Machine Learning (2026) 115:242
https://doi.org/10.1007/s10994-026-07165-0

Grid-based Decision Tree Learning

Qin-Cheng Zheng[1,2] · Shao-Qun Zhang[1,3] · Shen-Huan Lyu[4,5] · Yuan Jiang[1,2] · Zhi-Hua Zhou[1,2]

Received: 22 May 2024 / Revised: 22 July 2026 / Accepted: 3 September 2026
© The Author(s), under exclusive licence to Springer Science+Business Media LLC, part of Springer Nature 2026

Abstract
Decision tree learning algorithms, such as Classification and Regression Trees (CART), are commonly founded on heuristics that greedily maximize purity gain. Despite their practical success, the theoretical properties of these algorithms, such as consistency, remain unclear. This paper uncovers a substantial obstacle in analyzing the consistency of decision tree learning algorithms, in which the worst-case purity gain can be zero and a gain value of zero usually results in a non-convergent training error. Building upon this observation, our previous work introduced GridCART, a grid-based decision tree that can yield positive purity gain and achieve consistency under Lipschitz smoothness and a product feature distribution. This article extends the grid-based construction to random forests and gradient boosting decision trees, and establishes consistency-rate upper bounds for Grid-based Random Forests (GridForest) and a square-loss specialization of the Grid-based Gradient Boosting Decision Tree (GridGBDT). Experiments on 20 OpenML datasets show that the grid-based ensembles perform comparably to their conventional counterparts.

Keywords Decision tree · Consistency · Statistical learning theory · Ensemble learning · Random forest · Gradient boosting decision tree

1 Introduction

Decision trees stand out as one of the most favored techniques in machine learning and data mining. The widely used CART (Breiman et al., 1984) represents a heuristic-driven algorithm rooted in tree-based learning. At each step, CART strategically selects a feature and a corresponding cut point to maximize the purity gain. Subsequently, it divides every leaf t into two children, that is $t \cap \{\mathrm{x} \mid \mathrm{x}^{(k)} < s\}$ and $t \cap \{\mathrm{x} \mid \mathrm{x}^{(k)} \geq s\}$. This recursive process persists until all leaf nodes exclusively contain samples with identical labels. Despite its reliance on heuristics, this top-down methodology has demonstrated efficacy across diverse task domains. Moreover, CART commonly serves as the base learner in ensemble learning frameworks like the popular random forest (Breiman, 2001), gradient boosting decision

Editors: Michelangelo Ceci and Siegfried Nijssen.

Extended author information available on the last page of the article

tree (Chen & Guestrin, 2016; Ke et al., 2017) and deep forest (Zhou & Feng, 2019). Hence, understanding the underlying factors contributing to its success is of great importance.

Consistency describes whether a learning algorithm can eventually learn the optimal classifier from a large amount of training data with high probability (Devroye et al., 1997), providing a solid theoretical guarantee for the algorithm's performance. Despite empirical progress over three decades ago (Breiman et al., 1984), the general consistency of heuristic tree learning algorithms, such as CART, remains unclear. Numerous efforts have been made to address this issue. Some studies (Devroye et al., 1997; Biau et al., 2008; Gao & Zhou, 2020) have demonstrated the consistency of purely random trees, whose splits are independent of the observed labels. Note that the label dependency in node splitting is essential for CART and may cause great challenges in proving consistency. Scornet et al. (2015) was the first to analyze the label-dependent node splitting and showed that CART is consistent under the assumptions of additive target functions and uniformly random feature distribution. Klusowski and Tian (2024) relaxed the assumption on feature distribution and established the consistency of CART in high-dimensional settings with additive target functions. Nevertheless, in real-world tasks, we typically lack side information about the target function, rendering the assumption of additive target function being too ideal.

This journal article extends our AISTATS 2023 paper (Zheng et al., 2023). The conference paper identified the zero-purity-gain obstacle and introduced GridCART, for which it established a consistency rate of order $\mathcal{O}(n^{-1/(d+2)})$ without assuming an additive target function. The present article retains this material to make the development self-contained and extends the grid-based construction to ensemble learning. The additional contributions of the journal version are summarized as follows:

1. We formulate GridForest and establish a consistency-rate upper bound by combining existing control of subsampling and random feature selection with our control of grid shifting. We also extend the fast rates under additional margin and strong density assumptions.
2. We formulate GridGBDT and establish a consistency-rate upper bound for the square-loss. By exploiting the common-grid residual structure and the geometric contraction of the boosting updates, we show that, under the stated parameter choices, $\mathcal{O}(\log n)$ boosting rounds suffice to attain the rate $\mathcal{O}(n^{-1/(d+2)})$, matching the GridCART rate under the same assumptions.
3. We evaluate GridForest and GridGBDT on 20 OpenML datasets. The grid-based ensembles obtain performance comparable to their conventional counterparts on this benchmark.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 introduces some essential terminologies and notations. Section 4 highlights the challenges associated with proving the convergence of CART. Section 5 presents the GridCART with its consistency rate and time complexity for binary classification. Section 6 extends GridCART to encompass ensemble learning methods such as random forest-based and gradient boosting-based approaches. Section 7 further expands GridCART to address more general tasks, including multiclass classification and regression. Section 8 empirically evaluates GridCART and its ensemble extensions. Section 9 discusses the limitations of the present

results and directions for future work. Section 10 summarizes our findings and concludes the paper.

2 Related Work

Decision-tree algorithms: Classical decision-tree learning dates back to ID3 (Quinlan, 1979, 1986), followed by C4.5 for continuous features (Quinlan, 1993) and CART for classification and regression (Breiman et al., 1984). Subsequent work has considered, among other directions, multivariate splits (Brodley & Utgoff, 1995), alternative splitting criteria (Mingers, 1989), incremental tree construction (Utgoff, 1989), and extremely randomized trees (Geurts et al., 2006). Tree-based ensembles, including random forests (Breiman, 2001), XGBoost (Chen & Guestrin, 2016), LightGBM (Ke et al., 2017), and deep forests (Zhou & Feng, 2019), further improve predictive performance by combining multiple trees. Our methods remain within this algorithmic lineage: GridCART is a greedy, top-down, label-dependent tree learner, while GridForest and GridGBDT are its grid-based random-forest and gradient-boosting extensions, respectively. The grid is therefore not a new ensemble principle; it is the device that makes the Influence-based splitting quantities estimable and enables the consistency analysis developed in this paper.

Consistency of decision trees: One line of theory studies trees whose structures are fully or partly randomized. Purely randomized trees have been shown to be consistent (Biau et al., 2008), and subsequent work derived consistency rates for randomized trees and simplified variants of Breiman's CART (Gao & Zhou, 2020; Gao et al., 2022). Because their tree growth is label-independent, these analyses do not encounter the same difficulty as greedy CART-style methods, whose splits depend on the observed responses. GridCART preserves such label-dependent, greedy construction; consequently, its analysis must directly address the zero-purity-gain obstacle identified in Section 4.

A second line analyzes CART under structural conditions on the target function. Scornet et al. (2015) established consistency under uniformly distributed features and an additive regression function with Gaussian noise, and Klusowski and Tian (2024) extended the analysis of CART with additive target functions to high-dimensional settings. Related results include a lower bound for CART when learning additive functions (Tan et al., 2022) and rates under a sufficient-impurity-decrease condition (Mazumder & Wang, 2023). In contrast, our target function need not be additive or satisfy a sufficient-impurity-decrease condition: apart from Lipschitz regularity, we impose no structural assumption on it. We instead modify the impurity criterion and estimate it on a grid, obtaining an $\mathcal{O}(n^{-1/(d+2)})$ consistency rate for a greedy, label-dependent tree and extending the analysis to its random-forest and gradient-boosting variants in Section 6.

Influence-based tree learning: The work most closely related to our splitting criterion uses Influence (Kahn et al., 1988; O'Donnell, 2014) as an impurity measure for variants of CART (Fiat & Pechyony, 2004; Blanc et al., 2020b). Those studies analyze the training error of trees for Boolean functions. We adopt the same high-level idea of measuring coordinate-wise Influence to prevent the purity gain from vanishing. However, we consider a different statistical setting that involves real-valued feature spaces and unknown data distributions. Our analysis also focuses on population consistency rather than convergence of the training error. InfCART formalizes the population-level Influence criterion, whereas GridCART

estimates the required quantities from data. This connection allows us to derive finite-sample consistency rates and extend the construction to ensemble learning.

3 Preliminaries

Table 1 lists most of the notations used in this paper.

Setting: Let $\mathcal{X} = [0,1]^d$ and $\mathcal{Y} = \{0,1\}$ be the feature space and label space, respectively. Let $\mathrm{p} : \mathcal{X} \to \mathbb{R}^+$ be an underlying probability density function on $\mathcal{X}$, and let $\eta(\mathrm{x}) = \Pr[Y = 1 \mid \mathrm{X} = \mathrm{x}]$ be the conditional probability function. We observe data $D_n = \{(\mathrm{X}_1, Y_1), (\mathrm{X}_2, Y_2), \ldots, (\mathrm{X}_n, Y_n)\}$ drawn i.i.d. from $\mathrm{X}_i \sim \mathrm{p}$ and $Y_{i|\mathrm{X}_i} \sim \mathrm{Bernoulli}(\eta(\mathrm{X}_i))$, where $\mathrm{X}_i = (\mathrm{X}_i^{(1)}, \mathrm{X}_i^{(2)}, \ldots, \mathrm{X}_i^{(d)})$ is a d-dimensional vector. Let $R(h) \triangleq \Pr[h(\mathrm{X}) \neq Y]$ denote the generalization error of hypothesis h. We write the Bayes error and the target function (or Bayes optimal classifier) as

$$R^\star = \min_{h:\mathcal{X}\to\mathcal{Y}} R(h) \quad \text{and} \quad f \in \arg\min_{h:\mathcal{X}\to\mathcal{Y}} R(h)\,,$$

Table 1 Summary of notations

Notation	Description
$R^\star, f$	Bayes error and target function, respectively
p_X	The underlying probability density of the feature distribution
$\hat{\mathrm{p}}$	The estimated probability density of the feature distribution
$\hat{\mathrm{p}}^{(j\neq k)}$	The estimated probability density of all coordinates except $\mathrm{X}^{(k)}$
X	Random vector in $\mathcal{X} = [0,1]^d$, representing the features of a sample drawn from $\mathcal{D}$
Y	Random variable in $\mathcal{Y} = \{0,1\}$, representing the label of a sample drawn from $\mathcal{D}$
D_n	Sample set containing n random pairs (X_i, Y_i)
$\mathrm{X}^{(k)}$	The k-th coordinate of random vector X
$\mathrm{X}^{(j\neq k)}$	Vector consisting of all coordinates of X except the k-th
x	Deterministic vector in $\mathcal{X} = [0,1]^d$
G	Impurity function $G : [0,1] \to [0,1]$
$\eta(\mathrm{x}), \hat{\eta}(\mathrm{x})$	Ground-truth and estimated conditional probability functions, respectively
$T(\mathrm{x})$	Tree function $T : [0,1]^d \to \{0,1\}$
$\mathrm{leaves}(T)$	Leaf set of tree T
t, t_L, t_R	Leaf of a tree, its left child, and its right child, respectively
$R(T)$	Error of tree T evaluated on distribution $\mathcal{D}$
h_n	Window size for histogram density estimation or cube size for the histogram classifier
g_n^{hist}	Histogram classifier learned from n i.i.d. samples
T_K	A depth-K tree generated by GridCART
δ_k	Shift in the k-th feature coordinate

respectively. Given the unknown nature of the underlying distribution p and the target function f, we employ a learning algorithm to derive estimators $\hat{\mathrm{p}}$ and $\hat{f}$. For simplicity, we write $\mathbb{E}_{\mathrm{X}\sim\mathrm{p}}$ and $\Pr_{\mathrm{X}\sim\mathrm{p}}$ as $\mathbb{E}_{\mathrm{p}}$ and $\Pr_{\mathrm{p}}$, respectively, and similarly use $\mathbb{E}_{\hat{\mathrm{p}}}$ and $\Pr_{\hat{\mathrm{p}}}$. We say that a distribution p is a product distribution if $\mathrm{p}_{\mathrm{X}} \equiv \prod_{k=1}^{d} \mathrm{p}_{\mathrm{X}^{(k)}}$, i.e., $\mathrm{X}^{(1)}, \mathrm{X}^{(2)}, \ldots, \mathrm{X}^{(d)}$ are mutually independent. The convergence rate we analyze in this paper is about the excess error, which is defined by $R(h) - R^{\star}$. If the generalization errors of a sequence of classifiers $\{h_n\}_{n=1}^{\infty}$ converge to the Bayes error or equivalently $R(h_n) - R^{\star} \to 0$, as $n \to \infty$, we say the sequence $\{h_n\}_{n=1}^{\infty}$ is consistent.

Tree: Denote a tree that consists of many leaf nodes by T and let $\mathrm{leaves}(T) \triangleq \{t \mid t \text{ is a leaf of } T\}$ be the leaves set. Every leaf, which is a subset of $\mathcal{X}$, is the intersection of sets like $\{\mathrm{x} \mid \mathrm{x}^{(k)} < s_k\}$ or $\{\mathrm{x} \mid \mathrm{x}^{(k)} \geq s_k\}$, where x is a d-dimensional vector in $\mathcal{X}$ and $\mathrm{x}^{(k)}$ is its k-th dimension. For all $\mathrm{x} \in t$, the prediction $T(\mathrm{x})$ is decided as follows:

$$T(\mathrm{x}) = \arg\max_{y\in\{0,1\}} \sum_{i=1}^{n} \mathbb{I}\{Y_i = y, \mathrm{X}_i \in t\},\ \mathrm{x} \in t\ .$$

An impurity function $G : [0,1] \to [0,1]$ is a strongly concave function that satisfies $G(0) = G(1) = 0$, $G(1/2) = 1$, and $G(x) = G(1-x)$ for any x. Common impurity functions used in greedy tree learning algorithms include entropy and the Gini index. The impurity function provides an efficient and effective way to build a tree greedily. At each step, the learning algorithm chooses a feature $\mathrm{x}^{(k)}$ and a cut point s splitting t into two children $t_L \triangleq t \cap \{\mathrm{x} \mid \mathrm{x}^{(k)} < s\}$ and $t_R \triangleq t \cap \{\mathrm{x} \mid \mathrm{x}^{(k)} \geq s\}$. The selection of k and s aims to maximize the purity gain, given by the expression:

$$p_t\Big[G(E_t) - w_{t_L} G(E_{t_L}) - w_{t_R} G(E_{t_R})\Big],$$

where p_t is defined as the probability $\Pr_{\hat{\mathrm{p}}}[\mathrm{X} \in t]$, and E_t is defined as the conditional expectation $\mathbb{E}_{\hat{\mathrm{p}}}\big[\hat{f}(\mathrm{X}) \mid \mathrm{X} \in t\big]$. The proportions of samples in the two child nodes are represented by $w_{t_L} = p_{t_L}/p_t$ and $w_{t_R} = p_{t_R}/p_t$.

