Margin Distribution and Structural Diversity Guided Ensemble Pruning

Yi-Xiao He[1], Yu-Chang Wu[1], Chao Qian[1], Zhi-Hua Zhou[1]*

National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, 210023, China.

*Corresponding author(s). E-mail(s): zhouzh@nju.edu.cn;
Contributing authors: heyx@lamda.nju.edu.cn; wuyc@lamda.nju.edu.cn; qianc@lamda.nju.edu.cn;

Abstract

Ensemble methods that train and combine multiple learners have always been among the state-of-the-art learning methods, and ensemble pruning aims at generating a smaller-sized ensemble with even better generalization performance. Abundant ensemble pruning methods that use evaluation criteria such as diversity or margin together with validation error have been proposed. However, as these evaluation criteria are used together with the validation error, their effect on generalization performance is less clear. In this paper, we propose a margin distribution and structural diversity guided ensemble pruning framework, called Decoupled Ensemble Pruning (DEP). It decouples the optimization of margin distribution and structural diversity and the optimization of validation error into two stages. Our information-theoretic analysis reveals that the expected generalization gap is related to the *combination distribution*, i.e., validation error distribution of all the combinations of base learners. And show that optimizing *margin mean* and *structural diversity* benefits combination distribution. Concretely, we provide an instantiation of DEP framework in the classic tree-based ensemble pruning setting. Experimental results not only verify the effectiveness in optimizing the distribution, but also show that DEP enjoys better test accuracy than existing ensemble pruning methods.

Keywords: Ensemble learning, Ensemble pruning, Decision trees, Information theory

1 Introduction

Ensemble learning is a powerful machine learning method that combines multiple good and diverse learners to create a stronger learner (Dietterich, 2000; Zhou, 2012). Ensemble methods have wild applications, such as disease diagnosis (Lu et al, 2020; Schaefer et al, 2014), remote sensing (Zhang et al, 2022), anomaly detection (Liu et al, 2008), and model reuse (Wu et al, 2019). Besides, ensemble learning has also been extended to deep learning (Ganaie et al, 2022; Zhou and Feng, 2019; Lyu et al, 2022). Moreover, *open-environment machine learning* (Zhou, 2022) received much attention recently, where the machine learning process has to handle unknown changes that have never occurred in training data. It is evident that the sensitivity and robustness to new changes are fundamentally important, where ensemble learning has been found well helpful. In particular, *online ensemble* has been established as a sound and practical way to handle unknown changes in online data (Zhou, 2022).

Believing that using some instead of all the individual learners of the ensemble might be better (Zhou et al, 2002), ensemble pruning aims at combing only a subset of individual learners to achieve even better performance. Since the number of individual learners is reduced, ensemble pruning also helps reduce the storage and prediction overhead. Usually, after a collection of individual learners are produced using the training set, ensemble pruning chooses a combination of them to form the pruned ensemble with the help of the validation set.

Ensemble pruning methods can be roughly categorized into ordering-based methods (Margineantu and Dietterich, 1997; Martínez-Muñoz and Suárez, 2004; Martínez-Muñoz et al, 2008; Partalas et al, 2010), optimization-based methods (Zhang et al, 2006; Zhou et al, 2002; Qian et al, 2015) and clustering-based methods (Giacinto et al, 2000; Lazarevic and Obradovic, 2001). Most ensemble pruning methods use certain evaluation criteria together with the validation error. As *diversity* has been thought of as the key to ensemble performance, many diversity measures have been designed and incorporated into ensemble pruning, such as kappa (Margineantu and Dietterich, 1997; Martínez-Muñoz et al, 2008), disagreement (Li et al, 2012), complementariness (Martínez-Muñoz and Suárez, 2004), tree edit distance (Sun and Zhou, 2018), individual contribution (Lu et al, 2010), and objection (Bian et al, 2020). Another concept frequently encountered in ensemble pruning literature is *margin* (Tang et al, 2006; Guo and Boukir, 2013). Relevant notions include margin distance (Martínez-Muñoz and Suárez, 2004), orientation (Martínez-Muñoz and Suárez, 2006), and margin distribution (Wu et al, 2022).

The benefits of diversity and margin to the ensemble's generalization performance have been analyzed respectively. Li et al (2012) used voting diversity to bound the hypothesis space complexity, thus regarding diversity as regularization. Durrant and Lim (2020) modeled the relationship between diversity and generalization performance using a Polya-Eggenberger distribution. Tang et al (2006); Bian and Chen (2021) used margin to link ensemble diversity and its generalization performance. Gao and Zhou (2013); Lyu et al (2019) showed that the generalization performance of an ensemble is closely related to margin distribution.

However, in ensemble pruning literature, the use of evaluation criteria is often nested with validation error. So the benefits of diversity and margin are not as straightforward as the above analyses. Take ordering-based pruning as an example. The evaluation criterion, say diversity, determines the order in which the individual learners are added to the sub-ensemble, and the validation error determines when to stop the aggregation process. During the nested process, too much focus on ensemble diversity may result in poor validation error, while too much focus on validation error may lead to over-fitting on validation error.

In this paper, we decouple the optimization of diversity and margin from the optimization of the validation error. We propose a margin distribution and structural diversity guided ensemble pruning framework, called Decoupled Ensemble Pruning (DEP). Specifically, we first select a subset of base learners according to the margin mean and structural diversity on the training set, then conduct validation-error-based ensemble pruning on this subset.

Theoretical results confirm the rationality behind our method. With an information-theoretic analysis for validation-error-based ensemble pruning, we show that the generalization performance of ensemble pruning not only depends on the validation error of the selected combination of base learners, but also depends on the *combination distribution*, i.e., the validation error distribution of all the combinations of base learners. The analysis leverages mutual information, which is a useful tool that has been widely used recently (Russo and Zou, 2019; Zhang et al, 2023a). We then demonstrate that our optimization of the margin mean and structural diversity on the training set will benefit the combination distribution, thus enabling the following validation-error-based ensemble pruning stage to find a solution with better generalization performance.

We provide an application of our framework in the classic tree-based ensemble pruning setting. We design a new kind of structural diversity, the feature contribution diversity, which can distinguish trees well only using the training set information, and may be of independent interest for ensemble diversity. Experimental results verify the effectiveness of DEP compared to other state-of-the-art ensemble pruning methods.

The main contributions of this paper are summarized as follows.

- We propose a novel ensemble pruning method guided by margin distribution and structural diversity, named Decoupled Ensemble Pruning (DEP).
- Our theoretical analysis reveals that optimizing margin mean and structural diversity in the first stage of DEP is beneficial to the combination distribution, thus improving the generalization performance.
- We provide an instantiation of DEP framework in the classic tree-based ensemble pruning setting with a novel feature contribution diversity measure. Experiments show that DEP achieves better test accuracy than other state-of-the-art ensemble pruning methods.

2 Setting and notations

Throughout the ensemble pruning literature, the data is usually split into three parts. The base learners are generated on the training set, then the validation set is used for pruning the ensemble, and the test set is for reporting generalization performance.

Base learners. Denote the training set as $Tr = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m_1}$, where each instance is sampled i.i.d. from data distribution $\mathcal{D}$. n base learners are trained using the training set, where each base learner $h_t : \mathcal{X} \mapsto \{1, \dots, |\mathcal{Y}|\}$ is a classifier mapping from feature space $\mathcal{X}$ to label space $\mathcal{Y}$. Let $H = \{h_t\}_{t=1}^{n}$ denote the set of trained base learners.

Ensemble pruning. It mostly takes place on the validation set $V = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m_2}$, which may also be called the pruning set (Partalas et al, 2010; Guo and Boukir, 2013) or selection set (Martínez-Muñoz et al, 2008), where each instance is sampled i.i.d. from data distribution $\mathcal{D}$. Using the validation set, we aim at selecting a small subset from H to form a pruned ensemble with better performance. Let $H_\mathbf{s}$ denote a sub-ensemble corresponding to selector vector $\mathbf{s} \in \{0,1\}^n$, where $s_t = 1$ means that the base learner h_t is incorporated in $H_\mathbf{s}$.

Suppose majority voting is used when combining the selected base learners, the prediction on sample $\mathbf{x}$ made by a sub-ensemble $H_\mathbf{s}$ is

$$H_\mathbf{s}(\mathbf{x}) = \arg\max_j \sum_{t=1}^{n} s_t \cdot \mathbb{1}_{[h_t(\mathbf{x})=j]} \ ,$$

where $\mathbb{1}_{[\cdot]}$ is the indicator function, which returns 1 if $\cdot$ is true and 0 otherwise.

