Complex 2004

Proceedings of the 7th Asia-Pacific
Conference on Complex Systems
Cairns Converntion Centre, Cairns, Australia
6-10th December 2004

Ensemble Genetic Programming for Classifying Gene Expression Data

Jin Hyuk Hong and Sung Bae Cho
Dept. of Computer Science, Yonsei University
134 Sinchon-dong, Sudaemoon-ku
Seoul 120-749, Korea
Email: hjinh@sclab.yonsei.ac.kr, sbcho@cs.yonsei.ac.kr

Abstract

Ensemble is a representative technique for improving classification performance by combining a set of classifiers. It is required to maintain the diversity among base classifiers for effective ensemble. Conventional ensemble approaches construct various classifiers by estimating the similarity on the output patterns of them, and combine them with several fusion methods. Since they measure the similarity indirectly, it is restricted to evaluate the precise diversity among base classifiers. In this paper, we propose an ensemble method that estimates the similarity between classification rules by matching in representation-level. A set of comprehensive and precise rules is obtained by genetic programming. After evaluating the diversity, a fusion method makes the final decision with a subset of diverse classification rules. The proposed method is applied to cancer classificaiton using gene expression profiles, which requires high accuracy and reliability. Especially, the experiments on popular cancer datasets have demonstrated the usefulness of the proposed method.

1. Introduction

Classifier combination, known as ensemble, has received the attention in the past decade and is now one of the standard and most important techniques to improve classification performance in machine learning [1,2]. The ensemble classifier obtained by combining the outputs of multiple classifiers is aimed to be more accurate and reliable than an individual classifier, while both theoretical and empirical research has produced valuable results on that. Hansen and Salamon have provided the theoretical basis on ensemble [3], while Opitz and Maclin have performed empirical ensemble experiments comprehensively [4]. Zhou et al. have analyzed the effect on the number of participating classifiers into ensemble in both theoretical and empirical studies [5]. Bagging and boosting have been actively investigated to generate the base classifiers as popular ensemble learning techniques, while various fusion strategies have also been studied for effective ensemble [1,2,6]. A survey on generating diverse classifiers for ensemble has been conducted by Brown [7]. A hybrid model for efficient ensemble was studied by Bakker and Heskes [8], while Tan and Gilbert applied ensemble to cancer classification based on gene expression data [9].

Even though ensemble might be promising in improving classification accuracy, it is known that ensemble with the same classifiers does not produce any elevation on performance [6]. Selecting precise and diverse base classifiers is very important in making a good ensemble [7]. Simple ways to generate various classifiers are randomly initializing parameters or making a variation of training data. Bagging (bootstrap aggregating) that was introduced by Breimen generates individual classifiers by training with a randomly organized set of samples from the original data [10]. Ensemble classifiers with bagging aggregate the base classifiers based on a voting mechanism. Boosting, which is another popular ensemble learning method, is introduced by Schapire to produce a series of classifiers [11]. A set of samples for training a classifier is chosen based on the performance of the previous classifiers in the series. Examples incorrectly classified by previous classifiers have more chances to be selected as training samples for the current classifier. Arching [12] and Ada-Boosting [11] are the representative boosting methods.

Some others select diverse classifiers at combining them for ensemble [1,6]. The diversity among classifiers is estimated by some measures, and a set of the most discriminating classifiers is selected for the ensemble classifier. In general, the error patterns of classifiers are used to measure the diversity [7]. Bryll proposed a different approach that employs different sets of features to generate different classifiers [2]. All of the research is aiming at generating distinct base classifiers, but most of them do not provide explicit methods to measure the diversity among classifiers. They are usually based on the randomness of systems or the implicit diversity-estimation by error patterns of individual classifiers. Therefore, an explicit method estimating the diversity among classifiers might be helpful to prepare a set of diverse base classifiers. Genetic programming, which is adopted in this work, might be useful to generate comprehensive classification rules, allowing the estimation of the diversity among them [13].

As ensemble helps to increase the reliability and accuracy of classification, it has been applied to many applications. Recently, cancer classification using gene expression data comes to be one of popular problems in machine learning. Since the data have a few samples with a number of genes, it is difficult to obtain an accurate and reliable classifier [14]. Ensemble approach has been performed for improving the performance of cancer classification using gene expression profiles [9,15].

The objective of this paper is to investigate ensemble in classifying gene expression profiles. Classification rules are generated by genetic programming, while the rules might be interpreted to explicitly calculate the diversity. A set of rules is selected based on the diversity to construct an ensemble classifier in which rules may be distinct as much as possible from the others. Section 2 describes the overview of cancer classification based on gene expression profiles, and classification using genetic programming is explained in Section 3. The proposed method and results are presented in Section 4 and 5. Conclusion and future work are finally summarized in Section 6.

