Knowledge-Based Systems 67 (2014) 90–104

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Exploiting diversity for optimizing margin distribution in ensemble learning

Qinghua Hu [a,*], Leijun Li [a], Xiangqian Wu [a], Gerald Schaefer [b], Daren Yu [c]

[a] Biometric Computing Research Centre, School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China
[b] Department of Computer Science, Loughborough University, UK
[c] School of Energy Science and Engineering, Harbin Institute of Technology, Harbin 150001, China

ARTICLE INFO

Article history:
Received 9 December 2013
Received in revised form 29 March 2014
Accepted 9 June 2014
Available online 18 June 2014

Keywords:
Ensemble learning
Margin distribution
Diversity
Fusion strategy
Rotation

ABSTRACT

Margin distribution is acknowledged as an important factor for improving the generalization performance of classifiers. In this paper, we propose a novel ensemble learning algorithm named Double Rotation Margin Forest (DRMF), that aims to improve the margin distribution of the combined system over the training set. We utilise random rotation to produce diverse base classifiers, and optimize the margin distribution to exploit the diversity for producing an optimal ensemble. We demonstrate that diverse base classifiers are beneficial in deriving large-margin ensembles, and that therefore our proposed technique will lead to good generalization performance. We examine our method on an extensive set of benchmark classification tasks. The experimental results confirm that DRMF outperforms other classical ensemble algorithms such as Bagging, AdaBoostM1 and Rotation Forest. The success of DRMF is explained from the viewpoints of margin distribution and diversity.

1. Introduction

Ensemble learning has been an active research area in pattern recognition and machine learning domains for more than twenty years [1,29,38,45,59]. Ensemble learning, also referred to as multiple classifier systems, committees of learners, decision forest or consensus theory, is based on the idea of training a set of base classifiers or regressors for a given learning task and combining their outputs through a fusion strategy.

A significant amount of works have been focused on designing effective ensemble classifiers [15,20,25,39]. However, an exact explanation of the success of ensemble strategies is still an open problem. Some researchers explored how an ensemble's effectiveness is related to the large margin principle, which is regarded as an important factor for improving classification [42,51]. In this paper, we propose a novel ensemble learning algorithm named Double Rotation Margin Forest (DRMF), which is designed to improve the margin distribution of ensembles by enhancing the diversity of base classifiers and exploiting this diversity using an optimization technique.

In general, there are two well-accepted viewpoints – diversity and margin – to explain the success of ensemble learning. Roughly speaking, the margin of a sample is its distance from the decision boundary and thus reflects the confidence of the classification. The margin distribution is acknowledged as an important factor for improving the generalization performance of classifiers [2,6,11,43,49]. In [43], Shawe-Taylor et al. gave an upper bound of generalization error in terms of the margin, while in [6] a similar bound was derived for neural networks with small weights. The large margin principle has been employed to design classification algorithms in [8,14,21,26,50].

The performance of ensemble learning methods, especially boosting, has been attributed to the improvement of the margin distribution of the training set [42,51]. In AdaBoost, each new base classifier is trained by taking into account the performance of the previous base classifiers. Training samples that are misclassified by the current base classifiers play a more important role in training the subsequent one. The success of Adaboost can thus be explained from the margin distribution, where the optimization objective is to minimize a margin distribution based exponential loss function. In [42], an upper bound of the generalization error was derived in terms of the margins of the training samples, and it was shown that the generalization performance was determined by the margin distribution, the number of training samples and the number of base classifiers. The efficacy of AdaBoost thus lies in its ability of effectively improving the margin distribution. In [51], Wang et al. showed that a larger Equilibrium margin (Emargin) and a smaller Emargin error can reduce the generalization error,

* Corresponding author. Tel.: +86 22 27401839.
E-mail addresses: huqinghua@hit.edu.cn (Q. Hu), lileijun1985@163.com (L. Li), xqwu@hit.edu.cn (X. Wu), Gerald.Schaefer@ieee.org (G. Schaefer), yudaren@hit.edu.cn (D. Yu).

http://dx.doi.org/10.1016/j.knosys.2014.06.005
0950-7051/

and demonstrated that AdaBoost can produce a larger Emargin and a smaller Emargin error.

It is acknowledged that the diversity among the members of an ensemble is crucial for performance improvement. Intuitively, no improvement can be achieved when a set of identical classifiers are combined. Diversity thus allows different classifiers to offer complementary information for classification, which in turn can lead to better performance [28]. A number of techniques have been proposed to introduce diversity. In general, we can divide these into two categories: classifier perturbation and sample perturbation approaches. Classifier perturbation refers to the adoption of instability of learning algorithms [10,36] such as decision trees and neural networks. Since they are sensitive to initialization, trained predictors may converge to different local minima if started from different initializations, and diversity can thus be generated from trained classifiers. Sample perturbation techniques train classifiers on different sample subsets or feature subsets, and include bagging, boosting, random subspaces and similar approaches [4,7,17,41].

Since both diversity and margin are argued to explain the success of ensemble learning, it appears natural to question whether there is a connection between the two. Tang et al. [46] proved that maximizing the diversity among base classifiers is equivalent to optimizing the margin of an ensemble on the training samples if the average classification accuracy is constant and maximal diversity is achievable. Consequently, increasing the diversity among base classifiers is an effective method to improve the margin of ensembles. Our work is motivated by this conclusion, and our aim is to improve the margin distribution of ensembles.

In our proposed approach, we enhance the diversity of base classifiers by perturbing the samples using double random rotation. This idea is inspired by the PCA rotation proposed in the Rotation Forest algorithm [39]. In Rotation Forest, a candidate feature set is randomly split into K subsets and Principal Component Analysis (PCA) is conducted on each subset to create diverse training samples. Diversity is thus promoted through the random feature splits for different base classifiers. In our work, the feature sets are also randomly split into K subsets. In order to introduce further diversity between the base classifiers, we apply PCA and Locality Sensitive Discriminant Analysis (LSDA). In particular, we first perform unsupervised rotation with PCA, and then employ supervised large-margin rotation with LSDA. LSDA [12], as a supervised method, is able to derive a projection which maximizes the margin between data points from different classes. Our experimental results show that the applied Double Rotation can consistently enhance the diversity in a set of base classifiers.

We further exploit the diversity and improve the margin distribution with an optimal fusion strategy. In principle, there are two kinds of fusion strategies. One approach is to combine all available classifiers, e.g., in simple (plurality) voting (SV) [28] or through linear or non-linear combination rules [5,9,19,48]. The other method is to derive selective ensembles, or pruned ensembles such as LP-Adaboost [23] or genetic algorithm (GA)-based approaches [53], which only select a fraction of the base classifiers for decision making and discard the others. Clearly, the key problem here is how to find an optimal subset of base classifiers [32]. In the GASEN approach [55], neural networks are selected based on evolved weights to constitute the ensemble. In [54], the subset selection problem is formulated as a quadratic integer programming problem, and semi-definite programming is adopted to select the base classifiers. Both GASEN and semi-definite programming are global optimization methods and thus their computational complexity is rather high. Suboptimal ensemble pruning methods were proposed to overcome this drawback, including reduce-error pruning [31], margin distance minimization (MDM) [33], orientation ordering [34], boosting-based ordering [35], and expectation propagation [13]. In practice, users would prefer sparse ensembles since computational resources are often limited [57]. In this paper, we introduce a technique to improve the margin distribution by minimizing the margin induced classification loss. In our pruned ensembles, the weights of base classifiers are trained with L_1 regularized squared loss [56]. The base classifiers are then sorted according to their weights, and those with large weights are selected in the final ensemble.

Our presented work comprises three major contributions. First, since diversity is considered to be an important factor which affects the classification margin, Double Rotation is proposed to enhance the diversity among base classifiers. Second, we present a new pruned fusion method based on the Lasso technique for generating ensembles with optimal margin and sparse weight vectors, where the weights are learned through minimization of the regularized squared loss function. Third, we present an extensive set of experiments to evaluate the effectiveness and explain the rationality of the proposed algorithm. We convincingly show that it can improve the margin distribution to a great extent and lead to powerful ensembles.

The remainder of the paper is organized as follows. Related work is introduced in Section 2. Section 3 describes our proposed algorithm, while an analysis in terms of parameter sensitivity

Table 1
Statistics of classification tasks.

Dataset	Instances	Discrete features	Continuous features	Classes
Australian	690	8	6	2
Crx	690	9	6	2
Cmc	1473	7	2	3
Derm	366	0	34	6
German	1000	13	7	2
Glass	214	0	9	6
Heart	270	0	13	2
Horse	368	15	7	2
ICU	200	16	4	3
Iono	351	0	34	2
Iris	150	0	4	3
Movement	360	0	90	15
Pima	768	0	8	2
Rice	104	0	5	2
Spectf	269	0	44	2
Thyroid	215	0	5	3
Wiscon	699	0	9	2
Wdbc	569	0	30	2
Yeast	1484	0	7	2
Zoo	101	15	1	7

Table 2
Classification performance of DRMF with different numbers of splits.

Data set	$K=2$	$K=3$	$K=4$	$K=5$
Australian	**88.11 ± 3.48**	87.24 ± 4.10	87.10 ± 2.94	87.97 ± 3.63
Crx	86.37 ± 13.66	86.22 ± 15.01	**86.53 ± 13.58**	86.37 ± 14.28
Cmc	54.24 ± 3.25	54.45 ± 3.75	**55.54 ± 3.43**	52.89 ± 2.65
Derm	**96.75 ± 3.84**	95.95 ± 4.28	96.19 ± 3.22	95.71 ± 3.20
German	**77.80 ± 2.94**	76.70 ± 3.33	75.60 ± 3.63	76.40 ± 4.01
Glass	**76.64 ± 10.61**	73.37 ± 8.64	75.69 ± 11.01	62.06 ± 14.65
Heart	84.44 ± 4.88	83.70 ± 5.00	**86.30 ± 3.51**	80.00 ± 6.34
Horse	**93.49 ± 3.83**	91.85 ± 5.25	92.94 ± 4.06	93.21 ± 3.88
ICU	93.56 ± 4.80	89.45 ± 11.53	**93.61 ± 3.99**	90.98 ± 8.51
Iono	93.47 ± 4.76	**95.24 ± 3.69**	93.52 ± 5.05	95.21 ± 4.56
Iris	**98.67 ± 2.81**	94.00 ± 4.92	96.00 ± 3.44	96.00 ± 3.44
Movement	**82.44 ± 16.29**	81.56 ± 17.95	81.00 ± 16.63	80.22 ± 19.09
Pima	**78.78 ± 3.76**	77.35 ± 4.90	77.87 ± 4.61	73.57 ± 3.65
Rice	**89.82 ± 13.17**	79.05 ± 10.04	79.96 ± 10.70	79.05 ± 10.04
Spectf	82.48 ± 7.91	83.28 ± 3.08	82.58 ± 7.00	**83.94 ± 7.63**
Thyroid	**96.26 ± 4.86**	93.48 ± 7.92	93.96 ± 8.17	93.48 ± 5.93
Wiscon	**97.86 ± 2.36**	97.57 ± 3.16	96.57 ± 3.24	92.99 ± 3.71
Wdbc	**97.72 ± 1.66**	95.80 ± 3.42	97.21 ± 2.49	96.84 ± 2.58
Yeast	**73.25 ± 3.47**	72.57 ± 3.73	70.68 ± 5.80	70.68 ± 5.80
Zoo	**94.39 ± 8.39**	92.39 ± 11.40	93.14 ± 9.63	**94.39 ± 8.39**
Average	86.83	85.06	85.60	84.10

The best result for each data set is highlighted in bold face.

and robustness is presented in Section 4. Section 5 presents the experimental results and explores the rationality of DRMF. Finally, Section 6 offers conclusions and future work.

