Cancer Prediction Using Diversity-Based Ensemble Genetic Programming

Jin-Hyuk Hong and Sung-Bae Cho

Dept. of Computer Science, Yonsei University
134 Sinchon-dong, Sudaemoon-ku, Seoul 120-749, Korea
hjinh@sclab.yonsei.ac.kr, sbcho@cs.yonsei.ac.kr

Abstract. Combining a set of classifiers has often been exploited to improve the classification performance. Accurate as well as diverse base classifiers are prerequisite to construct a good ensemble classifier. Therefore, estimating diversity among classifiers has been widely investigated. This paper presents an ensemble approach that combines a set of diverse rules obtained by genetic programming. Genetic programming generates interpretable classification rules, and diversity among them is directly estimated. Finally, several diverse rules are combined by a fusion method to generate a final decision. The proposed method has been applied to cancer classification using gene expression profiles, which is one of the important issues in bioinformatics. Experiments on several popular cancer datasets have demonstrated the usability of the method. High performance of the proposed method has been obtained, and the accuracy has increased by diversity among the base classification rules.

1 Introduction

Genetic programming is a representative technique in evolutionary computation, which has several distinguished characteristics [1]. Especially, interpretable rules obtained by genetic programming provide not only useful information on classification but also many chances to combine with other approaches. Diversity that is important in ensemble might be directly estimated by comparing the rules [2].

Combining classifiers, known as ensemble, has received the attention to improve classification performance [3,4]. The ensemble classifier is obtained by combining the outputs of multiple classifiers, and the diversity among base classifiers is important besides the accuracy. Diversity implies how differently classifiers are formed, while accuracy represents how correctly a classifier categorizes. Many researchers have studied ensemble techniques and diversity measures. Hansen and Salamon have provided the theoretical basis on ensemble [5], while Opitz and Maclin have performed empirical ensemble experiments comprehensively [6]. Zhou *et al.* have analyzed the effect on the number of participating classifiers into ensemble in both theoretical and empirical studies [7]. 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 [3,4,8]. A survey on generating diverse classifiers for ensemble has been conducted by Brown [9]. A hybrid model for efficient ensemble was studied by Bakker and Heskes [10], while Tan and Gilbert applied ensemble to classifying gene expression data [11].

Since ensembling the same classifiers does not produce any elevation on performance [8], selecting diverse as well as accurate base classifiers is very important in

V. Torra et al. (Eds.): MDAI 2005, LNAI 3558, pp. 294–304, 2005.

making a good ensemble classifier [9]. Simple ways to generate various classifiers are randomly initializing parameters or making a variation of training data. Bagging (bootstrap aggregating) introduced by Breimen generates individual classifiers by training with a randomly organized set of samples from the original data [12]. 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 [13]. 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 one. Arching [14] and Ada-Boosting [13] are the representative boosting methods.

Some researchers select diverse classifiers to combine for ensemble [3,8]. Diversity among classifiers is estimated by some measures, and the most discriminating classifiers are selected to make an ensemble classifier. In general, the error patterns of classifiers are used to measure the diversity [9]. Bryll proposed a novel approach that employs different sets of features to generate different classifiers [4]. Most studies aim at generating distinct base classifiers, but they hardly provide explicit methods to measure diversity of classifiers and errors might be included into selecting classifiers. An explicit method estimating the diversity among classifiers might be helpful to minimize errors and to prepare a set of diverse base classifiers.

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

2 Cancer Classification Using Genetic Programming

Cancer classification based on gene expression profiles is one of the major research topics both in the medical field and in machine learning. DNA microarray technology recently developed provides an opportunity to take a genome-wide approach to the correct prediction of cancers. It captures the expression levels of thousands of genes simultaneously which contain information on diseases [15]. Since finding an understandable classification rule is required beside the accuracy, discovering classification rules using genetic programming was studied in the previous work [16]. Even though the rule is quite simple, it shows a good performance in classifying cancers.