The goal of ensemble pruning is to achieve better generalization error $\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\, g(H_\mathbf{s}(\mathbf{x}), y)$, where $g(f(\mathbf{x}), y) = \mathbb{1}_{[f(\mathbf{x})\neq y]}$ denotes the loss function of a classifier f on instance $(\mathbf{x}, y)$. Meanwhile, the number of base learners selected, i.e., ensemble size $|\mathbf{s}|$, should also be small. Note that usually, the generalization error is a more important goal, as we do not want to sacrifice generalization error much for a smaller ensemble size.

3 DEP method

In this section, we propose a novel ensemble pruning framework called Decoupled Ensemble Pruning (DEP). As illustrated in Figure 1, it is a two-stage selection process. In the first stage, it selects a subset of learners according to the margin distribution and structural diversity of the training set. Then in the second stage, it conducts pruning on this subset solely based on validation error.

Stage 1: Optimizing margin mean and structural diversity. In the first stage, we optimize the margin distribution and structural diversity. We formulate the bi-objective subset selection problem to be

$$\begin{aligned} &\arg\max_{\mathbf{z}\in\{0,1\}^n} \ \left(\mathbb{E}_{Tr}\left(\rho_{H_\mathbf{z}}\right), \mathrm{Div}(Tr, H_\mathbf{z})\right) \ , \\ &\qquad s.t. \ |\mathbf{z}| = n/2 \ . \end{aligned} \qquad (1)$$

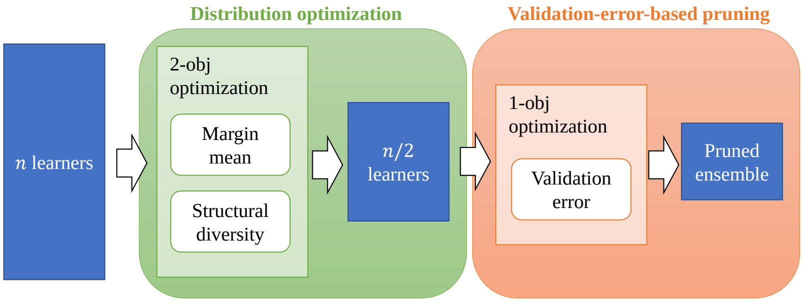


Fig. 1: An illustration of our proposed DEP framework.

The first objective is the margin mean on the training set, where $\mathbb{E}_{Tr}\left(\rho_{H_{\mathbf{z}}}\right)$ is short for $\mathbb{E}_{(\mathbf{x}_i,y_i)\in Tr}\left(\rho_{H_{\mathbf{z}}}\left(\mathbf{x}_i,y_i\right)\right)$, and

$$\rho_{H_{\mathbf{z}}}\left(\mathbf{x}_i,y_i\right)=\frac{1}{|\mathbf{z}|}\left(\sum_{t:h_t(\mathbf{x}_i)=y_i} z_t-\underset{j\neq y_i}{\arg\max}\sum_{t:h_t(\mathbf{x}_i)=j} z_t\right)$$

is the margin of $H_{\mathbf{z}}$ on instance $(\mathbf{x}_i,y_i)$. The second objective $\mathrm{Div}(Tr,H_{\mathbf{z}})$ is a measure of structural diversity of all the base learners selected by $\mathbf{z}$. It should be related to the structure of how the model makes predictions, and should be able to tell the difference between two base learners even if their predictions on the training set are the same. Considering it is a two-stage algorithm, we restricted the size $|\mathbf{z}|$ to $n/2$ so that half of the base learners are removed in the first stage, and the remaining half is left for further pruning in the second stage. The most suitable size for the first stage pruning may vary slightly depending on the task, but we currently fix it as the default setting. The rationale behind the design of the bi-objective optimization problem, including why these objectives are calculated on the training set, will be explained in Section 4.

As we have adopted multi-objective modeling in Eq. (1), the optimization problem may result in multiple optimal trade-off solutions, i.e., Pareto optimal solutions (Deb, 2014; Prajapati et al, 2023; Zhang et al, 2023b). And we choose one of the Pareto optimal solutions according to our needs.

Stage 2: Validation-error-based pruning. As we have already obtained a subset of base learners with good combination error distribution denoted by $\mathbf{z}$, in this stage, we conduct ensemble pruning on this subset solely based on validation error. We start with defining the $\preccurlyeq$ relation between two selector vectors.

Definition 1. Let $\mathbf{z}$ and $\mathbf{s}$ be two selector vectors. We say that $\mathbf{s}\preccurlyeq\mathbf{z}$ if and only if $H^{\mathbf{s}}\subseteq H^{\mathbf{z}}$, where $H^{\mathbf{z}}=\{h_t : z_t=1\}$.

With Definition 1, the single-objective optimization problem of validation-error-based pruning is represented as

$$\underset{\mathbf{s}\preccurlyeq\mathbf{z}}{\arg\max}\ \mathbb{E}_{(\mathbf{x},y)\in V}\ g(H_{\mathbf{s}}(\mathbf{x}),y)\ , \tag{2}$$

Algorithm 1 DEP

```
Input: Original ensemble $H = \{h_t\}_{t=1}^n$, training set $Tr$, validation set $V$
Output: Pruned ensemble $H_\mathbf{s}$
Randomly select solutions from $\{0,1\}^n$ to form the initial population $P_1$
for $t = 1$ : maximum #generations / 2 do
    Select solutions from $P_t$ by binary tournament selection to compose the mating pool
    Generate offspring population $P'$ by uniform crossover and bit-wise mutation
    Evaluate the objective vector($\mathbb{E}_{Tr}(\rho_{H_\mathbf{z}})$ , Div($Tr, H_\mathbf{z}$))$-\lambda*(|\mathbf{z}|-n/2)$ of $\mathbf{z} \in P'$
    Select next population $P_{t+1}$ from $P_t \cup P'$ based on non-dominated sorting and crowding distance
end for
Prune $H$ by selecting a Pareto optimal solution $\mathbf{z}$ from the final population to obtain $H_\mathbf{z}$
Randomly select solutions from $\{0,1\}^{\frac{n}{2}}$ to form the initial population $P_1$
for $t = 1$ : maximum #generations / 2 do
    Select solutions from $P_t$ by binary tournament selection to compose the mating pool
    Generate offspring population $P'$ by two-point crossover and bit-wise mutation
    Evaluate the objective validation error of $\mathbf{s} \in P'$
    Select next population $P_{t+1}$ from $P_t \cup P'$ based on fitness-based rank selection
end for
Prune $H_\mathbf{z}$ by selecting the best solution from the final population to obtain $H_\mathbf{s}$
```

where $\mathbf{s} \preccurlyeq \mathbf{z}$ means pruning on the $n/2$ base learners selected by $\mathbf{z}$.

The bi-objective optimization problem with equality constraint in Eq. (1) and the single-objective optimization problem in Eq. (2) can be solved using evolutionary algorithms (Deb et al, 2002; Zhou et al, 2019; Pan et al, 2023). The detailed algorithm design is illustrated in Algorithm 1.

4 Analysis for DEP

In this section, we first analyze the second stage of our DEP framework, the *validation-error-based* ensemble pruning. Theoretical results reveal that the key to good generalization performance of the second stage ensemble pruning is to obtain a subset of base learners with a good *combination distribution* in the first stage. Then we demonstrate that optimizing margin mean and structural diversity in the first stage benefits the combination distribution.

4.1 Combination distribution matters

A sub-ensemble can also be referred to as a *combination* of base learners. As the selector vector $\mathbf{s}$ can take 2^n possible values, we can sort it in order $(\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(2^n)})$. Let $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_{2^n}) : \mathcal{D}^{m_2} \mapsto \mathbb{R}^{2^n}$ denote the validation errors of the 2^n combinations, where $\phi_i = \mathbb{E}_{(\mathbf{x},y)\in V}\ g(H_{\mathbf{s}^{(i)}}(\mathbf{x}), y)$. Naturally, $\boldsymbol{\mu} = \mathbb{E}_{V\sim\mathcal{D}^{m_2}}[\boldsymbol{\phi}]$ is the corresponding generalization error vector, where $\mu_i = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\ g(H_{\mathbf{s}^{(i)}}(\mathbf{x}), y)$.

Then ensemble pruning can be viewed as a selection process $T : \mathcal{D}^{m_2} \mapsto \{1, \ldots, 2^n\}$ that picks one out of the 2^n combinations. For the conciseness of notion, we may use T as short for $T(V)$. During the validation-error-based ensemble pruning process, the selection rule T only has to do with validation error $\boldsymbol{\phi}$. Therefore, the validation error of the selected combination is mathematically of the form $\phi_{T(V)}(V)$. As $\boldsymbol{\phi}$ is only a noisy estimate of $\boldsymbol{\mu}$ due to the finite samples in V, and $T(V)$ is also a function of V, $\phi_{T(V)}(V)$ will be biased because of the selection procedure. It is called the *selection bias* (Russo and Zou, 2019). We analyze this selection bias in Theorem 2 and obtain the relationship between the expected generalization error μ_T and validation error ϕ_T of the pruned ensemble $H_{\mathbf{s}^{(T)}}$.