2. Related work

Assume that $x_i = [x_{i1}, \ldots, x_{in}]^T$ is a sample represented by a set F of n features and every sample is generated independently at random according to some fixed but unknown distribution $\mathfrak{D}$. Let X be an $N \times n$ matrix containing the training set and $Y = [y_1, \ldots, y_N]^T$ be an N-dimensional vector containing the class labels for the data, where y_i is a class label of x_i from the set of the class labels $\{\omega_1, \ldots, \omega_c\}$. Let $\{\mathcal{C}_1, \ldots, \mathcal{C}_L\}$ be the set of base classifiers in an ensemble. In this paper, our aim is to obtain an ensemble system with small generalization error via optimizing the margin distribution. Here, the generalization error of a classifier $\mathcal{C}_j$ is the probability of $\mathcal{C}_j(x) \neq y$ when an example (x, y) is chosen at random according to the distribution $\mathfrak{D}$ and denoted as $P_{\mathfrak{D}}[\mathcal{C}_j(x) \neq y]$. The margin distribution is a function of θ which gives the fraction of samples whose margin is smaller than θ. A good margin distribution means that most examples have large margins.

Definition 1. Given $x_i \in X, h_{ij} (j = 1, 2 \ldots, L)$ is the output of x_i from $\mathcal{C}_j$. We define

$$d_{ij} = \begin{cases} 1, & \text{if } y_i = h_{ij} \\ -1, & \text{if } y_i \neq h_{ij} \end{cases}, \tag{1}$$

where y_i is the real class label of x_i.

From this definition, we know that $d_{ij} = 1$ if x_i is correctly classified by $\mathcal{C}_j$; otherwise $d_{ij} = -1$.

Definition 2 [42]. Given $x_i \in X$, the margin of x_i in terms of the ensemble is defined as

$$m(x_i) = \sum_{j=1}^{L} w_j d_{ij}, \tag{2}$$

where w_j is the weight of $\mathcal{C}_j$ and $w_j > 0$.

In [42,51], it is shown that a small generalization error for a voting classifier can be obtained by a good margin distribution on the training set. Obviously, the performances of the base classifiers have a significant effect on the margin of x_i. At the same time, the diversity among base classifiers is another key factor. In [46], the underlying relationship between diversity and margin was analyzed.

Theorem 1 [46]. *Let Θ be the average classification accuracy of the base classifiers. If Θ is regarded as a constant and if maximum diversity is achievable, maximization of the diversity among base classifiers is equivalent to maximization of the minimal margin of the ensemble on the training samples.*

It should be noted that our aim is not to maximize the minimal margin of the ensemble, but to optimize the margin distribution. We use a disagreement measure [30] to measure the diversity of the base classifiers in our approach. The diversity between classifiers $\mathcal{C}_j$ and $\mathcal{C}_k$ is thus computed as

$$Dis_{jk} = \frac{N^{01} + N^{10}}{N^{11} + N^{10} + N^{01} + N^{00}}, \tag{3}$$

where N^{00} denotes the number of samples misclassified by both classifiers, N^{11} is the number of samples correctly classified by both, N^{10} denotes the number of samples which were correctly classified by $\mathcal{C}_j$ but misclassified by $\mathcal{C}_k$, and N^{01} denotes the number of samples misclassified by $\mathcal{C}_j$ but correctly classified by $\mathcal{C}_k$. For multiple base classifiers, the overall diversity is computed as the average diversity of classifier pairs.

In [39], Rodríguez and Kuncheva designed a method to generate ensembles based on feature transformation. The diversity of base classifiers is promoted by random splits of the feature set into different subsets. The original feature space is split into K subspaces (the subsets may be disjoint or may intersect). Then, PCA is applied to linearly rotate the subspaces along the "rotation" matrix. Diversity is obtained by random splits of the feature set.

Cai et al. [12] proposed a supervised algorithm for feature transformation, which can find a projection that maximizes the margin between different classes. For $x_i \in X$, denote by $\Upsilon(x_i) = \{x_i^1, \ldots, x_i^e\}$ the set of its e nearest neighbors and by y_i the class label of x_i. We define

$$\Upsilon_s(x_i) = \{x_i^j | y_i^j = y_i, 1 \leqslant j \leqslant e\}, \tag{4}$$

and

$$\Upsilon_b(x_i) = \{x_i^j | y_i^j \neq y_i, 1 \leqslant j \leqslant e\}, \tag{5}$$

so that $\Upsilon_s(x_i)$ contains the neighbors which share the same label with x_i, while $\Upsilon_b(x_i)$ is the set of the neighbors which belong to the other classes.

For any x_i and x_j, we define

$$V_{b,ij} = \begin{cases} 1 & \text{if } x_i \in \Upsilon_b(x_j) \text{ or } x_j \in \Upsilon_b(x_i) \\ 0 & \text{otherwise} \end{cases}, \tag{6}$$

Table 3
Classification performance with different numbers of candidate base classifiers.

Data set	$L = 20$	$L = 40$	$L = 60$	$L = 80$	$L = 100$
Australian	86.96 ± 3.01	88.13 ± 3.98	87.54 ± 2.90	87.97 ± 3.27	88.11 ± 3.48
Crx	85.66 ± 14.10	85.51 ± 14.66	85.64 ± 15.10	86.81 ± 13.20	86.37 ± 13.66
Cmc	52.82 ± 3.41	53.97 ± 3.30	53.70 ± 3.28	54.18 ± 2.95	54.24 ± 3.25
Derm	96.47 ± 3.90	96.98 ± 4.24	95.91 ± 5.26	96.47 ± 4.12	96.75 ± 3.84
German	75.40 ± 3.60	77.00 ± 3.40	77.00 ± 3.02	77.70 ± 2.45	77.80 ± 2.94
Glass	72.44 ± 12.14	74.44 ± 11.43	74.89 ± 13.25	78.14 ± 10.91	76.64 ± 10.61
Heart	83.33 ± 3.60	83.70 ± 4.68	84.81 ± 4.43	84.81 ± 4.77	84.44 ± 4.88
Horse	92.95 ± 4.06	92.94 ± 4.06	93.49 ± 3.38	93.22 ± 3.39	93.49 ± 3.83
ICU	94.04 ± 4.69	94.04 ± 4.69	93.56 ± 4.80	94.09 ± 5.21	93.56 ± 4.80
Iono	93.20 ± 4.00	92.92 ± 4.60	93.49 ± 4.95	93.77 ± 5.09	93.47 ± 4.76
Iris	94.67 ± 5.26	94.67 ± 5.26	94.67 ± 5.26	96.67 ± 4.71	98.67 ± 2.81
Movement	80.56 ± 15.15	82.78 ± 16.50	82.78 ± 16.57	82.44 ± 16.29	82.44 ± 16.29
Pima	77.87 ± 4.98	78.52 ± 3.90	78.39 ± 4.53	78.39 ± 4.32	78.78 ± 3.76
Rice	89.73 ± 10.18	88.82 ± 10.46	90.73 ± 12.86	89.82 ± 13.17	89.82 ± 13.17
Spectf	82.61 ± 5.18	83.34 ± 4.39	82.12 ± 7.21	83.25 ± 7.01	82.48 ± 7.91
Thyroid	94.83 ± 6.10	95.30 ± 6.31	94.83 ± 5.21	95.78 ± 5.19	96.26 ± 4.86
Wiscon	97.34 ± 2.59	97.43 ± 2.59	97.71 ± 2.15	97.34 ± 2.59	97.86 ± 2.36
Wdbc	97.19 ± 2.06	97.72 ± 1.66	98.43 ± 1.53	97.72 ± 1.66	97.72 ± 1.66
Yeast	73.11 ± 3.26	73.45 ± 3.12	73.45 ± 3.64	73.38 ± 3.61	73.25 ± 3.47
Zoo	94.39 ± 8.39	94.39 ± 8.39	94.39 ± 8.39	94.39 ± 8.39	94.39 ± 8.39
Average	85.78	86.30	86.38	86.82	86.83

Table 4
Classification performance and number of selected base classifiers for pruned simple voting (PSV) and pruned weighted voting (PWV).

Data set	PSV accuracy	PSV No. class.	PWV accuracy	PWV No. class.
Australian	88.11 ± 3.48	13.9	87.10 ± 3.83	71.4
Crx	86.37 ± 13.66	18.2	85.50 ± 12.22	80.6
Cmc	54.24 ± 3.25	26.9	52.55 ± 3.07	81.7
Derm	96.75 ± 3.84	4.2	96.47 ± 5.38	3.7
German	77.80 ± 2.94	20.6	76.70 ± 3.43	72.6
Glass	76.64 ± 10.61	9.8	78.51 ± 7.88	39.8
Heart	84.44 ± 4.88	15.4	82.59 ± 4.95	25.5
Horse	93.49 ± 3.83	4.4	91.03 ± 4.24	3.7
ICU	93.56 ± 4.80	2.4	94.14 ± 4.35	11.9
Iono	93.47 ± 4.76	4.9	90.95 ± 6.61	2.6
Iris	98.67 ± 2.81	6.6	98.00 ± 3.22	1
Movement	82.44 ± 16.29	11.5	79.33 ± 20.35	7.2
Pima	78.78 ± 3.76	17.2	77.35 ± 4.44	72.7
Rice	89.82 ± 13.17	4.8	84.98 ± 12.79	51.4
Spectf	82.48 ± 7.91	4.9	80.24 ± 8.33	2.6
Thyroid	96.26 ± 4.86	1.8	95.78 ± 5.19	1.2
Wiscon	97.86 ± 2.36	4.4	97.86 ± 2.54	70
Wdbc	97.72 ± 1.66	8.5	97.01 ± 2.05	3.9
Yeast	73.25 ± 3.47	24.8	71.69 ± 3.89	70.6
Zoo	94.39 ± 8.39	1.3	93.39 ± 8.24	1
Average	86.83	10.33	85.56	33.76

and

$$V_{s,ij} = \begin{cases} 1 & \text{if } x_i \in \Upsilon_s(x_j) \text{ or } x_j \in \Upsilon_s(x_i) \\ 0 & \text{otherwise} \end{cases}. \tag{7}$$

V_b and V_s thus give the weight matrices of the between-class graph G_b and the within-class graph G_s respectively.

The objective of Locality Sensitive Discriminant Analysis (LSDA) is to map the within-class graph and the between-class graph to a line so that the connected points of G_s stay as close as possible while the connected points of G_b are as distant as possible. Suppose $x_1, \ldots, x_N$ are mapped to $z_1, \ldots, z_N$ and $z_i = \vartheta^T x_i$ where ϑ is a projection vector. In order to compute $z_1, \ldots, z_N$, the following Locality Sensitive Discriminant (LSD) objective functions are optimized:

$$\min \sum_{ij} (z_i - z_j)^2 V_{s,ij}, \tag{8}$$

$$\max \sum_{ij} (z_i - z_j)^2 V_{b,ij}. \tag{9}$$

This optimization can be translated into maximum eigenvalue solutions to the generalized eigenvalue problem

$$X^T(p\Lambda_b + (1-p)V_s)X\vartheta = \lambda X^T Q_s X\vartheta, \tag{10}$$

where X is an $N \times n$ matrix, Q_s is a diagonal matrix whose entries are the column sums of V_s, and $\Lambda_b = Q_b - V_b$ where Q_b is a diagonal matrix whose entries are column sums of V_b. From the LSD objective functions, it can be seen that LSDA can discover both geometrical and discriminant structures in the data.