An individual of genetic programming is represented as a tree that consists of the function set {+, –, ×, /} and the terminal set {f_1, f_2, …, f_n, constant} where n is the number of features. The function set is designed to model the up and down regulations of the gene expression. The grammars for the classification rule are: G={V={EXP, OP, VAR}, T={+, –, ×, /, f_1, f_2, …, f_n, constant}, P, {EXP}}, where the rule set P is as the following.

$$\begin{aligned}&\text{EXP}\rightarrow\text{EXP OP EXP} \mid \text{VAR}\\ &\text{OP}\rightarrow + \mid - \mid \times \mid /\\ &\text{VAR}\rightarrow f_1 \mid f_2 \mid \ldots \mid f_n \mid \text{constant}\end{aligned}$$

The category of an instance is determined by evaluating it with the rule. An instance will be classified into class 1 if the evaluated value is over 0, while it will be classified into class 2 if the value is under 0. Conventional genetic operators for genetic programming are employed for evolution. Crossover randomly selects and changes sub-trees from two individuals, mutation changes a sub-tree into new one, and permutation exchanges two sub-trees of an individual. All genetic operations are conducted according to the predefined probabilities.

3 Combining Classification Rules with Diversity

The proposed method consists of 3 processes as shown in Figure 1: selecting features, discovering multiple rules, and selecting and combining the rules. 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 cancers. With the selected genes, genetic programming works to generate multiple classification rules. Diversity among these rules is estimated by the tree edit distance, and a subset of diverse classification rules is used to construct an ensemble classifier. Contrary to conventional ensemble learning methods that simply combine the outputs of individual classifiers, the proposed method picks up some classification rules that maximize diversity.

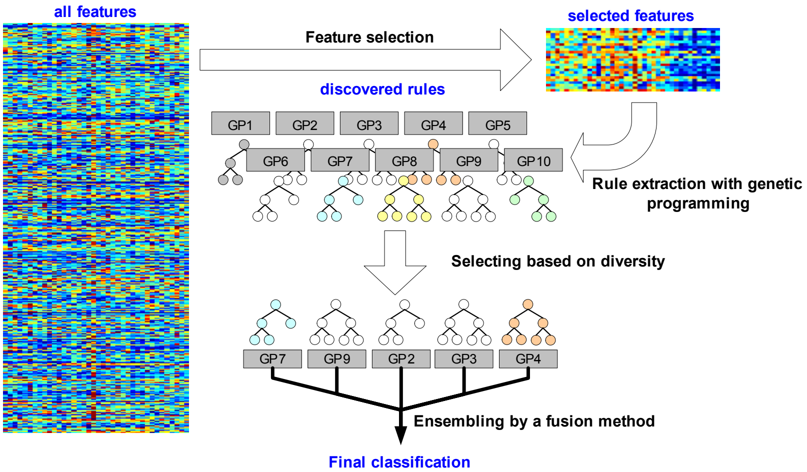

Fig. 1. Overview of the proposed method

3.1 Estimating Diversity in Genetic Programming

Diversity is concerned with the levels and types of variety between individuals in genetic programming. Features such as fitness values and structures are employed to design diversity measures [2]. Moreover, diversity of the population can be controlled during evolution so as to generate diverse individuals.

In genetic programming, diversity often refers to structural differences. Two identical structures might produce the same result, but this does not imply that the two structures with the same results are identical. Even if two individuals with different structures have the same outputs, they should be regarded as different ones possessing different potential characteristics. There are some representative methods for estimating diversity between two individuals such as edit distance, isolated-subtree distance, top-down distance, and alignment distance [17]. Edit distance, one of the most popular methods, measures the similarity between two individuals. It scores a distance of 1 if the nodes have different values, or 0. After scoring for all nodes in two trees, it sums up the distances and normalizes by dividing it by the size of the smaller tree. Sometimes a tree is interpreted as a string by a specific strategy, and the matching score between them is used to employ as the similarity of them.

3.2 Selecting Features

In general, microarrays include the expression information of thousands or even tens of thousands of genes, but only a small portion of them are related to the target cancer. A subset of informative genes should be selected by the feature selection process. Cutting down the number of features to a sufficient minimum is requisite to improve classification performance [18]. Two “ideal” marker gene expression profiles are designed as shown in Figure 2. The first one is a binary vector which is 1 among all the samples in class A and 0 among all the samples in class B, while the second one is another binary vector which is 0 among all the samples in class A and 1 among all the samples in class B. Three popular measures are employed such as Euclidean distance, cosine coefficient and signal-to-noise ratio. Fifty genes are selected by each feature selection method: the first 25 for ideal marker 1 and the rest for ideal marker 2.

