Identification of Informative Genes for Molecular Classification Using Probabilistic Model Building Genetic Algorithm

Topon Kumar Paul and Hitoshi Iba

Graduate School of Frontier Sciences, The University of Tokyo
Kashiwanoha 5-1-5, Kashiwa-shi, Chiba 277-8561, Japan
{topon,iba}@iba.k.u-tokyo.ac.jp

Abstract. DNA microarray allows the monitoring and measurement of the expression levels of thousands of genes simultaneously in an organism. A systematic and computational analysis of this vast amount of data provides understanding and insight into many aspects of biological processes. Recently, there has been a growing interest in classification of patient samples based on these gene expressions. The main challenge here is the overwhelming number of genes relative to the number of available training samples in the data set, and many of these genes are irrelevant for classification and have negative effect on the accuracy of the classifier. The choice of genes affects several aspects of classification: accuracy, required learning time, cost, and number of training samples needed. In this paper, we propose a new Probabilistic Model Building Genetic Algorithm (PMBGA) for the identification of informative genes for molecular classification and present our unbiased experimental results on three bench-mark data sets.

1 Introduction

The central dogma of molecular biology states that information is stored in DNA, transcribed to mRNA and then translated into proteins. The process by which mRNA and eventually protein is synthesized from the DNA template of each gene is called gene expression. 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. 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. Gene expression analysis 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

K. Deb et al. (Eds.): GECCO 2004, LNCS 3102, pp. 414–425, 2004.

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 [19]: filter and wrapper approaches for gene subset selection. In filter approach, the data are preprocessed and some top rank genes are selected independently of the classifier. Although filter approaches tend to be much faster, their major drawback is that an optimal subset of genes may not be independent of the representational biases of the classifier that will be used during the learning phase [19].

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 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)[6]. 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 $\frac{a(1-a)}{N}$ [6], 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 propose a new method based on Probabilistic Model Building Genetic Algorithm (PMBGA) [16], 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 separately Naive-Bayes (NB) classifier [3] and the weighted voting classifier [5,18]. 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 Multiobjective Evolutionary Algorithm (MOEA) [10].

2 Related Works in Molecular Classification Using Evolutionary Algorithms

Previously, Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [4], Multiobjective Evolutionary Algorithm(MOEA) [10] and Parallel Genetic Algorithm [9] 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 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 [7,17].

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 of the above 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.

3 Classifiers and Accuracy Estimation

3.1 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. The conditional probability takes the following form:

$$P(c_i|\mathbf{x}) \propto P(x_1|c_i)P(x_2|c_i)\cdots P(x_n|c_i)P(c_i) \ . \tag{1}$$

Taking logarithm we get,

$$\ln P(c_i|\mathbf{x}) \propto \ln P(x_1|c_i) + \cdots + \ln P(x_n|c_i) + \ln P(c_i) \ . \tag{2}$$

For a continuous gene, the conditional density is defined as

$$P(x_j|c_i) = \frac{1}{\sqrt{2\pi}\sigma_{ji}} e^{-\frac{(x_j-\mu_{ji})^2}{2\sigma_{ji}^2}} \tag{3}$$

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 (3) we get,

$$\ln P(x_j|c_i) = -\frac{1}{2}\ln(2\pi) - \ln\sigma_{ji} - \frac{1}{2}\left(\frac{x_j - \mu_{ji}}{\sigma_{ji}}\right)^2 \ . \tag{4}$$

Since the first term in (4) 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.

3.2 Classifier Based on Weighted Voting

Classifier based on weighted voting has been proposed in [5,18]. 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 the classes. The weight of a gene g is the correlation metric defined as

$$W(g) = \frac{\mu_1^g - \mu_2^g}{\sigma_1^g + \sigma_2^g} \tag{5}$$

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 x is

$$V(g) = W(g)\left(x_g - \frac{\mu_1^g + \mu_2^g}{2}\right) \tag{6}$$

where x_g is the value of gene g in that unknown sample. Then, the class of the sample x is

$$class(x) = sign\left\{\sum_{g\in G} V(g)\right\} \tag{7}$$

where G is the set of selected genes. If the computed value is positive, the sample x belongs to class 1; negative value means x belongs to class 2. But this kind of prediction does not make classification with reasonable confidence [5,18]. For more confident classification, we need to consider prediction strength. If we define V_+ and V_- are the absolute values of sum of all positive $V(g)$ and negative $V(g)$, respectively, the prediction strength,

$$ps = \left| \frac{V_+ - V_-}{V_+ + V_-} \right| \quad . \tag{8}$$

The classification according to (7) 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. This classifier is applicable to two-class classification tasks.