Our studies work with the following assumptions.

Assumption 1 Assume that $\eta(\mathrm{x})$ is L-Lipschitz, i.e., $\exists L \geq 0$, s.t. $|\eta(\mathrm{x}_1) - \eta(\mathrm{x}_2)| \leq L\|\mathrm{x}_1 - \mathrm{x}_2\|, \forall \mathrm{x}_1, \mathrm{x}_2 \in \mathcal{X}$.

Assumption 2 The probability density p is smooth enough. Specifically, for all k, the derivative $\mathrm{d}\mathrm{p}_{\mathrm{X}^{(k)}}/\mathrm{d}\mathrm{x}^{(k)}$ is absolutely continuous and $\int_{\mathcal{X}^{(k)}} \left(\mathrm{d}\mathrm{p}_{\mathrm{X}^{(k)}}/\mathrm{d}\mathrm{x}^{(k)}\right)^2 \mathrm{d}\mathrm{x}^{(k)} < \infty$.

Assumption 3 The underlying feature distribution is a product distribution, i.e., $\mathrm{p}(\mathrm{x}) \equiv \prod_{k=1}^{d} \mathrm{p}_{\mathrm{X}^{(k)}}(\mathrm{x}^{(k)})$.

It is worth mentioning that Assumptions 1 and 2 are standard in consistency analysis (Audibert & Tsybakov, 2007). Assumption 3 may appear strong, but it is usually employed by seminal studies (Takimoto & Maruoka, 2003; Kalai & Teng, 2008; Brutzkus et al., 2020; Tan et al., 2022). Assumption 3 is also a relaxation of uniform distribution in Scornet et al. (2015). Notice that our results in this paper do not depend on any assumptions on the target function used in Scornet et al. (2015); Klusowski and Tian (2024); Blanc et al. (2020a).

4 Difference Between CART and InfCART

This section introduces InfCART as an idealized intermediate algorithm between CART and the implementable GridCART. CART computes impurity after averaging over all coordinates in a leaf; InfCART instead measures coordinate-wise Influence and thereby avoids zero purity gain unless the error is already zero. Because the required expectations depend on the unknown distribution and target function, InfCART is not directly implementable. GridCART, introduced in Section 5, estimates these quantities on a finite grid. Thus, InfCART supplies the theoretical splitting principle, whereas GridCART supplies its data-driven realization. Section 4.1 shows how to prove a vanishing error, which is one of the main ingredients for consistency analysis. Section 4.2 shows that the zero purity gain of CART in the worst case leads to a serious obstacle proving a vanishing error. Section 4.3 introduces Influence-based Classification And Regression Trees (InfCART), a variant of CART that uses Influence as the impurity measure, which can always get purity gain even under no assumptions on the target function.

4.1 Roadmap for Bounding the Error

For any $K \in N^+$, CART builds a tree T_K with depth K. We define the potential C_K of T_K as follows:

$$C_K \triangleq \sum_{t \in \text{leaves}(T_K)} p_t \, \mathcal{G}(t) \quad \text{with} \quad \mathcal{G}(t) \triangleq G\Big(\mathbb{E}_{\hat{\mathrm{p}}}\big[\hat{f}(\mathrm{X}) \mid \mathrm{X} \in t\big]\Big) ,$$

where p_t follows the definition in Section 3 and $\mathcal{G}(t)$ is the impurity of leaf t. Here, we use $\hat{\mathrm{p}}$ and $\hat{f}$ rather than p and f because the true distribution and model are unknown. Thus, C_K measures the average impurity of the tree T_K. We then show that the potential C_K of T_K is an upper bound of the error $\hat{R}(T_K)$ in Proposition 1.

Proposition 1 *For any tree T_K with depth K and any impurity function G, we have the training error $\hat{R}(T_K) \leq C_K$.*

A crucial aspect of establishing consistency lies in demonstrating the convergence of the error to zero, as a consistent algorithm must strike a balance between the empirical error and model complexity. Thus, it suffices to ensure a vanishing potential C_K from Proposition 1, which will be detailed in Section 4.2.

4.2 Failure of Conventional Impurity Measures

Let Δ_K be the purity gain for a tree T_K growing from depth K to $K+1$ as follows:

$$\Delta_K \triangleq C_K - C_{K+1} .$$

Then we have the following conclusion.

Proposition 2 *There exists a probability density* p *and a conditional probability function* $\eta : \mathcal{X} \to [0,1]$*, such that* $\Delta_K = 0$ *for any possible cut in every leaf.*

Proposition 2 demonstrates that, in the worst case the potential gain Δ_K can be zero. This observation poses a significant challenge in establishing a vanishing error as indicated by Proposition 1. The proof of Proposition 2 is by giving a counterexample in which the maximal potential gain is zero but the error is not. Additional elaboration on this matter can be found in Appendix A.

4.3 Influence as the Impurity Measure

From the above discussions, the potential gain using conventional impurity measure, which is used in CART, tends to be zero in the worst case. In this section, we will present a new impurity measure called Influence, which never fails to achieve a nonzero purity gain. To begin with, we introduce the definition of Influence as follows:

Definition 1 *(G-Influence)* Let p and f be a product distribution and a mapping from $\mathcal{X}$ to $\mathcal{Y} = \{0,1\}$, respectively. The G-Influence of f on $\mathrm{X}^{(k)}$ is defined by

$$\mathrm{Inf}_k^G[f] \triangleq \mathbb{E}_{\mathrm{p}^{(j\neq k)}}\Big[G\Big(\mathbb{E}_{\mathrm{p}^{(k)}}[f(\mathrm{X})]\Big)\Big],$$

where

$$\mathbb{E}_{\mathrm{p}^{(j\neq k)}}[\cdot] \triangleq \mathbb{E}_{\mathrm{X}^{(1)}\sim\mathrm{p}^{(1)}} \cdots \mathbb{E}_{\mathrm{X}^{(k-1)}\sim\mathrm{p}^{(k-1)}} \mathbb{E}_{\mathrm{X}^{(k+1)}\sim\mathrm{p}^{(k+1)}} \cdots \mathbb{E}_{\mathrm{X}^{(d)}\sim\mathrm{p}^{(d)}}[\cdot],$$

or equivalently taking expectation over all the coordinates except $\mathrm{X}^{(k)}$.
It is natural to define the conditional Influence as follows:

Definition 2 *(Conditional G-Influence)* With the same notations as per Definition 1, the conditional Influence of function f at leaf t is defined as:

$$\mathrm{Inf}_k^G[f \mid t] \triangleq \mathbb{E}_{\mathrm{p}^{(j\neq k)}}\Big[G\Big(\mathbb{E}_{\mathrm{p}^{(k)}}[f(\mathrm{X}) \mid \mathrm{X} \in t]\Big)\Big].$$

With Definition 2 at hand, we can write the purity gain induced by Influence as follows:

$$\Delta^{\mathrm{Inf}}(t,k,s) = p_t\Big(\mathcal{G}^{\mathrm{Inf}}(t) - w_{t_L}\mathcal{G}^{\mathrm{Inf}}(t_L) - w_{t_R}\mathcal{G}^{\mathrm{Inf}}(t_R)\Big),$$

where we write $\mathcal{G}^{\mathrm{Inf}}(t) \triangleq \mathrm{Inf}_k^G[f \mid t]$ for simplicity. Then, InfCART splits every leaf t by maximizing the purity gain as follows:

$$\max_{k,s} \Delta^{\mathrm{Inf}}(t,k,s).$$

Table 2 Comparison of CART and InfCART

Method	Impurity Measure
CART	$\mathcal{G}(t) = \frac{1}{d}\sum_{k=1}^{d}\left[G\left(\mathbb{E}_{\mathrm{p}(j\neq k)}\mathbb{E}_{\mathrm{p}(k)}\left[f(\mathrm{X}) \mid \mathrm{X}\in t\right]\right)\right]$
InfCART	$\mathcal{G}(t) = \frac{1}{d}\sum_{k=1}^{d}\mathbb{E}_{\mathrm{p}(j\neq k)}\left[G\left(\mathbb{E}_{\mathrm{p}(k)}\left[f(\mathrm{X}) \mid \mathrm{X}\in t\right]\right)\right]$

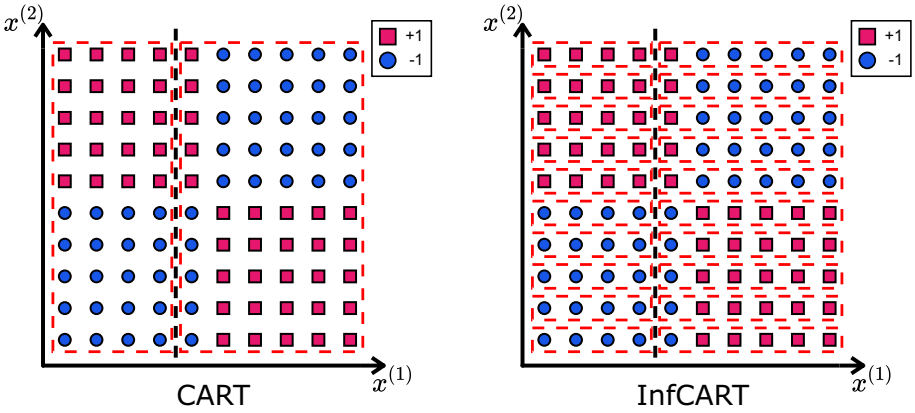


Fig. 1 Comparison of the calculation mechanism of purity gain between CART and InfCART for solving the "XOR" problem. In contrast to the CART which calculates impurities by taking an average over all the samples in the leaves and thus gets a zero purity gain, InfCART decomposes the leaves into lines and calculates the average impurities over the lines, which yields a positive purity gain

We compare CART with InfCART in Table 2, where we write the impurity measure of CART redundantly for comparison. Different from CART, InfCART takes an expectation taken later than the impurity function G. We will then show that the slight difference makes great help for lower bounding the purity gain nontrivially.

Proposition 3 *For any product distribution* p *on* $\mathcal{X}$ *and any target function* $\hat{f} : \mathcal{X} \to \mathcal{Y}$*, InfCART gets no purity gain if and only if it reaches zero error.*

Proposition 3 shows an advantage of InfCART compared with CART; the former can ensure non-trivial purity gain without any assumptions on the target functions (Scornet et al., 2015; Klusowski & Tian, 2024), while the latter cannot.

As shown in Figure 1, CART calculates the impurity by taking the average of labels in each child into impurity function G, leading to impurities equal to 1 in both the parent and the two children, and thus gets no purity gain for any cut in this example. Nevertheless, InfCART treats the leaf as many horizontal lines and calculates the purity gain by taking an average over the purity gain at every line, which always yields a nonzero purity gain.

InfCART is defined through population expectations and therefore requires an Influence oracle. GridCART addresses this implementation gap by estimating the required distributional quantities from data.

5 GridCART: A Refined CART with Consistency Guarantee

This section presents GridCART, whose key idea is to estimate the underlying distribution so that Influence can be obtained from the estimated distribution. Therefore, GridCART is a practical and provably consistent greedy algorithm. It consists of three steps: estimating the feature distribution, estimating the target function, and tree building, which are detailed in Algorithms 1, 2, 3 and 4.

5.1 Estimating the Feature Distribution

In this step, we will estimate a univariate distribution for each dimension, and the estimated density of the joint distribution is their multiplication. The key reason is that the curse of dimensionality in nonparametric density estimation (Devroye et al., 1997) may be avoided under Assumption 3, as shown in Lemma 4. The density of each coordinate can be estimated using well-known histogram methods (Devroye et al., 1997; Wasserman, 2006), as detailed in Algorithm 1.

Input: Training dataset D_n, grid size h
Output: An estimated probability density function $\hat{\mathbf{p}}_{\mathbf{X}}$
Procedure:

for $k \in [d]$ **do**
  Split the interval $[0,1]$ into bins: $\mathcal{B} \leftarrow \{[0,h), [h,2h), \ldots, [(1/h-1)h, 1]\}$
  for b in $\mathcal{B}$ **do**
    Estimate the density by the ratios: $\omega(b) \leftarrow 1/(nh) \sum_{i=1}^{n} \mathbb{I}\{\mathbf{X}_i^{(k)} \in b\}$
  end for
  $\hat{\mathbf{p}}_{\mathbf{X}^{(k)}}(x) \leftarrow \omega(b)$, for $x \in b, b \in \mathcal{B}$
end for
Compute the joint density by multiplication: $\hat{\mathbf{p}}_{\mathbf{X}} \leftarrow \prod_{k=1}^{d} \hat{\mathbf{p}}_{\mathbf{X}^{(k)}}$
return $\hat{\mathbf{p}}_{\mathbf{X}}$

Algorithm 1 Histogram Density Estimation (HDE)

Lemma 4 *Suppose that* $\mathbb{L}_1$*-error of estimators* $\hat{\mathrm{p}}_{\mathrm{X}^{(k)}}$ *has upper bound* $\mathbb{E}\big[\|\hat{\mathrm{p}}_{\mathrm{X}^{(k)}} - \mathrm{p}_{\mathrm{X}^{(k)}}\|_1\big] \le e_k$ *and Assumption 3 holds, then the following holds:*

$$\mathbb{E}\left[\left\|\prod_{k=1}^{d} \hat{\mathrm{p}}_{\mathrm{X}^{(k)}} - \mathrm{p}_{\mathrm{X}}\right\|_1\right] \le \sum_{k=1}^{d} e_k \,.$$

Generally, it is hard to estimate the distribution in high dimensions, due to the curse of dimensionality (Devroye et al., 1997). However, Lemma 4 simplifies the problem by reducing the d-dimensional case to multiple 1-dimensional cases, thus facilitating the estimation process. The total loss can be bounded by the sum of errors in each dimension,

which is linear in d in the worst case but does not depend exponentially on d in general situations (Tsybakov, 2009). By combining this result with the error bound of the histogram density estimation, we derive Lemma 5.

