Gene Expression Classification Using Optimal Feature/Classifier Ensemble with Negative Correlation

Jungwon Ryu
Yonsei University
Department of Computer Science
134 Shinchon-dong, Sudaemoon-ku, Seoul 120-749, Korea
rjungwon@candy.yonsei.ac.kr

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

Abstract – In order to predict the cancer class of patients, we have illustrated a classification framework that combines sets of classifiers trained with independent two features. Experimental results show that the feature sets that have negative or non-positive correlations produces very high recognition result.

I. INTRODUCTION

Although cancer detection and class discovery have been seriously studied over the past years, there has been no general and perfect way to work out this problem. It is because there can be 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, 2]. These gene expression profiles, however, are just simple sequences of numbers, and the necessity of tools analyzing these to get useful information has risen sharply.

DNA arrays consist of large numbers of DNA molecules spotted in a systemic order on a solid substrate. Depending on the size of each DNA spot on the array, DNA arrays can be categorized as microarrays, when the diameter of DNA spot is less than 250 microns, and macroarrays, when the diameter is bigger than 300 microns. When the solid substrate is small, arrays are also referred as DNA chips.

DNA microarrays consist 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 (equation 1) [3, 4, 5].

$$gene\_expression = \log_2 \frac{\text{Int(Cy5)}}{\text{Int(Cy3)}} \qquad (1)$$

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.

According to the divide-and-conquer principle, a complex computational task is solved by dividing it into a number of computationally simple tasks and then combining the solutions to those tasks [6]. Mixture-of-experts and committee machines are examples of this approach. In the previous work [7], in order to explore the optimal feature and classifier that efficiently detect the class of the disease, we have applied seven feature selection methods and five classifiers that are well known in the field of data mining and pattern recognition. We have realized that some of the feature selection methods are correlated, and several classifiers also commit the same mistakes on the particular samples. In this context, we have made a step further and suggest an efficient classification framework to enhance the classification performance by consisting of multiple classifiers with non-correlated or negative-correlated features.

The idea behind feature selection with negative correlation is to encourage set of classifiers, which consist of multiple classifiers, to learn different aspects of training data, so that ensembles of classifiers can search in a wide solution space. In order to classify the gene expression profile, we suggest an ensemble classifier that is composed of multiple classifiers and show the usefulness of negatively correlated features.

II. RELATED WORKS

Many have been working on the ensemble of the multiple modular neural networks. According to the Osherson's definition, a neural network is said to be modular if and only if the following statement is satisfied [8]:

> The computation performed by the network can be decomposed into two or more subsystems that operate on distinct inputs without communicating with each other.

After the notion of modular connectionist systems was first discussed in the mid of 1980's by Barto and Hinton, Pollack proposed the cascaded backpropagation architecture [9] and Jacobs developed taxonomy for a class of modular hierarchical connectionist models (hierarchical mixture-of-experts, HME) [10]. Hamshire and Waibel have proposed the Meta-Pi, which consists of a number of source-dependent sub networks that are integrated by a combinational time-delay neural network [11]. Lincoln and Skrzypek proposed clustering multiple backpropagation networks [12]. Battiti and Colla suggested the

0-7803-7278-6/02/$10.00 ©2002 IEEE

concept of democracy to combine the outputs of different neural network classifiers [13]. These early examples have shown that integrating the multiple modules, often referred as committee machines, could have enhanced the accuracy and generalization capacity.

Liu *et al.* studied evolutionary learning of neural network classifiers with negative correlation of classifiers [14, 15]. They used a penalty term in error function based on correlations between neural networks, so that classifiers learn to be negatively correlated. They also have shown that they can define the independent learning, often used in modular networks, as one of negative learning using lambda term.

In this paper, we focus on the negative correlation of features that will be used to train the ensemble classifiers.

