Neurocomputing 99 (2013) 581–591

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Letters

Dynamic classifier ensemble using classification confidence

Leijun Li, Bo Zou, Qinghua Hu*, Xiangqian Wu, Daren Yu

Harbin Institute of Technology, Harbin 150001, PR China

ARTICLE INFO

Article history:
Received 16 February 2012
Received in revised form
26 July 2012
Accepted 31 July 2012
Communicated by Zhouchen Lin
Available online 24 August 2012

Keywords:
Dynamic classifier ensemble
Classification confidence
Margin distribution

ABSTRACT

How to combine the outputs from base classifiers is a key issue in ensemble learning. This paper presents a dynamic classifier ensemble method termed as DCE-CC. It dynamically selects a subset of classifiers for test samples according to classification confidence. The weights of base classifiers are learned by optimization of margin distribution on the training set, and the ordered aggregation technique is exploited to estimate the size of an appropriate subset. We examine the proposed fusion method on some benchmark classification tasks, where the stable nearest-neighbor rule and the unstable C4.5 decision tree algorithm are used for generating base classifiers, respectively. Compared with some other multiple classifier fusion algorithms, the experimental results show the effectiveness of our approach. Then we explain the experimental results from the view point of margin distribution.

1. Introduction

Ensemble learning has been a hot topic in pattern recognition and machine learning domains for more than 20 years due to good generalization ability [1,25,26,36]. It means training a group of base learners which jointly solve a given classification or regression task with a fusion strategy. It has been theoretically and empirically demonstrated that combining multiple classifiers can substantially improve the classification performance of its constituent members [2,17,27,34].

How to effectively combine the outputs of the base classifiers is a key issue in ensemble learning. So far a number of fusion strategies have been proposed. In general, there are two basic fusion schemes to follow: one is to use the fixed base classifiers combination for all the test samples. The fixed combination can be constructed with all the base classifiers [4,10,18] or only a subset of them [6,15,20–23,33,35,37]. The other scheme is called dynamic classifier selection, which selects only one classifier to classify a given sample and the selected classifier is thought most likely to be correct for the given sample [12–14,30]. Inspired by the idea of dynamic classifier selection, we propose a dynamic classifier ensemble method in this paper based on the classification confidence of the test sample (termed as DCE-CC). However different from the dynamic classifier selection, DCE-CC dynamically selects a subset of classifiers for a given sample.

The fusion algorithms of using all the base classifiers include simple voting (SV) rule [18], linear weighted voting [4,10], and so on. These algorithms aim at combining all the outputs of the base classifiers in some way to improve the performance of the base classifiers. However it results in a large memory requirement and a slow classification speed [20].

In order to alleviate the drawbacks, selective ensemble algorithms, which select a fraction of the classifiers from the original ensemble and then combine them with simple or weighted voting, were proposed. The key problem is how to find the optimal subset of the base classifiers [20]. In [35], based on the evolved weights, GASEN was designed to select some neural networks to constitute the ensemble. Then in [15], the genetic algorithm was applied to find an approximate solution to the boosting pruning problem. In [33] the subset selection problem was viewed as a quadratic integer programming problem to search the classifiers subsets that have the optimal accuracy-diversity trade-off and semi-definite programming was used to get a good approximate solution. More recently, a new weighted combination method based on the linear programming was constructed for sparse ensemble [37]. However GASEN and semi-definite programming are all global optimization methods to search the appropriate classifiers subset and their computational costs are very high. To overcome this drawbacks, some suboptimal ensemble pruning methods were proposed, such as expectation propagation [6], margin distance minimization (MDM) [21], orientation ordering [22], boosting-based ordering [23], and so on.

These above fusion methods are based on the assumption that the classifiers are independent and equally reliable [8]. However, it is difficult to satisfy such an assumption in real applications. In the scheme of dynamic classifier selection [12–14,30], for each test sample, only one classifier is selected to classify it. The selected classifier for the given test sample is thought to most likely classify it correctly. Therefore it can avoid the error-independence assumption. These dynamic classifier selection algorithms include dynamic classifier selection based on classifier's local accuracy proposed in

* Corresponding author.
E-mail address: huqinghua@hit.edu.cn (Q. Hu).

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.07.026

[30], dynamic classifier selection based on multiple classifier behavior [12], and so on. In [30], in order to classify an unknown test sample, the ℓ-nearest neighbors surrounding the sample were firstly estimated and then the classifier with the highest accuracy in the local regions was selected to classify the test sample. Since this algorithm is devised based on the ℓ nearest neighbors, its performance is affected by the choice of ℓ.

Margin distribution is thought as an important factor to improve the generalization performance of classifiers [3,28] and the effectiveness of the ensemble learning methods, especially the boosting method, has to be explained from the improvement of the margin distribution on training sets [29,32]. Therefore improving the margin distribution on the training sets is an effective way to boost the generalization capability of ensemble learning. In this paper a dynamic classifier ensemble method called DCE-CC is proposed based on the classification confidence and the optimization of margin distribution on the training sets. It dynamically selects a subset of classifiers to classify a test sample with the weighted voting and the classification confidence of the test sample on the selected classifiers are the first K largest. In order to estimate the size K, we exploit the optimization of margin distribution based on the ordered aggregation technique [20]. Then the test sample is classified by the selected classifiers using the weighted voting and the weight is the corresponding classification confidence. It is worth remarking that since the classification confidence order for different samples are usually different, the selected classifiers for different samples is usually different.

In this paper, the ordered aggregation technique is utilized to find an appropriate classifier subset for each sample, where the weights of base classifiers are learned by minimization of margin loss on the training sets. This strategy has been used in the selective ensembles such as Complementarity Measure [21], margin distance minimization (MDM) [21], orientation ordering [22] and boosting-based ordering [23]. Then the performance of these algorithms has been analyzed in [20]. The key problem for the ordered aggregation technique is how to reorder the classifiers in the ensemble process. In DCE-CC, the order of aggregation of the classifiers is estimated according to the classification confidence of the sample.

The major contributions in this work are listed as follows. First, based on the classification confidence, DCE-CC and a new margin are proposed. Second, the optimization of margin distribution and the ordered aggregation technique are utilized for the estimation of the size of an appropriate subset. Besides, the weighted voting based on the classification confidence is proposed to combine the selected classifiers for an unseen sample. Third, we use the stable nearest-neighbor rule and the unstable C4.5 decision tree algorithm to train base classifiers, a set of experiments are presented to test the rationality and the effectiveness of the proposed algorithm. DCE-CC is competent compared with the single classifier, a dynamic classifier selection algorithm DCS-LA and a selective ensemble algorithm called MDM [21].

The rest of the paper is organized as follows. Related work and a margin based on the classification confidence are introduced in Section 2. DCE-CC algorithm and the generation algorithm of the base classifiers are presented in Section 3. Then we discuss the rationality of DCE-CC and present our experimental results in Section 4. Finally, Section 5 offers the conclusions and future work.

2. Related works

Denote by $X=[x_1,\ldots,x_n]$ the training set which contain n samples and $D_1,\ldots,D_L$ the classifiers in the ensemble. Let $Y=[y_1,\ldots,y_n]$ be the true class labels of training set and $H_i=[h_{1i},\ldots,h_{ni}]$ be class labels of training set estimated by the classifier D_i. Besides, every classifier D_i provides for the training set the classification confidence $R_i=[r_{1i},\ldots,r_{ni}](r_{ij}\in[0,1])$. Intuitively, the higher the confidence provided by the classifier, the higher the probability that the classifier has correctly classified the sample.

Since DCE-CC algorithm proposed in this paper utilizes the optimization of margin distribution, the definition of margin is first given. In [29], the margin of a sample is defined as the difference between the number of correct votes and the maximum number of votes received by any incorrect label.

Definition 1 (*Schapire et al. [29]*). For $x_i\in X(i=1,2,\ldots,n)$, let $\omega=\{\omega_1,\ldots,\omega_c\}$ be the set of class labels, $H=\{h_{ij}|h_{ij}\in\omega\}$ be the classification decision of x_i by the classifier D_j $(j=1,2,\ldots,L)$. The margin of the sample x_i is denoted by

$$M_1(x_i)=\frac{N(\omega_i)-\max\{N(\omega_j)|i\neq j\}}{L} \tag{1}$$

where L is the number of the classifiers, $N(\omega_i)$ means the number of ω_i in H and ω_i is the true label of x_i.

From Definition 1, we can see that the margin is a number in the range $[-1,1]$ and a sample x_i is classified correctly if and only if $M_1(x_i)>0$. A large positive margin can be interpreted as a "confident" correct classification, so the larger the margin on the test samples, the better the classification accuracy on the test samples. When the outputs of the classifiers are given, we expect the margin of each sample is as large as possible.

The margin distribution on the training sets is an important factor for the generalization performance of the ensemble learning methods. In [29], the generalization error of voting classifiers is bounded by the margin distribution, the number of training examples and the complexity of the set from which the base classifiers are chosen.

