MEPSI: An MDL-based Ensemble Pruning Approach with Structural Information

Xiao-Dong Bi, Shao-Qun Zhang*, Yuan Jiang

National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
School of Artificial Intelligence, Nanjing University, Nanjing 210023, China
{bixd, zhangsq, jiangy}@lamda.nju.edu.cn

Abstract

Ensemble pruning that combines a subset of individual learners generated in parallel to make predictions is an important topic in ensemble learning. Past decades have developed a lot of pruning algorithms that focus on the external behavior of learners on samples, which may lead to over-fitting. In this paper, we conjecture that the generalization performance of an ensemble is not only related to its external behavior on samples but also dependent on the internal structure of individual learners. We propose the general MEPSI approach based on Kolmogorov complexity and the Minimum Description Length (MDL) principle, which formulates the ensemble pruning task as the two-objective optimization problem that comprises the empirical error and structural information among individual learners. We also provide a concrete implementation of MEPSI on decision trees. The theoretical results provide generalization bounds for both the general MEPSI approach and tree-based implementation. The comparative experiments conducted on multiple real-world data sets demonstrate the effectiveness of our proposed method.

1 Introduction

Ensemble learning is a powerful learning paradigm that trains and combines multiple learners to solve a single learning problem (Dietterich 2000; Zhou 2012), and has already achieved state-of-the-art results in real-world tasks, such as gradient boosting (Chen and Guestrin 2016; Dorogush, Ershov, and Gulin 2018; Friedman 2001), voting (Zhou 2012), and stacking (Breiman 1996; Wolpert 1992). Ensemble pruning is a key topic in ensemble learning, which selects a subset of individual learners generated in parallel to make predictions. It is observed that ensemble pruning not only reduces storage and computation costs but also achieves better performance than an ensemble of all individual learners. Conventional ensemble pruning methods can be roughly categorized into three classes: the order-based method, the clustering-based method, and the optimization-based method. The order-based method generates a priority for each learner according to a certain criterion and only selects learners with a high priority (Martínez-Muñoz and Suárez 2006). The clustering-based method employs clustering techniques to partition the individual learners into several groups and combines the representative prototype learners in each group to make predictions (Giacinto, Roli, and Fumera 2000; Lazarevic and Obradovic 2001). The optimization-based method converts ensemble pruning tasks into optimization problems, which can be solved by some mature approaches, such as the relaxation techniques and evolutionary algorithms (Wu et al. 2022; Zhou and Tang 2003; Zhou, Wu, and Tang 2002).

Previous studies often used empirical errors as the optimization objectives or the criteria for order, and all of them only paid attention to the external behavior of learners on the samples, which may lead to over-fitting. Intuitively, the learner with both a lower empirical error and a simpler structure tends to achieve better generalization performance. Thus, we conjecture that the generalization performance of an ensemble is not only related to its external behavior on samples but also dependent on the internal structure of individual learners. In ensemble pruning, one can quantify the structural information among individual learners to upper bound the complexity of the combined learner. Nevertheless, quantifying structural information is one of the three great challenges for half-century-old computer science, summarized in the 50th year of Journal of the ACM by Brooks Jr (2003). For decades, it still remains open about how to measure structural information. Thus, we bridge the learner's structure and generalization performance.

In this paper, we investigate the structural information among individual learners by exploiting the Kolmogorov complexity and the MDL principle. We propose the MDL-based Ensemble Pruning approach with Structural Information, equally MEPSI approach in short. MEPSI converts the ensemble pruning task to the two-objective optimization problem that minimizes the weighted sum of empirical errors and structural information of individual learners, which is measured by the algorithmic mutual information. Thus, MEPSI induces a general learning approach for ensemble pruning by building the bridge between the learner's structure and generalization. Besides, we provide a concrete implementation for MEPSI on decision trees, in which we implement the structural information term by the edit distance of decision trees.

Our main contributions are summarized as follows.

- We propose a general ensemble pruning approach MEPSI based on the theory of Kolmogorov complex-

*Shao-Qun Zhang is the corresponding author.

ity and MDL principle, which formulates the ensemble pruning task as an optimization problem that comprises empirical error and structural information.

- We provide an implementation for the general MEPSI on decision trees, where we propose the edit similarity measure to approximate the structural information term with a concrete algorithm procedure.
- We give the generalization bounds for both general MEPSI and tree-based implementation, which completely match our proposed optimization problems.
- We conduct experiments to compare MEPSI with 13 ensemble pruning methods on 11 classification data sets. The numerical results show that our method achieves good results and sufficiently outperforms other methods on average.

The rest of this paper is organized as follows. Section 2 introduces related notations and terminologies. Section 3 presents the MEPSI and its implementation algorithms. Section 4 theoretically investigates the generalization abilities of MEPSI. Section 5 conducts experiments on multiple multi-classification data sets. Section 6 concludes this work with discussions and prospects.

2 Preliminaries

This section will introduce useful notations, terminologies, and related studies.

2.1 Notations

This work considers the classification tasks. Let $\mathcal{X}$ be the feature space, and $\mathcal{Y}$ denotes the set of labels $\{1, 2, \ldots, C\}$. Let $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$ be the data set, where $(\boldsymbol{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}$.

Provided a set of individual learners $H = \{h_t\}_{t=1}^{T}$, ensemble pruning selects a subset $S \subseteq H$ whose size is k and combines the individual learners in S as the combined learner $H_S = \sum_{i=1}^{k} w_i h_{S_i}$, where S_i denotes the index of the i_{th} learner in S and w_i is the weight to combine the learners, satisfying $w_i > 0$ and $\sum_{i=1}^{k} w_i = 1$. Provided the scoring function $h_i : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, we can get a classifier $f_i : \mathcal{X} \to \mathcal{Y}$ as follows

$$f_i(\boldsymbol{x}) = \arg\max_{y \in \mathcal{Y}} h_i(\boldsymbol{x}, y) \ . \tag{1}$$

For convenience, we denote h_i as both the scoring function and classifier simultaneously. We also refer to a learner hereafter as a model or a hypothesis.

For a character set $\{0, 1\}$, we denote $\{0, 1\}^*$ as the set of finite strings over the character set $\{0, 1\}$, and we have $\{0, 1\}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, \ldots\}$, where ϵ means the empty string. We also use $l(x)$ to denote the length of the string x. Let $x = x_1 x_2 \ldots x_n$ be a binary string whose length $l(x)$ is n. We call $\bar{x}$ the self-delimiting (prefix) version of x, i.e.,

$$\bar{x} = \underbrace{1\,1 \cdots 1}_{n \text{ times}}\ 0\ x_1 x_2 \ldots x_n \ .$$

Given binary strings x and y, we define the standard invertible function $\langle \cdot, \cdot \rangle : \{0, 1\}^* \times \{0, 1\}^* \to \{0, 1\}^*$ with $\langle x, y \rangle = \bar{x}y$, which maps the pair of binary strings (x, y) to another one $\langle x, y \rangle$. The standard invertible function above can be extended to the multivariate version as follows

$$\langle z_1, z_2, \ldots, z_k \rangle = \langle z_1, \langle z_2, z_3, \ldots, z_k \rangle \rangle \ ,$$

where $(z_1, z_2, \ldots, z_k)$ is a k-tuple of binary strings. It is efficient to recover the k-tuple $(z_1, z_2, \ldots, z_k)$ unambiguously from the $\langle z_1, z_2, \ldots, z_k \rangle$ without any delimiter. We denote $\overset{+}{<}$ and $\overset{+}{>}$ as inequalities within an additive constant. More precisely, for two functions $u : D \to \mathbb{R}, v : D \to \mathbb{R}$, we say $u(x) \overset{+}{<} v(x)$ if and only if there exists a constant c such that $\forall x \in \mathrm{D}$ we have $u(x) < v(x) + c$. Let $u(x) \overset{\pm}{=} v(x)$ if both $u(x) \overset{+}{<} v(x)$ and $u(x) \overset{+}{>} v(x)$ hold.

