Research Article

An Efficient Ensemble Learning Method for Gene Microarray Classification

Alireza Osareh and Bita Shadgar

Department of Computer Engineering, Islamic Azad University, Dezful Branch, Dezful 313, Iran

Correspondence should be addressed to Alireza Osareh; alireza.osareh@scu.ac.ir

The gene microarray analysis and classification have demonstrated an effective way for the effective diagnosis of diseases and cancers. However, it has been also revealed that the basic classification techniques have intrinsic drawbacks in achieving accurate gene classification and cancer diagnosis. On the other hand, classifier ensembles have received increasing attention in various applications. Here, we address the gene classification issue using RotBoost ensemble methodology. This method is a combination of Rotation Forest and AdaBoost techniques which in turn preserve both desirable features of an ensemble architecture, that is, accuracy and diversity. To select a concise subset of informative genes, 5 different feature selection algorithms are considered. To assess the efficiency of the RotBoost, other nonensemble/ensemble techniques including Decision Trees, Support Vector Machines, Rotation Forest, AdaBoost, and Bagging are also deployed. Experimental results have revealed that the combination of the fast correlation-based feature selection method with ICA-based RotBoost ensemble is highly effective for gene classification. In fact, the proposed method can create ensemble classifiers which outperform not only the classifiers produced by the conventional machine learning but also the classifiers generated by two widely used conventional ensemble learning methods, that is, Bagging and AdaBoost.

1. Introduction

Previous studies have shown that gene microarray data analysis is a powerful and revolutionary tool for biological and medical researches by allowing the simultaneous monitoring of the expression levels of tens of thousands of genes [1].

This is done by measuring the signal intensity of fluorescing molecules attached to DNA species that are bound to complementary strands of DNA localized to the surface of the microarray. Usually a ratio of intensities is calculated for each probe or gene, corresponding to two different labeled populations of reverse-transcribed mRNA.

Having captured the spot intensities, the obtained intensities undergo a normalization preprocessing stage to remove systematic errors within the data [2]. Early application of microarrays to the study of human disease conditions rapidly revealed their potential as a medical diagnostic tool [3, 4]. This is a class prediction problem to which supervised learning techniques are ideally suited.

Some studies have been reported on the application of microarray gene expression data analysis for molecular cancer classification [5, 6]. In fact, Microarray analysis has demonstrated that accurate cancer diagnosis can be achieved by performing microarray data classification, that is, by constructing classifiers to compare the gene expression profile of a tissue of unknown cancer status to a database-stored expression profile from tissues of known cancer status.

Usually, microarray classification process comprised of two successive steps, that is, feature selection and classification. So far, many machine learning algorithms have been introduced, and many of them have been employed for both steps, including the techniques of feature selection [7], classification techniques, for example, K-NN [8], support vector machines [9, 10], and neural networks [11]. Most of the existing research works attempt to choose an optimal subset of genes and then generalize an accurate classification model based on the selected genes.

The microarray data measures the expressions of tens of thousands of genes, producing a feature vector that is high in dimensionality and that contains much irrelevant information. This dimensionality degrades classification

performance. Moreover, datasets typically contain few samples for training (e.g., lung dataset [12] contains 12535 genes and only 181 samples), leading to the curse of dimensionality problem. It is essential, therefore, to find efficient methods for reducing the size of the feature set.

To avoid the curse of dimensionality problem, gene selection plays a crucial role in DNA microarray analysis. Another important reason to reduce dimensionality is to help biologists to identify the underlying mechanism that relates gene expression to diseases.

Indeed, the microarray data is associated with various uncertainties such as microarray data, gathering process which include fabrication, hybridization and image processing. These uncertainties always add various sources of noise [13]. Because of the impact of different uncertainties together with the lack of labeled training samples, the conventional machine learning techniques face complicated challenges to develop reliable classification models. Quite often selecting only a few genes can discriminate a majority of training instances correctly [14]. However, the generalization ability of such classifier model based on a few principal genes and a limited number of labeled training instances cannot be guaranteed.

It is therefore essential to develop general approaches and robust methods that are able to overcome the limitation of the small number of training instances and reduce the influence of uncertainties so as to produce reliable classification results. The motivation for this study is to utilize robust ensemble methods that are less sensitive to the selection of genes and are capable of removing the uncertainties of gene expression data.

Ensemble methodology is an efficient technique that has increasingly been adopted to combine multiple learning algorithms to improve overall prediction accuracy [15]. These ensemble techniques have the advantage to alleviate the small sample size problem by averaging and incorporating over multiple classification models to reduce the potential for overfitting the training data [16]. In this way the training data set may be used in a more efficient way, which is critical to many bioinformatics applications with small sample size.

Much research has shown the promise of ensemble learning for improving the accuracy in classifying data under uncertainties [15, 17]. However, a necessary and sufficient condition for an ensemble to outperform its individual members is that the base classifiers should be accurate and diverse [18]. An accurate classifier is one that has an error rate of better than randomly guessing classes for new unseen samples. On the other hand, two classifiers are said to be diverse if their decisions are different when classifying the same new instance, that is, if the individual classifiers do not always agree.

The most popular ensemble methods utilize a base classification algorithm to differently permutated training sets. Examples of these techniques include AdaBoost, Bagging, Random Subspace, Random Forest, and Rotation Forest [19]. AdaBoost has become a very popular choice for its simplicity and adaptability [20]. This algorithm builds an ensemble of classifiers by utilizing a specified base learning algorithm to successive obtained training sets that are formed by either resampling from the original training set or reweighting the original training set according to a set of weights maintained over the training set [20]. Thus, AdaBoost attempts to produce new "strong" classifiers that are able to better predict the hard instances for the previous ensemble "weak" members.

In Bagging, each base classifier is constructed on a bootstrap sample of the original training data, that is, a random sample of instances drawn with replacement and having the same size as the original training data. Ensemble classification is achieved by means of majority voting, where an unlabeled unseen data is assigned the class with the highest number of votes among the individual classifiers' predictions [21].