Theorem 1 (*Schapire et al. [29]*). *Let S be a sample of m examples chosen independently at random according to D. Assume that the base hypothesis space H is finite, and let $\delta>0$. Then with probability at least $1-\delta$ over the random choice of the training set S, every weighted average function f satisfies the following bound for all $\theta>0$:*

$$P_D[yf(x)\le 0]\le P_S[yf(x)\le\theta]+O(1/\sqrt{m}(\log m\log|H|/\theta^2+\log(1/\delta))^{1/2})$$

More generally, for finite or infinite H with VC-dimension d, the following bound holds as well:

$$P_D[yf(x)\le 0]\le P_S[yf(x)\le\theta]+O(1/\sqrt{m}(d\log^2(m/d)/\theta^2+\log(1/\delta))^{1/2})$$

In the theorem, H is the base classifier set, d is the VC dimension of H and θ is a threshold for the margin of an example (x,y), $P_D(yf(x)\le 0)$ denotes the probability of $yf(x)\le 0$ when an example (x,y) is chosen randomly according to the distribution D and $P_S(yf(x)\le\theta)$ denotes the probability with respect to choosing an example (x,y) uniformly at random from the training set S. This theorem states that with high probability $1-\delta$ the generalization error of any majority vote hypothesis can be bounded in terms of the number of training examples with margin below a threshold θ, the number of training examples S and the complexity measure of the base classifier set H.

Theorem 1 shows that a small generalization error for a voting classifier can be obtained by a good margin distribution on the training set. A good margin distribution refers to most training examples have large margins so that $P_S[yf(x)\le\theta]$ is small for not too small θ.

The margin proposed in [29] is based on the classification decision. In this paper, the information of classification confidence is added to the definition of margin.

Definition 2. For $x_i \in X(i=1,2,\ldots,n)$, let $\omega=\{\omega_1,\ldots,\omega_c\}$ be the set of class labels, $H=\{h_{ij}|h_{ij}\in\omega\}$ and $R=\{r_{ij}|r_{ij}\in[0,1]\}$ be the classification decision and the classification confidence of x_i by the classifier D_j $(j=1,2,\ldots,L)$, respectively. The margin of the sample x_i based on the classification confidence is denoted by

$$M_2(x_i)=S(\omega_i)-\max\{S(\omega_j)|i\neq j\} \tag{2}$$

where $S(\omega_i)$ means the sum of the classification confidence in R whose corresponding classification decision is ω_i and ω_i is the true label of x_i.

In DCE-CC, Definition 2 is used in the optimization of margin distribution.

3. The proposed algorithm: DCE-CC

Assume that a pool of L classifiers, providing both the classification decision and the classification confidence, are generated. We believe that the higher the confidence provided by the classifier, the higher the probability that the classifier has correctly classified the sample. Thus the basic idea is to classify a sample with a subset of classifiers containing $K\leq L$ classifiers whose classification confidence are the first K largest. Then the question is how to estimate the size K. In this paper, K is estimated based on the minimization of margin loss using the ordered aggregation technique. In particular, at first apply L classifiers on the training set X which contains n samples to get classification decision matrix $H=[h_{ij}](i=1,2,\ldots,n;\ j=1,2,\ldots,L)$ and the corresponding classification confidence matrix $R=[r_{ij}]$ (h_{ij} means the classification decision of the classifier D_j on the sample x_i and r_{ij} is the corresponding classification confidence for h_{ij}). Then for every row vector of R, sort its elements making the value of the new first element is the largest.... The resorted ith row of classification confidence matrix are denoted by $r'_{i1},r'_{i2},\ldots,r'_{iL}$ and its corresponding classification decision are denoted by $h'_{i1},h'_{i2},\ldots,h'_{iL}$. Furthermore, for $j=1,2,\ldots,L$, based on the first j elements in the resorted ith row $r'_{i1},\ldots,r'_{ij}$, $h'_{i1},\ldots,h'_{ij}$ and the true label $Y(i)$ of the sample x_i, the margin m_{ij} of the sample x_i is computed as Definition 2. The corresponding margin loss is computed as $l_{ij}=(1-m_{ij})^2$ and the sum of margin loss $l_{ij}(i=1,\ldots,n)$ for all the sample is denoted by $T(j)=\sum_{i=1}^{n} l_{ij}$. Finally, K is estimated with the minimum margin loss $T(K)$.

It is worth noting that since the order of classification confidence for different test samples is usually different, the selected classifiers are different. The pseudocode of DCE-CC is given in Algorithm 1.

Algorithm 1. DCE-CC.

Input:
•X: the training set which contain n samples $x_1,\ldots,x_n$.
•Y: the true labels of the training set.
•x: a test sample.
•D_j $(j=1,2,\ldots,L)$: the classifier in the ensemble which can provide both the classification decision and the classification confidence.
Output: the label of x.

1: Apply the classifiers on the training set X to get classification decision matrix $H=[h_{ij}](i=1,2,\ldots,n;\ j=1,2,\ldots,L)$ where h_{ij} means the classification decision of the classifier $D_j(j=1,2,\ldots,L)$ on the sample x_i and the corresponding classification confidence matrix $R=[r_{ij}]$.
2: For $i=1,2,\ldots,n$
3: Sort the elements of the ith row of classification confidence matrix R making the new first element is the largest.... Denote the elements of the resorted ith row of classification confidence matrix $r'_{i1},r'_{i2},\ldots,r'_{iL}$ and the corresponding ith row of classification decision matrix $h'_{i1},h'_{i2},\ldots,h'_{iL}$.
4: End for
5: For $j=1,2,\ldots,L$
6: Compute the margin m_{ij} of the sample x_i using $r'_{i1},\ldots,r'_{ij}$, $h'_{i1},\ldots,h'_{ij}$ and the true label $Y(i)$ as Definition 2 and the corresponding margin loss $l_{ij}=(1-m_{ij})^2$.
7: The sum of all the sample margin loss $l_{ij}(i=1,\ldots,n)$ is denoted by $T(j)$.
8: End for
9: Estimate K with the minimum margin loss $T(K)$.
10: Apply the classifiers on the test sample x to get classification decision vector $H_x=[h_{xj}](j=1,2,\ldots,L)$ where h_{xj} means the classification decision of the classifier $D_j(j=1,2,\ldots,L)$ on the test sample x and the corresponding classification confidence matrix $R_x=[r_{xj}]$.
11: Sort the elements of the classification decision vector H_x according to the corresponding classification confidence R_x making the new first element with the largest decision confidence.... Denote the elements of the resorted classification decision vector $H'_x=[h'_{x1},h'_{x2},\ldots,h'_{xL}]$.
12: Classify the test sample x using $h'_{x1},h'_{x2},\ldots,h'_{xK}$ with weighted voting where the weight of the h'_{xj} is its corresponding classification confidence r'_{xj}.

From the pseudocode of DCE-CC, we can see that the classification confidence is utilized as follows:

1. The reorder of base classifiers is based on the classification confidence.
2. The optimization of margin distribution is based on the classification confidence which is used to compute the margin.
3. The weighted voting to classify an unknown sample is based on the classification confidence which is used as the corresponding weight.

Thus the definition of classification confidence is crucial to the effectiveness of the algorithm and a good classification confidence for DCE-CC algorithm should have the property that the higher the confidence provided by the classifier, the higher the probability that the classifier has correctly classified the sample. Here we compute the classification confidence of the nearest-neighbor classifier and the C4.5 decision trees as follows.

Suppose X is a training set for the nearest-neighbor classifier and x is a test sample. x_1 is the nearest sample of x in X, denoted by $NH(x)$, and x_2 is the nearest sample of x in X out of the class of x_1, denoted by $NM(x)$. In this case, x will be classified into the class of x_1. So x_1 is the nearest hit and x_2 is the nearest miss of x. Then the classification margin of x is computed as $m(x)=|d(NM(x),x)-d(NH(x),x)|/2$, where d is a distance function. As we know the relationship between margin and classification confidence, in strictly, we use the margin as classification confidence of samples in this work [11].

As to decision trees, we use a set of features to match a sample when we compute its class. Suppose the subset of features $F'=\{f_1,f_2,\ldots,f_k\}$ are used. Then the classification confidence of a test sample x is computed as the minimal feature difference between x and the classification function in terms of F'. In fact, it can be understood as the distance between x and the decision function. For example, when the decision rule "$f_1<3$ and $f_2>8 \Rightarrow x\in\omega_1$" is used to classify the test sample $x=(1,9)$, then the classification confidence is computed as $\min(|1-3|, |8-9|)=1$.

If there are mixed numerical and categorical features, Heterogeneous Euclidean-Overlap Metric function can be introduced [31].

Although the focus of this paper is the fusion strategy, we still give the training phase of the base classifiers for the completeness of these experiments conducted in the next section.

It is known that in order to build a strong ensemble, the component classifiers should be with high accuracy as well as high diversity [19]. Inspired by the idea of multimodal perturbation proposed in [36], we propose the combination of double rotation and bootstrap sampling [9] to perturb the training set.

Double rotation proposed in this paper is based on the idea of PCA rotation in Rotation Forest algorithm [27]. Rotation Forest, introduced by Rodríguez and Kuncheva, is a method to generate classifier ensemble based on feature extraction. The diversity of the base classifiers is promoted by different splits of the feature set which can lead to different rotations and the accuracy is sought by keeping all principal components and also using the whole data set to train each base classifier.

