Multi-objective Evolutionary Ensemble Pruning Guided by Margin Distribution*

Yu-Chang Wu, Yi-Xiao He, Chao Qian, and Zhi-Hua Zhou

State Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing 210023, China
{wuyc,heyx,qianc,zhouzh}@lamda.nju.edu.cn

Abstract Ensemble learning trains and combines multiple base learners for a single learning task, and has been among the state-of-the-art learning techniques. Ensemble pruning tries to select a subset of base learners instead of combining them all, with the aim of achieving a better generalization performance as well as a smaller ensemble size. Previous methods often use the validation error to estimate the generalization performance during optimization, while recent theoretical studies have disclosed that margin distribution is also crucial for better generalization. Inspired by this finding, we propose to formulate ensemble pruning as a three-objective optimization problem that optimizes the validation error, margin distribution, and ensemble size simultaneously, and then employ multi-objective evolutionary algorithms to solve it. Experimental results on 20 binary classification data sets show that our proposed method outperforms the state-of-the-art ensemble pruning methods significantly in both generalization performance and ensemble size.

Keywords: Machine learning · Ensemble pruning · Multi-objective optimization · Margin distribution · Multi-objective evolutionary algorithm.

1 Introduction

For one machine learning task, ensemble methods [31] train and combine multiple base learners, which can achieve a better generalization performance than a single base learner, and has been one of the most successful learning algorithms. Based on the way how the base learners are generated, ensemble methods can be generally classified into two categories: sequential methods such as Boosting [26], and parallel methods such as Bagging [4]. After generating a set of trained base learners, ensemble pruning [31] selects and combines a subset of base learners instead of combining them all, which can not only save the storage space and accelerate the prediction speed, but also lead to a better generalization performance than the whole ensemble [7,20,24,34].

In the past twenty-five years, a number of effective ensemble pruning methods have been proposed, which can be roughly classified into two groups, ordering-based pruning and optimization-based pruning. Ordering-based methods are

* This work was supported by the National Science Foundation of China (62022039, 61921006). Chao Qian is the corresponding author.

usually based on greedy strategies. Given a set of trained base learners, this kind of method iteratively selects the base learner with the largest marginal gain on some specially designed evaluation criterion. Representative criteria include minimizing the error on the validation set (i.e., validation error) [10,21], maximizing the diversity [2], maximizing the complementarity [20], or combining different evaluation criteria [17]. It has been shown that compared with combining all base learners, ordering-based methods can often achieve a smaller error on the test set (i.e., test error) by selecting only a subset of base learners [20].

Different from ordering-based methods, optimization-based pruning methods formulate ensemble pruning as an optimization problem explicitly, and then apply optimization techniques to search for the optimal subset of base learners that constitutes the final pruned ensemble. As evolutionary algorithms (EAs) [1] inspired by natural evolution are a kind of general-purpose optimization algorithms, they have been naturally used for ensemble pruning. Indeed, the first work which opened the direction of optimization-based pruning [34] used a standard genetic algorithm to select a subset of base learners minimizing the validation error. Compared with the ordering-based methods, the generated pruned ensemble has a competitive test error, but also has a much larger ensemble size.

In order to obtain not only a good generalization performance but also a small ensemble size, Qian et al. [24] formulated ensemble pruning as an explicit bi-objective optimization problem that minimizes the validation error and ensemble size simultaneously, and proposed the Pareto Ensemble Pruning (PEP) method, which employs a simple MOEA [16,23] combined with a local search operator to solve the bi-objective problem. It has been shown [24] that PEP can be significantly better on both test error and ensemble size than various ordering-based methods [2,10,17,20,21] as well as the single-objective optimization-based method that minimizes the validation error only [34].

Ensemble pruning naturally has two goals: maximizing the generalization performance and minimizing the ensemble size. The above-mentioned works (e.g., [20,24,34]) mainly measured the generalization performance by the validation error during the optimization process. However, it has been revealed that the generalization performance depends on not only the error on a sampled data set, but also the margin, i.e., the distance from a sampled data to the decision boundary. Margin theory for Boosting was first presented by Schapire et al. [3] to explain the success of AdaBoost. Soon after, Breiman [5] proved that the minimum margin is crucial to the margin theory, but optimizing the minimum margin led to poor empirical generalization performance; this sentenced margin theory to death. Later, Reyzin and Schapire conjectured that it is margin distribution rather than minimum margin concerns [25]. Gao and Zhou [14] finally proved that it is crucial to optimize margin distribution, characterized by maximizing margin mean and minimizing margin variance simultaneously. Later, Grønlund et al. [15] proved that one cannot hope for much stronger upper bounds than Gao and Zhou's result. Gao and Zhou's result has inspired many advanced machine learning algorithms to maximize margin mean and minimize margin variance simultaneously [29,30], generally by taking one of them as an objective whereas

the other as a constraint. Lyu et al. [19] tried to take margin ratio, defined by the standard deviation of margin over margin mean, and applied it to improve deep forest. But to the best of our knowledge, the margin distribution has not been exploited for ensemble pruning.

In this paper, we propose a Margin Distribution guided multi-objective evolutionary Ensemble Pruning (MDEP) method, which formulates ensemble pruning as a three-objective optimization problem that minimizes the validation error, margin ratio [19] and ensemble size simultaneously, and then applies advanced multi-objective EAs (MOEAs) to solve it. Experiments have been conducted on 20 binary classification data sets. We first examine the performance of MDEP equipped with three typical MOEAs, i.e. NSGA-II [9], MOEA/D [28] and NSGA-III [8], suggesting that NSGA-III leads to the best performance. Then, we compare MDEP using NSGA-III against all the state-of-the-art pruning methods introduced before, showing that MDEP can achieve a better test error with a significantly smaller ensemble size. Finally, we also perform an ablation study to show that introducing the objective of minimizing the margin ratio (i.e., optimizing the margin distribution) really contributes to the advantage of MDEP.

2 MDEP Method

In this section, we first introduce the three-objective formulation (i.e., validation error, margin distribution and ensemble size) of the ensemble pruning problem, and then show how to solve this three-objective problem by MOEAs.

2.1 Three-objective Formulation with Margin Distribution

Given a set of n trained base learners $H = \{h_t\}_{t=1}^n$, where $h_t : \mathcal{X} \to \mathcal{Y}$ maps the instance space $\mathcal{X}$ to the label space $\mathcal{Y}$. Let $H_{\boldsymbol{s}}$ denote a pruned ensemble with the selector vector $\boldsymbol{s} \in \{0,1\}^n$, where $\forall t \in \{1,2,\ldots,n\}$, $s_t = 1$ and $s_t = 0$ mean that the base learner h_t is selected and unselected, respectively. Considering using voting to combine the base learners, the output of $H_{\boldsymbol{s}}$ on an instance $\boldsymbol{x} \in \mathcal{X}$ is calculated by taking an average of the selected base learners, i.e.,

$$H_{\boldsymbol{s}}(\boldsymbol{x}) = \frac{1}{|\boldsymbol{s}|} \sum_{t=1}^{n} s_t h_t(\boldsymbol{x}), \tag{1}$$

where $|\boldsymbol{s}| = \sum_{t=1}^{n} s_t$ represents the ensemble size. The goal of ensemble pruning is to select a pruned ensemble $H_{\boldsymbol{s}}$ that optimizes the generalization performance (i.e., the expected prediction error under the unknown data distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$) while containing as few base learners as possible.