2. Machine learning on gene expression profiles for cancer classification

Cancer classification based on gene expression profiles is one of the major research topics both in the medical field and in machine learning [11]. DNA microarray technology recently developed provides an opportunity to take a genome-wide approach to the correct prediction of cancers [14]. It captures the expression level of thousands of genes simultaneously which contains information on diseases. Figure 1 presents the process of gene expression profiling.

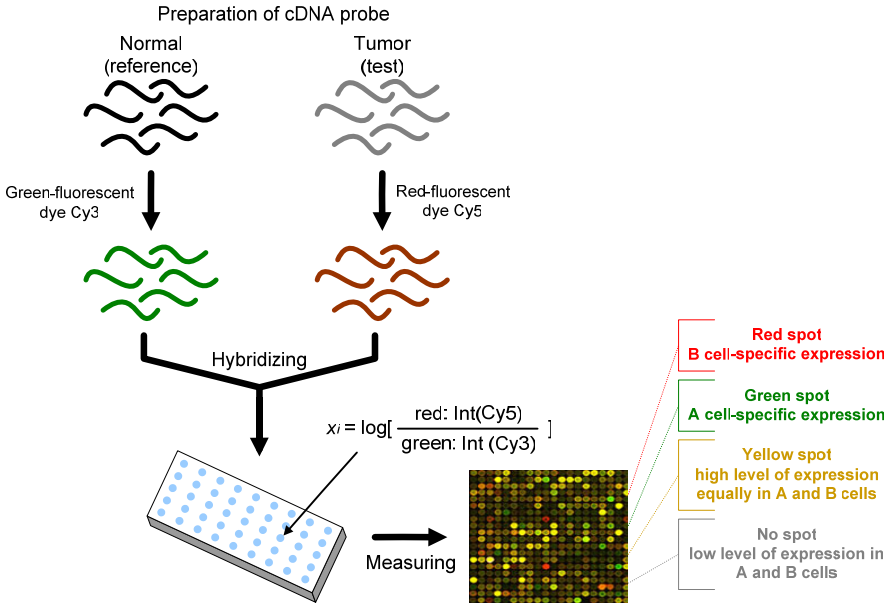

Figure 1. Overview of DNA microarray technology

The DNA microarray technology prints microscopic arrays of DNA sequences on glass slides, which are called DNA microarrays. They are constructed using cDNA, oligonucleotide, or genomic sequences according to applications. Especially cDNA arrays are made by spotting cDNAs onto glass slides, and the data are presented as microarrays through transcription. It measures the level of gene expression by hybridizing the fluorescently labeled tissue RNA and reference RNA labeled with a different fluorescent material. Typically, the reference RNA is dyed with green fluorescent materials while the tissue RNA is dyed with red fluorescent materials. The ratio of fluorescence intensities of the two dyes in a spot indicates the extent of a gene's up and down regulation, thereby the quantitative information of a cell is obtained. Moreover, it accesses to thousands of genes at once so as to obtain gene expression levels of those genes simultaneously [14,17].

Gene expression profiles are too huge to be analyzed manually by medical experts, and the functions and relationships of those genes are not recognized sufficiently. Therefore, automated analysis methods are requisite to extract useful knowledge from the data. Various machine learning techniques have been applied to analyzing gene expression profiles. Multi-layer perceptrons, support vector machines, k nearest neighbors classifiers, decision trees, and various discriminant methods are some of the popular classifiers that produce competitive results. The genetic algorithm (GA) has been also applied to selecting useful features, searching for the optimal ensemble, and so on [15,16].

Among these methods, the evolutionary algorithm has been emerged as a promising one for many applications in medical domain [18]. It searches not only optimized classification rules but also accurate and comprehensible rules. As GA-based classifiers, GABIL and GIL have been applied to generate a set of classification rules [19]. A hybrid method of the genetic algorithm and genetic programming has been tried by Tan et al. [18], and genetic programming has been applied to discover classification rules [20]. Many works have shown the comparative performance of genetic programming against neural networks, decision trees, and so on [15,16,21]. Although many of them have applied evolutionary methods to medical area, most data used in their work are not gene expression profiles but clinical samples. They have targeted clinical data that are composed of less than 100 attributes and values. The explicit knowledge discovery has been scarcely conducted for cancer classification based on gene expression profiles.

3. Arithmetic classification rule by genetic programming

A rule discovery method using genetic programming based arithmetic operators was proposed in the previous work [13]. A rule obtained is quite simple, but it shows good performance in classifying cancers based on gene expression profiles. The approach is composed of two parts: feature selection and rule discovery. Popular rank-based feature selection methods, such as Euclidean distance, cosine coefficient and signal-to-noise ratio, are employed to reduce the dimensionality of data, and then genetic programming is applied to find a classification rule.