A successful variation upon Bagging is the Rotation Forest. Rotation Forest is an ensemble classification approach which is built with a set of decision trees. For each tree, the bootstrap samples extracted from the original training set are adopted to construct a new training set. Then the feature set of the new training set is randomly split into some subsets, which are transformed individually. Since a small rotation of axes may build a complete different tree, the diversity of the ensemble system can be guaranteed by the transformation [22].

Compared with the other proposed ensemble approaches, such as AdaBoost [23], Bagging [24], and Random Forest [25], Rotation Forest is more robust because it can always enhance the generalization ability of the individual classifiers and the diversity in the ensemble at the same time.

C. Zhang and J. Zhang [19] proposed a novel ensemble classifier generation method RotBoost through combining Rotation Forest and AdaBoost. In this new ensemble method, the base classifier in Rotation Forest algorithm is replaced with AdaBoost. The experimental results show that RotBoost performs better than either Rotation Forest or AdaBoost when using some non-microarray gene-related data sets from the UCI repository.

Here, we inspired from RotBoost technique and suggest a two-stage ensemble learning methodology by integrating fast correlation-based filter feature selection and independent component analysis- (ICA-) based RotBoost ensemble classification. Indeed, to verify the efficiency of the proposed method on gene-related data, 8 publically available gene microarray benchmark datasets are analyzed.

To this end, we accomplish a comparative study of RotBoost efficiency against several other ensemble and single classifier systems including AdaBoost, Bagging, Rotation Forest single tree, and support vector machines (SVMs). Moreover, to achieve the optimum arrangement and parameters, different variations of RotBoost such as PCA-based and ICA-based are compared, following an implementation of several feature selection approaches.

In terms of generalization accuracy, ICA-based RotBoost ensemble in conjunction with fast correlation-based filter demonstrated superior average performance over all considered ensemble classifiers and is therefore recommended as an efficient classification technique for the prediction of new gene microarray class labels.

The rest of this paper is organized as follows. In Section 2, the framework of RotBoost is described in detail. In Section 3,

the experimental results and corresponding discussions are presented. Section 4 concludes the paper.

2. Materials and Methods

2.1. The Description of RotBoost Ensemble Classification. As it was stated before, RotBoost is constructed by integrating the ideas of Rotation Forest and AdaBoost ensemble classifier generation techniques with the aim of achieving even lower prediction error than either of these individual techniques.

Rotation Forest is an ensemble method which trains L decision trees independently, using a different set of extracted features for each tree [25]. Let $x = [x_1, \ldots, x_n]^T$ be an example described by n features (attributes) and let X be an $N \times n$ matrix containing the training examples. Assuming that w is the set of class labels $\{w_1, \ldots, w_m\}$, from which Y takes values. Training a base classifier C_i involves using the training data to formalize a mapping of the input variable space onto the binary response variable, Y. If the feature set F is split randomly into K subsets with approximate size, there will be the ensemble of L classifiers denoted by $D = \{D_1, \ldots, D_L\}$. To construct the training set for an individual classifier C_i, we split F into K disjoint subsets randomly where each feature subset contains $M = n/K$ features [23]. Let F_{ij} be the jth subset of features for training classifier C_i and X_{ij} the dataset X for the features in F_{ij}. Now, for each subset, we select a nonempty subset of classes from X_{ij} randomly. Then a bootstrap subset of input instances is drawn to form a new training set, which is denoted by X'_{ij}. Subsequently, a transformation such as PCA is applied on X'_{ij} to generate the coefficients in a matrix Q_{ij}, denoted by the coefficients $a_{ij}^{(1)}, \ldots, a_{ij}^{(Mj)}$. Therefore, the size of each X'_{ij} is $M \times 1$.

Finally, we canconstruct a sparse rotation matrix R_i with the obtained coefficients in matrix Q_{ij}, as follows [22]:

$$R_i = \begin{bmatrix} a_{i1}^{(1)}, \ldots, a_{i1}^{(M_1)} & \{0\} \ldots & \{0\} \\ \{0\} & a_{i2}^{(1)}, \ldots, a_{i2}^{(M_2)} & \{0\} \\ \{0\} & \{0\} & a_{iK}^{(1)}, \ldots, a_{iK}^{(M_K)} \end{bmatrix}. \quad (1)$$

The columns of R_i are rearranged according to the original feature sequence, and the rearranged rotation matrix is indicated by R^a_i. The new obtained transformed training set for classifier C_i is shown by XR^a_i. During the test stage, given a test input instance x, let $d_{ij}(xR^a_i)$ be the probability produced by the classifier C_i to the hypothesis that x belongs to class w_j. Then the confidence for a class can be obtained as follows:

$$\varphi_j(x) = \frac{1}{L} \sum_{i=1}^{L} d_{ij}(xR^a_i), \quad j = 1, \ldots, m. \quad (2)$$

Now, an unknown input sample x is assigned to the class with the largest confidence. In RotBoost, the base classifiers C_i in Rotation Forest are replaced by AdaBoost classifiers. RotBoost offers a potential computational advantage over AdaBoost in that it has the ability to execute in parallel. In fact, each subensemble classifier formed by AdaBoost can be learned independently of the other ones. Pseudocode 1 illustrates this algorithm.

When employing the RotBoost classification algorithm, some parameters are required to be defined beforehand. The values of the parameters S and T that, respectively, specify the numbers of iterations done for Rotation Forest and AdaBoost should be subjectively fine-tuned, and the value of K (or M which represents the number attributes in each subspace) can be selected to be a moderate value according to the size of the feature set F. Indeed, the decision trees are utilized as the individual base classifiers of the final constructed RotBoost ensemble predictor.

2.2. Transformation Methods. As it was already mentioned, the purpose of rotation-based ensemble classifiers such as Rotation Forest and RotBoost is to increase the individual classifier performance and the diversity within the ensemble. Thus, a full feature set is obtained with all the transformed features for each considered tree in the ensemble. Because a small rotation of axes may construct a complete different tree, the diversity of the ensemble classifier can be guaranteed by the selected transformation.