Theorem 2 (Combination error distribution is critical to generalization)**.** *Let V denote the validation set, $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_{2^n})$ denote the validation error of the 2^n combinations, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_{2^n})$ denote the corresponding generalization error. Let T denote the selection rule, and $\mathrm{P}(T \mid V)$ is the distribution of the selection rule output. Let E denote the validation error of a random combination. Then we have*

$$\mathop{\mathbb{E}}_{\substack{V\sim\mathcal{D}^{m_2},\\ T\sim\mathrm{P}(T|V)}} \mu_T \le \mathop{\mathbb{E}}_{\substack{V\sim\mathcal{D}^{m_2},\\ T\sim\mathrm{P}(T|V)}} \phi_T + \sqrt{\frac{-\mathop{\mathbb{E}}_{\substack{V\sim\mathcal{D}^{m_2},\\ T\sim P(T|V)}} \log_2 \mathrm{P}(E = \phi_T \mid V)}{2m_2}}\,. \tag{3}$$

where $\mathrm{P}(E = \phi_T \mid V) = \frac{\sum_{i=1}^{2^n} \mathbb{1}_{[\phi_i = \phi_T]}}{2^n}$.

Proof sketch. Firstly, we bound this selection bias in expectation using the mutual information $\mathbb{I}(T, \boldsymbol{\phi})$. Then we notice that, for those ϕ_is that take the same value, the selection rule T is equivalent to picking one of them randomly. When n is not too small ($n \gg \log_2 m_2$), the randomness in the selection process is considerable. Therefore we can bound the mutual information $\mathbb{I}(T, \boldsymbol{\phi})$ utilizing this randomness and obtain Theorem 2. The detailed proof is in the Appendix. □

Remark 1. Theorem 2 shows that the generalization performance not only depends on the validation error of the selected combination, but also depends on the *combination distribution*

$$\mathrm{P}(E = e \mid V) = \frac{\sum_{i=1}^{2^n} \mathbb{1}_{[\phi_i = e]}}{2^n}\,. \tag{4}$$

The value $\mathrm{P}(E = \phi_T \mid V)$ decides the expected generalization gap. A larger $\mathrm{P}(E = \phi_T \mid V)$ implies that the selection rule T picks a combination from a larger set randomly, hence the selection bias will be lower. For example, suppose all the combinations have the same validation error, i.e., $\mathrm{P}(E = \phi_T \mid V) = 1$, then the selection process is equivalent to completely random choice. In such cases, since we do not have a selection process on the validation set, the validation error is an unbiased estimate of the generalization error.

4.2 What is a better combination distribution

Consider selection rules Ts that can be modeled as one of the following:

(a) $\arg\min_{T} \phi_T,\ s.t.\, \mathrm{P}(E = \phi_T \mid V) > p_0$, where p_0 is a small probability mass. It models a search process that runs into the lower tail of the combination error distribution until the probability mass is too small.

(b) $T \in \{U : \phi_U = \varphi\}$, where φ is a small constant and can be related to the problem's difficulty. It models a search process that outputs a random combination that has fixed validation error φ.

For the above selection rules, we then discuss what kind of combination distribution is favorable for generalization. Notice that earlier we considered a fixed set of base learners H, and here we aim at manipulating the set of base learners to change the distribution, so we explicitly write the combination distribution as $\mathrm{P}(E = e \mid H, V)$.

Corollary 3 (Distribution that is heavier on the low error region leads to better generalization)**.** *Suppose the combination error distribution* $\mathrm{P}(E = e \mid H, V)$ *is unimodal for any base learner set* H *and* $V \sim \mathcal{D}^{m_2}$*. For two base learner sets* H *and* H'*, if there exists a small constant* e_0 *such that* $e_0 \geq \varphi$ *and* $\mathrm{P}(E = e_0 \mid H', V) \geq p_0$*, and for all* $e \leq e_0$ *and* $\mathrm{P}(E = e \mid H', V) > 0$, $\mathrm{P}(E = e \mid H, V) > \mathrm{P}(E = e \mid H', V)$ *holds, then the generalization error upper bound is lower for* H *than that for* H'*.*

4.3 Margin mean and structural diversity make better combination distribution

Generally speaking, finding a subset of learners with a lower mean of combination distribution and the same variance will satisfy the condition in Corollary 3. However, in finding such a favorable subset, there are two challenges. 1) For a combination distribution, the mean and variance affect each other. If the mean is lower, the variance tends to decrease because good base learners are more similar. Therefore, we should maximize the variance while minimizing the mean. 2) Once the validation information is utilized, the selection bias is introduced, and the combination distribution of the selected subset will be unchanged. Therefore, in the distribution optimization stage, we can only use the training set information.

In the bi-objective optimization problem as Eq. (1), the margin mean objective is for minimizing the distribution mean, and the structural diversity objective is for maximizing distribution variance. We will then explain in detail.

4.3.1 Margin mean for distribution mean

We first prove that optimizing $\mathbb{E}_{Tr}\left(\rho_{H_{\mathbf{z}}}\right)$, i.e., the margin mean of a single combination, is to optimize the average margin mean of all the combinations.

Proposition 4 (Optimizing ensemble margin mean is optimizing all the combinations)**.** *Suppose* $|\mathbf{z}|$ *is a constant. Then for binary classification,*

$$\arg\max_{\mathbf{z}} \mathbb{E}_{Tr}\left(\rho_{H_{\mathbf{z}}}\right) = \arg\max_{\mathbf{z}} \sum_{\mathbf{s} \preccurlyeq \mathbf{z}} \mathbb{E}_{Tr}\left(\rho_{H_{\mathbf{s}}}\right) \ .$$

According to the margin theory (Gao and Zhou, 2013), a voting classifier will generalize well if it has a better training margin mean. Therefore, maximizing $\mathbb{E}_{Tr}\left(\rho_{H_{\mathbf{z}}}\right)$ will lead to a lower mean of combination distribution. Note that although Proposition 4 proves only in the case of binary classification, we will show in experiments that optimizing margin mean is also effective in multi-class classification.

4.3.2 Structural diversity for distribution variance

Intuitively, if the base learners are diverse, there will be a certain number of very good combinations and very bad combinations. Even though for decades there has been no clear evidence that one combination with a large diversity must perform well (Kuncheva and Whitaker, 2003; Didaci et al, 2013), given that we are concerned with the distribution of all the combinations, existing a certain number of good combinations will suffice.

Most diversity measures only consider *behavioral diversity*, i.e., how the learners behave when making predictions (Sun and Zhou, 2018). However, as the learners are trained on the training data, we expect them to be quite similar in the perspective of behavioral diversity. On the other hand, the very limited differences in the labeling behavior have already been utilized in the margin mean objective. Therefore, we advocate using *structural* diversity as the diversity measure $\mathrm{Div}(Tr, H_{\mathbf{z}})$. Note that a specific design of structural diversity is required for the specific type of base learners.

5 Application to tree-based ensemble pruning

The general DEP framework in Section 3 is applicable to ensembles consisting of any kind of base learners. In this section, we provide a specific instantiation for tree-based ensemble pruning. The main consideration is on the design of diversity measure, and we design a new type of structural diversity for decision tree ensemble called feature contribution diversity.

5.1 Feature contribution diversity

Table 1 compares the discrimination ability of existing diversity measures and our feature contribution diversity. In the interpolation regime, each tree can classify all the training samples perfectly. In this regime, behavioral diversity cannot work. In the non-interpolation regime, behavioral diversity measures are applicable, but much less effective than those calculated on an independent validation set. Therefore, structural diversity is needed. However, existing structural diversity based on tree matching distance can only differentiate trees with different splitting features (Sun and Zhou, 2018). In contrast, our proposed feature contribution diversity is able to differentiate two trees in all the regimes.

Feature contribution has been proposed as an explaining tool of trees and forests (Palczewska et al, 2013; Saabas, 2014). It also has its application in debiased MDI feature importance (Li et al, 2019). Before introducing our definition of feature contribution diversity, we briefly show that the prediction of a tree h on $\mathbf{x}$ can be represented using the changes in label mean through each node on the decision path.

Table 1: Comparison between our feature contribution diversity and existing diversity measures. ‘$\surd$’ indicates the corresponding diversity measure is able to distinguish two trees under the given regime, while ‘$\times$’ indicates unable.

