AMSE JOURNALS-AMSE IIETA publication-2017-Series: Advances B; Vol. 60; N°1; pp 1-24

Submitted Jan. 2017; Revised March 15, 2017, Accepted April 15, 2017

https://doi.org/10.18280/ama_b.600101

An Improved Rotation Forest Algorithm Based on Heterogeneous Classifiers Ensemble for Classifying Gene Expression Profile

*Tao Chen

* School of Mathematics and Computer Science, Shaanxi University of Technology, Hanzhong 723000, China (ct79hz@126.com)

Abstract

Many machine learning methods can't obtain higher classification performance because of the characteristics of high dimension and small samplest of gene expression profile. This paper proposes an improved rotation forest algorithm based on heterogeneous classifiers ensemble to classify gene expression profile.Firstly, all the original genes are ranked by using relieff algorithm, and then some top-ranked genes are selected to build a new training subset from original training set. Secondly, because decision tree classifier in rotation forest algorithm has the disadvantages of local optimum and overfitting,an improved rotation forest algorithm based on heterogeneous classifiers is proposed to overcome above problems.Here,heterogeneous classifiers based on support vector machine, decision tree and extreme learning machine, replace decision tree in rotation forest algorithm and are used to train base classifiers, and then the heterogeneous base classifiers will have the higher diversity each other to improve ensemble performance furtherly.Experimental results on nine benchmark gene expression profile datasets show our proposed algorithm is better than traditional rotation forest, bagging and boosting. It improves not only classification accuracy, but also has high stability and time efficiency.

Key words

Gene expression profile, rotation forest, relief*f* algorithm, heterogeneous classifiers.

1. Introduction

DNA microarray technology is a technological breakthrough in the field of molecular biology in the 21st century, and it is possible to detect the expression levels of thousands of genes in a single experiment. It will help to classify diseases according to expression levels of genes in normal and tumor cells from molecular biology aspect. Therefore, the classification of gene expression profile has gained more and more attentions in recently years [1-4].

In the past years, Decision tree (DT)[5,6], Artificial neural network (ANN)[7], Bayesian networks[8],K-nearest neighbor (KNN)[9,10] and Support vector machine (SVM)[11-14] were widely used in gene expression profile classification. However, these methods always cannot obtain better classification performance because of small samples and high dimension of gene expression profile. Especially, since it is not known in advance which classifier is the best for a particular classification problem, and it is impossible that all the methods are implemented and compared, how to choose the appropriate classifier is very difficult for a particular problem. Furthermore, the researches show the best single classifier who classify all the gene expression profiles datasets is not exist.

In 1990, Ensemble learning is proposed to solve these problems and gained better performance than single classifier. Multiple base classifiers are trained according to certain strategies, and then outputs of all the base classifiers are combined to classify new samples. Because the errors of one classifier are averaged out by the correct classification of another classifier in ensemble. Therefore, ensemble learning can reduce the risk of selecting a poor performance classifier to improve classification performance, and gains more and more attentions in the fields of data mining [15]. Krogh indicate precision and diversity of base classifiers usually affect ensemble performance in 1995.Especially, increasing the diversity of base classifiers can improve ensemble performance on the premise of guarantying precision of every base classifiers. Bagging [16], Boosting [17], Random Subspace [18] and Random Forest [19] are effective ensemble methods and usually get higher classification performance in recently years.

Rotation Forest, proposed by Rodriguez in 2006, is a new ensemble algorithm. Its main idea is different feature spaces are generated by using different features, and original training set is mapped to different new feature spaces to generate many different training subsets with high diversity according to the above feature spaces, and then PCA is used to improve precision of training subsets [20-25]. Many researches indicate classification performance of rotation forest is significantly higher than that of traditional ensemble methods (bagging, boosting, random forest, etc) because of improving precision and diversity of base classifiers. However, decision tree is employed to train base classifier in rotation forest algorithm, and decision tree will lead to the overfitting problem due to the complexity of classifier and local optimum, and then it usually affect ensemble performance of rotation forest.

Extreme learning machine (ELM), proposed by Guang-Bin Huang in 2004, is a new neural network learning algorithm. The unique feature of ELM is the input weights and thresholds in the hidden layer are randomly assigned and never adjusted. In addition, ELM has faster learning speed and higher generalization performance because it uses single hidden layer feedforward neural network (SLFN) to reduce the learning time of the algorithm, and is widely used in regression and classification problems [26-29].

This paper proposes an improved rotation forest based on heterogeneous classifiers to classify gene expression profile. Firstly, In order to decrease the dimension of gene expression profile and eliminate irrelevant and redundant genes, all the genes are ranked by using relieff algorithm [30] and the top-ranked genes are selected to build new training subset. Secondly, due to decision tree has the disadvantages of overfitting problem and local optimum, decision tree are replaced the heterogeneous classifiers of extreme learning machine, support vector machine and decision tree in improved rotation forest algorithm. It not only overcomes the overfitting problem and local optimum, but also increases the diversity among base classifiers. Therefore, the classification performance of improved rotation forest is further improved.

The remainder of this paper is organized as follows: Materials and methods, including to relief*f*, classification algorithm (Decision tree, Support vector machine,Extreme learning machine) and improved rotation forest, are given in section 2.Section 3 gives basic ideas and steps of our proposed algorithm. Section 4 makes experiments on nine benchmark gene expression profiles and gives the experimental results and analysis. The conclusion is made in the end.

2. Materials and methods

2.1 Relief*f* algorithm

Relief is an algorithm based on the measuring of attribute importance proposed by Kira in 1992, and is an effective feature filter algorithm and obtains higher performance in data mining and pattern recognize[30].

The core idea of Relief algorithm is to identify the importance of each attribute by its ability to identify the class of samples in the vicinity of each attribute. Firstly, the importance of every attribute is calculated according relief criterion, which the larger value denotes the attribute has better recognition ability. Secondly, the attributes which corresponding value exceed the threshold are selected.

The purpose of the relief algorithm for a given sample is to find two nearest neighbor samples of the sample: a sample from the same class of the given sample (called *Nearest Hit*), and another sample from the different class of the given sample (called *Nearest Miss*).

In fact, the measure of the importance of attribute A in the relief method is an approximation of the difference between the following two conditional probabilities:

$W(A) = P$ (different values of A | the neighbor samples of different classes -P(different values of A | neighbor samples of the same class).

Relief algorithm can deal with discrete and continuous data, but can only be used for binary classification. Relief*f* algorithm is proposed on the basis of relief by Kononenko in 1994, which selects K samples to measure attributes importance according following formula.

$$W(g) = W(g) - (\sum_{j=1}^{k} diff(g, x_i, H_j)) / kn + \sum_{C \neq class(x_i)} (\frac{P(C)}{1 - P(class(x_i))} (\sum_{j=1}^{k} diff(g, x_i, M_j) / kn)) \tag{1}$$

where, $diff(g,x,y) = |value(g,x) - value(g,y)| / max(g) - min(g)$, $value(g,x)$ denotes the value of sample x in attribute g; $P(C) = num(C)/n$, $num(C)$ denotes the number of samples in the Cth class; *max(g)*, *min(g)* denotes the maximum and minimum value of all the samples in feature g, respectively.