There are largely two kinds of transformation methods, that is, PCA and ICA. PCA projects the data into a new space spanned by the principal components [26]. In contrast to PCA, ICA decomposes an input dataset into components so that each component is statistically as independent from the others as possible. It appears that ICA has a greater advantage over PCA in many aspects. First, it provides a better probabilistic model of the data, which can better identify where the data concentrate in n-dimensional space. Second, it can find a not necessarily orthogonal basis, which may reconstruct the data better than PCA in the presence of noise. Finally, it is sensitive to high-order statistics in the data, not just the covariance matrix [27]. Here, for the sake of comparison, we experiment with both PCA and ICA transformation methods and will report on their efficiency for our gene microarray classification task later on.

2.3. Gene Selection. Available training data sets for classification of cancer types generally have a fairly small sample size compared to the number of genes involved. This fact poses a challenging difficulty to some classification methodologies due to training data limitations. To be more specific, in a typical microarray dataset, there are thousands of gene features. Then if RotBoost ensemble classifier is applied to classify such dataset directly, a rotation matrix with thousands of dimensions is required for each tree, which greatly increases the computational complexity.

On the other hand, as only a small subset of genes is potentially relevant for distinguishing the sample classes, feature selection technique is crucial to reduce the number of features, removes irrelevant, noisy, and redundant data, and results in acceptable classification accuracy.

There are two broad categories for feature selection algorithms, filter model or wrapper [28]. The filter model relies on general characteristics of the training data to choose best features without involving any learning algorithm.

Input

    (i) L: a training set, $L = \{(x_i, y_i)\}_{i=1}^{N} = [X \;\; Y]$ where X is an $N \times n$ matrix containing the input values and Y is an N-dimensional column vector containing the class labels.

    (ii) K: number of attribute subsets (or M: number of input attributes contained in each subset).

    (iii) C: a base learning.

    (iv) S: number of iterations for Rotation Forest.

    (v) T: number of iterations for AdaBoost.

    (vi) x: a data point to be classified.

Training Phase

    For $s = 1, 2, \ldots, S$

        (1) use the steps similar to those in Rotation Forest to compute the Rotation matrix, say, R_s^a and let $L^a = [XR_t^a Y]$ be the training set for classifier C_s.

        (2) Initialize the weight distribution over L^a as $D_1(i) = 1/N$ $(i = 1, 2, \ldots, N)$.

        (3) For $t = 1, \ldots, T$

            (a) According to distribution D_t perform N extractions randomly from L^a with replacement to compose a new set L_t^a.

            (b) Apply w to L_t^a to train a classifier C_t^a and then compute the error of C_t^a as

$$\varepsilon_t = \Pr_{i \sim D_t}\left(C_t^a(x_i) \neq y_i\right) = \sum_{i=1}^{N}(C_t^a(x_i) \neq y_i))D_t(i)$$

            (c) If $\varepsilon_t > 0.5$ then set $D_t(i) = 1/N$ $(i = 1, 2, \ldots, N)$. and go to step (a); if $\varepsilon_t = 0$, then set $\varepsilon_t = 10^{-10}$ to continue the following iterations.

            (d) Choose $\alpha_t = 1/2 \ln\left((1 - \varepsilon_t)/\varepsilon_t\right)$

            (e) Update the distribution D_t over L^a as:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t}, & \text{if } C_t^a(x_i) = y_i, \\ e^{\alpha_t}, & \text{if } C_t^a(x_i) \neq y_i, \end{cases}$$

           where Z_t is a normalization factor being chosen so that D_{t+1} is a probability distribution over L^a.

           Endfor

        (4) Let $C_s(x) = \arg\max_{y \in \phi} \sum_{t=1}^{T} \alpha_t I(C_t^a(x) = y)$.

    Endfor

Output

    (i) The class label for x predicted by the final ensemble C^* as

$$C^*(x) = \arg\max_{y \in \phi} \sum_{s=1}^{S} I\left(C_s(x) = y\right),$$

    where $I(\cdot)$ is an indicator function.

Pseudocode 1: The RotBoost pseudocode.

The wrapper models, on the contrary, depends on feature addition or deletion to compose subset features and uses an evaluation function with a predetermined learning algorithm to estimate the subset features. Although these models tend to find features better suited to the learning algorithm resulting in superior learning performance, they also tend to be more computationally expensive than the filter model [29].

When the number of features becomes very large, the filter model is usually chosen due to its computational efficiency. Here, we utilize fast correlation-based filter (FCBF) as previous experiments [30] suggest that FCBF is an efficient and fast feature selection algorithm for classification of high dimensional data. FCBF model uses interdependence of features together with the dependence to the class and can achieve high degree of dimensionality reduction. This in turn improves classification accuracy with predominant selected features.

FCBF begins by selecting a subset of relevant features whose C-correlations are larger than a given threshold and then sorts the relevant features in descending order in terms of C-correlation. Using the sorted feature list, redundant features are eliminated one by one in a descending order. The

Table 1: Summary of benchmark gene microarray datasets.

Dataset	# Total genes (T)	# Instances (n)	# Classes (C)
Colon tumor	2000	62	2
Central nervous system	7129	60	2
Leukaemia	6817	72	2
Breast cancer	24481	97	2
Ovarian cancer	15154	253	2
MLL	12582	72	3
SRBCT	2308	83	4
Lung cancer	12533	181	5

remaining feature subset thus contains the predominant features with zero redundant features in terms of C-correlation [31].

2.4. Datasets. Here, we utilized 8 publicly available benchmark datasets [32]. A brief overview of these datasets is summarized in Table 1. Preprocessing is an important step

TABLE 2: Number of selected genes for each gene selection algorithm.