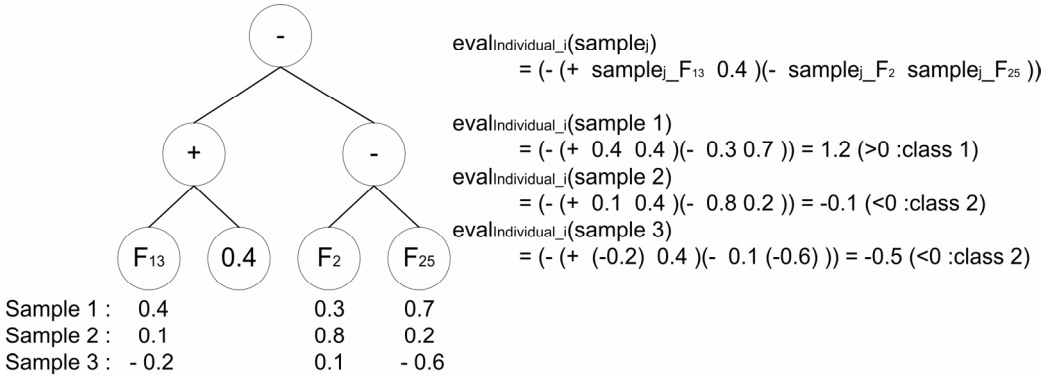

Figure 2. Classification rule with arithmetic operators

Table 1. Arithmetic operators used in this paper

Operator	Function	Description	Regulation
+	Addition	Positive on class 1(Negative on class 2)	Down regulation
–	Subtraction	Negative on class 1(Positive on class 2)	Up regulation
×	Multiplication	Multiplicative correlation	
/	Division	Divisive correlation	

The classification rule is represented as a tree that consists of several arithmetic operators, constants and features from gene expression data as shown in figure 2. A leaf node represents the corresponding gene or constants, while the other nodes signify arithmetic operators. The gene expression level implies up and down regulations for a cancer, so the rule might capture the implication. Here, a gene might have a tendency toward cancer as positive, negative or neutral. Arithmetic operators are able to consider those characteristics and combine the values of several genes related. If the estimated value of a sample is over 0, it will be classified into class 1, while if the value is under 0, it will be classified into class 2. Table 1 shows the arithmetic operators used in this paper and the meanings of them for analysis.

$$\text{IF } eval_{Individual\_i}(sample_j) > 0 \text{ THEN } class\ 1$$
$$\text{ELSE IF } eval_{Individual\_i}(sample_j) < 0 \text{ THEN } class\ 2$$
$$\text{ELSE reject to make a decision}$$

The rule not only decides samples' categories but also provides useful information on genes involved. A leaf node positively connected to the root can be interpreted as being

positive on class 1 and down regulation for cancer. A negatively connected node is negative on class 1 while it signifies up regulation for cancer at the same time.

Common genetic operators for genetic programming are employed for evolution. Since an individual is represented as a tree, crossover is conducted by changing randomly selected sub-trees of two individuals as shown in figure 3 (a). Mutation is performed by selecting a sub-tree and initializing it. In addition, permutation, which exchanges two sub-trees of an individual, is used to preserve the inherited character of an individual. Figure 3 (b) and figure 3 (c) show the mutation and the permutation of genetic programming. All genetic operations are conducted according to the predefined probabilities for experiments.

