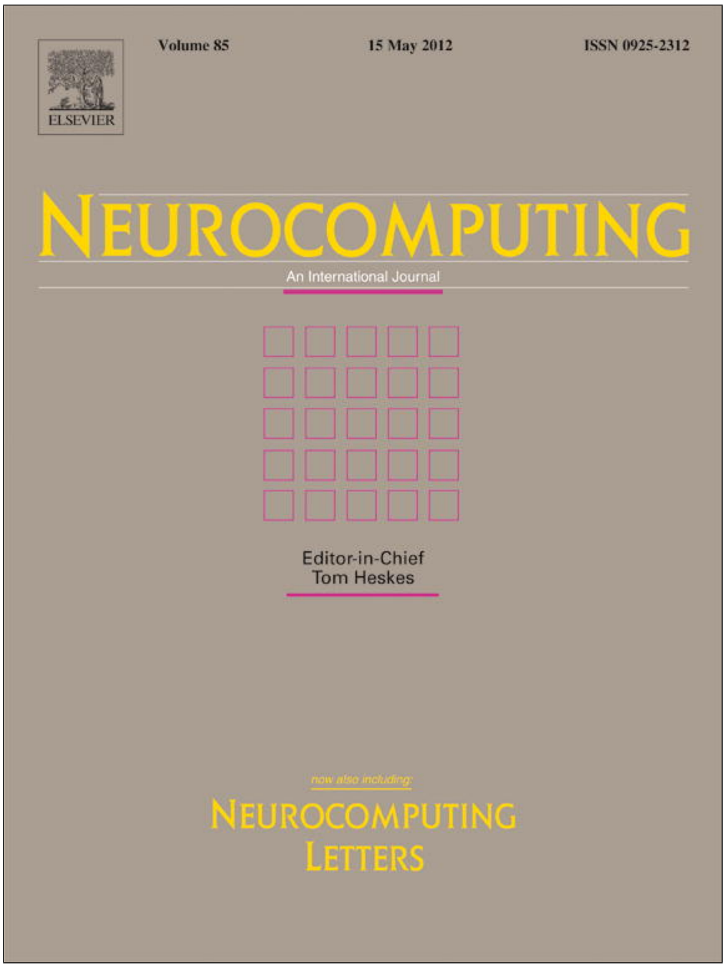
Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



Volume 85 15 May 2012 ISSN 0925-2312

ELSEVIER

NEUROCOMPUTING

An International Journal

Editor-in-Chief
Tom Heskes

now also including:

NEUROCOMPUTING
LETTERS

This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/copyright

Author's personal copy

Neurocomputing 85 (2012) 11–19

Contents lists available at SciVerse ScienceDirect

Neurocomputing

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

Margin distribution based bagging pruning

Zongxia Xie [a,*], Yong Xu [a], Qinghua Hu [b], Pengfei Zhu [b]

[a] Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China
[b] Harbin Institute of Technology, Harbin 150001, China

ARTICLE INFO

Article history:
Received 30 May 2011
Received in revised form
24 December 2011
Accepted 27 December 2011
Communicated by D. Wang
Available online 22 February 2012

Keywords:
Classification
Bagging
Margin distribution
Sparse ensemble

ABSTRACT

Bagging is a simple and effective technique for generating an ensemble of classifiers. It is found there are a lot of redundant base classifiers in the original Bagging. We design a pruning approach to bagging for improving its generalization power. The proposed technique introduces the margin distribution based classification loss as the optimization objective and minimizes the loss on training samples, which leads to an optimal margin distribution. Meanwhile, in order to derive a sparse ensemble, l_1 regularization is introduced to control the size of ensembles. By this way, we can obtain a sparse weight vector of base classifiers. Then we rank the base classifiers with respect to their weights and combine the base classifiers with large weights. We call this technique MArgin Distribution base Bagging pruning (MAD-Bagging). Simple voting and weighted voting are tried to combine the outputs of selected base classifiers. The performance of this pruned ensemble is evaluated with several UCI benchmark tasks, where base classifiers are trained with SVM, CART, and the nearest neighbor (1NN) rule, respectively. The results show that margin distribution based CART pruned Bagging can significantly improve classification accuracies. However, SVM and 1NN pruned Bagging improve little compared with single classifiers.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Bagging is one of the most popular methods in constructing classifier ensembles [1]. The technique trains a collection of base classifiers on bootstrap replicates of the training set and combines the outputs of base classifiers with simple voting. The effectiveness of the technique has been empirically verified in many pattern recognition tasks. In general, the error of Bagging becomes smaller as base classifiers aggregated in the ensemble increase [2]. Eventually, the error asymptotically approaches a constant level with a large ensemble size.

In order to get good performance, many base classifiers are usually required in Bagging. Much computational resources are occupied. Both space complexity and time complexity are very high. In fact, majority of base classifiers can be removed from the original ensemble, in the meanwhile the classification performance will not drop. Sparse ensembles were proposed to build such multiple classifier systems [3]. Sparse ensembles mean combining the outputs of base classifiers with a sparse weight vector, where each classifier is assigned a weight value, but only several weights are nonzero. The base classifiers with zero weights are not used in the final decision making. Thus most of the base classifiers are removed from the original ensembles. This technique is also called pruned ensembles or selective ensembles [4–7].

Tamon et al. proved that the problem of selecting the best combination of classifiers from an ensemble was NP-complete [8]. Since some optimization methods, such as GA [9] and semi-definite programming [10], have been introduced for selecting base classifiers with heuristic information. Some suboptimal ensemble pruning methods based on ordered aggregation were proposed, including reduce-error (RE) pruning [11], margin distance minimization (MDM) [4], orientation ordering [5], boosting-based ordering [7], expectation propagation [12], and so on. And the LP-Adaboost method in Yao and Liu [13], the GA-based method in [14], and the WV-LP method in [3] can be considered as sparse ensembles.

In the last decade, it was shown that the generalization performance of a classifier is related with the margin distribution on training samples and the generalization error of a classifier can be reduced by explicitly optimizing the margin distribution [15,16]. Lodhi et al. designed a boosting method to optimize the margin distribution based generalization bound [17]. This technique produced considerable improvements over AdaBoost. In 2010, Shen and Li proposed a margin-distribution boosting algorithm [18], which directly optimizes the margin distribution: maximizing the average margin and at the same time minimizing the variance of the margin distribution. This technique is built on the assumption that margins

[*] Corresponding author.
E-mail address: xiezongxia@hitsz.edu.cn (Z. Xie).

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2011.12.030

Author's personal copy

of samples satisfy Gaussian distribution. However, this assumption does not hold in real-world applications.

