Genetic Search for Optimal Ensemble of Feature-Classifier Pairs in DNA Gene Expression Profiles

Chanho Park
Yonsei University
Department of Computer Science
134 Shinchon-dong, Seodaemun-gu, Seoul Korea, 120-749.
cpark@candy.yonsei.ac.kr

Sung-Bae Cho
Yonsei University
Department of Computer Science
134 Shinchon-dong, Seodaemun-gu, Seoul Korea, 120-749
sbcho@cs.yonsei.ac.kr

***Abstract*-Gene expression profile is numerical data of gene expression levels from organism, measured on the microarray. In general, each specific tissue indicates different expression level in related genes, so that it is possible to classify disease by gene expression profile. For classification, it is needed to select related genes called feature selection, because all the genes are not useful for classification. We propose GA-based method for searching optimal ensemble of feature-classifier pairs of gene expression profile in seven feature selection methods based on correlation, distance, and information theory, and representative six classifiers. Experimental results on two gene expression profiles related to cancers show that GA finds good solution quickly. Especially, in Lymphoma dataset, GA finds the ensemble of 100% accuracy.**

I. INTRODUCTION

Although earlier cancer detection and correct class discovery have been seriously studied over the past years, there has been no perfect way to work out this problem. It is because there are so many pathways causing cancer, and there exist tremendous number of varieties. Recently, array technologies have made it straightforward to monitor the expression patterns of thousands of genes during cellular differentiation and response [1]. These gene expression profiles, however, are just simple sequences of numbers, and the necessity of tools analyzing them to get useful information has risen sharply.

General procedure for classifying gene expression profile is divided into several steps. Because it does not fit to use raw data directly, normalization procedure is needed [2]. Then, we select informative genes by feature selection methods, because the number of genes is much greater than that of samples, and most of them do not help in classification [3]. After feature selection, classifier is trained by training samples with selected genes, so that input patterns yield right outputs. Trained classifier is evaluated the performance with test set or validation set [4, 5].

On the other hand, it is attempted to make ensemble classifiers, because it is hard to find perfect and general feature-classifier pair (combination of feature selection and classifier) [6]. Ensemble is to combine classifier pairs and it is generally known for yielding stable result. Ensemble can also search much wider solution space than individual classifiers. For ensemble various methods can be used such as majority voting, weighted voting, and weighted average.

One important problem of ensemble is that it takes long time to try all the ensembles because there are so many possible ensembles. If there are m feature selection methods and n classifiers, mn feature-classifier pairs are possible. In our case, mn is 42, so we can figure out the number of possible ensembles as follows.

$$\sum_{k=1}^{42} {}_{42}C_k \cong 4 \times 10^{12} \qquad (1)$$

Even the newest Pentium 4 computer with 1 GB main memory would take several days to test all the ensembles. Moreover, the required time increases exponentially as feature selection methods or classifiers are added to the system.

In this paper, we propose a method based on genetic algorithm (GA) for finding optimal ensemble of feature-classifier pairs efficiently. We have used randomly selected initial chromosomes, and shown the tendency to the optimal ensemble by genetic operations. We have tried to test the proposed method in two benchmark cancer datasets, and systematically analyze its usefulness.

II. BACKGROUNDS

A. DNA Microarray

DNA microarray consists of a large number of DNA molecules spotted in a systemic order on a solid substrate. Especially, depending on the size of each DNA spot on the array, DNA microarray means the diameter of DNA spot is less than 250 microns. The arrays with the small solid substrate are also referred to as DNA chips. It is so powerful that we can investigate the gene information in short time, because at least hundreds of genes can be put on the DNA microarray to be analyzed.

DNA microarrays are composed of thousands of individual DNA sequences printed in a high density array on a glass microscope slide using a robotic arrayer. The relative abundance of these spotted DNA sequences in two DNA or RNA samples may be assessed by monitoring the differential hybridization of the two samples to the sequences on the array. For mRNA samples, the two samples are reverse-transcribed into cDNA, labeled using different fluorescent dyes mixed (red-fluorescent dye Cy5 and green-fluorescent dye Cy3). After the hybridization of these samples with the arrayed DNA probes, the slides are imaged using scanner that makes fluorescence measurements for each dye. The log ratio between the two intensities of each dye is used as the gene expression data [7].