2.2 Kolmogorov Complexity and Algorithmic Mutual Information

Kolmogorov complexity was originally proposed for measuring the randomness and complexity of individual objects (Solomonoff 1964; Kolmogorov 1965). In contrast to Shannon's information theory (Shannon 1948) which is based on the classical probability theory to measure the uncertainty and randomness of a known random source, Kolmogorov complexity is to measure the randomness of an individual object through computational theory. It is natural to say that an object is simple if it can be briefly described by a string or language. According to this motivation, Kolmogorov's theory provides a formal measurement of the object's complexity. Approximately, the Kolmogorov complexity of an object can be understood as the length of the shortest computer program written in the universal programming language that can print the object and then halt.

Formally, Li and Vitányi (2019) gives the definition of the prefix variant of Kolmogorov complexity; we also call it Kolmogorov complexity for convenience.

Definition 2.1. The Kolmogorov complexity $K : \{0, 1\}^* \to \mathbb{N}$ maps objects represented by binary strings to natural numbers. Let x, y, p be binary strings, and U be the reference prefix machine. The **conditional Kolmogorov complexity** of x conditional to y is defined by

$$K(x \mid y) = \min_p \{l(p) : U(\langle y, p \rangle) = x\} \ ,$$

where $U(\langle y, p \rangle)$ indicates the output of reference prefix machine whose input is $\langle y, p \rangle$. The **Kolmogorov complexity** of x is defined by

$$K(x) = K(x \mid \epsilon) = \min_p \{l(p) : U(p) = x\} \ ,$$

where ϵ denotes the empty string.

Algorithmic mutual information is derived from Kolmogorov complexity and formalizes the complexity-based similarity between two individual objects. Formally, one has the definition (Grünwald and Vitányi 2008; Li and Vitányi 2019) as follows

Definition 2.2. Let x, y be objects represented by binary strings. The **algorithmic information** about x contained in y is defined as

$$I(x : y) = K(y) - K(y \mid x^*) \, ,$$

where x^* represents the first shortest prefix program that prints x (Li and Vitányi 2019). It is observed that $I(x : y)$ is symmetric within an additive constant, i.e., $I(x : y) \stackrel{+}{=} I(y : x)$. Thus, we also call $I(x : y)$ the **algorithmic mutual information** between x and y.

Informally, the algorithmic mutual information between x and y can be understood as the description length of y that can be reduced with the help of x. It measures the structural similarity between x and y. The larger the algorithmic mutual information is, the more similarity objects x and y have.

2.3 The Minimum Description Length Principle

The MDL principle was first developed by Rissanen (1978), in which the best hypothesis that describes data x is the one that minimizes the two-part code length that is the sum of

- the length, in bits, of the description of the hypothesis,
- the length, in bits, of the description of the data x when the data is described with the help of the hypothesis.

The MDL principle provides a way of identifying hypotheses to avoid over-fitting, usually regarded as a concrete formulation of Occam's Razor. The ideal MDL principle relies on Kolmogorov complexity to measure the description length of the hypothesis and data (Li and Vitányi 2019).

Definition 2.3. Once both the hypothesis h and data set D can be expressed by strings, the ideal MDL principle selects the hypothesis h by minimizing

$$\underbrace{K(h)}_{\text{hypothesis description}} + \underbrace{K(D \mid h)}_{\text{data-to-model code}} \, , \tag{2}$$

where $K(h)$ measures the description length of the h and $K(D \mid h)$ measures the minimum description length of the D with the help of h.

The Kolmogorov-based MDL theory indicates that finding the MDL estimator in a hypothesis space of prescribed maximal Kolmogorov complexity gives the hypothesis that best fits the data (Vereshchagin and Vitányi 2004).

3 The MEPSI Approach

In this section, we propose the MEPSI, a general ensemble pruning approach, via the theory of Kolmogorov complexity and MDL principle. Section 3.1 introduces the general MDL-based optimization objective including empirical errors and structural information. Section 3.2 proposes an approximation method for the ensemble of tree-based models.

3.1 The General MDL-based Optimization Objective

We begin our work with the ensemble pruning task in which all individual learners are in a hypothesis class of prescribed maximal Kolmogorov complexity.

Assumption 3.1. Provided the set of individual learners $H = \{h_1, h_2, \ldots, h_T\}$, we assume that $\forall h \in H$, their Kolmogorov complexity $K(h)$ is upper bounded by a constant C_h, that is, $K(h) \leq C_h$ for $h \in H$.

This assumption is natural in pruning tasks for homogenous ensembles, such as random forests, in which all decision trees have similar node numbers and upper-bounded Kolmogorov complexities.

We follow the MDL principle in Definition 2.3 and represent the second term (data-to-model code) in Eq. (2) using the empirical error of the combined learner. Thus, the MDL principle can be regarded as minimizing the weighted sum of Kolmogorov complexity and the empirical error of combined learners. Hence, the ensemble pruning task can be re-formulated as the following two-objective optimization problem

$$\min_S \; \underbrace{\lambda K\left(H_S\right)}_{\text{complexity}} + \underbrace{\frac{1}{m}\sum_{i=1}^{m} \mathbb{I}\left(H_S(\boldsymbol{x_i}) \neq y_i\right)}_{\text{empirical error}}, \tag{3}$$

where $\lambda \geq 0$ is the trade-off parameter. Let $h_{S_{i:k}}$ denote the set $\{h_{S_i}, h_{S_2}, \ldots, h_{S_k}\}$, and $\langle h_{S_{i:k}} \rangle$ denotes the finite string $\langle h_{S_i}, h_{S_2}, \ldots, h_{S_k} \rangle$. Then the Kolmogorov complexity $K\left(H_S\right)$ of the combined learner H_S can be upper bounded by $K(h_{S_{i:k}})$, because H_S can be described with the help of the individual learners within a constant cost. Thus, the optimization objective in (3) could be upper bounded by

$$\lambda K(h_{S_{i:k}}) + \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}\left(H_S(\boldsymbol{x_i}) \neq y_i\right) \, . \tag{4}$$

We expand $K(h_{S_{i:k}})$ into the sum of the conditional Kolmogorov complexity of the individual learners $h_{S_{i:k}}$

$$K(h_{S_{1:k}}) \stackrel{+}{=} \sum_{i=1}^{k} K\left(h_{S_i} \mid \langle h_{S_{i+1:k}} \rangle^*\right) \, ,$$

which holds from the additivity property of Kolmogorov complexity (Li and Vitányi 2019). According to Definition 3.1, $K(h_{S_{1:k}})$ can be upper bounded by

$$K(h_{S_{1:k}}) \stackrel{+}{\leq} kC_h + \sum_{i=1}^{k} \left[K\left(h_{S_i} \mid \langle h_{S_{i+1:k}} \rangle^*\right) - K(h_{S_i})\right] \, .$$

According to Definition 2.2, we transform the conditional complexity into the algorithmic mutual information among different individual learners, then $K(h_{S_{1:k}})$ can be upper bounded by

$$K\left(h_{S_{1:k}}\right) \stackrel{+}{\leq} kC_h - \sum_{i=1}^{k} I\left(h_{S_i} : h_{S_{i+1:k}}\right) \, . \tag{5}$$

Substituting Inequality (5) into Eq. (4), we get the following optimization problem

$$\min_S \; \underbrace{-\lambda \sum_{i=1}^{k} I\left(h_{S_i} : h_{S_{i+1:k}}\right)}_{\text{structural information}} + \underbrace{\frac{1}{m}\sum_{i=1}^{m} \mathbb{I}\left(H_S(\boldsymbol{x_i}) \neq y_i\right)}_{\text{empirical error}} \, . \tag{6}$$

The surrogate objective above implies that minimizing the empirical error while maximizing the algorithmic mutual information among individual learners should be considered simultaneously. Recall Definition 2.2, the algorithmic mutual information $I(h_{S_i} : h_{S_{i+1:k}})$ not only means the description length of h_{S_i} that can be reduced with the help of the set $h_{S_{i+1:k}}$ but also measures the structural similarity between the individual learner h_{S_i} and the set of individual learners $h_{S_{i+1:k}}$. Thus, we call the first item of Eq. (6) the structural information term.