Double rotation aims to enhance the diversity of the base classifiers. To construct the training sets for the base classifier D_i, we first transform the data set X linearly into the new features as PCA rotation in Rotation Forest algorithm [27] and get rotation matrix R_i^a, then resplit the feature set into G subsets, run Locality Sensitive Discriminant Analysis (LSDA) separately on each subset, reassemble a new extracted feature set while keeping all the components and get new rotation matrix S_i^a, finally XR_i^a is transformed linearly into the new features and get $XR_i^aS_i^a$. In the second rotation, we replace PCA with LSDA to transform the data. LSDA, proposed by Cai, is an effective method for feature extraction [5]. Different from PCA [16], LSDA is supervised and can find a projection which maximizes the margin between data points from different classes.

Based on double rotation and bootstrap sampling, Algorithm 2 shows the pseudocode of the training phase for the base classifiers and then L different base classifiers can be obtained.

Algorithm 2. Base classifier generation based on multi-modal perturbation.

Input:
- X: the training set which contain n samples and every sample has N features ($n\times N$ matrix).
- Y: the labels of the training data set ($n\times 1$ matrix).
- L: the number of the classifiers in the ensemble.
- G: the number of the subsets.
- F: the feature set containing N features.
- γ: the ratio of bootstrap sample in the training set.

Output: the classifier D_i.

1: For $i=1,2,\ldots,L$
2: Split F randomly into G subsets: F_{ij} ($j=1,2,\ldots,G$) and each feature subset F_{ij} contains $M=N/G$ features.
3: For $j=1,2,\ldots,G$
4: Let X_{ij} be the data set X for the features in F_{ij} (it means X_{ij} is the subset of X and only contains the features in F_{ij}).
5: Eliminate from X_{ij} a random subset of classes.
6: Select a bootstrap sample from X_{ij} of γ of the number of objects in X_{ij} and denote the new set by X'_{ij}.
7: Apply PCA on X'_{ij} to obtain the principal component coefficients $a_{i,j}^1,\ldots,a_{i,j}^{M_j}$, each of size $M\times 1$.
8: End for
9: Organize the obtained vectors with coefficients in a sparse rotation matrix R_i showed as Eq. (3).

$$R_i=\begin{bmatrix} a_{i,1}^1,\ldots,a_{i,1}^{M_1} & 0 & \cdots & 0 \\ 0 & a_{i,2}^1,\ldots,a_{i,2}^{M_2} & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & a_{i,G}^1,\ldots,a_{i,G}^{M_G} \end{bmatrix} \tag{3}$$

10: Construct R_i^a by rearrange the columns of R_i so that they correspond to the original features and denote the rearranged rotation matrix R_i^a.
11: Use XR_i^a as the new training data set, rerun the above process (from step 2 to step 10, but replace the PCA with LSDA and replace X with XR_i^a) to get the new rotation matrix S_i^a.
12: Select a bootstrap sample from $XR_i^aS_i^a$ of γ of the number of objects in $XR_i^aS_i^a$. Denote the new set by X' and the corresponding labels Y'.
13: Build classifier D_i using (X',Y') as the training sets.
14: End for

4. Experimental evaluation

In this section, we introduce the stable nearest-neighbor rule and unstable C4.5 as base classification algorithms. Some experiments on UCI data sets are performed to validate the effectiveness of the proposed DCE-CC algorithm. Table 1 describes the 20 data sets used in the study.

The DCE-CC algorithm is based on the assumption that the higher the confidence provided by the classifier, the higher the probability that the classifier has correctly classified the sample. In order to validate whether the classification confidence of the nearest-neighbor rule and the C4.5 decision trees satisfy the assumption, some experiments on UCI data sets were set up. In these experiments, the nearest-neighbor rule and the C4.5 decision trees were respectively used as the base classifier and the number of the classifiers was 100. The parameters in Algorithm 2 were given as follows: the number of the subsets G was 2 and the ratio of bootstrap sample γ was 0.75.

Figs. 1 and 2 show the relationship between the classification accuracy and the ranking of classification confidence using the nearest-neighbor rule and the C4.5 decision tree as the base classifier, respectively. In particular, the x-axis is the ranking of

Table 1
Description of 20 data sets used in this study.

Data set	Instances	Features	Classes
Australian	690	14	2
Balancescale	625	4	3
crx	690	15	2
derm	366	34	6
ecoli	336	7	8
German	1000	20	2
Glass	214	9	6
Heart	270	13	2
Horse	368	22	2
ICU	200	20	3
iono	351	34	2
iris	150	4	3
pima	768	8	2
Segmentation	2310	19	7
Soybean	683	35	19
Thyroid	215	5	3
wdbc	569	30	2
Wine	178	13	3
Wiscon	699	9	2
Yeast	1484	7	2

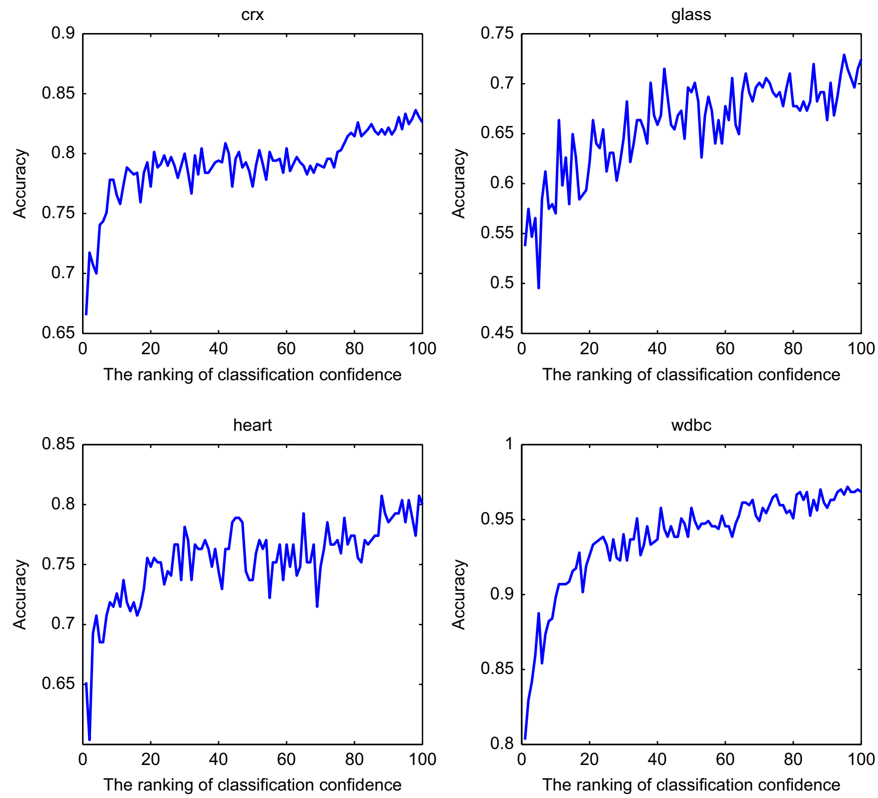

Fig. 1. Variation of classification accuracies with the ranking of the classification confidence using NN as the base classifier.

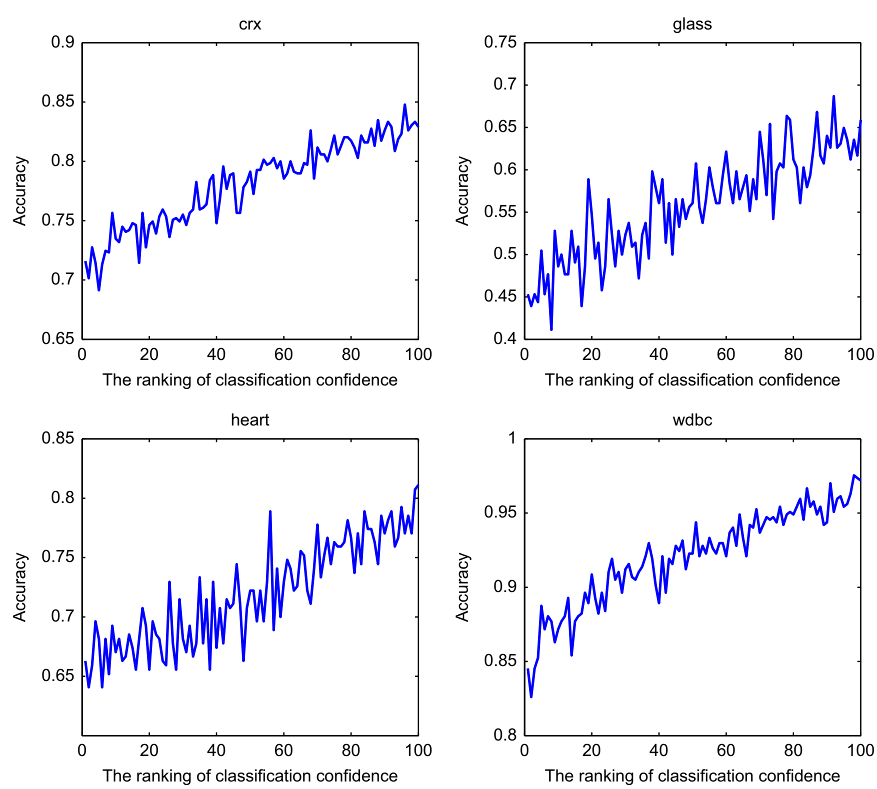

Fig. 2. Variation of classification accuracies with the ranking of the classification confidence using C4.5 as the base classifier.

classification confidence and the y-axis is the corresponding classification accuracy. On the x-axis, "1" means every test sample is classified by the classification decision with the minimum classification confidence and "100" means every test sample is classified by the classification decision with the maximal classification confidence.