0-7803-7898-9/03/$17.00 ©2003 IEEE

$$gene\_expression = \log_2 \frac{\text{Int(Cy5)}}{\text{Int(Cy3)}} \quad (2)$$

Since at least hundreds of genes are put on the DNA microarray, it is so helpful that we can investigate the genome-wide information in short time.

B. Genetic Algorithm

Genetic algorithms are stochastic search methods that have been successfully applied in many search, optimization, and machine learning problems [8]. Unlike most other optimization techniques, GAs maintain a population of encoded solution candidates that are competitively manipulated by applying some variation operators to find a global optimum. A population consists of many chromosomes that can be a candidate solution. A chromosome is composed of bit strings that express a specific status or value.

A sequential GA proceeds in an iterative manner by generating new populations of strings from the old ones. Every string is the encoded (binary, real, ...) version of a candidate solution. An evaluation function associates a fitness measure to every string indicating its fitness to the problem. The standard GA applies genetic operators such as selection, crossover, and mutation on an initially random population in order to compute a whole generation of new strings.

C. Related Works

Many people have been studying about gene expression profile classification. They have used various feature selection methods to select informative genes, such as information gain, signal to noise ratio, *t*-statistics, Euclidean distance, Pearson correlation coefficient, principal component analysis, genetic algorithm, and so on [2, 3, 9, 10]. As classifiers, they have used MLP, *k*NN, SVM, Fisher's linear discriminant analysis, logistic discriminant, decision tree, and so on [3, 4, 5, 11]. It is summarized in Table I.

On the other hand, many researchers have been working on the ensemble of the multiple classifiers to improve the performance of classification. The ensemble classifier also produces stable results. The ensemble methods used are majority voting, Bayesian average, neural network, and so on [6, 12].

III. Optimal Ensemble Classifier

The classification architecture for DNA microarray is composed of a series of processes to classify samples. The architecture contains feature selection, classifier, and finding optimal ensemble using GA. The system is as shown in Fig. 1.

A. Gene Selection Methods

Seven feature selection methods are used to select informative genes based on statistical correlation, distance, or information theory. The genes are selected by rank.

1) Statistical approach: Using the statistical correlation analysis, we can see the linear relationship and the direction of relation between two variables. Correlation coefficient r varies from –1 to +1, so that the data distributed near the line biased to (+) direction will have positive coefficients, and the data near the line biased to (-) direction will have negative coefficients.

Suppose that we have a gene expression pattern g_i ($i = 1 \sim m$, where m is the number of genes). Each g_i is a vector of gene expression levels from n samples, $g_i = (e_1, e_2, e_3, \ldots, e_n)$. Some elements are examples of class 1, and the others are those from class 0. An ideal gene pattern that belongs to class 1 is defined by $g_{ideal\_c1} = (1, \ldots, 1, 0, \ldots, 0)$, so that all the elements from class 1's samples are 1 and the others are 0. In this paper, we have calculated the correlation coefficients between this g_{ideal} and the expression pattern of each gene. When we have two vectors X and Y that contain N elements, $r_{Pearson}$ (PC) and $r_{Spearman}$ (SC) are calculated as follows:

TABLE I
Relevant works on cancer classification

Authors	Dataset	Method		Accuracy [%]
		Feature selection	Classifier	
Furey *et al.*	**Leukemia**	**Signal to noise ratio**	**SVM**	**94.1**
	Colon			**90.3**
Li *et al.*	**Lymphoma**	**Genetic Algorithm**	**KNN**	**84.6~**
	Colon			**94.1~**
Dudoit *et al.*	**Leukemia**	**The ratio of between-groups to**	**Nearest neighbor**	**95.0~**
	Lymphoma	**within-groups sum of squares**		**95.0~**
	Leukemia		**Diagonal linear discriminant analysis**	**95.0~**
	Lymphoma			**95.0~**
Nguyen *et al.*	**Leukemia**	**Principal component analysis**	**Logistic discriminant**	**94.2**
	Lymphoma			**98.1**
	Colon		**Diagonal linear discriminant analysis**	**87.1**
	Leukemia		**Quadratic discriminant analysis**	**95.4**
	Lymphoma		**BoostCART**	**97.6**
	Colon			**87.1**

