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

A R T I C L E   I N F O

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

A B S T R A C T

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 viewpoint of margin distribution.

© 2012 Elsevier B.V. All rights reserved.

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
dynamic classifier selection based on multiple classifier behavior [12], and so on. In [30], in order to classify an unknown test sample, the $\ell$-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 $\ell$ nearest neighbors, its performance is affected by the choice of $\ell$.

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 minimizing 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 $D_1,\ldots,D_h$ 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 $R_i=[r_{1i},\ldots,r_{ni}]\in[0,1]$. 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 $x_i\in\{x_1,\ldots,x_n\}$, let $\omega=[\omega_1,\ldots,\omega_h]$ be the set of class labels, $H=[h_j|j\in\omega]$ be the classification decision of $x_i$ by the classifier $D_j$ ($j=1,2,\ldots,h$). The margin of the sample $x_i$ is denoted by

$$M_i(x_i) = \frac{N(o_i) - \max[N(o_j)|j \neq i]}{L}$$

where $L$ is the number of the classifiers, $N(o_i)$ means the number of $o_i$ in $H$ and $o_i$ 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 $M_i(x_i) > 0$. A large positive margin can be interpreted as a “confident” correct classification, so the larger the margin on the test samples, the better the classifier 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 $\theta > 0$:

$$P_D[|f(x) - \hat{f}(x)| \leq \theta] \leq O(1/\sqrt{m}\log m \log |H|/\theta^2 + \log(1/\delta)^{1/2})$$

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

$$P_D[|f(x) - \hat{f}(x)| \leq \theta] \leq O(1/\sqrt{m}(d \log^2(m/d))/\theta^2 + \log(1/\delta)^{1/2})$$

In the theorem, $H$ is the base classifier set, $d$ is the VC dimension of $H$ and $\theta$ is a threshold for the margin of an example $(x,y)$, $P_D[f(x) \leq \theta]$ denotes the probability of $y=f(x) \leq 0$ when an example $(x,y)$ is chosen randomly according to the distribution $D$ and $P_D[f(x) \leq \theta]$ 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 $\theta$, 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 $P_D[f(x) \leq \theta]$ is small for not too small $\theta$. 

582 L. Li et al. / Neurocomputing 99 (2013) 581–591
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 \( \omega = \{\omega_1, \ldots, \omega_k\} \) be the set of class labels, \( H = \{h_{j|i}\} \) and \( R = \{r_{j|i}\} \in [0,1] \) 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 \( M_{\omega_i}(x) = S(\omega_i) - \max(S(\omega_i)) |i \neq j | \) \( (2) \)

where \( S(\omega_i) \) means the sum of the classification confidence in \( R \) whose corresponding classification decision is \( \omega_i \) and \( \omega_i \) 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 \( H = [h_{j|i}] \) \((i=1,2,\ldots,n; j = 1,2,\ldots,L)\) and the corresponding classification confidence matrix \( R = [r_{j|i}] \) \((h_{j|i}) \) means the classification decision of the classifier \( D_j \) on the sample \( x_i \) and \( r_{j|i} \) is the corresponding classification confidence for \( h_{j|i} \). Then for every row vector of \( R \), sort its elements making the value of the new first element is the largest. The sorted ith row of classification confidence matrix are denoted by \( r_{1}, r_{2}, \ldots, r_{L} \) and its corresponding classification decision are denoted by \( h_{1}, h_{2}, \ldots, h_{L} \). Furthermore, for \( j = 1,2,\ldots,L \), based on the first \( j \) elements in the sorted ith row \( r_{1}, r_{2}, \ldots, r_{j} \), \( h_{1}, h_{2}, \ldots, h_{j} \) and the true label \( Y(i) \) of the sample \( x_i \), the margin \( m_{y_i} \) of the sample \( x_i \) is computed as Definition 2. The corresponding margin loss is computed as \( l_{j} = (1- m_{y_i})^{2} \) and the sum of margin loss \( l_{j}(i = 1,\ldots,n) \) for all the sample is denoted by \( T(j) = \sum_{i=1}^{n} l_{j} \). 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 \( H = [h_{j|i}] \) \((i=1,2,\ldots,n; j = 1,2,\ldots,L)\) where \( h_{j|i} \) 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 \( R = [r_{j|i}] \).