Lemma 5 *Under Assumptions 2 and 3, the $\mathbb{L}_1$-error of histogram density estimator can be upper bounded by*

$$\mathbb{E}\left[\left\|\prod_{k=1}^{d}\hat{\mathrm{p}}_{\mathrm{X}^{(k)}}-\mathrm{p}_{\mathrm{X}}\right\|_1\right]\leq\mathcal{O}\left(d\left(h_n+\sqrt{\frac{1}{nh_n}}\right)\right).$$

Lemma 5 provides an error bound for the density estimator in Algorithm 1. Based on this lemma, the expected $\mathbb{L}_1$-error has order $\mathcal{O}(n^{-1/3})$, by selecting the grid size $h_n=\Theta(n^{-1/3})$. The error bound does not depend on d exponentially, implying that the curse of dimensionality is avoided under Assumption 3.

5.2 Estimating the Target Function

We use the well-known histogram classifier (Devroye et al., 1997; Wasserman, 2006) to estimate the target function, which is detailed in Algorithm 2.

Input: Training dataset D_n, grid size h
Output: A histogram classifier g_n^{hist}
Procedure:

Compute the set of all possible Cartesian products of intervals according to:

$$\mathfrak{C}\leftarrow\left\{\times_{k=1}^{d}\left[m_kh,(m_k+1)h\right)\mid\boldsymbol{m}\in\{0,\ldots,1/h-1\}^d\right\}$$

for $\mathcal{C}$ in $\mathfrak{C}$ **do**
  Compute the number of samples in $\mathcal{C}$.

$$N(\mathcal{C})\leftarrow\sum_{i=1}^{n}\mathbb{I}\{\mathbf{X}_i\in\mathcal{C}\}$$

  if $N(\mathcal{C})>0$ **then**
    Compute the conditional probability estimator by averaging the labels in $\mathcal{C}$.

$$\hat{\eta}(\mathcal{C})\leftarrow N(\mathcal{C})^{-1}\sum_{i=1}^{n}\mathbb{I}\{\mathbf{X}_i\in\mathcal{C}\}Y_i$$

  else
    Set $\hat{\eta}(\mathcal{C})\leftarrow 1/2$ when $\mathcal{C}$ contains no training samples
  end if
end for
Predict the label by voting: $g_n^{\mathrm{hist}}(\mathbf{x})=\mathbb{I}\{\hat{\eta}(\mathcal{C})>1/2\}$ for $\mathbf{x}\in\mathcal{C},\mathcal{C}\in\mathfrak{C}$
return g_n^{hist}

Algorithm 2 Learning Histogram Classifier (LHC)

We split the feature space $\mathcal{X} = [0,1]^d$ into many length-h_n disjoint cubes $[0,1]^d = \bigcup_{j\in\mathcal{J}} \mathcal{C}_j$. Every cube $\mathcal{C}_j$ has form $\times_{k=1}^d [m_k h_n, (m_k+1)h_n)$, where $m_k \in \mathbb{N}$, and we have the number of cubes $|\mathcal{J}| \propto \frac{1}{h_n^d}$ by simple calculations. The endpoint $1_{1\times d}$ is neglected for simplicity. Let $\mathcal{C}(\mathrm{x})$ be the cube that contains x for any $\mathrm{x} \in \mathcal{X}$, then we can define

$$\hat{\eta}_n(\mathrm{x}) \triangleq \frac{\sum_{i=1}^n Y_i \mathbb{I}\{\mathbf{X}_i \in \mathcal{C}(\mathrm{x})\}}{\sum_{i=1}^n \mathbb{I}\{\mathbf{X}_i \in \mathcal{C}(\mathrm{x})\}}$$

as the estimated conditional probability. If no samples fall into the cube $\mathcal{C}(\mathrm{x})$, we set $\hat{\eta}_n(\mathrm{x}) = 1/2$ and use the fixed tie-breaking rule in Algorithm 2. The histogram classifier g_n^{hist} can eventually be defined by

$$g_n^{\text{hist}}(\mathrm{x}) = \mathbb{I}\{\hat{\eta}_n(\mathrm{x}) > 1/2\},$$

or equivalently predict the label by voting in every cube. Based on the definitions, the following conclusion can be drawn.

Lemma 6 *Under Assumption 1, suppose that the histogram rule satisfies* $h_n \to 0$ *and* $nh_n^d \to \infty$ *as* $n \to \infty$. *For any* p, η, $n > 0$ *we have*

$$\mathbb{E}\left[R(g_n^{\text{hist}})\right] - R^\star \le \mathcal{O}\left(h_n + \sqrt{\frac{1}{nh_n^d}}\right).$$

Exclusively provided that $h_n = \Theta(n^{-1/(d+2)})$, *we obtain a consistency rate of order* $\mathcal{O}(n^{-1/(d+2)})$ *for histogram classifier* g_n^{hist}.

Lemma 6 demonstrates that the expected excess error of the histogram classifier can approach zero as n and h_n tend to infinity and zero, respectively. Intuitively, as the sample size n tends to infinity and h_n tends to zero, the number of cubes $|\mathcal{J}|$ increases infinitely, enabling a more detailed capture of the feature space. Additionally, the number of samples falling into each cube $|\{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n\} \cap \mathcal{C}|$ also tends to infinity in probability, resulting in increasingly accurate voting.

5.3 Constructing a Tree

With estimator $\hat{\mathrm{p}}$ and g_n^{hist} at hand, we can build a tree by running Algorithm 3, which is a recursive procedure that builds a tree of depth $K_{\max}$. To show the convergence of the error, we begin with the following definition:

Input: Training dataset D_n, a histogram density estimator $\hat{\mathbf{p}}$, a histogram classifier g_n^{hist}, the maximal tree depth $K_{\max}$, and grid size h
Output: A learned tree $T : \mathbb{R}^d \to \{0, 1\}$
Procedure:

Pass $\mathbf{p} \leftarrow \hat{\mathbf{p}}$ and $f \leftarrow g_n^{\text{hist}}$ to the definition of conditional Influence in Definition 2
Initialize the leaves set by the whole feature space $\mathcal{L} \leftarrow \{[0, 1]^d\}$
for $\ell = 1, \ldots, K_{\max}$ **do**
  Copy a leaves set $\mathcal{L}' \leftarrow \mathcal{L}$
  for $t \in \mathcal{L}'$ **do**
    if All the samples in t have the same label **then**
      continue
    else
      Solve the best split: $(k_0, s_0) \leftarrow \underset{k \in [d],\, s \in \{h, 2h, \ldots, 1-h\}}{\arg\max} \Delta^{\text{Inf}}(t, k, s)$
      Split the leaf t into two children:

$$t_L \leftarrow t \cap \{\mathbf{x} \mid \mathbf{x}^{(k_0)} < s_0\} \quad t_R \leftarrow t \cap \{\mathbf{x} \mid \mathbf{x}^{(k_0)} \geq s_0\}$$

      Update the leaves set $\mathcal{L} \leftarrow (\mathcal{L} \setminus \{t\}) \cup \{t_L, t_R\}$
    end if
  end for
end for
Predict the label by voting:

$$T(\mathbf{x}) \leftarrow \underset{y \in \{0,1\}}{\arg\max} \sum_{i=1}^{n} \mathbb{I}\{\mathbf{X}_i \in t, Y_i = y\} \text{ for } \mathbf{x} \in t$$

return T

Algorithm 3 Tree Building (TB)

Definition 3 *(Grid-aligned N-piece function)* Fix coordinate-wise cut-point sets $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_d)$, where $\mathcal{S}_k \subset (0, 1)$ and $|\mathcal{S}_k| \leq N - 1$. A function f is called a grid-aligned N-piece function with respect to $\mathcal{S}$ if, for every $k \in [d]$ and every fixed value of the other coordinates, $f_k(z)$ is piecewise constant with at most N pieces and all its discontinuities belong to $\mathcal{S}_k$, where $f_k(z) \triangleq f(\mathrm{x}^{(1)}, \ldots, \mathrm{x}^{(k-1)}, z, \mathrm{x}^{(k+1)}, \ldots, \mathrm{x}^{(d)})$.

Definition 3 serves as an extension of the piecewise-constant function employed in univariate scenarios. Figure 2 demonstrates an example of a 5-piece function, where the univariate function becomes a piecewise constant with 5 distinct parts when either $\mathrm{x}^{(1)}$ or $\mathrm{x}^{(2)}$ is given. For the regular grid $\mathcal{S}_k = \{h_n, 2h_n, \ldots, 1 - h_n\}$, the histogram classifier learned by Algorithm 2 is a grid-aligned $1/h_n$-piece function.

With the definition of a grid-aligned N-piece function, we can show that the tree built by Algorithm 3 can fit such a function well when its candidate cuts include the sets $\mathcal{S}_k$.

Theorem 7 *Suppose that the target function g is grid-aligned N-piece and that the candidate cuts in Algorithm 3 include the corresponding cut-point sets. Then, under Assumptions 1-3, for the tree T generated by Algorithm 3 with depth K, we have*

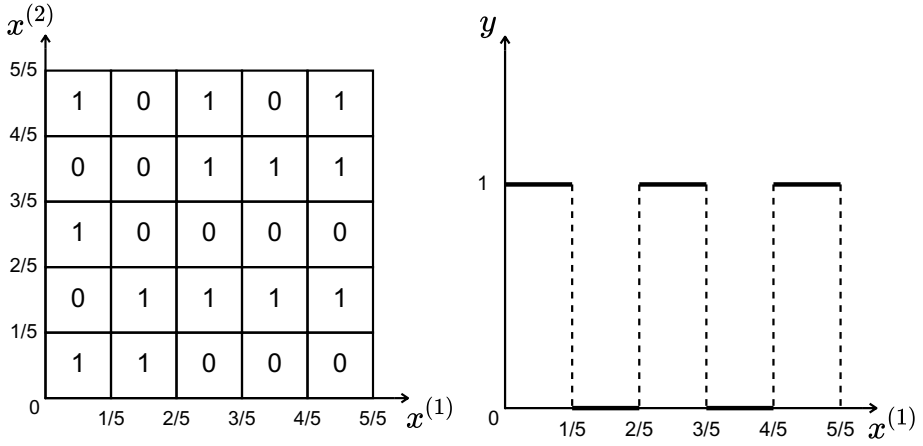


Fig. 2 An example of a 5-piece function from $[0,1]^2$ to $\{0,1\}$. The left figure shows the graph of the function and the right figure displays its projection on $x^{(2)} = 0.9$

$$\mathbb{E}\Big[\mathbb{I}\{T(\mathbf{X}) \neq g(\mathbf{X})\}\Big] \leq \mathcal{O}\left(N^3/K\right) .$$

Theorem 7 shows that Algorithm 3 can fit a grid-aligned N-piece function well under a product distribution. Comparing our result with the previous work (Blanc et al., 2020b), they considered binary features and size-s tree target functions, while we consider the setting where the feature space is real-valued and the target function is N-piece. Above all, we have introduced all the components of GridCART, the whole process of which is detailed in Algorithm 4.

Input: Training dataset $D_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$, maximal tree depth $K_{\max}$, and grid size h
Output: A learned tree $T_{K_{\max}} : \mathbb{R}^d \to \{0,1\}$ with depth at most $K_{\max}$
Procedure:

1: Learn the joint distribution by multiplying histogram estimates of the marginals:

$$\hat{\mathbf{p}}_{\mathbf{X}} \leftarrow \mathbf{HDE}(D_n, h)$$

2: Learn the histogram classifier with grid size h: $g_n^{\text{hist}} \leftarrow \mathbf{LHC}(D_n, h)$
3: Build a tree according to the estimated distribution and classifier

$$T \leftarrow \mathbf{TB}(D_n, \hat{\mathbf{p}}_{\mathbf{X}}, g_n^{\text{hist}}, K_{\max}, h)$$

4: **return** T

Algorithm 4 Grid-based Classification And Regression Tree (GridCART) for binary classification

5.4 Consistency Rate of GridCART

In this subsection, we will show the consistency rate of GridCART. We start with the error decomposition as follows:

$$R(T) - R^\star = \underbrace{\left(R(T) - R(g_n^{\text{hist}})\right)}_{\text{estimation error}} + \underbrace{\left(R(g_n^{\text{hist}}) - R^\star\right)}_{\text{approximation error}}, \tag{1}$$

where we define $R(T) - R(g_n^{\text{hist}})$ and $R(g_n^{\text{hist}}) - R^\star$ as the estimation error and the approximation error, respectively. Note that the definition may be slightly different from the known $R(T) - \inf_{h\in\mathcal{H}} R(h)$ and $\inf_{h\in\mathcal{H}} R(h) - R^\star$, since in other learning algorithms the learning objective is to fit the best classifier in the given hypothesis set, while the objective here is to fit the histogram classifier by a tree. We then decompose the estimation error as follows:

$$\begin{aligned} R(T) - R(g_n^{\text{hist}}) &= \mathbb{E}\left[\mathbb{I}\{T(\mathrm{X}) \neq Y\}\right] - \mathbb{E}\left[\mathbb{I}\{g_n^{\text{hist}}(\mathrm{X}) \neq Y\}\right] \\ &\leq \underbrace{\mathbb{E}_{D_n,\mathrm{p}}\left[\mathbb{I}\{T(\mathrm{X}) \neq g_n^{\text{hist}}(\mathrm{X})\}\right]}_{\text{term (a)}} + \underbrace{\mathbb{E}_{D_n}\left\|\hat{\mathrm{p}}_{\mathrm{X}} - \mathrm{p}_{\mathrm{X}}\right\|_1}_{\text{term (b)}}, \end{aligned} \tag{2}$$

where term (a) measures how the learned tree fits the histogram classifier, and term (b) measures the difference between the estimated distribution $\hat{\mathrm{p}}$ and the ground truth distribution p . We learn a tree on an estimated distribution; if the estimation is accurate enough, and the tree fits well, then the learned tree can perform as well as the histogram classifier. Taking the error bounds in Lemma 5, Lemma 6, and Theorem 7 into (1) and (2), we obtain the consistency rate of GridCART in the following Theorem.