The current ensemble pruning methods for Bagging do not consider the margin distribution of ensembles. In this paper, we propose a new sparse ensemble method for Bagging pruning, named as MArgin Distribution based Bagging (MAD-Bagging) pruning. This method is similar to the WV-LP method proposed in [3]. Both of them are focused on the training error of ensembles, instead of the classification performance of base classifiers. Here, we introduce a regularized classification loss function, where the margins of samples in an ensemble is used to compute the classification loss and l_1 regularization is added to the optimization objective for obtaining a sparse weight vector of base classifiers. However, WV-LP uses a weighted combination method to compute the training error and the continuous outputs of individual classifier are required. What is more, WV-LP used multiple feature subsets to generate the ensembles of KNN classifiers. In our work, Bagging is used to build multiple classifiers. We utilize SVM, CART, and 1NN algorithms in training base classifiers. Simple voting [19] and weighted voting [20] are tried in combining the predictions of the selected base classifiers. The objective is to find an optimal weight training technique and an effective approach to exploiting the trained weights.

The rest of this paper is organized as follows. In Section 2, we describe the original Bagging method and some current ensemble pruning techniques. Section 3 presents the framework of MAD-Bagging, including loss functions, the solution to the optimization objective, and the combination rule with the weights. Experimental analysis is presented in Section 4. Finally, conclusions are listed in Section 5.

2. Bagging and related research

Bagging is a popular ensemble method introduced by Breiman in 1996 [1]. The idea of Bagging is simple and appealing: the ensemble consists of base classifiers built on bootstrap replicates of the training set. The outputs of base classifiers are combined with the technique of the plurality voting.

Assume we have a training set $X=\{(x_i,y_i)|x_i \in \Re^{N_f}\}_{i=1}^{N}$ with $y_i \in \{1,2,\ldots,c\}$. N_f is the dimensionality of the sample space, c is the number of classes, and N is the number of training samples. More precisely, the Bagging algorithm can be described as follows.

1. Generate T bootstrap samples of N points $\{X_j\}_{j=1}^{T}$ from X with probability weights $p(i)$. In this paper, we use $p(i)=1/N$.
2. For $j=1,\ldots,T$, train a base classifier h_j on the bootstrap sample X_j.
3. Classify new points using the simple majority vote of the ensemble

$$\hat{y}(x) = \max_{m=1,2,\cdots c} \sum_{j=1}^{T} w_j h_{jm}(x), \tag{1}$$

where $w_j = 1$, and $h_{jm}(x)$ is the output of the j-th classifier for sample x associated with class m.

We can see that the original Bagging method combines the outputs of all classifiers. And the diversity of base classifiers in Bagging is generated by using different training data with the bootstrap method. Bootstrap is a technique for random sampling with replacement. So some objects could be represented in a new set once, twice or even more times and some objects may not be represented at all. Taking a bootstrap replicate of the training sample set, one can avoid or get less ‘outliers’ in the bootstrap training set. By this way the bootstrap estimates of the data distribution parameters are robust [21].

However, Bagging is not always effective. Breiman provided a qualitative description of the learners with which Bagging can be expected to work [1]: they have to be unstable, in the sense that small variations in the training set can lead to produce significantly different models. Decision trees and neural networks are examples of such learners. In contrast, the nearest-neighbor method is stable. Bagging is of little value when applied to stable classifiers. Domingos thought that Bagging worked because it effectively shifted the prior to a more appropriate region of model space [22]. The effectiveness of Bagging was also investigated by Tibshirani [23], and Wolpert and Macready [24], with the bias–variance decomposition to estimate the generalization error.

As to the size of Bagging, only some empirical guidelines have been given. It is well known that the misclassification rate of Bagging tends to an asymptotic value as the ensemble size increases. In 2008, Fumera, Roli and Serrau offered an analytical model of Bagging misclassification probability as a function of the ensemble size and showed preserving a few base classifiers is enough for obtaining good performance [25]. Several other researchers also proposed methods to select base classifiers generated by Bagging, with the aim of improving the ensemble accuracy and reducing its size.

There are six main algorithms for Bagging pruning. In 1997, Margineantu and Dietterich introduced some techniques for ensemble pruning [11], where reduce-error (RE) pruning was considered as a sophisticated algorithm. They proposed a backfitting search strategy, which starts with the best base classifier, and then adds a base classifier such that the voted combination has the lowest error. These two steps are the same as the greedy search. After that, backfitting revisits the selected classifiers one by one and replaces each selected classifier with another candidate for obtaining the best classification performance. Obviously, the time complexity of backfitting is very high.

In 2002, Zhou et al. derived a conclusion that selective ensemble is better than combining all base classifiers and developed a genetic algorithm based selectors GASEN, where the estimated error is used as the optimization objective [9].

In 2004, Martinez-Muoz and Suarez proposed a margin distance minimization algorithm (MDM) [4], where a matrix is defined. The element e_{ij} in the matrix is 1 or -1. If sample x_j is correctly classified by base classifier h_i, $e_{ij}=1$; otherwise, $e_{ij}=-1$. In this case, the mean of the jth column is the classification margin of sample x_j. Obviously, the margin takes values in $[-1, 1]$. If the samples are correctly classified by the ensemble, the margin is larger than zero. If we view the vector of sample margins as a point in N-dimensional space, then the samples are correctly classified when the corresponding points are located in the first quadrant. With this observation, the authors set a point in the first quadrant as an objective point. They selected the base classifier minimizing the distance between the objective point and the margin vectors in each step.

In 2006, Martínez-Muñoz and Suárez introduced a quantity to measure how a classifier maximizes the alignment of a signature vector of the ensemble with a direction that corresponds to perfect classification performance on the training set. Then they used this quantity to sort the base classifiers [5]. In 2007, boosting was introduced to compute ordering of base classifiers [7]. In 2009, Chen et al. designed an expectation propagation algorithm to approximate the posterior estimation of the weight vector and get a sparse weight vector [12].

3. Weight learning based on margin distribution

Much work on learning machines has been devoted to study how to control the generalization performance these years. Schapire et al. [26] gave an upper bound for the generalization

Author's personal copy

error of a voting classifier. This bound does not depend on how many classifiers are combined, but depends on the margin distribution over the training set, the number of the training examples and the VC dimension of base classifiers. This theory indicates that good margin distribution is the key to the success of AdaBoost. Some other results from the theoretical analysis also suggest that good margin distributions lead to good generalization performances [15,17,27–29].