2. For \( i = 1,2,\ldots,n \)
   - Sort the elements of the ith row of classification confidence matrix \( R \) making the first element is the largest. Denote the elements of the sorted ith row of classification confidence matrix \( r_{1}, r_{2}, \ldots, r_{L} \) and the corresponding ith row of classification decision matrix \( h_{1}, h_{2}, \ldots, h_{L} \).

3. For \( j = 1,2,\ldots,L \)
   - Compute the margin \( m_{y_j} \) of the sample \( x_i \) using \( r_{1}, r_{2}, \ldots, r_{L} \), \( h_{1}, h_{2}, \ldots, h_{L} \) and the true label \( Y(i) \) as Definition 2 and the corresponding margin loss \( l_{j} = (1- m_{y_i})^{2} \).

4. The sum of all the sample margin loss \( l_{j}(i = 1,\ldots,n) \) is denoted by \( T(j) \).

5. For \( j = 1,2,\ldots,L \)
   - Estimate \( K \) with the minimum margin loss \( T(K) \).

6. Apply the classifiers on the test sample \( x \) to get classification decision vector \( H = [h_{j}] \) \((j = 1,2,\ldots,L)\) where \( h_{j} \) 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 \( R = [r_{j}] \).

7. Sort the elements of the classification decision vector \( H \) according to the corresponding classification confidence \( R \) making the first element with the largest decision confidence. Denote the elements of the sorted classification decision vector \( H = [h_{1}, h_{2}, \ldots, h_{L}] \).

8. Classify the test sample \( x \) using \( h_{1}, h_{2}, \ldots, h_{L} \) with weighted voting where the weight of the \( h_{j} \) is its corresponding classification confidence \( r_{j} \).

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, Hetero-
genous 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 perturba-
tion 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 \( R_i \); 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 \( S_i \), finally \( X_{i,0} \) is
transformed linearly into the new features and get \( X_{i,0} X_{i,0}^T \). In
the second rotation, we replace PCA with LSDA to transform the data.
LSDA, proposed by Cai, is an effective method for feature extrac-
tion [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 classi-
ifiers 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.
- \( \gamma \): the ratio of bootstrap sample in the training set.

Output: the classifier \( D_i \).

1: For \( i = 1, \ldots, L \)
2: Split \( F \) randomly into \( G \) subsets: \( F_{ij} \) \( (j = 1, \ldots, G) \) and
  each feature subset \( F_{ij} \) contains \( M=N/G \) features.
3: For \( j = 1, \ldots, G \)
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 \( \gamma \) 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 \( a_{ij,1}, \ldots, a_{ij,M} \), 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).
\[
R_i = \begin{bmatrix}
    a_{i,1}^{1}, \ldots, a_{i,1}^{M} & 0 & \cdots & 0 \\
    0 & a_{i,2}^{1}, \ldots, a_{i,2}^{M} & \cdots & 0 \\
    0 & 0 & \cdots & a_{i,G}^{1}, \ldots, a_{i,G}^{M}
\end{bmatrix}
\]
10: Construct \( R_i^T \) by rearrange the columns of \( R_i \) so that they
  correspond to the original features and denote the
  rearranged rotation matrix \( R_i^T \).
11: Use \( X R_i^T \) 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 \( X R_i^T \)) to get the new rotation
  matrix \( S_i \).
12: Select a bootstrap sample from \( X R_i^T S_i^T \) of \( \gamma \) of the number
  of objects in \( X R_i^T S_i^T \). 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 experi-
ments 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 \( \gamma \) 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>
<thead>
<tr>
<th>Data set</th>
<th>Instances</th>
<th>Features</th>
<th>Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Australian</td>
<td>690</td>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>Balancescale</td>
<td>625</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>cxr</td>
<td>690</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>derm</td>
<td>366</td>
<td>34</td>
<td>6</td>
</tr>
<tr>
<td>ecoli</td>
<td>336</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>German</td>
<td>1000</td>
<td>20</td>
<td>2</td>
</tr>
<tr>
<td>Glass</td>
<td>214</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>Heart</td>
<td>270</td>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>Horse</td>
<td>368</td>
<td>22</td>
<td>2</td>
</tr>
<tr>
<td>ICU</td>
<td>200</td>
<td>20</td>
<td>3</td>
</tr>
<tr>
<td>ico3</td>
<td>351</td>
<td>34</td>
<td>2</td>
</tr>
<tr>
<td>iris</td>
<td>150</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>pima</td>
<td>768</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>Segmentation</td>
<td>2310</td>
<td>19</td>
<td>7</td>
</tr>
<tr>
<td>Soybean</td>
<td>683</td>
<td>35</td>
<td>19</td>
</tr>
<tr>
<td>Thyroid</td>
<td>215</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>wdbc</td>
<td>569</td>
<td>30</td>
<td>2</td>
</tr>
<tr>
<td>Wine</td>
<td>178</td>
<td>13</td>
<td>3</td>
</tr>
<tr>
<td>Wiscon</td>
<td>699</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>Yeast</td>
<td>1484</td>
<td>7</td>
<td>2</td>
</tr>
</tbody>
</table>
The ranking of 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.