From Figs. 1 and 2, we can see the trend that the higher the ranking of classification confidence, the higher the classification accuracy. It empirically demonstrates that the classification confidence of the nearest-neighbor rule and the C4.5 decision trees have the property that the higher the confidence provided by the classifier, the higher the probability that the classifier has correctly classified the sample.

In what follows, based on the nearest-neighbor rule and the C4.5 decision trees, experiments on the UCI data sets were respectively conducted to compare DCE-CC with the simple voting using all the classifiers (SV), the single classifier (the single NN and the single C4.5 decision tree), a dynamic classifier selection algorithm called DCS-LA [30] and a selective ensemble algorithm called MDM [21].

DCS-LA, proposed by Woods, is a dynamic classifier selection method based on local accuracy. In order to classify a test sample, the local accuracy of each classifier is estimated in a local region which is defined as ℓ-nearest neighbors of the test sample. Then the classifier with the highest value of this local accuracy is selected to classify the test sample.

MDM, proposed by Martínez-Muñoz, selects a suboptimal subset of classifiers from an ensemble based on the ordered aggregation technique. Given a labeled selection set X of size n, the signature vector c_j of classifier D_j is defined as the n dimensional vector whose ith component is $(c_j)_i$ whose quantity is equal to 1 if the classifier D_j correctly classifies the training sample x_i and -1 otherwise. The average ensemble signature vector $\langle c \rangle = 1/L\sum_{j=1}^{L} c_j$. The objective is to select a subensemble whose average signature vector is as close as possible to a reference position placed somewhere in the first quadrant. Usually this objective position is selected as a point o with equal components, namely,

$$o_i = p \quad \text{with } i = 1,2,\ldots,n \quad \text{and} \quad 0 < p < 1 \tag{4}$$

The first classifiers that are incorporated into the ensemble are those that reduce the distance from the vector $\langle c \rangle$ to the objective point o the most. In particular, the classifier selected in the uth iteration is the one that minimizes

$$s_u = \operatorname{argmin} d\left(o, \frac{1}{u}\left(c_i + \sum_{t=1}^{u-1} c_t\right)\right) \tag{5}$$

where i runs throughout the classifiers outside the subensemble and where $d(u,v)$ is the usual quadratic distance between vectors u and v.

Before these experiments, we divided every data set 10 fold: 8 fold was used for the training set, 1 fold for the validation set and 1 fold for the test set. In particular, the training set was used to train L classifiers in the ensemble, the validation set was used to evaluate the size of classifiers subset $K \le L$ whose classification confidence was the first largest and the test set was used to evaluate the performance of these algorithms. For each data set and fusion method, 10-fold cross-validation was performed.

In these experiments, the nearest-neighbor rule and the C4.5 decision trees were used as the base classifier and the number of the base classifiers is 100. The experimental settings in Algorithm 2 were shown as follows: the number of the subsets G was 2 and the ratio of bootstrap sample γ was 0.75. Table 2 shows classification accuracy and standard deviation of DCE-CC, SV, the single nearest-neighbor classifier, DCS-LA and MDM with the nearest-neighbor classifier. Table 3 shows classification accuracy and standard deviation of DCE-CC, SV, the single C4.5 decision tree, DCS-LA and MDM with the C4.5 decision trees. The bold one is the highest.

From Table 2, we can see that, for the stable nearest-neighbor rule, DCE-CC achieves the highest accuracy on 13 classification tasks, SV gets the highest accuracy in 1 data set, the single nearest-neighbor classifier obtains the highest accuracy in 1 data set, DCS-LA gets the highest accuracy in 3 data sets and MDM gets the highest accuracy in 2 data sets. From Table 3, we can see that, for the unstable C4.5 decision trees, DCE-CC achieves the highest accuracy on 12 classification tasks, SV gets the highest accuracy in 1 data set, the single C4.5 decision tree obtains the highest accuracy in 1 data set, DCS-LA gets the highest accuracy in 4 data sets and MDM gets the highest accuracy in 2 data sets.

Besides, Nemenyi test [24] was performed to compare DCE-CC with other methods from the statistical viewpoint and the significance level α was 0.05. In Nemenyi test, the critical difference [7] for 5 algorithms and 20 data sets at significance level $\alpha = 0.05$ is $CD = q_{0.05}\sqrt{k(k+1)/6N} = 2.728 \times \sqrt{5 \times (5+1)/(6 \times 20)} = 1.364$ where $q_{0.05}$ is the critical values for the two-tailed Nemenyi test, k is the number of algorithms and N is the number of data sets.

The average ranks for DCE-CC, SV, NN, DCS-LA and MDM in Tables 2 and 3 are respectively (1.40, 3.10, 4.65, 2.85, 3.00) and (1.60, 3.05, 4.15, 3.15, 3.05). Since the average rank differences between DCE-CC and the other methods are ($3.10-1.40 = 1.70 > 1.364$, $4.65-1.40 = 3.25 > 1.364$, $2.85-1.40 = 1.45 > 1.364$, $3.00-1.40 = 1.60 > 1.364$) and ($3.05-1.60 = 1.45 > 1.364$, $4.15-1.60 = 2.55 > 1.364$, $3.15-1.60 = 1.55 > 1.364$, $3.05-1.60 = 1.45 > 1.364$), thus

Table 2
Classification performance of NN and fusion methods with NN generated by Algorithm 2.

Data set	DCE-CC	SV	NN	DCS-LA	MDM
Australian	**84.21** ± **4.35**	80.29 ± 4.26	78.85 ± 4.69	81.74 ± 4.66	82.61 ± 2.71
Balancescale	75.18 ± 4.60	70.81 ± 7.11	70.76 ± 8.28	**78.74** ± **6.79**	72.38 ± 9.43
crx	**82.59** ± **14.96**	80.72 ± 13.78	78.98 ± 11.72	81.16 ± 11.46	80.87 ± 16.51
derm	96.63 ± 1.78	96.56 ± 2.00	96.35 ± 2.17	**96.79** ± **2.96**	96.61 ± 1.21
ecoli	**85.25** ± **2.53**	82.16 ± 5.49	79.56 ± 6.23	82.29 ± 6.82	81.18 ± 1.61
German	73.00 ± 4.52	70.50 ± 3.63	68.10 ± 3.87	70.20 ± 3.99	**73.69** ± **2.55**
Glass	**72.48** ± **15.60**	70.05 ± 15.23	65.77 ± 9.55	70.60 ± 11.22	69.73 ± 9.85
Heart	**80.00** ± **6.10**	77.04 ± 5.47	75.19 ± 9.88	78.89 ± 6.99	79.26 ± 4.22
Horse	**92.31** ± **4.86**	91.87 ± 4.65	89.70 ± 4.97	90.21 ± 6.11	91.89 ± 4.19
ICU	91.50 ± 8.92	89.97 ± 8.55	84.19 ± 17.78	**92.08** ± **3.39**	90.48 ± 5.16
iono	86.62 ± 6.32	**87.01** ± **6.33**	86.37 ± 4.62	86.71 ± 5.39	85.89 ± 4.97
iris	**96.53** ± **3.32**	96.00 ± 3.44	95.33 ± 4.06	95.67 ± 3.58	95.69 ± 5.06
pima	**74.34** ± **5.44**	72.39 ± 4.16	69.53 ± 3.78	71.87 ± 4.45	71.95 ± 4.74
Segmentation	**97.49** ± **1.23**	97.19 ± 1.60	96.67 ± 1.89	95.06 ± 1.55	96.12 ± 2.00
Soybean	**91.49** ± **4.60**	90.93 ± 4.59	90.77 ± 4.67	90.82 ± 3.95	90.96 ± 2.23
Thyroid	96.39 ± 3.58	96.23 ± 4.37	95.80 ± 4.16	95.85 ± 1.70	**98.18** ± **1.21**
wdbc	**96.85** ± **2.29**	96.67 ± 1.92	95.09 ± 3.05	96.14 ± 1.99	96.64 ± 3.00
Wine	**95.75** ± **3.04**	94.31 ± 4.42	93.86 ± 6.07	94.62 ± 4.58	94.00 ± 5.94
Wiscon	**97.28** ± **2.97**	95.86 ± 4.18	95.00 ± 3.70	96.00 ± 3.14	95.29 ± 4.40
Yeast	67.87 ± 4.52	66.39 ± 4.64	**70.36** ± **5.97**	67.53 ± 4.88	66.44 ± 3.52

Table 3
Classification performance of C4.5 and fusion methods with C4.5 generated by Algorithm 2.