III. FEATURE/CLASSIFIER ENSEMBLE

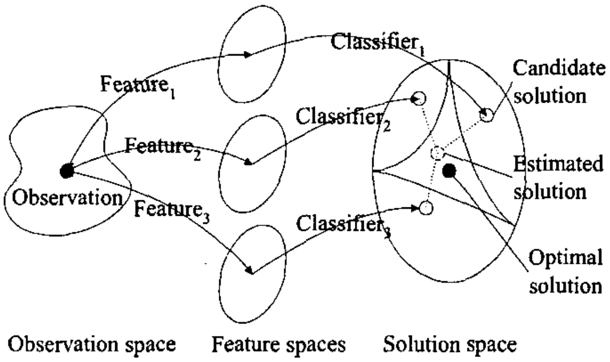


Fig. 1. Classification with multiple features

Fig. 1 illustrates the basic idea of multiple classifiers with multiple features. Classification can be defined as the process to approximate I/O mapping from the given observation to the optimal solution. Generally, classification tasks consist of two parts: feature selection and classification. Feature selection is a transformation process of observations to obtain the best pathway to get to the optimal solution. Therefore, considering multiple features encourages obtaining various candidate solutions, so that we can estimate more accurate solution to the optimal than any other local optima.

When we have multiple features available, it is important to know which of features should be used. Theoretically, as many features we may concern, it may be more effective for the classifier to solve the problems. But features that have overlapped feature spaces may cause the redundancy of irrelevant information and result in the counter effect such as overfitting. Therefore, it is more important to explore and utilize independent (negatively correlated) features to train classifiers, rather than increase the number of features we use. Correlation between feature sets can be induced from the distribution of feature numbers, or using mathematical analysis using statistics.

Meanwhile, there exists many algorithms for the classification from machine learning approach, but none of them is perfect. However, it is always difficult to decide what to use and how to set up its parameters. According to the environments the classifier is embedded, some algorithm works well and others not. It is because, according to the algorithm, feature and parameter used, the classifier searches in different solution space. These sets of classifiers produce their own outputs, and enable the ensemble classifier to explore more wide solution space.

This configuration of classifiers is different from that of HME, in that classifiers of ensemble use holistic features whereas HME uses partial information (subset of input space) when learning. These classifiers, therefore, are not modular, but they are still experts in their feature subsets.

We have applied this idea to a classification framework as shown in Fig. 2. If there exist k features, we choose the most independent features through the correlation analysis between ${}_kC_2$ possible combinations of features. Then classifiers are trained using the features selected, finally a neural network is accompanied to combine the outputs of these classifiers.

A. Feature Selection and Correlation Analysis

For efficient classification, we need to find out the informative features from input observation. This can be done by statistical tools, similarity measures, or information-theoretical methods. These are methods to score how much each feature may be informative and contain categorical information, and we finally choose k features from the top. There are three different approaches as follows:

Suppose that we have a $M \times N$ training set where M is the number of samples (input vector) and N is the number of features (dimensionality of input vector). The ith feature of samples, g_i, can be expressed:

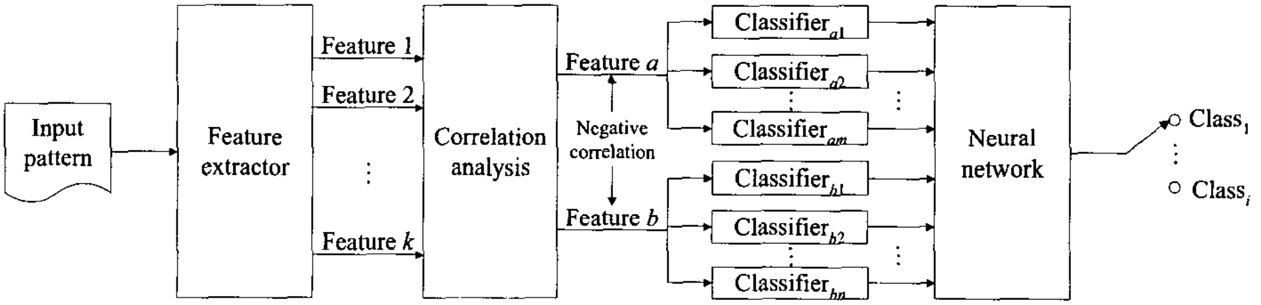


Fig. 2. Systematic diagram of feature/classifier ensemble with negative correlation

$$g_i = (e_1, e_2, e_3, \cdots, e_M) \tag{2}$$

where e is the data and i=1~N. We want to know the locations of informative k features out of M. If it is possible to know representative vector g_{ideal} for class c_j, we can simply measure the correlation and similarity of g_i to classes, which tells the feature-goodness. Modeling g_{ideal}, we should use prior knowledge and intuitional experience about classes.