Notice that the Kolmogorov complexity is not computable, and thus, one cannot precisely calculate the optimization objective in Eq. (6), but can only approximate it by some surrogate objectives. Besides, it is hard to estimate $K(H_S)$ directly since H_S is the average of multiple individual learners. Inspired by this recognition, we conjecture that it is easier to design similarity metrics to approximate the algorithmic mutual information among different structural individual learners.

3.2 Implementation with Tree-based Model

This subsection shows a specific implementation for the general MDL-based objective and presents the concrete ensemble pruning algorithm based on decision trees.

The key challenge for implementation is to approximate the structural information term. The decision tree is a model in which each internal node represents a test on a feature, and each leaf node represents a class label. There are many successful ensemble models based on decision trees, such as random forest (Breiman 2001), extremely randomized trees (Geurts, Ernst, and Wehenkel 2006), and rotation forest (Rodriguez, Kuncheva, and Alonso 2006). Here, we choose the decision tree as the individual learners for implementation. Thanks to the tree-like structure and good comprehensibility of decision trees, the Kolmogorov complexity of a decision tree can be approximated by its node number (Li and Vitányi 2019). Hence, we can design some metrics to approximate the structural information term for ensemble pruning with decision trees.

We first introduce the edit distance between decision trees, which is proposed by Sun and Zhou (2018).

Definition 3.2. Let h, g be two decision trees, where we neglect the leaf nodes and reserve the internal nodes. The **tree edit distance** $\mathrm{TED}(h, g)$ is defined as the minimum number of node operations to transform h into g, correspondingly, the sequence of node operations is called **tree edit sequence** $\mathrm{TES}(h, g)$, where three node operations are considered

- [INSERT] inserting a node.
- [REMOVE] removing a node and connecting its children node to its parent node.
- [UPDATE] updating the associated feature of a node.

The tree edit distance $\mathrm{TED}(h, g)$ can be efficiently calculated within polynomial time complexity (Pawlik and Augsten 2011; Zhang and Shasha 1989). Now, we define the edit similarity measure ESM according to the TED measure between decision trees.

Algorithm 1: The Tree-based Implementation for MEPSI

Input: A data set $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$ whose number of samples is m, a candidate decision trees $H = \{h_i\}_{i=1}^T$ whose size is T, the pruning size k, and the hyperparameter coefficient λ

Output: A subset of decision trees S whose size is k

Initialize $S \leftarrow \emptyset$
Initialize minObj $\leftarrow \infty$
for tree $h_i \in \{h_1, h_2, \ldots, h_{T-1}\}$ **do**
  for tree $h_j \in \{h_{i+1}, h_{i+2}, \cdots, h_T\}$ **do**
    error $\leftarrow \frac{1}{m}\sum_{i=1}^m \mathbb{I}(\frac{h_i+h_j}{2}(\boldsymbol{x}_i) \neq y_i)$
    comp $\leftarrow \mathrm{NC}(h_i) + \mathrm{NC}(h_j) - \mathrm{TED}(h_i, h_j)$
    if error $-\ \lambda$ comp $<$ minObj **then**
      $S \leftarrow \{h_i, h_j\}$
      minObj $\leftarrow$ error $-\ \lambda$ comp
    end if
  end for
end for
while $|S| < k$ **do**
  Initialize $A \leftarrow \emptyset$
  Initialize minObj $\leftarrow \infty$
  for tree $h \in \{h_1, h_2, \cdots, h_T\} \setminus S$ **do**
    error $\leftarrow \frac{1}{m}\sum_{i=1}^m \mathbb{I}(\frac{h+\sum_{g\in S} g}{|S|+1}(\boldsymbol{x}_i) \neq y_i)$
    $\mathrm{ESM}(h, S) \leftarrow \mathrm{NC}(h) - \min_{g\in S}\{\mathrm{TED}(g, h)\}$
    if error $-\ \lambda\ \mathrm{ESM}(h, S) <$ minObj **then**
      $A \leftarrow \{h\}$
      minObj $\leftarrow$ error $-\ \lambda\ \mathrm{ESM}(h, S)$
    end if
  end for
  $S \leftarrow S \cup A$
end while

Definition 3.3. Let h be the concerned decision tree, and $G = \{g_1, g_2, \ldots, g_k\}$ indicates a set of decision trees. We define the **edit similarity measure** $\mathrm{ESM}(h, G)$ as

$$\mathrm{ESM}(h, G) = \mathrm{NC}(h) - \min_{g\in G}\{\mathrm{TED}(g, h)\} \ ,$$

where $\mathrm{NC}(h)$ denotes the number of internal nodes of h.

With slight abuse of symbol, we use G to denote both set $\{g_1, g_2, \ldots, g_k\}$ and finite string $\langle g_1, g_2, \ldots, g_k\rangle$. Now, we claim that $\mathrm{ESM}(h, G)$ is a good approximation of the algorithmic mutual information $I(h : G)$. Firstly, the Kolmogorov complexity $K(h)$ can be approximated well by $\mathrm{NC}(h)$ according to (Li and Vitányi 2019). Secondly, for all $g \in G$, the structure of h could be recovered according to the structure of g and the node operation sequence $\mathrm{TES}(g, h)$ when G is given. Thus, the minimum value for all $g \in G$ of $\mathrm{TED}(g, h)$ could be taken as a good approximation of $K(h \mid G^*)$. Thus, $I(h : G)$ could be approximated well by $\mathrm{ESM}(h, G)$ according to Definition 2.2 that indicates $I(h : G) = K(h) - K(h \mid G^*)$.

Here, we implement the algorithmic mutual information in Eq. (6) with ESM, so that the ensemble pruning task can

be converted into the following optimization problem

$$\min_{S} -\lambda \sum_{i=1}^{k} \mathrm{ESM}\left(h_{S_i} : h_{S_{i+1:k}}\right) + \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}\left(H_S(\boldsymbol{x_i}) \neq y_i\right) .$$

Inspired by Kappa pruning (Margineantu and Dietterich 1997), the above problem can be solved by a heuristic tree-based MEPSI algorithm. We first enumerate all the pairs of individual learners to find the pair with the minimum optimization objective and add them to the selected subset S. Then, we heuristically select the individual learner with the minimum optimization objective and add it to the selected subset until the size of S equals the pruning size k. Algorithm 1 lists the detailed MEPSI algorithm.

Notice that the proposed ESM in Definition 3.3 is one of the feasible metrics for approximating the algorithmic mutual information $I(h : G)$. Besides, the MEPSI approach also applies to other homogenous models or even heterogeneous models besides decision trees. So, it is interesting to explore alternative similarity metrics that utilize the structural information to approximate $I(h : G)$ in the future.

4 Theoretical Results

Now, we present our first generalization theorem for the general MEPSI approach handling the pruning task, in which all individual learners are in a hypothesis class of prescribed maximal Kolmogorov complexity C_h as follows

Theorem 4.1. *Let $\mathcal{H}$ be the countable hypothesis class of the individual learners. If $\forall\ h \in \mathcal{H}$, it holds $K(h) \leq C_h$, then for every sample size m, confidence parameter δ, and probability distribution $\mathcal{D}$, with probability greater than $1-\delta$ over the choice of $D \sim \mathcal{D}$, the following bound holds (simultaneously) $\forall S \subseteq \mathcal{H}$ whose cardinality is k*

$$L_{\mathcal{D}}(h_S) \leq L_D(h_S) + \sqrt{\frac{kC_h + \beta - \sum_{i=1}^{k} I(h_{S_i} : h_{S_{i+1:k}}) + \ln(2/\delta)}{2m}} ,$$

where h_S is an ensemble of the individual learners in S, β is a constant that is independent to the choice of S and sufficiently less than C_h.