3. Algorithm description

As shown above, diversity is an important factor to improve margin distribution. In [42,51], the relationship between the generalization performance and the margin distribution of the training set was derived. It was found that if a voting classifier generates a good margin distribution, the generalization error will be small. Motivated by these results, we propose a novel technique to generate diverse base classifiers and to exploit diversity for producing good ensembles with an optimal margin distribution.

3.1. Double rotation

Double Rotation aims to enhance the diversity among base classifiers. In order to construct the training set for the base classifier $\mathcal{C}_j$, we first split the feature set F randomly into K subsets $F_{ij}(i = 1, 2 \ldots, K)$ which contain $M = \llcorner n/K \lrcorner$ features ($\llcorner n/K \lrcorner$ rounds n/K to the nearest integer) and denote by X_{ij} the data subset with features F_{ij}. We then eliminate a random subset of the classes and draw $\gamma \cdot N$ samples by bootstrapping from X_{ij} to obtain a new set X'_{ij}. We apply PCA on X'_{ij} to obtain the coefficients of the principal components $\mathrm{a}^1_{i,j}, \ldots, \mathrm{a}^{M_i}_{i,j}$.[1] From these we construct a "rotation" matrix R_j

$$R_j = \begin{bmatrix} \mathrm{a}^1_{1,j}, \ldots, \mathrm{a}^{M_1}_{1,j} & 0 & \cdots & 0 \\ 0 & \mathrm{a}^1_{2,j}, \ldots, \mathrm{a}^{M_2}_{2,j} & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \mathrm{a}^1_{K,j}, \ldots, \mathrm{a}^{M_K}_{K,j} \end{bmatrix}. \tag{11}$$

The columns of R_j are rearranged so that they correspond to the original features. If we denote the rearranged rotation matrix as R^a_j, then XR^a_j is taken as a new training set. We then repeat the above process but replace PCA with LSDA and obtain a new rotation matrix S^a_j. Finally, $\mathcal{C}_j$ is trained with $(XR^a_j S^a_j, Y)$.

The pseudocode of the Double Rotation algorithm is formulated in Algorithm 1.

Algorithm 1. Double Rotation.

Input:
X: the training set ($N \times n$ matrix)
Y: the labels of the training data set ($N \times 1$ matrix)
L: the number of classifiers in the ensemble
F: the feature set
K: the number of subsets
γ: the ratio of bootstrap sample in the training set
Output:
classifier $\mathcal{C}_j$
1: Split F randomly into K subsets $F_{ij}(i = 1, 2 \ldots, K)$ so that each feature subset contains $M = \llcorner n/K \lrcorner$ features
2: **for** $i = 1, 2 \ldots, K$ **do**
3: Let X_{ij} be the dataset X for the features in F_{ij}
4: Eliminate a random subset of classes X_{ij}
5: Select $\gamma \cdot N$ samples from X_{ij} by bootstrapping and denote the new set by X'_{ij}
6: Apply PCA on X'_{ij} to obtain the coefficients of the principal components $\mathrm{a}^1_{i,j}, \ldots, \mathrm{a}^{M_i}_{i,j}$
7: **end for**
8: Organize the obtained coefficients into a sparse "rotation" matrix R_j as defined in Eq. (11)
9: Construct R^a_j by rearranging the columns of R_j so that they correspond to the original features
10: Use XR^a_j as the new training data set and rerun the above process but replace PCA with LSDA to obtain a new rotation matrix S^a_j
11: Build the classifier $\mathcal{C}_j$ using $(XR^a_j S^a_j, Y)$ as the training set

[1] The reason for eliminating a random subset of classes and drawing $\gamma \cdot N$ samples by bootstrapping is to avoid identical coefficients of principal components when the same feature subset is chosen for different classifiers.

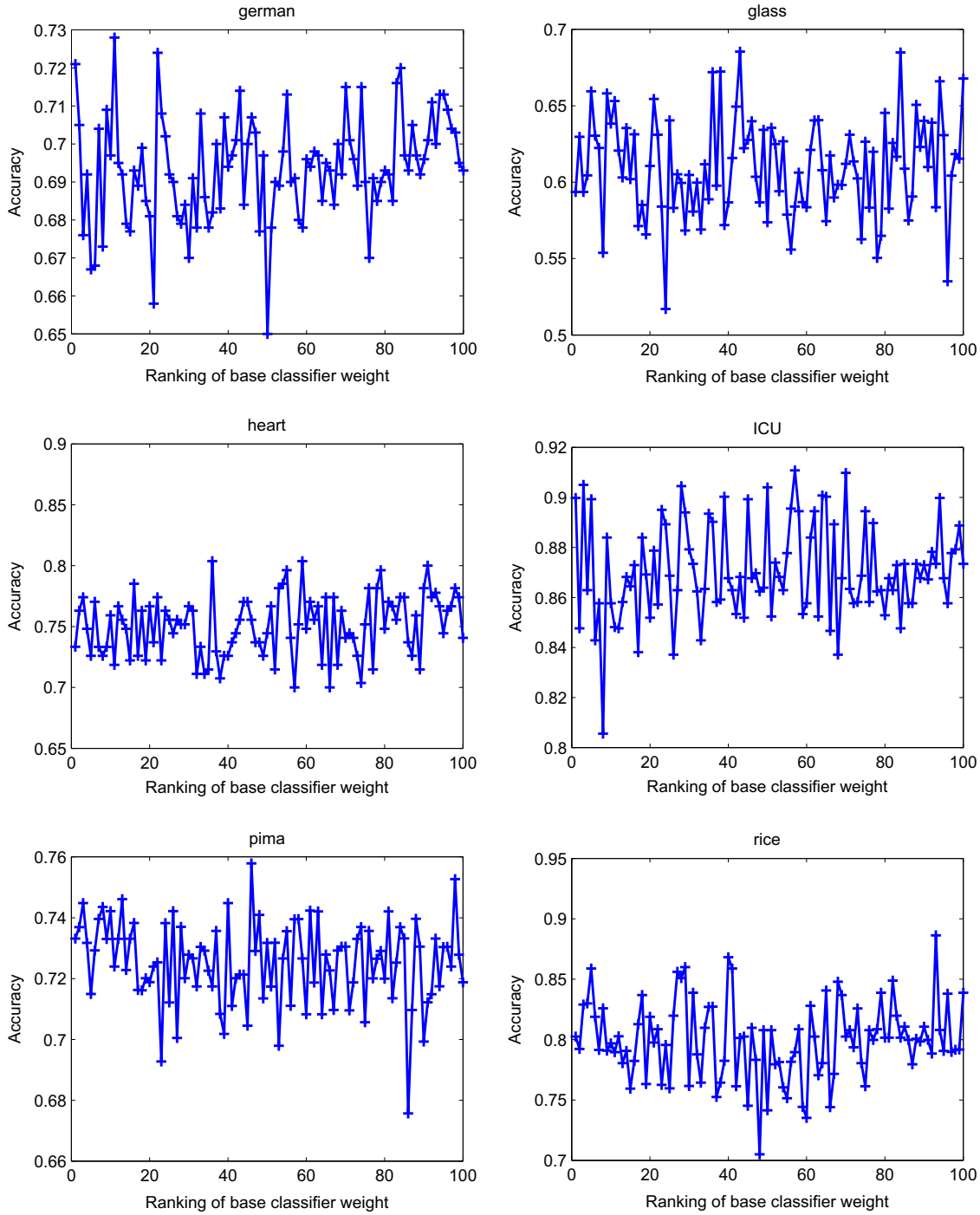

Fig. 1. Classification accuracies of base classifiers with different ranking in terms of their weights.

Double Rotation integrates two different feature transformation algorithms for boosting the diversity among base classifiers. In DRMF, the base classifiers are trained with the data $(XR_j^a S_j^a, Y)$ based on the J48 algorithm, an implementation of C4.5 in the WEKA library [27].

3.2. Ensemble Pruning by optimizing margin distribution

Based on the above procedure, we obtain a set of diverse decision tree classifiers. Now, we exploit this diversity to construct an optimal ensemble.

Given $x \in X, h_{xj} \in \{-1, 1\}$ as the output of x from C_j, and w_j as the weight of C_j, the final decision function is

$$f(x) = sgn\left(\sum_{j=1}^{L} w_j h_{xj}\right). \tag{12}$$

Here, $f(x)$ can be seen as a linear classifier in a new input space, where every sample x is represented as an L-dimensional vector $[h_{x1}, \ldots, h_{xL}]^T_{L\times 1}$, and then $w_j (j = 1, 2, \ldots, L)$ can be seen as the coefficients of this function. Based on the conclusion in [44], a bound of the generalization error for the linear classifier can be derived as follows.

Theorem 2. *For $\Delta > 0, t \in \Re$, consider a fixed but unknown probability distribution on the input space Φ with support in the ball of radius $\mathbb{R}$ about the origin. Then, with probability $1 - \delta$ over randomly*

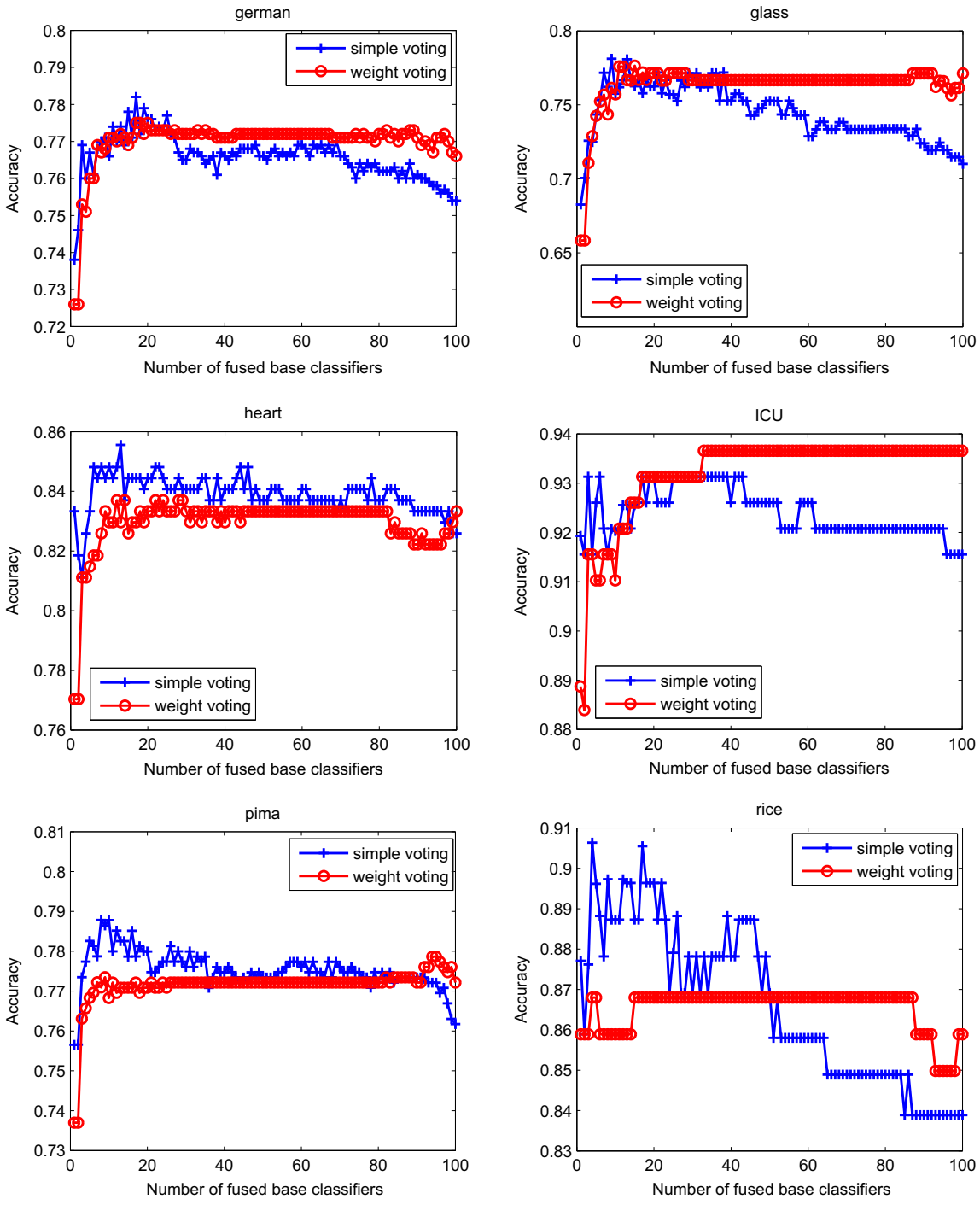

Fig. 2. Variation of classification accuracies with different numbers of selected base classifiers when simple voting and weighted voting are used.