Relief and Relief*f* are widely used in the fields of attribute reduction and feature selection because of their simple principle, easy understanding, high efficiency and good recognition performance.

Method 1 gives basic steps of Relief*f* algorithm.

Method1. Feature selection based on Relief*f* algorithm

```
Input: training set $X=(x_1,x_2,\cdots,x_n)$ ,features set $G=(g_1,g_2,\cdots,g_m)$
Output: weight vector of the features $W=(w_1,w_2,\cdots,w_m)$
Step 1: Initialize weight vector of the features: $W=(0,0,\cdots,0)$;
Step 2: For $i=1 to n$
    (1) $\forall x_i \in X$;
    (2) search for $k$ nearest neighbors of $x_i$ from the same class, called nearHist $H_j$;
    (3) For each class $C \neq class(x_i)$
        (a) search for $k$ nearest neighbors of $x_i$ from each of the different class, called nearMisses $M_j$
        (b) For $g=1 to m$
            $W(g)=W(g)-(\sum_{j=1}^{k} diff(g,x_i,H_j))/kn+\sum_{C\neq class(x_i)}(\frac{P(C)}{1-P(class(x_i)}(\sum_{j=1}^{k} diff(g,x_i,M_j)/kn))$
          End;
        End;
Step 3: End.
```

2.2 Classification algorithm

2.2.1 Decision tree (DT)

Decision tree is an effective graphical method to express the process of classifying or evaluating an object. Through this graphical approach, it is clear how decisions can be made and models can be automatically built from tag samples. Decision trees are generally constructed using a bottom-up recursive approach, which the internal nodes in the tree graph of the decision tree represent the tests on the attributes and the branches represent the outputs of the tests, and each leaf node represents a different class.

At present, a series of specific decision tree learning models are produced under the general framework of decision tree, such as ID3, C4.5 and CART. Decision tree has the advantages of simple algorithm description, easy to understand, fast classification speed and high classification accuracy, and has been used in pattern recognize. However, the decision tree is poorly scalable and is especially easy to occur over-fitting phenomenon in the data containing noise.

2.2.2 Support vector machine (SVM)

Support Vector Machine (SVM) is a new effective machine learning algorithm based on structural risk minimization for resolving high dimensions, small samples and nonlinear problems, and is widely used in data mining field. The main idea of SVM is original feature space is mapped onto a high-dimension space by using an appropriate nonlinear function based on Mercer kernel theorem n, and the original nonlinear classification is converted to a linear classification problem in this high-dimension space, and then the optimal hyperplane is found to separate the samples in new feature space [31].

2.2.3 Extreme learning machine (ELM)

Extreme learning machine (ELM), proposed by Guang-Bin Huang in 2004, is a new neural network learning algorithm.ELM has faster learning speed and higher generalization performance because it uses single hidden layer feedforward neural network to reduce the learning time of the algorithm, and is widely used in regression and classification problems.

Figure 1 displays the network structure of ELM. The ELM is a three-layer network structure including one input layer, one hidden layer and one output layer. The unique feature of ELM is the input weights and thresholds in the hidden layer are randomly assigned and never adjusted. Compare with traditional learning algorithms, ELM has some advantages of simple structure, fast learning speed and high generalization performance.

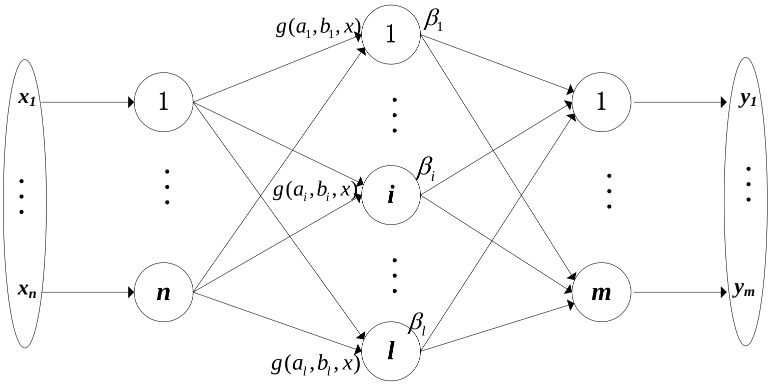


Fig.1. The network structure diagram of ELM

In the following, a multi-class classification task is assumed.

Suppose $T=(X,Y)$ is a training set containing Q samples , where $X_{n\times Q}$ is the inputs of training set containing Q samples and n attributes, $Y_{m\times Q}$ is the outputs of training set containing Q samples and m attributes, that is

$$X=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1Q} \\ x_{21} & x_{22} & \cdots & x_{2Q} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nQ} \end{bmatrix}_{n\times Q} \quad , \quad Y=\begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1Q} \\ y_{21} & y_{22} & \cdots & y_{2Q} \\ \vdots & \vdots & & \vdots \\ y_{m1} & y_{m2} & \cdots & y_{mQ} \end{bmatrix}_{m\times Q} .$$

ELM network is build according training set and has three layers structure with $n-l-m$, that is the number of nodes in input layer is n , the number of nodes in hidden layer is l and the number of nodes in output layer is m.

Suppose w denotes the weight vector connecting the hidden nodes and the input nodes, and β denotes the weight vector connecting the hidden node and the output nodes, that is

$$w=\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & & \vdots \\ w_{l1} & w_{l2} & \cdots & w_{ln} \end{bmatrix}_{l\times n} \quad , \quad \beta=\begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1m} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2m} \\ \vdots & \vdots & & \vdots \\ \beta_{l1} & \beta_{l2} & \cdots & \beta_{lm} \end{bmatrix}_{l\times m} .$$

Where w_{ji} denotes the weight vector connecting the *j*th hidden nodes and the *i*th input nodes, and β_{jk} denotes the weight vector connecting the *j*th hidden node and the *k*th output nodes.

Suppose b is the threshold vector in hidden layer, that is $b=[b_1,b_2,...,b_l]'_{l\times1}$.

So, $T=\begin{bmatrix}t_1, & t_2, & \cdots, & t_Q\end{bmatrix}'_{m\times Q}$ is the actual output of the network.

Where, $$t_j=\begin{bmatrix}t_{1j}\\ t_{2j}\\ \vdots\\ t_{mj}\end{bmatrix}_{m\times1}=\begin{bmatrix}\sum_{i=1}^{l}\beta_{i1}g(w_ix_j+b_i)\\ \sum_{i=1}^{l}\beta_{i2}g(w_ix_j+b_i)\\ \vdots\\ \sum_{i=1}^{l}\beta_{im}g(w_ix_j+b_i)\end{bmatrix}_{m\times1},\ j=1,2,3,\cdots,Q \qquad (2)$$

Where, $w_i=\begin{bmatrix}w_{i1},w_{i2},\cdots,w_{in}\end{bmatrix}, x_j=\begin{bmatrix}x_{j1},x_{j2},\cdots,x_{jn}\end{bmatrix}^T$, $g(x)$ is activation function in hidden layer.