Dataset	Initial gene numbers	FCBF	ReliefF	CFS	mRMR	GSNR
Colon	2000	14	25	31	100	100
CNS	7129	28	28	40	356	356
Leukaemia	7129	51	104	50	356	356
Breast	24481	90	131	130	1224	1224
Lung	12553	100	432	299	628	628
Ovarian	15154	30	120	103	587	587
MLL	12582	97	295	327	629	629
SRBCT	2308	82	97	82	115	115

for handling gene expression data. This includes two steps: filling missing values and normalization. For both training and test dataset, missing values are filled using the average value of that gene. Normalization is performed so that every gene expression has mean equal to 0 and variance equal to 1. In summary, the 8 datasets had between 2 and 5 distinct diagnostic categories, 60–253 instances, and 2000–24481 genes.

3. Results and Discussion

In our experiments, we choose four representative feature selection algorithms, that is, ReliefF, correlation-based filter selection (CFS), minimum redundancy maximum relevance (mRMR), and general signal to noise ratio (GSNR) in comparison with FCBF.

ReliefF [33] is an extension of the original Relief algorithm [34] that adds the ability of dealing with multiclass problems, and it is more robust and capable of dealing with incomplete and noisy data. The Relief family methods are especially attractive because they may be applied in all situations, have low bias, include interaction among features, and may capture local dependencies that other methods miss [35].

The CFS method is based on test theory concepts and relies on a set of heuristics to assess the adequacy of subsets of features. These heuristics take into account both the usefulness of individual features to predict the class label as well as their correlation [34].

The mRMR criterion computes both the redundancy between features and the relevance of each feature. Redundancy is computed by the mutual information (MI) between pairs of features whereas relevance is measured by the MI between each feature and the class labels. The mRMR method has also been applied successfully to microarray data [36].

The GSNR is a measure of the ratio between intergroup and intragroup variations. Higher GSNR values indicate higher discrimination power for the gene. GSNR selects m genes in the descent order, and the best subset of genes is selected based on a predefined description.

In order to reduce the computational complexity of the problem at hand and select the most informative genes, we run all 5 feature selection algorithms against each dataset and obtain the number of selected features for each algorithm. Table 2 shows the number of genes which are selected by these feature selection algorithms for each individual microarray
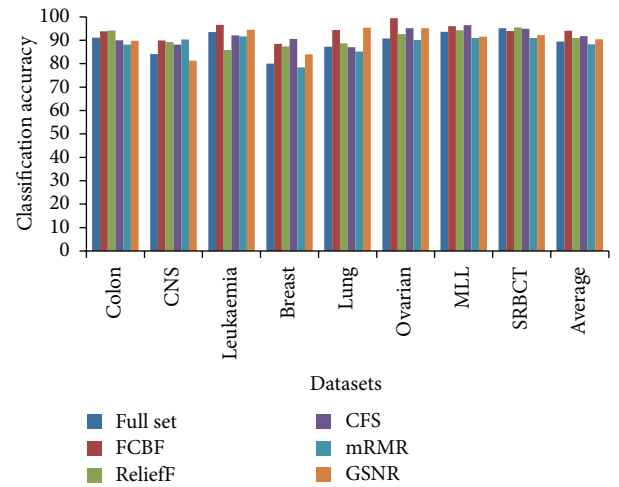

FIGURE 1: Classification accuracy of decision tree classifier on selected genes of 8 datasets based on different feature selection algorithms.

gene dataset. As it can be seen, the number of selected genes for each processed gene dataset is different and depends on the choice of a feature selection algorithm.

It should be noted that both mRMR and GSNR algorithms provide an ordered list of the initial genes (features) according to the genes importance and discrimination power. Here, for the sake of comparison, we experimentally retained the top 10% of the sorted genes by each of these two feature selection algorithms. This in turn leads to less computational cost in experiments.

From Table 2, it is obvious that FCBF achieves the highest level of dimensionality reduction by selecting the least number of discriminative genes. This is consistent with the theoretical analysis about FCBF's ability to identify and ignore redundant features.

To evaluate the gene classification accuracy of selected top genes by each feature selection algorithm, a single decision tree learning algorithm is utilized. The learning algorithm is applied to the original gene datasets as well as each newly obtained dataset containing only the selected genes, and in each case the final overall accuracy is measured.

Figure 1 summarizes the learning accuracy of decision tree classifier on different feature sets. Considering the

averaged accuracy over all data sets, we observe that, in general, FCBF improves the accuracy of decision tree classifier; it also outperforms the other four feature selection algorithms. Indeed, from individual accuracy values, we observe that for all the datasets except SRBCT, FCBF can highly increase the overall gene classification accuracy. On the other hand, CFS method achieves the second best classification accuracy, and both relief and GSNR accomplish more than 90% average accuracy.

However, as it was already mentioned, a single classifier such as decision tree is far from an accurate classifier when applied to the problem of gene microarray classification which usually confronts several challenges such as curse of dimensionality, small sample size datasets, and huge amount of noise and uncertainties.

To cope with these challenges and to develop a more robust and accurate learning method, ensemble learning methodology is utilized. The datasets are first preprocessed, and then to reduce the computational complexity and select the most informative genes, FCBF is applied to these datasets. Having chosen the best discriminative features, the ensemble classifiers including RotBoost, Rotation Forest, AdaBoost, and Bagging are developed and learned using these features. In all ensemble experiments a classification tree [16] was exploited as the base learner because it is sensitive to the changes in its training data.

In order to provide a fair comparison, for all utilized ensemble techniques 100 trees are trained to constitute the corresponding ensemble classifiers. With respect to RotBoost, the number of iterations for Rotation Forest and AdaBoost both fine-tuned to be $S = T = 10$ (to properly balance the trade-off between these two algorithms).

In theory the parameter M should be selected to be a moderate value according to the size of the feature set F. Therefore, it seems that the performances of RotBoost would be changed with different number of features contained in each feature subset (M). But when comparing the results obtained by setting M ranged between 1 and 20, it is found that the overall gene classification results under different conditions vary slightly and none of the values take obvious advantage. So there was no consistent relationship between the classification accuracy and M, which was also pointed out in [37]. Here, the option $M = 3$ was the optimum choice to establish a proper balance between the overall classification accuracy and the diversity of the based learners for most examined gene datasets.