Suppose g_{ideal} is an ideal vector representing class c_j.

$$g_{ideal} = (e_1', e_2', e_3', \cdots, e_M') \tag{3}$$

Cross analysis, such as Pearson's (PR) and Spearman's (SP) correlation coefficients, can measure the correlation between g_i and g_{ideal}. Coefficient can have numbers varying from –1 to +1: higher coefficient implies that g_i is much correlated to c_j.

The similarity between two variables can be thought of as the distance of those. Distance is a measure on how far the two vectors are located, and the distance between g_i and g_{ideal} implies how much g_i is likely to the class c_j. In this paper, we have adopted Euclidean distance (ED) and cosine coefficient (CC).

However, it is not always possible to know the ideal vectors for classes. Then, we have alternative ways to measure the relevance of feature using the frequency of g_i satisfying condition Q under the categorical situations. Information gain (IG) and mutual information (MI) are good examples. For equations 8 and 9, $P(g_i)$ is the number of g_i satisfying Q, $P(\bar{g}_i)$ is the number of features satisfying Q other than g_i, and $P(g_i, c_j)$ is the number of cases when g_i satisfying Q and c_j occurred simultaneously.

Moreover, signal to noise ratio (SN) measures features from the information of the distribution of features in each class. g_i is composed of two parts: one from c_j and the other from $\bar{c}_j$. When 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 is defined as equation 10. Table 1 summarizes the mathematical formula.

From the results of feature selection methods, we get a set of informative features from the data. In order to choose negatively (or not positively) correlated features, we have plotted the distribution of g_i from two feature selection methods. If the two features are negatively correlated, the distribution will be in the (-) direction, otherwise (+) direction.

TABLE I. MATHMETICAL FORMULA FOR EACH FEATURE SELECTION METHOD

Feature selection methods
$PR(g_i, g_{ideal}) = \dfrac{\sum g_i g_{ideal} - \dfrac{\sum g_i \sum g_{ideal}}{N}}{\sqrt{\left(\sum g_i^2 - \dfrac{(\sum g_i)^2}{N}\right)\left(\sum g_{ideal}^2 - \dfrac{(\sum g_{ideal})^2}{N}\right)}} \qquad (4)$
$SP(g_i, g_{ideal}) = 1 - \dfrac{6\sum(D_g - D_{ideal})^2}{N(N^2-1)} \qquad (5)$
(D_g and D_{ideal} are the rank matrices of g_i and g_{ideal})
$ED(g_i, g_{ideal}) = \sqrt{\sum(g_i - g_{ideal})^2} \qquad (6)$
$CC(g_i, g_{ideal}) = \dfrac{\sum g_i g_{ideal}}{\sqrt{\sum g_i^2 \sum g_{ideal}^2}} \qquad (7)$
$IG(g_i, c_j) = P(g_i, c_j) \log \dfrac{P(g_i, c_j)}{P(c_j) \cdot P(g_i)} + P(\bar{g}_i, c_i) \log \dfrac{P(\bar{g}_i, c_j)}{P(c_j) \cdot P(\bar{g}_i)} \qquad (8)$
$MI(g_i, c_j) = \log \dfrac{P(g_i, c_j)}{P(g_i) \cdot P(c_j)} \qquad (9)$
$SN(g_i, c_j) = \dfrac{\mu_{c_j}(g_i) - \mu_{\bar{c}_j}(g_i)}{\sigma_{c_j}(g_i) - \sigma_{\bar{c}_j}(g_i)} \qquad (10)$

B. Classifiers

Many promising machine learning algorithms successfully applied for classification problems. We have used multiplayer perceptron (MLP), support vector machine (SVM), k-nearest neighbor (KNN) as classifiers. Each classifier has been trained independently with the feature selection methods that are negatively correlated.

Multilayer perceptron (MLP) is commonly used in such field of pattern recognition due to its powerful and stable learning algorithms [16]. The power of the backpropagation algorithm on MLP 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 [17].

The Support Vector Machine (SVM) introduced by V. Vapnic is a method to estimate the function classifying the data into two classes [18, 19]. The basic idea of SVM is to construct a hyperplane as the decision surface in such a way that the margin of separation between positive and negative examples is maximized. SVM achieves this by the structural risk minimization principle that is based on the fact 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:

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

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 nonzeros. This paper has used SVM^{light} module that imports quadratic programming techniques. Linear (SVM$_{linear}$) and RBF (SVM$_{RBF}$) kernel have been also used.