The similarity between an ideal marker *ideal* and a gene *g* can be regarded as a distance, while the distance presents how far they are located in. A gene is regarded as an informative gene if the distance is small, while the gene is regarded as an uncorrelated gene if the distance is large. Euclidean distance (ED) and cosine coefficient (CC), where *n* is the number of samples, estimate the distance as follows:

$$ED = \sqrt{\sum_{i=1}^{n} \left(ideal_i - g_i\right)^2} \tag{1}$$

$$CC = \frac{\sum_{i=1}^{n} ideal_i \times g_i}{\sqrt{\sum_{i=1}^{n} ideal_i^{\,2} \times \sum_{i=1}^{n} g_i^{\,2}}} \tag{2}$$

Given the mean □ and standard deviation □ from the distribution of gene expressions within their classes, the signal to noise ratio of a gene *g*, SN, is defined as follows:

$$SN = \frac{\mu_{class\,A}(g) - \mu_{class\,B}(g)}{\sigma_{class\,A}(g) + \sigma_{class\,B}(g)}, \tag{3}$$

where $\mu_{class\,i}(g)$ is the mean of *g* and *ideal* whose label is *class i*

and $\sigma_{class\,i}(g)$ is the standard deviation of *g* and *ideal* whose label is *class i*

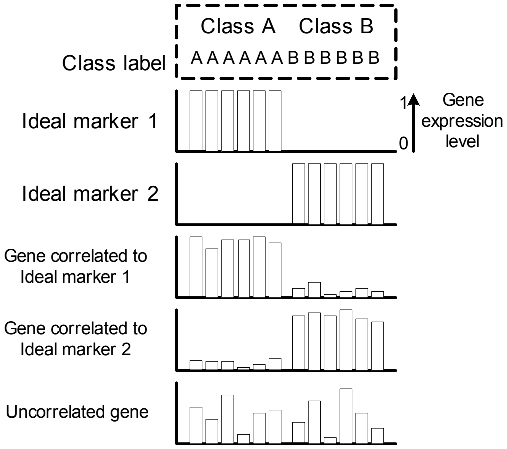

Fig. 2. Illustration of the feature selection

3.3 Generating Multiple Classification Rules

In order to generate multiple classification rules, genetic programming operates in parallel as shown in Figure 1. Each genetic programming obtains a classification rule that consists of a subset of features and arithmetic operators. Four fifths of the whole training data is randomly selected to construct a training set for evolving a rule. Generating a classification rule was described in Section 2.

In evolution process, genetic programming evaluates individuals in classification accuracy, while it also considers the simplicity of rules. The concept of Occam's razor also supports the introduction of simplicity [19]. 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.

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

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

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

3.4 Selecting Diverse Rules

Selecting a subset of attributes is also benefit for learning diverse classifiers as well as constructing a training set dynamically [4]. 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 and structures [20]. Before combining classification rules, diversity is measured by the edit distance of structures and the appearance of genes used. The edit distance between the structures of two rules r_i and r_j is estimated as follows:

$$edit\_distance(r_i, r_j) = \begin{cases} d(p,\, q), & \text{if neither } r_i \text{ nor } r_j \text{ have any children} \\ d(p,\, q) + edit\_distance(\text{RS of } r_i, \text{RS of } r_j) & \\ \qquad + edit\_distance(\text{LS of } r_i, \text{LS of } r_j), & \\ & \text{otherwise (RS : right subtree, LS : left subtree)} \end{cases}$$

$$where\ \ d(p,\, q) = \begin{cases} 1, & \text{if } p \text{ and } q \text{ overlap} \\ 0, & \text{if } p \text{ and } q \text{ do not overlap} \end{cases}$$

The appearance of genes in rules is also compared with each other. The diversity decreases when there is the same gene, while it increases if different genes are used. 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 3. Since some fusion methods might result in a tie, 5 rules are selected for ensemble in this paper.

```
$R$: A set of extracted rules $\{r_1, r_2, \ldots, r_{10}\}$
$S$: A set of selected rules $\{s_1, s_2, \ldots, s_5\}$

int calculate_diversity($r_i$, $r_j$) {
    $cf_{ij}$ = common_feature_number($r_i$, $r_j$);
    $df_{ij}$ = different_feature_number($r_i$, $r_j$);
    $ed_{ij}$ = edit_distance($r_i$, $r_j$);
    return $df_{ij} - cf_{ij} + 0.5 \times ed_{ij}$;
}
For $i$=1 to 10 {
    For $j$=$i$+1 to 10 {
        $d_{ij}$ = calculate_diversity($r_i$, $r_j$);
}}
Find a set $S$ in which rules' diversity is maximized
$S = \{s_1, s_2, \ldots, s_5\}$
```

Fig. 3. An algorithm for selecting 5 diverse classification rules

3.5 Combining Multiple Classification Rules

Four fusion methods are used: Majority vote (MAJ), maximum (MAX), minimum (MIN) and average (AVG) [3]. They are described as the following formula, where m_i is the margin of the classifier i.