To find the most promising transformation method and preserve both diversity and accuracy of the base classifiers, we conduct experiments with two well-known transforms, that is, PCA and ICA. When the initial genes are transformed by either PCA or ICA, all of the principal components or independent components are kept to preserve the discriminatory information. Following transformation, the axes are rotated optimally. Despite the conventional approach of choosing some directions for good discriminate capability, the rotation mainly contributes to the generation of diversity among the classifiers without weakening the individual classifiers. Thus, an acceptable trade-off between diversity and accuracy can be maintained simultaneously.

TABLE 3: Classification results obtained by RotBoost ensemble learning against typical 8 gene datasets in terms of PCA/ICA transformation methods.

Dataset	ICA_based RootBoost	PCA_based RootBoost
Colon	96.10 ± 0.59	$95.48 \pm 0.61^{\bullet}$
CNS	95.00 ± 0.28	$94.80 \pm 0.59^{\bullet}$
Leukaemia	98.77 ± 0.03	98.75 ± 0.31
Breast	97.88 ± 0.45	$94.39 \pm 0.49^{\bullet}$
Lung	99.54 ± 0.11	$98.11 \pm 0.17^{\bullet}$
Ovarian	99.40 ± 0.26	99.82 ± 0.08^{o}
MLL	99.31 ± 0.55	$98.86 \pm 0.23^{\bullet}$
SRBCT	99.59 ± 0.16	99.50 ± 0.31
Win tie loss		5/2/1

$^{\bullet}$Specifies that RotBoost is significantly better, and ^{o}points out that RotBoost is notably worse at the significance level $\alpha = 0.05$.

In many earlier works, researchers typically split the original dataset into two parts, that is, a training set and a test set in a random fashion. Gene selection is then performed on the training set, and the goodness of selected genes is assessed from the unseen test set [31].

However, due to the small number of instances in gene microarray datasets, such an approach can lead to unreliable results. Instead, Ambroise and McLachlan [38] suggested splitting the data using 10-fold cross-validation or 0.632 + bootstrap. A comparative study of several different error estimation techniques on microarray classification shows that 0.632 + bootstrap can be more appropriate than other estimators including resubstitution estimator, cross-validation, and leave-one-out estimation [39].

Therefore, in this work, we deployed a balanced 0.632 + bootstrap technique to evaluate the performance of the gene selection algorithm considered in this study. The 0.632 + bootstrap requires sampling a training set with a replacement manner from the original dataset. The test set is then made by those samples excluded from the training dataset. Finally, the 0.632 + bootstrap is repeated n times, and the final bootstrap error is estimated as follows:

$$E = \frac{1}{n} \sum_{i=1}^{n} \left(0.368\alpha_i + 0.632\beta_i \right), \tag{3}$$

where α_i and β_i are the training error and test error on the ith resampling stage. Following the work in [14], here, the bootstrap samples are experimentally formed with $n = 15$ replicates. It is worth to note that the feature selection is then carried out using only the training samples. Finally, the test error (classification accuracy) is estimated on the unseen test samples using (3).

Table 3 presents the RotBoost mean classification accuracy against the considered 8 gene datasets when transformation matrix is chosen to be either PCA or ICA where the values following "±" denote the related standard deviations. In order to explore whether RotBoost is significantly better or worse than other ensemble/nonensemble methods statistically, a one-tailed paired t-test is considered with significance level $\alpha = 0.05$ and the results for which a

TABLE 4: Mean classification accuracy of each classification method against 8 different gene datasets.

Dataset	ICA-based RotBoost	Single Tree	Rotation Forest	AdaBoost	Bagging	SVMs
Colon	96.10 ± 0.59	93.80 ± 0.82•	95.21 ± 0.43•	94.97 ± 0.63•	94.92 ± 0.50•	96.13 ± 0.12
CNS	95.00 ± 0.28	89.92 ± 0.61•	92.37 ± 0.83•	95.09 ± 0.64	93.50 ± 0.79•	93.34 ± 0.10•
Leukemia	98.77 ± 0.03	96.60 ± 00.46•	97.97 ± 0.38•	98.22 ± 0.55•	97.47 ± 0.51•	95.64 ± 0.49•
Breast	97.88 ± 0.45	88.50 ± 0.72•	98.60 ± 0.63º	98.89 ± 0.47º	92.74 ± 0.45•	96.84 ± 0.02•
Lung	99.54 ± 0.11	94.36 ± 0.42•	97.56 ± 0.23•	96.30 ± 0.39•	97.08 ± 0.37•	95.56 ± 0.55•
Ovarian	99.40 ± 0.26	99.37 ± 0.12	99.77 ± 0.07º	99.57 ± 0.11	99.76 ± 0.08º	98.66 ± 0.35•
MLL	99.31 ± 0.55	96.03 ± 0.59•	97.61 ± 0.31•	97.63 ± 0.45•	97.11 ± 0.55•	96.80 ± 0.31•
SRBCT	99.59 ± 0.16	93.96 ± 0.59•	97.44 ± 0.41•	98.16 ± 0.39•	96.46 ± 0.58•	97.23 ± 0.44•
Win Tie Loss		7/1/0	6/0/2	5/2/1	7/0/1	7/1/0

•Specifies that RotBoost is significantly better, and ºpoints out that RotBoost is notably worse at the significance level $\alpha = 0.05$.

significant difference with RotBoost was found are marked with a bullet or an open circle next to them. A bullet next to a result indicates that RotBoost is significantly better than the corresponding method. An open circle next to a result denotes that RotBoost performs significantly worse than the corresponding method. In the triplet labeled "Win-Tie-Loss" in the last row of Table 3, the first value denotes the number of gene datasets on which RotBoost operates considerably better than the corresponding algorithm; the second value stands for the number of datasets on which the difference between the performance of RotBoost and that of the corresponding algorithm is not significant; the third one indicates the number of datasets on which RotBoost performs significantly worse than the compared algorithm.