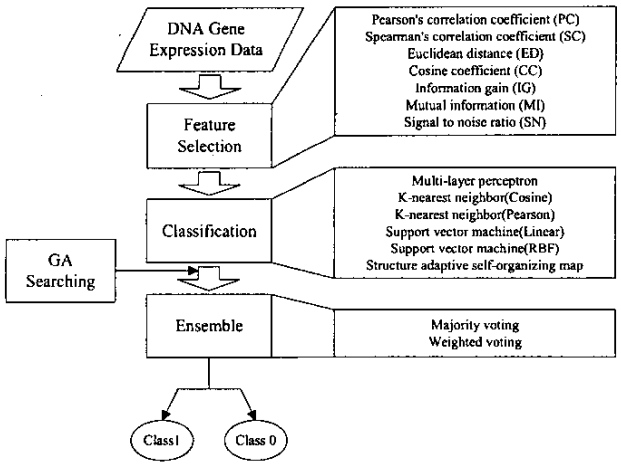

Fig. 1. The ensemble classification system

$$r_{Pearson} = \frac{\sum XY - \frac{\sum X \sum Y}{N}}{\sqrt{\left(\sum X^2 - \frac{(\sum X)^2}{N}\right)\left(\sum Y^2 - \frac{(\sum Y)^2}{N}\right)}} \tag{3}$$

$$r_{Spearman} = 1 - \frac{6\sum (D_x - D_y)^2}{N(N^2-1)} \tag{4}$$

where D_x and D_y are the rank matrices of $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively.

2) Distance approach: The similarity between two input vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$ can be thought of as distance. Distance is a measure on how far the two vectors are located, and the distance between g_{ideal_c1} and g_i tells us how much the g_i is likely to the class 1. Calculating the distance between them, if it is bigger than certain threshold, the gene g_i would belong to class 1, otherwise g_i belongs to class 0. In this paper, we have adopted Euclidean distance ($r_{Eclidean}$, ED) and cosine coefficient (r_{Cosine}, CC) represented by the following equations:

$$r_{Eclidean} = \sqrt{\sum (X-Y)^2} \tag{5}$$

$$r_{Cosine} = \frac{\sum XY}{\sqrt{\sum X^2 \sum Y^2}} \tag{6}$$

3) Information-theoretic approach: We have utilized the information gain and mutual information that are widely used in many fields such as text categorization and data mining. If we count the number of genes excited ($P(g_i)$) or not excited ($P(\bar{g}_i)$) in category c_j ($P(c_j)$), the coefficients of the information gain (IG) and mutual information (MI) become as follows:

$$IG(g_i, c_j) = P(g_i \mid c_j) \log \frac{P(g_i \mid c_j)}{P(c_j) \cdot P(g_i)} + P(\bar{g}_i \mid c_j) \log \frac{P(\bar{g}_i \mid c_j)}{P(c_j) \cdot P(\bar{g}_i)} \tag{7}$$

$$MI(g_i, c_j) = \log \frac{P(g_i, c_j)}{P(c_j) \cdot P(g_i)} \tag{8}$$

Mutual information tells us the dependency between two probabilistic variables of events. If two events are completely independent, the mutual information is 0. The more they are related, the higher the mutual information gets. Information gain is used when the features of samples are extracted by inducing the relationship between gene and class by the presence frequency of the gene in the sample. Information gain measures the goodness of gene using the presence and absence within the corresponding class.

For each gene g_i, some are from class 1, and some are from class 0. If we calculate the mean μ and standard deviation σ from the distribution of gene expressions within their classes, the signal to noise ratio of gene g_i, $SN(g_i)$, is defined by:

$$SN(g_i) = \frac{\mu_{c1}(g_i) - \mu_{c0}(g_i)}{\sigma_{c1}(g_i) + \sigma_{c0}(g_i)} \tag{9}$$

B. Classifiers

1) Multilayer perceptron: Error backpropagation neural network is a feed-forward multilayer perceptron (MLP) that is applied in many fields due to its powerful and stable learning algorithm [4]. The neural network learns the training examples by adjusting the synaptic weight of neurons according to the error occurred on the output layer. The power of the backpropagation algorithm lies in two main aspects: local for updating the synaptic weights and biases, and efficient for computing all the partial derivatives of the cost function with respect to these free parameters.