Theorem 8 *(Consistency rate for binary classification) Under Assumptions 1-3, the expected excess error of tree T_{K_n} learned by Algorithm 4 has the following upper bound*

$$\mathbb{E}_{D_n}\left[R(T_{K_n})\right] - R^\star \leq \mathcal{O}\left(\frac{1}{K_n h_n^3} + h_n + \sqrt{\frac{1}{n h_n^d}}\right).$$

Choosing $h_n = \Theta\left(n^{-1/(d+2)}\right)$ and $K_n = \Omega(n^{4/(d+2)})$, we obtain a consistency rate of order $\mathcal{O}\left(n^{-1/(d+2)}\right)$ for GridCART.

Theorem 8 establishes a consistency rate for a greedy, label-dependent tree learning algorithm without imposing structural assumptions on the target function. Table 3 lists the detailed comparison between previous studies and our result on the consistency. Previous consistency rates have primarily focused on randomized trees whose split structures are generated independently of the observed labels. On the other hand, approaches that rely on label-dependent generation have typically relied on strong assumptions and failed to provide a consistency rate. It is worth noting that the parameter K_n in Theorem 8 determines the maximum depth to which the tree can grow. The optimal convergence rate, as demonstrated in the theorem, is achieved by selecting $K_n = \Omega(n^{4/(d+2)})$. However, this choice may lead to inefficient running times. In this study, we set $K_n = \omega(\log^3(n))$ and $h_n = \Theta(1/\log n)$ to strike a balance between efficient running time and consistency.

Theorem 8 provides an upper bound on the excess error of GridCART. To determine whether this rate can be improved, we first recall the classical minimax lower bound for nonparametric classification over Lipschitz conditional probability functions without a margin assumption (Yang, 1999, Theorem 2 and Section V-B).

Table 3 Comparison of our results with previous works.

Algorithms	Feature distributions	Target functions	Results
Pure Random	Any	Any	$\mathcal{O}(n^{-1/(8d+2)})$
Tree			(Gao & Zhou, 2020)
Centered Random	Any	Any	$\mathcal{O}((\ln n/n)^{1/(d+2)})$
Tree			(Gao & Zhou, 2020)
CART	Uniformly random	Additive model	$o(1)$
	in $[0,1]^d$		(Scornet et al., 2015)
CART	Any	Additive model	$\mathcal{O}(1/\log n)$
			(Klusowski & Tian, 2024)
GridCART	Product distribution	Any	$\mathcal{O}(n^{-1/(d+2)})$
			(our result)

Theorem 9 *(Minimax lower bound for Lipschitz classification) Let* X *be uniformly distributed on* $[0,1]^d$*, and let* $\mathrm{Lip}(L)$ *be the class of L-Lipschitz conditional probability functions for some* $L > 0$*. There exist constants* $c > 0$ *and* $n_0 \geq 1$ *such that, for all* $n \geq n_0$*,*

$$\inf_{\hat{f}_n} \sup_{\eta \in \mathrm{Lip}(L)} \left\{ \mathbb{E}_{D_n}\left[R(\hat{f}_n)\right] - R^\star \right\} \geq c n^{-1/(d+2)} ,$$

where the infimum ranges over all measurable classifiers constructed from n i.i.d. samples.

Corollary 10 *(Minimax tightness) Under Assumptions 1-3, the* $\mathcal{O}(n^{-1/(d+2)})$ *rate in Theorem 8 is minimax optimal over all measurable classifiers, including non-tree-based classifiers.*

This is the classical minimax rate for nonparametric classification with an L-Lipschitz conditional probability function and no margin assumption (Stone, 1982; Yang, 1999). The lower bound continues to hold under Assumption 3. The standard bump-function construction that yields this rate applies when X is uniformly distributed on $[0,1]^d$, and the uniform distribution is a special product distribution. Thus, the product-distribution restriction retains a worst-case instance. In the upper-bound analysis, independence among the coordinates of X is used to control the density-estimation term (b) in (2) through Lemma 4. It does not reduce the intrinsic difficulty of estimating an arbitrary L-Lipschitz η, which determines the lower bound. Consequently, Theorem 8 is minimax-optimal over the function class considered here. The GridForest rate in Theorem 14 also matches this lower bound. These minimax conclusions concern the baseline function class without a margin assumption.

5.5 Discussions

Time complexity: Here, we conduct a comparison of the time complexity between CART and our proposed GridCART, with a focus on analyzing the complexity associated with the splitting process at each leaf node. We deliberately exclude the comparison of tree struc-

tures, as the total time required for tree construction simply amounts to the sum of the splitting complexities at all leaf nodes.

As for CART, at each leaf node, all dimensions are traversed, and the samples falling into the leaf are sorted along dimensions. Subsequently, CART calculates the purity gains for all possible cuts. The time complexity of traversing all possible cuts is mainly dominated by the sorting process, which costs time of order $\mathcal{O}(n \log n)$ for each dimension, resulting in a total time of order $\mathcal{O}(dn \log n)$ at each leaf.

On the other hand, GridCART adopts a different approach. It initially estimates the underlying distribution using observed samples with time complexity of order $\mathcal{O}(n)$. When splitting a node, there are d/h_n potential cuts since each dimension allows $1/h_n$ possible cuts. To calculate the purity gain of a specific cut, only one traverse over all the cubes is required, leading to a time complexity of order $\mathcal{O}(1/h_n^d)$. Consequently, the total complexity of one split is $\mathcal{O}(d/h_n^{(d+1)})$. It is possible to achieve $\mathcal{O}(n + dn^{(d+1)/(d+2)})$, provided that $h_n = \Theta(n^{-1/(d+2)})$ as demonstrated in Theorem 8, making GridCART marginally faster than CART.

Based on the above analysis, we can conclude that GridCART is faster than CART for splitting since GridCART converts data to $o(n)$ cubes and stores data into an ordered form, in which no sorting is required which eliminates a log term.

General rid-based decision tree earning: The core concepts of GridCART lie in the construction of a tree to fit the estimated feature distribution and target function. This process works no matter which density estimator and classifier are used. Hence, we have the flexibility to incorporate any density estimator and classifier.

Building upon the above discussions, we call this broader paradigm Grid-based Decision Tree Learning (**GridDTL**). For example, GridDTL may replace the histogram density estimator with a kernel density estimator (Devroye et al., 1997; Wasserman, 2006) or a Gaussian mixture model (Hastie et al., 2009). Once the chosen density estimator is consistent and satisfies the estimation requirements used in (2), the argument of Theorem 8 yields consistency. In Section 6, we demonstrate how GridDTL can be effectively applied to popular ensemble methods.

6 Applications to Ensemble Learning

In this section, we will show that the proposed grid-based learning paradigm can be applied to ensemble methods, such as random forests and gradient boosting decision trees, which are extensively employed in practical settings and have proven highly successful across numerous real-world applications. Additionally, we highlight the consistency of the proposed grid-based ensemble methods.

6.1 Grid-based Random Forests

Random forests (Breiman, 2001) are ensemble methods that combine multiple decision trees. The main idea behind this approach is to train several decision trees on different subsets of the training data and then average their predictions. The process of constructing the forest can be summarized as follows:

1. Randomly sample a subset of training data, with replacement, that has the same size as the original training data.
2. Train a decision tree using the sampled data. The growth process of the tree is as follows:

   2.1 If the maximum tree depth is reached or all the data points have the same labels, the tree construction is stopped.
   2.2 Randomly select m features from the total d features and determine the best split based on these selected features.
   2.3 Split the data based on the best split and recursively repeat the above steps for the two resulting subsets.

3. Repeat the above steps for M times.
4. Output the average of the predictions of the M decision trees.

The original random forests are based on CART, so the theoretical guarantees are as weak as CART, which was discussed in Section 1. In this section, we propose Grid-based Random Forests (GridForest), an ensemble of GridCART that admits a consistency guarantee.

6.1.1 Diversity Encouragement

In order to incorporate the proposed GridCART technique into random forests, it is crucial to emphasize the importance of *diversity*, as it serves as a fundamental element in ensemble learning. However, it should be noted that the conventional grid-based learner often exhibits a tendency to produce similar outcomes due to its fixed grid structure. For instance, as depicted in Figure 3, the decision boundaries of the decision trees tend to align, resulting in minimal enhancement when combined as an ensemble. This lack of diversity is unfavorable and undermines the desired objective.

To avoid diversity vanishing, we randomly shift the grid structure in each tree, and the shifted grid structure is used to train the decision tree. Formally, when generating a tree f_t, we first generate $\delta_1, \delta_2, \ldots, \delta_d$ independently from $\mathcal{U}[0, \delta_{\max}]$, where $\delta_{\max}$ is a hyperparameter. Then, both the histogram density estimation and histogram classifier are learned

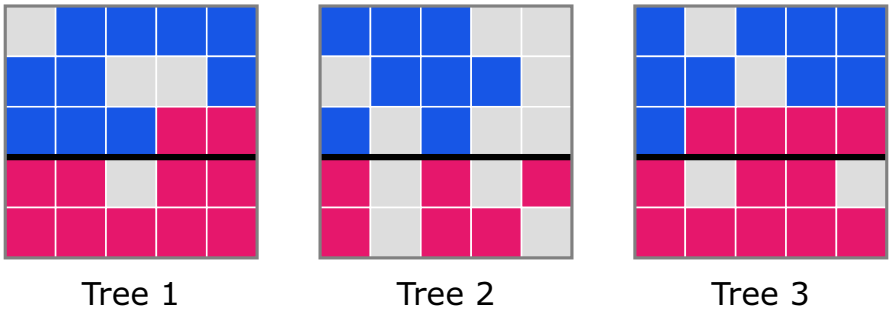


Fig. 3 Illustrations of diversity vanishing. The ground truth decision function is defined by $f^\star = \mathbb{I}\{x_2 \leq 0.5\}$. Blue and red rectangles are two classes of data points, while gray rectangles are the undetermined regions. As shown in the figure, although bagging is applied, the decision boundaries tend to be the same

based on the shifted grid structure. For histogram density estimation, we replace the step in Algorithm 1 [Line 2] as follows:

$$\tilde{\mathcal{B}}_k \leftarrow \left\{ [0, \delta_k), [\delta_k, h + \delta_k), [h + \delta_k, 2h + \delta_k), \ldots, [(1/h - 1)h + \delta_k, 1] \right\},$$

where $\tilde{\mathcal{B}}_k$ is the bin of the histogram for dimension k.

The histogram classifier in Algorithm 2 [Line 1] is modified by

$$\tilde{\mathfrak{C}} \leftarrow \left\{ \times_{k=1}^{d} b_k \,\middle|\, b_k \in \tilde{\mathcal{B}}_k \right\},$$

where $\tilde{\mathfrak{C}}$ represents the Cartesian product of the shifted intervals. The remaining steps of the shifted histogram density estimation and histogram classifier remain the same as in Algorithm 1 and Algorithm 2, respectively. Figure 4 provides a visual representation of the shifted grid structure.

As shown in Figure 4, shifting the grid structure can encourage diversity, enabling the ensemble to learn a more complex decision function. The complete process of GridForest is summarized in Algorithm 5.

Input: Training dataset D_n, number of trees M, number of random features m, maximal tree depth $K_{\max}$, grid size h, maximal shift distance $\delta_{\max}$
Output: A binary random-forest classifier
Procedure:

1: **for** $t = 1, \ldots, M$ **do**
2: Sample a training dataset D_t from D_n with replacement, with $|D_t| = |D_n|$
3: Generate $\delta_1, \delta_2, \ldots, \delta_d$ independently from Uniform($[0, \delta_{\max}]$)
4: Learn a histogram density estimation $\hat{\mathbf{p}}_t$ on D_t using the shifted grid structure
5: Learn a histogram classifier $g_{n,t}^{\text{hist}}$ on D_t using the shifted grid structure
6: Train a decision tree f_t with maximal depth $K_{\max}$ using $\hat{\mathbf{p}}_t$ and $g_{n,t}^{\text{hist}}$; at each internal node, independently sample m candidate features uniformly without replacement
7: **end for**
8: **return** $f(\mathbf{x}) \leftarrow \mathbb{I}\{M^{-1} \sum_{t=1}^{M} f_t(\mathbf{x}) > 1/2\}$

Algorithm 5 Grid-based Random Forest (GridForest)

6.1.2 Consistency Guarantee

We begin with the formulation of GridForest. Let the average voting score of GridForest be

$$\bar{f}(\mathrm{x};\ \ , \boldsymbol{\delta}) = M^{-1} \sum_{t=1}^{M} f_t(\mathrm{x};\ \ _t, \boldsymbol{\delta}_t).$$

The binary classifier returned by Algorithm 5 is $f(\mathrm{x};\ \ , \boldsymbol{\delta}) = \mathbb{I}\{\bar{f}(\mathrm{x};\ \ , \boldsymbol{\delta}) > 1/2\}$. It depends on random variables $\ \ = (\ _1,\ _2, \ldots,\ _M)$ and $\boldsymbol{\delta} = (\boldsymbol{\delta}_1, \boldsymbol{\delta}_2, \ldots, \boldsymbol{\delta}_M)$, which are both independent of the training data D_n. Note that $\ _t$ comes from the random sampling and random feature

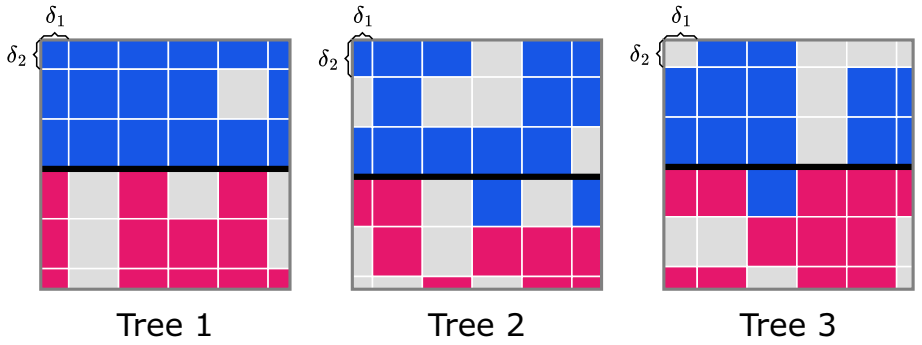


Fig. 4 Illustrations of diversity encouragement. The ground truth decision function is defined by $f^\star = \mathbb{I}\{x^{(2)} \leq 0.5\}$. Blue and red rectangles are two classes of data points, while gray rectangles are the undetermined regions. As shown in the figure, the decision boundaries of the three decision trees are different, so their ensemble can eventually approximate the ground truth decision function

selection, and $\boldsymbol{\delta_t}$ comes from the random shifting. $\boldsymbol{\delta_t} = 0$ means that the grid structure is not shifted, which recovers the original GridCART.