Provided we have a training sample set $X=\{(x_i,y_i)\}_{i=1}^{N}$ with $y_i\in\{-1,1\}$ for a binary classification task. Here, a base classifier h_j is a mapping from X to $\{-1,1\}$. The voted ensemble $f(x)$ is of the form

$$f(x)=\sum_{j=1}^{T}w_jh_j(x)$$

$$\sum_{j=1}^{T}w_j=1,\quad w_j\ge 0,\ j=1,2,\ldots,T, \tag{2}$$

where w_j is the weight assigned to the base classifier h_j and T is the number of base classifiers in the system. An error occurs to sample x_i if and only if the output of voting classifier and the label y_i do not have the same sign, i.e.

$$y_if(x_i)\le 0. \tag{3}$$

Since $h_j\in\{-1,1\}$,

$$y_if(x_i)=\sum_{i:y_i=h_i(x_i)}w_i-\sum_{i:y_i\ne h_i(x_i)}w_i.$$

Hence $y_if(x_i)$ is the difference between the weights assigned to the correct label and the weights assigned to the wrong label. $y_if(x_i)$ is considered as the sample margin ρ_i with respect to the voting classifier f [26]. Obviously, ρ_i takes values in the interval $[-1,1]$. We have

$$\rho_i=y_if(x_i)=y_i\sum_{j=1}^{T}w_jh_j(x_i)=\sum_{j=1}^{T}w_jy_ih_j(x_i). \tag{4}$$

With this definition, we can see if w_j is large, the base classifier h_j contributes much to the margin of samples. So the classifiers with larger weights play a more important role than others. We should select the base classifiers with large weights in selective ensembles.

Note that $y_ih_j(x_i)\in\{-1,1\}$ can reflect whether x_i is correctly classified by classifier h_j. If $y_ih_j(x_i)=1$ x_i is correctly classified while $y_ih_j(x_i)=-1$ x_i is misclassified. $y_ih_j(x_i)$ is the margin with respect to the base classifier d_{ij}. We obtain that the margin ρ on the whole training set is

$$\rho=\begin{bmatrix}\rho_1\\ \rho_2\\ \vdots\\ \rho_N\end{bmatrix}=\begin{bmatrix}d_{11},d_{12},\ldots,d_{1T}\\ d_{21},d_{22},\ldots,d_{2T}\\ \vdots,\quad \vdots,\ddots,\vdots\\ d_{N1},d_{N2},\ldots,d_{NT}\end{bmatrix}\begin{bmatrix}w_1\\ w_2\\ \vdots\\ w_T\end{bmatrix}$$
$$=[D_1,D_2,\ldots,D_j,\ldots,D_T]w=Dw \tag{5}$$

where D_j is the vector of margins with respect to the base classifier h_j on the whole training set.

As to multi-class tasks, $y\in\{1,2,\ldots,c\}$. d_{ij} cannot be computed through $y_ih_j(x_i)$ directly. We define that $d_{ij}=1$ if x_i is correctly classified by the individual classifier h_j; otherwise $d_{ij}=-1$.

In order to obtain better generalization ability, the above voting model $f(x)$ should minimize the loss criterion $\sum_i C(y_if(x_i))$ which is a function of the margin distribution $\rho_i=y_if(x_i)$ of this model on the data. Here, we use the squared hinge loss

$$\sum_i C(y_if(x_i))=\sum_i(1-y_if(x_i))^2=\sum_i(1-\rho_i)^2=\|1-Dw\|^2. \tag{6}$$

The above optimization cannot output sparse weight vectors. The regularization technique can be utilized to control the complexity of the model $f(x)$. Thus, the quantity actually minimized on the data is a regularized version of the loss function:

$$w(\lambda)=\min_w\sum_i C(y_if(x_i))+\lambda\|w\|_p^p$$
$$=\min_w\|1-Dw\|^2+\lambda\|w\|_p^p$$

$$\text{s.t. } w_j\ge 0 \tag{7}$$

Table 1
Data sets.

Number	Data	Samples	Features	Classes
1	Credit	690	15	2
2	German	1000	20	2
3	Glass	214	9	6
4	Heart	270	13	2
5	Hepatitis	155	19	2
6	Horse	368	22	2
7	Iono	351	34	2
8	Sonar	208	60	2
9	Votes	435	16	2
10	WDBC	569	31	2
11	Wine	178	13	3
12	WPBC	198	33	2

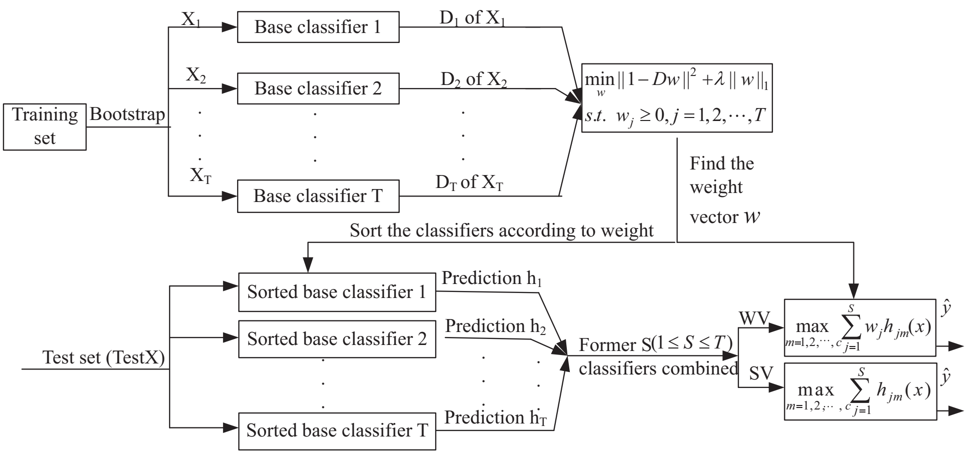


Fig. 1. Framework of MAD-Bagging.

Author's personal copy

Table 2
Number of selected base classifiers in different ensembles.

Data	SVM				CART				1NN			
	RE	MDM	SV	WV	RE	MDM	SV	WV	RE	MDM	SV	WV
Credit	7.0	7.4	2.2	3.1	11.6	11.3	13.3	10.3	3.9	4.3	3.8	3.6
German	15.9	23.0	12.5	7.3	29.7	64.1	20.8	21.0	5.1	3.9	4.8	7.3
Glass	17.0	1.9	6.8	2.8	13.6	9.8	47.5	4.0	3.0	2.5	3.3	3.9
Heart	3.8	3.0	4.0	2.4	13.1	24.6	10.7	5.6	2.8	2.7	2.7	2.4
Hepatitis	20.7	15.9	20.9	1.9	2.2	10.1	4.6	3.3	2.0	2.6	1.5	1.8
Horse	3.1	3.5	16.1	4.2	2.2	14.2	8.3	2.4	2.3	1.8	1.8	1.6
Iono	3.5	3.6	1.8	1.8	14.1	9.5	16.8	3.3	1.6	3.7	1.5	2.6
Sonar	1.9	11.0	4.9	2.3	35.8	32.0	18.7	7.6	4.9	2.8	3.4	3.2
Votes	2.3	2.3	2.3	1.8	12.6	2.0	2.7	4.5	3.0	6.1	4.3	3.4
WDBC	16.7	9.5	10.3	1.4	29.8	11.9	11.8	7.2	2.9	2.5	2.0	4.6
Wine	2.3	1.5	10.6	1.0	15.6	6.0	17.4	4.6	1.0	1.0	1.5	1.5
WPBC	1.0	1.0	1.0	1.0	40.8	46.3	28.0	11.1	4.1	3.6	5.7	7.3
Ave.	7.9	7.0	7.8	2.58	18.4	20.1	16.7	7.1	3.1	3.1	3.0	3.6

Table 3
Classification performance with SVM and its ensembles.