KNN extracts k most closest vectors in the reference set based on similarity measures, and makes decision for the label of input vector using information of distribution and labels of k neighbors. Pearson's correlation has been used as the similarity measure. When we have an input X and a reference set $D = \{d_1, d_2, \cdots, d_N\}$, the probability that X may belong to class c_j, $P(X, c_j)$ is defined as follows:

$$P(X,c_j) = \sum_{d_i \in kNN} \mathrm{Sim}(X,d_i)P(d_i,c_j) - b_j \tag{12}$$

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

C. *Combining Outputs of Classifiers*

A neural network combines the outputs of single classifiers in our system. Outputs of single classifiers can be thought of as CSVs (classification status values), which contain information on answer patterns of classifiers. Neural network has $m+n$ (dimensionality of CSV) input nodes and j output nodes. Using neural network, we can have the adaptivity of thresholds based on the entropy as opposed to *ad hoc* and hard thresholds [20]. For the comparison, majority voting has been also used [13].

IV. EXPERIMENTS

A. *Data Set*

Dataset that we have used is a collection of expression measurements reported by Golub *et al* [1]. Gene expression profiles have been constructed from 72 people who have either acute lymphoblastic leukemia (ALL) or acute myeloid leukemia (AML). Each person has submitted one sample of DNA microarray, so that the database consists of 72 samples. Each sample is comprised of 7,129 gene expression numbers, and finally the whole database is a 7,129×72 matrix.

38 samples are for training and the other 34 are for test of the classification. The training data has 27 ALL and 11 AML samples, whereas the test data has 20 ALL and 14 AML samples.

B. *Gene Selection and Classification*

Before the classification, we need to find out genes that are related to distinction of the cancer class out of 7,129. For the use of feature selection methods summarized in Table 1, feature (gene) $g_i = (e_1, e_2, e_3, \cdots, e_{38})$ as there exist 38 samples. Suppose that the first 27 numbers $(e_1, e_2, \cdots, e_{27})$ are examples of ALL, and the other 11 $(e_{28}, e_{29}, \cdots, e_{38})$ are those from AML. Then we can define an ideal gene pattern which belongs to ALL class, called $g_{ideal\_ALL} = (1, 1, \cdots, 1, 0, \cdots, 0)$, so that all numbers from the ALL samples are 1, and the others are 0. We have measured the correlation coefficient between g_{ideal} and each gene's expression pattern, when N = 7,129 is the dimension of gene pattern.

For *IG* and *MI*, the condition Q is "if it is induced." The gene expression has positive values if the gene is more induced than the background condition, and negative values if repressed. Therefore, we simply count the number of positive and negative gene expression numbers to get $P(g_i)$ and $P(\bar{g}_i)$.

For the classifiers, we have used 3-layered MLP, with 5~15 hidden nodes, 2 output nodes, 0.03~0.5 of learning rate and 0.1~0.9 of momentum. $\gamma = 0.5$ is used for RBF kernel of SVM and KNN uses k = 1~7.

Additionally, self-organizing map (SOM), decision tree (DT) and KNN with cosine coefficient similarity measure ($\mathrm{KNN}_{\mathrm{cosine}}$) have been used for the comparison. SOM uses 2×2~5×5 maps with rectangular topology and 0.05 of learning rate. Quinlan's C4.5 has been adopted for DT. The parameters that produce the best results on training set have been chosen. The final results are averaged by 10 runs.

C. *Results*

1) Single Classifier with Single Feature: Totally, 175 (25 ×7) genes, including overlapping, are selected as features. However, only 30 genes are appeared in more than two feature selection methods as shown in Fig. 3. g_{1882}, g_{2288} and g_{6201} appear in three feature selection methods at the same time. This fact indicates that each feature selection method selects disjoint sets of genes (in feature spaces) and search for the optimal solution exclusively in the solution space.

```
2,    5,    6,    8,    12,   13,   14,   18,
22,   461,  1249, 1745, 1834, 1882, 2020, 2043,
2111, 2242, 2288, 2402, 2759, 3258, 3320, 3847,
4196, 4847, 5039, 6200, 6201, 6990
```

Fig. 3. IDs of genes chosen by more than two methods

Fig. 4 shows the examples of frequent misclassification made by MLP, SVM and KNN, which are single classifiers to be combined in ensemble. Samples 15, 23 and 34 are apt to be misclassified by $\mathrm{SVM}_{\mathrm{linear}}$ and $\mathrm{SVM}_{\mathrm{RBF}}$, but they are quite well classified by MLP and KNN. Samples 5 and 10 are missed by KNN with *SN*, but well classified by other classifiers. Even between SVMs, SVM with linear kernel always fails to recognize sample 27, but RBF kernel correctly recognizes.