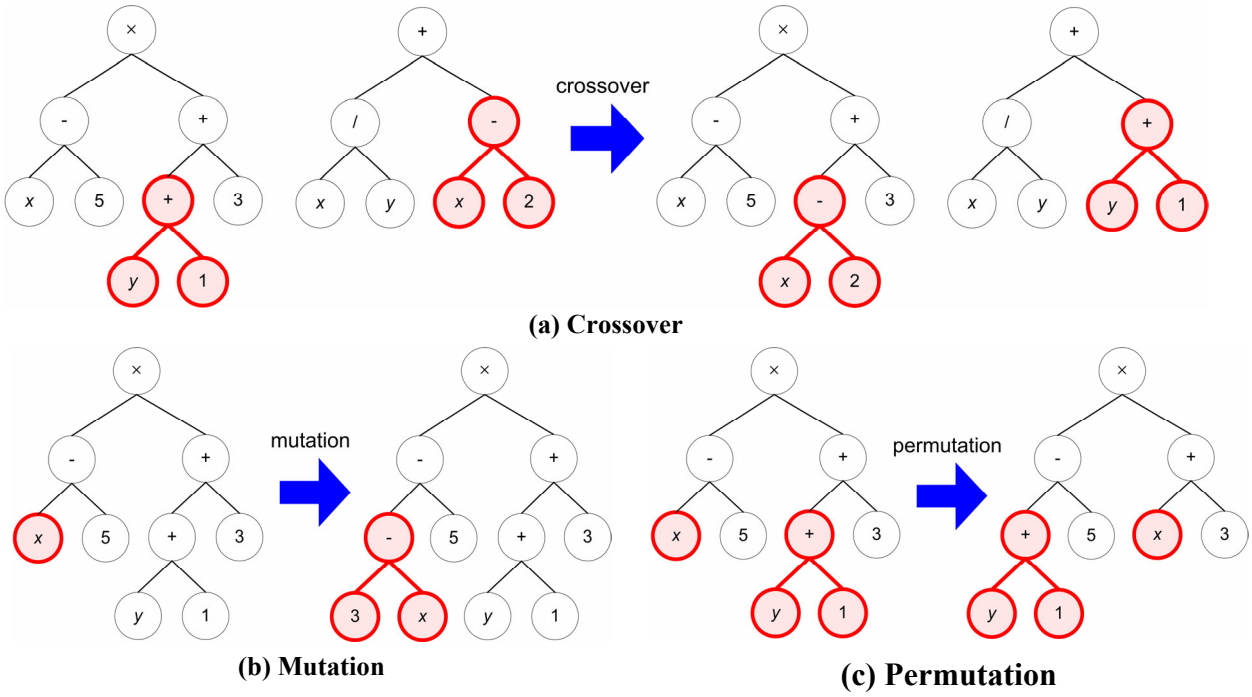

Figure 3. Genetic operations of genetic programming

4. Ensemble of diverse classification rules by genetic programming

4.1 Overview

The proposed method consists of 3 processes as shown in figure 4: feature selection, multiple rule discovery, and fusion. Based on the previous work, Euclidean distance, cosine coefficient and signal-to-noise ratio are employed to score the degree of association of genes with diseases. With the selected genes, a set of genetic programming works to generate multiple classification rules. The similarity between these rules are measured by a matching method designed, and several diverse classification rules are selected to make the final output. Contrary to conventional ensemble learning methods that simply combine the outputs of individual classifiers, the proposed method picks up some classification rules that maximize the diversity among them.

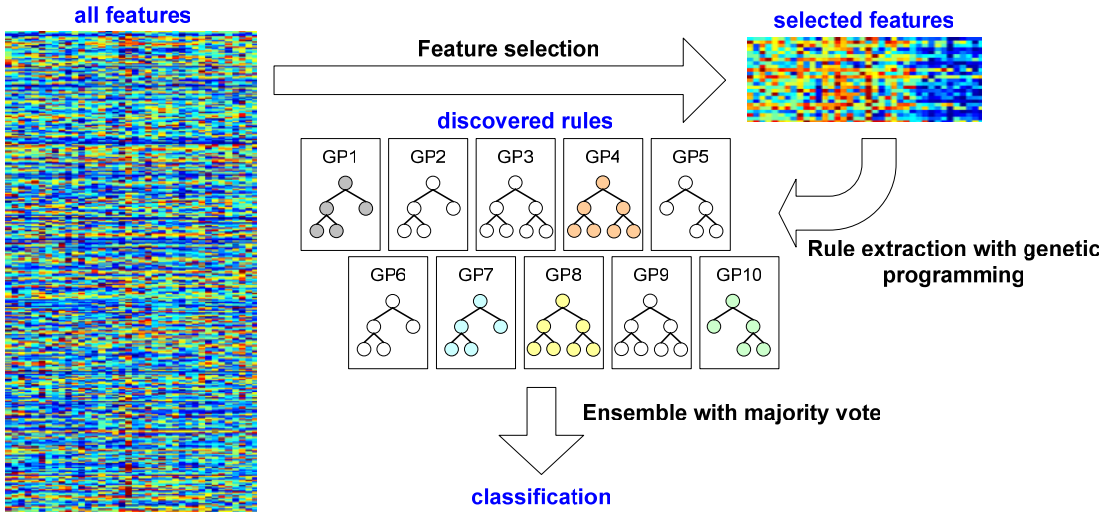

Figure 4 Overview of the proposed method

4.2 Feature selection

In general, microarrays include the expression information of thousands or even tens of thousands of genes. That kind of huge scale data might be redundant and discriminable in many real world applications contrary to theoretical ideas [22]. Cutting down the number of features to a sufficient minimum is requisite to the improvement of classification. Feature selection, sometimes called as gene selection, may select a subset of genes, which might be involved in the pathways or biological interpretation. The paper employs three popular rank-based feature selection methods such as Euclidean distance, cosine coefficient and signal-to-noise ratio. Thirty genes are selected by each feature selection methods and used in the rule discovery process.