Theorem 4.1 shows that empirical errors of the combined learner should be reduced while increasing the algorithmic mutual information among individual learners, which matches the two-objective optimization in the general MEPSI approach. The complete proof of Theorem 4.1 can be found in the appendix.

For the tree-based implementation shown in Section 3.2, we assume that the node number of decision trees is similar to their Kolmogorov complexities and further present the second theorem as follows

Theorem 4.2. *Let $\mathcal{H}$ be the countable hypothesis class of the decision trees. If $K(h) \leq C_h, \forall\ h \in \mathcal{H}$ and $|\mathrm{NC}(h) - K(h)| \leq \tau, \forall\ h \in \mathcal{H}$, then for every sample size m, confidence parameter δ, and probability distribution $\mathcal{D}$, with probability greater than $1-\delta$ over the choice of $D \sim \mathcal{D}$,the following bound holds (simultaneously) $\forall S \subseteq \mathcal{H}$ whose cardinality is k*

$$L_{\mathcal{D}}(h_S) \leq L_D(h_S) + \sqrt{\frac{k(C_h + \tau) + \gamma - \sum_{i=1}^{k} \mathrm{ESM}(h_{S_i}, h_{S_{i+1:k}}) + \ln(2/\delta)}{2m}} ,$$

where h_S is the ensemble of the individual learners in S, C_h and α are constants, γ is a constant that is independent to the choice of S and sufficiently less than C_h, $\mathrm{NC}(\cdot)$ is the node number of decision trees, and $\mathrm{ESM}(\cdot,\cdot)$ is the edit similarity measure in implementation.

Theorem 4.2 shows that empirical errors of the combined learner should be reduced while increasing ESM among individual learners. The theoretical guarantee for tree-based implementation is practical since we can easily estimate the upper bound C_h of $K(h)$ and τ. The complete proof of Theorem 4.2 can be found in the appendix.

We claim that the above bounds and our methods match. The generalization bound for the general MEPSI approach in Theorem 4.1 derives the following optimization problem

$$S^* = \arg\min_{S}\ L_D(h_S) + \lambda_1 \sqrt{\Delta - \sum_{i=1}^{k} \mathrm{I}(h_{S_i}, h_{S_{i+1:k}})} , \tag{7}$$

where $\lambda_1 \geq 0$, λ_1 and Δ are constants given by the generalization bound. This optimization problem can be written equivalently as

$$\begin{aligned} S^* &= \arg\min_{S}\ L_D(h_S) \\ \text{s.t. } & \sqrt{\Delta - \sum_{i=1}^{k} \mathrm{I}(h_{S_i}, h_{S_{i+1:k}})} \leq t , \end{aligned} \tag{8}$$

where t is a constant that corresponds one-to-one with λ_1. Actually, Eq. (7) is obtained by applying the method of Lagrange multipliers to Eq. (8), and λ_1 is a Lagrange variable. Eq. (8) also can be written equivalently as

$$\begin{aligned} S^* &= \arg\min_{S}\ L_D(h_S) \\ \text{s.t. } & -\sum_{i=1}^{k} \mathrm{I}(h_{S_i}, h_{S_{i+1:k}}) \leq t^2 - \Delta , \end{aligned} \tag{9}$$

By adding a Lagrange multiplier, Eq. (9) can be written equivalently as

$$S^* = \arg\min_{S}\ -\lambda_2 \sum_{i=1}^{k} \mathrm{I}(h_{S_i}, h_{S_{i+1:k}}) + L_D(h_S) , \tag{10}$$

where λ_2 is the Lagrange multiplier that corresponds one-to-one with t and λ_1. It is observed that Eq. (10) is identical to our designed optimization objective in Eq. (6). Thus, we can achieve the best generalization performance in Theorem 4.1 by adjusting λ in Eq. (6), which verifies the effectiveness of our results in Theorem 4.1. The above conclusion also applies to the tree-based implementation in Section 3.2.

Table 1: Comparison of test accuracy (mean ± std) for MEPSI and other methods on multiple data sets. Bold highlights the top three methods with the highest average accuracy for each data set. The last line counters the number of bold highlights of all methods on multiple data sets.

	Ours	Baseline		Order/Optimization/Clustering-based Pruning					Diversity-based Pruning					
Data Set	**MEPSI**	**All**	**Random**	**Kappa**	**Orient**	**Boost**	**SDP**	**HAC**	**TreeMatch**	**QStat**	**Disagree**	**Entropy**	**KWVar**	**InAgree**
Sklearn-Digits	**82.1±1.7**	78.2±1.1	73.8±1.9	75.7±3.1	67.6±4.3	80.1±2.0	74.7±3.1	79.4±2.6	78.3±1.8	75.3±3.2	**81.4±1.5**	78.6±2.0	**81.3±1.5**	79.0±2.2
USPS	**59.1±2.2**	45.6±0.7	45.4±2.1	49.7±4.4	47.4±6.1	58.6±2.3	46.8±2.8	58.8±2.2	56.1±2.6	57.6±2.6	**59.0±2.6**	57.5±2.5	**59.0±2.6**	58.9±2.4
Breast-Cancer	**79.1±3.2**	71.4±0.0	71.4±0.0	75.2±2.8	78.2±2.1	75.3±2.7	71.4±0.0	78.7±0.4	77.6±1.4	**79.5±0.4**	78.7±0.4	**78.9±1.0**	78.7±0.4	78.8±0.5
Breast-W	95.6±0.9	**96.0±0.3**	95.4±0.7	94.9±0.3	93.2±0.4	92.7±1.4	95.2±0.7	**95.7±0.5**	95.6±0.3	93.6±0.6	95.1±0.3	94.5±0.5	95.1±0.3	**95.8±0.5**
Vowel	**43.7±3.4**	42.9±2.2	36.4±4.0	29.5±3.0	23.5±3.8	42.9±2.6	38.4±2.9	40.7±3.6	**44.6±4.0**	26.9±4.6	**43.7±1.9**	42.1±2.1	**44.1±2.2**	39.6±3.9
Mfeat-Factors	**87.1±1.1**	84.6±1.1	78.8±3.6	79.1±3.9	73.3±1.6	**85.6±2.1**	78.2±4.0	80.5±3.0	84.4±1.5	76.3±1.9	**85.6±1.9**	84.3±1.9	85.5±2.0	80.4±3.5
Splice	68.9±4.7	50.2±0.0	53.3±5.6	58.5±0.0	50.2±0.0	60.7±8.8	50.6±1.8	**74.7±3.0**	**78.5±4.2**	71.0±3.3	70.8±3.7	63.4±4.3	70.8±3.7	**75.7±2.8**
Credit-A	**89.8±0.0**	86.7±3.8	81.4±9.1	83.2±2.1	50.0±0.0	61.8±8.8	77.4±8.9	87.4±2.4	87.1±3.3	**89.6±0.4**	86.7±3.2	**89.5±0.4**	86.8±3.2	89.4±0.6
Tic-Tac-Toe	78.3±7.8	64.9±0.0	64.9±0.0	**83.5±0.7**	64.9±0.0	**83.3±2.2**	65.0±0.4	80.1±5.6	**83.6±1.9**	64.9±0.0	75.2±7.2	75.7±7.9	75.2±7.2	82.0±3.4
Vehicle	**61.9±0.9**	60.3±2.8	54.3±7.9	47.3±3.4	50.7±9.3	61.9±0.6	56.3±4.6	59.5±4.8	61.1±1.0	39.9±7.6	**62.0±0.8**	61.7±0.5	**62.0±0.8**	61.2±1.7
Sick	**96.4±0.6**	91.8±0.0	91.8±0.0	91.8±0.1	**96.4±0.0**	**96.8±1.2**	91.8±0.0	96.2±0.6	95.1±0.8	92.5±1.4	91.8±0.0	91.8±0.0	91.8±0.0	96.4±0.5
Top3 Count	8/11	1/11	0/11	1/11	1/11	3/11	0/11	2/11	3/11	2/11	5/11	2/11	4/11	3/11

5 Experiments

This section conducts comparative experiments on several real-world image data sets and tabular data sets for multi-class ensemble pruning tasks to demonstrate the effectiveness of our proposed MEPSI.