As GridCART has been shown to be consistent in Section 5, the key to showing the consistency of GridForest is to theoretically explain the effect of randomness and $\boldsymbol{\delta}$, which distinguishes GridForest from GridCART. The dependence induced by bootstrap bagging is difficult to analyze directly. We therefore analyze a subsampling variant in which each tree is trained on a_n observations sampled without replacement. Algorithm 5 retains bootstrap bagging for the empirical implementation, whereas the theorem below concerns this subsampling variant. Klusowski and Tian (Klusowski & Tian, 2024) analyze random feature selection for CART regression forests with additive bounded-variation comparators. Their oracle inequality is specific to that setting, so we establish the corresponding effect directly for the Influence-based splitting rule used by GridCART.

Lemma 11 *(Random-feature tree-fitting bound) Let g be a grid-aligned N-piece binary target. Consider the depth-K tree generated by Algorithm 3, except that, at every internal node, the candidate set $\mathcal{S}_t \subseteq [d]$ is sampled independently and uniformly from all subsets of size m, and the split is optimized only over features in $\mathcal{S}_t$. Under Assumptions 1–3,*

$$\mathbb{E}_{\boldsymbol{\Theta}}\mathbb{E}_{\hat{\mathrm{p}}}\left[\mathbb{I}\{T_{K,\boldsymbol{\Theta}}(\mathrm{X}) \neq g(\mathrm{X})\}\right] \leq \mathcal{O}\left(\frac{d}{m}\frac{N^3}{K}\right).$$

Lemma 12 *Let $T_1, \ldots, T_M$ be arbitrary binary classifiers, and define*

$$F_M(\mathrm{x}) = \mathbb{I}\left\{\frac{1}{M}\sum_{b=1}^{M} T_b(\mathrm{x}) > \frac{1}{2}\right\}.$$

Then

$$R(F_M) - R^\star \leq \frac{2}{M}\sum_{b=1}^{M}\left(R(T_b) - R^\star\right).$$

The proofs are given in Appendix F. A uniformly selected subsample of a_n distinct observations from the i.i.d. dataset D_n is itself i.i.d. in its unconditional distribution. Lemmas 5 and 6 can therefore be applied with sample size a_n. Combining those estimation bounds with Lemma 11 gives the following base-tree bound when the grid is not shifted:

$$\mathbb{E}_{D_n,\boldsymbol{\Theta}}[R(T_{K_n,\boldsymbol{\Theta}})] - R^\star \leq \mathcal{O}\left(\frac{d}{m}\frac{1}{K_n h_n^3} h_n \sqrt{\frac{1}{a_n h_n^d}}\right). \tag{3}$$

Lemma 12 transfers this bound to the binary forest classifier, up to an absolute factor of two. The bound does not decrease with the number of trees M and therefore does not establish a theoretical diversity benefit. Empirically, aggregating accurate and diverse learners can improve predictive performance (Breiman, 2001; Zhou, 2012), but explaining this effect remains an open theoretical problem.

We will then show that the effect of random shifting is negligible when the sample size n is large enough. Intuitively, the size of cubes near the boundary of the feature space $[0,1]^d$ may be too small to reach a sufficient performance, thus samples in these cubes may be wrongly predicted. But, fortunately, the probability of a sample falling into these cubes tends to be zero when grid size h_n tends to zero, which happens when sample size n tends to infinity. Formally, we have the following proposition:

Proposition 13 *Under Assumptions 2 and 3, the density* p *is bounded on* $[0,1]^d$. *Define*

$$\bar{\mathrm{p}} \triangleq \sup_{\mathrm{x}\in[0,1]^d} \mathrm{p}(\mathrm{x}) < \infty.$$

For $0 \leq \delta_{\max} \leq 1/2$, *we have*

$$\Pr\left[\exists k \in [d] \quad \text{s.t.} \quad \mathrm{X}^{(k)} \in [0,\delta_{\max}] \cup [1-\delta_{\max},1]\right] \leq 2d\bar{\mathrm{p}}\delta_{\max} = \mathcal{O}(d\delta_{\max}).$$

The proof is given in Appendix G. Proposition 13 shows that if we do not shift the grid structure too much, the shifting will not hurt the performance too much, and vice versa. A smaller δ_{max} reduces the boundary term in Proposition 13, whereas a larger δ_{max} may increase empirical diversity among the trees. In practice, δ_{max} can be selected to balance this theoretical error term and the empirical effect of diversity.

With the above analysis, we can show the consistency of GridForest.

Theorem 14 *(Consistency rate of GridForest for binary classification) Under Assumptions 1-3, the expected excess error of grid-based forest* $f(\boldsymbol{\Theta},\boldsymbol{\delta})$ *learned by the subsampling variant of Algorithm 5 described above has the following upper bound*

$$\mathbb{E}_{D_n,\boldsymbol{\Theta},\boldsymbol{\delta}}\left[R(f(\boldsymbol{\Theta},\boldsymbol{\delta}))\right] - R^\star \leq \mathcal{O}\left(\frac{d}{m}\frac{1}{K_n h_n^3} + h_n + \sqrt{\frac{1}{a_n h_n^d}} + d\delta_{\max,n}\right).$$

By choosing $h_n = \Theta\left(n^{-1/(d+2)}\right)$, $K_n = \Omega\left((d/m)n^{4/(d+2)}\right)$, $a_n = \Theta(n)$, *and* $\delta_{\max,n} = \Theta\left(n^{-1/(d+2)}\right)$, *we obtain a consistency rate of order* $\mathcal{O}\left(n^{-1/(d+2)}\right)$.

For the subsampling variant described above, the proof of Theorem 14 follows by combining the base-tree bound (3), Lemma 12, and Proposition 13. This result gives an upper bound of the same asymptotic order as GridCART, with additional constants and error terms due to feature subsampling, subsample size, and grid shifting. By Corollary 10, this order is already minimax-optimal under Assumptions 1-3, so the rate itself cannot be tightened by a sharper analysis of either GridCART or GridForest. This is a different statement from an ensemble improvement, however: the matching lower bound shows that neither a single tree nor the forest can be pushed below this worst-case floor, but it does not show that the forest actually improves on the single tree, since both are shown to attain the same floor. Quantifying such a diversity benefit would require an assumption beyond Lipschitz smoothness and product features, and is left open here. It should be noted that this does not imply that GridForest performs no better than GridCART, as the analysis is based on worst-case scenarios. Empirically, ensembles of diverse decision trees can outperform a single decision tree (Breiman, 2001; Zhou, 2012), and GridForest is designed to exploit this mechanism through random feature selection, subsampling, and grid shifting. This empirical motivation is distinct from Theorem 14, which does not establish an improvement over a single GridCART tree. The GridForest construction can also be formulated for multiclass classification and regression. The present ensemble analysis focuses on binary classification, while Section 7 and Appendix E establish the corresponding extensions for the GridCART base learner.

The preceding rate is minimax optimal without a restriction on the probability mass near the Bayes decision boundary. We next give an optional refinement under the standard margin condition of Audibert and Tsybakov (2007).

Assumption 4 *(Margin condition)* There exist constants $C_0 > 0$ and $\alpha \geq 0$ such that, for every $\tau > 0$,

$$\Pr_{\mathrm{p}}\left[0 < |\eta(\mathrm{X}) - 1/2| \leq \tau\right] \;\leq\; C_0 \tau^{\alpha} \,.$$

Assumption 4 limits the probability mass in a neighborhood of the Bayes decision boundary. The case $\alpha = 0$ imposes no effective restriction, while a larger α describes a sharper separation between the two classes.

Corollary 15 *(Margin fast rate for GridForest) Suppose Assumptions 1–3 and 4 hold. Suppose also that there exist constants* $0 < p_{\min} \leq p_{\max} < \infty$ *such that*

$$p_{\min} \leq \mathrm{p}(\mathrm{x}) \leq p_{\max}, \qquad \mathrm{x} \in [0,1]^d \,.$$

If $0 \leq \delta_{\max,n} \leq h_n$*, then for some constant* $c > 0$ *the expected excess error of GridForest satisfies*

$$\mathbb{E}_{D_n,\boldsymbol{\Theta},\boldsymbol{\delta}}\big[R(f(\boldsymbol{\Theta},\boldsymbol{\delta}))\big] - R^{\star} \leq \mathcal{O}\left(\frac{d}{m}\frac{1}{K_n h_n^3} + h_n^{\alpha+1} + \left(\frac{1}{a_n h_n^d}\right)^{(\alpha+1)/2} + d\delta_{\max,n} + \frac{d}{h_n} e^{-c a_n h_n}\right) .$$

Choosing $a_n = \Theta(n)$, $h_n = \Theta(n^{-1/(d+2)})$,

$$K_n = \Omega\left(\frac{d}{m} n^{(\alpha+4)/(d+2)}\right) \quad \text{and} \quad \delta_{\max,n} = \mathcal{O}\left(n^{-(\alpha+1)/(d+2)}\right)$$

yields

$$\mathbb{E}_{D_n,\boldsymbol{\Theta},\boldsymbol{\delta}}\left[R(f(\boldsymbol{\Theta},\boldsymbol{\delta}))\right] - R^\star = \mathcal{O}\left(n^{-(\alpha+1)/(d+2)}\right) .$$

Corollary 15 recovers the rate of Theorem 14 when $\alpha = 0$ and is strictly faster when $\alpha > 0$. The lower-density bound is used to compare the fitted-tree error under the estimated feature distribution with its population counterpart.

6.2 Grid-based Gradient Boosting Decision Tree

Gradient Boosting Decision Trees (GBDT) are a widely used ensemble learning method that combines the predictions of multiple weak learners, typically decision trees. Several implementations of GBDT have been developed, including XGBoost (Chen & Guestrin, 2016), LightGBM (Ke et al., 2017), and CatBoost (Prokhorenkova et al., 2018). Here we extend GridCART to the boosting setting by taking grid-based trees as the weak learners, and refer to the resulting method as the Grid-based Gradient Boosting Decision Tree (**GridGBDT**).

6.2.1 The Algorithmic Details of GridGBDT

GBDT is an iterative greedy algorithm that constructs an ensemble in a forward stage-wise manner. At each iteration, a new weak learner is trained to fit the negative gradient of the loss function, thereby greedily reducing the ensemble loss. Formally, GridGBDT is defined for a generic differentiable loss $\ell(y,u)$ evaluated at $u = F_{t-1}(\mathrm{X}_i)$, and round t fits a tree to the corresponding negative gradients. The last iterate is turned into an output by a prediction map Ψ. As in standard GBDT implementations, we instantiate the pair (ℓ, Ψ) as the logistic loss with a log-odds threshold for binary classification,

$$\ell(y,u) = \log(1+e^u) - yu\,, \qquad \Psi(u) = \mathbb{I}\{u > 0\}\,,$$

and as the square loss with the identity prediction map for regression,

$$\ell(y,u) = \frac{1}{2}(y-u)^2\,, \qquad \Psi(u) = u\,.$$

Incorporating GridCART into the boosting framework requires additional care. At round t, the tree T_t is trained to fit the negative gradients $\rho_{t,i}$, while the current ensemble F_{t-1} is itself constructed from trees fitted to the gradients generated in previous rounds. Consequently, the N-piece structure required by Theorem 7 does not arise automatically. In particular, there is no a priori guarantee that the negative-gradient target can be represented as a piecewise-constant function of x with a controlled number of pieces. Addressing this difficulty is the key step in establishing a consistency guarantee for the GridGBDT extension proposed here.

To recover the required piecewise-constant structure, at each boosting round, **TB** fits a histogram approximation of the negative gradients rather than the raw gradients themselves. Specifically, we first compute the sample negative gradients

$$\rho_{t,i} \triangleq -\frac{\partial \ell(Y_i, u)}{\partial u}\Big|_{u=F_{t-1}(\mathrm{X}_i)}, \qquad i = 1, \ldots, n.$$

We then independently sample a random grid shift $\boldsymbol{\delta}_t$ and average these values within each cube $\mathcal{C}$ of the shifted grid. This gives the piecewise-constant target

$$\bar{\rho}_t(\mathcal{C}) \triangleq \frac{\sum_{i=1}^n \rho_{t,i} \mathbb{I}\{\mathbf{X}_i \in \mathcal{C}\}}{\sum_{i=1}^n \mathbb{I}\{\mathbf{X}_i \in \mathcal{C}\}}.$$

For any $\mathrm{x} \in \mathcal{C}$, we define the target as follows:

$$\bar{\rho}_t(\mathrm{x}) = \bar{\rho}_t(\mathcal{C}).$$

Thus, $\bar{\rho}_t$ is a piecewise-constant function on the feature space. This construction reuses Algorithm 2 without its final thresholding step and with the negative gradients in place of the labels. Resampling the shift at each round allows successive trees to use different grid boundaries. We also introduce a shrinkage parameter $\nu \in (0, 1]$. Algorithm 6 summarizes the resulting procedure.