As it can be noted from Table 3, in 5 cases out of 8, ICA-based RotBoost learners could outperform their PCA-based counterparts in terms of higher classifications accuracies and lower standard deviations. On the other hand, in 2 cases, that is, Leukaemia and SRBCT datasets, the results obtained by both methods are not significantly different, and in 1 case (Ovarian dataset) the PCA-based RotBoost classifier could surpass the ICA-based learner. It is also necessary to point out that all these experiments have been accomplished on the best discriminative genes already selected by FCBF algorithm.

It is well known that no algorithm can hold a general advantage in terms of generalization capability over another one across all possible classification tasks. However, the relative advantage of an algorithm is possible across a set of real-world tasks. Considering that the 8 gene microarray datasets include different characteristics in terms of number of samples, genes, classes and the type of the cancer to which these data is related to; overall, ICA-based RotBoost classifier seems to be more effective than PCA-based RotBoost.

Following our decision on the choice of the transformation method, we accomplish a comparative study of ICA-based RotBoost accuracy performance against other ensemble and an independent single classifier system including AdaBoost, Bagging, Rotation Forest, single tree, and SVMs.

Specifying a SVMs classifier requires two parameters, that is, the kernel function and the regularization parameter C. In this study, the SVMs classifiers are evaluated based on Gaussian radial bases functions (RBF). In order to obtain the optimal value for the SVMs regularization parameter C and the parameter of kernel functions (σ), we experiment with different SVMs classifiers using a 5-fold cross-validation technique. The performance of the optimum selected SVMs is measured in terms of classification accuracy (Table 4).

Table 4 summarizes the mean classification accuracy of each classification method on the considered datasets. As can be seen from this table, RotBoost methodology performs significantly better than Single Tree, SVMs and Bagging, algorithms. When compared with Rotation Forest, the statistically significant difference is favorable in 6 datasets, although the Rotation Forest could surpass the RotBoost when working on Breast and Ovarian datasets. Indeed, RotBoost is seen to outperform AdaBoost in most cases even though the advantage of RotBoost is not significant in 1 dataset and tie is occurred on the remaining 2 datasets.

An important issue in ensemble learning methodology is the ability to establish a proper balance between the diversity and the accuracy of the constituted base learners. That is, the base learners utilized in a robust ensemble classifier should be of high classification accuracy and avoid making coincident misclassification errors which in turn necessitate the diverse learners. Thus, a sample misclassified by a base learner will be corrected by others, so the fused outputs are more accurate than that of the best individual classifier [37].

On the other hand, the diversity usually conflicts with the accuracy of base learners; that is, the more accurate the base learners are, the lower the diversity among them is. In practice, it appeared to be difficult to define a single measure of diversity and even more difficult to relate that measure to the ensemble performance in a neat and expressive dependency. Here, to investigate the ability of the proposed ICA-based RotBoost ensemble to build accurate and diverse base learners efficiently, the pairwise diversity measure is utilized [40].

This diversity measure evaluates the level of agreement between a pair of base learners while correcting for chance, which is named as Kappa statistic. For c class labels, k is defined on the $c \times c$ coincidence matrix $\mathbf{M}$ of the two classifiers. The entry $m_{k,s}$ of $\mathbf{M}$ is the proportion of the data
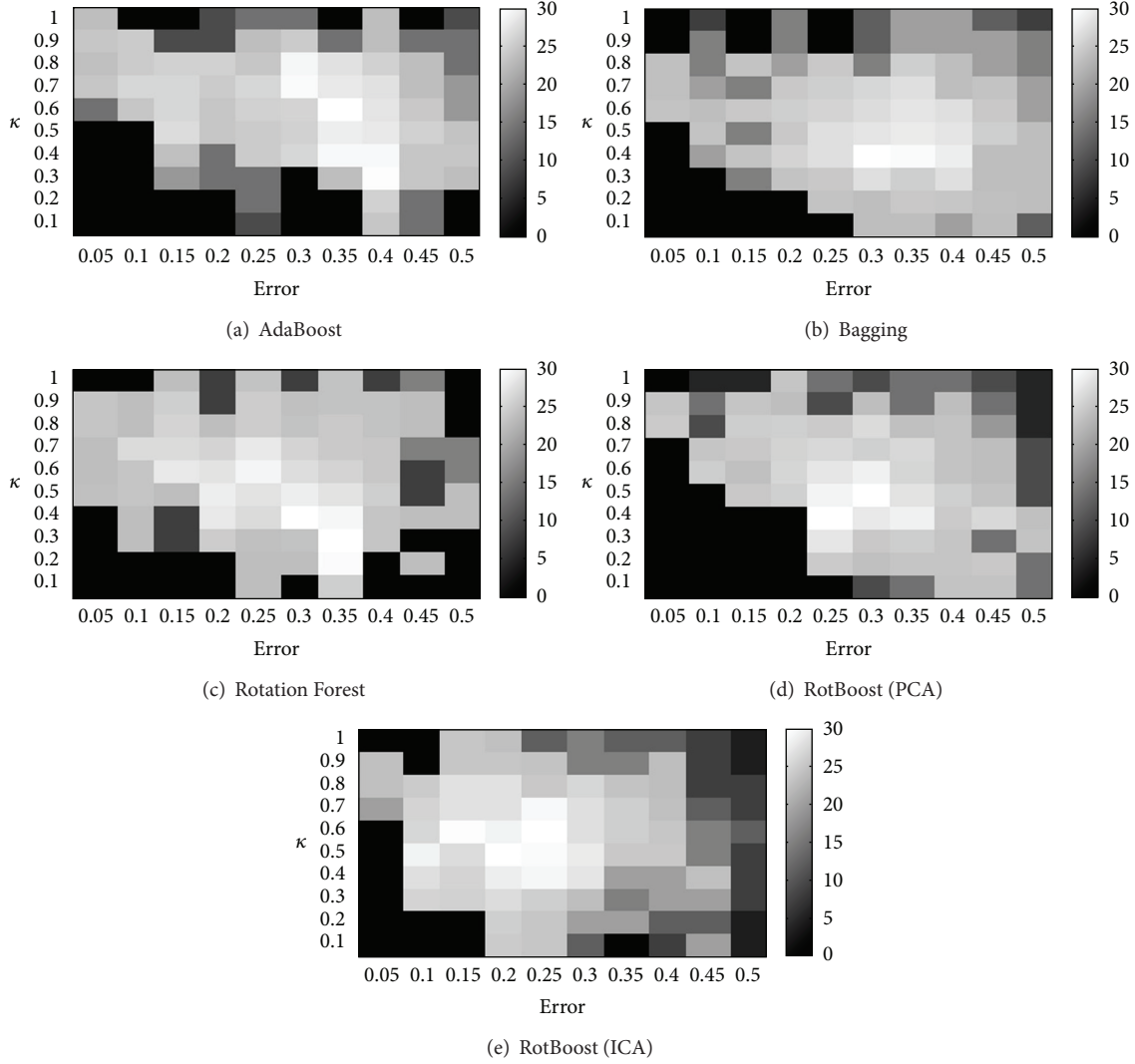
(a) AdaBoost