Data	SingleSVM	Bagging	RE	MDM	MAD-SV(λ)	MAD-WV(λ)
Credit	82.46 ± 10.67	82.17 ± 10.88	84.64 ± 10.16	85.21 ± 8.75	$85.51\pm8.32(10)$	$84.93\pm7.95(10)$
German	74.00 ± 3.40	74.20 ± 3.22	75.90 ± 3.41	76.70 ± 3.33	$77.00\pm3.13(10)$	$75.70\pm3.37(10)$
Glass	63.83 ± 14.81	64.22 ± 13.45	70.19 ± 11.21	68.83 ± 11.28	$70.74\pm10.30(10)$	$68.37\pm10.90(10)$
Heart	82.96 ± 6.10	82.96 ± 6.10	85.19 ± 5.52	84.81 ± 5.90	$85.93\pm5.47(0.01)$	$84.44\pm3.83(10)$
Hepatitis	83.83 ± 3.34	84.33 ± 3.87	89.00 ± 6.30	89.67 ± 6.37	$89.00\pm7.04(1)$	$87.33\pm7.98(10)$
Horse	90.49 ± 3.86	90.49 ± 4.08	92.68 ± 3.36	92.95 ± 3.40	$92.94\pm3.64(50)$	$92.39\pm3.78(1)$
Iono	84.99 ± 7.05	85.29 ± 7.11	88.68 ± 5.18	89.23 ± 4.69	$89.54\pm5.04(1)$	$89.24\pm5.01(10)$
Sonar	69.69 ± 9.23	70.14 ± 10.13	83.21 ± 8.59	82.69 ± 6.51	$83.69\pm7.85(50)$	$82.26\pm8.69(50)$
Votes	96.26 ± 3.63	96.03 ± 3.77	97.45 ± 2.33	96.75 ± 2.55	$97.21\pm2.68(1)$	$96.75\pm2.53(1)$
WDBC	95.26 ± 2.75	95.26 ± 2.75	95.61 ± 2.38	95.96 ± 2.49	$96.14\pm2.16(10)$	$95.79\pm2.21(1)$
Wine	98.33 ± 2.68	98.33 ± 2.68	98.89 ± 2.34	98.89 ± 2.34	$99.44\pm1.76(10)$	$98.33\pm2.68(0.01)$
WPBC	76.32 ± 3.04	76.32 ± 3.04	76.32 ± 3.04	76.32 ± 3.04	$76.32\pm3.04(0.01)$	$76.32\pm3.04(0.01)$
Ave.	83.20	83.31	86.48	86.50	86.95	85.99

Table 4
Classification performance with 1NN and its ensembles.

Data	Single1NN	Bagging	RE	MDM	MAD-SV(λ)	MAD-WV(λ)
Credit	79.10 ± 11.62	79.10 ± 11.62	82.59 ± 10.68	81.58 ± 10.86	$82.16\pm11.41(0.01)$	$81.73\pm11.06(0.01)$
German	68.80 ± 3.22	68.80 ± 3.22	72.40 ± 3.10	71.40 ± 2.22	$72.50\pm2.22(1)$	$71.10\pm3.07(10)$
Glass	65.42 ± 12.85	65.42 ± 12.85	70.12 ± 12.95	72.57 ± 12.11	$70.55\pm14.37(50)$	$69.64\pm13.53(50)$
Heart	76.67 ± 9.41	76.67 ± 9.41	80.37 ± 10.04	81.48 ± 10.90	$81.11\pm7.08(100)$	$78.89\pm8.56(100)$
Hepatitis	82.50 ± 7.59	82.50 ± 7.59	85.67 ± 8.61	86.17 ± 6.28	$85.17\pm8.62(10)$	$85.17\pm8.62(10)$
Horse	87.26 ± 4.22	87.26 ± 4.22	88.86 ± 2.37	90.23 ± 4.01	$90.51\pm4.40(50)$	$89.43\pm4.76(50)$
Iono	86.40 ± 4.93	86.40 ± 4.93	88.37 ± 4.78	88.40 ± 5.70	$88.09\pm5.65(100)$	$88.09\pm5.35(1)$
Sonar	87.05 ± 7.56	87.05 ± 7.56	88.98 ± 5.10	89.00 ± 6.36	$90.40\pm5.10(50)$	$88.95\pm6.89(0.01)$
Votes	93.32 ± 5.54	93.53 ± 4.63	94.90 ± 4.00	95.82 ± 5.36	$95.60\pm3.23(50)$	$94.90\pm4.00(100)$
WDBC	95.44 ± 3.32	95.44 ± 3.32	95.96 ± 3.09	96.67 ± 2.39	$96.84\pm2.72(1)$	$96.67\pm2.67(1)$
Wine	94.86 ± 5.07	94.86 ± 5.07	96.60 ± 3.93	95.42 ± 4.63	$96.60\pm2.94(10)$	$96.60\pm2.94(10)$
WPBC	70.68 ± 6.67	70.68 ± 6.67	76.29 ± 4.59	74.24 ± 6.85	$75.79\pm8.37(0.01)$	$75.79\pm9.31(0.01)$
Ave.	82.29	82.31	85.09	85.25	85.44	84.75

where the second term penalizes the l_p norm of the coefficient vector w ($p\geq1$, and in practice usually $p\in\{1,2\}$), and $\lambda\geq0$ is a tuning regularization parameter. In order to get a sparse solution, we set $p{=}1$ [18].

The above optimization task is an l_1 regularized least square problems (l_1-*LS*) [30]. Here, all weights should be no smaller than zero. The l_1-*LS* problem with nonnegativity constraints can be rewritten as

$$\min \quad \|Ax-y\|^2+\lambda\sum_{i=1}^{n}x_i$$

$$\text{s.t.} \quad x_i\geq0, \quad i=1,2,\ldots,n. \tag{8}$$

where the variable $x\in R^n$ and the data are $A\in R^{m\times n}$ and $y\in R^m$.

Let $y=1$ and $A{=}D$ in Eq. (8), it is easy to see Eq. (8) is equal to Eq. (7) when $p{=}1$. Thus, we can obtain the solutions of this optimization task with some existing algorithms [31].

When the weights of base classifiers are obtained, we can rank the base classifiers in the descending order with respect to their weights. As pointed out before, the base classifiers with larger weights contribute more to the margin than other

Author's personal copy

classifiers, we should first consider the base classifiers with large weights. So the classifier with the largest weight is first selected, then classifiers are sequentially included in the ensemble one by one until the accuracy of combined voting does not increase. The simple plurality voting and weighted voting methods are used to combine the predictions of classifiers.

Table 5
Classification performance with CART and its ensembles.