For the sake of simplicity, the above formula can also be written as follows : $H\beta=T$

$$H(w_1,w_2,\cdots,w_l,b_1,b_2,\cdots,b_l,x_1,x_2,\cdots,x_Q)$$

$$=\begin{bmatrix}g(w_1x_1+b_1) & g(w_2x_1+b_2) & \cdots & g(w_lx_1+b_l)\\ g(w_1x_2+b_1) & g(w_2x_2+b_2) & \cdots & g(w_lx_1+b_l)\\ \vdots & \vdots & & \vdots\\ g(w_1x_Q+b_1) & g(w_2x_Q+b_2) & \cdots & g(w_lx_Q+b_l)\end{bmatrix}_{Q\times l} \qquad (3)$$

Suppose $T=(x_i,y_i)$ is training set, and parameters (w,b) are generated randomly, H is calculated according to above formula. Then, the weight vector β is calculated according following formula.

$$\beta=H^{+}T \qquad (4)$$

Where, H^{+} is the generalized inverse matrix of H .

The basic process of the Extreme Learning machine algorithm is shown in Method 2.

Method 2. Extreme Learning Machine algorithm

Input: Training set $T=\{(x_{1\times n},y_{1\times m}),(x_{2\times n},y_{2\times m}),...,(x_{Q\times n},y_{Q\times m})\}$,activation function $g(w_t,b_t,x_i)$,the number of hidden nodes L

Output: The weight vector connecting hidden nodes and output nodes β .

Step 1: Determine the structure of ELM network according to practical problem ,that is the number of nodes of every layers;

Step 2: Randomly generate the weight vector connecting hidden nodes and input nodes w_t and the thresholds of hidden nodes b_t ;

Step 3: Calculate hidden layer output matrix H according to formula (3);

Step 4: Calculate weight vector connecting hidden nodes and output nodes β according to formula (4);

2.3 The improved rotation forest algorithm

Rotation Forest is a new and effective ensemble classification method proposed by Rodriguez in 2006. The main success of rotation forest is to construct an ensemble classifier based on the feature disturbance and PCA transform. The purpose of rotating forest algorithm is to train multiple decision trees, and each decision tree is trained by using following way.

Firstly, the original feature set is randomly divided into several feature subsets, and then training subsets are generated according to above feature subsets obtained. Secondly, each feature subset is transformed into a new subset by a linear transformation, such as PCA. Thirdly, all the new feature subsets obtained are integrated according to a certain principle to reconstruct the original feature set, and then base classifiers are generated by using decision tree on above training set. Here, the linear transformation of the feature subset corresponds to the rotation of the feature axis, even if the feature axis has little rotation, the training set obtained by this method also has a large difference and is used to train base classifiers. The diversity of base classifiers has higher to improve generalization performance of rotation forest. Finally, the output results of all the decision trees are integrated to obtain the output of the ensemble system. Compare to bagging and boosting algorithm, Rotation forest has better generalization performance and robustness, and is widely used in the field of pattern recognize.

However, decision tree is employed to train base classifier in rotation forest algorithm, and decision tree will lead to the overfitting problem due to the complexity of classifier and local optimum, and it usually affect ensemble performance of rotation forest. In order to overcome the overfitting problem of decision tree and increase the diversity among base classifiers in rotation forest algorithm, an improved rotation forest is proposed. In improved rotation forest algorithm, extreme learning machine (ELM), support vector machine (SVM) and decision tree (DT) are employed to train base classifiers in the same ensemble to improve performance of rotation forest. The one hand, the ensemble of heterogeneous classifiers can increase diversity among base classifiers, on the other hand, extreme learning machine, support vector machine and decision tree complement each other.

Method 3 gives basic steps of improved rotation forest algorithm.

Method 3. The improved rotation forest algorithm

```
Input: training set $Q=\{X,Y\}=\{(x_i,y_i)\}_{i=1}^{N}$ contains $N$ samples and $p+1$ features, $X \in R^{N\times p}, Y \in R^{N\times 1}$; $T$ is the number of base classifiers; $f \in \{DT,SVM,ELM\}$ is a base classification algorithm; $x$ is a new sample to be classified; $C=\{c_1,c_2,\cdots,c_m\}$ is class labels set.
Output: ensemble classification result.
1.Generating phase of base classifiers
Step 1: For $t=1,2,3\ldots,T$
Step 2: Calculate the rotation matrix $R_t^a$ for the tth classifier $C_t$.
    Step 2.1: Randomly split the original attribute set $F$ into $K$ subsets $F_{t,k}(k=1,2,\ldots,K)$.Where any two subsets $F_{t,i}$, $F_{t,j}$ don't intersect and the attribute number of every subsets almost equal.
    Step 2.2: For $k=1,2,3\ldots,K$
        (1)Select the column of $X$ that correspond to the attributes in $F_{t,k}$ to build a training set $X_{t,k}$.
        (2)Generate a training subset $X'_{t,k}$ (with 75% sample size of $X_{t,k}$) from $X_{t,k}$ by using bootstrap.
        (3)Obtain $D_{t,k}$ from $X'_{t,k}$ based on PCA, whose ith column consist of the coefficients of the ith principal component.
    Step 2.3: End
    Step 2.4: Arrange $D_{t,k}$ $(k=1,2,\ldots,K)$ into a diagonal matrix $R_t=\begin{bmatrix} D_{t1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & D_{tk} \end{bmatrix}$.
    Step 2.5: Rearrange the rows of $R_t$ to construct the rotation matrix $R_t^a$ so that they correspond to the original features in $F$.
    Step 2.6: Build the base classifier $C_t$ from training set $[XR_t^a,Y]$ by using $f \in \{DT,SVM,ELM\}$.
Step 3: End
2.Integrating phase of base classifiers
Step 4: New sample $x$ is rotated according to rotation matrix $R_t^a$ , that is $x'=x\cdot R_t^a$
Step 5: Calculate the probability that the sample $x$ is assigned to the class $c_j$
```

$$u_j(x) = \frac{1}{T}\sum_{i=1}^{T} d_{ij}(x'), j = 1, 2, \cdots, m$$

where $d_{ij}(x')$ is a probability that the sample x is assigned to the class c_j by classifier C_i

Step 6: The samples x is classified into the class with the highest probability, i.e. the classification results are integrated by following formula: $C^*(x) = \arg(\max x_{j=1}^{c}(u_j))$.

3. Our proposed algorithm

In rotation forest, the diversity among base classifiers is enhanced by feature segmentation and the accuracy of base classifiers is increased by using PCA and keeping all principal components. Therefore, generalization performance of rotation forest is improved further.

However, decision tree is too complex to lead to over fitting problem in the learning processing of decision tree and affect ensemble performance of rotation forest algorithm.ELM can resolve over fitting problem because of adaptation and the training speed of ELM is quickly. In addition, in order to further increase the diversity among base classifiers and the precision of base classifiers, the heterogeneous classifiers of extreme learning machine, support vector machine and decision tree are employed to train base classifiers, and then it improves the ensemble generalization performance.

This paper proposes an improved rotation forest algorithm based on heterogeneous classifiers ensemble to improve classification performance of gene expression profiles. The steps of our algorithm are given as follows.

(1) Feature gene selection based on relieff algorithm

In order to remove irrelevant and redundant genes from original gene expression profile to improve the quality of the data and decrease computation complexity, all the genes are ranked by using relieff algorithm and top-ranked genes are selected to build a new training subset.