drawn training set X of size N for all $\beta > 0$ *the generalization of the linear classifier* $f(x)$ *on the input space is bounded by*

$$\epsilon(N,\eta,\delta) = \frac{2}{N}\left(\eta \log_2\left(\frac{8eN}{\eta}\right)\log_2(32N) + \log_2\left(\frac{8N}{\delta}\right)\right), \tag{13}$$

where

$$\eta = \left\lfloor \frac{64.5(\mathbb{R}^2 + \Delta^2)(\|W\|^2 + E(X,(W,t),\beta)^2/\Delta^2)}{\beta^2} \right\rfloor, \tag{14}$$

provided $N \geqslant 2/\epsilon$, *and* $\eta \leqslant eN$.

In Theorem 2, $W = [w_1,\ldots,w_L]^T_{L\times 1}, x = [h_{x1},\ldots,h_{xL}]^T_{L\times 1}, y$ is the real class label of x and $t = 0$. Besides, $E(X,(W,t),\beta) = \sqrt{\sum_{(x,y)\in X}\varphi((x,y),(W,t),\beta)^2}$ and $\varphi((x,y),(W,t),\beta) = \max\{0, \beta - y(\langle W^T\cdot x\rangle - t)\}$. We can see that with β given, a small $\mathbb{R}$ and $E(X,(W,t),\beta)$ can lead to a good linear classifier. In fact, $\mathbb{R}$ is related to the number of base classifiers in the ensemble since if there are fewer base classifiers, $\mathbb{R}$ will become smaller. On the other hand, $y(\langle W^T\cdot x\rangle - t) = y(\langle W^T\cdot x\rangle) = y(\sum_{j=1}^L w_j h_{xj}) = \sum_{j=1}^L w_j y h_{xj} = \sum_{j=1}^L w_j d_{xj} = m(x)$ when $t = 0$ and $y, h_{xj} \in \{-1, 1\}$. Thus, $E(X,(W,t),\beta)$ can be understood as the root of the squared loss of ensembles,

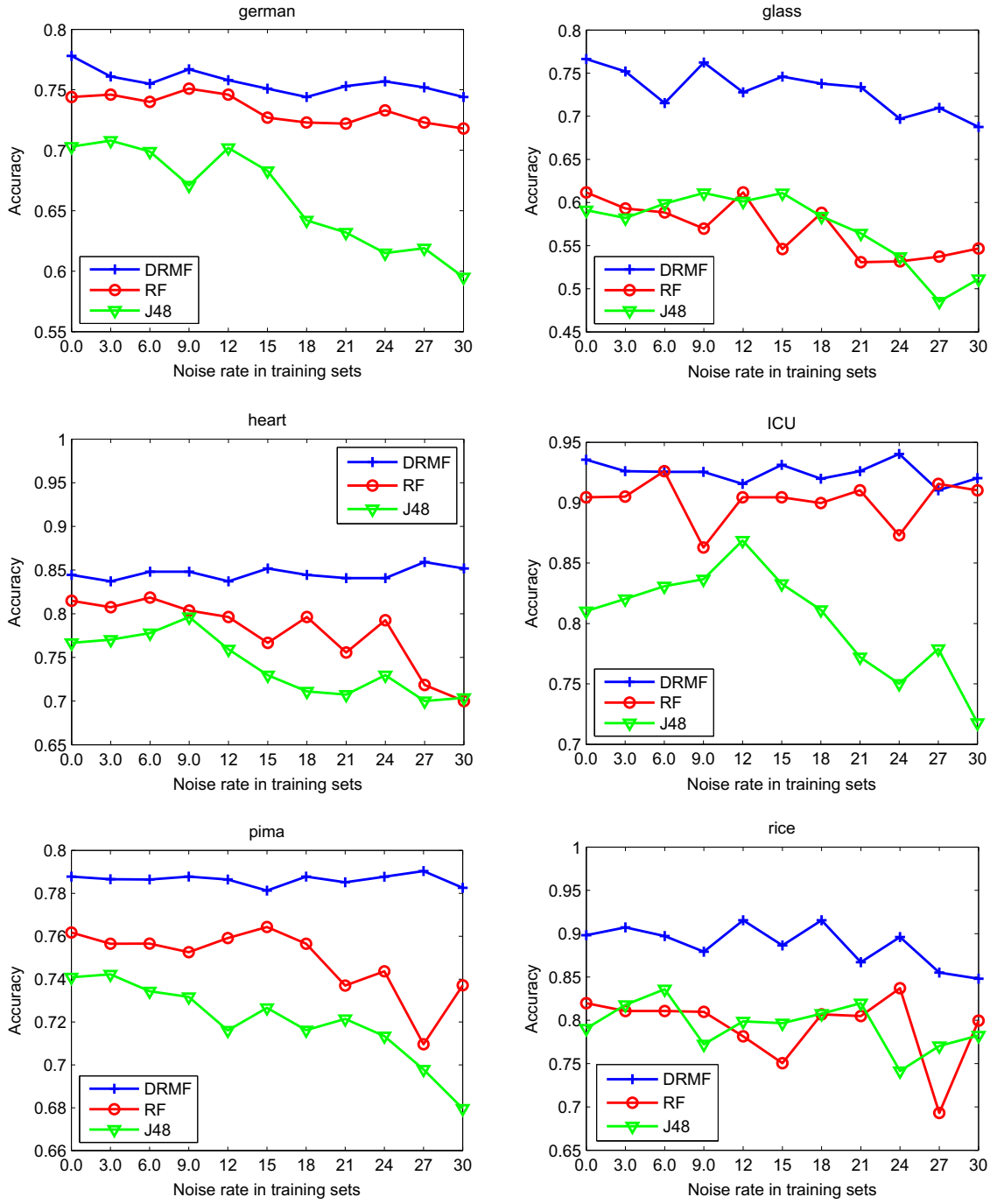

Fig. 3. Variation of classification accuracies when varying the rate of class noise.

which is determined by the margin distribution of the ensemble. Based on the theorem, we can design an optimization objective to learn the weights of the base classifiers.

Definition 3. Given $x_i \in X$, the classification loss of x_i is computed as

$$l(x_i) = [1 - m(x_i)]^2, \tag{15}$$

where $m(x_i)$ is the margin of x_i. Then, the classification loss of X is

$$l(X) = \sum_{i=1}^{n} l(x_i) = \|U - DW\|_2^2, \tag{16}$$

where $U = [1, \ldots, 1]^T_{N\times 1}, W = [w_1, \ldots, w_L]^T_{L\times 1}$ and $D = \{d_{ij}\}_{N\times L}$.

The above function only considers the classification loss related to the margin distribution. The number of base classifiers is not taken into account. However, from Theorem 2 we know that only few base classifiers should be included in ensembles, and consequently the weight vector should be sparse. A sparse model is expected to improve the generalization performance [18,22,58]. In order to obtain a sparse weight vector, we add the L_1 norm regularization term of the weight vector into the loss function. The regularized loss function is then

$$J_W = argmin\|U - DW\|_2^2 + \lambda\|W\|_1. \tag{17}$$

This is the well-known Lasso problem [47]. While Lasso has been employed in regression ensembles [24], we here apply it to

Table 5
Comparison of different strategies used in selecting base classifiers (FA = fusion accuracy, AA = average accuracy).

Data set	FA (LASSO)	FA (DRMF)	Diversity (LASSO)	Diversity (DRMF)	AA (LASSO)	AA (DRMF)
Australian	86.09 ± 3.88	88.11 ± 3.48	0.1122	0.1318	83.73	84.22
Crx	83.61 ± 18.39	86.37 ± 13.66	0.0993	0.1357	82.25	82.54
Cmc	52.41 ± 3.70	54.24 ± 3.25	0.3493	0.3517	47.08	47.19
Derm	97.86 ± 2.45	96.75 ± 3.84	0.1872	0.1776	86.27	87.59
German	75.40 ± 3.17	77.80 ± 2.94	0.2713	0.2986	69.37	69.15
Glass	71.01 ± 11.00	76.64 ± 10.61	0.2951	0.3232	60.98	62.30
Heart	82.59 ± 4.95	84.44 ± 4.88	0.2407	0.2562	75.07	75.81
Horse	91.04 ± 4.76	93.49 ± 3.83	0.1465	0.1585	86.89	88.12
ICU	92.08 ± 2.31	93.56 ± 4.80	0.0295	0.0317	87.06	89.58
Iono	95.73 ± 4.51	93.47 ± 4.76	0.2112	0.2330	86.73	87.61
Iris	95.33 ± 6.32	98.67 ± 2.81	0.0164	0.0180	94.49	98.57
Movement	80.44 ± 16.59	82.44 ± 16.29	0.3307	0.3255	59.48	61.37
Pima	76.17 ± 4.57	78.78 ± 3.76	0.2236	0.2332	72.52	73.27
Rice	83.89 ± 8.78	89.82 ± 13.17	0.1208	0.1212	79.97	83.37
Spectf	81.00 ± 6.09	82.48 ± 7.91	0.2420	0.2362	75.50	77.07
Thyroid	95.78 ± 4.16	96.26 ± 4.86	0.1083	0.1451	92.55	95.58
Wiscon	97.28 ± 2.65	97.86 ± 2.36	0.0366	0.0520	95.77	95.61
Wdbc	97.55 ± 2.05	97.72 ± 1.66	0.0838	0.0789	93.40	94.13
Yeast	71.62 ± 4.32	73.25 ± 3.47	0.1195	0.1383	70.55	70.86
Zoo	94.39 ± 8.39	94.39 ± 8.39	0.1045	0.0667	87.29	93.89
Average	85.06	86.83	0.1664	0.1757	79.35	80.89

classification tasks. Lasso can be explained as a large margin solution in classification in terms of the infinite norm [40]. By minimizing J_W, we obtain the weights $w_j (j = 1, 2, \ldots, L)$ of the base classifiers.

Given the weight coefficients, the base classifiers are sorted, and then a suboptimal subset is selected for classifying previously unseen samples. Algorithm 2 describes the approach in pseudocode.