(b) Bagging

(c) Rotation Forest

(d) RotBoost (PCA)

(e) RotBoost (ICA)

FIGURE 2: Kappa error diagrams for the Lung dataset using different ensemble algorithms.

set, which D_i labels as w_k and D_j labels as w_s. Now, the agreement between D_i and D_j is given by

$$k_{i,j} = \frac{\sum_k m_{k,k} - ABC}{1 - ABC},$$ (4)

where $\sum_k m_{k,k}$ is the observed agreement between the classifiers, and "ABC" which is named agreement-by-chance is defined as follows [41]:

$$ABC = \sum_k \left(\sum_s m_{k,s}\right)\left(\sum_s m_{s,k}\right).$$ (5)

If the classifiers decisions are the same, $k = 1$ and $k = 0$ represent the case when the classifiers are entirely independent, and the agreement of the two classifiers equals that expected by chance [41]. In theory, there are $(P-1) \times P/2$ pairs of classifiers D_a and D_b for an ensemble with P base learners. For a typical kappa-error diagram, x-axis is the k for the pair, and y-axis is the averaged individual error, which is calculated by $E_{a,b} = (E_a + E_b)/2$, where E_a and E_b refer to

the error rates of pairs of classifiers D_a and D_b, respectively. It is worth to note that a small value of k denotes superior diversity among the learners, and in turn a small value of $E_{a,b}$ presents less error rate between the classifiers.

Table 5 summarizes the kappa-error values for typical Lung cancer dataset with FCBF gene selection method in terms of the centroids of different ensembles. From this table it is clear that the ICA-based RotBoost provides the highest pairwise accuracy, and the second best accuracy is achieved by PCA-based RotBoost. On the contrary, AdaBoost presents the best diversity at the cost of relatively poor classification accuracy. Bagging also presents the second best diversity but low classification accuracy.

When outlining the kappa-error diagrams, the diagrams of different ensembles are greatly overlapping, and the distances between the centroids are small. Therefore, a three dimensional histogram with a density underneath plot is usually utilized to clearly demonstrate the distribution of the results achieved by each ensemble method. Figure 2 illustrates these histograms against the typical Lung cancer dataset.

Table 5: Kappa error diagram for Lung dataset (the centroids of ensembles).

Ensemble method	Kappa	Error
AdaBoost	0.22	0.30
Bagging	0.24	0.25
Rotation Forest	0.29	0.23
RotBoost (PCA)	0.58	0.09
RotBoost (ICA)	0.59	0.07

We perform these experiments on all 8 gene datasets, and overall the ICA and PCA RotBoost methods perform best in terms of accuracy with average classification accuracy about 98.0% and 96.3%, respectively. Furthermore, these ICA and PCA RotBoost ensemble methods achieve average kappa values of 0.48 and 0.55, respectively. This is coincident with the observation previously given that the diversity usually conflicts with the accuracy of the base learners. However our experiments indicate that the ICA-based RotBoost ensemble could establish a proper balance between the diversity and the accuracy of the constituted base learners.

4. Conclusions

In this work, we addressed RotBoost ensemble classification method to cope with gene microarray classification problems. This ensemble classifier method is a combination of Rotation Forest and AdaBoost techniques which in turn preserve both desirable features of an ensemble architecture, that is, accuracy and diversity. To overcome the limitation of the small number of gene instances and curse of dimensionality and in order to select a small subset of most informative genes, 5 representative feature selection algorithms, that is, Relief F, CFS, mRMR, GSNR, and FCBF were applied. Then, RotBoost was employed on the selected genes. Here, we experimented with 2 different transformation matrixes, that is, PCA and ICA towards RotBoost implementation.

To assess the efficiency of RotBoost algorithm different ensemble/nonensemble techniques including Rotation Forest, AdaBoost, Bagging single tree, and SVMs were also deployed. The experimental results revealed that the combination of the FCBF feature selection and ICA-based RotBoost ensemble with several base learners is a robust method for microarray classification. The proposed method achieved the highest averaged generalization ability compared to its counterparts and denoted an acceptable level of diversity among the learners for majority of the analyzed benchmark datasets.

References

[1] P. O. Brown and D. Botstein, "Exploring the new world of the genome with DNA microarrays," *Nature Genetics*, vol. 21, no. 1, pp. 33–37, 1999.

[2] J. Quackenbush, "Microarray data normalization and transformation," *Nature Genetics*, vol. 32, no. 5, pp. 496–501, 2002.

[3] J. Khan, J. S. Wei, M. Ringnér et al., "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nature Medicine*, vol. 7, no. 6, pp. 673–679, 2001.

[4] H. Bhaskar, D. C. Hoyle, and S. Singh, "Machine learning in bioinformatics: a brief survey and recommendations for practitioners," *Computers in Biology and Medicine*, vol. 36, no. 10, pp. 1104–1125, 2006.