The similarity between two input vectors X and Y can be regarded as a distance, while the distance presents how far they are located in. The distance between gi and gideal_c1 tells us how similar the gi is to class 1. If it is larger than a threshold, the gene gi would belong to class 1, otherwise gi belong to class 0. In this paper, Euclidean distance (ED) and cosine coefficient (CC) are used as follows:

$$ED = \sqrt{\sum (X - Y)^2} \tag{1}$$

$$CC = \frac{\sum XY}{\sqrt{\sum X^2 \sum Y^2}} \tag{2}$$

Given the mean μ and standard deviation σ from the distribution of gene expressions within their classes, the signal to noise ratio of gene gi, SN(gi), is defined as:

$$SN(g_i) = \frac{\mu_{c1}(g_i) - \mu_{c0}(g_i)}{\sigma_{c1}(g_i) + \sigma_{c0}(g_i)} \tag{3}$$

4.3 Multiple rules discovery

In order to generate multiple classification rules, a set of genetic programming as shown in figure 4 is performed in parallel. Each genetic programming obtains a classification rule that

consists of a subset of features and arithmetic operators. Each genetic programming evolves using overall training samples. Four fifth of whole training data is randomly selected to construct a training set for evolving a rule. The process of generating a classification rule is described in Section 3.

In evolution process, genetic programming evaluates individuals in the accuracy of classification, while it also considers the simplicity of rules. The concept of Occam's razor also supports the introduction of simplicity [23]. The accuracy is estimated as correct classification rate for training samples, and the simplicity is measured as the number of nodes used in a rule. The following formula show the fitness function used in this paper, and the weights for each criterion are set as 0.9 and 0.1, respectively.

$$fitness\ \ of\ \ individual_i = \frac{\text{number of correct samples}}{\text{number of total train data}} \times w_1 + simplicity \times w_2$$

$$\text{where}\ \ simplicity = \frac{\text{number of nodes}}{\text{number of maximum nodes}},$$

$$w_1 = \text{weight for training rate, and}\ w_2 = \text{weight for simplicity}$$

4.4 Diversity based rule selection

Selecting a subset of attributes is also benefit for learning diverse classifiers as well as constructing a training set dynamically [2]. The classification rules obtained by genetic programming have different structures and use different genes. It signifies that the parallel genetic programming might naturally generate diverse rules by selecting different sets of attributes [24].

```
R: A set of extracted rules {r1, r2, ..., r10}
S: A set of selected rules { s1, s2, ..., s5}

// Calculate similarity
For i=1 to 10 {
For j=i+1 to 10 {
sij = calculate_similarity(ri, rj);
}}
// Select 5 diverse classification rules
SR = sorting(sij);
sij = SR(first);
While k≤5 {
if(i ∉ S)
  include i into S
  k++;
if(j ∉ S)
  include j into S
  k++;
sij = SR(next);
}
```

Figure 5. An algorithm for selecting diverse classification rules

Before combining classification rules obtained in the previous stage, the diversity is measured by the edit distance of structures and the appearance of genes. The structure of a rule is represented into an in-order string, and the edit distance between two rules is calculated for all pairs of rules. Genes used in rules are also compared with each other, so if

there are same genes, the diversity of them is decreased. A good ensemble can be made when base classifiers are distinct from one another, so some classification rules are selected to compose an ensemble classifier from 10 rules by the algorithm described in figure 5. Since some fusion methods might result in a tie, the number of selected classification rules is set as 5 in this paper.

4.5 Fusion method

Two fusion methods are used: Summation and majority vote. Since the associated values of a classification rule for samples imply the degree of belonging to a category, we simply sum the values from five classification rules to make a final output as shown in the following formula.

$$Fusion\_value(sample_j) = \sum_{i=1}^{5} eval_{Individual\_i}(sample_j)$$

$$\text{IF}(Fusion\_value(sample_j) > 0)\ \text{THEN}\ class1$$

$$\text{ELSE IF}(Fusion\_value(sample_j) < 0)\ \text{THEN}\ class2$$

$$\text{ELSE reject to make a decision}$$

Majority vote is a popular fusion method, which will assign the right class label, wi, to a sample x if at least (Total number of rules)/2 classification rules vote for wi [1]. The following formula describes how to make a final output using majority vote.

$$\text{FOR}(i = 1;\ i < 5;\ i++)\ \{$$

$$\quad \text{IF}\ (eval_{Individual\_i}(sample_j) > 0)\ \text{THEN}\ result++;$$

$$\}$$

$$\text{IF}(result > 2.5)\ \text{THEN}\ class1$$

$$\text{ELSE}\ class2$$

5. Experimental results

5.1 Experimental environment

Two popular datasets are used in this study: Lymphoma cancer (http://limpp.nih.gov/lymphoma) and GCM cancer (http://www-genome.wi.mit.edu/MPR/GCM.html). Both of them are normalized from 0 to 1 in advance.