Feature	MLP	$\mathrm{SVM}_{\mathrm{linear}}$	$\mathrm{SVM}_{\mathrm{RBF}}$	KNN
PR	34	23 27 29 30 31 32 33 34	23 25 26 28 29 30 31 32 34	21 22 31
SP	15 16 20 22 23 24 25 27 28 29	27 34	28 29 31 33	2 29 34
ED	22	24 25 27 28 29 30 31 33 34	17 20 22	23 29 32 34
CC	11 16 22 24 25 26 28	15 24 25 27 34	15 20 28 30	9 15 16 24 25 27 28 29
IG	16 22 26	23 26 27 28	21 22 24 25 26 29 30	5 15 18 21 22 23 27
MI	4 9 12 16 18 19 22 24 25 26 31	21 22 24 25 26 27 29 30	15 34	15 18 24 25 27 28 29 33 34
SN	28 29	28 29	15 34	4 5 10 32

Fig. 4. Frequently misclassified samples

The results of recognition rate on the test data with single feature and classifier are as shown in Table 2. The MLP seems to have the best recognition rate among the classifiers on the average. 97.1% of accuracy is the best throughout all the classifiers and features. However, the performance of classifiers seems to be somewhat dependent on the feature it uses. $\mathrm{SVM}_{\mathrm{linear}}$, for example, has 97.1% of accuracy with Pearson's correlation, but 58.8% with information gain and

TABLE II. RECOGNITION RATES BY FEATURE AND CLASSIFIER [%]

Feature / Classifier	Pearson	Spearman	Euclidean distance	Cosine coefficient	Information gain	Mutual information	S/N ratio	Average
MLP	97.1	70.6	97.1	79.4	72.9	62.1	94.1	81.9
KNN	88.2	73.5	82.4	76.5	70.6	58.8	94.1	77.7
SVM_{linear}	97.1	70.6	91.2	70.6	58.8	58.8	94.1	77.3
SVM_{RBF}	97.1	70.6	78.6	70.6	58.8	58.8	94.1	75.5
KNN_{cosine}	91.2	61.8	82.4	70.6	61.8	58.8	97.1	74.8
SOM	74.1	67.4	70.6	70.6	63.8	68.8	97.1	73.2
DT	97.1	61.8	82.4	73.5	47.1	55.9	91.2	72.7
Average	89.6	68.3	82.2	72.8	61.1	61.4	94.7	76.2

mutual information, which is the worst. KNNs with mutual information do not seem to do the classification at all.

Signal to noise ratio and Pearson's correlation are the best among the feature selection methods, obtaining 94.7% and 89.6% on the average, respectively.

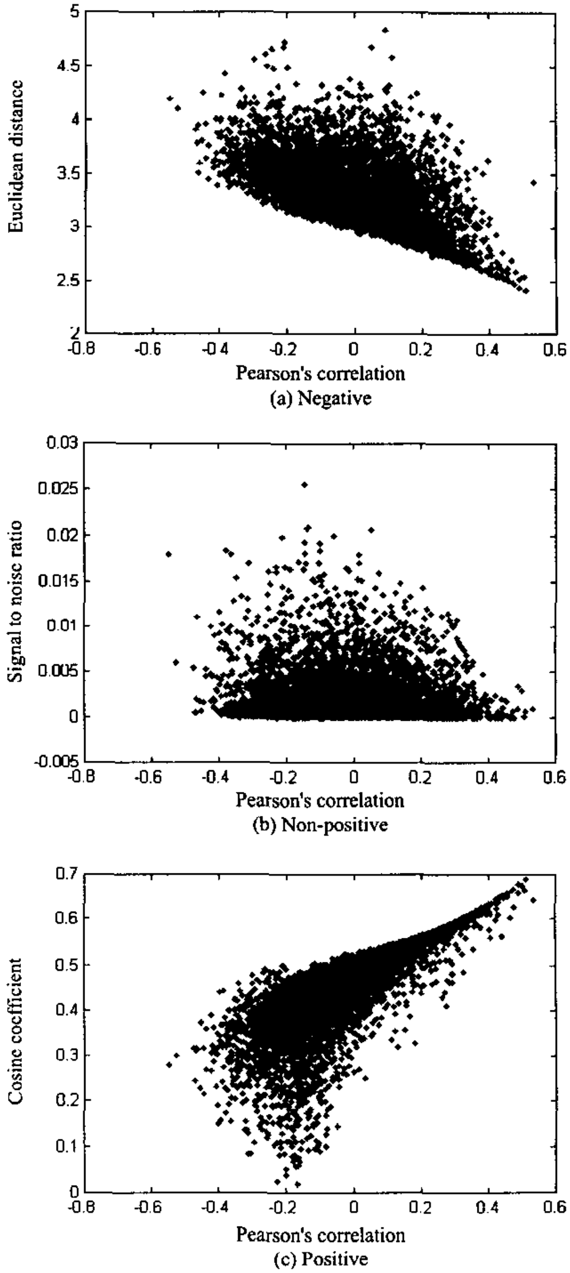
2) *Ensemble Classifiers:* Fig. 5 illustrates the dependency between two feature selection methods. These are three representative cases of all 21 possible feature pairs ($_7C_2$). Each axis is the feature selection method, and 7,129 genes' scores by the corresponding feature selection methods are marked as dark dots in the figure.