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.
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 tree 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 Martinez-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 of classifier Dj is defined as the n dimensional vector whose ith component is (cij) whose quantity is 1 if the classifier Dj correctly classifies the training sample x; and 0 otherwise. The average ensemble signature vector = 1/ |x| 1 1 cij is defined as the average accuracy of each classifier 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  with equal components, namely,

\[ o_i = p \quad \text{with } i = 1, 2, \ldots, n \quad \text{and } 0 < p < 1 \]  

(4)

The first classifiers that are incorporated into the ensemble are those that reduce the distance from the vector \( \langle \mathbf{c} \rangle \) to the objective point q the most. In particular, the classifier selected in the ith iteration is the one that minimizes

\[ s_i = \arg\min\left( d\left( o_i, \frac{1}{n} \sum_{j=1}^{n} c_{ij} \right) \right) \]

(5)

where i runs through 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 ≤ 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 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 α = 0.05 is

\[ CD = q_{0.05} \sqrt{\frac{k(k+1)}{6N}} = 2.728 \times \sqrt{\frac{5}{(5-1)(6 \times 20)}} = 1.364 \]

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, 3.64, 4.65−1.40 = 3.25 > 1.364, 2.85−1.40 = 1.45 > 1.364, 3.10−1.40 = 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.

<table>
<thead>
<tr>
<th>Data set</th>
<th>DCE-CC</th>
<th>SV</th>
<th>NN</th>
<th>DCS-LA</th>
<th>MDM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Australian</td>
<td>84.21 ± 4.35</td>
<td>80.29 ± 4.26</td>
<td>78.85 ± 4.69</td>
<td>81.74 ± 4.66</td>
<td>82.61 ± 2.71</td>
</tr>
<tr>
<td>Balancescale</td>
<td>75.18 ± 4.60</td>
<td>70.81 ± 7.11</td>
<td>70.16 ± 8.28</td>
<td>78.74 ± 6.79</td>
<td>72.38 ± 9.43</td>
</tr>
<tr>
<td>ccx</td>
<td>82.59 ± 14.96</td>
<td>80.72 ± 13.78</td>
<td>78.98 ± 11.72</td>
<td>81.16 ± 11.46</td>
<td>80.87 ± 16.51</td>
</tr>
<tr>
<td>derm</td>
<td>96.63 ± 1.78</td>
<td>96.56 ± 2.00</td>
<td>96.35 ± 2.17</td>
<td>96.79 ± 2.96</td>
<td>96.61 ± 1.21</td>
</tr>
<tr>
<td>ecoli</td>