Data	SingleCART	Bagging	RE	MDM	MAD-SV(λ)	MAD-WV(λ)
Credit	82.88 ± 14.92	84.04 ± 15.68	86.95 ± 14.14	87.66 ± 14.00	88.11 ± 12.12(50)	87.09 ± 13.50(0.01)
German	70.80 ± 3.49	76.40 ± 3.27	79.20 ± 3.94	79.70 ± 3.20	80.10 ± 4.70(50)	78.40 ± 4.60(50)
Glass	43.62 ± 15.68	49.80 ± 12.40	57.95 ± 7.96	56.90 ± 9.21	57.81 ± 11.64(1)	51.57 ± 14.99(10)
Heart	74.07 ± 6.30	82.22 ± 9.53	85.19 ± 7.41	85.93 ± 9.69	86.30 ± 8.56(10)	85.19 ± 9.88(0.01)
Hepatitis	91.67 ± 6.14	92.33 ± 8.02	95.00 ± 5.72	95.50 ± 5.44	96.83 ± 4.61(10)	95.50 ± 5.45(10)
Horse	95.65 ± 2.61	96.73 ± 1.73	97.82 ± 1.74	98.36 ± 1.92	98.36 ± 1.92(0.01)	97.82 ± 1.74(1.00)
Iono	86.43 ± 7.22	91.22 ± 4.85	95.74 ± 3.87	95.46 ± 3.04	94.61 ± 3.87(100)	93.78 ± 5.42(50)
Sonar	73.02 ± 14.91	80.29 ± 8.05	85.07 ± 10.25	86.07 ± 6.58	88.90 ± 6.10(1.00)	86.98 ± 11.13(50)
Votes	96.50 ± 3.04	96.96 ± 2.81	97.89 ± 2.39	97.88 ± 2.91	98.12 ± 2.23(0.01)	97.89 ± 2.13(0.01)
WDBC	90.50 ± 4.55	95.60 ± 3.64	96.66 ± 2.68	97.54 ± 1.89	98.07 ± 1.54(10)	96.83 ± 2.16(50)
Wine	89.86 ± 6.35	96.60 ± 2.94	97.15 ± 3.01	98.33 ± 2.68	98.26 ± 2.80(0.01)	97.15 ± 3.01(0.01)
WPBC	70.63 ± 7.54	78.21 ± 6.69	81.24 ± 6.62	82.34 ± 4.23	84.39 ± 4.85(1)	80.32 ± 3.62(0.01)
Ave.	80.47	85.03	87.99	88.47	89.16	87.38

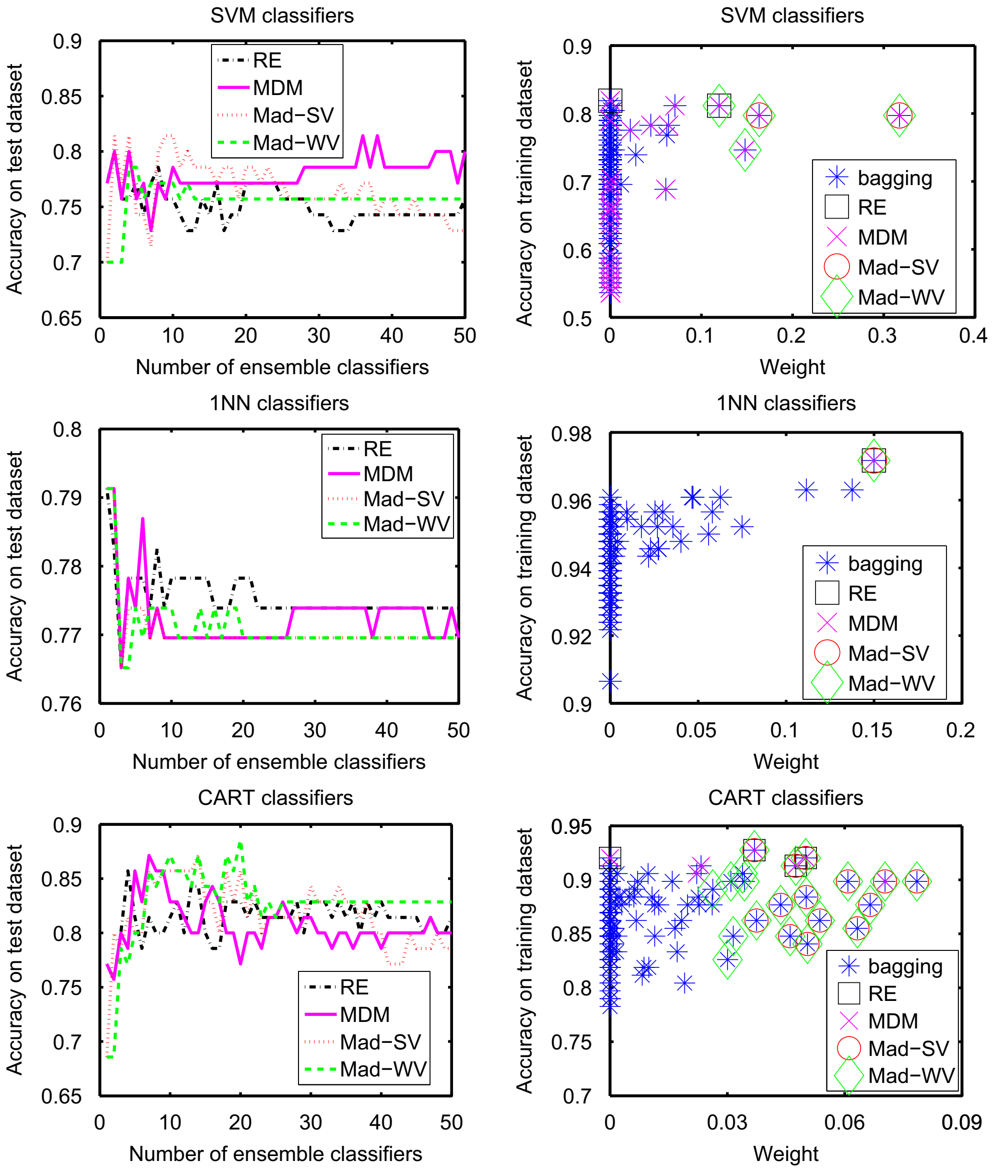


Fig. 2. Relation of accuracy and weights of Credit dataset with $\lambda = 0.001$.

Author's personal copy

The whole framework of MAD-Bagging is described in Fig. 1. There are four main steps in the framework:

1. Obtaining the bootstrap sample X_j from training set.
2. Computing the margin vector D_j with respect to each base classifier on the whole X_j.
3. Computing the weight vector w with l_1-LS optimal methods.
4. Combining the sorted base classifiers one by one to give the prediction of test data.