Ensemble pruning can be naturally formulated as a bi-objective optimization problem that optimizes the generalization performance of $H_{\boldsymbol{s}}$ and minimizes the ensemble size $|\boldsymbol{s}|$, simultaneously. Previous work [24,34] measured the generalization performance by the validation error only. However, it has been proved by Gao and Zhou [14] that the generalization performance depends on not only the error on a sampled data set, but also the margin distribution.

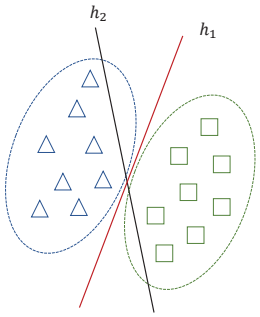

Figure 1: A simple illustration of two linear classifiers h_1, h_2 with the same validation error but different margin distributions. Dotted ellipses are two underlying distributions, from which blue triangles and green squares are validation instances sampled for two classes. This illustration takes the idea from [32].

To intuitively show that the generalization performance of a learner is related to the margin distribution, we consider an example of binary classification in Figure 1. The margin of an instance with respect to a learner is the distance from the instance to the learner's decision boundary, which can also be viewed as a measure of confidence in classification. The larger the margin, the better it is. Figure 1 illustrates the importance of margin distribution. h_1 and h_2 are different linear classifiers with equal validation errors which cannot be distinguished if we only consider the validation error. But when we also consider the margin distribution, h_1 has larger margins on most sampled instances and will be selected, which is the true better classifier that separates the two classes perfectly.

For achieving a better generalization performance, it is thus required to optimize both the error and the margin distribution on the validation set. Let $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$ denote the given validation set. Considering binary classification, i.e., $\mathcal{Y} = \{+1, -1\}$, the validation error of a pruned ensemble $H_{\boldsymbol{s}}$ can be represented as

$$\text{error}_D(H_{\boldsymbol{s}}) = \frac{1}{m}\sum_{i=1}^{m}\left(I(y_i H_{\boldsymbol{s}}(\boldsymbol{x}_i) < 0) + \frac{I(y_i H_{\boldsymbol{s}}(\boldsymbol{x}_i) = 0)}{2}\right), \tag{2}$$

where $I(\cdot)$ is the indicator function that is 1 if the inner expression is true and 0 otherwise. Note that $y_i H_{\boldsymbol{s}}(\boldsymbol{x}_i) < 0$ implies that the pruned ensemble $H_{\boldsymbol{s}}$ makes the wrong prediction; $y_i H_{\boldsymbol{s}}(\boldsymbol{x}_i) = 0$ implies that $H_{\boldsymbol{s}}(\boldsymbol{x}_i)$ in Eq. (1) is equal to 0, and the pruned ensemble will make a random guess, resulting in an error with probability 1/2. The margin of the labeled instance $(\boldsymbol{x}_i, y_i)$ with respect to a pruned ensemble $H_{\boldsymbol{s}}$ is

$$\rho_{H_{\boldsymbol{s}}}(\boldsymbol{x_i}, y_i) = y_i H_{\boldsymbol{s}}(\boldsymbol{x}_i) = \frac{1}{|\boldsymbol{s}|}\left(\sum_{t:y_i=h_t(\boldsymbol{x}_i)} \boldsymbol{s}_t - \sum_{t:y_i\neq h_t(\boldsymbol{x}_i)} \boldsymbol{s}_t\right). \tag{3}$$

Gao and Zhou [14] have revealed that a smaller margin variance and a larger margin mean will lead to a better margin distribution, and Lyu et al. [19] have further proved that margin distribution can be characterized by margin ratio

related to the margin standard deviation against the margin mean. For a pruned ensemble $H_{\boldsymbol{s}}$, its margin ratio on the validation set D can be calculated as

$$\rho_D^{\text{ratio}}(H_{\boldsymbol{s}}) = \sqrt{\frac{\text{Var}_D(\rho_{H_{\boldsymbol{s}}})}{\text{Mean}_D^2(\rho_{H_{\boldsymbol{s}}})}} = \sqrt{\frac{m\sum_{i\neq j}\left(\rho_{H_{\boldsymbol{s}}}(\boldsymbol{x_i}, y_i) - \rho_{H_{\boldsymbol{s}}}(\boldsymbol{x_j}, y_j)\right)^2}{2(m-1)(\sum_{i=1}^{m}\rho_{H_{\boldsymbol{s}}}(\boldsymbol{x_i}, y_i))^2}}, \quad (4)$$

where $\text{Var}_D(\rho_{H_{\boldsymbol{s}}})$ and $\text{Mean}_D(\rho_{H_{\boldsymbol{s}}})$ denote the margin variance and mean, respectively, of the instances in D with respect to $H_{\boldsymbol{s}}$, and $\rho_{H_{\boldsymbol{s}}}(\boldsymbol{x_i}, y_i)$ is the margin of $(\boldsymbol{x_i}, y_i)$ with respect to $H_{\boldsymbol{s}}$, as calculated in Eq. (3). The smaller the margin ratio, the better the margin distribution and thus the generalization performance.

Based on the above analysis, we formulate ensemble pruning as a three-objective minimization problem

$$\arg\min_{\boldsymbol{s}\in\{0,1\}^n} \quad \left(\text{error}_D(H_{\boldsymbol{s}}), \rho_D^{\text{ratio}}(H_{\boldsymbol{s}}), |\boldsymbol{s}|\right). \quad (5)$$

That is, the validation error, the margin ratio and the ensemble size are minimized simultaneously. Note that minimizing the first two objectives corresponds to optimizing the generalization performance. To the best of our knowledge, this is the first time that margin distribution is utilized for ensemble pruning.

By solving the three-objective problem formulated in Eq. (5) by MOEAs, we propose the Margin Distribution guided multi-objective evolutionary Ensemble Pruning method, briefly called MDEP. Though the margin in Eq. (3) is defined for binary classification, it can be adapted to multi-class classification and regression accordingly [11,22], and thus MDEP can also be applied to these tasks.