		Interpolation regime		Non-interpolation regime	
		Different tree structure	Same tree structure	Same splitting features Different splitting points	Different tree structures
Behavior diversity	Kappa (Margineantu and Dietterich, 1997; Martínez-Muñoz et al, 2008)	×	√	√	√
	Disagreement (Li et al, 2012)	×	√	√	√
	Complementarity (Martínez-Muñoz and Suárez, 2004)	×	√	√	√
Structural diversity	Tree matching distance (Sun and Zhou, 2018)	√	×	×	√
	Feature contribution (ours)	√	√	√	√

Suppose $\mathbf{x}$ goes through nodes $t_0, t_1, \ldots, t_l$, where t_0 is the root node and t_l is the leaf node. Then

$$h(\mathbf{x}) = u(t_0) + \sum_{0<i\leq l} \Delta u(t_i) \ , \tag{5}$$

where $\Delta u(t_i) = u(t_i) - u(t_{i-1})$, $u(t_i)$ is the label mean of training samples in node t_i. In the root node, $u(t_0)$ is the label mean of the whole tree's training data.

The feature contribution vector defined below is able to characterize the prediction of a sample with the tree structure information encoded.

Definition 5 (Feature contribution vector)**.** Let $s(t)$ denote the splitting feature of node t. Let K denote the number of features. Then $\forall(\mathbf{x}, y)$, the prediction made by tree h can be represented using a feature contribution vector $[f_{h,0}(\mathbf{x}), f_{h,1}(\mathbf{x}), \ldots, f_{h,K}(\mathbf{x})]$, where

$$f_{h,0}(\mathbf{x}) = u(t_0)[y] \ ; \quad f_{h,k}(\mathbf{x}) = \sum_{0<i\leq l:s(t_{i-1})=k} \Delta u(t_i)[y] \ , 1 \leq k \leq K \ . \tag{6}$$

Note that $f_{h,\cdot}(\mathbf{x})$, $u(t_0)$ and $\Delta u(t_i)$ are originally $|\mathcal{Y}|$-dimensional vectors, and we use the ground truth label y to turn them into scalers. $f_{h,0}(\mathbf{x})$ is introduced because different trees may use different training samples according to the data manipulation scheme. Figure 2 shows an example of the decision path and the feature contribution vector. We can see that so long as there is a difference in the label mean of any node, the feature contribution vector will exhibit the difference. We use the variation of the feature contribution vectors to characterize the diversity of a set of trees.

Definition 6 (Feature contribution diversity)**.** The feature contribution diversity with respect to a set of trees H on data set S is

$$\mathrm{Div}(S, H) \triangleq \frac{1}{(K+1)|S|} \sum_{k=0}^{K} \sum_{\mathbf{x}\in S} \sqrt{\frac{\sum_{h\in H} (f_{h,k}(\mathbf{x}) - \bar{f}_k(\mathbf{x}))^2}{|H|-1}} \ , \tag{7}$$

where $\bar{f}_k(\mathbf{x}) = \frac{1}{|H|} \sum_{h\in H} f_{h,k}(\mathbf{x})$.

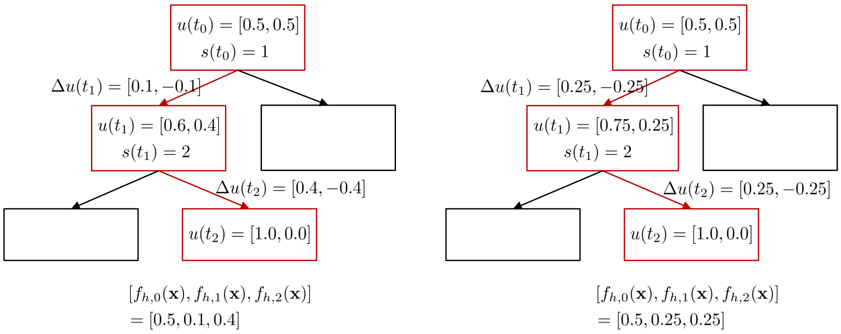

$u(t_0) = [0.5, 0.5]$
$s(t_0) = 1$
$\Delta u(t_1) = [0.1, -0.1]$
$u(t_1) = [0.6, 0.4]$
$s(t_1) = 2$
$\Delta u(t_2) = [0.4, -0.4]$
$u(t_2) = [1.0, 0.0]$
$[f_{h,0}(\mathbf{x}), f_{h,1}(\mathbf{x}), f_{h,2}(\mathbf{x})]$
$= [0.5, 0.1, 0.4]$
$u(t_0) = [0.5, 0.5]$
$s(t_0) = 1$
$\Delta u(t_1) = [0.25, -0.25]$
$u(t_1) = [0.75, 0.25]$
$s(t_1) = 2$
$\Delta u(t_2) = [0.25, -0.25]$
$u(t_2) = [1.0, 0.0]$
$[f_{h,0}(\mathbf{x}), f_{h,1}(\mathbf{x}), f_{h,2}(\mathbf{x})]$
$= [0.5, 0.25, 0.25]$

Fig. 2: An example of the feature contribution vector. It is a binary classification problem ($\mathcal{Y} = \{0, 1\}$), hence the label mean vector $u(t_i)$ has two dimensions. For each tree, the decision path of a sample with label $y = 1$ is marked in red. Even though the splitting features of each node and the predictions in leaf nodes are the same, the feature contribution vectors reveal the differences.

To speed up the calculation, we can calculate the feature contribution diversity on a subset of Tr instead of the whole training set.

6 Experiments

6.1 Experimental setup

Compared methods. We compare DEP to seven state-of-the-art ensemble pruning methods, including four ordering-based methods and three evolutionary optimization-based methods:

- Kappa (Margineantu and Dietterich, 1997; Martínez-Muñoz et al, 2008): An ordering-based method that considers κ statistic and validation error.
- CP (Martínez-Muñoz and Suárez, 2004): An ordering-based method that considers complementariness and validation error.
- DREP (Li et al, 2012): An ordering-based method that considers disagreement and validation error.
- MD (Martínez-Muñoz and Suárez, 2004): An ordering-based method that considers margin distance.
- EA (Zhou and Tang, 2003): A single-objective evolutionary algorithm that minimizes the validation error only.
- PEP (Qian et al, 2015): A bi-objective evolutionary algorithm that minimizes the validation error and ensemble size simultaneously.
- MDEP (Wu et al, 2022): A three-objective evolutionary algorithm that optimizes the validation error, margin distribution, and ensemble size.

The compared methods also include a baseline “All” that uses all the base learners.

Configurations. Each dataset is evenly and randomly partitioned into training, validation, and test sets, and this partitioning process is repeated 30 times independently. For each partition, 100 base learners (decision trees) are trained on the training set, and the hyperparameters of each base learner are randomly chosen from the predefined hyperparameter set: sampling rate in $\{1.0, 0.8, 0.6, 0.4\}$, number of candidate features in each node in $\{\text{all}, \text{sqrt}, 1\}$, leaf size in $\{1, 5, 20, 50\}$, splitting rule in $\{\text{best}, \text{random}\}$. Each pruning method is performed on each partition and the average and standard deviation of performance are reported. The hyperparameter ρ of DREP is selected from $\{0.2, 0.25, \ldots, 1.0\}$ on the validation set, and the hyperparameter p of MD is set to 0.075. The total number of fitness evaluations used by EA, PEP, and MDEP is set to $50,000$, with a population size of 100 and 500 generations. For DEP (and variants), the distribution optimization and the validation-error-based pruning each take $25,000$ evaluations, which means the total number of evaluations is the same as EA, PEP, and MDEP.

Datasets. We conduct experiments on 20 binary and 10 multi-class classification data sets from the UCI repository (Dua and Graff, 2017). Several binary classification datasets are derived from multi-class datasets as in previous works (Qian et al, 2015; Wu et al, 2022): *letter-ah* is based on *letter* data set and classifies 'a' against 'h', as do *letter-br* and *letter-oq*; *optdigits* classifies '01234' against '56789'; *satimage-12v57* is based on *satimage* and classifies labels '1' and '2' against '5' and '7', as does *satimage-2v5*; *vehicle-bo-vs* is based on the *vehicle* data set and classifies 'bus' and 'opel' against 'van' and 'saab', as does *vehicle-b-v*.

6.2 Effectiveness of optimizing combination distribution

We first show that optimizing margin mean and structural diversity brings about better combination distribution. We compare the validation error distribution of 10000 random combinations from four sets of base learners. The four sets are generated by choosing 50 with different strategies out of 100 decision trees trained on the 6-dim 2-class Gaussian quantile dataset (Hastie et al, 2009). The four strategies are 1) Random: randomly chosen 50 base learners. 2) Margin: single-objective evolutionary optimization of $\mathbb{E}_{Tr}(\rho_{H_{\mathbf{z}}})$. 3) Diversity: single-objective evolutionary optimization of $\mathrm{Div}(V, H_{\mathbf{z}})$. 4) 2-obj: bi-objective optimization of margin and feature contribution diversity, respectively. As we can see in Figure 3, 2-obj achieves the best combination error distribution, with the largest probability density in the low error region compared to the other three strategies. Margin and Diversity that optimize $\mathrm{Mean}_{Tr}(\rho_{H_{\mathbf{z}}})$ and $\mathrm{Div}(V, H_{\mathbf{z}})$ respectively are only slightly better than random choice.