5.1 Configurations

Data sets. The tree-based MEPSI is evaluated on 11 real-world image data sets and tabular data sets, including the scikit-learn digits (Pedregosa et al. 2011), USPS (Hull 1994), and multiple UCI data sets (Kelly, Longjohn, and Nottingham 2017), which are most frequently used in ensemble learning and ensemble pruning (Li and Zhou 2009; Rodriguez, Kuncheva, and Alonso 2006; Sun and Zhou 2018; Zhou and Tang 2003; Zhou and Feng 2019). Because the data sets are commonly used, we left out the details of the data. The default methods to split train and test data are employed if the data sets have been split into the training and testing parts. Otherwise, we randomly split the data for training and testing. The detailed split configurations of data sets are shown in the appendix.

Settings. For the generation of decision trees, we employ both bootstrap sampling and random feature selection approaches to train 200 CART decision trees (Breiman 2017) and select 20 of them to combine and make predictions. The trees are generated according to the implementation of scikit-learn's random forest (Pedregosa et al. 2011), and the detailed settings and hyperparameters of random forests are shown in the appendix. For the hyperparameter of Algorithm 1, we also show the setting of the trade-off weight λ in the appendix. The training and pruning processes are conducted according to the training data and are repeated 20 times randomly, and we report the mean and variance of the combined learners' accuracies on test sets.

Contenders. We employ 13 other pruning methods as contenders for our proposed MEPSI. Specifically, we compare the tree-based MEPSI approach with 2 baseline methods, including the ensemble of all learners and the ensemble of 20 randomly selected learners. We also compare our method with 3 order-based ensemble pruning approaches, including Kappa pruning (Margineantu and Dietterich 1997), orientation pruning (Martínez-Muñoz and Suárez 2006), and Boosting-based pruning (Martinez-Munoz and Suárez 2007), 1 optimization-based approach named SDP-Relaxation (Zhang et al. 2006), and 1 clustering-based approach named HAC Pruning (Giacinto, Roli, and Fumera 2000). We call them typical pruning methods in the experiments. Besides, we also compare our method with 6 other ensemble pruning approaches, which employ the same heuristic optimization method as Kappa pruning (Margineantu and Dietterich 1997) but use ensemble diversity measures as optimization objectives. Ensemble diversity indicates the difference among individual learners, and combining diverse learners often achieves better per-

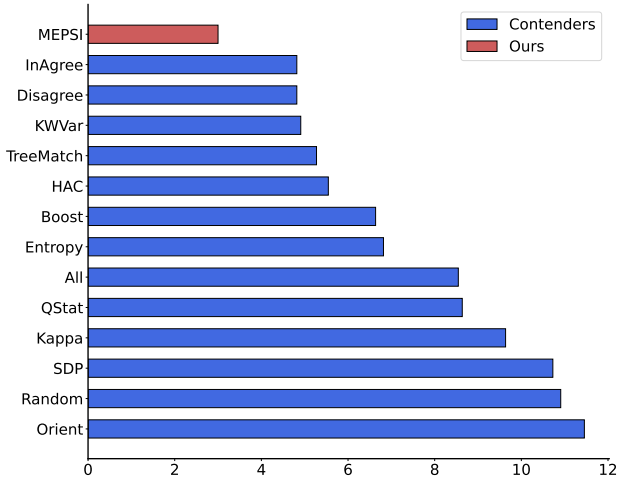

Figure 1: The average ranks of MEPSI and contenders on 11 data sets, where the horizontal axis and vertical axis indicate the average ranking and pruning method, respectively.

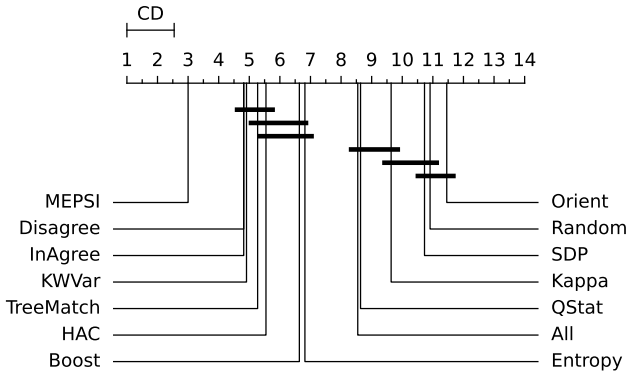

Figure 2: Comparison results of the Friedman test and Nimenyi test about the rank of MEPSI and contenders on 11 data sets, where the CD bar indicates the critical difference with the Nimenyi test at a significance level of 0.95.

formance (Banfield et al. 2005). Many works explicitly optimize diversity measures to select the subset of individual learners (Margineantu and Dietterich 1997; Martinez-Munoz, Hernández-Lobato, and Suárez 2008; Sun and Zhou 2018). To demonstrate the superiority of our proposed optimization objective that exploits the structural information among individual learners, we use the same heuristic optimization method as Kappa pruning to optimize pairwise diversity measures including Tree Match Diversity (Sun and Zhou 2018), Q-Statistic (Yule 1900), Disagreement Measure (Skalak 1996), and non-pairwise diversity measures including Entropy (Cunningham and Carney 2000), Kohavi-Wolpert Variance (Kohavi and Wolpert 1996), and Interrater agreement (Kuncheva and Whitaker 2003). We call them diversity-based pruning methods in the experiments.

5.2 Results

Table 1 shows the mean and variance of test accuracies of the proposed MEPSI and its contenders, which comprise 2 baseline methods, 5 typical pruning methods, and 6 diversity-based methods. Our method ranks top three among 14 methods on 8 data sets. As demonstrated, MEPSI performs well on most data sets. We also show the ranks of all methods on each data set in the Figure 1 and appendix, from which it is observed that our method ranks first on average.

In order to further compare the performance of different methods, we perform the Friedman test in conjunction with the Nimenyi test at a significance level of 0.95 to compare the performance of all methods according to their ranks on 20 repeated times pruning processes for multiple data sets. The test results are shown in Figure 2, which demonstrates that our method sufficiently outperforms other methods on average and verifies the effectiveness of MEPSI.

We also compute the Jaccard indexes between each pair of the learner sets returned by MEPSI and the other 13 methods. The Jaccard index is a measure of the similarity between two finite sets. The smaller the Jaccard indexes between the learner sets returned by MEPSI and the other methods, the more different the behavior of MEPSI and the other methods will be. The comparison results of the Jaccard index are shown in the appendix, from which we can find that the MEPSI approach selects quite different sets of decision trees for pruning from other methods.

In summary, our proposed MEPSI performs better than and is different from all other conducted pruning methods. The fact that our method performs better than diversity-based methods demonstrates that our proposed optimization objective is superior to other diversity measures for ensemble pruning when employing the heuristic optimization algorithm in Algorithm 1. Thus, it is necessary for MEPSI to exploit the structural information among individual learners when ensemble pruning.

6 Conclusions, Discussions, and Prospects

In this paper, we proposed an ensemble pruning approach, the MEPSI, by considering the internal structural information of individual learners. The proposed MEPSI converts the pruning task into a general optimization problem that comprises empirical error and structural information by exploiting the Kolmogorov complexity and MDL principle. We also provided an implementation of MEPSI on decision trees. The generalization bounds were given for both the general MEPSI approach and the concrete tree-based implementation, which leverage the effects of structural information on pruning approaches. Experiments conducted on several real-world image data sets and tabular data sets showed the superiority of our method.