Input: Training dataset D_n, number of rounds M, maximal tree depth K, grid size h_n, shrinkage $\nu \in (0, 1]$, maximal shift distance $\delta_{\max}$, differentiable loss ℓ, prediction map Ψ
Output: A grid-based GBDT estimator $\hat{f}_M$
Procedure:

1: Initialize $F_0(\mathbf{x}) \leftarrow \arg\min_c \sum_{i=1}^n \ell(Y_i, c)$ for all $\mathbf{x}$
2: **for** $t = 1, \ldots, M$ **do**
3: Compute the sample negative gradients $\rho_{t,i} \leftarrow -\partial \ell(Y_i, u)/\partial u\big|_{u=F_{t-1}(\mathbf{X}_i)}$ for $i = 1, \ldots, n$
4: Sample an independent random shift $\boldsymbol{\delta}_t \sim \mathcal{U}([0, \delta_{\max}]^d)$
5: Learn the shifted density estimate $\hat{\mathbf{p}}_t \leftarrow \mathbf{HDE}(D_n, h_n)$ using the $\boldsymbol{\delta}_t$-shifted grid
6: Construct the gradient histogram $\bar{\rho}_t$ by applying Algorithm 2 without thresholding to $\{(\mathbf{X}_i, \rho_{t,i})\}_{i=1}^n$ on the $\boldsymbol{\delta}_t$-shifted grid
7: Build a depth-K grid regression tree $T_t \leftarrow \mathbf{TB}(D_n, \hat{\mathbf{p}}_t, \bar{\rho}_t, K, h_n)$, splitting by the regression purity gain $\Delta_{\mathrm{reg}}^{\mathrm{Inf}}$ of Section 7, restricted to cuts on the $\boldsymbol{\delta}_t$-shifted grid
8: Update: $F_t(\mathbf{x}) \leftarrow F_{t-1}(\mathbf{x}) + \nu T_t(\mathbf{x})$
9: **end for**
10: **return** $\hat{f}_M(\mathbf{x}) \leftarrow \Psi\big(F_M(\mathbf{x})\big)$

Algorithm 6 Grid-based Gradient Boosting Decision Tree (GridGBDT)

6.2.2 Consistency Guarantee

Algorithm 6 is stated for a generic pair (ℓ, Ψ), but the analysis below instantiates it with the square loss $\ell(y, u) = \frac{1}{2}(y - u)^2$ and the prediction map $\Psi(u) = \mathbb{I}\{u > 1/2\}$. For this analyzed specialization, the update is $F_t(\mathrm{x}) \leftarrow \mathrm{clip}_{[0,1]}(F_{t-1}(\mathrm{x}) + \nu T_t(\mathrm{x}))$. The clipping is required by the proof because it keeps every iterate bounded in [0, 1], as assumed by the

residual and contraction arguments. The negative gradient of Line 3 is then simply the residual $\rho_{t,i} = Y_i - F_{t-1}(\mathrm{X}_i)$ and the initializer of Line 1 is $n^{-1}\sum_i Y_i$, so GridGBDT boosts a real-valued surrogate of η and thresholds it only at the end, rather than solving a regression problem.

Several studies have examined the theoretical foundations of boosting methods, including Zhang and Yu (2005); Wang et al. (2008); Zhou (2012); Gao and Zhou (2013); Lyu et al. (2019). Our argument specializes the boosting-with-early-stopping framework of Zhang and Yu (2005) to the grid-tree base learner. Rather than invoking a generic weak-learning or edge condition, we use the fact that GridCART's construction provides a fitting guarantee for *any* bounded N-piece target (Theorem 16). Under the unshifted choice used for the analysis, histogramming the residuals on the common grid ensures that the target at every round is an N-piece function with $N = 1/h_n$ (Lemma 17).

Boosting the square loss for classification is a standard construction rather than an artifact of the analysis, since it is the LS-Boost of Friedman (2001) and the L_2-Boosting of Bühlmann and Yu (2003), whose classification behavior has been studied in precisely this plug-in form. It is also the tractable choice for a *grid-based* weak learner, because the two properties the construction relies on hold exactly rather than approximately. The negative gradient $y - u$ is affine in the current iterate, so the residual averaged over a grid cube is exactly $\hat{\eta}_n(\mathcal{C}) - F_{t-1}(\mathcal{C})$ (Lemma 17) and every round targets the same fixed histogram $\hat{\eta}_n$, which supplies the N-piece structure required by Theorem 16 and which the raw gradients of a general loss do not provide. Moreover $\partial_u^2 \ell \equiv 1$, so the induced potential is exactly quadratic in the grid-cube values and an inexact gradient step contracts it geometrically at a rate depending only on the shrinkage ν (Lemma 18). The per-cube minimizer $\hat{\eta}_n(\mathcal{C}) \in [0,1]$ is moreover always finite, so the quantity the contraction controls is the same L^2 quantity that the final excess-error conversion consumes.

Algorithm 6 allows the grid to be shifted independently at each boosting round. The consistency result below analyzes the admissible choice $\delta_{\max} = 0$, for which $\boldsymbol{\delta}_t = 0$ at every round. All trees then share one unshifted grid and one density estimate, denoted by $\hat{\mathrm{p}}$. We write $\hat{\eta}_n$ for the unthresholded histogram regressor obtained by averaging the original labels Y on this grid, as defined in Section 5.2.

The proof proceeds in three steps. We first establish a regression-tree approximation bound for bounded real-valued targets. We then use the common grid to express every residual histogram in terms of the fixed target $\hat{\eta}_n$ and derive a contraction bound for the boosting recursion. Finally, we combine this contraction with the usual plug-in inequality to control the classification excess error. Appendix I contains the proofs.

The first step extends Theorem 7 from binary-valued targets to bounded real-valued targets such as $\bar{\rho}_t$. Its proof requires two replacements for arguments that are specific to classification. First, the Efron–Stein inequality (Boucheron et al., 2013) replaces the Bourgain-type inequality of Lemma 28. Since a GridCART leaf is an axis-aligned box, conditioning a product distribution on a leaf preserves the product structure. The Efron–Stein inequality therefore gives

$$\mathrm{Var}(f \mid \mathrm{X} \in t) \le \sum_{k=1}^{d} \mathrm{Inf}_k^{\mathrm{Var}}[f \mid t] = d\,\mathcal{G}(t)$$

on every leaf t. Second, Lemma 34 replaces the binary split-gain bound of Lemma 27. It shows that an N-piece target taking values in $[-B, B]$ admits a split at a piece boundary whose purity gain is at least $\mathrm{Var}^2[f]/(4B^2(N-1)^2)$. Substituting these two bounds into the depth-K potential argument from Appendix D yields the following result.

Theorem 16 *Suppose the target $g : \mathcal{X} \to [-B, B]$ is grid-aligned N-piece (Definition 3) and that the candidate cuts include the corresponding cut-point sets. Then, under Assumptions 1–3, the regression tree T generated by Algorithm 3 with the impurity* Var *in place of G (Section 7) and depth K satisfies*

$$\mathbb{E}_{\hat{\mathrm{p}}}\big[(T(\mathrm{X}) - g(\mathrm{X}))^2\big] \;\le\; \mathcal{O}\left(\frac{B^2 N^3}{K}\right).$$

This theorem supplies the per-round approximation guarantee needed for GridGBDT. It also makes explicit the fitting-error term left implicit in the proof of Theorem 22.

The second step uses the common grid to identify the target fitted at each round. Since **TB** splits only at grid points, every tree and every clipped iterate is constant on each grid cube. Averaging the residuals within a cube therefore gives the following exact identity.

Lemma 17 *Let F_{t-1} be an iterate generated by Algorithm 6 with $\delta_{\max} = 0$. Then F_{t-1} is constant on every unshifted grid cube $\mathcal{C} \in \mathcal{J}$, and the residual histogram satisfies $\bar{\rho}_t(\mathcal{C}) = \hat{\eta}_n(\mathcal{C}) - F_{t-1}(\mathcal{C})$ for every $\mathcal{C} \in \mathcal{J}$.*

Thus, although the residual histogram changes with t, it is always the difference between the current iterate and the same fixed histogram target $\hat{\eta}_n$. Theorem 16 applies with $B = 1$ and $N = 1/h_n$. Indeed, clipping ensures that both $\hat{\eta}_n(\mathcal{C})$ and $F_t(\mathcal{C})$ lie in $[0, 1]$. We measure the discrepancy from $\hat{\eta}_n$ using

$$e_t \;\triangleq\; \Phi(F_t) \;=\; \sum_{\mathcal{C}\in\mathcal{J}} \Pr_{\hat{\mathrm{p}}}[\mathrm{X} \in \mathcal{C}]\,\big(\hat{\eta}_n(\mathcal{C}) - F_t(\mathcal{C})\big)^2 .$$

On the grid cubes, the update in Algorithm 6 is an inexact gradient step for this quadratic potential. The tree approximation bound controls the inexactness and gives the following recursion.

Lemma 18 *Suppose every round of Algorithm 6 satisfies the guarantee of Theorem 16 with $B = 1$, $N = 1/h_n$, i.e., $\mathbb{E}_{\hat{\mathrm{p}}}\big[(T_t(\mathrm{X}) - \bar{\rho}_t(\mathrm{X}))^2\big] \le \gamma$ for $\gamma = \mathcal{O}\big(1/(Kh_n^3)\big)$. Then*

$$\sqrt{e_t} \;\le\; (1-\nu)\sqrt{e_{t-1}} + \nu\sqrt{\gamma}\,, \qquad t = 1, \dots, M\,,$$

and consequently $\sqrt{e_M} \le \sqrt{\gamma} + (1-\nu)^M \sqrt{e_0}$, with $e_0 \le 1/4$.

The identity

$$\hat{\eta}_n - F_t = (1-\nu)(\hat{\eta}_n - F_{t-1}) + \nu(\bar{\rho}_t - T_t)$$

explains the recursion. Cauchy–Schwarz controls the approximation term, and clipping cannot increase the distance to $\hat{\eta}_n(\mathcal{C}) \in [0, 1]$. Consequently, the bound decreases geomet-

rically with M until it reaches the accuracy level certified for the base learner, which is $\gamma = \mathcal{O}(1/(Kh_n^3))$.

The final step converts this squared-error bound into a classification guarantee. Since F_M takes values in $[0, 1]$, the classifier $\hat{f}_M = \mathbb{I}\{F_M > 1/2\}$ is a plug-in classifier. We apply Lemma 24 directly to F_M, rather than comparing $\hat{f}_M$ with the thresholded histogram g_n^{hist}. The direct route does not require a margin condition. Pointwise, a plug-in classifier can disagree with the Bayes classifier only when $|\eta - 1/2| \le |\eta - F_M|$. The resulting bound depends on $\sqrt{e_M}$ because $\mathbb{E}_{\hat{\mathrm{p}}}|\hat{\eta}_n - F_M| \le \sqrt{e_M}$. Combining this term with the approximation and estimation errors of the histogram gives the main result.

Theorem 19 *(Consistency rate of GridGBDT for binary classification) Under Assumptions 1–3, fix any shrinkage $\nu \in (0,1)$ and let $\hat{f}_M$ be the output of Algorithm 6 with $\delta_{\max} = 0$, so that all boosting rounds share the same unshifted grid. The expected excess error satisfies*

$$\mathbb{E}_{D_n}\big[R(\hat{f}_M)\big] - R^\star \;\le\; \mathcal{O}\left(\sqrt{\frac{1}{Kh_n^3}} + (1-\nu)^M + h_n + \sqrt{\frac{1}{nh_n^d}}\right).$$

Choosing $h_n = \Theta\big(n^{-1/(d+2)}\big)$, $K = K_n = \Omega\big(n^{5/(d+2)}\big)$, and $M = M_n = \Omega(\log n)$, we obtain a consistency rate of order $\mathcal{O}\big(n^{-1/(d+2)}\big)$ for GridGBDT, matching the rate of GridCART (Theorem 8) under the same assumptions.

The proof is given in Appendix I. This result provides a guarantee for the unshifted specialization of GridGBDT that was absent from the conference paper. The four terms in the bound have distinct sources. The first is the error certified for the tree fitted at each round. The second is the optimization error after M rounds. The last two are respectively the approximation and estimation errors of the histogram.

Three consequences are worth noting.

1. GridGBDT is consistent under the same assumptions as GridCART and GridForest. No condition is imposed on the mass near the decision boundary. The assumption-free conversion from squared error to classification error introduces $\sqrt{e_M}$, which leads to the depth requirement $K_n = \Omega(n^{5/(d+2)})$. GridCART instead requires $K_n = \Omega(n^{4/(d+2)})$ in Theorem 8.
2. For fixed $\nu \in (0,1)$, the optimization term $(1-\nu)^M$ decreases geometrically at a rate independent of n, d, and h_n. Thus, $M_n = \Omega(\log n)$ makes this term negligible at the target statistical rate. The present bound does not improve further once the certified base-learner error dominates.
3. The common-grid structure is essential to the current proof because it allows every boosting round to target the same histogram estimator.

For comparison, a sufficiently strong empirical margin condition can reduce the required tree depth. This condition is not used in Theorem 19 and differs from the population margin condition in Assumption 4.

Assumption 5 *(Empirical margin condition)*] There exist constants $C_0 > 0$ and $\beta \ge 0$ such that, for every $\tau > 0$,

$$\Pr_{\hat{\mathrm{p}}}\left[\,|\hat{\eta}_n(\mathrm{X}) - 1/2| < \tau\,\right] \;\le\; C_0\,\tau^{\beta}\,.$$

Proposition 20 *(Refinement under a margin condition) Suppose Assumption 5 also holds with exponent $\beta \ge 0$. In the setting of Theorem 19, the combined tree-fitting and optimization contribution may then be bounded by*

$$\mathcal{O}\left(\left(\frac{1}{Kh_n^3} + (1-\nu)^{2M}\right)^{\beta/(\beta+2)}\right).$$

For $\beta > 0$, the choice $K_n = \Omega\big(n^{(4\beta+2)/(\beta(d+2))}\big)$ suffices to retain the rate $\mathcal{O}(n^{-1/(d+2)})$.
Since $(4\beta+2)/\beta = 4 + 2/\beta$, this refinement reduces the assumption-free depth requirement when $\beta > 2$ and approaches the GridCART requirement $\Omega(n^{4/(d+2)})$ as $\beta \to \infty$.

Theorem 19 is stated for binary classification. The same argument applies to the unthresholded regression histogram in Section 7. In that setting no thresholding step is needed, and the regression risk is controlled directly by the squared-error potential e_M. Applying the construction separately to each class gives the corresponding multiclass extension.

7 Beyond Binary Classification

In this section, we extend GridCART to multiclass classification and regression.

7.1 Multiclass Classification

We consider multiclass classification with $\mathcal{Y} = \{1, 2, \ldots, m\}$ for $m \in \mathbb{N}^+$. We first define the conditional G-Influence for multiclass classification.