Diffuse large B-cell lymphoma (DLBCL) is one disease, which is a subtype of non-Hodgkin's lymphoma [25]. There are various subtypes of lymphoma cancer needed different treatments, but it is not easy to distinguish them clinically. Hence, research on lymphoma cancer classification using gene expression profiles has been investigated [26]. The dataset consists of 47 samples: 24 samples of GC B-like and 23 samples of activated B-like. Each sample has 4,026 gene expression levels.

GCM dataset is originally composed of 14 different cancer classes and normal tissue [27]. Since the proposed method targets on categorizing into 2 classes, we use a special dataset consisting of 2 classes: tumor and normal. Total 190 tumor samples and 90 normal tissue samples are employed to evaluate the proposed method. Each sample has 16,063 gene expression levels.

Since each dataset consists of few samples with many features, we conduct 5 folds cross-validation. One fifth samples are evaluated as test data while the others are used as train data, and it is repeated 10 times for the average results, resulting in total 50 (5×10)

experiments. The parameters for genetic programming are set as shown in Table 2. We use roulette wheel selection with elite preserving strategy.

Table 2. Experimental environments

Parameter	Value	Parameter	Value
Population size	100	Mutation rate	0.1~0.3
Maximum generation	50,000	Permutation rate	0.1
Selection rate	0.6~0.8	Maximum depth of a tree	3~5
Crossover rate	0.6~0.8	Elitism	yes

5.2 Result analysis

Table 3 summarizes the predictive accuracy of the proposed method comparing with a single classification rule; the highlighted values represent the highest accuracy obtained by the method. The result shows that the ensemble of classification rules increases the performance of classification. The overall performance of the proposed method is better than that of the single rule as shown in figure 6.

Table 3. The predictive accuracy of the proposed method

Dataset	FS method	Single rule	Summation ensemble	Voting ensemble
Lymphoma	CC	94.1%	**98.3%**	97.9%
	ED	94.0%	98.3%	**98.7%**
	SN	93.7%	**98.9%**	**98.9%**
GCM	CC	90.4%	91.1%	**92.9%**
	ED	90.6%	91.4%	**92.5%**
	SN	88.3%	89.3%	**90.7%**

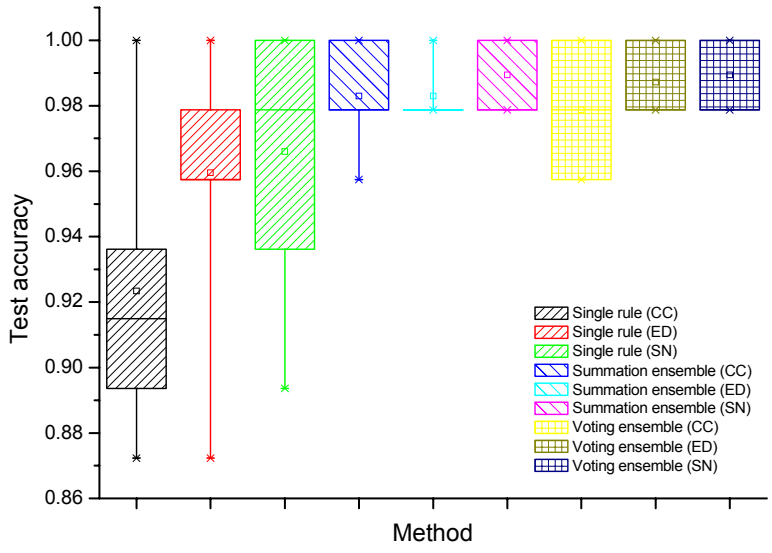

Figure 6. Predictive accuracy on Lymphoma cancer data

Maximizing the diversity of rules by the proposed method has obtained higher performance than the conventional bagging ensemble learning on Lymphoma cancer dataset as shown in figure 7. Not only the diversity on output patterns of classification rules, but also

the diversity on the representation of classification rules may result a good ensemble classifier. Figure 8 presents the diversity among rules of common bagging and the proposed method. An example ensemble classifier obtained by the proposed method is shown in figure 9.