2.2 Multi-objective Evolutionary Algorithms

Next, we will show how MDEP applies MOEAs to solve the three-objective problem in Eq. (5). The input of MDEP is a set of trained base learners $H = \{h_t\}_{t=1}^n$ and a validation data set D. As introduced before, a pruned ensemble can be naturally represented by a Boolean vector $\boldsymbol{s} \in \{0,1\}^n$, where the t-th bit $s_t = 1$ if and only if the base learner h_t is selected. Note that the solution with all 0s is excluded during optimization. The procedure of MDEP is presented in Algorithm 1. In fact, MDEP can be equipped with any existing MOEA, e.g., NSGA-II [9], MOEA/D [28] and NSGA-III [8]. Here, we mainly introduce the special initialization, crossover and mutation operations that MDEP adopts.

Initialization. With the goal of improving the search efficiency of MDEP in the solution space with a small ensemble size, we evaluate all the solutions with size 1 in line 1 of Algorithm 1, and select the non-dominated solutions among them as initial solutions in line 2. Note that these solutions must be Pareto optimal, because solutions with size larger than 1 cannot dominate them. To fill in the initial population P_1, the remaining initial solutions are randomly selected from the whole solution space $\{0,1\}^n$ in line 3. Note that this setting implicitly requires that the population size is at least the number of Pareto optimal solutions with size 1. In our experiments, the population size will be set to n, which obviously satisfies this requirement.

Algorithm 1 MDEP Method

Input: Original ensemble $H = \{h_t\}_{t=1}^{n}$, validation data set $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$
Output: Pruned ensemble $H_{\boldsymbol{s}}$

1: Evaluate all the solutions $\{\boldsymbol{s}^i\}_{i=1}^{n}$ with size 1, where $\boldsymbol{s}^i$ has value 1 on the i-th bit, and 0 otherwise;
2: Select the non-dominated solutions among $\{\boldsymbol{s}^i\}_{i=1}^{n}$, and add them into the initial population P_1;
3: For the remaining required initial solutions, randomly select them from $\{0,1\}^n$;
4: **for** $t = 1$: maximum #generations **do**
5: Select solutions from P_t to compose the mating pool;
6: Generate offspring population P' by uniform crossover and bit-wise mutation;
7: **for** each offspring solution $\boldsymbol{s}' \in P'$ **do**
8: **if** $|\boldsymbol{s}'| \leq 1$ **then**
9: **repeat**
10: Apply the bit-wise mutation operator to update $\boldsymbol{s}'$
11: **until** $|\boldsymbol{s}'| > 1$
12: **end if**
13: Evaluate $\boldsymbol{s}'$
14: **end for**
15: Select next population P_{t+1} from $P_t \cup P'$
16: **end for**
17: Select a non-dominated solution $\boldsymbol{s}$ from the final population

Reproduction. To reproduce offspring solutions from the selected parent solutions, we employ the common operators over Boolean vector representation: uniform crossover and bit-wise mutation [13], as shown in line 6 of Algorithm 1. The uniform crossover operator generates the first offspring solution by inheriting each bit from the first parent solution independently with probability 1/2, and otherwise from the second parent. The second offspring is created using inverse mapping. The bit-wise mutation operator flips each bit of a solution independently with probability $1/n$. Since we have explored all the solutions with size 1 in the initialization procedure, when an offspring solution $\boldsymbol{s}'$ with $|\boldsymbol{s}'| \leq 1$ is generated, the bit-wise mutation operator is applied repeatedly to update $\boldsymbol{s}'$ until $|\boldsymbol{s}'| > 1$ (i.e., lines 8–12).

Though the above settings are simple, they have been sufficient to lead to a good performance of MDEP, which will be shown in our experiments. More careful designs may further improve the performance. Note that the uniform crossover and bit-wise mutation operators are usually applied with some probabilities, denoted as P_{c} and P_{m}, respectively. They are treated as two hyperparameters. The parent selection strategy for reproduction in line 5 as well as the survival selection strategy for updating the population in line 15 depends on the concrete MOEA employed by MDEP. For example, if NSGA-II [9] is employed, binary tournament selection is used to select parent solutions and the survival selection strategy is based on non-dominated sorting and crowding distance.

MDEP will continue to run until a predefined number of generations (i.e., maximum #generations in line 4 of Algorithm 1) is reached. After MDEP terminates, we will get a set of solutions, and the final output solution can be selected

according to the user's preference. Here we propose to select the solution with the smallest validation error from the final population. If such a solution is not unique, we select the solution with the smallest ensemble size among them. This strategy of selecting the final solution will be used in our experiments.

3 Experiments

In this section, we empirically examine the performance of MDEP. Section 3.1 introduces the general experimental settings. As MDEP can be equipped with any MOEA, we compare the performance of MDEP using three typical MOEAs, i.e., NSGA-II [9], MOEA/D [28] and NSGA-III [8], in Section 3.2, showing that NSGA-III is the best choice. Next, we compare MDEP equipped with NSGA-III against state-of-the-art pruning methods in Section 3.3. Finally, Section 3.4 performs an ablation study to examine whether considering the margin distribution in problem formulation, i.e., introducing the objective of minimizing the margin ratio in Eq. (5), really contributes to the advantage of MDEP.

3.1 Settings

We conduct experiments on 20 binary classification data sets from the UCI repository [12]. Some of the binary classification data sets are generated from multi-class data sets: *letter-ah* is based on the *letter* data and classifies 'a' against 'h', and alike *letter-br* and *letter-oq*; *optdigits* classifies '01234' against '56789'; *satimage*-12*v*57 is based on the *satimage* data and classifies labels '1' and '2' against '5' and '7', and alike *satimage*-2*v*5; *vehicle-bo-vs* is based on the *vehicle* data and classifies 'bus' and 'opel' against 'van' and 'saab', and alike *vehicle-b-v*.

To evaluate each method on each data set, a data set is evenly and randomly split into three parts: training set, validation set and test set. We use Bagging [4] to train 100 C4.5 decision trees [6] on the training set as the original ensemble $H = \{h_t\}_{t=1}^{n}$, and then prune the ensemble by a pruning method on the validation set. Finally, we report the performance of the pruned ensemble on the test set. In order to reduce the influence of randomness, each data set is randomly partitioned 30 times independently, and each method will be performed on each partition of the data set and the average performance will be reported.

3.2 Comparison of MDEP Using Various MOEAs

Since MDEP can employ various MOEAs to solve the three-objective problem in Eq. (5), we first compare the performance of MDEP equipped with NSGA-II [9], MOEA/D [28] and NSGA-III [8]. Because the optimization process of an MOEA is inherently stochastic, for each partition of each data set, the MOEA is repeated 5 times further. That is, each MOEA on each data set is repeated 150 times (30 partitions $\times$ 5 times per partition). For fairness of comparison, we use the same hyperparameter setting for each MOEA. The population size is 100, the number of generations is 500. The probabilities P_{c} and P_{m} of applying

Table 1: The test errors and ensemble sizes (mean+std.) of the compared methods on 20 binary data sets. The smallest error and size on each data set are bolded. In the row of "count of the best", the largest values are bolded. The "w/t/l to Bagging" denotes the number of data sets where the test error of MDEP using a specific MOEA is smaller, same, or larger, compared to Bagging.