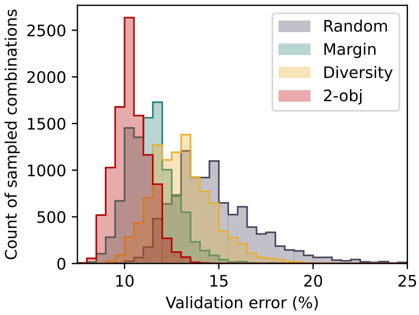
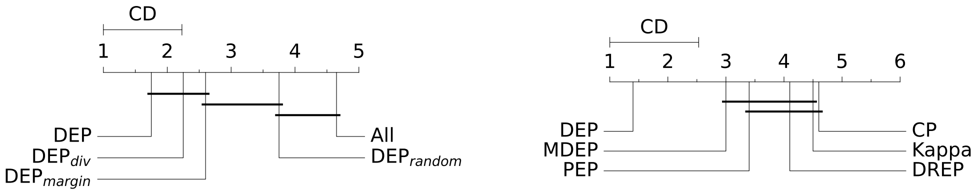


Fig. 3: Validation error distribution of 10000 random combinations from four sets of base learners chosen by different strategies.



(a) Ablation studies.

(b) DEP compared to SOTA methods.

Fig. 4: Friedman-Nemenyi test at significance level 0.1. If two algorithms are connected by a CD (critical difference) line, then there is no significant difference between them. (a) Friedman-Nemenyi test for DEP and variants that use other strategies in the distribution optimization stage, and All. (b) Friedman-Nemenyi test for the six top-ranked methods in Tabel 2.

We then show that better combination distribution leads to better ensemble pruning performance. We compare DEP with the aforementioned variants that use other strategies in the distribution optimization stage as the ablation study. The variants are named DEP$_{\text{random}}$, DEP$_{\text{margin}}$ and DEP$_{\text{div}}$, respectively. Figure 4a shows the Friedman-Nemenyi test (Demšar, 2006) of the ablation study on 20 binary data sets. It can be observed that DEP achieves the best average rank while DEP$_{\text{random}}$ gets the worst. This experimental result confirms our theoretical analysis that optimizing combination distribution through margin mean and structural diversity will benefit the generalization performance.

6.3 DEP vs. state-of-the-art pruning methods

Table 2 compares DEP to other state-of-the-art methods. We can see that DEP has the best average rank among all the eight methods. And it is significantly better than Kappa, CP, DREP, MD, EA, and All on most data sets by the Wilcoxon rank-sum test (Wilcoxon, 1945) with confidence level 0.1. We then conduct the Friedman-Nemenyi test for the six top-ranked methods as shown in Figure 4b. We can observe that DEP has a significant advantage in generalization performance over the others.

We can also observe that MDEP and PEP are two highly competitive methods. According to Theorem 2, the generalization bound is decided by the validation error and the generalization gap (which is determined by the combination distribution). Although MDEP and PEP do not optimize combination distribution, they are able to find solutions with a better validation error, therefore they can achieve good test performance as well. DEP optimizes combination distribution, but on the other hand, it limits the search space of the validation-error-based pruning. Note that although DEP is not significantly better than MDEP and PEP on every data set in Table 2, it can be seen from Figure 4b that DEP is significantly better than MDEP and PEP on average ranking by the Friedman-Nemenyi test, which demonstrates the effectiveness of our optimization of combination distribution.

Table 2: Test error (mean±std.) of the compared methods on 20 binary data sets. An entry is marked with a bullet '•' (or circle '∘') if DEP is significantly better (or worse) than the corresponding method based on the Wilcoxon rank-sum test with confidence level 0.1. For each data set, the entry with the lowest average error is bolded. The average ensemble size of each method is summarized in the last row.

Dataset	DEP	Kappa	CP	DREP	MD	EA	PEP	MDEP	All
australian	**.132±.016**	.141±.017•	.141±.017•	.135±.018	.139±.020	.136±.017	.136±.015	.138±.014	.135±.015
breast	**.035±.010**	.040±.010•	.045±.011•	.044±.012•	.057±.020•	.038±.011•	.041±.010•	.041±.010•	.047±.010•
bupa	.319±.033	.337±.041•	.324±.052•	.328±.042	.339±.044•	.331±.037•	**.318±.041**	.328±.040	.392±.024•
diabetes	**.244±.024**	.252±.023•	.248±.023	.246±.025	.255±.020•	.246±.021	.244±.019	.250±.024	.270±.020•
german	**.257±.019**	.260±.019	.265±.019•	.258±.019	.267±.021•	.270±.014•	.263±.017•	.263±.016•	.292±.007•
haberman	.266±.025	.280±.027•	.279±.040	.279±.028•	.288±.043•	.265±.014	.268±.024	.274±.027	**.265±.001**
heart-statlog	**.176±.033**	.182±.037	.184±.038	.181±.037	.222±.042•	.197±.033•	.187±.043	.183±.041	.290±.060•
ionosphere	**.088±.028**	.098±.025•	.093±.028	.101±.029•	.119±.033•	.103±.022•	.090±.024	.090±.024	.201±.040•
letter-AH	**.009±.006**	.010±.006	.012±.005•	.015±.005•	.024±.007•	.016±.007•	.012±.006•	.009±.005	.031±.011•
letter-BR	**.040±.008**	.043±.010•	.041±.009	.042±.009	.062±.013•	.045±.009•	.042±.010	.041±.010	.062±.015•
letter-OQ	**.035±.009**	.043±.012•	.039±.010	.041±.008•	.063±.017•	.044±.010•	.039±.010•	.039±.007•	.060±.016•
optdigits-b	**.033±.005**	.041±.006•	.035±.004•	.033±.005	.051±.007•	.037±.005•	.034±.006	.033±.005	.051±.006•
phishing	.040±.004	.044±.003•	.044±.004•	.040±.004	.048±.004•	.042±.004•	**.040±.003**	.041±.003	.052±.004•
satimage-12v57	**.024±.004**	.025±.005	.027±.005•	.026±.004•	.037±.007•	.028±.004•	.026±.005•	.026±.005•	.036±.006•
satimage-25	.024±.008	.024±.008	**.024±.009**	.026±.008	.034±.009•	.026±.010	.025±.008	.024±.007	.035±.009•
sonar	.222±.042	.243±.051•	.246±.038•	.232±.059	.280±.071•	.253±.046•	**.217±.036**	.228±.057	.394±.062•
spambase	.058±.006	.066±.006•	.061±.006•	.060±.006	.091±.014•	.063±.007•	.059±.006	**.058±.005**	.089±.013•
vehicle-bo-vs	.232±.023	.250±.023•	.240±.024	.233±.019	.252±.019•	.236±.018	.232±.020	**.227±.023**	.249±.023•
vehicle-b-v	**.015±.009**	.020±.012•	.023±.015•	.033±.020•	.030±.016•	.021±.012•	.022±.016•	.020±.012•	.047±.033•
vote	**.048±.012**	.051±.013•	.061±.016•	.059±.015•	.049±.012	.052±.014•	.056±.015•	.055±.013•	.055±.014•
count of the best	13	0	1	0	0	0	3	2	1
average rank	1.50	5.20	5.25	4.60	7.95	5.60	3.80	3.45	7.65
average size	21.4	27.0	16.3	16.1	36.2	43.5	17.2	9.8	100

6.4 Multi-class extension of DEP

We also compare DEP with the state-of-the-art methods on 10 multi-class UCI data sets. Although the analysis in Proposition 4 only involves binary classification, the calculation of margin mean and structural diversity in DEP is directly applicable to multi-class problems. Note that the compared methods Kappa, CP, and DREP are initially designed for binary classification. To extend them for multi-class classification, we generalize their "equal" and "unequal" tests to apply to multiple classes. The detailed results are reported in Table 3. It shows that DEP has the best average rank (1.60), while the runner-up among the compared methods has an average rank of 2.90. By the Wilcoxon rank-sum test with confidence level 0.1, DEP is significantly better than Kappa, CP, DREP, MD, EA, and All on most data sets, and is never significantly worse compared with any other method, since no '∘' appears in Table 3.