4. Experimental analysis

In order to test the performance of MAD-Bagging, experiments on 12 UCI data sets [32] are performed. The information about these data sets is listed in Table 1. These data sets are normalized in advance so that continuous features are valued in the interval [0,1]. In the experiment, 10-fold cross validation method is used to compute the performance of each dataset. First, the samples in each class are divided into 10 subsets randomly. Second, we carry out 10 trials for each dataset. In every trial, 9 subsets of each class are composed as training dataset and the left one is used as test dataset. For a given parameter λ, we can construct an optimization problem as Formulation (7) according to margin distribution and use $l1$-LS algorithm to obtain the weights of each base classifier w_j. We sort the base classifier according to the weights in the descending order. And the sizes of ensembles producing the best accuracy are output. Third, the mean accuracy and mean size of ensembles of the 10 trials are computed. Fourth, we repeat the above process for different values of λ. Finally, the results presented in the tables are the best average accuracies among all λ and size of ensembles according to the accuracy.

In order to compare with the proposed technique, we perform the same processing to RE and MDM. We sort the base classifier according to the accuracies of base classifiers in RE and distance to the optimal solution in MDM, respectively. Then we add the first k classifiers one by one and use the test set to estimate the classification performance. We output the best accuracies of the nested classification systems.

In this work, we try both stable (1NN and SVM) and unstable CART learners in training base classifiers. We discuss the influence of these algorithms on the final performance of MAD-Bagging. SVM is implemented by LIBSVM software [33]

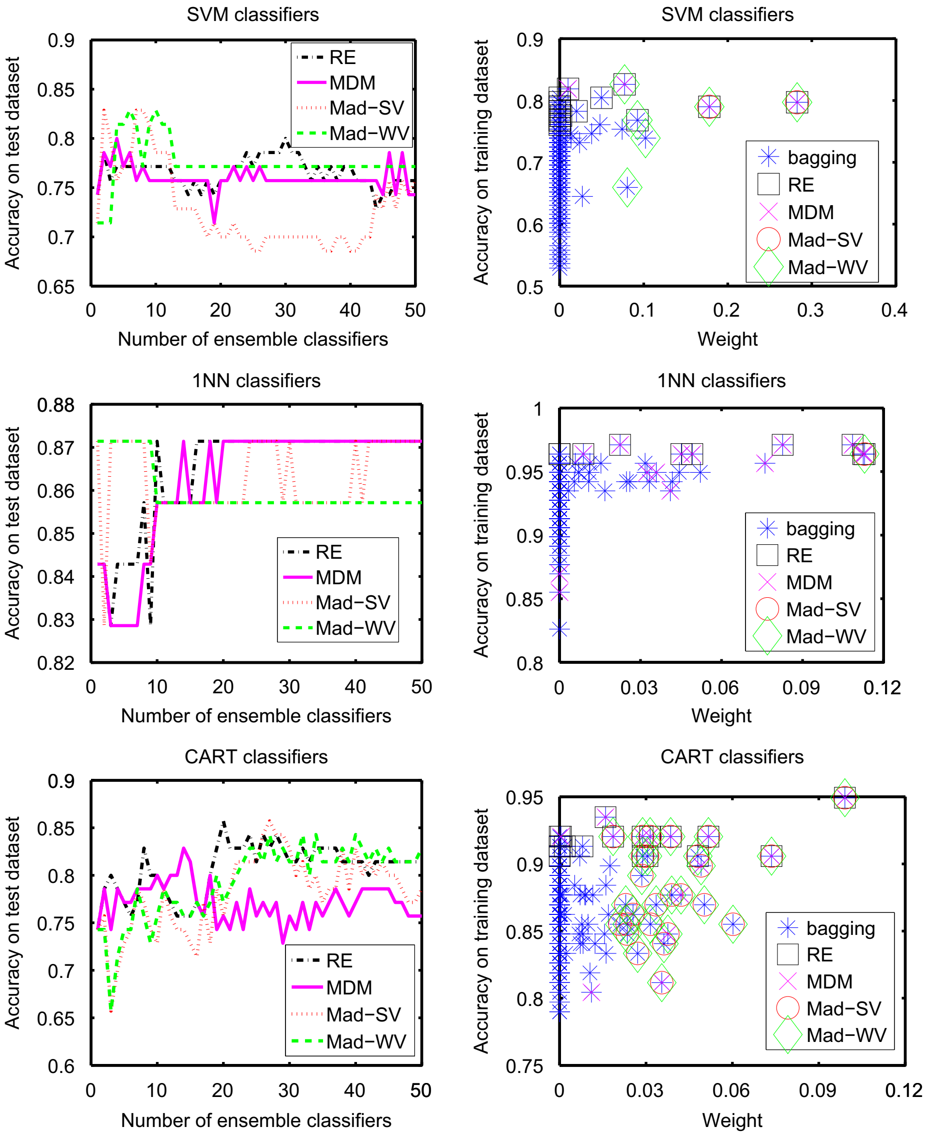


Fig. 3. Relation of accuracy and weights of Sonar dataset with $\lambda = 0.01$.

Author's personal copy

with default parameters. We use the functions of decision tree method in matlab 7.1 with default parameters to model CART classifiers, and 1NN classifier employs the Euclidean distance.

We compare the performance of MAD-Bagging using simple voting (MAD-SV), and MAD-Bagging using weighted voting (MAD-WV) with single classifiers, Bagging, RE, and MDM. For all ensembles in our experiments, we train 200 base classifiers at first. For pruned ensemble, the sizes of ensembles producing the best accuracy are reported. The results are shown in Tables 2–5, respectively.

From Table 2, we can see that the ensemble size of four pruned ensemble methods is much smaller than 200. Most base classifiers are removed from the ensembles. These methods only utilize a small part of classifiers in the ensemble. For SVM and CART classifiers, the ensembles of MAD-WV method are the smallest, and the sizes of the other three methods are nearly the same. For 1NN classifiers, the ensemble size of the four pruned ensemble methods is much smaller. We also can see that SVM and 1NN based MAD-Bagging consist of much fewer base classifiers than CART based MAD-Bagging. This result suggests ensemble of many stable classifiers is not useful for improving classification performance. If the base classifiers are diverse, more base classifiers would enhance the classification power of the ensembles.

Now we discuss the classification performance of different ensembles. From Tables 3 to 5, we can see that the average accuracy of single SVMs or 1NN is higher than that of CART, however the performance of CART based Bagging is the best. Compared with single classifiers, the performances of SVM and 1NN based ensembles do not improve much, which shows us unstable weak classification algorithms are more useful for constructing powerful ensembles.

Among different ensemble techniques, it is easy to derive that MAD-SV obtains the best average performance, which is better than a single CART by 9%. Then MDM and MAD-WV also obtain significant improvement. As a whole, we see all these four pruned ensembles outperform single classifiers and the original Bagging. This result tells us that pruning is effective for improving performances of ensemble learning.