Discussions. We are not the first to consider the generalization performance for ensemble pruning. Wu et al. (2022) employed margin distribution items into the optimization objective to measure the generalization performance of the combined learner. However, we are the first to consider generalizing ensemble pruning through the internal structure of individual learners. In addition, our method is practical for the pruning task of homogeneous ensembles but is hard to apply to heterogeneous ensembles due to the lack of structural information metrics. The measurement of structural information has been a well-known great challenge (Brooks Jr 2003), and it is extremely challenging to measure the structural information among heterogeneous models. However, our approach has taken a meaningful step toward measuring models' structural information.

Prospects. In Section 3.2, we designed an edit similarity measure to implement the structural information of decision trees. It is worth mentioning that the edit similarity measure is not the only feasible metric for approximating structural information; various implementations are worthy of exploring, especially when one considers the more detailed substructure of decision trees. Besides, this work only provides a concrete scheme for handling the ensemble of tree-based models; however, the MEPSI approach is also applicable to the ensemble of other models, such as neural networks (Zhang et al. 2023). It is attractive for further study to design and explore more metrics.

Acknowledgments

This research was supported by the Collaborative Innovation Center of Novel Software Technology and Industrialization, the Jiangsu Provincial Natural Science Foundation Youth Project (BK20230782), and the National Natural Science Foundation of China Project (Machine Learning for Heterogeneous Multi-View Data, No.62176117).

References

Banfield, R. E.; Hall, L. O.; Bowyer, K. W.; and Kegelmeyer, W. P. 2005. Ensemble diversity measures and their application to thinning. *Information Fusion*, 6(1): 49–62.

Breiman, L. 1996. Stacked regressions. *Machine Learning*, 24(1): 49–64.

Breiman, L. 2001. Random forests. *Machine Learning*, 45(1): 5–32.

Breiman, L. 2017. *Classification and Regression Trees*. Routledge.

Brooks Jr, F. P. 2003. Three great challenges for half-century-old computer science. *Journal of the ACM*, 50(1): 25–26.

Chen, T.; and Guestrin, C. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

Cunningham, P.; and Carney, J. 2000. Diversity versus quality in classification ensembles based on feature selection. In *Proceedings of the 11th European Conference on Machine Learning*, 109–116.

Dietterich, T. G. 2000. Ensemble methods in machine learning. In *Proceedings of the 1st International Workshop on Multiple Classifier Systems*, 1–15.

Dorogush, A. V.; Ershov, V.; and Gulin, A. 2018. CatBoost: Gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363*.

Friedman, J. H. 2001. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 1189–1232.

Geurts, P.; Ernst, D.; and Wehenkel, L. 2006. Extremely randomized trees. *Machine Learning*, 63(1): 3–42.

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

Grünwald, P. D.; and Vitányi, P. M. 2008. Algorithmic information theory. *Handbook of the Philosophy of Information*, 281–320.

Hoeffding, W. 1963. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301): 13–30.

Hull, J. J. 1994. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5): 550–554.

Kelly, M.; Longjohn, R.; and Nottingham, K. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml. Accessed: 2023-09-25.

Kohavi, R.; and Wolpert, D. H. 1996. Bias plus variance decomposition for zero-one loss functions. In *Proceedings of the 13th International Conference on Machine Learning*, 275–83.

Kolmogorov, A. N. 1965. Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1(1): 1–7.

Kuncheva, L. I.; and Whitaker, C. J. 2003. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2): 181–207.

Lazarevic, A.; and Obradovic, Z. 2001. Effective pruning of neural network classifier ensembles. In *Proceedings of the 2001 International Joint Conference on Neural Networks*, 796–801.

Li, M.; and Vitányi, P. M. B. 2019. *An Introduction to Kolmogorov Complexity and its Applications, 4th Edition*. Springer.

Li, N.; and Zhou, Z.-H. 2009. Selective ensemble under regularization framework. In *Proceedings of the 8th International Workshop on Multiple Classifier Systems*, 293–303.

Margineantu, D. D.; and Dietterich, T. G. 1997. Pruning adaptive boosting. In *Proceedings of the 14th International Conference on Machine Learning*, 211–218.

Martinez-Munoz, G.; Hernández-Lobato, D.; and 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.

Martínez-Muñoz, G.; and Suárez, A. 2006. Pruning in ordered bagging ensembles. In *Proceedings of the 23rd International Conference on Machine learning*, 609–616.

Martinez-Munoz, G.; and Suárez, A. 2007. Using boosting to prune bagging ensembles. *Pattern Recognition Letters*, 28(1): 156–165.

Pawlik, M.; and Augsten, N. 2011. RTED: A robust algorithm for the tree edit distance. *arXiv preprint arXiv:1201.0230*.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; and Dubourg, V. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830.

Rissanen, J. 1978. Modeling by shortest data description. *Automatica*, 14(5): 465–471.

Rodriguez, J. J.; Kuncheva, L. I.; and Alonso, C. J. 2006. Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10): 1619–1630.

Shannon, C. E. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3): 379–423.

Skalak, D. B. 1996. The sources of increased accuracy for two proposed boosting algorithms. In *Working Notes of the AAAI'96 Workshop on Integrating Multiple Learned Models*, 1133.

Solomonoff, R. J. 1964. A formal theory of inductive inference. Part I and Part II. *Information and Control*, 7: 1–22,224–254.

Sun, T.; and Zhou, Z.-H. 2018. Structural diversity for decision tree ensemble learning. *Frontiers of Computer Science*, 12(3): 560–570.

Vereshchagin, N. K.; and Vitányi, P. M. 2004. Kolmogorov's structure functions and model selection. *IEEE Transactions on Information Theory*, 50(12): 3265–3290.

Wolpert, D. H. 1992. Stacked generalization. *Neural Networks*, 5(2): 241–259.

Wu, Y.-C.; He, Y.-X.; Qian, C.; and Zhou, Z.-H. 2022. Multi-objective evolutionary ensemble pruning guided by margin distribution. In *Proceedings of the 17th International Conference on Parallel Problem Solving from Nature*, 427–441.

Yule, G. U. 1900. On the association of attributes in statistics. *Philosophical Transactions of the Royal Society of London*, 194(252-261): 257–319.

Zhang, K.; and Shasha, D. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal on Computing*, 18(6): 1245–1262.

Zhang, S.-Q.; Wu, J.-H.; Zhang, G.; Xiong, H.; Gu, B.; and Zhou, Z.-H. 2023. On the Generalization of Spiking Neural Networks via Minimum Description Length and Structural Stability. *arXiv preprint arXiv:2207.04876*.

Zhang, Y.; Burer, S.; Nick Street, W.; Bennett, K. P.; and Parrado-Hernández, E. 2006. Ensemble Pruning via Semi-definite Programming. *Journal of Machine Learning Research*, 7(48): 1315–1338.

Zhou, Z.-H. 2012. *Ensemble Methods: Foundations and Algorithms*. CRC press.

Zhou, Z.-H.; and Feng, J. 2019. Deep forest. *National Science Review*, 6(1): 74–86.

Zhou, Z.-H.; and Tang, W. 2003. Selective ensemble of decision trees. In *Proceedings of the 9th International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*, 476–483.

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

Appendix

In this Appendix, we provide the supplementary materials for our work "MEPSI: An MDL-based Ensemble Pruning Approach with Structural Information", constructed according to the corresponding sections therein.

A Full Proof for Theorem 4.1

Proof of Theorem 4.1. We first introduce Hoeffding's Inequality, which quantifies the gap between the empirical average of random variables and their expected value.