6.5 Further improvement of DEP

In the previous experiments, we have verified that DEP is able to achieve significantly better generalization performance. However, it only has a medium ensemble size. Among the compared methods, we notice that MDEP has the second-best generalization performance just after DEP while possessing a remarkably small ensemble size. This can be attributed to its use of a unique objective, the margin ratio (Lyu

Table 3: Test error (mean±std.) of the compared methods on 10 multi-class data sets. An entry is marked with a bullet '•' (or circle '○') if DEP is significantly better (or worse) than the corresponding method based on the Wilcoxon rank-sum test with confidence level 0.1. For each data set, the entry with the lowest average error is bolded. The average ensemble size of each method is summarized in the last row.

Dataset	DEP	Kappa	CP	DREP	MD	EA	PEP	MDEP	All
glass	.305±.046	.352±.040●	.308±.052	.354±.054●	.448±.077●	.330±.046●	.317±.050	**.302±.051**	.567±.089●
heart	**.424±.030**	.429±.034	.436±.031●	.437±.027●	.460±.035●	.451±.020●	.431±.026●	.426±.038	.469±.004●
iris	**.051±.030**	.056±.024	.058±.026	.077±.060●	.093±.090●	.057±.030	.061±.036	.061±.027	.222±.210●
libras	.349±.050	.461±.082●	.358±.045	.441±.060●	.445±.067●	.358±.045	**.347±.045**	.360±.046	.457±.081●
segment	.039±.006	.043±.008●	.040±.007●	.054±.011●	.054±.011●	.042±.008●	.039±.006	**.038±.006**	.056±.011●
soybean	**.092±.018**	.143±.025●	.100±.020●	.143±.087●	.209±.066●	.101±.017●	.095±.018	.096±.017	.284±.058●
vehicle	.274±.019	.307±.027●	.281±.026●	.312±.030●	.311±.022●	.283±.019●	.279±.021	**.272±.024**	.318±.019●
vowel	.171±.028	.260±.035●	.169±.027	.261±.034●	.261±.041●	.181±.027●	.176±.029	**.167±.022**	.274±.040●
wine	**.038±.025**	.058±.028●	.066±.032●	.129±.051●	.165±.129●	.060±.034●	.058±.027●	.068±.029●	.389±.157●
zoo	**.062±.041**	.064±.044	.069±.033	.084±.044●	.302±.183●	.125±.108●	.072±.039	.068±.037	.559±.060●
count of the best	5	0	0	0	0	0	1	4	0
average rank	1.60	4.90	3.80	6.90	7.70	5.00	3.30	2.90	8.90
average size	21.4	34.3	13.3	23.1	35.8	39.6	17.4	9.7	100

et al, 2019; Wu et al, 2022)

$$\rho_V^{\text{ratio}}\left(H_{\boldsymbol{s}}\right)=\sqrt{\frac{\operatorname{Var}_V\left(\rho_{H_{\boldsymbol{s}}}\right)}{\operatorname{Mean}_V^2\left(\rho_{H_{\boldsymbol{s}}}\right)}}=\sqrt{\frac{m_2\sum_{i\neq j}\left(\rho_{H_{\boldsymbol{s}}}\left(\boldsymbol{x}_i,y_i\right)-\rho_{H_{\boldsymbol{s}}}\left(\boldsymbol{x}_j,y_j\right)\right)^2}{2(m_2-1)\left(\sum_{i=1}^{m_2}\rho_{H_{\boldsymbol{s}}}\left(\boldsymbol{x}_i,y_i\right)\right)^2}}\,.$$

To further improve DEP, we may add the margin ratio as another objective aside from the validation error in the second stage of the DEP algorithm, and name this variant DEP$_\rho$. In Table 4, we compare DEP$_\rho$ to the original version of DEP, which has the best test error, and MDEP, which has the second-best test error and the best ensemble size. We can observe that DEP$_\rho$ has an even slightly better average rank in test error than DEP, and there is also a remarkable improvement in ensemble size than DEP. Compared with the other methods in Table 2, we can conclude that DEP$_\rho$ enjoys significantly better test error than other state-of-the-art methods, while having an ensemble size that is better than all the state-of-the-art methods except for MDEP.

7 Conclusion

In this paper, we propose a margin distribution and structural diversity guided ensemble pruning framework called Decoupled Ensemble Pruning (DEP). It has two stages. In the first stage, it selects a subset of learners with good margin mean and structural diversity on the training set. In the second stage, it conducts validation-error-based ensemble pruning on this subset. Theoretical analysis shows that optimizing margin mean and structural diversity on the training set benefits the combination distribution, thus benefiting the validation-error-based ensemble pruning in the second stage. Experimental results verify the effectiveness of our DEP method. In addition, we propose feature contribution diversity, a new measure of structural diversity that is specifically applicable to decision trees, which may be of independent interest for tree-based ensemble literature.

Table 4: Test error (mean±std.) of the compared methods on 20 binary data sets. An entry is marked with a bullet '•' (or circle '○') if DEP$_\rho$ is significantly better (or worse) than the corresponding method based on the Wilcoxon rank-sum test with confidence level 0.1. For each data set, the entry with the lowest average error is bolded. The average ensemble size of each method is summarized in the last row.

Dataset	DEP$_\rho$	DEP	MDEP
australian	0.134±0.017	**0.132±0.016**	0.138±0.014
breast	0.036±0.009	**0.035±0.010**	0.041±0.010●
bupa	**0.315±0.040**	0.319±0.033	0.328±0.040
diabetes	0.254±0.018	**0.244±0.024**○	0.250±0.024
german	0.261±0.016	**0.257±0.019**	0.263±0.016
haberman	0.279±0.024	**0.266±0.025**○	0.274±0.027
heart-statlog	**0.175±0.038**	0.176±0.033	0.183±0.041
ionosphere	**0.080±0.024**	0.088±0.028	0.090±0.024●
letter-AH	**0.008±0.003**	0.009±0.006	0.009±0.005
letter-BR	**0.036±0.008**	0.040±0.008●	0.041±0.010●
letter-OQ	0.035±0.007	**0.035±0.009**	0.039±0.007●
optdigits-b	**0.031±0.005**	0.033±0.005	0.033±0.005
phishing	**0.040±0.003**	0.040±0.004	0.041±0.003
satimage-12v57	0.025±0.004	**0.024±0.004**	0.026±0.005
satimage-25	**0.023±0.007**	0.024±0.008	0.024±0.007
sonar	**0.220±0.038**	0.222±0.042	0.228±0.057
spambase	0.058±0.006	0.058±0.006	**0.058±0.005**
vehicle-bo-vs	**0.226±0.019**	0.232±0.023	0.227±0.023
vehicle-b-v	**0.014±0.011**	0.015±0.009	0.020±0.012●
vote	0.049±0.014	**0.048±0.012**	0.055±0.013●
count of the best	11	8	1
average rank	1.55	1.70	2.75
average size	13.3	21.4	9.8

Recently there is a proposal called "Learnware" which advocates exploiting all kinds of trained machine learning models, submitted by developers all over the world to a *learnware market*, to enable future users not to build their own machine learning application from scratch, without disclosing the data of developers and users (Zhou, 2016). The key is a carefully designed *Learnware specification* which enables the identification and reassembly of helpful models without data disclosure. Ensemble mechanisms are very useful in reassembling the identified models to tackle users' tasks, even for tasks never considered by developers of the original models (Zhou and Tan, 2023). It can be expected that better ensemble pruning methods can help build better learnwares consisting of fewer models.

Acknowledgments. This research was supported by the National Science Foundation of China (62250069, 62022039).

Declarations

- Funding: This work is funded by the National Science Foundation of China (62250069, 62022039).

- Conflict of interest/Competing interests: The authors declare that they have no conflict of interest.
- Ethics approval: Not applicable.
- Consent to participate: Not applicable.
- Consent for publication: Not applicable.
- Availability of data and materials: The data used in our experiments are all public.
- Code availability: The code will be provided after the paper is accepted.
- Authors' contributions: Yi-Xiao He conceived and developed the procedure and wrote the manuscript. Yu-Chang Wu performed the experiments. Zhi-Hua Zhou and Chao Qian conceived the study and were in charge of overall direction and planning. All authors discussed the results and contributed to the final manuscript. All authors approved the final version of the manuscript.