<td>85.25 ± 2.53</td>
<td>82.16 ± 5.49</td>
<td>79.56 ± 6.23</td>
<td>82.29 ± 6.82</td>
<td>81.18 ± 1.61</td>
</tr>
<tr>
<td>German</td>
<td>73.00 ± 4.52</td>
<td>70.50 ± 3.63</td>
<td>68.10 ± 3.87</td>
<td>70.20 ± 3.99</td>
<td>73.69 ± 2.55</td>
</tr>
<tr>
<td>Glass</td>
<td>72.48 ± 15.60</td>
<td>70.05 ± 15.23</td>
<td>65.77 ± 9.55</td>
<td>70.60 ± 11.22</td>
<td>69.73 ± 9.85</td>
</tr>
<tr>
<td>Heart</td>
<td>80.00 ± 6.10</td>
<td>77.04 ± 5.47</td>
<td>75.19 ± 9.88</td>
<td>78.89 ± 6.99</td>
<td>79.26 ± 4.22</td>
</tr>
<tr>
<td>Horse</td>
<td>92.31 ± 0.86</td>
<td>91.87 ± 4.65</td>
<td>80.70 ± 4.97</td>
<td>90.21 ± 6.11</td>
<td>91.89 ± 4.19</td>
</tr>
<tr>
<td>ICU</td>
<td>91.50 ± 8.92</td>
<td>89.97 ± 8.55</td>
<td>84.19 ± 17.78</td>
<td>92.08 ± 3.39</td>
<td>90.48 ± 5.16</td>
</tr>
<tr>
<td>iono</td>
<td>86.62 ± 6.32</td>
<td>87.01 ± 6.33</td>
<td>86.37 ± 4.62</td>
<td>86.71 ± 5.39</td>
<td>85.89 ± 4.97</td>
</tr>
<tr>
<td>iris</td>
<td>96.53 ± 3.32</td>
<td>96.00 ± 3.44</td>
<td>95.33 ± 4.06</td>
<td>95.67 ± 3.58</td>
<td>95.69 ± 5.06</td>
</tr>
<tr>
<td>pima</td>
<td>74.34 ± 5.44</td>
<td>72.39 ± 4.16</td>
<td>69.53 ± 3.78</td>
<td>71.87 ± 4.45</td>
<td>71.95 ± 4.74</td>
</tr>
<tr>
<td>Segmentation</td>
<td>97.49 ± 1.23</td>
<td>97.19 ± 1.60</td>
<td>96.67 ± 1.89</td>
<td>95.06 ± 1.55</td>
<td>96.12 ± 2.00</td>
</tr>
<tr>
<td>Soybean</td>
<td>91.49 ± 4.60</td>
<td>90.93 ± 4.59</td>
<td>90.77 ± 4.67</td>
<td>90.82 ± 3.95</td>
<td>90.96 ± 2.23</td>
</tr>
<tr>
<td>Thyroid</td>
<td>96.39 ± 3.58</td>
<td>96.23 ± 4.37</td>
<td>95.80 ± 4.16</td>
<td>95.85 ± 1.70</td>
<td>98.18 ± 1.21</td>
</tr>
<tr>
<td>wdbc</td>
<td>96.85 ± 2.29</td>
<td>96.67 ± 1.92</td>
<td>95.09 ± 3.05</td>
<td>96.14 ± 1.99</td>
<td>96.64 ± 3.00</td>
</tr>
<tr>
<td>Wine</td>
<td>95.75 ± 3.04</td>
<td>94.31 ± 4.42</td>
<td>93.86 ± 6.07</td>
<td>94.62 ± 4.58</td>
<td>94.00 ± 5.94</td>
</tr>
<tr>
<td>Wiscon</td>
<td>97.28 ± 2.97</td>
<td>95.86 ± 4.18</td>
<td>95.00 ± 3.70</td>
<td>96.00 ± 3.14</td>
<td>95.29 ± 4.40</td>
</tr>
<tr>
<td>Yeast</td>
<td>67.87 ± 4.52</td>
<td>66.39 ± 4.64</td>
<td>70.36 ± 5.97</td>
<td>67.53 ± 4.88</td>
<td>66.44 ± 3.52</td>
</tr>
</tbody>
</table>
Table 3
Classification performance of C4.5 and fusion methods with C4.5 generated by Algorithm 2.