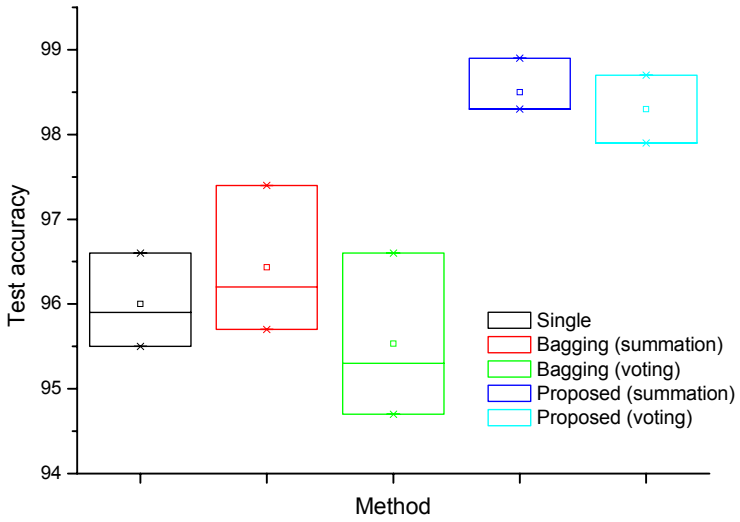

Figure 7. Comparision with the conventional ensemble

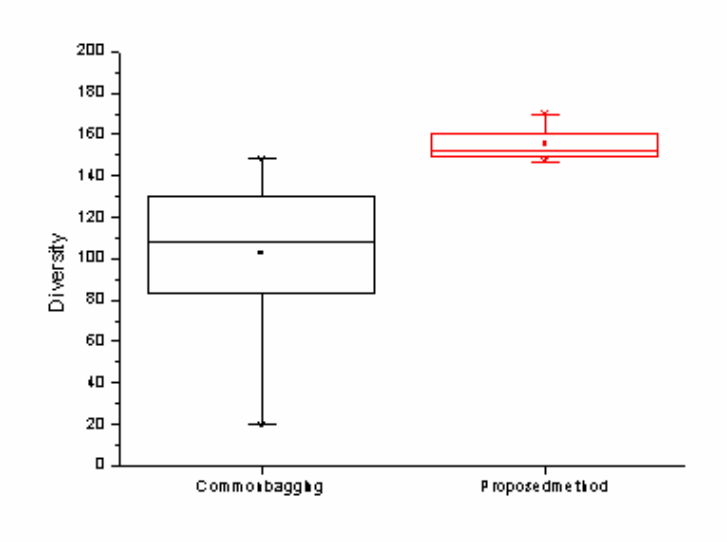

Figure 8. Comparing the methods in terms of diversity

6. Conclusion

In this paper, we have proposed an effective ensemble method for genetic programming. Since gene expression data is composed of a few samples having a number of features, feature selection is applied to reduce the dimensionality. Then, genetic programming generates various classification rules with arithmetic operators based on the gene selected. The classification rules obtained by genetic programming might be comprehensive so as to be possible to directly estimate the similarity between them. Contrary to conventional ensembel learning, the proposed method maximizes the diversity of base classifiers by the direct distance calculation. After all, a fusion method combines those various rules.

We have applied the proposed method to cancer classification using gene expression. Especially, Lymphoma cancer dataset and GCM cancer dataset have been employed for the demonstration. The proposed ensemble method using genetic programming obtained higher performance than a individual classification rule as presented in the result. Moreover, the experiments show that the diversity calculated by directly matching representations of rules increases the performance of ensembling.

As the future work, we will compare the method with various conventional ensemble learning methods such as Arcing, Ada-boosting, attibute bagging, etc. The other popular benchmark datasets in bioinformatics will be investigated with the proposed method.

Acknowledgements

This work was supported by Brain Science and Engineering Research Program sponsored by Korean Ministry of Science and Technology.

References

[1] L. Kuncheva, "A theoretical study on six classifier fusion strategies," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 281-286, 2002.

[2] R. Bryll et al., "Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets," *Pattern Recognition*, vol. 36, no. 6, pp. 1291-1302, 2003.

[3] L. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993-1001, 1990.

[4] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *J. of Artificial Intelligence Research*, vol. 11, pp. 160-198, 1999.

[5] Z. Zhou et al., "Ensembling neural networks : Many could be better than all," *Artificial Intelligence*, vol. 137, no. 1-2, pp. 239-263, 2002.

[6] D. Ruta and B. Gabrys, "Classifier selection for majority voting," *Information Fusion*, 2004.

[7] G. Brown et al., "Diversity creation methods: a survey and categorization," *Information Fusion*, 2004.