MAJ $$\#\text{ of classifiers selecting class1} > \#\text{ of classifiers selecting class2 ? class1 : class2} \tag{4}$$

MAX $$\text{ABS\_MAX}(m_1, m_2, m_3, m_4, m_5) > 0 \text{ ? class1 : class2} \tag{5}$$

MIN $$\text{ABS\_MIN}(m_1, m_2, m_3, m_4, m_5) > 0 \text{ ? class1 : class2} \tag{6}$$

AVG $$\sum_{i=1}^{5} m_i > 0 \text{ ? ? class 1 : class2} \tag{7}$$

4 Experimental Results

4.1 Experimental Environment

Three popular gene expression datasets are used in this paper: Types of diffuse large B-cell lymphoma cancer dataset [21], lung cancer dataset [22], and ovarian cancer dataset [23]. All of them are normalized from 0 to 1 at first.

Diffuse large B-cell lymphoma (DLBCL) is one disease, which is the common subtype of non-Hodgkin's lymphoma [21]. There are various subtypes of lymphoma cancer needed different treatments, but it is not easy to distinguish them clinically. Hence, lymphoma cancer classification using gene expression profiles has been investigated [24,25]. The dataset consists of 47 samples: 24 samples of germinal centre B-like group and 23 samples of activated B-like group. Each sample has 4,026 gene expression levels.

Lung cancer dataset has been exploited in classifying between malignant pleural mesothelioma (MPM) and adenocarcinoma (ADCA) of the lung. There are 181 tissues: 31 MPM tissues and 150 ADCA tissues. Each tissue has 12,533 gene expression levels [22].

Ovarian cancer dataset aims to identify proteomic pattern in serum so as to distinguish the ovarian cancer. It has 91 controls (normal) and 162 ovarian cancer tissues where each sample has 15,154 gene expression levels [23].

Since each dataset consists of few samples with many features, we conduct 5 folds cross-validation. One fifth of samples are evaluated as test data while the others are used as training data, and it is repeated 10 times for the average results, leading to 50 (5×10) experiments in total. The parameters for genetic programming are set as shown in Table 1. We use roulette wheel selection with elite preserving strategy.

Table 1. Experimental environments

Parameter	Value	Parameter	Value
Population size	200	Mutation rate	0.1~0.3
Maximum generation	3,000	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

4.2 Classification Accuracy

Table 2~4 summarize the predictive accuracy of the proposed method for each dataset; the highlighted values represent the highest accuracy obtained. The result shows that the ensembling improves the performance of classification, while the proposed method that considers diversity obtains the highest accuracy in most cases against the combination of 10 classifiers and the combination of 5 classifiers. The ensemble that uses 10 rules was inferior to the ensemble that uses 5 rules, even though the former procedure includes more information than the latter. This implies that error is increased with increasing numbers of base classifiers. Finally, the proposed approach not only supports the same degree of useful information with the ensemble that uses 10 rules, but also minimizes the increment of the error.

Table 2. Test accuracy on lymphoma cancer dataset (%)

Features	Fusion method	10 classifiers	5 classifiers	5 diverse classifiers	Individual classifier
ED	MAJ	92.2	92.9	**94.4**	88.9
	MAX	96.8	95.7	**100**	
	MIN	78.2	80.1	81.6	
	AVG	96.8	95.4	**98.9**	
CC	MAJ	92.2	93.7	**100**	91.3
	MAX	96.7	95.5	**100**	
	MIN	76.3	82.5	84.3	
	AVG	95.6	94.9	**100**	
S2N	MAJ	95.6	95.4	**99.7**	89.7
	MAX	97.8	96	**99.1**	
	MIN	74.1	80	**96.1**	
	AVG	98.9	96.5	**99.1**	

Table 3. Test accuracy on lung cancer dataset (%)

Features	Fusion method	10 classifiers	5 classifiers	5 diverse classifiers	Individual classifier
ED	MAJ	97.8	**98.3**	97.4	97.5
	MAX	**99.2**	98.9	99	
	MIN	94.2	**95.7**	93.9	
	AVG	**99.4**	98.8	98.2	
CC	MAJ	99.2	99.1	**99.9**	97.8
	MAX	98.9	98.9	**99.4**	
	MIN	94.5	**95.9**	95.8	
	AVG	99.4	99.2	**99.9**	
S2N	MAJ	**99.7**	99.6	99.6	99
	MAX	99.4	**99.5**	99.4	
	MIN	95.3	**96.7**	96.5	
	AVG	**100**	99.8	**100**	

