm's bias [10].

During our experiments, we have used two classifiers separately to show that our algorithm is not biased towards a specific classifier. Naive-Bayes classifier is applicable to multi-class classification whereas the weighted voting classifier for two-class problem. To get more confident results, we have used prediction strength threshold of 30% with the weighted voting classifier.

VIII. Summary and Future Work

In this paper, the selection of the most useful subset of genes for cancer class prediction in three well-known microarray data sets has been done by a new Probabilistic Model Building Genetic Algorithm (PMBGA) using either Naive-Bayes or weighted voting classifier. This new algorithm is a variant of PBIL. During the estimation of probability of each variable, we have emphasized on the fact that weight of a gene should play role in the selection of that gene, and this has been justified by the empirical results. The two objectives of the task have been combined into a single one by the weighted sum of the accuracy and the dimensionality of the gene subset. Since the number of available training samples compared to number of genes is very smaller, we have used the wrapper approach Leave-One-Out-Cross-Validation to calculate the accuracy of a gene subset on training data. The classification accuracy is notably improved and the number of genes selected is reduced with respect to both PBIL and MOEA.

However, there remain many unresolved issues that we we want to address in future. For example, DNA microarray data may contain noise, and dealing with noisy data is very important in Bioinformatics. In our future works, we want to work with these noisy data. Naive-Bayes classifier can be used for multiclass classification problems. We want to perform experiment using this classifier on multiclass data sets with a reasonable confidence level which we have not considered here. During our experiments, we found that some runs were not selecting some gene subsets with a few more genes although they would provide better test accuracy. We will pay attention to this in our future work.

References

[1] Alizadeh, A. A., Eisen, M. B., et al., "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling", *Nature*, vol. 403, pp. 503–511, 2000.

[2] Alon, U., Barkai, N., et al., "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays", in *Proceedings of National Academy of Science, Cell Biology*, vol. 96, 1999, pp. 6745–6750.

[3] Baluja, S. , "Population based incremental learning: A method for integrating genetic search based function optimization and competitive learning", *Technical Report No. CMU-CS-94-163*, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1994.

[4] Cestnik, B., "Estimating probabilities: a crucial task in machine learning", in *Proceedings of the European Conference on Artificial Intelligence*, 1990, pp. 147–149.

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

[6] Golub, G.R., et al., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring", *Science*, vol. 286, no.15, pp. 531–537, 1999.

[7] Holland, J.H., *Adaptation in Natural and Artificial Systems*, Ann Arbor: University of Michigan Press, 1975.

[8] Keller, A.D., Schummer, M., Hood, L. and Ruzzo, W.L., "Bayesian Classification of DNA Array Expression Data", *Technical Report UW-CSE-2000-08-01*, Department of Computer Science and Engineering, University of Washington, 2000.

[9] Kohavi, R., "A study of cross-validation and bootstrap for accuracy estimation and model selection", in *Proceedings of the International Joint Conference on Artificial Intelligence*, 1995.

[10] Kohavi, R. and John, G. H., "Wrappers for feature subset selection", *Artificial Intelligence*, vol. 97, no.1-2, pp. 273–324, 1997.

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

[12] Liu, J. and Iba, H., "Selecting Informative Genes using a Multiobjective Evolutionary Algorithm", in *Proceedings of the World Congress on Computation Intelligence(WCCI-2002)*, 2002, pp. 297–302.

[13] Liu, J. and Iba, H., "Selecting Informative Genes with Parallel Genetic Algorithms in Tissue Classification", in *Genome Informatics*, vol. 12, pp. 14–23, 2001.

[14] Mühlenbein, H. and Paaß, G., "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*, Berlin, Germany: Springer-Verlag, 1996, pp. 178–187.

[15] Paul, T. K. and Iba, H., "Linear and Combinatorial Optimizations by Estimation of Distribution Algorithms", in *Proceedings of the 9th MPS Symposium on Evolutionary Computation*, IPSJ, Japan, 2002, pp. 99–106.

[16] Paul, T. K. and Iba, H., "Reinforcement Learning Estimation of Distribution Algorithm", in *Proceedings of the Genetic and Evolutionary Computation Conference 2003 (GECCO2003), Lecture Notes in Computer Science (LNCS) 2724*, Springer-Verlag, 2003, pp. 1259–1270.

[17] Paul, T. K. and Iba, H.,"Optimization in Continuous Domain by Real-coded Estimation of Distribution Algorithm", in *Design and Application of Hybrid Intelligent Systems*, IOS Press, 2003, pp. 262–271.

[18] Pelikan, M., Goldberg, D.E. and Lobo, F.G., "A Survey of Optimizations by Building and Using Probabilistic Models", *Technical Report*, Illigal Report no. 99018, University of Illinois at Urbana-Champaign, USA, 1999.

[19] Pelikan, M., Goldberg, D.E. and Cantú-paz, E., "Linkage Problem, Distribution Estimation and Bayesian Networks", *Evolutionary Computation*, vol. 8, no.3, pp. 311-340, 2000.

[20] Rowland, J.J, "Generalization and Model Selection in Supervised Learning with Evolutionary Computation", *EvoWorkshops 2003, LNCS 2611*, Springer, 2003, pp. 119–130.

[21] Slonim, D. K., Tamayo, P., et al., "Class Prediction and Discovery Using Gene Expression Data", in *Proceedings of the* 4^{th} *Annual International Conference on Computational Molecular Biology*, 2000, pp. 263–272.