Data set	DCE-CC	SV	C4.5	DCS-LA	MDM
Australian	**86.92 ± 4.47**	85.08 ± 4.94	79.85 ± 4.33	82.76 ± 1.88	86.67 ± 6.67
Balancescale	81.59 ± 10.26	78.88 ± 6.31	78.46 ± 9.91	**82.44 ± 8.66**	78.39 ± 3.98
crx	**83.90 ± 17.69**	83.75 ± 18.41	76.98 ± 5.72	79.69 ± 14.53	82.61 ± 20.13
derm	**97.16 ± 2.81**	94.36 ± 2.65	88.81 ± 5.74	96.79 ± 2.96	94.44 ± 3.26
ecoli	**85.54 ± 3.86**	84.41 ± 4.29	81.16 ± 6.19	84.63 ± 4.85	81.12 ± 3.36
German	**75.20 ± 2.49**	74.60 ± 3.37	69.20 ± 4.98	74.70 ± 4.11	75.16 ± 9.20
Glass	**67.37 ± 14.42**	66.38 ± 12.58	61.24 ± 6.85	63.79 ± 16.23	66.45 ± 8.86
Heart	**82.22 ± 7.16**	81.85 ± 7.08	69.26 ± 10.48	75.93 ± 8.60	82.00 ± 6.73
Horse	91.61 ± 4.91	91.30 ± 4.73	**93.22 ± 4.61**	91.57 ± 4.12	91.41 ± 3.16
ICU	92.03 ± 9.28	92.03 ± 7.44	78.77 ± 13.67	89.97 ± 5.28	**92.43 ± 2.13**
iono	**92.38 ± 7.07**	92.09 ± 4.86	77.67 ± 9.66	87.08 ± 7.26	92.21 ± 5.05
iris	97.33 ± 2.13	96.13 ± 6.32	95.33 ± 5.49	**97.63 ± 2.32**	95.26 ± 5.26
pima	**76.83 ± 5.18**	76.70 ± 5.03	69.53 ± 5.00	71.23 ± 4.64	76.79 ± 3.10
Segmentation	94.11 ± 1.96	91.26 ± 2.47	92.60 ± 1.85	**96.10 ± 1.71**	90.31 ± 6.80
Soybean	90.58 ± 5.75	**92.23 ± 3.96**	90.04 ± 7.44	89.60 ± 9.43	88.53 ± 2.63
Thyroid	**97.13 ± 1.96**	96.23 ± 3.61	93.96 ± 5.77	93.90 ± 5.03	96.36 ± 1.80
wdbc	**97.90 ± 2.15**	97.72 ± 2.20	92.80 ± 3.93	94.04 ± 2.02	97.54 ± 2.00
Wine	**96.92 ± 2.63**	96.60 ± 3.93	93.26 ± 4.37	93.21 ± 4.78	96.69 ± 2.19
Wiscon	96.42 ± 3.10	96.42 ± 3.03	91.42 ± 4.36	**97.00 ± 2.31**	91.35 ± 3.10
Yeast	71.23 ± 3.73	71.36 ± 3.96	73.58 ± 4.59	72.01 ± 3.54	**73.96 ± 1.45**

Table 4
Classification performance using different NN selection and voting strategies.

Data set	SV	SDCE-CC	WV	DCE-CC
Australian	80.29 ± 4.26	84.09 ± 3.61	83.63 ± 3.94	84.21 ± 4.35
Balancescale	70.81 ± 7.11	72.15 ± 5.92	70.73 ± 7.01	75.18 ± 4.60
crx	80.72 ± 13.78	81.89 ± 14.96	82.19 ± 13.90	82.59 ± 14.96
derm	96.56 ± 2.00	96.59 ± 3.16	96.59 ± 4.67	96.63 ± 1.78
ecoli	82.16 ± 5.49	84.16 ± 3.09	84.14 ± 2.00	85.25 ± 2.53
German	70.50 ± 3.63	72.61 ± 4.52	71.60 ± 3.44	73.00 ± 4.52
Glass	70.05 ± 15.23	72.48 ± 1.56	72.44 ± 14.68	72.48 ± 15.60
Heart	77.04 ± 5.47	80.00 ± 6.10	78.89 ± 7.21	80.00 ± 6.10
Horse	91.87 ± 4.65	91.26 ± 5.19	91.60 ± 4.32	92.31 ± 4.86
ICU	89.97 ± 8.55	89.97 ± 8.55	91.50 ± 8.92	91.50 ± 8.92
iono	87.01 ± 6.33	86.62 ± 6.32	86.54 ± 6.97	86.62 ± 6.32
iris	96.00 ± 3.44	96.37 ± 3.48	95.98 ± 3.16	96.53 ± 3.32
pima	72.39 ± 4.16	73.08 ± 5.14	71.87 ± 4.82	74.34 ± 5.44
Segmentation	97.19 ± 1.60	97.03 ± 1.51	97.01 ± 1.48	97.49 ± 1.23
Soybean	90.93 ± 4.59	91.16 ± 5.36	91.07 ± 4.37	91.49 ± 4.60
Thyroid	96.23 ± 4.37	93.94 ± 6.31	95.30 ± 4.50	96.39 ± 3.58
wdbc	96.67 ± 1.92	96.85 ± 2.29	96.02 ± 1.84	96.85 ± 2.29
Wine	94.31 ± 4.42	95.06 ± 4.16	95.12 ± 5.42	95.75 ± 3.04
Wiscon	95.86 ± 4.18	97.28 ± 2.97	96.57 ± 3.24	97.28 ± 2.97
Yeast	66.39 ± 4.64	67.54 ± 4.12	67.67 ± 3.90	67.87 ± 4.52

Table 5
Classification performance using different C4.5 selection and voting strategies.

Data set	SV	SDCE-CC	WV	DCE-CC
Australian	85.08 ± 4.94	84.96 ± 4.50	85.36 ± 4.34	86.92 ± 4.47
Balancescale	78.88 ± 6.31	76.26 ± 9.81	82.60 ± 8.07	81.59 ± 10.26
crx	83.75 ± 18.41	83.89 ± 18.08	83.90 ± 17.76	83.90 ± 17.69
derm	94.36 ± 2.65	95.19 ± 3.68	97.34 ± 2.35	97.16 ± 2.81
ecoli	84.41 ± 4.29	83.62 ± 5.31	85.57 ± 4.51	85.54 ± 3.86
German	74.60 ± 3.37	72.70 ± 3.56	75.20 ± 2.57	75.20 ± 2.49
Glass	66.38 ± 12.58	64.49 ± 14.73	66.78 ± 14.66	67.37 ± 14.42
Heart	81.85 ± 7.08	81.11 ± 6.86	81.85 ± 5.64	82.22 ± 7.16
Horse	91.30 ± 4.73	91.36 ± 5.14	91.31 ± 4.35	91.61 ± 4.91
ICU	92.03 ± 7.44	89.45 ± 10.41	90.98 ± 12.35	92.03 ± 9.28
iono	92.09 ± 4.86	92.29 ± 6.06	92.19 ± 4.61	92.38 ± 7.07
iris	96.13 ± 6.32	97.33 ± 4.66	95.33 ± 6.32	97.33 ± 2.13
pima	76.70 ± 5.03	74.88 ± 4.83	76.31 ± 5.43	76.83 ± 5.18
Segmentation	91.26 ± 2.47	93.64 ± 1.94	96.21 ± 1.30	94.11 ± 1.96
Soybean	92.23 ± 3.96	88.96 ± 5.16	91.93 ± 4.52	90.58 ± 5.75
Thyroid	96.23 ± 3.61	96.35 ± 3.17	96.41 ± 2.36	97.13 ± 1.96
wdbc	97.72 ± 2.20	97.89 ± 1.99	97.71 ± 2.10	97.90 ± 2.15
Wine	96.60 ± 3.93	95.42 ± 4.19	97.15 ± 3.01	96.92 ± 2.63
Wiscon	96.42 ± 3.03	95.71 ± 4.20	96.28 ± 2.95	96.42 ± 3.10
Yeast	71.36 ± 3.96	71.23 ± 3.00	71.16 ± 3.66	71.23 ± 3.73

DCE-CC performs significantly better than SV, NN, DCS-LA and MDM. These experiments validate the effectiveness of the proposed fusion algorithm DCE-CC.

In what follows, we analyze why DCE-CC can improve the fusion performance. DCE-CC algorithm mainly contains two parts: the dynamic classifier ensemble for different test sample and the weighted voting based on the classification confidence. First, experiments were given to test whether they were all necessary for improving the fusion performance. The experiments results include:

1. The fusion results of the simple voting using all the classifiers (for short SV);
2. The fusion results of the simple voting based on the dynamic classifier ensemble (for short SDCE-CC);
3. The fusion result of the weighted voting using all the classifiers (for short WV);
4. The fusion result of the weighted voting based on the dynamic classifier ensemble (for short DCE-CC);

Tables 4 and 5 show the experiment results using the nearest-neighbor rule and the C4.5 decision trees, respectively.

From the experiment results, we can see that the dynamic ensemble and the weighted voting are all necessary for boosting the classification accuracy. For example, for the crx data set in Table 4, using SDCE-CC and WV can both improve the classification accuracy and using DCE-CC, the accuracy is the highest.