(2) Gene expression profile classification based on improved rotation forest algorithm

In order to overcome the overfitting problem of decision tree and increase the diversity among base classifiers in rotation forest algorithm, extreme learning machine, support vector machine and decision tree are employed to train base classifiers in the same ensemble to improve performance of rotation forest. The one hand, the ensemble of heterogeneous classifiers can increase diversity among base classifiers, on the other hand, extreme learning machine, support vector machine and decision tree complement each other.

4. Experiment

4.1 Experimental datasets

In order to verify the performance of our proposed algorithm, nine well-known benchmark cancer gene expression profiles are selected to implement in our experiment. The characteristics of nine datasets are described in table 1.

Table 1. Benchmark cancer gene expression profiles

No	Data set	classes	genes	samples	training	testing
1	Colon	2	2000	62	43	19
2	CNS	2	7129	60	42	18
3	DLBCL	2	7129	77	32	45
4	Gliomas	2	12625	50	20	30
5	Ovarian	2	15154	253	177	76
6	Leukemia	3	7129	72	38	34
7	MLLLeukemia	3	12582	72	27	45
8	SRBCT	4	2308	83	63	20
9	ALL	6	12625	248	148	100

(1) Colon contains 2000 genes and 62samples, where 22 are normal and 40 are normal.

(2) CNS(Central Nervous System) contains 7129 genes and 60 samples, where 21 are survivors and 39 are failures.

(3) DLBCL contains 7129 genes and 77 samples, where 19 are Follicular lymphoma and 58 are Diffuse large B-cell lymphoma.

(4) Gliomas contains 12625 genes and 50 samples, where 28 are normal and 22 are patients.

(5) Ovarian contains 15154 genes and 253 samples, where 162 are normal and 91 are patients.

(6) Leukemia contains 7129 genes and 72 samples, where 9 are acute lymphoblastic leukemia T-cell (ALL–T), 38 are acute lymphoblastic leukemia B-cell (ALL- B) and 25 are acute myelogenous leukemia (AML).

(7) MLLLeukemia contains 12582 genes and 83 samples, where 24 are ALL,20 are MLL and 28 are AML.

(8) SRBCT contains 2308 genes and 72 samples, where 29 are EWS (Ewing sarcoma), 11 are BL(Burkitt lymphoma),18 are NB (Neuroblastoma) and 25 are RMS (Rhabdomyosarcoma).

(9) ALL(Acute lymphoblastic leukemia) contains 12625 genes and 248 samples, where15 are BCR-ABL,27 are E2A-PBX1,64 are Hyperdiploid>50, 20 are MLL, 43 are T-ALL and 79 are TEL-AML.

4.2 Experimental algorithms and the parameters setting

In order to compare effectiveness of our proposed algorithm, five popular ensemble algorithms are used to compare with our method. In addition, the experiment are repeated 20 times independently, and then the average results of 20 times are as final results to guarantee non-contingency of the results of different algorithms.

algorithm 1:Relief+Bagging (Decision Tree); algorithm 2:Relief+AdaBoost (Decision Tree);
algorithm 3:Relief+Rotation Forest (SVM); algorithm 4: Relief+Rotation Forest (Decision Tree);
algorithm 5: Relief+Rotation Forest (ELM); our algorithm: Relief+Improved Rotation Forest.

For the SVM classifier in our algorithm and algorithm 3, the gamma in the kernel function and the parameter C of C-SVC are randomly selected.

Runtime environment: All methods used in this paper are coded in MATLAB with 64 bit running on an Inter(R) Core(TM) i3PC with dual-core3.0 GHz CPU and 4G memory.

4.3 Experimental results and analysis

4.3.1 The comparison of classification accuracy of different algorithms

In order to investigate the relationship between base classifier number and ensemble performance, the number of the base classifiers in the ensemble respectively equals 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55 and 60 in our experiments.

Table 2 displays the average results of different algorithms when the number of base classifiers is equal to 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55 and 60.

Table 2. The average classification accuracy (%)

Dataset	algorithm1	algorithm2	algorithm3	algorithm4	algorithm5	our algorithm
Colon	69.74	68.86	73.68	71.93	75.39	**78.68**
CNS	63.43	56.48	77.78	76.44	87.69	**93.24**
DLBCL	84.44	84.81	80.00	88.52	84.54	**88.89**
Gliomas	66.39	66.67	83.33	74.33	78.75	**84.25**
Ovarian	98.79	98.90	97.37	97.45	**100.00**	99.96
Leukemia	**96.81**	92.16	76.47	88.43	89.46	92.43
MLLLeukemia	53.15	54.07	86.67	83.17	92.65	**92.68**
SRBCT	86.67	97.50	90.00	92.75	98.92	**99.38**
ALL	94.75	96.00	93.00	96.38	97.77	**98.04**
avg	*79.35*	*79.50*	*84.26*	*85.49*	*89.46*	***91.92***

It is clean our algorithm has the highest classification accuracy on the most datasets of nine datasets from table 2.The detailed conclusions are as follows.

(1) Our algorithm yields top-notch performance among six methods on Colon, CNS, DLBCL, Gliomas,MLLLeukemia,SRBCT and ALL dataset.

For Colon, the average accuracy of our algorithm is 78.68%, which is 8.94% higher than that of algorithm 1, 9.82 % higher than that of algorithm 2,5% higher than that of algorithm 3, 6.75% higher than that of algorithm 4, 3.29% higher than that of algorithm 5.

For CNS, the average accuracy of our algorithm is 93.24%, which is 29.81% higher than that of algorithm 1, 36.76% higher than that of algorithm 2, 15.46% higher than that of algorithm 3, 16.8 % higher than that of algorithm 4, 5.55 % higher than that of algorithm 5.

For DLBCL, the average accuracy of our algorithm is 88.89%, which is 4.45% higher than that of algorithm 1, 4.08% higher than that of algorithm 2, 8.89% higher than that of algorithm 3, 0.37% higher than that of algorithm 4, 4.35 % higher than that of algorithm 5.

For Gliomas, the average accuracy of our algorithm is 84.25%, which is 17.86% higher than that of algorithm 1, 17.58% higher than that of algorithm 2, 0.92% higher than that of algorithm 3, 9.92% higher than that of algorithm 4, 5.5 % higher than that of algorithm 5.

For MLLLeukemia, the average accuracy of our algorithm is 92.68%, which is 39.53% higher than that of algorithm 1, 38.61% higher than that of algorithm 2, 6.01% higher than that of algorithm 3, 9.51% higher than that of algorithm 4, 0.03% higher than that of algorithm 5.

For SRBCT, the average accuracy of our algorithm is 99.38%, which is 12.71% higher than that of algorithm 1, 1.88% higher than that of algorithm 2, 9.38% higher than that of algorithm 3, 6.63% higher than that of algorithm 4, 0.46% higher than that of algorithm 5.

For ALL, the average accuracy of our algorithm is 98.04%, which is 3.29% higher than that of algorithm 1,2.04% higher than that of algorithm 2, 5.04% higher than that of algorithm 3, 1.66% higher than that of algorithm 4, 0.27% higher than that of algorithm 5.