Appendix A Omitted proofs

A.1 Proof of Theorem 2

Proof. Since $g(f(\mathbf{x}), y) = \mathbb{1}_{[f(\mathbf{x}) \neq y]}$, then $\forall \mathbf{s} \in \{0,1\}^n$, $g(h_{\mathbf{s}}(\mathbf{x}), y) - \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} g(h_{\mathbf{s}}(\mathbf{x}), y)$ is 1/2-sub-Gaussian with respect to $(\mathbf{x}, y) \in \mathcal{D}$. Then, $\phi_i - \mu_i$ is $1/2\sqrt{m_2}$-sub-Gaussian for each $i \in \{1, \ldots, 2^n\}$. Then according to Russo and Zou (2019), we have

$$\left| \underset{\substack{V\sim\mathcal{D},\\ T\sim P(T|V)}}{\mathbb{E}} [\phi_T - \mu_T] \right| \leq \sqrt{\frac{\mathbb{I}(T;\boldsymbol{\phi})}{2m_2}} \ .$$

We have assumed $\mathbf{s}^{(T)}$ is selected based on the validation performance $\boldsymbol{\phi}$. However, there might be more than one solution that has the same validation error as ϕ_T. In fact, these solutions have an equal probability to be chosen, since they are indistinguishable in terms of the validation error. The exact number of these indistinguishable solutions is $\sum_{i=1}^{2^n} \mathbb{1}_{[\phi_i = \phi_T]}$. Therefore, the conditional entropy

$$\mathbb{H}(T \mid \boldsymbol{\phi}) = \log_2 \sum_{i=1}^{2^n} \mathbb{1}_{[\phi_i = \phi_T]} \ .$$

It follows that

$$\begin{aligned}
\mathbb{I}(T;\boldsymbol{\phi}) &= \mathbb{H}(T) - \mathbb{H}(T \mid \boldsymbol{\phi}) \\
&= \log_2 2^n - \underset{\substack{V\sim\mathcal{D}^{m_2},\\ T\sim \mathrm{P}(T|V)}}{\mathbb{E}} \log_2 \sum_{i=1}^{2^n} \mathbb{1}_{[\phi_i = \phi_T]} \\
&= - \underset{\substack{V\sim\mathcal{D}^{m_2},\\ T\sim \mathrm{P}(T|V)}}{\mathbb{E}} \log_2 \frac{\sum_{i=1}^{2^n} \mathbb{1}_{[\phi_i = \phi_T]}}{2^n} \ .
\end{aligned}$$

Consequently, we have Eq. (3). □

A.2 Proof of Corollary 3

Proof. For selection rule (a), we have that for any H, V and the corresponding $\boldsymbol{\phi}$ and $\boldsymbol{\mu}$,

$$\mathop{\mathbb{E}}_{\substack{V\sim\mathcal{D}^{m_2},\\ T\sim\mathrm{P}(T|V)}} \mu_T \leq \mathop{\mathbb{E}}_{\substack{V\sim\mathcal{D}^{m_2},\\ T\sim\mathrm{P}(T|V)}} \phi_T + \sqrt{\frac{-\mathop{\mathbb{E}}_{V\sim\mathcal{D}^{m_2}} \log_2 p_0}{2m_2}} \ .$$

As $\mathrm{P}(E=e \mid H,V)$ is unimodal, e_0 is a small constant that it must be on the left side of the modal, therefore for $e<e_0$, $\mathrm{P}(E=e)$ is monotonically increasing. As $\mathrm{P}(E=e_0 \mid H',V) \geq p_0$, the solution found by selection rule (a) is no larger than e_0. Since $\mathrm{P}(E=e \mid H,V) > \mathrm{P}(E=e \mid H',V)$, according to selection rule (a), we have $\phi_T < \phi'_T$. Therefore, the upper bound for H is lower than that for H'.

For selection rule (b), $\phi_T = \phi'_T = \varphi$. As $\varphi \leq e_0$, then $\mathrm{P}(E=\varphi|H,V) > \mathrm{P}(E=\varphi|H',V)$. Therefore, according to Eq. (3), the upper bound for H is lower. □

A.3 Proof of Proposition 4

Proof. For binary classification,

$$\rho_{H_{\mathbf{s}}}\left(\boldsymbol{x}_i, y_i\right) = \frac{1}{|\mathbf{s}|}\left(\sum_{t:h_t(\mathbf{x}_i)=y_i} s_t - \arg\max_{j\neq y_i} \sum_{t:h_t(\mathbf{x}_i)=j} s_t\right) \ .$$

Therefore,

$$\begin{aligned}
\mathbb{E}_{Tr}\left(\rho_{H_{\mathbf{z}}}\right) &= \frac{1}{m_1}\sum_{i=1}^{m_1} \rho_{H_{\mathbf{z}}}\left(\mathbf{x}_i, y_i\right) \\
&= \frac{1}{m_1}\sum_{i=1}^{m_1} \frac{1}{|\mathbf{z}|}\left(\sum_{t:y_i=h_t(\boldsymbol{x}_i)} z_t - \sum_{t:y_i\neq h_t(\boldsymbol{x}_i)} z_t\right) \\
&= \frac{1}{m_1|\mathbf{z}|}\sum_{i=1}^{m_1}\sum_{t=1}^{n} z_t \cdot \left(\mathbb{1}_{[y_i=h_t]} - \mathbb{1}_{[y_i\neq h_t]}\right) \ .
\end{aligned}$$

Meanwhile, according to Definition 1,

$$\begin{aligned}
&\sum_{\mathbf{s}\preccurlyeq\mathbf{z}} \mathbb{E}_{Tr}\left(\rho_{H_{\mathbf{s}}}\right) = \sum_{\mathbf{s}\preccurlyeq\mathbf{z}} \frac{1}{m_1}\sum_{i=1}^{m_1} \rho_{H_{\mathbf{s}}}\left(\mathbf{x}_i, y_i\right) \\
=&\sum_{\mathbf{s}\preccurlyeq\mathbf{z}} \frac{1}{m_1}\sum_{i=1}^{m_1} \frac{1}{|\mathbf{s}|}\left(\sum_{t:y_i=h_t(\mathbf{x}_i)} s_t - \sum_{t:y_i\neq h_t(\mathbf{x}_i)} s_t\right) \\
=&\frac{1}{m_1}\sum_{i=1}^{m_1}\sum_{\mathbf{s}\preccurlyeq\mathbf{z}} \frac{1}{|\mathbf{s}|}\left(\sum_{t:y_i=h_t(\mathbf{x}_i)} s_t - \sum_{t:y_i\neq h_t(\mathbf{x}_i)} s_t\right)
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{m_1} \sum_{i=1}^{m_1} \sum_{\mathbf{s} \preccurlyeq \mathbf{z}} \frac{1}{|\mathbf{s}|} \sum_{t=1}^{n} s_t \cdot \left(\mathbb{1}_{[y_i = h_t]} - \mathbb{1}_{[y_i \neq h_t]} \right) \\
&= \frac{1}{m_1} \sum_{i=1}^{m_1} \Bigg(\sum_{\mathbf{s}:\mathbf{s} \preccurlyeq \mathbf{z}, |\mathbf{s}|=1} \sum_{t=1}^{n} \left(\mathbb{1}_{[y_i = h_t]} - \mathbb{1}_{[y_i \neq h_t]} \right) + \sum_{\mathbf{s}:\mathbf{s} \preccurlyeq \mathbf{z}, |\mathbf{s}|=2} \frac{1}{2} \sum_{t=1}^{n} C^{1}_{|\mathbf{z}|-1} \left(\mathbb{1}_{[y_i = h_t]} - \mathbb{1}_{[y_i \neq h_t]} \right) \\
&\quad + \sum_{\mathbf{s}:\mathbf{s} \preccurlyeq \mathbf{z}, |\mathbf{s}|=3} \frac{1}{3} \sum_{t=1}^{n} C^{2}_{|\mathbf{z}|-1} \left(\mathbb{1}_{[y_i = h_t]} - \mathbb{1}_{[y_i \neq h_t]} \right) + \dots \\
&\quad + \sum_{\mathbf{s}:\mathbf{s} \preccurlyeq \mathbf{z}, |\mathbf{s}|=|\mathbf{z}|-1} \frac{1}{|\mathbf{z}|-1} \sum_{t=1}^{n} C^{|\mathbf{z}|-2}_{|\mathbf{z}|-1} \left(\mathbb{1}_{[y_i = h_t]} - \mathbb{1}_{[y_i \neq h_t]} \right) \\
&\quad + \sum_{\mathbf{s}:\mathbf{s} \preccurlyeq \mathbf{z}, |\mathbf{s}|=|\mathbf{z}|} \frac{1}{|\mathbf{z}|} \sum_{t=1}^{n} C^{|\mathbf{z}|-1}_{|\mathbf{z}|-1} \left(\mathbb{1}_{[y_i = h_t]} - \mathbb{1}_{[y_i \neq h_t]} \right) \Bigg) \\
&= \frac{1}{m_1} \left(1 \cdot C^{0}_{|\mathbf{z}|-1} + \frac{C^{1}_{|\mathbf{z}|-1}}{2} + \frac{C^{2}_{|\mathbf{z}|-1}}{3} + \dots + \frac{C^{|\mathbf{z}|-2}_{|\mathbf{z}|-1}}{|\mathbf{z}|-1} + \frac{C^{|\mathbf{z}|-1}_{|\mathbf{z}|-1}}{|\mathbf{z}|} \right) \sum_{i=1}^{m_1} \sum_{t=1}^{n} \left(\mathbb{1}_{[y_i = h_t]} - \mathbb{1}_{[y_i \neq h_t]} \right) \ .
\end{aligned}
$$