By Theorem 1, we know that if the fraction of training examples with small margin is small, then the generalization ability of the voting classifier can be improved. Now we analyze why, compared with the simple voting using all the classifiers, DCE-CC can boost the classification accuracy from the view of the margin distribution, where the margin of a sample is defined as Definition 1. Figs. 3 and 4 show the margin distribution on the validation set using the nearest-neighbor classifier and the C4.5 decision trees, respectively. On Figs. 3 and 4, the x-axis is the margin and the y-axis is the fraction of examples whose margin is at most $x \in [-1,1]$. The margin distribution of DCE-CC and the simple voting using all the classifiers are indicated by solid blue curves and short-dashed red curves, respectively.

From Figs. 3 and 4, we can see that compared with SV, DCE-CC improves the margin distribution on most data sets.

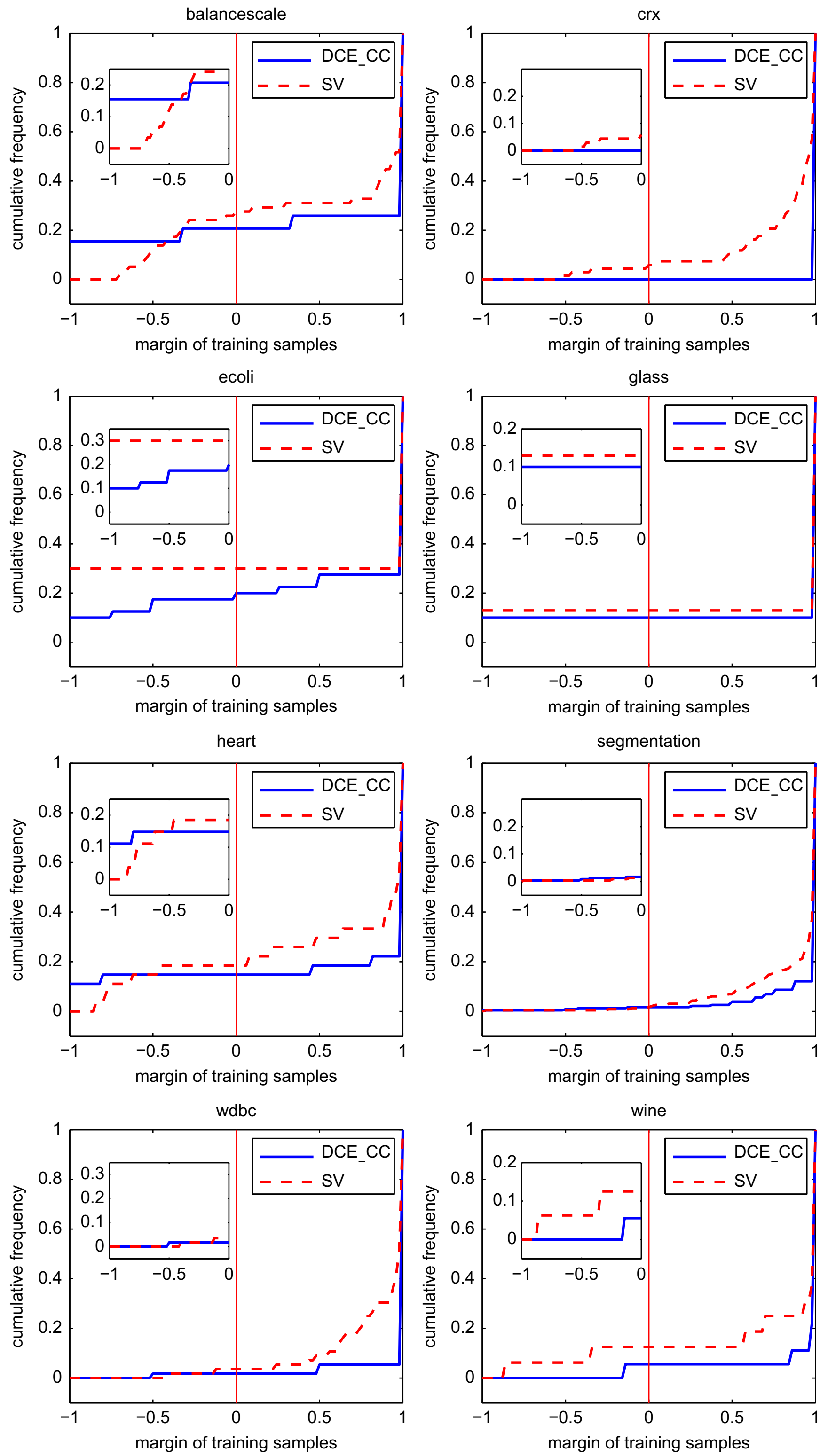

Fig. 3. Margin cumulative frequency of training samples using NN as the base classifier.

These above experiments were conducted with the base classifiers generated by Algorithm 2. Then can DCE-CC still perform well with the base classifiers generated by other strategies? In other word, whether the performance of DCE-CC depends upon a special diversity strategy? In order to answer these question, the experiments with the base classifiers generated by Random Feature Selection (RFS) were conducted. In these experiments, the nearest-neighbor rule and the C4.5 decision trees were still respectively used as the base classifier and the number of the classifiers was 100. The ratio of sampling from original feature set was 0.75. Tables 6 and 7 show classification accuracy and standard deviation of DCE-CC, SV, the single classifier, DCS-LA and MDM with the base classifiers generated by Random Feature Selection. Nemenyi test was also performed to compare DCE-CC

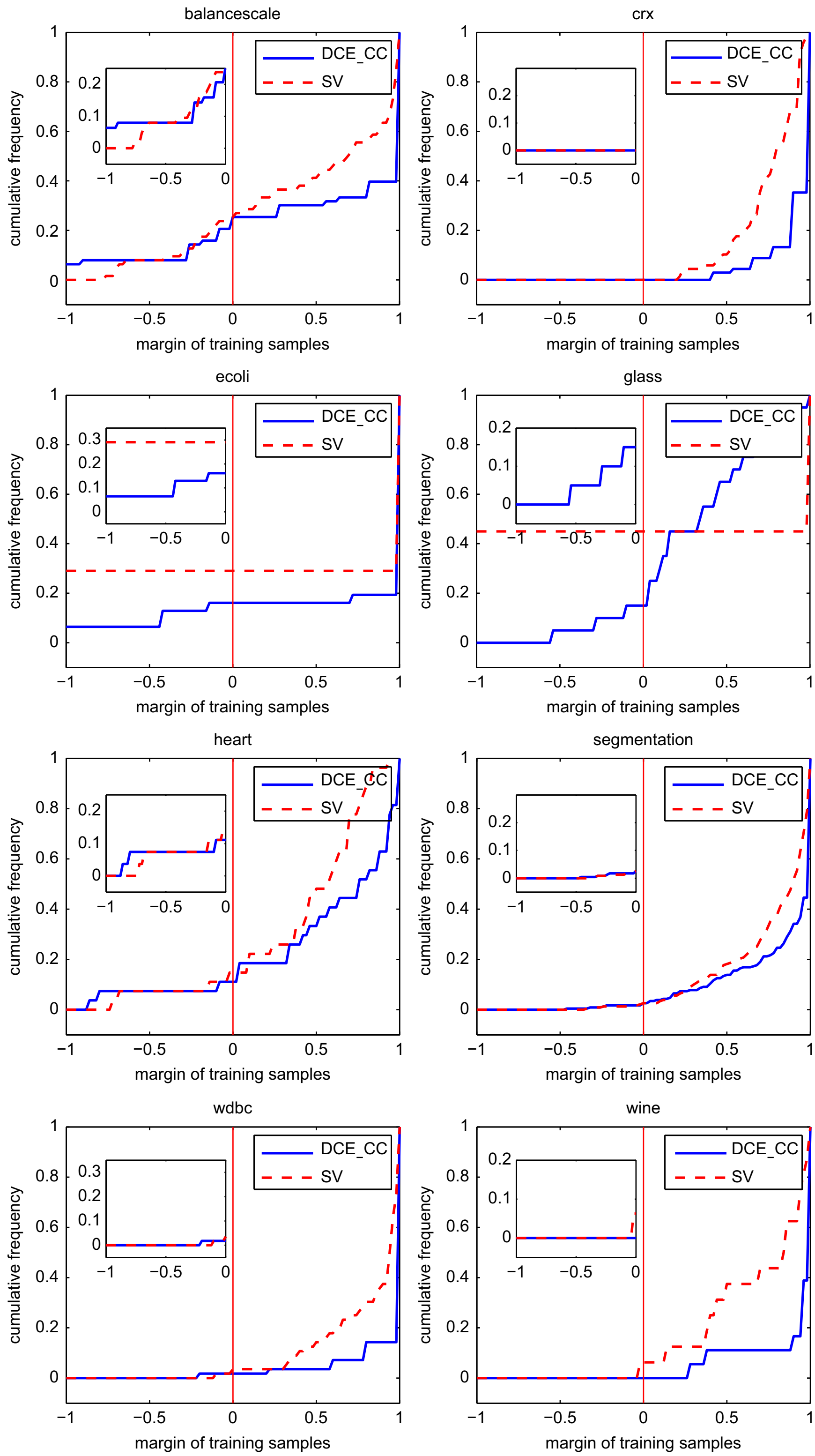

Fig. 4. Margin cumulative frequency of training samples using C4.5 as the base classifier.

with other methods from the statistical viewpoint and the significance level was still 0.05. The average ranks for DCE-CC, SV, NN, DCS-LA and MDM in Tables 6 and 7 are respectively (1.45, 3.15, 4.30, 3.05, 3.05) and (1.50, 2.95, 4.50, 2.95, 3.10). Since the average rank differences between DCE-CC and the other methods are $(3.15-1.45=1.70>1.364,\ 4.30-1.45=2.85>1.364,\ 3.05-1.45=1.60>1.364,\ 3.05-1.45=1.60>1.364)$ and $(2.95-1.50=1.45>1.364,\ 4.50-1.50=3.00>1.364,\ 2.95-1.50=1.45>1.364,\ 3.10-1.50=1.60>1.364)$, thus DCE-CC performs significantly better than SV, NN, DCS-LA and MDM.