(2) Our algorithm (99.96%) don't obtain the best classification accuracy on Ovarian, which simply fall below algorithm 5(100%).Our algorithm (92.43%) don't obtain the best classification accuracy on Leukemia, which simply fall below algorithm 1 (96.81%).

"*avg*" shows summarized result which is calculates by averaging the accuracy over all the datasets. The average accuracy of our algorithm is 91.92%, and is improved 12.57%, 12.42%, 7.66%, 6.43% and 2.46% to compare with that of algorithm 1, 2, 3, 4 and 5, respectively.

In addition, we find that algorithm 5 yields top-notch performance among three algorithms (algorithm 3,4 and 5).The reason is ELM is weak classifier and it's stability is weak than SVM and Decision Tree. Hence, ELM benefits to improve ensemble performance because ELM can increase the diversity among base classifiers.

4.3.2 The comparison of different algorithms based on geometry accuracy ratio

Table 3 displays geometry mean accuracy ratio of different algorithms on all the dataset. Geometry accuracy ratio (GMAR) are employed to compare relative classification performance of different algorithms on all the datasets [32].

The definition of Geometry accuracy ratio is as follows: $GMAR = (\prod_{i=1}^{n} \frac{E_{iA}}{E_{iB}})^{\frac{1}{n}}$ (5)

where E_{iA} and E_{iB} represent accuracy of algorithm A and B on the $i\ th$ dataset, respectively. n is the number of datasets.

In table 3, "r" represents geometry mean value of row/col , "s" represents win/tie/loss, where win, tie, loss represents the number of datasets of col>row, col=row and col<row, respectively. where “row” represents the classification accuracy of each row corresponding algorithm on all the datasets, “col” represents the classification accuracy of each column corresponding algorithm on all the datasets.

Give an example to explain the statistic r and s:

$r = 1.0023 = (\frac{69.74}{68.86} \times \frac{63.43}{56.48} \times \frac{84.44}{84.81} \times \frac{66.39}{66.67} \times \frac{98.79}{98.90} \times \frac{96.81}{92.16} \times \frac{53.15}{54.07} \times \frac{86.67}{97.50} \times \frac{94.75}{96})^{\frac{1}{9}}$,it is shows algorithm 1 has better than algorithm 2 as a whole because of 1.0023>1; *s*=6/0/3 means algorithm 2 outperform algorithm 1 on six datasets, and algorithm 2 don not outperform algorithm 1 on three dataset, respectively.

Table 3. The comparison of different algorithms on all the datasets

		algorithm2	algorithm3	algorithm4	algorithm5	our algorithm
algorithm1	*r*	1.0023	0.9262	0.9143	0.8726	0.8476
	s	6/0/3	5/0/4	7/0/2	8/0/1	8/0/1
algorithm2	*r*		0.9241	0.9123	0.8707	0.8457
	s		4/0/5	6/0/3	7/0/2	9/0/0
algorithm3	*r*			0.9872	0.9421	0.9151
	s			5/0/4	8/0/1	9/0/0
algorithm4	*r*				0.9544	0.9270
	s				8/0/1	9/0/0
algorithm5	*r*					0.9713
	s					8/0/1

The following conclusions are obtained by analyzing r and s in table 3.

(1) *r*=0.9713<1 means our algorithm is better than algorithm 5. Similarly, our algorithm is better than algorithm 4, algorithm 3,algorithm 2 and algorithm 1 because corresponding *r* is less than 1.In addition, we find that our algorithm defeats the others on the 90% datasets according s .

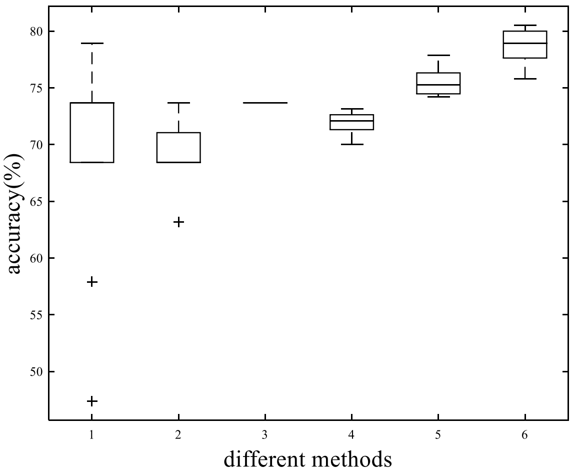
(2) The classification results from good to bad is as follows: our algorithm, algorithm 5, algorithm 4, algorithm 3, algorithm 1 and algorithm 2.

4.3.3 The stability of different algorithms

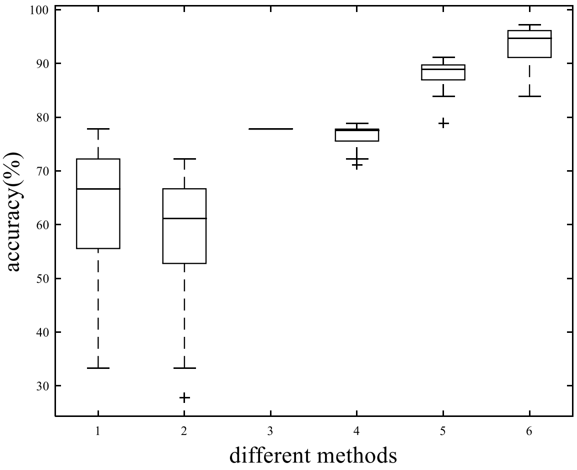
The stability is an important performance of classification algorithm, and the boxplot is used to evaluate the stability. Figure 2 displays the boxplot of different algorithms on all the datasets. Where, 1, 2, 3, 4, 5 and 6 on the horizontal axis represents algorithm 1, 2, 3, 4, 5 and our algorithm.

We find that algorithm 3 has the best stability from figure 2, and the reason is SVM is a strong learning algorithm, which is insensitive for the change of samples. However, Because of this, the ensemble performance by using SVM is relatively poor.

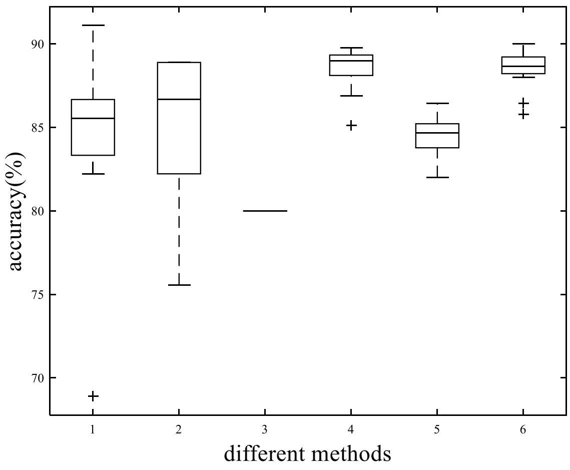
The stability of our algorithm is not lower than algorithm 4 and algorithm 5 except on Colon, CNS and Leukemia dataset, and is much better than algorithm 1 and algorithm 2 on all the datasets. Overall, the stability of our algorithm is relatively good on the most datasets.