	Test Error					Ensemble Size		
Data Set	NSGA-III	NSGA-II	MOEA/D	Bagging	BI	NSGA-III	NSGA-II	MOEA/D
australian	**.143±.020**	.144±.021	**.143±.020**	**.143±.017**	.152±.023	8.2±3.4	**7.5±3.3**	**7.5±3.1**
breast-cancer	**.273±.035**	.279±.038	.278±.035	.279±.037	.298±.044	7.4±2.7	6.9±1.6	**6.8±2.2**
liver-disorders	.312±.033	.313±.033	**.310±.035**	.327±.047	.365±.047	11.2±3.8	**10.6±3.7**	10.9±3.3
heart-statlog	**.192±.037**	.197±.040	.195±.040	.195±.038	.235±.049	**7.7±2.4**	7.9±.2.7	**7.7±2.1**
house-votes-84	.044±.018	.045±.019	.043±.020	**.041±.013**	.047±.016	**3.0±1.4**	3.1±1.8	**3.0±1.9**
ionosphere	**.083±.022**	.085±.025	**.083±.023**	.092±.025	.117±.022	5.0±1.6	**4.9±1.7**	5.1±1.7
kr-vs-kp	**.009±.003**	.010±.003	**.009±.003**	.015±.007	.011±.004	**3.8±1.4**	4.0±1.2	4.4±1.8
letter-AH	**.012±.006**	.014±.006	**.012±.006**	.021±.006	.023±.008	5.1±2.0	**4.9±1.9**	5.1±1.7
letter-BR	**.045±.011**	.047±.012	.048±.010	.059±.013	.078±.012	9.8±2.2	**9.4±2.5**	10.7±2.9
letter-OQ	**.041±.009**	.042±.010	.043±.009	.049±.012	.078±.017	9.9±2.5	**9.8±2.7**	10.7±2.9
optdigits-b	.035±.005	**.034±.005**	.037±.005	.038±.007	.095±.008	**21.1±4.1**	21.7±4.5	21.5±5.3
satimage-12v57	**.028±.004**	**.028±.004**	**.028±.004**	.029±.004	.052±.006	**13.7±3.1**	14.3±4.6	14.7±4.2
satimage-25	.022±.006	**.021±.007**	**.021±.006**	.023±.009	.033±.010	**5.4±1.3**	5.6±1.9	5.7±1.9
sick	**.015±.003**	**.015±.003**	.016±.003	.018±.004	.018±.004	5.8±2.2	**5.6±2.7**	6.2±1.8
sonar	**.244±.052**	.257±.057	.257±.040	.266±.052	.310±.051	10.9±3.5	**9.9±2.7**	10.9±3.5
spambase	**.065±.006**	.066±.007	.066±.006	.068±.007	.093±.008	14.0±4.9	**13.7±3.7**	14.0±3.4
tic-tac-toe	**.128±.024**	.131±.021	**.128±.022**	.164±.028	.212±.028	12.4±3.2	**11.2±3.2**	12.0±3.1
vehicle-bo-vs	.226±.022	**.223±.021**	.229±.021	.228±.026	.257±.025	13.1±4.6	**11.9±4.1**	12.6±3.6
vehicle-b-v	**.019±.011**	.020±.012	**.019±.013**	.027±.014	.024±.013	**2.8±1.0**	**2.8±1.1**	2.9±1.5
vote	**.044±.018**	.046±.019	.046±.020	.047±.018	.046±.016	2.9±1.5	**2.7±1.1**	2.8±1.3
count of the best	**15**	5	9	2	0	7	**13**	4
w/t/l to Bagging	18/1/1	16/1/3	16/2/2	-	-	-	-	-

crossover and mutation are set arbitrarily to 0.7 and 1, respectively. The more careful setting may achieve better results.

The comparative methods also include two baselines: Bagging which uses the original ensemble (i.e., all 100 trained base learners), and Best Individual (BI) which selects the base classifier with the smallest validation error. Table 1 gives the detailed results, i.e., the mean and standard deviation of test error and ensemble size of each method on each data set. To save space, MDEP equipped with a specific MOEA is denoted by the name of the MOEA in Table 1. For example, NSGA-III actually means MDEP equipped with NSGA-III. Among all the comparison methods, BI has the worst test error on all data sets, which is consistent with the fact that an ensemble of multiple classifiers usually achieves better generalization performance than a single classifier. From the row of "w/t/l to Bagging", we can observe that MDEP using any MOEA achieves a smaller test error than Bagging on at least 80% (16/20) data sets. Furthermore, by the Wilcoxon rank-sum test [27] with confidence level 0.1, MDEP using any MOEA can be significantly better than Bagging on 60% (12/20) of the data sets.

By the row of "count of the best", we can observe that MDEP using NSGA-III achieves the smallest test error on 75% (15/20) data sets, which is better than using other MOEAs. This may be because NSGA-III is proposed to improve the performance of NSGA-II for problems with more objectives. Though using NSGA-II most often achieves the smallest ensemble size, the average ensemble

size of NSGA-III, NSGA-II, and MOEA/D on 20 data sets is similar, which is 8.66, 8.42 and 8.76, respectively. That is, MDEP using any MOEA will reduce the original ensemble size greatly.

In conclusion, MDEP using any MOEA can result in better generalization performance with significantly reduced ensemble size. Furthermore, using NSGA-III leads to the best performance of MDEP, which achieves a smaller test error with a similar ensemble size, compared with using other MOEAs.

3.3 MDEP vs. State-of-the-art Pruning Methods

Next, we compare MDEP with state-of-the-art ensemble pruning methods. Note that MDEP uses NSGA-III here, which has been shown to be the best choice in Section 3.2. We implement seven state-of-the-art pruning methods, including five ordering-based methods: Reduce-Error (RE) [7], Kappa [2], ComPlementarity (CP) [20], Margin Distance (MD) [21] and DREP [17]; two optimization-based methods: EA [33,34] that employs a standard genetic algorithm to minimize the validation error only, and PEP [24] that employs a simple MOEA [16] combined with a local search operator to minimize the validation error and ensemble size simultaneously. Note that EA and PEP output the pruned ensemble with the smallest validation error from the final population [24,33,34]. The hyperparameter p of MD is set to 0.075 [21], and the hyperparameter ρ of DREP is selected from $\{0.2, 0.25, \ldots, 0.5\}$ [17]. As suggested by [24], the total number of fitness evaluations used by EA and PEP is set to $n^4 \log n$, which is much greater than that (i.e., population size 100×500 #generations $= 50,000$) of MDEP as $n = 100$. Though this is unfair MDEP, better performance on test error and ensemble size can still be achieved by MDEP, and will be shown later.