[8] B. Bakker and T. Heskes, "Clustering ensembles of neural network models," *Neural Networks*, vol. 16, no. 2, pp. 261-269, 2003.

[9] A. Tan and D. Gilbert, "Ensemble machine learning on gene expression data for cancer classification," *Applied Bioinformatics*, vol. 2, no. 3, pp. 75-83, 2003.

[10] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123-140, 1996.

[11] Y. Freund and R. Schapire, "Experiments with a new boosting algorithm," *Proc. Of the 13th Int. Conf. on Machine Learning*, pp. 148-156, 1996.

[12] L. Breiman, "Bias, variance, and arcing classifiers," *Tech. rep. 460, UC-Berkeley*, 1996.

[13] J.-H. Hong and S.-B. Cho, "Lymphoma cancer classification using genetic programming with SNR features," *Lecture Notes in Computer Science*, vol. 3003, pp. 78-88, 2004.

[14] H. Zhang et al., "Cell and tumor classification using gene expression data: Construction of forests," *Proc. of the National Academy of Sciences of the United States of America*, vol. 100, no. 7, pp. 4168-4172, 2003.

[15] C.-H. Park and S.-B. Cho, "Evolutionary computation for optimal ensemble classifier in lymphoma cancer," *Lecture Notes in Computer Science*, vol. 2871, pp. 539-543, 2003.

[16] Y. Lu and J. Han, "Cancer classification using gene expression data," *Information Systems*, vol. 28, no. 4, pp. 243-268, 2003.

[17] R. Wooster, "Cancer classification with DNA microarrays: Is less more?," *Trends in Genetics*, vol. 16, no. 8, pp. 327-329, 2000.

[18] K. Tan et al., "Evolutionary computing for knowledge discovery in medical diagnosis," *Artificial Intelligence in Medicine*, vol. 27, no. 2, pp. 129-154, 2003.

[19] C. Zhou et al., "Discovery of classification rules by using gene expression programming," *Proc. of the 2002 Int. Conf. on Artificial Intelligence*, pp. 1355-1361, 2002.

[20] I. Falco et al., "Discovering interesting classification rules with genetic programming," *Applied Soft Computing*, vol. 1, no. 4, pp. 257-269, 2002.

[21] J. Kishore et al., "Application of genetic programming for multicategory pattern classification," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 242-258, 2000.

[22] M. Xiong et al., "Feature selection in gene expression-based tumor classification," *Molecular Genetics and Metabolism*, vol. 73, no. 3, pp. 239-247, 2001.

[23] M. Brameier and W. Banzhaf, "A comparison of linear genetic programming and neural networks in medical data mining," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 1, pp. 17-26, 2001.

[24] Y. Zhang and S. Bhattacharyya, "Genetic programming in classifying large-scale data: an ensemble method," *Information Sciences*, vol. 163, no. 1-3, pp. 85-101, 2004.

[25] M. Shipp et al., "Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning," *Nature Medicine*, vol. 8, no. 1, pp. 68-74, 2002.

[26] T. Ando et al., "Selection of causal gene sets for lymphoma prognostication from expression profiling and construction of prognostic fuzzy neural network models," *J. of Bioscience and Bioengineering*, vol. 96, no. 2, pp. 161-167, 2003.

[27] S. Ramaswamy et al., "Multiclass cancer diagnosis using tumor gene expression signature," *Proc. The Natl. Acad. Sci.*, vol. 98, pp. 15149-15154, 2001.

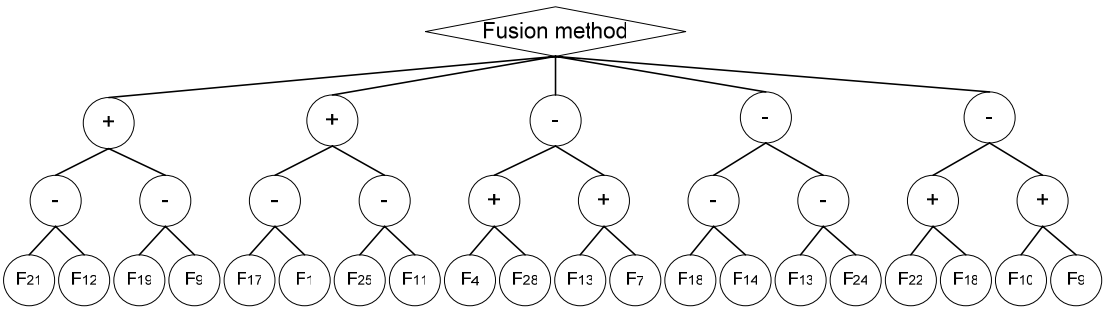

Figure 9. An ensemble classifier by the proposed method