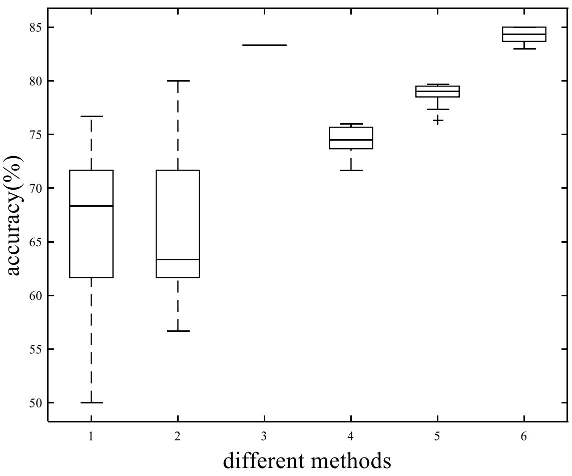
Colon



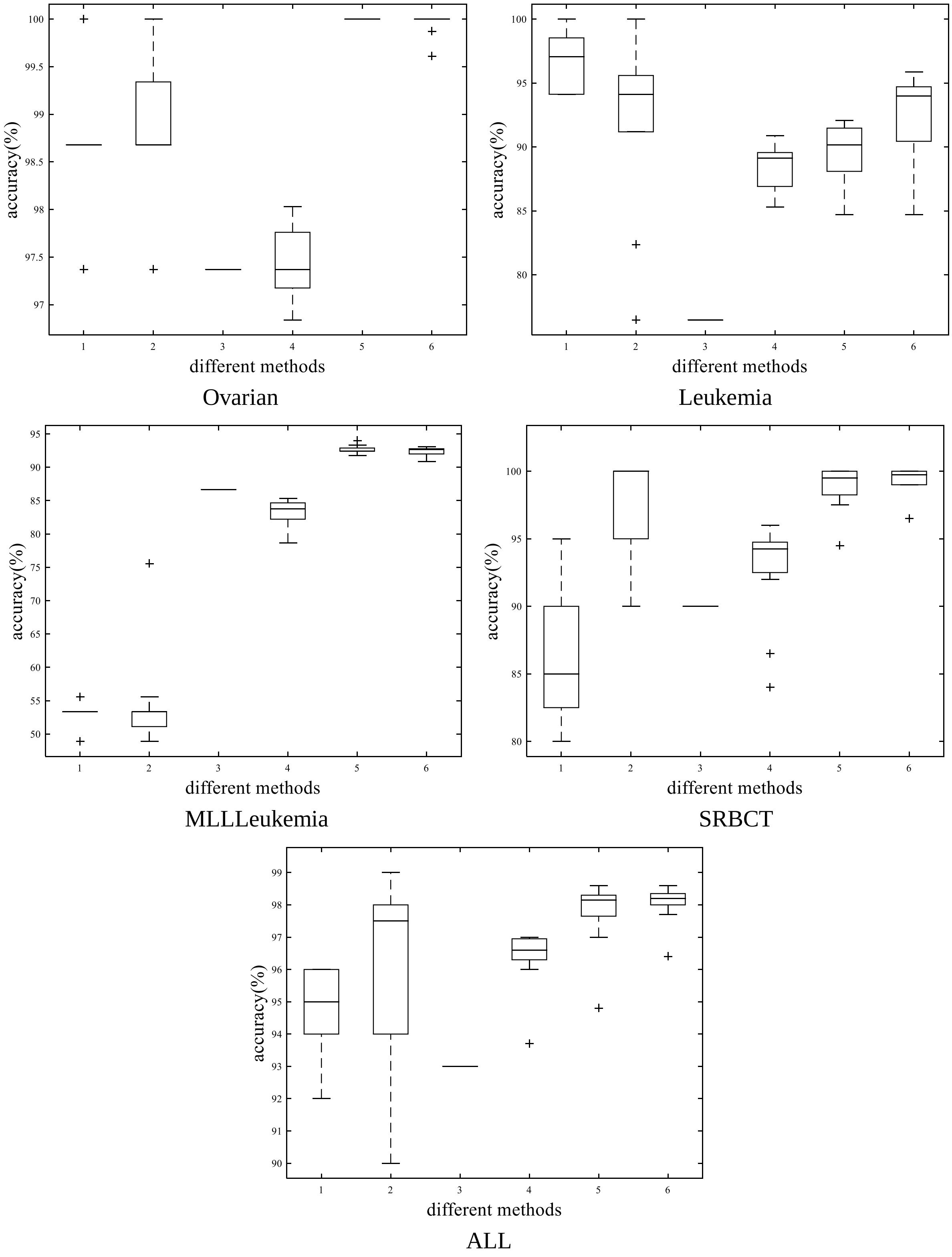
CNS



DLBCL



Gliomas



Ovarian

Leukemia

MLLLeukemia

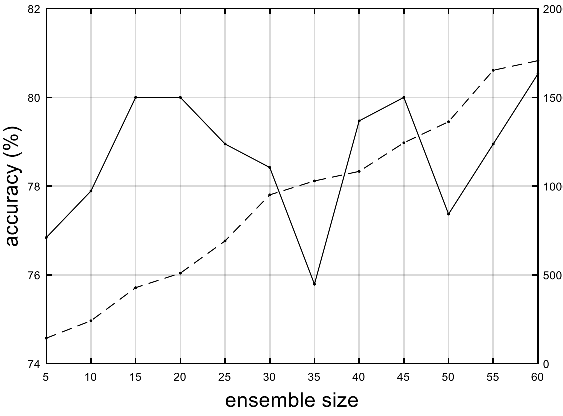
SRBCT

ALL

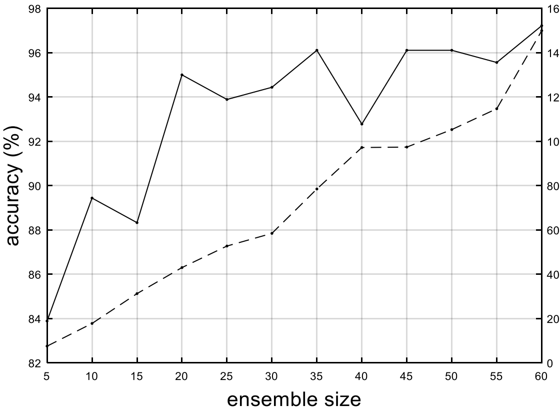
Fig.2. The boxplot of different algorithms

4.3.4 The relationship between the number of base classifiers and classification performance

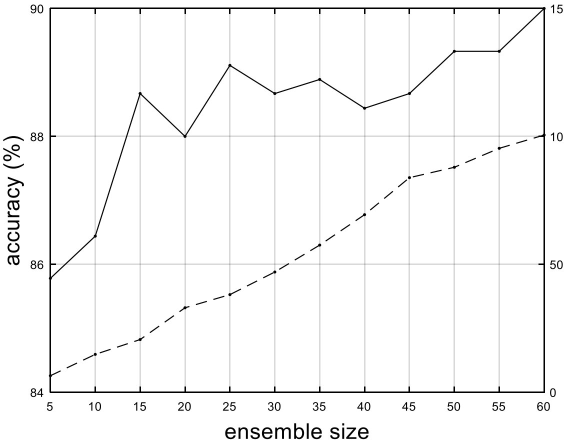
Figure 3 displays influence of number of base classifiers on classification performance by using our algorithm .In figure 3, “solid line” and “dotted line” represent classification accuracy and run time, respectively.