2) k-nearest neighbor: *k*-nearest neighbor (KNN) is one of the most common methods for memory based induction. Given an input vector, KNN extracts *k* closest vectors in the reference set based on similarity measures, and makes decision for the label of input vector using the labels of the *k* nearest neighbors [13].

Pearson's correlation and cosine coefficient have been used as the similarity measure. When we have an input X and a reference set $D = \{d_1, d_2, \ldots, d_N\}$, the probability that X may belong to class c_j, $P(X, c_j)$ is defined as follows:

$$P(X, c_j) = \sum_{d_i \in kNN} \mathrm{Sim}(X, d_i) P(d_i, c_j) - b_j \tag{10}$$

where $\mathrm{Sim}(X, d_i)$ is the similarity between X and d_i, and b_j is a bias term.

3) Support vector machine: Support vector machine (SVM) estimates the function classifying the data into two classes [6, 14]. SVM builds up a hyperplane as the decision surface in such a way to maximize the margin of separation between positive and negative examples. SVM achieves this by the structural risk minimization principle that the error rate of a learning machine on the test data is bounded by the sum of the training-error rate and a term that depends on the Vapnik-Chervonenkis (VC) dimension. Given a labeled set of M

training samples (X_i, Y_i), where $X_i \in R^N$ and Y_i is the associated label, $Y_i \in \{-1, 1\}$, the discriminant hyperplane is defined by:

$$f(X) = \sum_{i=1}^{M} Y_i \alpha_i k(X, X_i) + b \qquad (11)$$

where $k(\,.\,)$ is a kernel function and the sign of $f(X)$ determines the membership of X. Constructing an optimal hyperplane is equivalent to finding all the nonzero α_i (support vectors) and a bias b. We have used SVM^{light} module in this paper.

4) Structure adaptive SOM: Even though SOM is well known for its good performance of topology preserving, it is difficult to apply it to practical classification since the topology should be fixed before training. A structure adaptive self-organizing map (SASOM) was proposed to overcome this shortcoming [15]. SASOM starts with 4×4 map, and dynamically splits the output nodes of the map, where the data from different classes are mixed, trained with the LVQ learning algorithm.

C. Ensemble Classifier

It is hard to find the optimal method for classification, because there are many algorithms for the classification, and depending on the data, algorithms, features, and parameters used, the classifier would yield its result differently. To solve these problems, ensemble method is attempted and studied. It is informed that one can get improved result by combining classifiers that produce their own output.

If we just use one feature-classifier pair, we can get 42 results. However, we can get about 4-tera results by combining them. It means that we can search much wider solution space.

We have applied this idea to the classification framework as shown in Fig. 1. We have chosen majority voting and weighted voting method, among various methods for ensemble, and they are explained in the Table II.

D. Searching Optimal Ensemble Using GA

It takes so long time to test all possible ensembles. Moreover, if we added one feature-classifier pair, it would increase the necessary time exponentially. Therefore efficient method for finding optimal ensemble is needed, and we have proposed the method using GA. The structure of chromosome is as shown in Fig. 2.

TABLE II

Ensemble methods (when x is input, $C_{1i}(x)=1$ if $e_i(x)=1$, otherwise $C_{1i}(x)=0$, $c_{0i}(x)=1$ if $e_i(x)=0$, otherwise $c_{0i}(x)=0$, w_i is accuracy of $e_i(x)$, and $e_i(x)$ is element feature-classifier)

Ensemble method	Output	Condition
Majority voting	1	$\sum_i (c_{1i}(x)) > \sum_i (c_{0i}(x))$
	0	otherwise
Weighted voting	1	$\sum_i (c_{1i}(x) w_i) > \sum_i (c_{0i}(x) w_i)$
	0	otherwise