Algorithm 2. Margin Based Pruning

Input:
X: the training set ($N \times n$ matrix)
Y: the labels of the training set ($N \times 1$ matrix)
$\mathcal{C}_j (j = 1, 2, \ldots, L)$: the base classifiers
x: a test sample
Output:
the label of x;
1: Apply $\mathcal{C}_j (j = 1, 2, \ldots, L)$ on the training set X, and compare the classification results with Y to obtain D from Definition 3
2: Minimize $J_W = argmin\|U - DW\|_2^2 + \lambda\|W\|_1$ to obtain the weights $w_j (j = 1, 2, \ldots, L)$
3: Sort the base classifiers according to their weights in descending order $\mathcal{C}_{sj} (sj = 1, 2, \ldots, L)$
4: **for** $j = 1, 2, \ldots, L$ **do**
5: Classify the training set X with the classifiers $\{\mathcal{C}_{s1}, \mathcal{C}_{s2}, \ldots, \mathcal{C}_{sj}\}$ and combine their outputs using simple voting to obtain the predicted labels of X
6: Compare the classification results with Y and compute the corresponding accuracy ψ_j
7: **end for**
8: Choose the subset of base classifier $\{\mathcal{C}_{s1}, \mathcal{C}_{s2}, \ldots, \mathcal{C}_{sB}\}$ that produces the maximal accuracy ψ_B in $\{\psi_1, \psi_2, \ldots, \psi_L\}$
9: Use the ensemble $\{\mathcal{C}_{s1}, \mathcal{C}_{s2}, \ldots, \mathcal{C}_{sB}\}$ to classify the unseen sample x

Essentially, our proposed technique is an ordered aggregation pruning method based on the weights of the base classifiers, where the weights are trained by minimizing a margin induced classification loss.

4. Algorithm analysis

There are several parameters to be set in our proposed algorithm. In this section, we discuss how these parameters affect the performance of the generated classifier.

First, we discuss how to set K, i.e. the number of splits. We do this based on numerical experiments on various UCI [3] datasets. We set K to 2, 3, 4, 5, and then compare the resulting performances of DRMF. The ratio of bootstrap samples was set to 0.75, the number of candidate base classifiers was 100 and J48 decision trees from the WEKA library [27] were used as base classifiers.

Table 1 describes the 20 classification tasks we used. For every classification task, standard 10-fold cross validation is performed.

In Table 2, we report the classification performance of DRMF with different random splits. As we can see from there, K has some influence on the performance of the generated ensembles. Since random splits of the features lead to different rotations, diverse base classifiers are generated. However, if K is too large, the number of features in each subset may become too small and hence not sufficient to effectively represent the learning task, thus leading to a drop in performance. The experimental results in Table 2 indicate that DRMF produces good performance if $K = 2$, and we consequently set $K = 2$ in the remainder of experiments.

In DRMF, the number of candidate base classifiers should be set before training. Hence, next we investigate how many candidate base classifiers are sufficient to lead to a good ensemble. We thus perform experiments varying the number of base classifiers L as 20, 40, 60, 80 and 100, respectively. Table 3 compares the classification performance of DRMF with these settings for L. As we can see from there, the overall performance improves when L becomes larger. However, if $L \geqslant 80$, the difference is not significant, and we consequently use $L = 100$ in the following experiments.

In Margin Based Pruning (Algorithm 2), we utilize simple voting in lines 5 and 9. That is, the class that receives the largest number of votes is considered as the final decision. In contrast, in weighted voting, the votes are weighted and the ensemble decision is the class with the largest sum of weights of votes. Since we calculate the weights in line 2 of Algorithm 2, a natural question that arises is whether applying a weighted voting strategy would give better results. To answer this, we give, in Table 4, the classification accuracies and the number of selected base classifiers[2] using the two

[2] Different fusion strategies will lead to different ensemble sizes.

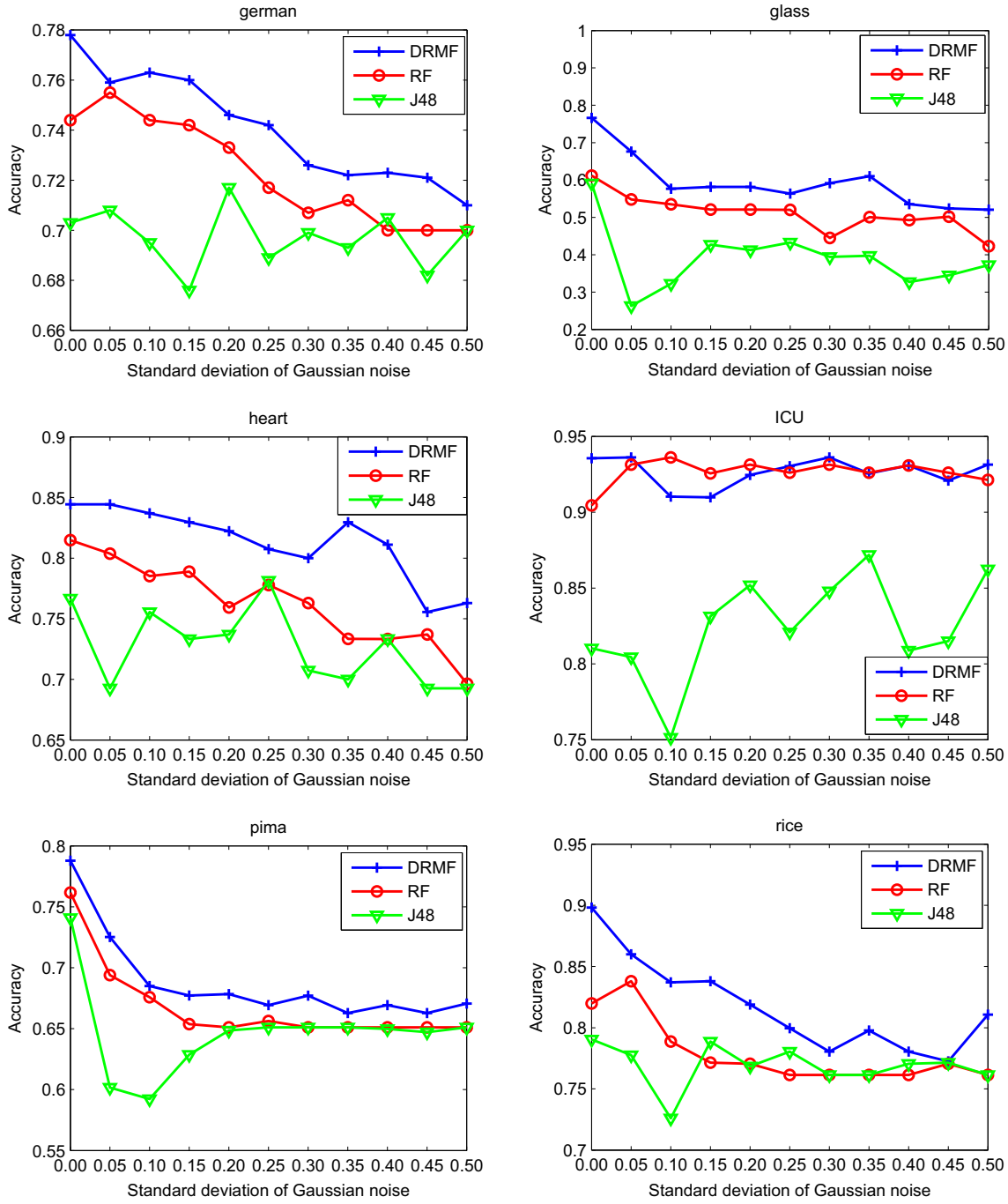

Fig. 4. Variation of classification accuracies when varying the rate of attribute noise.

fusion strategies. From Table 4 we can observe that simple voting performs typically better than weighted voting, while it also leads to smaller ensembles.

To further investigate why simple voting is better than weighted voting, we perform some additional experiments to explore the relationship between the weights of the base classifiers and their classification performances. Fig. 1 shows the relationship between the classification accuracy and the ranking of their weights. Here, the x-axis indicates the ranking of the weight with 1 indicating the largest weight of the base classifier and 100 the smallest. From Fig. 1 we can conclude that base classifiers with large weights do not necessarily have better classification performance compared to those with small weights. Weight learning considers both the diversity among base classifiers and their performances and hence while the weights can be used to rank the base classifiers, they do not reflect their classification performances.

In our algorithm, the base classifiers are added to the ensemble one by one. Base classifiers with the large weights are included first. Fig. 2 plots the classification accuracies when the base classifiers are added, for both simple and weighted voting. From there, we can notice that the best classification accuracy obtained by simple voting is often higher than that of weighted voting, confirming that simple voting can produce better performance than weighted voting. We can also see that fusion based on only a subset of base classifiers is better than combining all of them, which is consistent with the findings from [55].

Table 6
Performance comparison with other classifiers.

Data set	DRMF	J48	Bagging	AdaBoostM1	Rotation Forest
Australian	**88.11 ± 3.48**	83.05 ± 5.40	86.55 ± 6.07	86.10 ± 5.01	85.83 ± 5.09
Crx	**86.37 ± 13.66**	82.74 ± 13.38	83.91 ± 15.48	85.21 ± 12.94	83.04 ± 17.89
Cmc	54.24 ± 3.25	54.65 ± 2.65	53.84 ± 3.32	50.24 ± 2.49	**54.72 ± 2.62**
Derm	96.75 ± 3.84	93.73 ± 4.50	96.39 ± 3.72	95.12 ± 3.30	**97.10 ± 3.19**
German	**77.80 ± 2.94**	70.30 ± 3.40	75.60 ± 3.17	76.00 ± 3.53	74.40 ± 4.79
Glass	**76.64 ± 10.61**	59.11 ± 13.53	68.17 ± 11.17	72.89 ± 17.04	61.17 ± 11.68
Heart	**84.44 ± 4.88**	76.67 ± 5.25	82.22 ± 7.77	80.00 ± 5.84	81.48 ± 6.98
Horse	93.49 ± 3.83	96.19 ± 2.65	97.27 ± 2.27	**97.56 ± 0.86**	91.02 ± 4.64
ICU	**93.56 ± 4.80**	81.03 ± 29.12	84.14 ± 29.72	84.14 ± 29.83	90.45 ± 12.03
Iono	93.47 ± 4.76	89.24 ± 7.90	91.21 ± 5.83	93.22 ± 4.62	**94.34 ± 4.94**
Iris	**98.67 ± 2.81**	96.00 ± 3.44	94.67 ± 6.13	96.00 ± 3.44	95.33 ± 3.22
Movement	**82.44 ± 16.29**	62.44 ± 16.89	68.89 ± 17.73	74.11 ± 18.82	82.00 ± 20.92
Pima	**78.78 ± 3.76**	74.09 ± 5.87	76.43 ± 4.89	73.57 ± 3.74	76.17 ± 3.39
Rice	**89.82 ± 13.17**	79.05 ± 10.04	83.89 ± 8.78	83.89 ± 8.78	81.98 ± 8.95
Spectf	**82.48 ± 7.91**	73.47 ± 8.47	78.78 ± 10.74	78.46 ± 9.85	80.26 ± 6.15
Thyroid	**96.26 ± 4.86**	93.48 ± 5.93	94.87 ± 6.43	94.00 ± 10.12	95.78 ± 7.25
Wiscon	**97.86 ± 2.36**	94.57 ± 2.10	96.43 ± 3.03	95.71 ± 3.01	96.57 ± 2.87
Wdbc	**97.72 ± 1.66**	92.98 ± 3.96	96.31 ± 3.36	97.19 ± 1.90	97.19 ± 2.22
Yeast	73.25 ± 3.47	74.53 ± 4.44	**76.68 ± 5.55**	73.52 ± 3.59	71.89 ± 3.88
Zoo	94.39 ± 8.39	90.76 ± 10.26	93.30 ± 7.07	**96.38 ± 5.75**	90.65 ± 9.13

The best result for each data set is highlighted in bold face.

Table 7
Classification performances of RF with different numbers of splits.