<table>
<thead>
<tr>
<th>Data set</th>
<th>DCE-CC</th>
<th>SV</th>
<th>C4.5</th>
<th>DCS-LA</th>
<th>MDM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Australian</td>
<td>86.92 ± 4.47</td>
<td>85.08 ± 4.94</td>
<td>79.85 ± 4.33</td>
<td>82.76 ± 1.88</td>
<td>86.67 ± 6.67</td>
</tr>
<tr>
<td>Balancescale</td>
<td>81.59 ± 10.26</td>
<td>78.88 ± 6.31</td>
<td>78.46 ± 9.91</td>
<td>84.24 ± 8.66</td>
<td>78.39 ± 3.98</td>
</tr>
<tr>
<td>crx</td>
<td>83.90 ± 17.69</td>
<td>83.75 ± 18.41</td>
<td>76.98 ± 5.72</td>
<td>79.69 ± 14.53</td>
<td>82.61 ± 20.13</td>
</tr>
<tr>
<td>derm</td>
<td>97.16 ± 2.81</td>
<td>94.36 ± 2.65</td>
<td>88.81 ± 5.74</td>
<td>96.79 ± 2.96</td>
<td>94.44 ± 3.26</td>
</tr>
<tr>
<td>ecoli</td>
<td>85.54 ± 3.86</td>
<td>84.41 ± 4.29</td>
<td>81.16 ± 6.19</td>
<td>84.63 ± 4.85</td>
<td>81.12 ± 3.36</td>
</tr>
<tr>
<td>German</td>
<td>75.20 ± 2.49</td>
<td>74.60 ± 3.37</td>
<td>69.20 ± 4.98</td>
<td>74.70 ± 4.11</td>
<td>75.16 ± 9.20</td>
</tr>
<tr>
<td>Glass</td>
<td>67.37 ± 14.42</td>
<td>66.38 ± 12.58</td>
<td>61.24 ± 6.85</td>
<td>63.76 ± 16.23</td>
<td>66.45 ± 8.86</td>
</tr>
<tr>
<td>Heart</td>
<td>82.22 ± 7.16</td>
<td>81.85 ± 7.08</td>
<td>69.26 ± 10.48</td>
<td>75.93 ± 8.60</td>
<td>80.22 ± 6.73</td>
</tr>
<tr>
<td>Horse</td>
<td>91.61 ± 4.91</td>
<td>91.30 ± 4.73</td>
<td>91.57 ± 4.12</td>
<td>91.41 ± 3.16</td>
<td></td>
</tr>
<tr>
<td>ICU</td>
<td>92.03 ± 5.28</td>
<td>92.03 ± 7.44</td>
<td>78.77 ± 13.67</td>
<td>89.97 ± 5.28</td>
<td>92.43 ± 1.13</td>
</tr>
<tr>
<td>iono</td>
<td>92.38 ± 7.07</td>
<td>92.09 ± 4.86</td>
<td>77.67 ± 9.66</td>
<td>87.08 ± 7.26</td>
<td>92.21 ± 5.05</td>
</tr>
<tr>
<td>iris</td>
<td>97.33 ± 2.13</td>
<td>96.13 ± 6.32</td>
<td>93.53 ± 5.49</td>
<td>97.63 ± 2.32</td>
<td>95.26 ± 5.26</td>
</tr>
<tr>
<td>pima</td>
<td>76.83 ± 5.18</td>
<td>76.70 ± 5.03</td>
<td>69.53 ± 5.00</td>
<td>71.23 ± 4.64</td>
<td>79.76 ± 3.10</td>
</tr>
<tr>
<td>Segmentation</td>
<td>94.11 ± 1.96</td>
<td>92.16 ± 2.47</td>
<td>92.60 ± 1.85</td>
<td>96.10 ± 1.71</td>
<td>90.31 ± 6.80</td>
</tr>
<tr>
<td>Soybean</td>
<td>90.58 ± 5.75</td>
<td>92.23 ± 3.96</td>
<td>90.04 ± 7.44</td>
<td>89.60 ± 9.43</td>
<td>88.53 ± 2.63</td>
</tr>
<tr>
<td>Thyroid</td>
<td>97.13 ± 3.96</td>
<td>96.23 ± 3.61</td>
<td>93.96 ± 5.77</td>
<td>93.90 ± 5.03</td>
<td>96.36 ± 1.80</td>
</tr>
<tr>
<td>wdbc</td>
<td>97.90 ± 2.15</td>
<td>97.72 ± 2.20</td>
<td>92.80 ± 3.93</td>
<td>94.04 ± 2.02</td>
<td>97.54 ± 2.00</td>
</tr>
<tr>
<td>Wine</td>
<td>96.92 ± 2.63</td>
<td>96.60 ± 3.93</td>
<td>93.26 ± 4.37</td>
<td>93.21 ± 4.78</td>
<td>96.69 ± 2.19</td>
</tr>
<tr>
<td>wiscon</td>
<td>96.42 ± 3.10</td>
<td>96.42 ± 3.03</td>
<td>91.42 ± 4.36</td>
<td>97.00 ± 2.31</td>
<td>91.35 ± 1.10</td>
</tr>
<tr>
<td>Yeast</td>
<td>71.23 ± 3.73</td>
<td>71.36 ± 3.96</td>
<td>73.58 ± 4.59</td>
<td>72.01 ± 3.54</td>
<td>73.96 ± 1.45</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Data set</th>
<th>SV</th>
<th>SDCE-CC</th>
<th>WV</th>