The average test error and ensemble size are shown in Table 2. In terms of test error, MDEP performs the best on 65% (13/20) of the data sets, while the other methods are at most 40% (8/20). Compared with any other method, MDEP is better on at least 55% (11/20) of the data sets, and is never significantly worse since no '$\circ$' appears in the upper half of Table 2. In terms of ensemble size, MDEP and PEP perform the best on 85% (17/20) and 20% (4/20) of the data sets, respectively, while the other methods never achieve the smallest size. This may be because only MDEP and PEP minimize the ensemble size explicitly. EA minimizes the validation error only, and generates ensembles with the largest size on all data sets, which is consistent with previous observation [18,34]. Compared with the runner-up PEP, MDEP achieves a smaller ensemble size on 80% (16/20) of the data sets, and is significantly better on 45% (9/20) of the data sets. To sum up, MDEP can achieve better generalization performance than other pruning methods, while with a significantly smaller ensemble size.

We further make a more comprehensive comparison between MDEP and the runner-up PEP [24]. We map all the solutions in their final population into the space of test error and ensemble size. Figure 2(a) shows the results on the data set *spambase*. It can be observed that MDEP obtains a much larger solution set than PEP, with more solutions in the lower-left corner of the figure. Note that PEP does not maintain a fixed-size population, and thus may obtain few

Table 2: The test errors and ensemble sizes (mean+std.) of the compared methods on 20 binary data sets. The smallest error and size on each data set are bolded, and '•/○' denotes that MDEP is significantly better/worse than the corresponding method by the Wilcoxon rank-sum test with confidence level 0.1. In the rows of "count of the best", the largest values are bolded. The "MDEP: w/t/l" denotes the number of data sets where the test error (or ensemble size) of MDEP is smaller, same or larger, compared to the corresponding method.

Test Error										
Data Set	MDEP	DREP	Kappa	CP	MD	RE	EA	PEP	Bagging	BI
australian	**.143±.020**	.144±.019	**.143±.021**	.145±.022	.148±.022	.144±.020	**.143±.020**	.144±.020	**.143±.017**	.152±.023•
breast-cancer	**.273±.035**	.275±.036	.287±.037•	.282±.043	.295±.044•	.277±.031	.275±.032	.275±.041	.279±.037	.298±.044•
liver-disorders	.312±.033	.316±.045	.326±.042•	.306±.039	.337±.035•	.320±.044	.317±.046	**.304±.039**	.327±.047•	.365±.047•
heart-statlog	.192±.037	.194±.044	.201±.038	.199±.044	.226±.048•	**.187±.044**	.196±.032	.197±.037	.195±.038	.235±.049•
house-votes-84	.044±.018	.045±.017	.044±.017	.045±.017	.048±.018	.043±.018	**.041±.012**	.045±.019	**.041±.013**	.047±.016
ionosphere	**.083±.022**	.085±.021	.084±.020	.089±.021•	.100±.026•	.086±.021	.093±.026•	.088±.021•	.092±.025•	.117±.022•
kr-vs-kp	**.009±.003**	.011±.003	.010±.003	.011±.003	.011±.005	.010±.004	.012±.004•	.010±.003	.015±.007•	.011±.004
letter-AH	**.012±.006**	.014±.005•	**.012±.006**	.015±.006•	.017±.007•	.015±.006•	.017±.006•	.013±.005	.021±.006•	.023±.008•
letter-BR	**.045±.011**	.048±.009	.048±.014	.048±.012	.057±.014•	.048±.012	.053±.011•	.046±.008	.059±.013•	.078±.012•
letter-OQ	**.041±.009**	**.041±.010**	.042±.011	.042±.010	.046±.011•	.046±.011•	.044±.011	.043±.009	.049±.012•	.078±.017•
optdigits-b	**.035±.005**	**.035±.006**	**.035±.005**	.036±.005	.037±.006	.036±.006	**.035±.006**	**.035±.006**	.038±.007•	.095±.008•
satimage-12v57	**.028±.004**	.029±.004	**.028±.004**	.029±.004	.029±.004	.029±.004	.029±.004	**.028±.004**	.029±.004	.052±.006•
satimage-25	.022±.006	.022±.008	.022±.007	**.021±.008**	.026±.010•	.023±.007	**.021±.008**	**.021±.007**	.023±.009	.033±.010•
sick	**.015±.003**	.016±.003	.017±.003	.016±.003	.017±.003•	.016±.003	.017±.004•	**.015±.003**	.018±.004•	.018±.004•
sonar	**.244±.052**	.257±.056	.249±.059	.250±.048	.268±.055	.267±.053•	.251±.041	.248±.056	.266±.052•	.310±.051•
spambase	**.065±.006**	**.065±.006**	.066±.006	.066±.006	.068±.007•	.066±.006	.066±.006	**.065±.006**	.068±.007•	.093±.008•
tic-tac-toe	**.128±.024**	.129±.026	.132±.023	.132±.026	.145±.022•	.135±.026	.138±.020•	.131±.027	.164±.028•	.212±.028•
vehicle-bo-vs	.226±.022	.234±.026	.233±.024	.234±.024	.244±.024•	.226±.022	.230±.024	**.224±.023**	.228±.026	.257±.025•
vehicle-bus-van	.019±.011	.019±.013	.019±.012	.020±.011	.021±.011	.020±.011	.026±.013•	**.018±.011**	.027±.014•	.024±.013
vote	.044±.018	.043±.019	**.041±.016**	.043±.016	.045±.014	.044±.017	045±.015	.044±.018	.047±.018	.046±.016
count of the best	**13**	3	5	1	0	1	4	8	2	0
MDEP: w/t/l	-	14/5/1	12/7/1	17/0/3	20/0/0	16/2/2	16/2/2	11/5/4	18/1/1	20/0/0
Ensemble Size										
australian	**8.2±3.4**	11.7±4.7•	14.7±12.6•	11.0±9.7	8.5±14.8	12.5±6.0•	41.9±6.7•	10.6±4.2•	–	–
breast-cancer	**7.4±2.7**	9.2±3.7•	26.1±21.7•	8.8±12.3	7.8±15.2	8.7±3.6•	44.6±6.6•	8.4±3.5•	–	–
liver-disorders	**11.2±3.8**	13.9±5.9•	24.7±16.3•	15.3±10.6	17.7±20.0	13.9±4.2•	42.0±6.2•	14.7±4.2 •	–	–
heart-statlog	**7.7±2.4**	11.3±2.7•	17.9±11.1•	13.2±8.2	13.6±21.1	11.4±5.0•	44.2±5.1•	9.3±2.3	–	–
house-votes-84	3.0±1.4	4.1±2.7•	5.5±3.3•	4.7±4.4	5.9±14.1	3.9±4.0	46.5±6.1	**2.9±1.7**	–	–
ionosphere	**5.0±1.6**	8.4±4.3•	10.5±6.9•	8.5±6.3•	10.7±14.6	7.9±5.7•	48.8±5.1•	5.2±2.2	–	–
kr-vs-kp	**3.8±1.4**	7.1±3.9•	10.6±9.1•	9.6±8.6•	7.2±15.2	5.8±4.5	45.9±5.8	4.2±1.8	–	–
letter-AH	5.1±2.0	7.8±3.6•	7.1±3.8•	8.7±4.7•	11.0±10.9	7.3±4.4•	42.5±6.5•	**5.0±1.9**	–	–
letter-BR	**9.8±2.2**	11.3±3.5•	13.8±6.7•	12.9±6.8•	23.2±17.6•	15.1±7.3•	38.3±7.8•	10.9±2.6	–	–
letter-OQ	**9.9±2.5**	13.7±4.9•	13.9±6.0•	12.3±4.9•	23.0±15.6•	13.6±5.8•	39.3±8.2•	12.0±3.7•	–	–
optdigits-b	**21.1±4.1**	25.0±8.0•	25.2±8.1•	21.4±7.5•	46.8±23.9•	25.0±9.3•	41.4±7.6•	22.7±3.1•	–	–
satimage-12v57	**13.7±3.1**	18.1±4.9•	22.1±10.3•	21.2±10.0•	37.6±24.3•	20.8±9.2•	42.7±5.2•	17.1±5.0•	–	–
satimage-25	**5.4±1.3**	7.7±3.5•	7.6±4.2•	10.9±7.0•	26.2±28.1•	6.8±3.2•	44.1±4.8	5.7±1.7	–	–
sick	**5.8±2.2**	11.6±6.7•	10.9±6.0•	11.5±10.0•	8.3±13.6	7.5±3.9•	44.7±8.2•	6.9±2.8	–	–
sonar	**10.9±3.5**	14.4±5.9•	20.6±9.3•	13.9±7.1	20.6±20.7	11.0±4.1	43.1±6.4	11.4±4.2	–	–
spambase	**14.0±4.9**	16.7±4.6•	20.0±8.1•	19.0±9.9•	28.8±17.0•	18.5±5.0•	39.7±6.4•	17.5±4.5•	–	–
tic-tac-toe	**12.4±3.2**	13.6±3.4	17.4±6.5•	15.4±6.3	28.0±22.6•	16.1±5.4•	39.8±8.2•	14.5±3.8•	–	–
vehicle-bo-vs	**13.1±4.6**	13.2±5.0	16.5±8.2•	11.2±5.7	21.6±20.4	15.7±5.7•	41.9±5.6•	16.5±4.5•	–	–
vehicle-bus-van	**2.8±1.0**	4.0±3.9	4.5±1.6•	5.3±7.4	2.8±3.8	3.4±2.1	48.0±5.6	**2.8±1.1**	–	–
vote	2.9±1.5	3.9±2.5	5.1±2.6•	5.4±5.2	6.0±9.8	3.2±2.7	47.8±6.1	**2.7±1.1**	–	–
count of the best	**17**	0	0	0	0	0	0	4	-	-
MDEP: w/t/l	-	20/0/0	20/0/0	20/0/0	20/0/0	20/0/0	20/0/0	16/1/3	-	-