Table 4. Test accuracy on ovarian cancer dataset (%)

Features	Fusion method	10 classifiers	5 classifiers	5 diverse classifiers	Individual classifier
ED	MAJ	96.6	96.8	**97.5**	96.4
	MAX	**97.6**	97	97.3	
	MIN	95.1	95.4	**96.1**	
	AVG	97	96.8	**97.3**	
CC	MAJ	89.3	89.2	**92.6**	87.7
	MAX	90.3	90.1	**91.5**	
	MIN	80.6	83.7	83.4	
	AVG	89.7	89.8	**92.6**	
S2N	MAJ	98.6	98.9	**99.9**	98.5
	MAX	99	99	**99.9**	
	MIN	97.2	97.9	**99.3**	
	AVG	99.2	99	**99.9**	

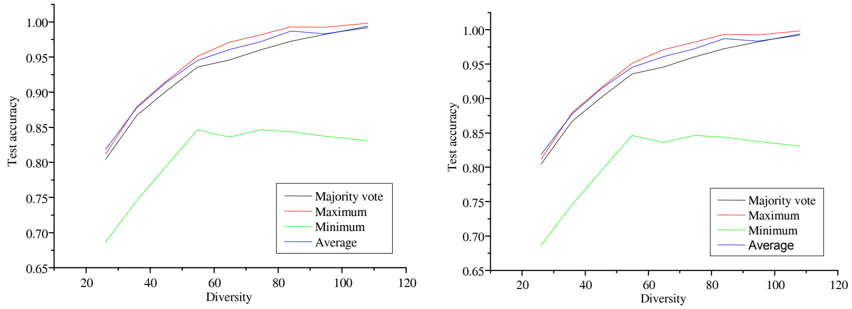

Fig. 4. Test accuracy for diversity for lymphoma cancer (left: CC and right: S2N)

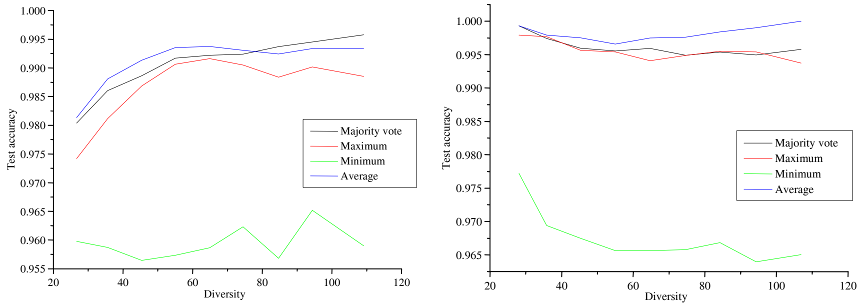

Fig. 5. Test accuracy for diversity for lung cancer (left: CC and right: S2N)

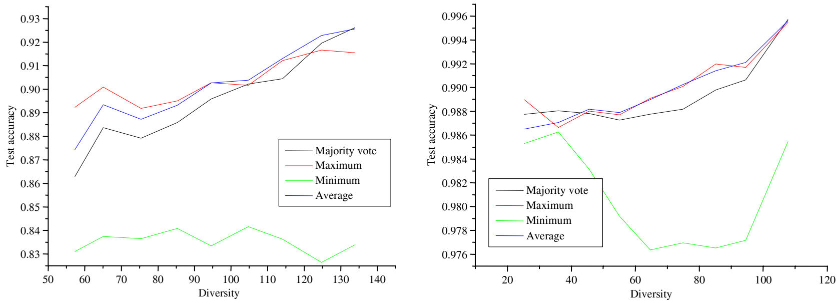

Fig. 6. Test accuracy for diversity for ovarian cancer (left: CC and right: S2N)

4.3 Diversity Performance

The relationship between diversity and performance is also analyzed and shown in Fig. 4-6. The results indicate that classification accuracy increases according to the increment of diversity in most cases. A decline in accuracy occasionally appears, because diversity is apt to increase when there is a peculiar rule. This can be solved by a non-pair-wise approach for estimating diversity in ensemble genetic programming. MIN often selects poor rules among the diverse rules, while the others use mutually cooperative rules from the rule set.

5 Conclusion

In this paper, we have proposed an effective ensemble method with 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 genes selected. The classification rules might be comprehensive so as to be possible to directly estimate diversity between them. Contrary to the conventional ensemble learning, the proposed method selects a set of base classification rules whose diversity is maximized. After all, a fusion method combines the diverse rules selected as shown in Figure 7.