<th>DCE-CC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Australian</td>
<td>80.29 ± 4.26</td>
<td>84.09 ± 3.61</td>
<td>83.63 ± 3.94</td>
<td>84.21 ± 4.35</td>
</tr>
<tr>
<td>Balancescale</td>
<td>70.81 ± 7.11</td>
<td>72.15 ± 5.92</td>
<td>70.73 ± 7.01</td>
<td>75.18 ± 4.60</td>
</tr>
<tr>
<td>crx</td>
<td>80.72 ± 13.78</td>
<td>81.89 ± 14.96</td>
<td>82.19 ± 13.90</td>
<td>82.59 ± 14.96</td>
</tr>
<tr>
<td>derm</td>
<td>96.56 ± 2.00</td>
<td>96.59 ± 3.16</td>
<td>96.59 ± 4.87</td>
<td>96.63 ± 1.78</td>
</tr>
<tr>
<td>ecoli</td>
<td>82.16 ± 5.49</td>
<td>84.16 ± 3.09</td>
<td>84.14 ± 2.00</td>
<td>85.25 ± 2.53</td>
</tr>
<tr>
<td>German</td>
<td>70.50 ± 3.63</td>
<td>72.61 ± 4.52</td>
<td>71.60 ± 3.44</td>
<td>73.00 ± 4.52</td>
</tr>
<tr>
<td>Glass</td>
<td>70.05 ± 15.23</td>
<td>72.48 ± 1.56</td>
<td>74.44 ± 14.68</td>
<td>72.48 ± 15.60</td>
</tr>
<tr>
<td>Heart</td>
<td>77.04 ± 5.47</td>
<td>80.00 ± 6.10</td>
<td>78.89 ± 7.21</td>
<td>80.00 ± 6.10</td>
</tr>
<tr>
<td>Horse</td>
<td>91.87 ± 4.65</td>
<td>91.26 ± 5.19</td>
<td>91.60 ± 4.32</td>
<td>92.31 ± 4.86</td>
</tr>
<tr>
<td>Soybean</td>
<td>89.97 ± 8.55</td>
<td>89.97 ± 8.55</td>
<td>91.50 ± 8.92</td>
<td>91.50 ± 8.92</td>
</tr>
<tr>
<td>iono</td>
<td>87.01 ± 6.33</td>
<td>86.62 ± 6.32</td>
<td>86.54 ± 6.97</td>
<td>86.62 ± 6.32</td>
</tr>
<tr>
<td>iris</td>
<td>90.60 ± 3.44</td>
<td>96.37 ± 3.48</td>
<td>95.98 ± 3.16</td>
<td>96.53 ± 3.32</td>
</tr>
<tr>
<td>pima</td>
<td>72.39 ± 4.16</td>
<td>73.08 ± 5.14</td>
<td>71.87 ± 4.82</td>
<td>74.34 ± 5.44</td>
</tr>
<tr>
<td>Segmentation</td>
<td>97.19 ± 1.60</td>
<td>97.03 ± 1.51</td>
<td>97.01 ± 1.48</td>
<td>97.49 ± 1.23</td>
</tr>
<tr>
<td>Soybean</td>
<td>90.93 ± 4.59</td>
<td>91.16 ± 5.36</td>
<td>91.07 ± 4.37</td>
<td>91.49 ± 4.60</td>
</tr>
<tr>
<td>Thyroid</td>
<td>96.23 ± 4.37</td>
<td>93.94 ± 6.30</td>
<td>93.50 ± 4.50</td>
<td>96.39 ± 3.58</td>
</tr>
<tr>
<td>wdbc</td>
<td>96.67 ± 1.92</td>
<td>96.85 ± 2.29</td>
<td>96.82 ± 1.84</td>
<td>96.85 ± 2.29</td>
</tr>
<tr>
<td>Wine</td>
<td>94.31 ± 4.42</td>
<td>95.06 ± 4.16</td>
<td>95.12 ± 5.42</td>
<td>95.75 ± 3.04</td>
</tr>
<tr>
<td>wiscon</td>
<td>95.86 ± 4.18</td>
<td>97.28 ± 2.97</td>
<td>96.57 ± 3.24</td>
<td>97.28 ± 2.97</td>
</tr>
<tr>
<td>Yeast</td>
<td>66.39 ± 4.64</td>
<td>67.54 ± 4.12</td>
<td>67.67 ± 3.90</td>
<td>67.87 ± 4.52</td>
</tr>
</tbody>
</table>

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 ∈ [−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.
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. 3. Margin cumulative frequency of training samples using NN 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.

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

That is to say, we dynamically select a subset of base classifiers in dynamic ensembles we select the classifiers with large classi-

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 61017119, 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).

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.