final solutions, as observed. Figure 2(b) shows the non-dominated solutions in Figure 2(a). It can be more clearly observed that for each solution obtained by PEP, MDEP has at least one solution that can dominate it.

Since MDEP optimizes the margin distribution explicitly, we also visualize the margin distribution of the final pruned ensemble by plotting the histogram of the frequency on each margin. Figure 3 shows the results of MDEP and PEP on the data set *heart-statlog*. It can be seen that MDEP obtains larger margins (e.g., margins greater than 0.7). Although MDEP also gets more very negative margins (e.g., margins no greater than -0.7), the overall frequency of non-positive margins is less than that of PEP, implying that fewer samples are misclassified by MDEP. Thus, MDEP achieves an overall better margin distribution, suggesting a better generalization performance as observed before.

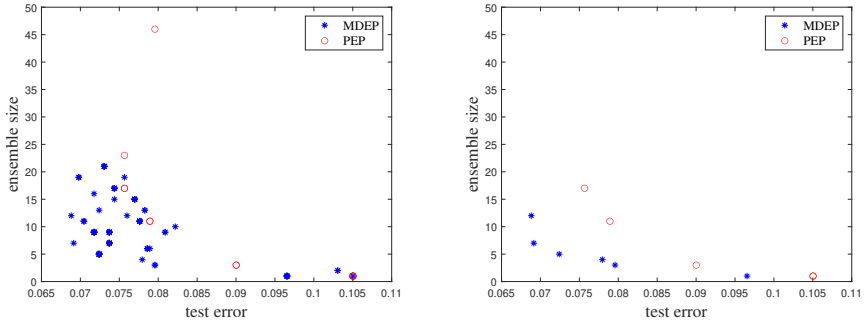

Figure 2: The final solution sets of MDEP (blue stars) and PEP (red dots) in the space of test error and ensemble size on the data set *spambase*. (a) All solutions. (b) Non-dominated solutions in (a).

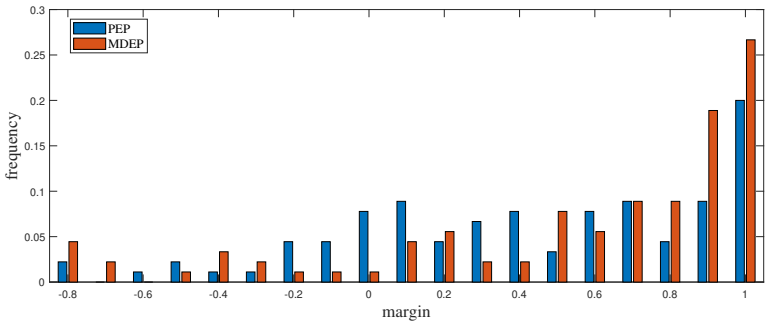

Figure 3: The histogram of the margin distributions (i.e., the frequency on each margin) obtained by MDEP and PEP on the data set *heart-statlog*.

3.4 Ablation Study

The above experiments have shown the clear advantage of MDEP, which employs NSGA-III to minimize the three-objective problem in Eq. (5). Then, a natural question is whether explicitly minimizing the margin ratio really contributes to the advantage of MDEP. Though PEP [24] is to minimize the validation error and ensemble size simultaneously, it employs a simple MOEA [16] combined with local search for optimization, and thus the superiority of MDEP over PEP cannot answer the question due to the difference of the employed optimizer.