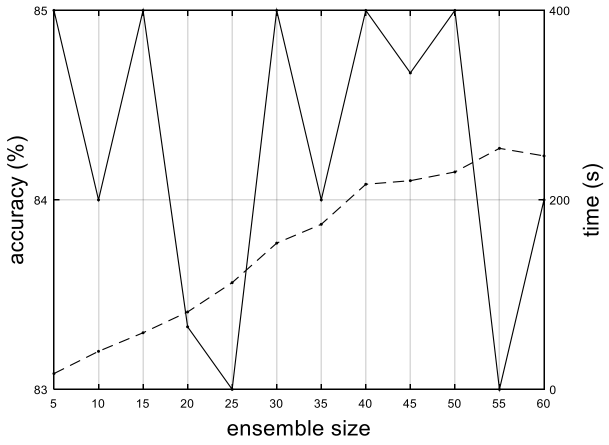
Colon



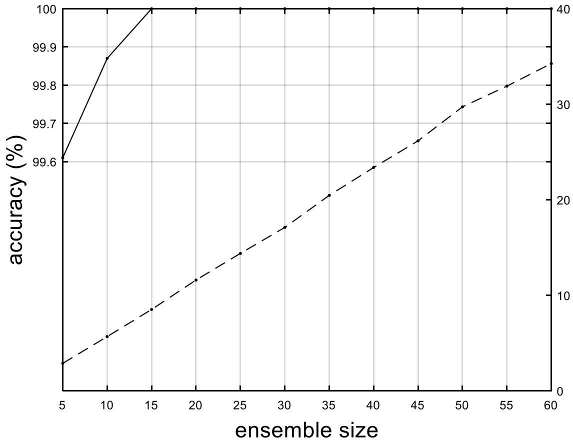
CNS



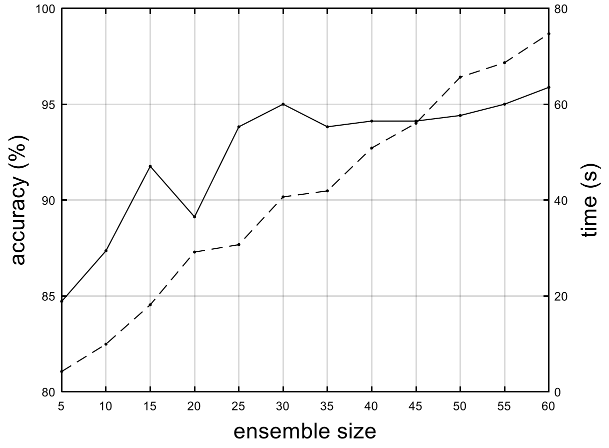
DLBCL



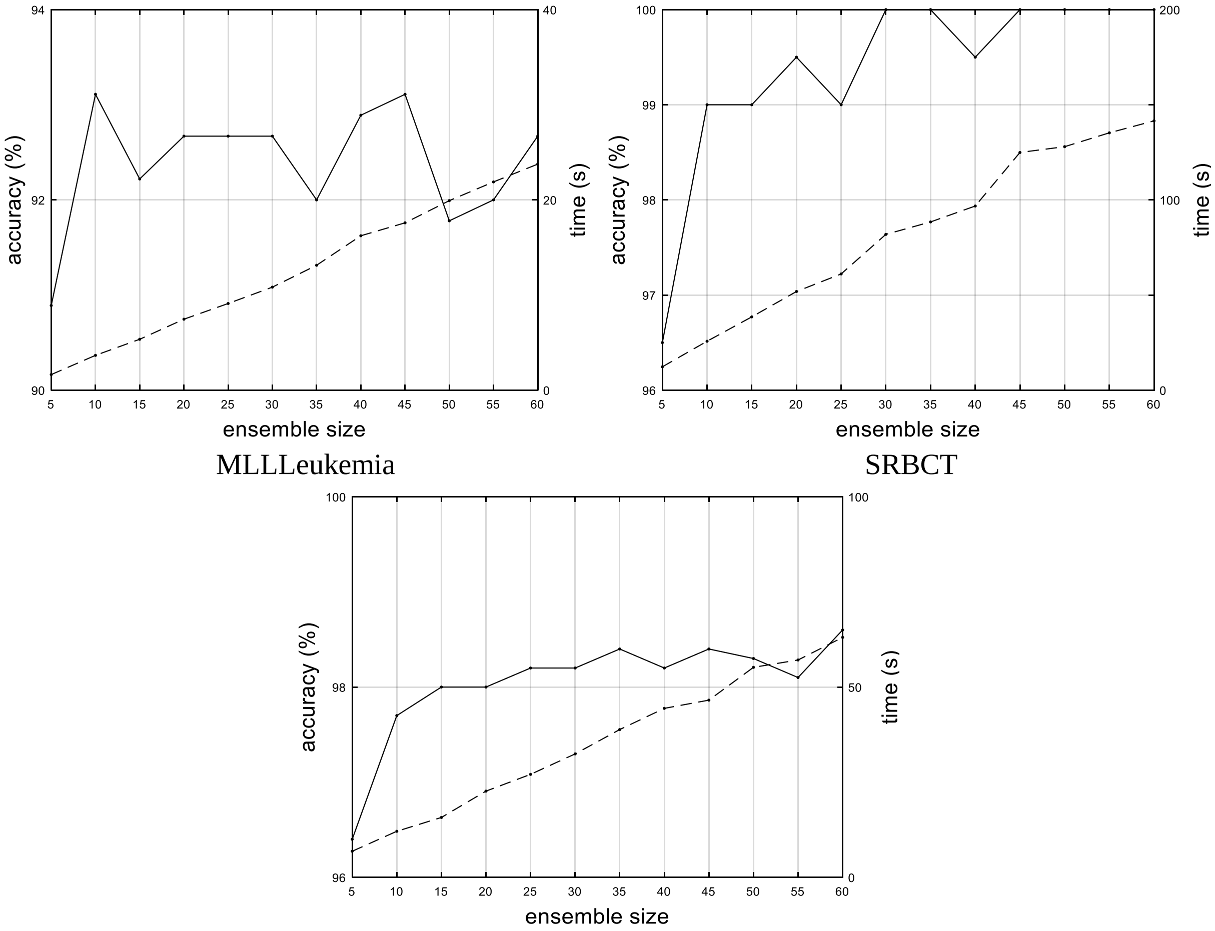
Gliomas



Ovarian



Leukemia



MLLLeukemia

SRBCT

ALL

Fig.3. Variation of number of base classifiers and classification performance

(1) Classification accuracy

It is clearly that the number of base classifiers has a great influence on the classification accuracy from figure 3. The classification accuracy of our algorithm is low when the number of base classifiers is 5, and then the classification accuracy quickly increases with the number of base classifiers, but the classification accuracy basically maintains at a high level when the number of base classifiers is about between 20 and 60.

(2) Run times

According to figure 3, the run time of our algorithm is linearly increased with the growth of the number of base classifiers.