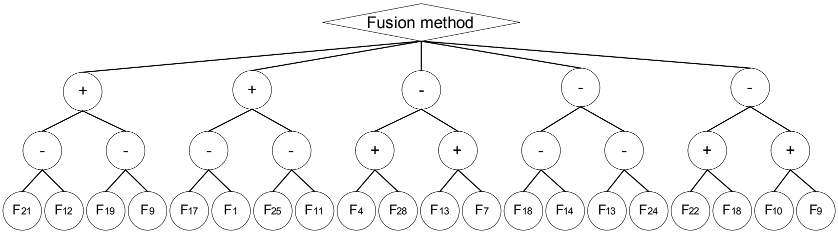

Fig. 7. An ensemble classifier obtained by the proposed method

We have applied the proposed method to cancer classification using gene expression. Especially, 3 cancer datasets have been employed for the demonstration. The proposed ensemble method using genetic programming produces higher performance than the others as presented in the results. 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 diversity measures, and extend it by combining ensemble learning methods such as Arcing, Ada-boosting, attribute bagging, etc. Other popular benchmark datasets in bioinformatics will be also investigated with the proposed method.

Acknowledgement

This work was supported by Brain Science and Engineering Research Program sponsored by Korean Ministry of Commerce, Industry and Energy.

References

1. J. Koza, “Genetic programming,” *Encyclopedia of Computer Science and Technology*, vol. 39, pp. 29-43, 1998.
2. E. Bruke, et al., “Diversity in genetic programming: An analysis of measures and correlation with fitness,” *IEEE Trans. Evolutionary Computation*, vol. 8, no. 1, pp. 47-62, 2004.
3. L. Kuncheva, “A theoretical study on six classifier fusion strategies,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 281-286, 2002.
4. 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.

5. L. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993-1001, 1990.
6. D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *J. of Artificial Intelligence Research*, vol. 11, pp. 160-198, 1999.
7. Z. Zhou, et al., "Ensembling neural networks : Many could be better than all," *Artificial Intelligence*, vol. 137, no. 1-2, pp. 239-263, 2002.
8. D. Ruta and B. Gabrys, "Classifier selection for majority voting," *Information Fusion*, 2004.
9. G. Brown, et al., "Diversity creation methods: A survey and categorization," *Information Fusion*, vol. 6, no. 1, pp. 5-20, 2005.
10. B. Bakker and T. Heskes, "Clustering ensembles of neural network models," *Neural Networks*, vol. 16, no. 2, pp. 261-269, 2003.
11. 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.
12. L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123-140, 1996.
13. Y. Freund and R. Schapire, "Experiments with a new boosting algorithm," *Proc. the 13th Int. Conf. Machine Learning*, pp. 148-156, 1996.
14. L. Breiman, "Bias, variance, and arcing classifiers," *Tech. Rep. 460, UC-Berkeley*, 1996.
15. C. Peterson and M. Ringner, "Analyzing tumor gene expression profiles," *Artificial Intelligence in Medicine*, vol. 28, no. 1, pp. 59-74, 2003.
16. 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.
17. J. Wang and K. Zhang, "Finding similar consensus between trees: An algorithm and a distance hierarchy," *Pattern Recognition*, vol. 34, no. 1, pp. 127-137, 2001.
18. M. Xiong, et al., "Feature selection in gene expression-based tumor classification," *Molecular Genetics and Metabolism*, vol. 73, no. 3, pp. 239-247, 2001.
19. M. Brameier and W. Banzhaf, "A comparison of linear genetic programming and neural networks in medical data mining," *IEEE Trans. Evolutionary Computation*, vol. 5, no. 1, pp. 17-26, 2001.
20. 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.
21. A. Alizadeh, et al., "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling," *Nature*, vol. 403, no. 6769, pp. 503-511, 2000.
22. G. Gordon, et al, "Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma," *Cancer Research*, vol. 62, no. 17, pp. 4963-4967, 2002.
23. E. Petricoin III, et al., "Use of proteomic patterns in serum to identify ovarian cancer," *The Lancet*, vol. 359, no. 9306, pp. 572-577, 2002.
24. 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.
25. T. Ando, et al., "Selection of causal gene sets for lymphoma prognostication from expression profiling and construction of prognostic fuzzy neural network models," *J. Bioscience and Bioengineering*, vol. 96, no. 2, pp. 161-167, 2003.