3.3 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.

4 Gene Selection Method

A new method based on Probabilistic Model Building Genetic Algorithm (PMBGA) [16] 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 generates offspring. 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 [11]. A PMBGA has the follow components: encoding of

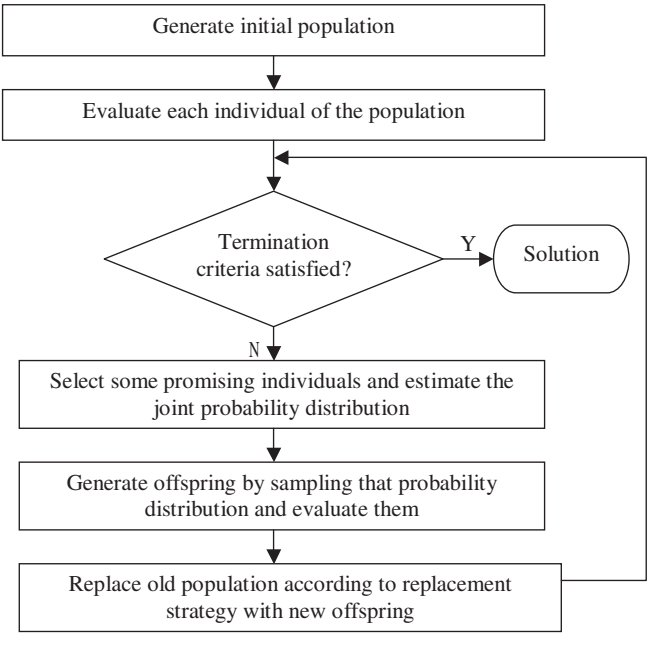

Fig. 1. Flowchart of a PMBGA

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. Reviews on PMBGA can be found in [8,12,13,14,15]. For our experiments, we propose another one which is described in the next subsection.

4.1 Proposed Method

Before we describe our proposed method, we need to give some notations. Let $X = \{X_1, X_2, \ldots, X_n\}$ be the set of n binary variables corresponding to n genes

in the data set, and $x = \{x_1, x_2, \ldots, x_n\}$ be the set of values of those variables with $x_i \in \{0, 1\}(i = 1, \ldots, n)$ being the value of the variable X_i. N 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 $p(x, t) = \prod_{i=1}^{n} p(x_i, t|pa_i)$ 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 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

$$p(x_i, t + 1) = \alpha p(x_i, t) + (1 - \alpha) M(x_i, t) \overline{w}(g_i) \tag{9}$$

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:

$$\overline{w}(g_i) = \frac{|W(g_i)|}{MAX\{|W(g_1)|, |W(g_2)|, \ldots |, W(g_n)|\}} \tag{10}$$

where each $W(g_i)$ is calculated according to (5). The marginal distribution of X_i is calculated as follows:

$$M(x_i, t) = \frac{\sum_{j=1}^{N} \delta_j^i}{N} \tag{11}$$

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. Let us give an example of generating an offspring. Suppose, there are 4 genes in the data set; the normalized weight vector of genes and the probability vector of the corresponding variables at generation t are $\overline{w}(g) = (0.01, 1.0, 0.75, 0.05)$ and $p(x, t) = (0.001, 0.8, 0.4, 0.5)$, respectively. The selected individuals from a population at generation t are $(0, 1, 1, 0), (0, 1, 0, 1)$and $(1, 1, 1, 0)$. Then, the corresponding marginal probabilities of variables will be $M(x, t) = (0.33, 1.0, 0.67, 0.33)$. If $\alpha = 0.1$, the updated probability according to (9) will be $p(x, t + 1) =$(0.00307,0.98,0.49225, 0.06485). Now generate 4 random numbers from uniform distribution and suppose they are $(0.001, 0.45, 0.6, 0.07)$. Comparing each random value with corresponding $p(x_i, t)$, we get an offspring $(1, 1, 0, 0)$.

4.2 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

$$fitness(X) = w_1 * a(X) + w_2 * (1 - d(X)/n) \tag{12}$$

where w_1 and w_2 are weights from $[0, 1]$, $a(X)$ is the accuracy of X on training data, $d(X)$ the number of genes selected in X, and n is the total number of genes. This kind of fitness calculation was used in [9].

5 Experiments

5.1 Data Sets

We evaluate our method on three cancer data sets: Leukemia, Lymphoma and Colon. Leukemia and Lymphoma data sets need some preprocessing because the first one has some negative values while the second one has some missing values; we used the preprocessed data from
http://www.iitk.ac.in/kangal/bioinformatics.