To answer the question, we next compare the performance of the same MOEA (i.e., NSGA-III, NSGA-II or MOEA/D) optimizing the three objectives and two objectives (i.e., only the validation error and ensemble size), respectively. The hyper-parameters of all MOEAs are the same as in the previous experiments. The results are shown in Table 3. We can observe that for the same MOEA, minimizing the margin ratio additionally (corresponding to the columns of '3-obj') usually results in a smaller test error, which also supports the margin distri-

Table 3: The test errors and ensemble sizes (mean+std.) of each MOEA optimizing three or two objectives on 20 binary data sets. For each MOEA on each data set, the smaller error and size are bolded. The "3-obj vs. 2-obj: w/t/l" denotes the number of data sets where the test error (or ensemble size) of an MOEA optimizing three objectives is smaller, same or larger, compared to that of the MOEA optimizing two objectives.

Data Set	Test Error						Ensemble Size					
	NSGA-III		NSGA-II		MOEA/D		NSGA-III		NSGA-II		MOEA/D	
	3-obj	2-obj	3-obj	2-obj	3-obj	2-obj	3-obj	2-obj	3-obj	2-obj	3-obj	2-obj
australian	**.143±.020**	**.143±.019**	**.144±.021**	.147±.022	**.143±.020**	.144±.022	8.2±3.4	**7.0±3.1**	7.5±3.3	**6.7±2.6**	**7.5±3.1**	7.9±3.2
breast-cancer	**.273±.035**	.283±.037	.279±.038	**.275±.038**	**.278±.035**	.280±.039	7.4±2.7	**5.5±2.2**	6.9±1.6	**5.9±1.6**	6.8±2.2	**5.6±2.3**
liver-disorders	**.312±.033**	.325±.043	**.313±.033**	.325±.040	**.310±.035**	.313±.040	**11.2±3.8**	11.5±4.2	10.6±3.7	**10.2±4.1**	**10.9±3.3**	11.6±3.9
heart-statlog	**.192±.037**	.193±.042	**.197±.040**	.209±.039	**.195±.040**	.202±.033	7.7±2.4	**6.6±2.4**	7.9±.2.7	**6.9±2.2**	7.7±2.1	**7.2±2.3**
house-votes-84	**.044±.018**	.045±.019	**.045±.019**	.046±.018	**.043±.020**	.044±.018	3.0±1.4	**2.9±1.4**	3.1±1.8	**2.9±1.3**	3.0±1.9	**2.7±1.3**
ionosphere	**.083±.022**	.092±.021	**.085±.025**	.095±.025	**.083±.023**	.090±.023	5.0±1.6	**4.5±1.5**	4.9±1.7	**4.0±1.2**	**5.1±1.7**	**5.1±2.2**
kr-vs-kp	**.009±.003**	.010±.004	**.010±.003**	**.010±.003**	**.009±.003**	.010±.003	3.8±1.4	**3.7±1.4**	4.0±1.2	**3.6±1.2**	4.4±1.8	**3.7±1.4**
letter-AH	**.012±.006**	.013±.006	.014±.006	**.012±.006**	**.012±.006**	.013±.005	5.1±2.0	**4.7±1.7**	**4.9±1.9**	**4.9±1.8**	**5.1±1.7**	**5.1±1.5**
letter-BR	**.045±.011**	.049±.010	**.047±.012**	.048±.010	.048±.010	**.047±.011**	9.8±2.2	**9.2±3.5**	9.4±2.5	**8.3±3.1**	10.7±2.9	**8.9±3.5**
letter-OQ	**.041±.009**	.046±.010	**.042±.010**	.044±.011	**.043±.009**	.045±.011	9.9±2.5	**8.9±2.2**	9.8±2.7	**9.0±3.0**	**10.7±2.9**	11.1±2.6
optdigits-b	**.035±.005**	.036±.006	**.034±.005**	.036±.006	**.037±.005**	**.037±.006**	21.1±4.1	**20.2±4.9**	21.7±4.5	**20.3±5.6**	21.5±5.3	**18.9±5.0**
satimage-12v57	**.028±.004**	.029±.004	**.028±.004**	.029±.004	**.028±.004**	.030±.004	13.7±3.1	**13.2±4.3**	14.3±4.6	**12.5±3.7**	14.7±4.2	**13.7±3.3**
satimage-25	**.022±.006**	**.022±.008**	**.021±.007**	.022±.006	**.021±.006**	.022±.007	**5.4±1.3**	5.7±2.4	5.6±1.9	**4.7±1.2**	5.7±1.9	**5.1±1.9**
sick	**.015±.003**	.016±.003	**.015±.003**	.016±.003	**.016±.003**	.017±.003	5.8±2.2	**5.7±2.2**	5.6±2.7	**5.1±2.0**	6.2±1.8	**5.3±2.3**
sonar	**.244±.052**	.267±.071	**.257±.057**	.263±.064	.257±.040	**.255±.048**	10.9±3.5	**8.5±3.5**	9.9±2.7	**8.6±3.4**	10.9±3.5	**8.8±3.3**
spambase	**.065±.006**	.067±.006	**.066±.007**	**.066±.005**	**.066±.006**	.067±.007	**14.0±4.9**	15.1±4.8	13.7±3.7	**12.2±3.7**	**14.0±3.4**	14.2±3.9
tic-tac-toe	**.128±.024**	.133±.024	**.131±.021**	.135±.024	**.128±.022**	.137±.020	12.4±3.2	**11.0±3.5**	11.2±3.2	**10.7±3.6**	**12.0±3.1**	12.4±3.2
vehicle-bo-vs	**.226±.022**	.228±.023	**.223±.021**	.225±.019	**.229±.021**	.233±.024	**13.1±4.6**	13.7±5.0	11.9±4.1	**11.1±4.1**	12.6±3.6	**12.1±5.1**
vehicle-b-v	**.019±.011**	.021±.013	.020±.012	**.018±.012**	**.019±.013**	**.019±.013**	2.8±1.0	**2.7±1.1**	**2.8±1.1**	**2.8±1.1**	2.9±1.5	**2.8±1.2**
vote	**.044±.018**	.045±.020	.046±.019	**.045±.019**	.046±.020	**.045±.017**	2.9±1.5	**2.7±1.2**	**2.7±1.1**	**2.7±1.1**	2.8±1.3	**2.5±1.1**
3-obj vs. 2-obj: w/t/l	18/2/0		14/2/4		15/2/3		4/0/16		0/3/17		5/2/13	

bution theory [14,19]. We note that the ensemble size obtained by optimizing the three objectives is relatively larger, which may be because a solution with a larger ensemble size is easier to be dominated under the bi-objective formulation. In fact, the difference in the ensemble size is very small. For the three-objective formulation, the average ensemble size of NSGA-III, NSGA-II and MOEA/D on the 20 data sets is 8.66, 8.42 and 8.76, respectively; while for the bi-objective formulation, the average size is 8.15, 7.66 and 8.20, respectively. Furthermore, as shown in Section 3.3, the ensemble size achieved by NSGA-III under the three-objective formulation is still significantly smaller than other state-of-the-art pruning methods. Thus, these results give a positive answer, i.e., confirm that optimizing the margin distribution explicitly brings advantages.