Lemma A.1. *Let $X_1, X_2, \cdots, X_m$ be a sequence of i.i.d. random variables and we assume for $\forall\, i \in \{1, 2, \ldots, m\}$ we have $X_i \in [a, b]$. Let $\bar{X}$ be the empirical average, i.e.*

$$\bar{X} = \frac{1}{m} \sum_{i=1}^{m} X_i \,,$$

then the following inequality holds for any ϵ:

$$\mathrm{P}\left[\bar{X} - \mathbb{E}\left[\bar{X}\right] \geq \epsilon\right] \leq \exp\left(\frac{-2m\epsilon^2}{(b-a)^2}\right) . \tag{11}$$

The proof can be found in (Hoeffding 1963).

According to Eq. (5), we have

$$K(h_{S_{1:k}}) \leq kC_h - \sum_{i=1}^{k} I(h_{S_i} : h_{S_{i+1:k}}) + \beta, \quad \forall S \subseteq \mathcal{H} \text{ and } |S| = k \,, \tag{12}$$

where β is a constant. According to the theory of Kolmogorov complexity, β is sufficiently less than C_h and does not depend on the choice of S (Li and Vitányi 2019).

Because the $K(\cdot)$ is the length of the prefix program, according to the Kraft inequality we have

$$\sum_{S \subseteq \mathcal{H} \text{ and } |S|=k} 2^{-K(h_{S_{1:k}})} \leq 1 \,. \tag{13}$$

Combining the Eq. (12) and Eq. (13), we have

$$\sum_{S \subseteq \mathcal{H} \text{ and } |S|=k} 2^{-\left(kC_h - \sum_{i=1}^{k} I(h_{S_i} : h_{S_{i+1:k}}) + \beta\right)} \leq 1 \,. \tag{14}$$

For a fixed subset of individual learners S, the corresponding combined learner is H_S. Because the 0-1 loss function $L_D(\cdot)$ can be represented with $\frac{1}{m}\sum_{i=1}^{m} \mathbb{I}(H_S(\boldsymbol{x}_i) \neq y_i)$ and the indicator function $\mathbb{I}(\cdot)$ is lower bounded by $a = 0$ and upper bounded by $b = 1$, we have the following result according to Theorem A.1:

$$\mathrm{P}\left[L_D(H_S) - L_{\mathcal{D}}(H_S) \geq \epsilon\right] \leq \exp\left(-2m\epsilon^2\right), \quad \forall S \subseteq \mathcal{H} \text{ and } |S| = k \,. \tag{15}$$

For $\forall \delta \in (0, 1)$, and let $\exp(-2m\epsilon^2) = 2^{-(kC_h - \sum_{i=1}^{k} I(h_{S_i} : h_{S_{i+1:k}}) + \beta)}\delta$, we have

$$\epsilon = \sqrt{\frac{kC_h - \sum_{i=1}^{k} I(h_{S_i} : h_{S_{i+1:k}}) + \ln(2/\delta)}{2m}} \,, \tag{16}$$

then substituting Eq. (16) into Eq. (15), for $\forall S \subseteq \mathcal{H}$ and $|S| = k$, we have

$$\begin{aligned} &\mathrm{P}\left[L_D(H_S) - L_{\mathcal{D}}(H_S) \geq \sqrt{\frac{kC_h - \sum_{i=1}^{k} I(h_{S_i} : h_{S_{i+1:k}} + \beta) + \ln(2/\delta)}{2m}}\right] \\ &\leq 2^{-(kC_h - \sum_{i=1}^{k} I(h_{S_i} : h_{S_{i+1:k}}) + \beta)}\delta \,. \end{aligned} \tag{17}$$

Applying the union bound and according to Eq. (14), we have

$$\begin{aligned} &\mathrm{P}\left[\exists S \subseteq \mathcal{H} \text{ and } |S| = k : L_D(H_S) - L_{\mathcal{D}}(H_S) \geq \sqrt{\frac{kC_h - \sum_{i=1}^{k} I(h_{S_i} : h_{S_{i+1:k}}) + \beta + \ln(2/\delta)}{2m}}\right] \\ &\leq \sum_{S \subseteq \mathcal{H} \text{ and } |S|=k} 2^{-(kC_h - \sum_{i=1}^{k} I(h_{S_i} : h_{S_{i+1:k}}) + \beta)}\delta \\ &\leq \delta \,, \end{aligned} \tag{18}$$

which implies

$$\mathrm{P}\left[\forall S \subseteq \mathcal{H}\ and\ |S| = k : L_{\mathcal{D}}(H_S) \leq L_D(H_S) + \sqrt{\frac{kC_h - \sum_{i=1}^{k} I(h_{S_i} : h_{S_{i+1:k}}) + \beta + \ln(2/\delta)}{2m}}\right] \geq 1 - \delta\ .$$

The theorem is proved. □

B Full Proof for Theorem 4.2

Proof of Theorem 4.2. The edit similarity measure ESM is defined in Definition 3.3. For the decision tree h_{S_i} and the set of decision tree $h_{S_{i+1:k}}$, the edit similarity measure between them is defined as

$$\mathrm{ESM}(h_{S_i}, h_{S_{i+1:k}}) = \mathrm{NC}(h_{S_i}) - \min_{j \in S_{i+1:k}} \left\{\mathrm{TED}(h_{S_j}, h_{S_i})\right\}\ , \tag{19}$$

where $NC(h)$ denotes the number of internal nodes of h, and TED is the tree edit distance defined in Definition 3.2.

We first introduce the following lemma which provides the lower bound of the second item in Eq. (19):

Lemma B.1. *The minimal tree edit distance* $\min_{j \in S_{i+1:k}} \left\{\mathrm{TED}(h_{S_j}, h_{S_i})\right\}$ *is lower bounded by the condition Kolmogorov complexity* $K(h_{S_i}|h_{S_j})$ *within an additivity constant, i.e.*

$$\min_{j \in S_{i+1:k}} \left\{\mathrm{TED}(h_{S_j}, h_{S_i})\right\} \geq K(h_{S_i}|\langle h_{S_{i+1:k}}\rangle^*) + \gamma_i\ ,$$

where γ_i *is a constant that does not depend on the choice of* S*.*

Proof of Theorem B.1. Because we can recover h_{S_i} according to the structure of h_{S_j} and the tree edit sequence $\mathrm{TES}(h_{S_j}, h_{S_i})$ defined in Definition 3.2, we have

$$\mathrm{TED}(h_{S_j}, h_{S_i}) \overset{+}{\geq} K(h_{S_i}|h_{S_j}) \overset{+}{\geq} K(h_{S_i}|h_{S_{i+1:k}}) \overset{+}{\geq} K(h_{S_i}|\langle h_{S_{i+1:k}}\rangle^*), \quad \forall j \in \{i+1, i+2, \ldots, k\}\ .$$

Thus, there is a sequence of constant γ_i that do not depend on the choice of S such that

$$\min_{j \in S_{i+1:k}} \left\{\mathrm{TED}(h_{S_j}, h_{S_i})\right\} \geq K(h_{S_i}|\langle h_{S_{i+1:k}}\rangle^*) + \gamma_i\ .$$

According to the theory of Kolmogorov complexity, γ_i is sufficiently less than C_h and does not depend on the choice of S (Li and Vitányi 2019). □

Because we assume that the node number of decision trees is similar to their Kolmogorov complexities, i.e.

$$|\mathrm{NC}(h) - K(h))| \leq \tau, \quad \forall h \in \mathcal{H}\ .$$

Thus, we have

$$\mathrm{NC}(h_{S_i}) \leq K(h_{S_i}) + \tau\ . \tag{20}$$

Combining Theorem B.1 and Eq. (20), we have the upper bound of edit similarity measure, i.e

$$\begin{aligned}\mathrm{ESM}(h_{S_i}, h_{S_{i+1:k}}) \quad &\leq K(h_{S_i}) - K(h_{S_i}|\langle h_{S_{i+1:k}}\rangle^*) + \tau - \gamma_i \\ &\leq I(h_{S_i} : h_{S_{i+1:k}}) + \tau - \gamma_i\ .\end{aligned} \tag{21}$$

Combining the inequality Eq. (21) and Eq. (12), we have

$$K(h_{S_{1:k}}) \leq k(C_h + \tau) - \sum_{i=1}^{k} \mathrm{ESM}(h_{S_i}, h_{S_{i+1:k}}) + \gamma, \quad \forall S \subseteq \mathcal{H} \text{ and } |S| = k\ , \tag{22}$$

where γ denotes $\beta - \sum_{i=1}^{k} \gamma_i$, which is a constant not depending on the choice of S and is sufficiently less than C_h.