Leukemia Data Set. This is a collection of gene expressions of 7129 genes of 72 leukemia samples reported by Golub et al. [5]. 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 [4]. Then the data has been normalized after taking logarithm of the values.

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 expressed in lymphoid cells or which are 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 [4]. Then the expression values have been normalized, and the data set is randomly divided into mutually exclusive training and test sets of equal size.

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 training and test sets of equal size. The samples in one set are exclusive of the other set.

5.2 Experimental Results

For each data set and each experiment, the initial population is generated 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 settings of other parameters are: population size=500, maximum generation=10, elite=10%, α=0.1, w_1=0.75 and w_2=0.25. Elitism replacement has been used to prevent the so far found best individual of previous generations from being lost. We use both Naive-Bayes and weighted voting classifiers separately to predict the class of a sample. The algorithm terminates when either there is no improvement of the fitness value of the best individual in 5 consecutive generations or maximum number of generations has passed.

For all data sets, the average classification accuracy returned and number of genes selected by our algorithm in 50 independent runs are provided. For comparison, we provide only the experimental results of MOEA by Liu and Iba [10]. Though it is stated in the paper that the accuracy presented is on training set, it is actually the accuracy on 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 1, 2 and 3. In the tables, WVC stands for weighted voting classifier and θ is the prediction strength threshold.

From the experimental results, we find that our algorithm using either Naive-Bayes or weighted voting classifier with $\theta = 0$ returns the same average accuracy on all three training data, but using weighted voting classifier (θ=0) provides better accuracy on test data as compared to using Naive-Bayes classifier. Weighted voting classifier with $\theta = 0.30$, provides 100% and 90%, 97% and 92%, 92% and 74% average accuracy on training and test sets of Leukemia, Lymphoma and Colon data, respectively. Our algorithm performs badly on colon data as compared to on other two data sets. According to our knowledge, there have been reported no algorithms and no classifiers that return 100% accuracy on this data set. The overall average accuracy returned by our method on each data set using both classifiers is superior to the accuracy returned by MOEA.

During the experiments, we found that our algorithm was selecting in each independent run only 2 genes using both Naive-Bayes and weighted voting classifier with prediction strength threshold 0 in the case of Leukemia data set. Weighted voting classifier with prediction strength threshold of 0.30 selects 2.02, 2.04 and 2.48 genes on the average for the three data sets: Leukemia, Lymphoma and Colon, respectively. In the case of Lymphoma and Colon data sets, Naive-

Bayes classifier selects higher number of genes with higher standard deviations than weighted voting classifier. The number of genes selected by applying MOEA are larger than those selected by our algorithm.

Table 1. The average accuracy returned by our algorithm on training and test data using weighted voting and Naive-Bayes classifiers

Data Set	WVC(θ=0)		WVC(θ=0.30)		Naive-Bayes Classifier	
	Train Set	Test set	Train Set	Test set	Train Set	Test Set
Leukemia	1.0 ± 0.0	0.92 ± 0.05	1.0 ± 0.0	0.90 ± 0.03	1.0 ± 0.0	0.91 ± 0.09
Lymphoma	0.99 ± 0.01	0.92 ± 0.04	0.97 ± 0.02	0.92 ± 0.05	0.99 ± 0.01	0.91 ± 0.05
Colon	0.96 ± 0.03	0.80 ± 0.07	0.92 ± 0.04	0.74 ± 0.09	0.96 ± 0.04	0.79 ± 0.06

Table 2. The average number of genes selected by our algorithm using weighted voting and Naive-Bayes classifiers

Data Set	WVC (θ=0)	WVC (θ=0.30)	NB Classifier	MOEA
Leukemia	2.0 ± 0.0	2.02 ± 0.14	2.0 ± 0.0	15.20 ± 4.54
Lymphoma	2.66 ± 2.22	2.04 ± 0.20	3.96 ± 4.70	12.90 ± 4.40
Colon	2.30 ± 0.61	2.48 ± 0.81	3.11 ± 1.32	11.4 ± 4.27

Table 3. The overall average accuracy reported by our algorithm on three data sets using weighted voting and Naive-Bayes classifiers

Data Set	WVC (θ=0)	WVC (θ=0.30)	NB Classifier	MOEA
Leukemia	0.96 ± 0.02	0.95 ± 0.02	0.95 ± 0.05	0.90 ± 0.07
Lymphoma	0.95 ± 0.02	0.95 ± 0.02	0.95 ± 0.02	0.90 ± 0.03
Colon	0.88 ± 0.03	0.83 ± 0.05	0.88 ± 0.03	0.80 ± 0.08

6 Discussion

Identification 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 MOEA by selecting the most useful gene subset resulting in better classification accuracy.