Case (a) is the correlation between Pearson's correlation and Euclidean distance. Dots are distributed in negative direction. Genes ranked high in Pearson's correlation get low scores from Euclidean distance, and vice versa. Therefore, the feature sets chosen by Pearson's correlation and Euclidean distance must be very disjoint, and the classifiers with these feature selection methods are trained in independent feature spaces each other.

Case (b) is the correlation between Pearson's correlation and signal to noise ratio method. Dots are distributed in a triangular form. We cannot explicitly see the direction of the correlation. However, when we see around the part of the right vertex of triangle, it seems to have a tendency that genes chosen by on feature selection method are devaluated by the other, just as in case (a), leading to non-positive (or week negative) correlation. Actually, most of the cases in the experiments show non-positive correlations, but (b) is one typical example of this category.

Case (c) shows a positive correlation, which is from between Pearson's correlation and cosine coefficient methods. Genes selected by one method with high score also appear in the list of top-ranking genes by the other method. In this case, there must be many common genes between two feature sets, so that the ensemble classifier will learn from highly correlated feature sets. Since two sets of classifiers are trained in mutually dependent feature spaces, it is hard to expect the performance improvement when the ensemble classifiers are combined.

MLP, KNN, SVM_{linear}, and SVM_{RBF} have been trained simultaneously from the same feature sets chosen by each feature selection method, and the outputs from this set of classifiers of two features (case (a), (b) and (c)) are combined by a neural network and voting method. For the comparison, we have also combined outputs of classifiers trained using all features.



(a) Negative

(b) Non-positive

(c) Positive

Fig. 5. Correlations of feature selection methods

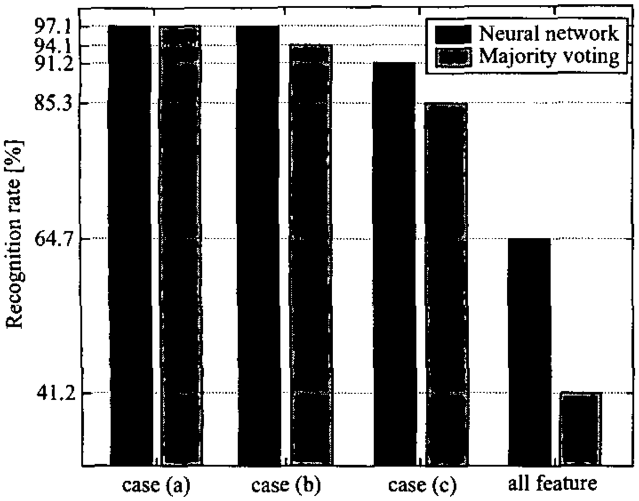


Fig. 6. Results of ensemble classifiers

Fig. 6 is the result of the ensemble classifiers. Case (a), (b) and (c) are investigated and we also have combined all the features, for the comparison with others. Case (a) produces best recognition rates, 97.1% in both neural network and voting method, which is also the best that we could get in the experiments with single feature/classifier. Case (b) also produces relatively high rates. Case (c) and 'all feature,' however, turn to be bad, which implies that combining independent features is efficient, producing much higher performance than when all features are considered.