4 Conclusion

In this paper, we introduce the three-objective (i.e., validation error, margin ratio and ensemble size) formulation of ensemble pruning, and propose a new optimization-based ensemble pruning method MDEP, which employs MOEAs to solve the three-objective problem. Experimental results show that MDEP using NSGA-III is better than using NSGA-II and MOEA/D, and more importantly, it can outperform state-of-the-art pruning methods significantly in both generalization performance and ensemble size. In the future, it would be interesting to perform theoretical analysis [35], as well as to design the components of MDEP more carefully or apply more advanced MOEAs, which may bring further performance improvement.

References

1. Back, T.: Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. Oxford University Press, Oxford, UK (1996)
2. Banfield, R.E., Hall, L.O., Bowyer, K.W., Kegelmeyer, W.P.: Ensemble diversity measures and their application to thinning. Information Fusion **6**(1), 49–62 (2005)
3. Bartlett, P., Freund, Y., Lee, W.S., Schapire, R.E.: Boosting the margin: A new explanation for the effectiveness of voting methods. The Annals of Statistics **26**(5), 1651–1686 (1998)
4. Breiman, L.: Bagging predictors. Machine Learning **24**(2), 123–140 (1996)
5. Breiman, L.: Prediction games and arcing algorithms. Neural Computation **11**(7), 1493–1517 (1999)
6. Breiman, L., Friedman, J., Olshen, R.A., Stone, C.J.: Classification and regression trees. Wadsworth and Brooks, Monterey, CA (1984)
7. Caruana, R., Niculescu-Mizil, A., Crew, G., Ksikes, A.: Ensemble selection from libraries of models. In: Proceedings of the 21st International Conference on Machine Learning (ICML'04). pp. 18–25. Banff, Canada (2004)
8. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation **18**(4), 577–601 (2013)
9. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6**(2), 182–197 (2002)
10. Dietterich, T., Margineantu, D.: Pruning adaptive boosting. In: Proceedings of the 14th International Conference on Machine Learning (ICML'97). pp. 211–218. Nashville, TN (1997)
11. Drucker, H., Burges, C.J.C., Kaufman, L., Smola, A.J., Vapnik, V.: Support vector regression machines. In: Advances in Neural Information Processing Systems 9 (NIPS'96). pp. 155–161. Denver, CO (1996)
12. Dua, D., Graff, C.: UCI machine learning repository (2017), `http://archive.ics.uci.edu/ml`
13. Eiben, A., Smith, J.: Introduction to Evolutionary Computing. Springer, Bering, Germany (2015)
14. Gao, W., Zhou, Z.H.: On the doubt about margin explanation of boosting. Artificial Intelligence **203**, 1–18 (2013)
15. Grønlund, A., Kamma, L., Larsen, K.G., Mathiasen, A., Nelson, J.: Margin-based generalization lower bounds for boosted classifiers. In: Advances in Neural Information Processing Systems 32 (NeurIPS'19). pp. 11940–11949. Vancouver, Canada (2019)
16. Laumanns, M., Thiele, L., Zitzler, E.: Running time analysis of multiobjective evolutionary algorithms on pseudo-Boolean functions. IEEE Transactions on Evolutionary Computation **8**(2), 170–182 (2004)
17. Li, N., Yu, Y., Zhou, Z.H.: Diversity regularized ensemble pruning. In: Proceedings of the 23rd European Conference on Machine Learning (ECML'12). pp. 330–345. Bristol, UK (2012)
18. Li, N., Zhou, Z.H.: Selective ensemble under regularization framework. In: Proceedings of the 8th International Workshop on Multiple Classifier Systems (MCS'09). pp. 293–303. Reykjavik, Iceland (2009)

19. Lyu, S.H., Yang, L., Zhou, Z.H.: A refined margin distribution analysis for forest representation learning. In: Advances in Neural Information Processing Systems 32 (NeurIPS'19). pp. 5531–5541. Vancouver, Canada (2019)
20. Martínez-Muñoz, G., Hernández-Lobato, D., Suárez, A.: An analysis of ensemble pruning techniques based on ordered aggregation. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(2), 245–259 (2008)
21. Martínez-Muñoz, G., Suárez, A.: Pruning in ordered bagging ensembles. In: Proceedings of the 23rd International Conference on Machine Learning (ICML'06). pp. 609–616. Pittsburgh, PA (2006)
22. Mohri, M., Rostamizadeh, A., Talwalkar, A.: Foundations of Machine Learning. MIT Press, Cambridge, MA (2018)
23. Qian, C., Yu, Y., Zhou, Z.H.: An analysis on recombination in multi-objective evolutionary optimization. Artificial Intelligence **204**, 99–119 (2013)
24. Qian, C., Yu, Y., Zhou, Z.H.: Pareto ensemble pruning. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15). pp. 2935–2941. Austin, TX (2015)
25. Reyzin, L., Schapire, R.E.: How boosting the margin can also boost classifier complexity. In: Proceedings of the 23rd International Conference on Machine Learning (ICML'06). pp. 753–760. Pittsburgh, PA (2006)
26. Schapire, R.E.: The strength of weak learnability. Machine Learning **5**(2), 197–227 (1990)
27. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics Bulletin **1**(6), 80–83 (1945)
28. Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation **11**(6), 712–731 (2007)
29. Zhang, T., Zhou, Z.H.: Optimal margin distribution clustering. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI'18). pp. 4474–4481. New Orleans, LA (2018)
30. Zhang, T., Zhou, Z.H.: Optimal margin distribution machine. IEEE Transactions on Knowledge and Data Engineering **32**(6), 1143–1156 (2019)
31. Zhou, Z.H.: Ensemble Methods: Foundations and Algorithms. Chapman & Hall/CRC Press, Boca Raton, FL (2012)
32. Zhou, Z.H.: Large margin distribution learning. In: Proceedings of the 6th International Workshop on Artificial Neural Networks in Pattern Recognition (ANNPR'14). pp. 1–11. Montreal, Canada (2014)
33. Zhou, Z.H., Tang, W.: Selective ensemble of decision trees. In: Proceddings of the 9th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC'03). pp. 476–483. Chongqing, China (2003)
34. Zhou, Z.H., Wu, J., Tang, W.: Ensembling neural networks: Many could be better than all. Artificial Intelligence **137**(1-2), 239–263 (2002)
35. Zhou, Z.H., Yu, Y., Qian, C.: Evolutionary Learning: Advances in Theories and Algorithms. Springer, Singapore (2019)