In microarray data, overfitting 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 accuracy that was used during the gene selection phase as the final accuracy. 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 classifications. 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 [7].

7 Summary and Future Work

In this paper, we have proposed a novel PMBGA for selection of informative genes aimed at maximizing classification accuracy for classification of DNA microarray data using either Naive-Bayes or weighted voting classifier. In our algorithm, the normalized weight of a gene is incorporated into the equation of updating the probability of the corresponding variable. The two objectives of the problem have been scalarized into one objective. We used the Leave-One-Out-Cross-validation technique to calculate the accuracy of an individual (a gene subset) on training data. By performing experiments, we found that the the accuracy was notably improved and the number of gene selected was smaller as compared to MOEA.

However, we have not attempted to identify the accession numbers of the selected genes and to maintain population diversity during the experiments; which are very important for cancer diagnosis and multimodal optimization. In the future, we would like to take care of these issues during experiments. We also plan to extend our algorithm for noisy DNA microarray data. As Naive-Bayes classifier is applicable to multi-class classification, we would like to apply our algorithm using this classifier on larger multi-class cancer data sets.

References

1. Alizadeh, A. A., Eisen, M. B., et al.: Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature **403**(2000), 503–511.
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. Proceedings of National Academy of Science, Cell Biology, vol. 96, 1999, 6745–6750.
3. Cestnik, B.: Estimating probabilities: a crucial task in machine learning. Proceedings of the European Conference on Artificial Intelligence, 1990, 147–149.
4. Deb, K. and Reddy, A.R.: Reliable classification of two-class cancer data using evolutionary algorithms. BioSystems **72**(2003),111–129.
5. Golub, G.R., et al.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science **286**(15)(1999), 531–537.

6. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the International Joint Conference on Artificial Intelligence, 1995.
7. Kohavi, R. and John, G. H.: Wrappers for feature subset selection. Artificial Intelligence **97**(1-2)(1997), 273–324.
8. Larrañaga, P. and Lozano, J.A: Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publishers, Boston, USA, 2001.
9. Liu, J. and Iba, H.: Selecting Informative Genes with Parallel Genetic Algorithms in Tissue Classification. Genome Informatics **12**(2001), 14–23.
10. Liu, J. and Iba, H.: Selecting Informative Genes Using a Multiobjective Evolutionary Algorithm. Proceedings of the World Congress on Computation Intelligence(WCCI-2002), 2002, 297–302.
11. Mühlenbein, H. and Paaß, G. : From Recombination of Genes to the Estimation of Distribution I. Binary parameters. Parallel Problem Solving from Nature-PPSN *IV*, Lecture Notes in Computer Science (LNCS) 1411, Springer-Verlag, Berlin, Germany, 1996, 178–187.
12. Paul, T. K. and Iba, H.: Linear and Combinatorial Optimizations by Estimation of Distribution Algorithms. Proceedings of the 9th MPS Symposium on Evolutionary Computation, IPSJ, Japan, 2002, 99–106.
13. Paul, T. K. and Iba, H.: Reinforcement Learning Estimation of Distribution Algorithm. Proceedings of the Genetic and Evolutionary Computation Conference 2003 (GECCO2003), Lecture Notes in Computer Science (LNCS) 2724, Springer-Verlag, 2003, 1259–1270.
14. Paul, T. K. and Iba, H.:Optimization in Continuous Domain by Real-coded Estimation of Distribution Algorithm. Design and Application of Hybrid Intelligent Systems, IOS Press, 2003, pp. 262–271.
15. Pelikan, M., Goldberg, D.E. and Cantú-paz, E.: Linkage Problem, Distribution Estimation and Bayesian Networks. Evolutionary Computation **8**(3)(2000), 311-340.
16. 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).
17. Rowland, J.J: Generalization and Model Selection in Supervised Learning with Evolutionary Computation. EvoWorkshops 2003, LNCS 2611, Springer, 2003, pp. 119–130.
18. Slonim, D. K., Tamayo, P., et al.: Class Prediction and Discovery Using Gene Expression Data. Proceedings of the 4^{th} Annual International Conference on Computational Molecular Biology, 2000, 263–272.
19. Yang, J. and Honavar, V.: Feature subset selection using a genetic algorithm. Feature extraction, construction and selection, Kluwer Academic Publishers, 1998, pp. 118–135.