Data set	$K = 2$	$K = 3$	$K = 4$	$K = 5$
Australian	85.83 ± 5.09	85.80 ± 3.83	**86.09 ± 3.53**	84.34 ± 3.62
Crx	83.04 ± 17.89	82.18 ± 17.54	82.89 ± 15.68	**83.75 ± 16.00**
Cmc	**54.72 ± 2.62**	52.68 ± 2.60	53.02 ± 4.37	52.89 ± 2.65
Derm	97.10 ± 3.19	**97.78 ± 2.87**	97.26 ± 2.93	97.02 ± 2.99
German	74.40 ± 4.79	**75.30 ± 3.97**	75.10 ± 5.09	74.80 ± 4.87
Glass	61.17 ± 11.68	66.33 ± 10.64	**71.42 ± 13.53**	60.22 ± 15.34
Heart	81.48 ± 6.98	**84.81 ± 4.08**	83.33 ± 6.11	77.41 ± 9.31
Horse	91.02 ± 4.64	91.84 ± 3.66	91.56 ± 3.77	**92.66 ± 3.39**
ICU	**90.45 ± 12.03**	**90.45 ± 13.92**	87.29 ± 17.86	87.29 ± 21.75
Iono	**94.34 ± 4.94**	93.50 ± 4.95	94.06 ± 5.21	93.50 ± 4.95
Iris	95.33 ± 3.22	94.00 ± 4.92	**96.00 ± 3.44**	**96.00 ± 3.44**
Movement	**82.00 ± 20.92**	80.89 ± 21.25	78.11 ± 23.44	80.33 ± 20.03
Pima	**76.17 ± 3.39**	75.13 ± 5.20	74.74 ± 5.04	73.57 ± 3.65
Rice	**81.98 ± 8.95**	79.05 ± 10.04	79.96 ± 10.70	79.05 ± 10.04
Spectf	80.26 ± 6.15	**80.99 ± 7.46**	78.86 ± 7.67	81.01 ± 5.52
Thyroid	**95.78 ± 7.25**	93.48 ± 7.92	93.96 ± 8.17	93.48 ± 5.93
Wiscon	96.57 ± 2.87	**97.43 ± 2.92**	96.00 ± 3.29	92.99 ± 3.71
Wdbc	97.19 ± 2.22	97.38 ± 2.64	**97.73 ± 2.33**	97.02 ± 2.62
Yeast	**71.89 ± 3.88**	70.55 ± 5.80	70.68 ± 5.80	70.68 ± 6.05
Zoo	90.65 ± 9.13	90.28 ± 8.34	88.65 ± 9.04	**92.39 ± 9.24**
Average	84.07	83.99	83.84	83.02

The best result for each data set is highlighted in bold face.

Margin Based Pruning has two main components. One is to learn the weights for the base classifiers via the minimization of J_W, while the other is to select the base classifiers. Some of the weights might be zero, and we consequently tested whether the classification performance is affected if we remove base classifier with zero weights. For this, in Table 5, we compare the fusion accuracy of base classifiers that receive non-zero weights (denoted by LASSO in Table 5) with the fusion accuracy of DRMF. It can be seen that DRMF performs better than LASSO, and thus that Steps 3–8 in Algorithm 2 are indeed useful for improving the classification performance. We also analyze the differences between the base classifiers selected by the two strategies. From Table 5 it is apparent that the average accuracy of the base classifiers in DRMF is higher than that of the base classifiers that receive non-zero weights, and the diversity among base classifiers in DRMF is also higher than that of the base classifiers with non-zero weights.

Finally, we compare the robustness of DRMF with that of Rotation Forest and J48. For that, we first generate noisy samples by randomly revising the labels of some training samples with the percentage of relabeled samples varying from 3% to 30%. Fig. 3 shows the variation of classification accuracies when we increase the noise rate in the training set. As is apparent, DRMF shows superior robustness compared to both Rotation Forest and J48. In particular, for DRMF the variation of the classification accuracies remains small as the rate of mislabeled samples increases.

We further consider the robustness of the algorithms with respect to attribute noise and add Gaussian noise to the features of the training data. The mean of the noise is zero, while we vary the standard deviation from 0 to 0.5, and show the results in Fig. 4. As we can observe from there, DRMF is more robust with respect to attribute noise than J48, and performs similarly compared to Rotation Forest.

5. Simulation and experimental analysis

In this section, we compare DRMF with some other representative classification algorithms including J48, Rotation Forest, AdaBoostM1,and Bagging, on the 20 datasets from Table 1. For all ensembles, base classifiers are generated using J48. For DRMF and Rotation Forest, we set $K = 2$, the rate of bootstrap sampling to 0.75 and the number L of base classifiers to 100. The parameters of Bagging and AdaBoostM1 were kept as their default values in WEKA, while the number of base classifiers was also set to 100. We performed standard 10-fold cross validation to compute the classification performance. Table 6 gives the classification accuracies on all datasets together with the standard deviations for all evaluated algorithms.

We further employ a test for statistical significance, namely the Nemenyi test [37], to compare the algorithms. In this test, the critical difference [16] for the five algorithms and 20 data sets at significance level $\alpha = 0.05$ is

$$CD = q_{0.05}\sqrt{\frac{k(k+1)}{6\mathbb{N}}} = 2.728 \times \sqrt{\frac{5 \times (5+1)}{6 \times 20}} = 1.364, \tag{18}$$

where $q_{0.05}$ is the critical value for the two-tailed Nemenyi test, k is the number of the algorithms and $\mathbb{N}$ is the number of data sets.

The average ranks for DRMF, J48, Bagging, AdaBoostM1 and Rotation Forest were thus found to be 1.55, 4.36, 3.00, 3.05, and 3.03, respectively, and the average rank differences between DRMF and the other methods were $4.375 - 1.55 = 2.825 > 1.364$, $3.00 - 1.55 = 1.45 > 1.364$, $3.05 - 1.55 = 1.50 > 1.364$, and

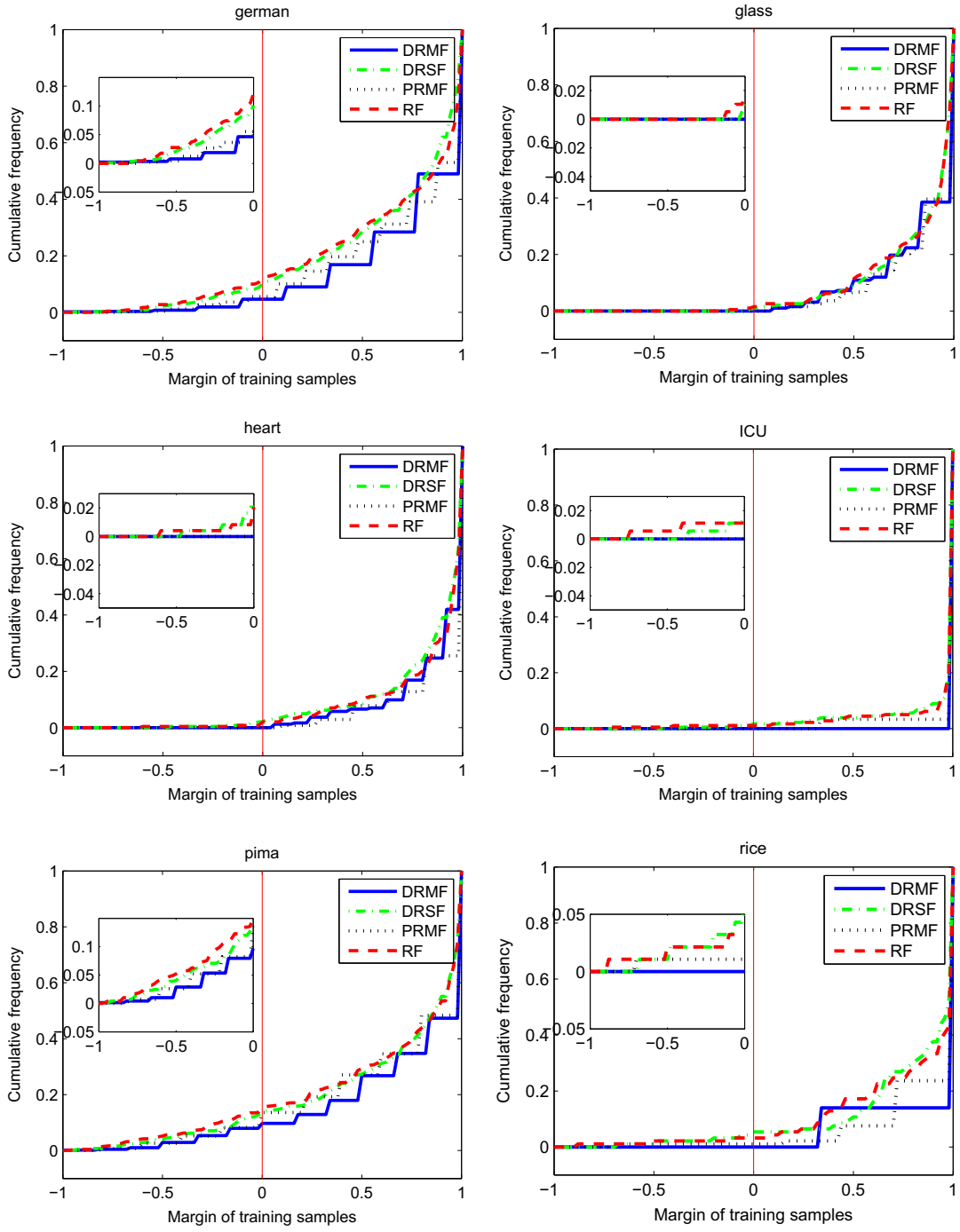

Fig. 5. Margin cumulative frequency of training samples for RF, DRSF, PRMF and DRMF.

$3.025 - 1.55 = 1.475 > 1.364$. Consequently, DRMF was shown to perform statistically significantly better than all other methods.

Next, we discuss why, compared with Rotation Forest, DRMF is able to further boost the classification performance. First, it can be seen that the parameter K in DRMF and Rotation Forest was set to 2 in the above experiments. From Table 2 we know, that $K = 2$ is suitable for DRMF. In order to verify whether $K = 2$ is also suitable for Rotation Forest, the classification performances of Rotation Forest with different numbers of splits are given in Table 7. From there, we can see that Rotation Forest also produces good performance if $K = 2$.

The difference between DRMF and Rotation Forest mainly comprises two parts: the use of Double Rotation to generate the base classifiers, and Margin Based Pruning. Thus, we explore whether they are both necessary for improving the classification performance of the ensemble. For this, we test four combinations: Rotation Forest (RF); a combination of Double Rotation and the fusion strategy in Rotation Forest i.e. simple voting of all base classifiers (DRSF), a combination of PCA rotation and Margin Based Pruning (PRMF), and DRMF. The results are presented in Table 8. As we can confirm from there, both Double Rotation and Margin Based Pruning are indeed useful for improving the classification performance.

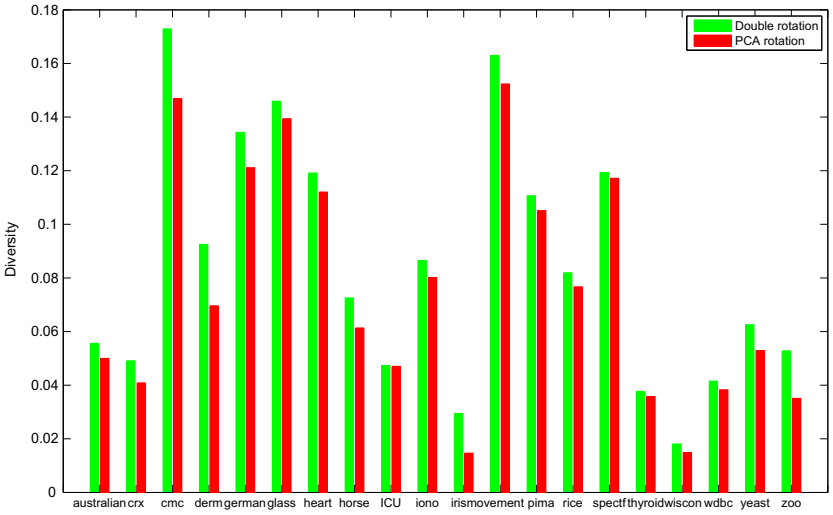

Fig. 6. Diversity of Double Rotation and PCA Rotation.