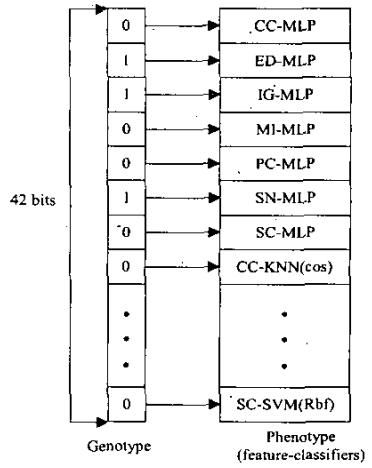

Fig. 2. Structure of chromosome

Each chromosome is composed of 42 bits string, and each bit indicates whether the corresponding feature-classifier pair is included or not. Each bit corresponding specific feature-classifier such as the first bit to CC-MLP, the second bit to ED-MLP, and so on. Fig. 2 shows the ensemble of second, third, and sixth feature-classifier pairs because their bits are 1.

After initial population is created, fitness is evaluated based on the performance of the ensemble.

IV. Experiments

A. Experimental Dataset

There are several microarray datasets from published cancer gene expression studies. In this paper, we have used two representative datasets among them.

Lymphoma cancer dataset consists of 24 samples of GC B-like and 23 samples of activated B-like [16]. Each sample contains 4026 genes. 22 out of 47 samples were used as training data and the remaining were used as test data. (Available at: http://genome-www.stanford.edu/lymphoma)

Colon cancer dataset consists of 62 samples taken from colon-cancer patients. Each sample contains 2000 gene expression levels. 40 samples are colon cancer samples and the remaining are normal samples. Each sample was taken from tumors and normal healthy parts of the colons of the same patients [17]. 31 out of 62 samples were used as training data and the remaining were used as test data. (Available at: http://www.sph.uth.tmc.edu:8052/hgc/default.asp)

B. Experimental Environment

The experiment is composed of gene selection, classification, and searching optimal ensemble using GA. For gene selection, we rank the genes according to its feature score, and select 25 highly scored genes. For classifier, we have used a two-layered MLP with 8 hidden nodes, 2 output nodes, 0.01~0.50 of learning rate, 0.9 of momentum, 500 of maximum iterations,

and 98% of target accuracy. In the case of *k*NN, we have set *k* from 1 to 8, and used Pearson correlation coefficient and cosine coefficient as similarity measures. We have used SVM with linear and RBF kernel function. In SASOM, we have used initial 4×4 map which has rectangular shape.

For GA, we have used roulette wheel rule for selection method. Preliminary results have indicated that it converges local minimum when we have used less than 100 chromosomes in a population. Stable result can be obtained when we use more than 100 chromosomes. In evaluation, a chromosome gets higher fitness when the number of feature-classifier pairs is smaller. Finally we have tested with crossover rates to 0.3, 0.5, 0.7 and 0.9, and mutation rates to 0.01 and 0.05.

C. *Results of Element Feature-Classifier*

Tables III and IV show the results of all element feature-classifier pairs for two datasets. In Lymphoma dataset, IG shows good performance for gene selection and *k*NN(Cosine) shows good accuracy for classification. Overall, feature selection methods based on information-theoretic approach show better performance than others in this dataset.

Also, *k*NN(Cosine) shows the best average performance in Colon dataset. However, the best performance of element feature-classifier pair, 83.9%, is worse compared with that in Lymphoma (92.0%) datasets. In the case of SASOM classifier, it has shown the worst performance that does not reach to 50% accuracy.

Table III

Recognition rates (%) of element feature-classifier pairs in Lymphoma dataset

	MLP	SASOM	SVM		KNN	
			Linear	RBF	Cosine	Pearson
PC	64.0	48.0	56.0	60.0	76.0	60.0
SC	60.0	68.0	44.0	44.0	60.0	60.0
ED	56.0	52.0	56.0	56.0	68.0	56.0
CC	68.0	52.0	56.0	56.0	72.0	60.0
IG	92.0	84.0	92.0	92.0	92.0	92.0
MI	72.0	64.0	64.0	64.0	64.0	80.0
SN	76.0	76.0	72.0	76.0	80.0	76.0
Avg.	69.7	63.4	62.9	63.4	73.1	69.1