This clearly shows that the suggested framework works and we may improve the classification performance by combining mutually exclusive sets of classifiers learned from two independent features, even when we use simple combination method of voting.

V. CONCLUDING REMARKS

In order to predict the cancer class of patients, we have illustrated a classification framework that combines sets of classifiers using the correlation information of seven feature selection methods. We have shown the usefulness of this framework with cancer gene expression data. Experimental results show that the feature sets that have negative or non-positive correlations yield high recognition result.

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] P. Tamayo, "Interpreting patterns of gene expression with self-organizing map: Methods and application to hematopoietic differentiation," *Proc. of the Natl. Acad. of Sci. USA,* vol. 96, pp. 2907-2912, 1999.

[3] D. Lashkari, J. Derisi, J. McCusker, A. Namath, C. Gentile, S. Hwang, P. Brown, and R. Davis, "Yeast microarrays for genome wide parallel genetic and gene expression analysis," *Proc. of the Natl. Acad. of Sci. USA*, vol. 94, pp. 13057-13062, 1997.

[4] J. Derisi, V. Iyer and P. Brosn, "Exploring the metabolic and genetic control of gene expression on a genomic scale," *Science*, vol. 278, pp. 680-686, 1997.

[5] M. Eisen, P. Spellman, P. Brown and D. Bostein, "Cluster analysis and display of genome-wide expression patterns," *Proc. of the Natl. Acad. of Sci. USA*, vol. 95, pp. 14863-14868. 1998.

[6] S. Haykin, *Neural Networks*, 2nd Eds., Chapter 7, pp. 351-391, Prentice Hall, 1999.

[7] J. Ryu and S.-B. Cho, "Towards optimal feature and classifier for gene expression classification of cancer," *Proc. of Asian Fuzzy Systems Society*, 2002. (To appear)

[8] D. N. Osherson, S. Weinstein and M. Stob, "Modular learning," *Computational Neuroscience*, E. L. Schwartz, Eds., pp. 369-377, Cambridge, MA: MIT Press, 1990.

[9] J. Pollack, "Cascaded back-propagation on dynamic connectionist networks," *Proc. of Ninth Annl. Conf. Cognitive Sci., Soc.*, pp. 391-404, 1987.

[10] R. Jacobs, "Initial experiments on constructing domains of expertise and hierarchies in connectionist systems," *Proc. of Connectionist Models Summer School*, pp. 144-153, San Mateo: CA, 1988.

[11] J. B. Hamshire and A. Waibel, "The Meta-Pi network: Building distributed knowledge representations for robust multisource pattern recognition," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 14, pp. 751-769, July 1992.

[12] W. P. Lincoln and J. Skrzypek, "Synergy of clustering multiple backpropagation networks," *Advances in Neural Information Processing Systems,* D. S. Touretzky, Eds., CA: Morgan Kauffmann, vol. 1, pp. 650-657, 1990.

[13] R. Battiti and A. M. Colla, "Democracy in neural nets: Voting schemes for classification," *Neural Networks*, vol. 7, pp. 691-707, July 1994.

[14] Y. Liu and X. Yao, "Ensemble learning via negative correlation," *Neural Networks*, vol. 12, no. 10, pp. 1399-1404, December 1999.

[15] Y. Liu and X. Yao, "Evolutionary ensembles with negative correlation learning," *IEEE Trans. on Evolutionary Comput.* vol. 4, no. 4, November 2000.

[16] R. P. Lippman, "An introduction to computing with neural nets," *IEEE ASSP Magazine*, pp. 4-22, April 1987.

[17] H. D. Beale, *Neural Network Design*, PWS Publish Company, vol. 11, pp. 1-47, 1996.

[18] V. N. Vapnik, *The Nature of Statistical Learning Theory*, New York: Springer, 1995.

[19] B. Moghaddam and M.-H. Yang, "Gender classification with support vector machines," *Proc. of 4th IEEE Intl. Conf. on Automatic Face and Gesture Recognition*, pp. 306-311, 2000.

[20] J. Ryu and S.-B. Cho, "Gender recognition of human behaviors using neural network ensembles," *Proc. of IEEE-INNS Intl. Joint Conf. on Neural Networks*, pp. 571-576, Washington DC, USA, July 2001.