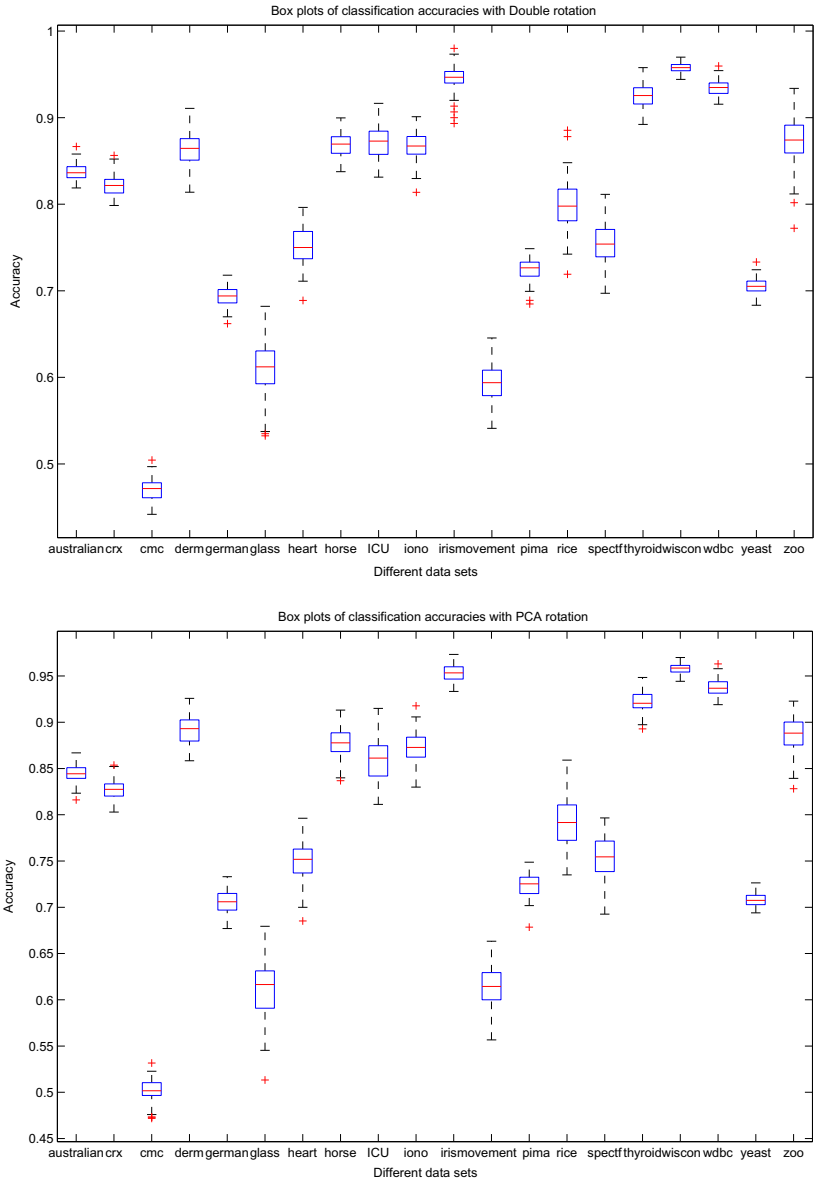

Fig. 7. Box plots of classification accuracies with different rotation strategies.

Further, we compute the margin distribution of the ensembles generated using the above four classification algorithms, where the margin of a sample is computed as the difference between the number of correct votes and the maximum number of the votes received by any wrong label [42]. A large margin is understood as a "confident" classification, and thus we would desire that large margins of the training samples are derived. Fig. 5 shows the margin distribution when RF, DRSF, PRMF and DRMF are used. We can observe, that compared with RF, DRMF improves the margin distribution on the training set, which confirms that both Double Rotation and Margin Based Pruning are helpful for improving the margin distribution.

So, why does Double Rotation improve the margin distribution of the training set? From Theorem 1, we know that the margin of the training samples has an underlying relationship with the diversity of the base classifiers. We thus compare the diversity of the base classifiers when employing Double Rotation and PCA rotation, and show the results in Fig. 6. From there we can notice that the diversity among base classifiers is consistently enhanced after Double Rotation. However, from Fig. 7 we see that the average accuracies of the two kinds of base classifiers are almost the same. Consequently, we can derive that the improvements of the margin distribution come from the diversity, and not from the improved accuracies of the base classifiers.

Finally, we conducted some experiments to validate the effectiveness of the proposed Margin Based Pruning (Margin-P) algorithm and compared it with other ensemble pruning techniques based on the margin including MeanD-M [52] and the improved version of MDM [32,33]. MeanD-M optimizes the average margin via a backward elimination strategy. In particular, it ranks the contribution and importance of every base classifier $\mathcal{C}_j$ in the temporary ensemble Γ by observing the decrease of the average margin when removing $\mathcal{C}_j$ from the ensemble. During each step, the least important classifier $\mathcal{C}_{min}$ with the minimum decrease of the average margin is eliminated from the ensemble and the ensemble thus shrinks to its subset $\Gamma' = \Gamma \setminus \mathcal{C}_{min}$. Then, the base classifiers in Γ' are reordered and the above process is repeated. MDM selects base classifiers via a forward selection strategy where base classifiers are sequentially added based on a specified rule. In particular, the classifier selected in the u-th iteration is

$$s_u = \arg\min_j \mathrm{d}\left(\mathbf{o}, \frac{1}{u}\left(\mathbf{c}_j + \sum_{t=1}^{u-1}\mathbf{c}_{s_t}\right)\right), \tag{19}$$

where $\mathbf{c}_j$ is the N-dimensional signature vector of $\mathcal{C}_j$ whose i-th component $(\mathbf{c}_j)_i$ is 1 if the sample x_i is correctly classified by $\mathcal{C}_j$ and is -1 otherwise. j runs through the classifiers outside the temporary ensemble and $\mathrm{d}(\mathbf{v}_1, \mathbf{v}_2)$ is the distance between vectors $\mathbf{v}_1$ and $\mathbf{v}_2$. In [33], the objective point $\mathbf{o}$ is placed in the first quadrant with equal components $\mathbf{o}_i = p$ (e.g., $p = 0.075$). An improved version of MDM is proposed [32], which uses a moving objective point $\mathbf{o}$ that allows $p(u)$ to vary with the size of the sub-ensemble u. Exploratory experiments show that a value $p(u) \propto \sqrt{u}$ is appropriate. Here the improved version is used for comparison with our method.

In MeanD-M, base classifiers are eliminated from the original ensemble one by one and the sub-ensemble with the best accuracy on the test set is used to estimate its performance. Thus, we also use the best accuracy on the test set to estimate the classification performance for MDM and Margin-P. The results of the experiment are given in Table 9. From there, we can confirm that our algorithm does indeed performs better than the pruning strategies in most cases.

Table 8
Classification performances with different rotation and fusion strategies.

Data set	RF	DRSF	PRMF	DRMF
Australian	85.83 ± 5.09	86.09 ± 3.88	88.02 ± 3.95	**88.11 ± 3.48**
Crx	83.04 ± 17.89	83.61 ± 18.39	85.94 ± 14.06	**86.37 ± 13.66**
Cmc	54.72 ± 2.62	52.41 ± 3.70	**55.67 ± 2.15**	54.24 ± 3.25
Derm	97.10 ± 3.19	**97.86 ± 2.45**	97.26 ± 2.62	96.75 ± 3.84
German	74.40 ± 4.79	75.40 ± 3.17	76.50 ± 5.10	**77.80 ± 2.94**
Glass	61.17 ± 11.68	58.81 ± 11.78	75.28 ± 8.38	**76.64 ± 10.61**
Heart	81.48 ± 6.98	82.59 ± 4.95	83.70 ± 5.58	**84.44 ± 4.88**
Horse	91.02 ± 4.64	91.04 ± 4.76	92.67 ± 4.42	**93.49 ± 3.83**
ICU	90.45 ± 12.03	91.55 ± 3.34	91.98 ± 11.08	**93.56 ± 4.80**
Iono	94.34 ± 4.94	**95.73 ± 4.51**	95.16 ± 2.74	93.47 ± 4.76
Iris	95.33 ± 3.22	95.33 ± 6.32	96.00 ± 3.44	**98.67 ± 2.81**
Movement	82.00 ± 20.92	80.44 ± 16.59	**82.89 ± 18.13**	82.44 ± 16.29
Pima	76.17 ± 3.39	76.17 ± 4.57	76.05 ± 3.99	**78.78 ± 3.76**
Rice	81.98 ± 8.95	83.89 ± 8.78	87.82 ± 10.99	**89.82 ± 13.17**
Spectf	80.26 ± 6.15	81.00 ± 6.09	82.39 ± 5.96	**82.48 ± 7.91**
Thyroid	95.78 ± 7.25	95.78 ± 4.16	95.76 ± 6.12	**96.26 ± 4.86**
Wiscon	96.57 ± 2.87	97.28 ± 2.65	97.28 ± 2.47	**97.86 ± 2.36**
Wdbc	97.19 ± 2.22	97.55 ± 2.05	97.18 ± 3.04	**97.72 ± 1.66**
Yeast	71.89 ± 3.88	71.62 ± 4.32	73.11 ± 3.18	**73.25 ± 3.47**
Zoo	90.65 ± 9.13	93.76 ± 10.06	92.28 ± 8.00	**94.39 ± 8.39**
Average	84.07	84.40	86.15	86.83

The best result for each data set is highlighted in bold face.

Table 9
The best test performances based on different pruning methods.

Data set	MeanD-M	MDM	Margin-P
Australian	88.41 ± 3.48	87.54 ± 2.99	**88.98 ± 3.16**
Crx	86.22 ± 15.11	87.23 ± 14.20	**87.81 ± 13.71**
Cmc	55.94 ± 2.97	56.08 ± 4.00	**56.62 ± 3.48**
Derm	**98.41 ± 2.21**	98.13 ± 2.16	**98.41 ± 2.21**
German	78.60 ± 2.17	79.00 ± 2.49	**79.50 ± 3.06**
Glass	77.60 ± 10.62	75.24 ± 7.89	**79.94 ± 9.89**
Heart	86.30 ± 3.92	85.93 ± 4.55	**87.04 ± 3.15**
Horse	94.59 ± 4.37	93.50 ± 3.64	**94.85 ± 3.70**
ICU	93.61 ± 5.31	94.56 ± 5.06	**94.61 ± 4.29**
Iono	**97.12 ± 3.66**	96.60 ± 3.71	96.84 ± 3.27
Iris	98.00 ± 4.50	97.33 ± 3.44	**98.67 ± 2.81**
Movement	85.33 ± 16.04	83.67 ± 16.06	**85.44 ± 15.87**
Pima	78.91 ± 3.15	79.43 ± 4.10	**80.86 ± 3.39**
Rice	91.55 ± 8.24	91.34 ± 8.29	**94.36 ± 6.48**
Spectf	85.86 ± 4.25	85.11 ± 7.27	**86.62 ± 5.56**
Thyroid	97.19 ± 4.58	96.69 ± 4.51	**97.64 ± 4.62**
Wiscon	98.00 ± 2.25	**98.43 ± 1.96**	98.14 ± 2.13
Wdbc	**98.60 ± 1.38**	98.25 ± 1.42	98.42 ± 1.29
Yeast	73.92 ± 3.41	74.12 ± 4.08	**74.46 ± 2.32**
Zoo	**95.39 ± 8.41**	**95.39 ± 8.41**	**95.39 ± 8.41**
Average	87.98	87.68	88.73

The best result for each data set is highlighted in bold face.