Table IV

Recognition rates (%) of element feature-classifier pairs in Colon dataset

	MLP	SASOM	SVM		KNN	
			Linear	RBF	Cosine	Pearson
PC	74.2	74.2	64.5	64.5	77.4	71.0
SC	58.1	45.2	64.5	64.5	67.7	61.3
ED	67.8	67.6	64.5	64.5	83.9	83.9
CC	83.9	64.5	64.5	64.5	80.7	80.7
IG	71.0	71.0	71.0	71.0	80.7	74.2
MI	71.0	71.0	71.0	71.0	80.7	74.2
SN	64.5	45.2	64.5	64.5	71.0	64.5
Avg.	70.1	62.7	66.4	66.4	77.4	72.8

D. *Results of Optimal Ensemble with GA*

Before finding optimal ensemble, we investigate the change of average fitness to see if GA evolves, because optimal ensemble can be found by chance. Fig. 3 shows the change of average fitness in Lymphoma dataset. We can see that average fitness increases as iteration goes. It shows to converge after about 100 iterations, and it is similar in colon datasets.

We have not been able to obtain good performance with element feature-classifier pairs. It means that there exists high possibility to get improved performance, and GA practically finds outstanding ensembles that are better than any other element feature-classifier pairs.

In the case of Lymphoma dataset, GA finds the ensemble that shows the best performance. In this dataset, the accuracy of element feature-classifier pairs is 44~92%, leading to average of 67%. Nevertheless, GA finds the ensemble of 100% that is composed of complementary pairs of feature-classifier. Ensembles searched by GA are of 92~100% accuracy, and of 96.7% on average. Table V is some ensembles that show 100% accuracy.

Also, in the case of Colon dataset, GA finds the ensemble that shows higher performance than any other element feature-classifier pairs. The best accuracy of element feature-classifier pair is 83.9%, but the ensemble searched by GA is of 93.5% accuracy. Ensemble results in this dataset have ranged 87.1~93.5%, and average accuracy is 90.8%. These results are summarized in Fig. 4.

The practical usefulness of GA lies in its time-efficiency. We have experimented with all the ensembles that are composed of 7 feature-classifier pairs, and it takes about 3 hours. Therefore, it is almost impossible to test all the ensembles of 42. It takes less than 2 minutes using GA, however, when we use 2000 chromosomes with 500 iterations. We can find optimal ensemble in 100 iterations with 2000 chromosomes, and 900 iterations with 100 chromosomes. With these results, we can confirm that it is very efficient to use GA when searching space is very large.

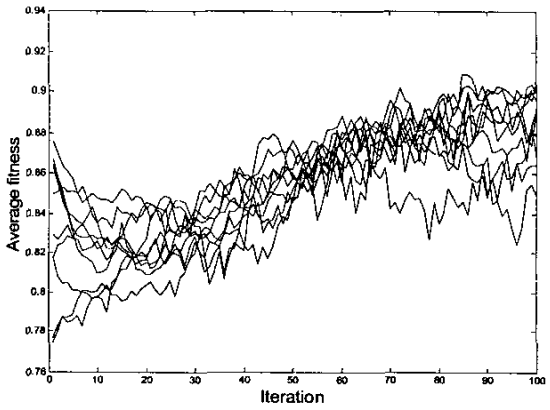

Fig. 3. Change of average fitness on Lymphoma dataset

Table V

Optimal ensembles of Lymphoma dataset

Methods	Feature-classifier pairs
Majority voting	CC-KNN(P), MI-KNN(C), SN-KNN(C), SS-SASOM, IG-SVM(L)
	IG-KNN(C), MI-KNN(C), PC-KNN(C), SN-KNN(P), SN-SASOM
Weighted voting	IG-KNN(C), IG-KNN(P), PC-KNN(P), SN-KNN(P), CC-SASOM
	IG-KNN(C), MI-KNN(C), SN-KNN(C), SN-KNN(P), CC-SASOM, IG-SASOM, PC-SVM(R)