It is interesting to know which base classifiers are selected by the optimization technique. Figs. 2 and 3 give the relation between weights of base classifiers and their training accuracies. From Table 2, we know that the ensemble size of pruned ensembles is smaller than 50. Thus, we just give the best 50 classifiers with respect to the weights in these figures. If the base classifiers are selected with the pruning techniques, they are marked in the figures. We see that MAD-SV and MAD-WV are not necessary to select the accurate base classifiers. That is to say, the base classifiers with high classification accuracies do not necessarily obtain large weights, which are computed with margin distribution. Thus some base classifiers producing good performances are not selected by MAD-SV or MAD-WV. However, RE usually selects the best base classifiers.

As to SVM, MAD-SV and MDM produce the best performance, followed by MAD-WV and RE for Credit dataset. There are 37 base classifiers in MDM ensembles, MAD-SV just uses 2 classifiers to obtain the same accuracy. For Sonar dataset, MAD-SV and MAD-WV get the best performance, followed by MDM and RE.

For 1NN classifiers, MAD-SV, MAD-WV, RE, and MDM obtain the same accuracy with selecting the same one classifier for Credit dataset. As to Sonar dataset, these four ensemble methods obtain the same accuracy. But there are only one classifier in MAD-WV and MAD-SV ensembles. MDM includes 15 classifiers and RE uses 10 classifiers to obtain the same accuracy.

As to CART, MAD-WV gets the best performance, followed by MDM, MAD-SV, and RE for Credit dataset. However, the number of classifiers selected by MAD-WV is the most. For Sonar, RE and MAD-SV get the best performance, followed by MAD-WV and MDM. 6 base classifiers are selected by MAD-SV, more than that selected by RE.

Fig. 4 presents the sparseness of the learned weights with different parameter values for λ. It is easy to see if λ increases, the number of nonzero weights decreases. As to MAD-WV, the ensemble size is smaller if λ increases. However, as to MAD-SV, there is no significant difference when λ varies.

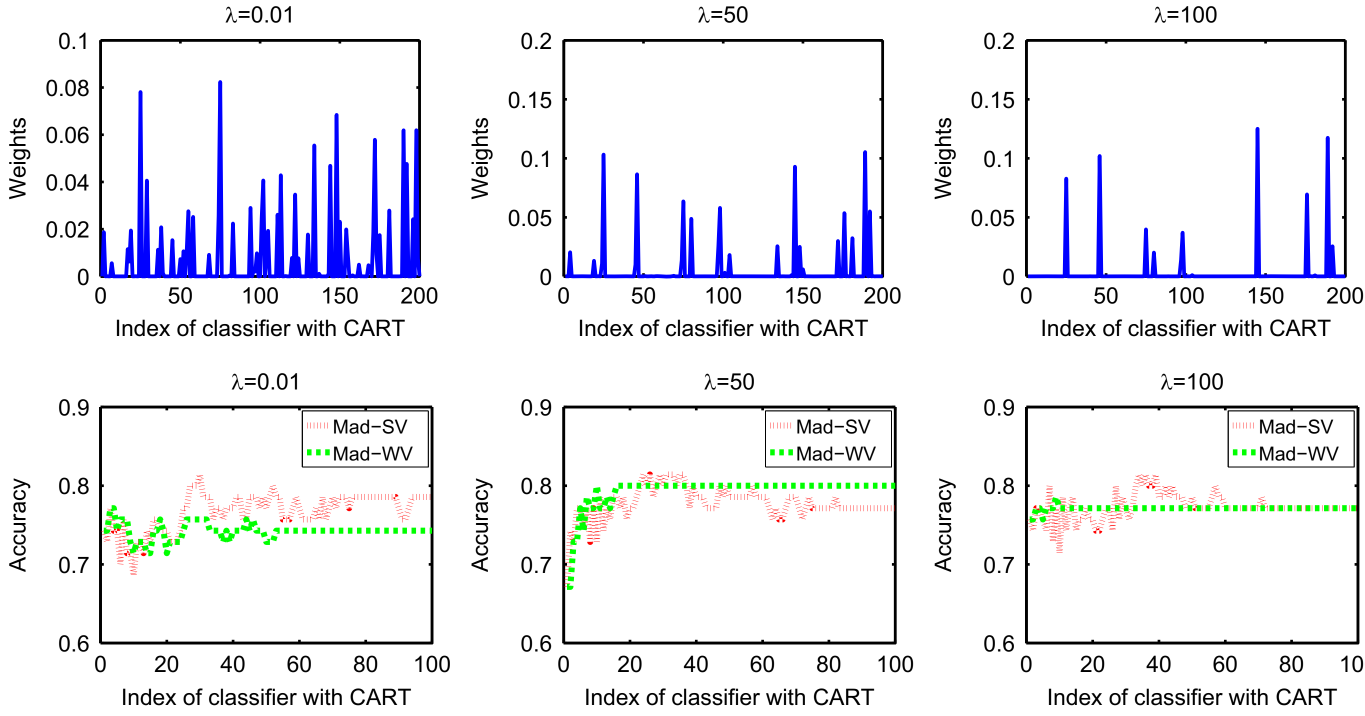


Fig. 4. Sparse characteristic of weights.

Author's personal copy

5. Conclusion and future work

In this paper, we introduce margin distribution of ensembles to select the subset of base classifiers for Bagging. The squared hinge loss and l_1 regularization are combined in the objective function. The large margin leads to a smaller classification loss. We optimize the weights of base classifiers such that the classification loss is minimized. At the same time, the sizes of ensembles are also controlled through l_1 regularization. This optimal problem is converted to an $l1$-LS problem. Thus, a collect of existing techniques can be introduced to derive the solution. It is notable that there is a parameter λ to be set for controlling the sparsity of weights. If λ increases, the solution may become sparser.

In the experiments, we compare our methods MAD-SV and MAD-WV with single classifiers, Bagging, RE and MDM on classification algorithms SVM, CART, and 1NN. Experimental results on 12 UCI datasets are given. We can draw some conclusions from the analysis. First, unstable base classifiers can lead to more powerful ensembles. Second, the base classifiers producing high classification accuracies may not be useful for constructing powerful ensembles. Both diversity and accuracy should be considered. Third, pruning is very effective for improving performance of ensembles. Last, optimizing margin distribution, instead of minimal margin or classification accuracy, improves the classification of ensembles. We should learn the weights of base classifiers based on margin distribution.

In this work, we just consider the squared hinge loss in the optimization objective. In fact there are a collection of loss functions to be used, such as the exponential loss and logistic loss. Moreover l_2 regularization can also be considered and combined with different loss functions. We will work along these directions in future.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants 61105054, 61071179 and 10978011. And this article was partly supported by Key Laboratory of Network Oriented Intelligent Computation, Program for New Century Excellent Talents in University (No. NCET-08-0156); Dr Xie is supported by China Postdoctoral Science Foundation funded project.