6. Conclusions and future work

Ensemble learning is an effective approach to improve the generalization performance of a classification system. In this paper, we have proposed Double Rotation Margin Forest (DRMF) as an effective new ensemble learning algorithm. The idea of DRMF is to improve the generalization performance by improving the margin distribution on the training set. Extensive experimental results on 20 benchmark datasets confirm that DRMF provides a competent ensemble learner, and allows us to draw several conclusions: (1) Double Rotation with PCA and LSDA is able to generate diverse base classifiers; (2) The margin distribution of the ensemble system is improved if a set of diverse base classifiers is exploited by optimizing a regularized loss function, and consequently the classification performance of the ensemble is enhanced; and (3) The DRMF algorithm outperforms classical ensemble learning techniques such as Bagging, AdaBoostM1 and Rotation Forest.

Our work is motivated by the idea that diversity among base classifiers can improve the margin distribution of the ensemble. However, no deep discussion on this issue has been reported so far. Further theoretical analysis on the relationship between diversity and margin is thus required. While in this paper, we use Double Rotation to create diversity, some other techniques could also be introduced. A systematic discussion on generating diversity would therefore present an important task. Although DRMF is presented as an approach to create homogenous ensembles (e.g., based only on decision trees as the base classifiers), it is straightforward to use DRMF to learn and prune heterogeneous classifiers for ensemble learning. Exploring the effectiveness of DRMF in such a setting might be an interesting avenue.

Acknowledgments

This work is supported by the National Program on Key Basic Research Project under Grant 2013CB329304, National Natural Science Foundation of China under Grants 61222210, 61170107, 61073125, 61350004 and 11078010, the Program for New Century Excellent Talents in University (No. NCET-12-0399), and the Fundamental Research Funds for the Central Universities (Grant No. HIT.NSRIF.2013091 and HIT.HSS.201407).

References

[1] M. Aksela, J. Laaksonen, Using diversity of errors for selecting members of a committee classifier, Pattern Recogn. 39 (2006) 608–623.
[2] F. Aiolli, A. Sperduti, A re-weighting strategy for improving margins, Artif. Intell. 137 (2002) 197–216.
[3] C. Blake, E. Keogh, C.J. Merz, UCI Repository of Machine Learning Databases. Dept. Inf. Comput. Sci., Univ. California, Irvine, CA, 1998. <http://www.ics.uci.edu/mlearn/MLRepository.html>.
[4] L. Breiman, Bagging predictors, Mach. Learn. 24 (1996) 123–140.
[5] L. Breiman, Stacked regressions, Mach. Learn. 24 (1996) 49–64.
[6] P.L. Bartlett, For valid generalization, the size of the weights is more important than the size of the network, in: Advances inNeural Information Processing Systems, vol. 9, 1997.
[7] R. Bryll, R. Gutierrez-Osuna, F. Quek, Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets, Pattern Recogn. 36 (2003) 1291–1302.
[8] B.E. Boser, I.M. Guyon, V.N. Vapnik, A training algorithm for optimal margin classifiers, in: Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory, 1992, pp. 144–152.
[9] 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.
[10] K.J. Cherkauer, Human expert level performance on a scientific image analysis task by a system using combined artificial neural networks, in: Proceedings of 13th AAAI Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Algorithms, 1996, pp. 15–21.
[11] K. Crammer, R. Gilad-Bachrach, A. Navot, A. Tishby, Margin analysis of the LVQ algorithm, Adv. Neural Inf. Process. Syst. 15 (2003) 462–469.
[12] D. Cai, X.F. He, K. Zhou, J.W. Han, H.J. Bao, Locality Sensitive Discriminant Analysis, International Joint Conference on Artificial Intelligence, 2007, pp. 708–713.
[13] H. Chen, P. Tino, X. Yao, Predictive ensemble pruning by expectation propagation, IEEE Trans. Knowl. Data Eng. 21 (7) (2009) 999–1013.
[14] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[15] Q. Dai, A competitive ensemble pruning approach based on cross-validation technique, Knowl.-Based Syst. 37 (2013) 394–414.
[16] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (1) (2006) 1–30.
[17] Y. Freund, Boosting a weak learning algorithm by majority, Inf. Comput. 121 (1996) 256–285.
[18] S. Floyd, M. Marmuth, Sample compression learnability, and the VapnikC Chervonenkis dimension, Mach. Learn. 21 (3) (1995) 269–304.
[19] G. Fumera, F. Roli, A theoretical and experiment alanalysis of linear combiners for multiple classifier systems, IEEE Trans. Pattern Anal. Mach. Intell. 27 (6) (2005) 942–956.
[20] Y. Freund, R.E. Schapire, A decision-theoretic generalization of online learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.
[21] Y. Freund, R.E. Schapire, Large Margin Classification Using the Perceptron Algorithm, Mach. Learn. 37 (3) (1999) 277–296.
[22] T. Graepel, R. Herbrich, J. Shawe-Taylor, Generalization error bounds for sparse linear classifiers, in: 13th Annual Conference on Computational Learning Theory, 2000, pp. 298–303.
[23] A.J. Grove, D. Schuurmans, Boosting in the limit: maximizing the margin of learned ensembles, in: Proceedings of the 15th National Conference on Artificial Intelligence, American Association for Artificial Intelligence, Menlo Park, CA, USA, 1998, pp. 692–699.
[24] D. Hernández-Lobato, G. Martínez-Muñoz, A. Suárez, Empirical analysis and evaluation of approximate techniques for pruning regression bagging ensembles, Neurocomputing 74 (12–13) (2011) 2250–2264.
[25] Q.H. Hu, D.R. Yu, M.Y. Wang, Constructing rough decision forests, D. Slezak et al. (Eds.), RSFDGrC 2005, Lect. Notes Artif. Intell. 3642 (2005) 147–156.
[26] Q.H. Hu, P.F. Zhu, Y.B. Yang, D.R. Yu, Large-margin nearest neighbor classifiers via sample weight learning, Neurocomputing 74 (2011) 656–660.
[27] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, SIGKDD Explor. 11 (1) (2009) 10–18.
[28] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas, On combining classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 20 (3) (1998) 226–239.
[29] L.I. Kuncheva, Switching between selection and fusion in combining classifiers: an experiment, IEEE Trans. Syst. Man Cybern. Part-B: Cybern. 32 (2) (2002) 146–156.
[30] L. Kuncheva, C. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, Mach. Learn. 51 (2003) 181–207.
[31] D.D. Margineantu, T.G. Dietterich, Pruning adaptive boosting, in: Proceedings of the 14th International Conference on Machine Learning, Morgan Kaufman Publishers Inc., San Francisco, CA, USA, 1997, pp. 211–218.
[32] 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.
[33] G. Martínez-Muñoz, A. Suárez, Aggregation ordering in bagging, in: Proceedings of the International Conference on Artificial Intelligence and Applications, 2004, pp. 258–263.
[34] G. Martínez-Muñoz, A. Suárez, Pruning in ordered bagging ensembles, in: Proceedings of the 23th International Conference on Machine Learning, 2006, pp. 609–616.
[35] G. Martínez-Muñoz, A. Suárez, Using boosting to prune bagging ensembles, Pattern Recogn. Lett. 28 (1) (2007) 156–165.
[36] R. Maclin, J.W. Shavlik, Combining the predictions of multiple classifiers: using competitive learning to initialize neural networks, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence, Morgan Kaufman, San Mateo, CA, 1995, pp. 524–530.
[37] P.B. Nemenyi, Distribution-Free Multiple Comparisons, PhD Thesis, Princeton University, 1963.
[38] A. Rahman, B. Verma, Ensemble classifier generation using non-uniform layered clustering and Genetic Algorithm, Knowl.-Based Syst. 43 (2013) 30–42.
[39] J.J. Rodríguez, L.I. Kuncheva, Rotation forest: a new classifier ensemble method, IEEE Trans. Pattern Anal. Mach. Intell. 28 (10) (2006) 1619–1630.
[40] S. Rosset, Z. Ji, T. Hastie, Boosting as a regularized path to a maximum margin classifier, J. Mach. Learn. Res. 5 (2004) 941–973.
[41] R.E. Schapire, The strength of weak learnability, Mach. Learn. 5 (1990) 197–227.
[42] R.E. Schapire, Y. Freund, P. Bartlett, W.S. Lee, Boosting the margin: A new explanation for the effectiveness of voting methods, Ann. Statist. 26 (5) (1998) 1651–1686.
[43] 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.
[44] J. Shawe-Taylor, N. Cristianini, Margin Distribution Bounds on Generalization, EuroCOLT 1999, pp. 263–273.
[45] C.H. Shen, H.X. Li, Boosting through optimization of margin distributions, IEEE Trans. Neural Networks 21 (4) (2010) 659–666.
[46] E.K. Tang, P.N. Suganthan, X. Yao, An analysis of diversity measures, Mach. Learn. 65 (2006) 247–271.
[47] R. Tibshirani, Regression shrinkage and selection via the lasso, J. Royal. Statist. Soc B. 58 (1) (1996) 267–288.
[48] N. Ueda, Optimal linear combination of neural networks for improving classification performance, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2) (2000) 207–215.
[49] V.N. Vapnik, Estimation of Dependences Based on Empirical Data, Springer-Verlag, New York, 1982.
[50] K. Weinberger, J. Blitzer, L. Saul, Distance metric learning for large margin nearest neighbor classification, Adv. Neural Inf. Process. Syst. (NIPS) 18 (2006) 1473–1480.
[51] L.W. Wang, M. Sugiyama, Z.X. Jing, C. Yang, Z.H. Zhou, J.F. Feng, A refined margin analysis for boosting algorithms via equilibrium margin, J. Mach. Learn. Res. 12 (2011) 1835–1863.
[52] F. Yang, W.H. Lu, L.K. Luo, T. Li, Margin optimization based pruning for random forest, Neurocomputing 94 (2012) 54–63.
[53] X. Yao, Y. Liu, Making use of population information in evolutionary artificial neural networks, IEEE Trans. Syst. Man Cybern., Part-B: Cybern. 28 (3) (1998) 417–425.
[54] Y. Zhang, S. Burer, W.N. Street, Ensemble pruning via semi-definite programming, J. Mach. Learn. Res. 7 (2006) 1315–1338.
[55] 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.
[56] Z.H. Zhou, Y. Yu, Ensembling local learners through multimodal perturbation, IEEE Trans. Syst. Man Cybern. Part-B: Cybern. 35 (4) (2005) 725–735.

[57] L. Zhang, W.D. Zhou, Sparse ensembles using weighted combination methods based on linear programming, Pattern Recogn. 44 (2011) 97–106.
[58] L. Zhang, W.D. Zhou, On the sparseness of 1-norm support vector machines, Neural Networks 23 (3) (2010) 373–385.
[59] X.Q. Zhu, P. Zhang, X.D. Lin, Y. Shi, Active learning from stream data using optimal weight classifier ensemble, IEEE Trans. Syst. Man Cybern. Part-B: Cybern. 40 (6) (2010) 1607–1621.