Definition 4 (*Conditional G-Influence for class c*) Suppose that X is drawn from a product distribution. Conditional Influence of function $f : \mathcal{X} \to \mathcal{Y}$ on variable $\mathrm{X}^{(k)}$ *w.r.t.* class c at leaf t is defined by

$$\mathrm{Inf}_k^G[f, c \mid t] \triangleq \mathbb{E}_{\mathrm{p}^{(j\neq k)}}\left[G\Big(\Pr_{\mathrm{p}^{(k)}}\left[f(\mathrm{X}) = c \mid t\right]\Big)\right],$$

where we here abbreviate $\mathbb{E}_{\mathrm{p}^{(k)}}\Big[\mathbb{I}\{f(\mathrm{X}) = c\} \mid \mathrm{X} \in t\Big]$ as $\Pr_{\mathrm{p}^{(k)}}\left[f(\mathrm{X}) = c \mid t\right]$.
With the definition of Influence at hand, we define the impurity measure as follows:

$$\mathcal{G}(t) = \frac{1}{dm}\sum_{k=1}^{d}\sum_{c=1}^{m}\mathrm{Inf}_k^G[f, c \mid t]\,.$$

Then, every leaf node is split by solving

$$\max_{k,s}\; \mathcal{G}(t) - w_{t_L}\mathcal{G}(t_L) - w_{t_R}\mathcal{G}(t_R)\,.$$

Note that when $m = 2$, by using the fact that $G(x) = G(1-x)$ for all impurity functions G, we have

$$\begin{aligned}\mathrm{Inf}_k^G[f, 1 \mid t] &= \mathbb{E}_{\mathrm{p}^{(j\neq k)}}\Big[G\Big(\Pr_{\mathrm{p}^{(k)}}[f(\mathrm{X}) = 1 \mid t]\Big)\Big] \\ &= \mathbb{E}_{\mathrm{p}^{(j\neq k)}}\Big[G\Big(1 - \Pr_{\mathrm{p}^{(k)}}[f(\mathrm{X}) = 2 \mid t]\Big)\Big] \\ &= \mathbb{E}_{\mathrm{p}^{(j\neq k)}}\Big[G\Big(\Pr_{\mathrm{p}^{(k)}}[f(\mathrm{X}) = 2 \mid t]\Big)\Big] \\ &= \mathrm{Inf}_k^G[f, 2 \mid t]\,,\end{aligned}$$

which implies that the definition of impurity measure recovers the binary classification setting in Definition 2.

The binary consistency argument extends to this multiclass construction by applying the influence bound to the one-versus-rest indicator of each class and summing over the classes. Appendix E gives the complete argument, including the multiclass plug-in inequality, the histogram-estimation bound, and the tree-fitting bound. The resulting guarantee is stated below.

Theorem 21 *(Consistency rate for multiclass classification) Suppose that Assumptions 1–3 hold, where Assumption 1 is imposed on every class-probability function $\eta(\cdot, c)$, and let $\mathcal{Y} = \{1, \ldots, m\}$. Choose $G = \mathrm{Var}$ in the multiclass impurity. The expected excess error of the depth-K_n multiclass GridCART tree satisfies*

$$\mathbb{E}_{D_n}[R(T_{K_n})] - R^\star \leq \mathcal{O}\left(m\left(\frac{1}{K_n h_n^3} + h_n + \sqrt{\frac{1}{n h_n^d}}\right)\right).$$

Choosing $h_n = \Theta(n^{-1/(d+2)})$ and $K_n = \Omega(n^{4/(d+2)})$ gives the rate $\mathcal{O}(mn^{-1/(d+2)})$.

7.2 Regression

We consider regression with $\mathcal{Y} = [-M, M]$ for some $M > 0$. For a regression predictor g, define its squared-loss risk and the optimal squared-loss risk by

$$R_{\mathrm{reg}}(g) \triangleq \mathbb{E}[(Y - g(\mathrm{X}))^2] \quad \text{and} \quad R^\star_{\mathrm{reg}} \triangleq \inf_{g:\mathcal{X}\to\mathbb{R}} R_{\mathrm{reg}}(g),$$

respectively. Similarly, we first formulate the conditional Influence for regression tasks.

Definition 5 *(Conditional Influence for regression)* Let p and f be a product distribution and a mapping from $\mathcal{X}$ to $\mathcal{Y} = \mathbb{R}$, respectively. The conditional Influence of function f on variable $\mathrm{X}^{(k)}$ is defined by

$$\mathrm{Inf}_k[f \mid t] \triangleq \mathbb{E}_{\mathrm{p}^{(j\neq k)}}\Big[\mathrm{Var}_{\mathrm{p}^{(k)}}\big[f(\mathrm{X}) \mid \mathrm{X} \in t\big]\Big]\,,$$

where we choose variance as the impurity measure, which is widely used in regression tasks (Breiman et al., 1984).
For regression tasks, we replace the splitting criterion in Algorithm 3 as follows:

$$(k_0, s_0) \leftarrow \arg\max_{k,s} \ \Delta_{\text{reg}}^{\text{Inf}}(t, k, s) \,,$$

where

$$\Delta_{\text{reg}}^{\text{Inf}}(t, k, s) \triangleq \text{Inf}_k[f \mid t] - w_{t_L}\text{Inf}_k[f \mid t_L] - w_{t_R}\text{Inf}_k[f \mid t_R] \,.$$

For regression, the histogram method directly outputs $g_n^{\text{hist}}(\text{x}) \leftarrow \hat{\eta}(\mathcal{C})$ for $\text{x} \in \mathcal{C}$ instead of applying the thresholding step in Algorithm 2.

Theorem 22 *(Consistency rate for regression) Suppose that Assumptions 1-3 hold and $\mathcal{Y} = [-M, M]$ for some $M > 0$. The excess error of tree T_{K_n} generated by GridCART for regression has the following upper bound:*

$$\mathbb{E}_{D_n}\left[R_{\text{reg}}(T_{K_n})\right] - R_{\text{reg}}^{\star} \leq \mathcal{O}\left((L^2 + M^2)\Big(\frac{1}{K_n h_n^3} + h_n + \sqrt{\frac{1}{nh_n^d}}\Big)\right) .$$

Choosing $h_n = \Theta(n^{-1/(d+2)})$ and $K_n = \Omega(n^{4/(d+2)})$ gives the rate $\mathcal{O}((L^2 + M^2)n^{-1/(d+2)})$.

Theorem 22 shows that GridCART is consistent for regression with order $\mathcal{O}((L^2 + M^2)n^{-1/(d+2)})$. The quadratic dependence on M is natural because regression performance is measured by squared loss. Appendix E gives the complete proof.

8 Experiments

This section conducts experiments on both synthetic and real-world datasets. We first compare GridCART with conventional CART and examine the consistency rate of GridCART. We then evaluate the two ensemble extensions by comparing GridForest with random forests and GridGBDT with conventional gradient boosting decision trees on 20 OpenML datasets.

8.1 Experiment Setup

This subsection introduces the datasets used in our experiments and the algorithmic setup for GridCART and CART.

8.1.1 Datasets

The datasets comprise two categories.

Synthetic datasets: The synthetic experiments evaluate the approximation ability for different target functions, as shown in Table 4. Each synthetic dataset consists of 10, 000 samples with four-dimensional features drawn uniformly from $[0, 1]^4$. To model noisy labels, we independently flip each label with probability 0.05. Formally, we have

$$\Pr[Y = f(\mathrm{x}) \mid \mathrm{X} = \mathrm{x}] = 0.95, \Pr[Y = 1 - f(\mathrm{x}) \mid \mathrm{X} = \mathrm{x}] = 0.05\,,$$

where $f(\mathrm{x})$ is the target function.

Real-world datasets: We use 20 binary classification datasets from OpenML (Bischl et al., 2021; Vanschoren et al., 2013). These datasets and their characteristics are listed in Tables 6 and 7.

8.1.2 Algorithm Details

Histogram density estimation: The histogram density estimator and the histogram classifier can be calculated simultaneously for efficiency. As the assumption of a product distribution may not hold for real-world datasets, we use histogram density estimation directly for the joint distribution instead of multiplying the estimated marginals.

The choice of h_n **.** We choose h_n from the set $\{1/N \mid N \in \mathbb{N}^+\}$, which partitions the interval [0, 1] into N equal parts. Because the datasets are not large, choosing $N-1$, N, or $N+1$ may affect performance. Therefore, we choose the best value of N from $\{N-1, N, N+1\}$ by cross-validation. Table 5 reports the resulting comparison between GridCART and CART on the synthetic datasets.

8.2 Consistency Rates of GridCART

We plot the decreasing errors as sample sizes increase. For each pair of datasets and sample sizes, we repeat the experiment ten times. The curves of excess errors as well as our error bound about training sample sizes are shown in Figure 5. Note that the upper bound we present is relevant to unknown constants so we have to rescale the upper bound to fit the mean values for the comparison.

As shown in Figure 5, the blue lines represent the mean value and the blue regions are filled between the minimal and maximal values. The shape of the two curves in most of the subplots can nearly match, implying that the order of our upper bound is not so far away from the worst-case error in practice. For the Ball and XOR problems, the different curve shapes do not contradict the theoretical result, which is a worst-case upper bound rather than a pointwise prediction of empirical error. The gap indicates that the bound can be conservative for problems that are comparatively easy for GridCART.

8.3 Comparison with CART

We empirically compare the proposed GridCART with CART. We use the same 20 OpenML datasets, preprocessing, data splits, validation criterion, and paired random seeds as in the

Table 4 Formulation of synthetic datasets used in our experiments.

Target	Formulation
Sin	$f(\mathrm{x}) = \mathbb{I}\left\{\mathrm{x}^{(4)} \le \sin\left[5(\mathrm{x}^{(1)} + \mathrm{x}^{(2)} + \mathrm{x}^{(3)})\right]\right\}$
Ball	$f(\mathrm{x}) = \mathbb{I}\left\{(\mathrm{x}^{(1)})^2 + (\mathrm{x}^{(2)})^2 + (\mathrm{x}^{(3)})^2 + (\mathrm{x}^{(4)})^2 \le 1/2\right\}$
Ring	$f(\mathrm{x}) = \mathbb{I}\left\{1/3 \le (\mathrm{x}^{(1)})^2 + (\mathrm{x}^{(2)})^2 + (\mathrm{x}^{(3)})^2 + (\mathrm{x}^{(4)})^2 \le 2/3\right\}$
XOR	$f(\mathrm{x}) = \mathbb{I}\{\mathrm{x}^{(1)} < 0.5\} \oplus \mathbb{I}\{\mathrm{x}^{(2)} < 0.5\} \oplus \mathbb{I}\{\mathrm{x}^{(3)} < 0.5\} \oplus \mathbb{I}\{\mathrm{x}^{(4)} < 0.5\}$
Poly1	$f(\mathrm{x}) = \mathbb{I}\left\{4(\mathrm{x}^{(1)})^1 + 3(\mathrm{x}^{(2)})^2 + 2(\mathrm{x}^{(3)})^3 + (\mathrm{x}^{(4)})^4 \le 4\right\}$
Poly2	$f(\mathrm{x}) = \mathbb{I}\left\{(\mathrm{x}^{(1)})^4 + 2(\mathrm{x}^{(2)})^3 + 3(\mathrm{x}^{(3)})^2 + 4(\mathrm{x}^{(4)})^1 \le 4\right\}$

ensemble experiment below. CART and GridCART select the maximal tree depth from $\{5, 10, \log_2 n_{\text{train}}\}$, while GridCART additionally selects N from $\{4, 8, 16, 32\}$. Table 6 reports test macro-F1 using the same presentation format as the ensemble results.

Across the 20 datasets, GridCART obtains an average macro-F1 of .811, compared with .808 for CART. Its dataset-level mean is higher or lower on 13 and 7 datasets, respectively. Thus, GridCART remains competitive with CART while retaining its consistency guarantee under the stated assumptions. For real-world data, the product-distribution assumption may not hold, while the joint-histogram implementation remains empirically effective. This motivates us to relax the assumption for the analysis of our proposed GridCART in the future.

8.4 Comparison of Grid-based Ensembles

We next examine the empirical performance of the grid-based construction in bagging and boosting ensembles. Specifically, we compare GridForest with RF and GridGBDT with GBDT.

OpenML datasets: We conduct experiments on the same 20 binary classification datasets from OpenML (Bischl et al., 2021; Vanschoren et al., 2013). Table 7 reports the OpenML data identifier, the number of instances n, the number of input features d, and the ratio r between the minority- and majority-class sizes. The datasets vary in sample size, dimension, and class distribution.

Experiment setup: For each dataset, we impute missing numerical values with the median of the corresponding feature, treat missing categorical values as an additional category, and ordinally encode categorical features. We use stratified splits containing 60% training data, 20% validation data, and 20% test data. Hyperparameters and the decision threshold are selected by macro-F1 on the validation split, while the test split is used only for final evaluation. We use the same search space for the shared hyperparameters within each ensemble pair. RF and GridForest select the maximal tree depth from $\{5, 10, \log_2 n_{\text{train}}\}$ and the

Table 5 Comparison of GridCART with CART on synthetic datasets

Synthetic datasets	Test accuracy		Depth		Number of leaves		Time (seconds)	
	CART	GC	CART	GC	CART	GC	CART	GC
Sin	77.9	**83.3**	26.9	**12.3**	1253.7	**290.4**	72.2	**1.0**
	±1.2	±1.2	±2.9	±0.6	±33.3	±7.8	±18.7	±0.2
Ball	88.8	**92.8**	31.0	**10.9**	722.0	**100.0**	80.1	**0.5**
	±1.0	±0.7	±5.5	±0.7	±29.7	±13.0	±19.6	±0.1
Ring	87.0	**90.8**	26.9	**11.5**	808.7	**144.7**	72.5	**0.6**
	±1.0	±0.9	±2.4	±0.5	±27.5	±11.1	±15.7	±0.1
XOR	83.1	**94.8**	33.9	**10.4**	998.6	**49.6**	138.3	**0.7**
	±3.5	±0.7	±6.6	±0.7	±150.3	±11.8	±38.0	±0.2
Poly1	86.2	**88.8**	29.2	**11.3**	828.6	**143.7**	79.5	**0.7**
	±0.9	±1.1	±3.0	±0.7	±26.0	±11.6	±22.1	±0.2
Poly2	86.3	**88.9**	28.5	**11.6**	825.6	**142.7**	81.2	**0.7**
	±1.2	±1.1	±3.2	±0.7	±24.2	±10.1	±21.3	±0.2

Bold font indicates the algorithm outperforms the other one (higher accuracy, lower depth, smaller number of leaves, and less running time)

GridCART is abbreviated as GC due to space limitations.