[5] U. Alon, N. Barka, D. A. Notterman et al., "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 96, no. 12, pp. 6745–6750, 1999.

[6] L. J. Van't Veer, H. Dai, M. J. Van de Vijver et al., "Gene expression profiling predicts clinical outcome of breast cancer," *Nature*, vol. 415, no. 6871, pp. 530–536, 2002.

[7] R. Blanco, P. Larrañaga, I. Inza, and B. Sierra, "Gene selection for cancer classification using wrapper approaches," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 18, no. 8, pp. 1373–1390, 2004.

[8] S. Cho and H. Won, "Machine learning in DNA microarray analysis for cancer classification," in *Proceedings of the 1st Asia-Pacific Bioinformatics Conference on Bioinformatics*, pp. 189–198, 2003.

[9] M. P. S. Brown, W. N. Grundy, D. Lin et al., "Knowledge-based analysis of microarray gene expression data by using support vector machines," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 97, no. 1, pp. 262–267, 2000.

[10] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, no. 10, pp. 906–914, 2000.

[11] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian networks to analyze expression data," in *Proceedings of the 4th Annual International Conference on Computational Molecular Biology (RECOMB '00)*, pp. 127–135, April 2000.

[12] G. J. Gordon, R. V. Jensen, L.-L. Hsiao 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.

[13] X. Wang, M. J. Hessner, Y. Wu, N. Pati, and S. Ghosh, "Quantitatative quality control in microarray experiments and the application in data filtering, normalization and false positive rate prediction," *Bioinformatics*, vol. 19, no. 11, pp. 1341–1347, 2003.

[14] Y. Peng, "A novel ensemble machine learning for robust microarray data classification," *Computers in Biology and Medicine*, vol. 36, no. 6, pp. 553–573, 2006.

[15] T. Dietterich, "Ensemble methods in machine learning," in *Proceedings of the Multiple Classifier System Conference*, pp. 1–15, 2000.

[16] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization," *Machine Learning*, vol. 40, no. 2, pp. 139–157, 2000.

[17] P. Yang, Y. H. Yang, B. B. Zhou, and A. Y. Zomaya, "A review of ensemble methods in bioinformatics," *Current Bioinformatics*, vol. 5, no. 4, pp. 296–308, 2010.

[18] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.

[19] C.-X. Zhang and J.-S. Zhang, "RotBoost: a technique for combining rotation forest and adaboost," *Pattern Recognition Letters*, vol. 29, no. 10, pp. 1524–1536, 2008.

[20] C.-X. Zhang and J.-S. Zhang, "A variant of rotation forest for constructing ensemble classifiers," *Pattern Analysis and Applications*, vol. 13, no. 1, pp. 59–77, 2010.

[21] K. W. De Bock, K. Coussement, and D. Van den Poel, "Ensemble classification based on generalized additive models," *Computational Statistics and Data Analysis*, vol. 54, no. 6, pp. 1535–1546, 2010.

[22] K.-H. Liu and D.-S. Huang, "Cancer classification using Rotation Forest," *Computers in Biology and Medicine*, vol. 38, no. 5, pp. 601–610, 2008.

[23] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

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

[25] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[26] S.-I. Lee and S. Batzoglou, "Application of independent component analysis to microarrays," *Genome Biology*, vol. 4, no. 11, article r76, 2003.

[27] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski, "Face recognition by independent component analysis," *IEEE Transactions on Neural Networks*, vol. 13, no. 6, pp. 1450–1464, 2002.

[28] P. Mitra, C. A. Murthy, and S. K. Pal, "Unsupervised feature selection using feature similarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 301–312, 2002.

[29] C.-H. Yang, L.-Y. Chuang, and C.-H. Yang, "IG-GA: a hybrid filter/wrapper method for feature selection of microarray data," *Journal of Medical and Biological Engineering*, vol. 30, no. 1, pp. 23–28, 2010.

[30] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," *Journal of Machine Learning Research*, vol. 5, no. 12, pp. 1205–1224, 2004.

[31] Z. Zhu, Y.-S. Ong, and M. Dash, "Markov blanket-embedded genetic algorithm for gene selection," *Pattern Recognition*, vol. 40, no. 11, pp. 3236–3248, 2007.

[32] T. Li, C. Zhang, and M. Ogihara, "A comparative study of feature selection and multiclass classfication methods for tissue classification based on gene expression," *Bioinformatics*, vol. 20, no. 15, pp. 2429–2437, 2004.

[33] I. Kononenko, "Estimating attributes: analysis and extensions of RELIEF," in *Proceedings of the European Conference on Machine Learning*, pp. 171–182, 1994.

[34] K. Kira and L. Rendell, "A practical approach to feature selection," in *Proceedings of the 9th International Workshop on Machine Learning*, pp. 249–256, 1992.

[35] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "An ensemble of filters and classifiers for microarray data classification," *Pattern Recognition*, vol. 45, no. 1, pp. 531–539, 2012.

[36] C. Ding and H. Peng, "Minimum redundancy feature selection from microarray gene expression data," in *Proceedings of the 2nd IEEE Computational Systems Bioinformatics*, pp. 523–528, 2003.

[37] L. Kuncheva and J. Rodriguez, "An experimental study on rotation forest ensembles," in *Multiple Classifier Systems*, vol. 4472 of *Lecture Notes on Computer Science*, pp. 459–468, 2007.

[38] C. Ambroise and G. J. McLachlan, "Selection bias in gene extraction on the basis of microarray gene-expression data," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 10, pp. 6562–6566, 2002.

[39] U. M. Braga-Neto and E. R. Dougherty, "Is cross-validation valid for small-sample microarray classification?" *Bioinformatics*, vol. 20, no. 3, pp. 374–380, 2004.

[40] D. Margineantu and T. Detterich, "Pruning adaptive boosting," in *Proceedings of the 14th International Conference of Machine Learning*, pp. 378–387, 1997.

[41] L. Kuncheva, *Combining Pattern Classifiers*, John Wiley & Sons, 2004.