Combining the (22) and Eq. (13), we have

$$\sum_{S \subseteq \mathcal{H} \text{ and } |S|=k} 2^{-\left(k(C_h+\tau) - \sum_{i=1}^{k} \mathrm{ESM}(h_{S_i}, h_{S_{i+1:k}}) + \gamma\right)} \leq 1\ . \tag{23}$$

For the immediate result Eq. (23), we apply the same techniques as Appendix A and follow the same approach as Eq. (15), Eq. (16), Eq. (17), and Eq. (18). Thus, for $\forall \delta \in (0, 1)$, we have

$$\mathrm{P}\left[\forall S \subseteq \mathcal{H}\ and\ |S| = k : L_{\mathcal{D}}(H_S) \leq L_D(H_S) + \sqrt{\frac{k(C_h + \tau) - \sum_{i=1}^{k} \mathrm{ESM}(h_{S_i}, h_{S_{i+1:k}}) + \gamma + \ln(2/\delta)}{2m}}\right] \geq 1 - \delta\ .$$

The theorem is proved. □

Claim B.1. *The theoretical bounds in both Theorem 4.1 and Theorem 4.2 are practical, i.e. it is easy to estimate the constants in the bounds, which include C_h, τ, and γ.*

When h is a decision tree, $K(h)$ can be upper bounded by $\mathrm{NC}(h)$ because the decision tree can be described by its nodes, τ can be upper bounded by $\max\{K(h), \mathrm{NC}(h)\}$ that is also upper bounded by $\mathrm{NC}(h)$. γ is a constant, which is sufficiently less than C_h and can be almost ignored in practice. Therefore, our theoretical guarantees are practical.

C Experiments

C.1 Experimental Settings

Table S1: Special settings for each data set. The 'Split' column means the way to split the data for the training part and testing part, where 'random (test_size=0.3)' represents the random split with 0.3 test proportion, and 'default' represents the default split given by the data set. The 'ccp_alpha' means the complexity parameter, which is a parameter in the scikit-learn decision tree (Pedregosa et al. 2011).

Data sets	Split	ccp_alpha	Data sets	Split	ccp_alpha
Sklearn-Digits	random (test_size=0.3)	0.2	USPS	default	0.2
Breast-Cancer	random (test_size=0.3)	0.1	Breast-W	random (test_size=0.3)	0.2
Vowel	default	0.2	Mfeat-Factors	random (test_size=0.3)	0.2
Splice	random (test_size=0.3)	0.2	Credit-A	random (test_size=0.3)	0.2
Tic-Tac-Toe	random (test_size=0.3)	0.05	Vehicle	random (test_size=0.3)	0.2
Sick	random (test_size=0.3)	0.05			

Table S2: General experimental settings of training and pruning for the random forest. The 'pruning_size' means the number of decision trees to select in the ensemble pruning task, and the other parameters can be found in the scikit-learn decision tree (Pedregosa et al. 2011).

Parameters	Value	Parameters	Value
pruning_size	20	n_estimators	200
criterion	entropy	max_features	sqrt
bootstrap	True	max_samples	1.0

The setting of trade-off weight λ in tree-based MEPSI implementation is related to the node's number of decision trees. We set λ as $\frac{C}{\mathrm{ESM}_{avg}}$, where ESM_{avg} denotes the average value of ESM term over all subsets of trees and can be estimated by sampling. Moreover, C is the weight that trades off between the empirical error term and structural information term in the objective after such scaling. The setting of C for different data sets is shown in Table S3. Here, We explain why λ is set as above. The two terms in the objective have different scales, i.e., the value of the empirical error term is in the interval $[0, 1]$ while the structural information term ESM may be at most as large as the number of tree nodes. Thus, we divide ESM by ESM_{avg} to scale the structural information term down to the same scale as the empirical error term.

Table S3: The setting of C about the trade-off weight λ for each data set.

Dataset	C	Dataset	C
Sklearn-Digits	0.8	USPS	0.8
Breast-Cancer	0.5	Breast-W	0.75
Vowel	0.75	Mfeat-Factors	0.75
Splice	0.75	Credit-A	0.75
Tic-Tac-Toe	0.5	Vehicle	0.75
Sick	0.5		

C.2 Additional Experimental Results

Table S4: Ranks of test accuracy for MEPSI and other methods on multiple data sets. The last line shows the average rank for each method on multiple data sets.

	Ours	**Baseline**		**Order/Optimization/Clustering-based Pruning**					**Diversity-based Pruning**					
Data Set	**MEPSI**	**All**	**Random**	**Kappa**	**Orient**	**Boost**	**SDP**	**HAC**	**TreeMatch**	**QStat**	**Disagree**	**Entropy**	**KWVar**	**InAgree**
Sklearn-Digits	1	9	13	10	14	4	12	5	8	11	2	7	3	6
USPS	1	13	14	10	11	6	12	5	9	7	2	8	2	4
Breast-Cancer	2	12	12	11	8	10	12	5	9	1	5	3	5	4
Breast-W	5	1	6	10	13	14	7	3	4	12	8	11	8	2
Vowel	3	5	11	12	14	6	10	8	1	13	3	7	2	9
Mfeat-Factors	1	5	11	10	14	3	12	8	6	13	2	7	4	9
Splice	7	13	11	10	13	9	12	3	1	4	5	8	5	2
Credit-A	1	8	11	10	14	13	12	5	6	2	8	3	7	4
Tic-Tac-Toe	6	11	11	2	11	3	10	5	1	11	8	7	8	4
Vehicle	3	8	11	13	12	4	10	9	7	14	1	5	1	6
Sick	3	9	9	8	2	1	9	5	6	7	9	9	9	3
Average Rank	3.00	8.55	10.91	9.64	11.45	6.64	10.73	5.55	5.27	8.64	4.82	6.82	4.91	4.82

Table S5: The Jaccard indexes between all methods except 'All', averaged over all datasets and 20 rounds.

	MEPSI	**Random**	**Kappa**	**Orient**	**Boost**	**SDP**	**HAC**	**TreeMatch**	**QStat**	**Disagree**	**Entropy**	**KWVar**	**InAgree**
MEPSI	1	0.0512	0.2695	0.0849	0.2632	0.1196	0.3608	0.2166	0.33	0.3713	0.362	0.3838	0.3723
Random		1	0.0560	0.0568	0.0571	0.0660	0.0552	0.0652	0.0609	0.0626	0.0616	0.0643	0.05713
Kappa			1	0.0599	0.1913	0.0943	0.3687	0.1688	0.2576	0.368	0.3147	0.3594	0.3574
Orient				1	0.0679	0.0488	0.1001	0.0571	0.1178	0.0874	0.0976	0.0868	0.0977
Boost					1	0.0799	0.2966	0.2471	0.2148	0.349	0.3425	0.3481	0.293
SDP						1	0.0939	0.0759	0.1091	0.1095	0.1157	0.1166	0.0975
HAC							1	0.2186	0.3818	0.541	0.4258	0.533	0.715
TreeMatch								1	0.1318	0.2423	0.2314	0.2423	0.229
QStat									1	0.3792	0.3877	0.393	0.3896
Disagree										1	0.6504	0.961	0.5186
Entropy											1	0.6816	0.4158
KWVar												1	0.512
InAgree													1