References

[1] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[2] G. Martinez-Muoz, 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.
[3] L. Zhang, W. Zhou, Sparse ensembles using weighted combination methods based on linear programming, Pattern Recognition 44 (1) (2011) 97–106.
[4] G. Martınez-Munoz, A. Suárez, Aggregation ordering in bagging, in: Proceedings of the IASTED International Conference on Artificial Intelligence and Applications, Citeseer, 2004, pp. 258–263.
[5] G. Martínez-Muñoz, A. Suárez, Pruning in ordered bagging ensembles, in: Proceedings of the 23rd International Conference on Machine Learning, ACM, 2006, pp. 609–616.
[6] G. Maríne-Muñoz, D. Hernández-Lobato, A. Suárez, Selection of decision stumps in bagging ensembles, in: Artificial Neural Networks—ICANN, 2007, pp. 319–328.
[7] G. Martínez-Mu noz, A. Suárez, Using boosting to prune bagging ensembles, Pattern Recognition Lett. 28 (1) (2007) 156–165.
[8] C. Tamon, J. Xiang, On the boosting pruning problem, in: Proceedings of the 11th European Conference on Machine Learning, Springer-Verlag, 2000, pp. 404–412.
[9] Z. Zhou, J. Wu, W. Tang, Ensembling neural networks: many could be better than all, Artif. Intell. 137 (1–2) (2002) 239–263.
[10] Y. Zhang, S. Burer, W. Street, Ensemble pruning via semi-definite programming, J. Mach. Learn. Res. 7 (2006) 1315–1338.
[11] D. Margineantu, T. Dietterich, Pruning adaptive boosting, in: Machine Learning—International Workshop, Morgan Kaufmann Publishers, Inc., 1997, pp. 211–218.
[12] H. Chen, P. Tino, X. Yao, Predictive ensemble pruning by expectation propagation, IEEE Trans. Knowl. Data Eng. 21 (7) (2009) 999–1013.
[13] 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.
[14] A. Grove, D. Schuurmans, Boosting in the limit: maximizing the margin of learned ensembles, in: Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence, American Association for Artificial Intelligence, 1998, pp. 692–699.
[15] A. Garg, D. Roth, Margin distribution and learning. in: Machine Learning—International Workshop, vol. 20, 2003, p. 210.
[16] L. Reyzin, R. Schapire, How boosting the margin can also boost classifier complexity, in: Proceedings of the 23rd International Conference on Machine Learning, ACM, 2006, pp. 753–760.
[17] H. Lodhi, G. Karakoulas, J. Shawe-Taylor, Boosting the margin distribution, in: Intelligent Data Engineering and Automated Learning. Data Mining, Financial Engineering, and Intelligent Agents, vol. 55, 2009, pp. 54–59.
[18] C. Shen, H. Li, Boosting through optimization of margin distributions, IEEE Trans. Neural Networks 21 (4) (2010) 659–666.
[19] D. Ruta, B. Gabrys, Classifier selection for majority voting, Inf. Fusion 6 (1) (2005) 63–81.
[20] K. Ali, M. Pazzani, Error reduction through learning multiple descriptions, Mach. Learn. 24 (3) (1996) 173–202.
[21] P. Rousseeuw, A. Leroy, J. Wiley, Robust Regression and Outlier Detection, vol. 3, Wiley Online Library, 1987.
[22] P. Domingos, Why does bagging work? A Bayesian account and its implications, in: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, Citeseer, 1997, pp. 155–158.
[23] R. Tibshirani, Bias, variance and prediction error for classification rules, 1996.
[24] D. Wolpert, W. Macready, An efficient method to estimate bagging' generalization error, Mach. Learn. 35 (1) (1999) 41–55.
[25] G. Fumera, R. Fabio, S. Alessandra, A theoretical analysis of bagging as a linear combination of classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 30 (7) (2008) 1293.
[26] R. Schapire, Y. Freund, P. Bartlett, W. Lee, Boosting the margin: a new explanation for the effectiveness of voting methods, Ann. Stat. 26 (5) (1998) 1651–1686.
[27] A. Garg, S. Har-Peled, D. Roth, On generalization bounds, projection profile, and margin distribution, in: Machine Learning—International Workshop, Citeseer, 2002, pp. 171–178.
[28] J. Shawe-Taylor, N. Cristianini, Further results on the margin distribution, in: Proceedings of the Twelfth Annual Conference on Computational Learning Theory, ACM, 1999, pp. 278–285.
[29] F. Aiolli, G. Da San Martino, A. Sperduti, A kernel method for the optimization of the margin distribution, in: Artificial Neural Networks—ICANN, 2008, pp. 305–314.
[30] S. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinevsky, An interior-point method for large-scale l1-regularized least squares, IEEE J. Sel. Top. Signal Process. 1 (4) (2007) 606–617.
[31] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B (Methodological) 55 (1996) 267–288.
[32] A. Asuncion, D. Newman, UCI Machine Learning Repository 〈http://www.ics.uci.edu/~mlearn/mlrepository.html〉. University of California, School of Information and Computer Science, Irvine, CA.
[33] C. Chang, C. Lin, Libsvm: A Library for Support Vector Machines, Software available at 〈http://www.csie.ntu.edu.tw/cjlin/libsvm〉.

Zongxia Xie received her B.S. from Dalian Maritime University in 2003, and M.S. and Ph.D. from Harbin Institute of Technology in 2005 and 2010, respectively. Now she is a postdoctoral fellow with Shenzhen Graduate School, Harbin Institute of Technology. Her major interests include machine learning and pattern recognition with rough sets and SVM, solar image processing and knowledge discovery. She has published more than 20 conference and journal papers on related topics.

Yong Xu received his B.S. and M.S. degrees at Air Force Institute of Meteorology (China) in 1994 and 1997, respectively. He then received his Ph.D. degree in pattern recognition and intelligence system at the Nanjing University of Science and Technology (NUST) in 2005. From May 2005 to April 2007, he worked at Shenzhen Graduate School, Harbin Institute of Technology (HIT) as a postdoctoral research fellow. Now he is a professor at Shenzhen Graduate School, HIT. He also acts as a research assistant researcher at the Hong Kong Polytechnic University from August 2007 to June 2008. His current interests include pattern recognition, biometrics, and machine learning. He has published more than 50 scientific papers.

Author's personal copy

Qinghua Hu received B.S., M.S. and Ph.D. degrees from Harbin Institute of Technology, Harbin, China, in 1999, 2002 and 2008, respectively. Now he is an associate professor with Harbin Institute of Technology and a postdoctoral fellow with the Hong Kong Polytechnic University. His research interests are focused on intelligent modeling, data mining, knowledge discovery for classification and regression. He is a PC co-chair of RSCTC 2010 and severs as referee for a great number of journals and conferences. He has published more than 70 journal and conference papers in the areas of pattern recognition and fault diagnosis.

Pengfei Zhu received his B.Sc. and M.Sc from Harbin Institute of Technology. Now he is working towards his Ph.D. degree in Department of Computing, The Hong Kong Polytechnic University. His research interests are focused on machine learning and pattern recognition.