As $|\mathbf{z}|$ is fixed to be a constant, then

$$
c = |\mathbf{z}| \left(1 \cdot C^{0}_{|\mathbf{z}|-1} + \frac{C^{1}_{|\mathbf{z}|-1}}{2} + \frac{C^{2}_{|\mathbf{z}|-1}}{3} + \dots + \frac{C^{|\mathbf{z}|-2}_{|\mathbf{z}|-1}}{|\mathbf{z}|-1} + \frac{C^{|\mathbf{z}|-1}_{|\mathbf{z}|-1}}{|\mathbf{z}|} \right)
$$

is a constant. Therefore we have

$$
\mathbb{E}_{Tr}\left(\rho_{H_{\mathbf{z}}}\right) = \frac{1}{c} \sum_{\mathbf{s} \preccurlyeq \mathbf{z}} \mathbb{E}_{Tr}\left(\rho_{H_{\mathbf{s}}}\right) \ .
$$

Then it follows

$$
\arg\max_{\mathbf{z}} \mathbb{E}_{Tr}\left(\rho_{H_{\mathbf{z}}}\right) = \arg\max_{\mathbf{z}} \sum_{\mathbf{s} \preccurlyeq \mathbf{z}} \mathbb{E}_{Tr}\left(\rho_{H_{\mathbf{s}}}\right) \ .
$$

□

References

Bian Y, Chen H (2021) When does diversity help generalization in classification ensembles? IEEE Transactions on Cybernetics 52(9):9059–9075

Bian Y, Wang Y, Yao Y, et al (2020) Ensemble pruning based on objection maximization with a general distributed framework. IEEE Transactions on Neural Networks and Learning Systems 31(9):3766–3774

Deb K (2014) Multi-objective optimization. In: Search Methodologies. p 403–449

Deb K, Pratap A, Agarwal S, et al (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2):182–197

Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7:1–30

Didaci L, Fumera G, Roli F (2013) Diversity in classifier ensembles: Fertile concept or dead end? In: Proceedings of the 11th International Workshop on Multiple Classifier Systems, pp 37–48

Dietterich TG (2000) Ensemble methods in machine learning. In: International Workshop on Multiple Classifier Systems, pp 1–15

Dua D, Graff C (2017) UCI machine learning repository. URL http://archive.ics.uci.edu/ml

Durrant B, Lim N (2020) A diversity-aware model for majority vote ensemble accuracy. In: Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, pp 4078–4087

Ganaie MA, Hu M, Malik A, et al (2022) Ensemble deep learning: A review. Engineering Applications of Artificial Intelligence 115:105151

Gao W, Zhou ZH (2013) On the doubt about margin explanation of boosting. Artificial Intelligence 203:1–18

Giacinto G, Roli F, Fumera G (2000) Design of effective multiple classifier systems by clustering of classifiers. In: Proceedings of the 15th International Conference on Pattern Recognition, pp 160–163

Guo L, Boukir S (2013) Margin-based ordered aggregation for ensemble pruning. Pattern Recognition Letters 34(6):603–609

Hastie T, Rosset S, Zhu J, et al (2009) Multi-class adaboost. Statistics and its Interface 2(3):349–360

Kuncheva LI, Whitaker CJ (2003) Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Machine Learning 51(2):181–201

Lazarevic A, Obradovic Z (2001) Effective pruning of neural network classifier ensembles. In: International Joint Conference on Neural Networks, pp 796–801

Li N, Yu Y, Zhou ZH (2012) Diversity regularized ensemble pruning. In: Proceedings of the 12th European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, pp 330–345

Li X, Wang Y, Basu S, et al (2019) A debiased mdi feature importance measure for random forests. In: Advances in Neural Information Processing Systems 32, pp 8047–8057

Liu FT, Ting KM, Zhou ZH (2008) Isolation forest. In: Proceedings of the 8th IEEE International Conference on Data Mining, pp 413–422

Lu J, Song E, Ghoneim A, et al (2020) Machine learning for assisting cervical cancer diagnosis: An ensemble approach. Future Generation Computer Systems 106:199–205

Lu Z, Wu X, Zhu X, et al (2010) Ensemble pruning via individual contribution ordering. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 871–880

Lyu SH, Yang L, Zhou ZH (2019) A refined margin distribution analysis for forest representation learning. Advances in Neural Information Processing Systems 32 pp 5531–5541

Lyu SH, Chen YH, Zhou ZH (2022) A region-based analysis for the feature concatenation in deep forests. Chinese Journal of Electronics 31(6):1072–1080

Margineantu DD, Dietterich TG (1997) Pruning adaptive boosting. In: Proceedings of the 14th International Conference on Machine Learning, pp 211–218

Martínez-Muñoz G, Suárez A (2004) Aggregation ordering in bagging. In: Proceeding of the 14th International Conference on Artificial Intelligence and Applications, pp 258–263

Martínez-Muñoz G, Suárez A (2006) Pruning in ordered bagging ensembles. In: Proceedings of the 23rd International Conference on Machine Learning, pp 609–616

Martínez-Muñoz G, Hernández-Lobato D, Suárez A (2008) An analysis of ensemble pruning techniques based on ordered aggregation. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(2):245–259

Palczewska A, Palczewski J, Robinson RM, et al (2013) Interpreting random forest classification models using a feature contribution method. In: Integration of Reusable Systems, pp 193–218

Pan S, Ma Y, Wang Y, et al (2023) An improved master-apprentice evolutionary algorithm for minimum independent dominating set problem. Frontiers of Computer Science 17(4):174326

Partalas I, Tsoumakas G, Vlahavas I (2010) An ensemble uncertainty aware measure for directed hill climbing ensemble pruning. Machine Learning 81:257–282

Prajapati A, Parashar A, Rathee A (2023) Multi-dimensional information-driven many-objective software remodularization approach. Frontiers of Computer Science 17(3):173209

Qian C, Yu Y, Zhou ZH (2015) Pareto ensemble pruning. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence

Russo D, Zou J (2019) How much does your data exploration overfit? controlling bias via information usage. IEEE Transactions on Information Theory 66(1):302–323

Saabas A (2014) Interpreting random forests. https://blog.datadive.net/interpreting-random-forests

Schaefer G, Krawczyk B, Celebi ME, et al (2014) An ensemble classification approach for melanoma diagnosis. Memetic Computing 6:233–240

Sun T, Zhou ZH (2018) Structural diversity for decision tree ensemble learning. Frontiers of Computer Science 12:560–570

Tang K, Suganthan PN, Yao X (2006) An analysis of diversity measures. Machine Learning 65:247–271

Wilcoxon F (1945) Individual comparisons by ranking methods. Biometrics Bulletin 1(6):80–83

Wu XZ, Liu S, Zhou ZH (2019) Heterogeneous model reuse via optimizing multiparty multiclass margin. In: International Conference on Machine Learning, pp 6840–6849

Wu YC, He YX, Qian C, et al (2022) Multi-objective evolutionary ensemble pruning guided by margin distribution. In: Proceedings of the 17th International Conference on Parallel Problem Solving from Nature, pp 427–441

Zhang C, Lu X, Cao P, et al (2023a) A nonprofiled side-channel analysis based on variational lower bound related to mutual information. Science China Information Sciences 66(1):112302

Zhang K, Xu L, Yi X, et al (2023b) Predefined-time distributed multiobjective optimization for network resource allocation. Science China Information Sciences 66(7):1–15

Zhang Y, Burer S, Nick Street W, et al (2006) Ensemble pruning via semi-definite programming. Journal of Machine Learning Research 7(7)

Zhang Y, Liu J, Shen W (2022) A review of ensemble learning algorithms used in remote sensing applications. Applied Sciences 12(17):8654

Zhou ZH (2012) Ensemble Methods: Foundations and Algorithms. Chapman & Hall/CRC, Boca Raton, FL

Zhou ZH (2016) Learnware: on the future of machine learning. Frontiers of Computer Science 10(4):589–590

Zhou ZH (2022) Open-environment machine learning. National Science Review 9(8):nwac123

Zhou ZH, Feng J (2019) Deep forest. National science review 6(1):74–86

Zhou ZH, Tan ZH (2023) Learnware: Small models do big. Science China Information Sciences https://doi.org/10.1007/s11432-023-3823-6

Zhou ZH, Tang W (2003) Selective ensemble of decision trees. In: Proceedings of the 9th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, pp 476–483

Zhou ZH, Wu J, Tang W (2002) Ensembling neural networks: Many could be better than all. Artificial Intelligence 137(1-2):239–263

Zhou ZH, Yu Y, Qian C (2019) Evolutionary Learning: Advances in Theories and Algorithms. Springer, Singapore