The experimental results showed in Tables 6 and 7 validate that the performance of DCE-CC does not depend upon a special diversity strategy and it can also performs well for other diversity strategies which can make these base classifiers be diverse.

Table 6
Classification performance of NN and fusion methods with NN generated by RFS.

Data set	DCE-CC	SV	NN	DCS-LA	MDM
Australian	80.87 ± 4.28	$\mathbf{82.02 \pm 2.99}$	78.85 ± 4.69	81.76 ± 3.64	79.71 ± 3.69
Balancescale	66.70 ± 9.00	65.95 ± 9.98	$\mathbf{70.76 \pm 8.28}$	63.83 ± 15.09	66.46 ± 10.95
crx	81.06 ± 13.89	80.43 ± 13.70	78.98 ± 11.72	80.86 ± 10.33	$\mathbf{81.16 \pm 14.20}$
derm	$\mathbf{97.92 \pm 1.70}$	96.95 ± 3.00	96.35 ± 2.17	93.37 ± 5.94	97.22 ± 3.93
ecoli	$\mathbf{81.95 \pm 4.84}$	80.80 ± 4.64	79.56 ± 6.23	80.77 ± 5.18	81.76 ± 3.86
German	73.51 ± 4.02	73.20 ± 3.66	68.10 ± 3.87	73.36 ± 4.22	$\mathbf{75.00 \pm 2.00}$
Glass	66.81 ± 8.53	66.72 ± 9.21	65.77 ± 9.55	$\mathbf{69.69 \pm 11.29}$	63.64 ± 5.57
Heart	$\mathbf{79.69 \pm 6.31}$	78.89 ± 7.82	75.19 ± 9.88	78.96 ± 5.64	79.01 ± 7.12
Horse	$\mathbf{91.06 \pm 5.19}$	90.79 ± 4.73	89.70 ± 4.97	88.32 ± 5.09	89.19 ± 5.16
ICU	$\mathbf{92.61 \pm 5.89}$	91.55 ± 5.61	84.19 ± 17.78	91.69 ± 5.95	91.96 ± 5.01
iono	$\mathbf{88.31 \pm 5.50}$	87.55 ± 5.48	86.37 ± 4.62	87.97 ± 4.76	87.61 ± 6.33
iris	$\mathbf{96.16 \pm 3.39}$	95.68 ± 3.89	95.33 ± 4.06	95.98 ± 3.49	95.81 ± 5.58
pima	$\mathbf{72.61 \pm 6.91}$	70.96 ± 4.70	69.53 ± 3.78	71.05 ± 3.02	71.16 ± 4.16
Segmentation	97.06 ± 1.50	97.01 ± 1.42	96.67 ± 1.89	$\mathbf{97.16 \pm 1.37}$	95.70 ± 1.63
Soybean	$\mathbf{92.86 \pm 3.73}$	92.12 ± 4.60	90.77 ± 4.67	92.21 ± 6.23	89.12 ± 3.22
Thyroid	$\mathbf{96.32 \pm 5.50}$	94.85 ± 5.25	95.80 ± 4.16	94.37 ± 6.64	95.45 ± 3.21
wdbc	$\mathbf{96.82 \pm 2.30}$	96.32 ± 2.39	95.09 ± 3.05	96.31 ± 1.93	96.38 ± 1.89
Wine	95.06 ± 3.16	$\mathbf{95.42 \pm 4.63}$	93.86 ± 6.07	95.01 ± 6.03	94.44 ± 3.93
Wiscon	$\mathbf{96.91 \pm 2.31}$	96.43 ± 3.82	95.00 ± 3.70	96.61 ± 3.14	96.57 ± 4.36
Yeast	72.31 ± 4.57	$\mathbf{72.86 \pm 4.62}$	70.36 ± 5.97	71.03 ± 5.02	70.34 ± 2.78

Table 7
Classification performance of C4.5 and fusion methods with C4.5 generated by RFS.

Data set	DCE-CC	SV	C4.5	DCS-LA	MDM
Australian	81.79 ± 3.56	81.74 ± 3.13	79.85 ± 4.33	80.03 ± 4.29	$\mathbf{84.35 \pm 5.06}$
Balancescale	80.21 ± 6.93	79.18 ± 8.97	78.46 ± 9.91	$\mathbf{81.53 \pm 5.07}$	77.30 ± 2.88
crx	77.49 ± 9.28	$\mathbf{79.87 \pm 8.16}$	76.98 ± 5.72	77.08 ± 8.10	78.26 ± 14.38
derm	91.56 ± 6.35	90.23 ± 5.61	88.81 ± 5.74	$\mathbf{93.13 \pm 4.02}$	90.35 ± 7.19
ecoli	$\mathbf{83.69 \pm 5.26}$	82.73 ± 5.55	81.16 ± 6.19	82.28 ± 5.20	82.94 ± 7.32
German	$\mathbf{71.60 \pm 4.72}$	69.40 ± 5.06	69.20 ± 4.98	70.20 ± 4.32	69.00 ± 2.35
Glass	$\mathbf{68.63 \pm 10.75}$	67.26 ± 7.62	61.24 ± 6.85	65.81 ± 15.79	63.64 ± 3.80
Heart	67.04 ± 8.81	65.19 ± 9.75	$\mathbf{69.26 \pm 10.48}$	66.67 ± 6.98	63.70 ± 4.83
Horse	95.89 ± 3.24	$\mathbf{96.21 \pm 4.77}$	93.22 ± 4.61	94.86 ± 4.09	95.68 ± 2.42
ICU	91.51 ± 6.37	88.35 ± 15.18	78.77 ± 13.67	$\mathbf{91.60 \pm 3.05}$	91.43 ± 2.13
iono	$\mathbf{80.56 \pm 6.20}$	77.96 ± 9.17	77.67 ± 9.66	79.36 ± 10.02	76.67 ± 10.13
iris	95.61 ± 4.22	94.67 ± 5.26	95.33 ± 5.49	95.29 ± 3.19	$\mathbf{96.00 \pm 3.65}$
pima	$\mathbf{73.39 \pm 4.90}$	72.92 ± 4.64	69.53 ± 5.00	71.08 ± 5.14	70.13 ± 3.67
Segmentation	$\mathbf{96.33 \pm 1.38}$	94.50 ± 1.41	92.60 ± 1.85	94.16 ± 1.62	92.80 ± 1.77
Soybean	$\mathbf{94.61 \pm 3.05}$	93.01 ± 6.05	90.04 ± 7.44	91.21 ± 5.66	91.47 ± 3.19
Thyroid	94.56 ± 3.16	94.42 ± 3.27	93.96 ± 5.77	93.05 ± 6.21	$\mathbf{95.45 \pm 3.62}$
wdbc	$\mathbf{95.44 \pm 2.37}$	94.91 ± 3.56	92.80 ± 3.93	94.56 ± 3.15	94.04 ± 3.64
Wine	$\mathbf{96.97 \pm 2.06}$	94.93 ± 4.88	93.26 ± 4.37	95.42 ± 4.63	96.67 ± 2.97
Wiscon	$\mathbf{95.69 \pm 3.98}$	93.99 ± 3.48	91.42 ± 4.36	94.00 ± 3.48	94.86 ± 3.29
Yeast	$\mathbf{76.91 \pm 3.42}$	74.33 ± 5.21	73.58 ± 4.59	76.02 ± 5.47	75.57 ± 2.71

5. Conclusions and future work

Effective fusion strategy in ensemble learning has attracted much attention in recent years. In this paper, a dynamic classifier ensemble algorithm DCE-CC is proposed and some explicit experiments show the effectiveness of this algorithm. We systematically discuss the rationality of DCE-CC algorithm and explore the reason of improving classification performance. The following conclusions can be drawn.

1. It is shown that the classifier with large classification confidence can provide good generalization performance. Thus, in dynamic ensembles we select the classifiers with large classification confidence.

2. It is also shown that good performance can be achieved by dynamically combining a subset of classifiers with weighted voting. That is to say, we dynamically select a subset of base classifiers according to the classification confidence, and then combine them with weighted voting. Good generalization performance is obtained.

3. We explain the success of the proposed algorithm with the margin distribution. It is found that the margin distribution is improved after dynamic ensemble.