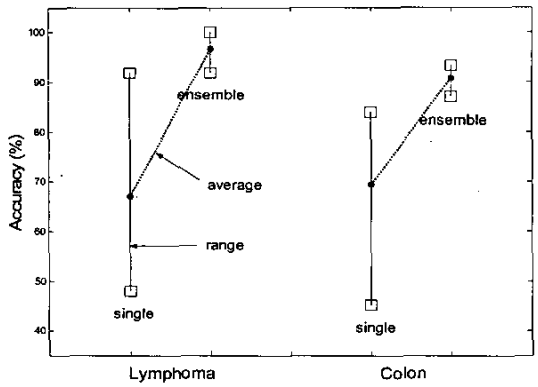

Fig. 4. Comparison of recognition rates for feature-classifier pairs (single) and ensemble (%)

V. CONCLUDING REMARKS

This paper uses GA that imitates the evolution of organism for efficiently finding the optimal ensemble classifier to analyze gene expression profiles. Experimental results show that GA finds ensembles that produce higher performance than any element feature-classifier pairs. Especially in Lymphoma dataset, GA finds the ensemble that yields 100% accuracy. GA finds optimal solution quickly, and it is more useful when the number of feature selection methods or classifiers is large. In addition, we can see that GA finds optimal solution through the increase of average fitness.

On the other hand, though we have just used simple GA and simple ensemble methods, we will investigate alternative methods later, such as rank-based or tournament selection method in GA and Bayesian ensemble method. Since we could not investigate the meaning of genes that do with the optimal ensemble found by GA, it might be interesting to see the biological implication.

ACKNOWLEDGEMENTS

This work was supported by Biometrics Engineering Research Center and a grant of Korea Health 21 R&D Project, Ministry of Health & Welfare, Republic of Korea.

REFERENCES

[1] T. R. Golub, *et al.* "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, pp. 531-537, 1999.

[2] Y. H. Yang, et al., "Normalization for cDNA microarray data," *SPIE Bios 2001*, San Jose, California, pp. 1-12, January 2001.

[3] L. Li, et al., "Gene selection for sample classification based on gene expression data: Study of sensitivity to choice of parameters of the GA/KNN method," *Bioinformatics*, vol. 17, no. 12, pp. 1131-1142, June 2001.

[4] J. Khan, et al., "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nature*, vol. 7, no. 6, pp. 673-679, June 2001.

[5] M. P. S. Brown, et al., "Support vector machine classification of microarray gene expression data," *USCS-CRL-99-09*, pp. 1-23, June 1999.

[6] S.-B. Cho, and J. Ryu, "Classifying gene expression data of cancer using classifier ensemble with mutually exclusive features," *Proc. of the IEEE*, vol. 90, no. 11, pp. 1744-1753, 2002.

[7] M. B. Eisen, et al., "Cluster analysis and display of genome-wide expression patterns," *Proc. of the Natl. Acad. of Sci. USA*, vol. 95, pp. 14863-14868, 1998.

[8] T. M. Mitchell, *Machine Learning*, Carnegie Mellon University, 1997.

[9] S. Fuhrman, et al., "The application of Shannon entropy in the identification of putative drug targets," *BioSystems*, vol. 55. pp. 5-14, 2000.

[10] D. V. Nguyen, et al., "Tumor classification by partial least squares using microarray gene expression data," *Bioinformatics*, vol. 18, no. 1, pp. 39-50, 2002.

[11] Y. Xu, et al., "Artificial neural networks and gene filtering distinguish between global gene expression profiles of Barrett's esophagus and esophageal cancer," *Cancer Research*, vol. 62, pp. 3493-3497, 2002.

[12] A. Ben-Dor, et al., "Tissue classification with gene expression profiles," *Journal of Computational Biology*, vol. 7. pp. 559-584, 2000.

[13] R. O. Duda, et al., *Pattern Classification*, 2nd Ed., Wiley Interscience, 2001.

[14] T. S. Furey, et al., "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, no. 10, pp. 906-914, 2000.

[15] H. D. Kim, and S.-B. Cho, "Genetic optimization of structure-adaptive self-organizing map for efficient classification," *Proc. of Int. Conf. on Soft Computing*, pp. 34-39, October 2000.

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

[17] U. Alon et al., "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proc. Natl. Acad. Sci. USA*. vol. 96, pp. 6745-6750, June 1999.