Therefore, according to classification accuracy and run time, the performance of our algorithm is good when the number of base classifiers is 20 to 40. It is a reference for selecting the number of base classifiers in ensemble from classification accuracy and time efficiency.

5. Conclusion

In this paper, an improved rotation forest algorithm is proposed to improve the classification performance. In this algorithm, extreme learning machine, support vector machine and decision tree were used to train multiple heterogeneous base classifiers in ensemble, and it can increase diversity among base classifiers to improve ensemble performance further. Experimental results indicate our algorithm has higher classification and better stability than rotation forest algorithm, and it is effective for classifying gene expression profile.

Acknowledgements

This paper is supported by Scientific Research Program Funded by Shaanxi Provincial Education Department (16JK1149) and Scientific Research Program Funded by Shaanxi University of Technology (SLGKY16-15,SLGKY-29).

References

1. M.B Kursa, Robustness of random forest-based gene selection methods, 2014, BMC bioinformatics, vol.15, no.1, pp.1-8.
2. T. Chen, H.F. Xue, Z.L. Hong, M. Cui, H. Zhao, A hybrid ensemble method based on double disturbance for classifying microarray data, 2015, Bio-Medical Materials and Engineering, vol.26, no.1, pp.1961-1968.
3. Y. Xiao, T.H. Hsiao, U.Suresh, H.I. Chen, X. Wu, S.E. Wolf. Y. Chen, A novel significance score for gene selection and ranking, 2014, Bioinformatics, vol.30, no.6, pp. 801-807.
4. T. Chen, Z.L. Hong, H. Zhao, J. Wei, A novel feature gene selection method based on neighborhood mutual Information, 2015, International Journal of Hybrid Information Technology, vol.8, no.7, pp.277-292.
5. K.H. Chen, K.J. Wang, M.L. Tsai, K.M. Wang, A.M. Adrian, W.C. Cheng, T.S. Yang, N.C. Teng, K.P. Tan, K.S. Chang, Gene selection for cancer identification: a decision tree model empowered by particle swarm optimization algorithm, 2014, BMC bioinformatics, vol.5, no.1, pp.49-56.
6. R.C. Barros, M.P. Basgalupp, A.A. Freitas, A. De Carvalho, Evolutionary design of decision-tree algorithms tailored to microarray gene expression data sets, 2014, IEEE Transactions on Evolutionary Computation, vol.18, no.6, pp.873-892.

7. B. Chandra, K.V.N. Babu, Classification of gene expression data using spiking wavelet radial basis neural network,2014,Expert systems with applications, vol.41, no.4, pp.1326-1330.

8. C. Bazot, N. Dobigeon, J.Y. Tourneret, A.K. Zaas, G.S. Ginsburg, A.O. Hero, Unsupervised bayesian linear unmixing of gene expression microarrays, 2013, BMC bioinformatics, vol.14, no.1, pp.99-108.

9. S. Kar, K.D. Sharma, M. Maitra, Gene selection from microarray gene expression data for classification of cancer subgroups employing PSO and adaptive K-nearest neighborhood technique, 2015, Expert Systems with Applications, vol. 42, no.1, pp. 612-627.

10. C. Das, S. Bose, M. Chattopadhyay, S. Chattopadhyay, A novel distance-based iterative sequential KNN algorithm for estimation of missing values in microarray gene expression data, 2016, International Journal of Bioinformatics Research and Applications, vol.12, no.4, pp. 312-342.

11. H. Saberkaria, M. Shamsi, M. Joroughi, F. Golabi, M.H. Sedaaghi, Cancer classification in microarray data using a hybrid selective independent component analysis (SICA) and υ-Support Vector Machine (υ-SVM) Algorithm, 2014, Journal of medical signals and sensors, vol.4, no.4, pp.291-299.

12. T. Chen, Classification algorithm on gene expression profiles of tumor using neighborhood rough set and support vector machine, 2014, Advanced Materials Research, vol.850, pp.1238-1242.

13. H. Zhao, Intrusion detection ensemble algorithm based on bagging and neighborhood rough set, 2013, International Journal of Security and Its Applications, vol.7, no.5, pp.193-204.

14. T. Chen, Z.L. Hong, A combined svm ensemble algorithm based on KICA and KFCM, 2012, Software Engineering and Knowledge Engineering: Theory and Practice, China, pp.585-592.

15. L. Shi, L. Xi, X. Ma, M. Weng, X. Hu, A novel ensemble algorithm for biomedical classification based on ant colony optimization, 2011, Applied Soft Computing, vol.11, no.8, pp.5674-5683.

16. L. Breiman, Bagging predictors, 1996, Mach. Learn, vol.24, no.1, pp.123-140.

17. R. Schapire, The strength of weak learnability, 1990, Mach. Learn, vol.5, no.2, pp.197-227.

18. T.K. Ho, The random subspace method for constructing decision forests, 1998, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.20, no.8, pp.832-844.
19. R. Díaz-Uriarte, S.A. DeAndres, Gene selection and classification of microarray data using random forest, 2006, BMC bioinformatics, vol.7, no.1, pp.1-13.
20. C.X. Zhang, J.S. Zhang, RotBoost: A technique for combining Rotation Forest and AdaBoost, 2008, Pattern recognition letters, vol.29, no.10, pp.1524-1536.
21. J.J. Rodriguez, L.I. Kuncheva, C.J. Alonso, Rotation forest: a new classifier ensemble method, 2006, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.28, no.10, pp.1619-1630.
22. L. Zhang, P.N. Suganthan, Random forests with ensemble of feature spaces, 2014, Pattern Recognition, vol.47, no.10, pp.3429-3437.
23. H. Lu, L. Yang, K. Yan, Y. Xue, Z. Gao, A cost-sensitive rotation forest algorithm for gene expression data classification, 2016, Neurocomputing, vol.228, no.8, pp.270-276.
24. S. Kotsiantis, Combining bagging, boosting, rotation forest and random subspace methods, 2011, Artificial Intelligence Review, vol.35, no.3, pp.223-240.
25. A. Ozcift, A. Gulten, Classifier ensemble construction with rotation forest to improve medical diagnosis performance of machine learning algorithms, 2011, Computer methods and programs in biomedicine, vol.104, no.3, pp.443-451.
26. G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: theory and applications, 2006, Neurocomputing , vol.70, no.1, pp. 489-501.
27. G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, 2004, IEEE International Joint Conference, pp.985-990.
28. G.B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, 2012, Systems, Man, and Cybernetics, Part B: Cybernetics,vol.42,no.2,pp.513-529.
29. Y. Lan, Y.C. Soh, G.B. Huang, Two-stage extreme learning machine for regression, 2010, Neurocomputing, vol.73, no.16, pp.3028-3038.

30. I. Kononenko, Estimating attributes: analysis and extensions of RELIEF, 1994, Proceedings of the European conference on machine learning, Lecture notes in computer science, pp.784:171-182.

31. M. Aly, Survey on multiclass classification methods, 2005, Neural Network, pp.1-9.

32. G.I. Webb, Multiboosting: A technique for combining boosting and wagging, 2000, Machine learning, vol.40, no.2, pp.159-196.