In this work, we just consider the base classifiers trained with the NN and the C4.5. In fact, we think this idea is also suitable for the SVM and other learning algorithms if we define the good index of classification confidence. We will extend this idea to other learning algorithms in future.

Acknowledgments

This work is supported by National Natural Science Foundation of China under Grant 61222210, 61170107, 60873140, 61073125 and 61071179, the Program for New Century Excellent Talents in University (no. NCET-08-0155 and NCET-08-0156), and the Fok Ying Tong Education Foundation (no. 122035).

References

[1] M. Aksela, J. Laaksonen, Using diversity of errors for selecting members of a committee classifier, Pattern Recognition 39 (2006) 608–623.
[2] L. Breiman, Random forests, Ann. Stat. 45 (1) (2001) 5–32.

[3] P.L. Bartlett, For valid generalization, the size of the weights is more important than the size of the network, in: Advances in Neural Information Processing Systems, vol. 9, 1997.

[4] J.A. Benediktsson, J.R. Sveinsson, O.K. Ersoy, P.H. Swain, Parallel consensual neural networks, IEEE Trans. Neural Networks 8 (1) (1997) 54–64.

[5] D. Cai, X.F. He, K. Zhou, J.W. Han, H.J. Bao, Locality sensitive discriminant analysis, in: International Joint Conference on Artificial Intelligence, 2007, pp. 708–713.

[6] H. Chen, P. Tino, X. Yao, Predictive ensemble pruning by expectation propagation, IEEE Trans. Knowl. Data Eng. 21 (7) (2009) 999–1013.

[7] J. Demsar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (1) (2006) 1–30.

[8] T.G. Dietterich, Ensemble methods in marching learning, in: Proceedings of the First International Workshop on Multiple Classifier Systems, MCS, Lecture Notes in Computer Science, vol. 1857, Springer Publication, Berlin, 2000.

[9] B. Efron, R. Tibshirani, An Introduction to the Bootstrap, Chapman and Hall, New York, 1993.

[10] G. Fumera, F. Roli, A theoretical and experimental analysis of linear combiners for multiple classifier systems, IEEE Trans. Pattern Anal. Mach. Intell. 27 (6) (2005) 942–956.

[11] R. Gilad-Bachrach, A. Navot, N. Tishby, Margin based feature selection-theory and algorithms, in: Proceedings of the 21st International Conference on Machine Learning, ACM, 2004, pp. 43–50.

[12] G. Giacinto, F. Roli, Dynamic classifier selection based on multiple classifier behaviour, Pattern Recognition 34 (2001) 1879–1881.

[13] G. Giacinto, F. Roli, G. Fumera, Selection of image classifiers, Electron. Lett. 36 (5) (2000) 420–422.

[14] G. Giacinto, F. Roli, A theoretical framework for dynamic classifier selection, in: Proceedings of the 15th International Conference on Pattern Recognition, IEEE Computer Society Press, 2000, pp. 8–11.

[15] D. Hernández-Lobato, J.M. Hernández-Lobato, R. Ruiz-Torrubiano, Á Valle, Pruning adaptive boosting ensembles by means of a genetic algorithm, in: E. Corchado, H. Yin, V.J. Botti, C. Fyfe (Eds.), Proceedings of the Seventh International Conference on Intelligent Data Engineering and Automated Learning, 2006, pp. 322–329.

[16] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2001.

[17] Q.H. Hu, D.R. Yu, Z.X. Xie, X.D. Li, EROS: ensemble rough subspaces, Pattern Recognition 40 (2007) 3728–3739.

[18] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas, On combining classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 20 (3) (1998) 226–239.

[19] A. Krogh, J. Vedelsby, Neural network ensembles, cross validation, and active learning, in: G. Tesauro, D.S. Touretzky, T.K. Leen (Eds.), Advances in Neural Information Processing Systems, vol. 7, MIT Press, Cambridge, MA, 1995, pp. 231–238.

[20] G. Martínez-Muñoz, D. Hernandez-Lobato, A. Suarez, An analysis of ensemble pruning techniques based on ordered aggregation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 245–259.

[21] G. Martínez-Muñoz, A. Suárez, Aggregation ordering in bagging, in: Proceedings of the International Conference on Artificial Intelligence and Applications, 2004, pp. 258–263.

[22] G. Martínez-Muñoz, A. Suárez, Pruning in ordered bagging ensembles, in: Proceedings of the 23th International Conference on Machine Learning, 2006, pp. 609–616.

[23] G. Martínez-Muñoz, A. Suárez, Using boosting to prune bagging ensembles, Pattern Recognition Lett. 28 (1) (2007) 156–165.

[24] P.B. Nemenyi, Distribution-Free Multiple Comparisons. Ph.D. Thesis, Princeton University, 1963.

[25] P. Piro, R. Nock, F. Nielsen, M. Barlaud, Leveraging k-NN for generic classification boosting, Neurocomputing 80 (2012) 3–9.

[26] Z.Q. Qi, Y.T. Xu, L.S. Wanga, Y. Song, Online multiple instance boosting for object detection, Neurocomputing 74 (2011) 1769–1775.

[27] J.J. Rodríguez, L.I. Kuncheva, Rotation forest: a new classifier ensemble method, IEEE Trans. Pattern Anal. Mach. Intell. 28 (10) (2006) 1619–1630.

[28] J. Shawe-Taylor, P.L. Bartlett, R.C. Williamson, M. Anthony, A framework for structural risk minimisation, in: Proceedings of the Ninth Annual Conference on Computational Learning Theory, 1996, pp. 68–76.

[29] R.E. Schapire, Y. Freund, P. Bartlett, W.S. Lee, Boosting the margin: a new explanation for the effectiveness of voting methods, in: Machine Learning: Proceedings of the 14th International Conference, 1997.

[30] K. Woods, W.P. Kegelmeyer, K. Bowyer, Combination of multiple classifiers using local accuracy estimates, IEEE Trans. Pattern Anal. Mach. Intell. 19 (4) (1997) 405–410.

[31] D.R. Wilson, T.R. Martinez, Improved heterogeneous distance functions, J. Artif. Intell. Res. 6 (1997) 1–34.

[32] L.W. Wang, M. Sugiyama, C. Yang, Z.H. Zhou, J.F. Feng, On the margin explanation of boosting algorithms, in: Proceedings of COLT, 2008, pp. 479–490.

[33] Y. Zhang, S. Burer, W.N. Street, Ensemble pruning via semi-definite programming, J. Mach. Learn. Res. 7 (2006) 1315–1338.

[34] C.S. Zhang, Q.T. Cai, Y.Q. Song, Boosting with pairwise constraints, Neurocomputing 73 (2010) 908–919.

[35] Z.H. Zhou, J.X. Wu, W. Tang, Ensembling neural networks: many could be better than all, Artif. Intell. 137 (1–2) (2002) 239–263.

[36] Z.H. Zhou, Y. Yu, Ensembling local learners through multimodal perturbation, IEEE Trans. Syst. Man Cybern. B Cybern. 35 (4) (2005) 725–735.

[37] L. Zhang, W.D. Zhou, Sparse ensembles using weighted combination methods based on linear programming, Pattern Recognition 44 (2011) 97–106.

Leijun Li got his B.Sc., M.Sc. from Hebei Normal University in 2007 and 2010, respectively. Now he is a Ph.D. candidate with School of Computer Science and Technology, Harbin Institute of Technology. His research interests include ensemble learning, margin theory and rough sets, etc.

Bo Zou got his B.Sc., M.E. and Ph.D. degrees from Harbin Institute of Technology, Harbin, China in 2001, 2005 and 2009, respectively. He was a postdoctoral fellow with School of Economics and Management from 2009 to 2011. Now he is an associate professor with this school. His main interests are knowledge management, knowledge discovery and data mining.

Qinghua Hu received B.Sc., M.E. and Ph.D. degrees from Harbin Institute of Technology, Harbin, China in 1999, 2002 and 2008, respectively. He started working with Harbin Institute of Technology from 2006, and was a postdoctoral fellow with the Hong Kong Polytechnic University from 2009 to 2011. Now he is a full professor with Tianjin University. His research interests are focused on intelligent modeling, data mining, knowledge discovery for classification and regression. He is a PC co-chair of RSCTC 2010 and severs as referee for a great number of journals and conferences. He has published more than 90 journal and conference papers in the areas of pattern recognition and fault diagnosis.

Xianqian Wu received his B.Sc., M.E. and Ph.D. degrees from Harbin Institute of Technology, Harbin, China in 1997, 1999 and 2004, respectively. Now he is a full professor with School of Computer Science and Technology, Harbin Institute of Technology. He once visited The Hong Kong Polytechnic University and Michigan State University. His main interests are focused on biometrics, image processing and pattern recognition. He has published more than 50 peer revived papers in these domains.

Daren Yu received the M.Sc. and D.Sc. degrees from Harbin Institute of Technology, Harbin, China, in 1988 and 1996, respectively. Since 1988, he has been working at the School of Energy Science and Engineering, Harbin Institute of Technology. His main research interests are in modeling, simulation, and control of power systems. He has published more than one hundred conference and journal papers on power control and fault diagnosis.