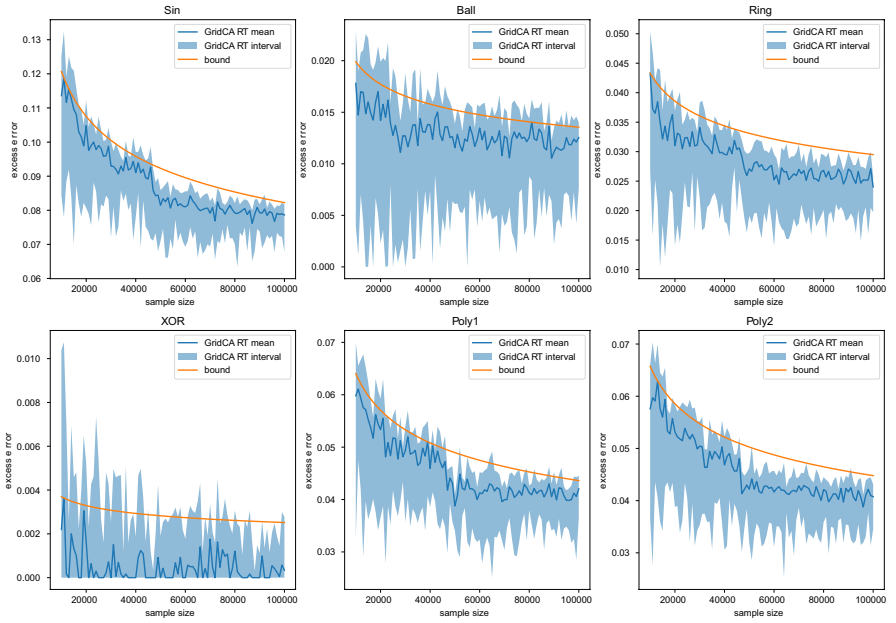


Fig. 5 Comparison of excess errors for GridCART trained with increasing training sample sizes and our error bounds. The blue curves are the mean empirical excess errors (test error minus the Bayes error), and the blue regions are filled between the minimal and the maximal values among multiple repeats. The orange curves are the error bound we presented in Theorem 8. The order of excess error nearly matches our presented upper bound in many cases

number of candidate features from $\{\sqrt{d}, d\}$. GBDT and GridGBDT select the learning rate from $\{0.1, 0.2, 0.3\}$ and the maximal tree depth from $\{3, 5, 10\}$. GridForest and GridGBDT additionally select N from $\{4, 8, 16, 32\}$. We repeat the complete procedure with ten random seeds and use the same split for all models within each seed.

We use macro-F1 to account for class imbalance and report the mean and standard deviation over the ten paired random seeds.

Across the 20 datasets, GridForest obtains an average macro-F1 of .833, compared with .831 for RF. Its dataset-level mean is higher or lower on 11 and 9 datasets, respectively. GridGBDT obtains an average macro-F1 of .831, compared with .823 for GBDT, and its corresponding counts are 13 and 7. The datasets range from 506 to 40,768 instances, from 3 to 41 features, and from .165 to .999 in minority-to-majority ratio. Overall, the grid-based ensembles show performance comparable to their conventional counterparts on this benchmark.

9 Limitations

The analysis assumes a product feature distribution because the definition of Influence requires independence among the feature coordinates. Consequently, the guarantees do not cover dependent features or the joint-histogram implementation used in the experiments on real-world datasets.

Table 6 Test macro-F1 of CART and GridCART on the 20 OpenML datasets

Dataset (OpenML data ID)	n	d	r	CART	GridCART
2D planes (727)	40,768	11	.996	.930 ± .002	.932 ± .002
Ailerons (734)	13,750	41	.736	.854 ± .004	.856 ± .009
Arsenic female bladder (949)	559	5	.167	.623 ± .071	.555 ± .051
Banana (1460)	5,300	3	.813	.884 ± .005	.892 ± .008
Banknote authentication (1462)	1,372	5	.801	.977 ± .009	.981 ± .013
Boston (853)	506	14	.704	.809 ± .047	.829 ± .032
Breast W (15)	699	10	.526	.926 ± .014	.926 ± .029
Churn (46915)	5,000	20	.165	.860 ± .013	.864 ± .014
COMPAS two years (42192)	5,278	14	.888	.662 ± .013	.669 ± .016
Credit approval (46535)	690	16	.802	.834 ± .020	.867 ± .014
Diabetes (37)	768	9	.536	.687 ± .033	.687 ± .028
Glioma grading (46846)	839	24	.723	.855 ± .022	.855 ± .010
KDD El Nino (839)	782	9	.539	.921 ± .029	.890 ± .031
Mammographic mass (45557)	961	5	.862	.779 ± .035	.792 ± .028
PBCSeq (802)	1,945	19	.999	.675 ± .035	.720 ± .018
Quake (772)	2,178	4	.801	.520 ± .020	.519 ± .014
RMFTSA sleep (741)	1,024	3	.988	.703 ± .040	.718 ± .032
Stock (841)	950	10	.947	.942 ± .015	.932 ± .012
Tour and Travels churn (45545)	954	7	.307	.817 ± .014	.817 ± .029
WDBC (1510)	569	31	.594	.905 ± .024	.922 ± .025

Values are means ± standard deviations over ten paired random seeds. Boldface indicates the higher unrounded mean within each method pair

The data identifier is shown in parentheses

The quantity r is the ratio between the minority- and majority-class sizes

The GridForest analysis establishes worst-case consistency rates but does not capture the empirical benefit of tree diversity, whose theoretical characterization remains open.

The GridGBDT guarantee considers a common unshifted grid. With random shifts, the current iterate is no longer constant on the new grid cells, so the fixed-target residual identity and the resulting contraction argument no longer apply.

10 Conclusions

Our AISTATS 2023 work identified the zero-purity-gain obstacle in the consistency analysis of CART and introduced GridCART, for which it established a consistency rate of order $\mathcal{O}(n^{-1/(d+2)})$ without assuming an additive target function. This journal extension develops Grid-based Random Forests (GridForest) and Grid-based Gradient Boosting Decision Tree (GridGBDT). It establishes a consistency rate of order $\mathcal{O}(n^{-1/(d+2)})$ for GridForest and for a theoretically tractable specialization of GridGBDT. Under an additional margin condition and a feature density bounded away from zero, GridForest attains the faster rate $\mathcal{O}(n^{-(\alpha+1)/(d+2)})$. On the 20 OpenML datasets, the grid-based ensembles achieve average

Table 7 Test macro-F1 of conventional and grid-based ensembles on the 20 OpenML datasets

Dataset (OpenML data ID)	n	d	r	RF	GridForest	GBDT	GridGBDT
2D planes (727)	40,768	11	.996	.930 ± .001	.931 ± .002	.932 ± .003	.931 ± .002
Ailerons (734)	13,750	41	.736	.875 ± .006	.870 ± .006	.878 ± .006	.878 ± .005
Arsenic female bladder (949)	559	5	.167	.669 ± .078	.624 ± .059	.625 ± .045	.638 ± .073
Banana (1460)	5,300	3	.813	.893 ± .006	.905 ± .005	.896 ± .006	.905 ± .006
Banknote authentication (1462)	1,372	5	.801	.990 ± .006	1.000 ± .000	.982 ± .005	.993 ± .009
Boston (853)	506	14	.704	.869 ± .026	.871 ± .018	.831 ± .028	.865 ± .028
Breast W (15)	699	10	.526	.953 ± .016	.960 ± .017	.938 ± .021	.942 ± .014
Churn (46915)	5,000	20	.165	.906 ± .014	.905 ± .012	.894 ± .016	.903 ± .013
COMPAS two years (42192)	5,278	14	.888	.673 ± .015	.670 ± .017	.676 ± .012	.669 ± .019
Credit approval (46535)	690	16	.802	.877 ± .028	.874 ± .023	.851 ± .029	.869 ± .020
Diabetes (37)	768	9	.536	.714 ± .039	.711 ± .040	.711 ± .039	.716 ± .048
Glioma grading (46846)	839	24	.723	.858 ± .021	.860 ± .015	.854 ± .018	.852 ± .020
KDD El Nino (839)	782	9	.539	.927 ± .021	.923 ± .017	.916 ± .029	.936 ± .026
Mammographic mass (45557)	961	5	.862	.787 ± .028	.797 ± .027	.785 ± .023	.787 ± .030
PBCSeq (802)	1,945	19	.999	.757 ± .027	.782 ± .016	.775 ± .018	.760 ± .021
Quake (772)	2,178	4	.801	.512 ± .021	.523 ± .025	.526 ± .028	.522 ± .015
RMFTSA sleep (741)	1,024	3	.988	.723 ± .027	.737 ± .022	.707 ± .029	.734 ± .027
Stock (841)	950	10	.947	.953 ± .015	.952 ± .018	.937 ± .011	.956 ± .010
Tour and Travels churn (45545)	954	7	.307	.820 ± .019	.835 ± .020	.831 ± .022	.826 ± .026
WDBC (1510)	569	31	.594	.939 ± .018	.939 ± .021	.915 ± .031	.945 ± .013

Values are means ± standard deviations over ten paired random seeds. Boldface indicates the higher unrounded mean within each ensemble pair

The data identifier is shown in parentheses

The quantity r is the ratio between the minority- and majority-class sizes

macro-F1 values comparable to their conventional counterparts. Additionally, we extended GridCART to multiclass classification and regression and provided consistency rates of order $\mathcal{O}(mn^{-1/(d+2)})$ and $\mathcal{O}((L^2+M^2)n^{-1/(d+2)})$, respectively. Taken together, these results demonstrate that the proposed grid-based approach provides a principled path toward narrowing the gap between the empirical success of widely used tree-based methods and their theoretical understanding.

Appendix A: Proof of Proposition 2 and Proposition 3

Here, we will prove the two propositions in Section 4. We begin with the strict definition of the impurity function, which was not mentioned before due to space limitations.

Definition 6 *(Impurity function)* An impurity function $G : [0,1] \to [0,1]$ is a function that measures the impurity given the ratio of positive samples or negative samples, which satisfies the following three properties:

1. (Normalized) $G(0) = G(1) = 0$ and $G(1/2) = 1$, which means that if all the samples are in a positive class or negative class, then there is no impurity. If the numbers of the two classes are the same, then the maximal impurity is reached.
2. (Symmetric) $G(x) = G(1-x)$, as we only care about the ratio of the positive or negative no matter which of them.
3. (Strongly concave) There exists an $\alpha > 0$ such that for all $\theta \in [0,1], x_1, x_2 \in [0,1]$ we have

$$G\big(\theta x_1 + (1-\theta)x_2\big) - \theta\, G(x_1) - (1-\theta)\, G(x_2) \geq \alpha\theta(1-\theta)\,(x_1 - x_2)^2\,,$$

which is useful to lower bound the purity gain.

Proof of Proposition 2

Proposition 2 There exists a probability density p and a conditional probability function $\eta : \mathcal{X} \to [0,1]$, such that $\Delta_K = 0$ for any possible cut in every leaf.

Proof We give a counterexample, an "XOR"-like problem in real-valued space, which can be proved with $\Delta_K = 0$ for all possible split, even when the error is not zero. In the counterexample, the features X uniformly distributes in $[0,1]^2$ and the target function is defined as follows:

$$f(\mathrm{x}^{(1)}, \mathrm{x}^{(2)}) = \begin{cases} 0, & \mathbb{I}\{\mathrm{x}^{(1)} < 0.5\} \neq \mathbb{I}\{\mathrm{x}^{(2)} < 0.5\}\,, \\ 1, & \text{otherwise}\,, \end{cases}$$

or equivalently the exclusive OR of $\mathrm{x}^{(1)} > 0.5$ and $\mathrm{x}^{(2)} > 0.5$. We let the conditional probability function $\eta(\mathrm{x})$ exactly be f, i.e., the label is with probability one the target function $f(\mathrm{x})$ for all x. Then we can prove that there is no purity gain for CART, when $K = 1$, no matter which dimension k and which cut point s we choose.

$$\begin{aligned} \Delta_K &= C_K - C_{K+1} \\ &= \sum_{t \in \text{leaves}(T_K)} p_t \mathcal{G}(t) - \sum_{t \in \text{leaves}(T_{K+1})} p_t \mathcal{G}(t) \\ &= \mathcal{G}\big([0,1]^2\big) - s\, \mathcal{G}\big(t_L\big) - (1-s)\, \mathcal{G}\big(t_R\big) \\ &= G\big(\mathbb{E}[f \mid \mathrm{X} \in [0,1]^2]\big) - s\, G\big(\mathbb{E}\big[f \mid \mathrm{X} \in t_L\big]\big) - (1-s)\, G\big(\mathbb{E}\big[f \mid \mathrm{X} \in t_R\big]\big) \\ &= G(1/2) - s\, G(1/2) - (1-s)\, G(1/2) \\ &= 0\,, \end{aligned} \tag{A1}$$

where we write $\{\mathrm{x} \mid \mathrm{x}^{(k)} < s\}$ and $\{\mathrm{x} \mid \mathrm{x}^{(k)} \geq s\}$ as t_L and t_R, respectively. (A1) is because both the positive and the negative classes have the same ratio, no matter which cutpoint we choose. □

Proof of Proposition 3

proposition 3 *For any product distribution* p *on* $\mathcal{X}$ *and any target function* $\hat{f} : \mathcal{X} \to \mathcal{Y}$, *InfCART gets no purity gain if and only if it reaches zero error.*

Proof The sufficiency is also trivial so we will only show the proof of necessity. Let the maximal purity gain of InfCART in every leaf be 0, by the strong concavity of the impurity function, we have

$$\begin{aligned} 0 &= \max_{k,s} \ \mathrm{Inf}_k\left[f \mid [0,1]^2\right] - s\,\mathrm{Inf}_k\left[f \mid t_L\right] - (1-s)\,\mathrm{Inf}_k\left[f \mid t_R\right] \\ &= \max_{k,s} \ \mathbb{E}_{\mathrm{p}^{(j\neq k)}}\left[G\left(\mathbb{E}_{\mathrm{p}^{(k)}}\left[f \mid \mathrm{X} \in t\right]\right)\right] - w_{t_L}\mathbb{E}_{\mathrm{p}^{(j\neq k)}}\left[G\left(\mathbb{E}_{\mathrm{p}^{(k)}}\left[f \mid \mathrm{X} \in t_L\right]\right)\right] \\ &\qquad - w_{t_R}\mathbb{E}_{\mathrm{p}^{(j\neq k)}}\left[G\left(\mathbb{E}_{\mathrm{p}^{(k)}}\left[f \mid \mathrm{X} \in t_R\right]\right)\right] \qquad \text{(A2)} \\ &\geq \max_{k,s} \ \mathbb{E}_{\mathrm{p}^{(j\neq k)}}\left[\alpha w_{t_L} w_{t_R}\left(\mathbb{E}_{\mathrm{p}^{(k)}}\left[f \mid \mathrm{X} \in t_L\right] - \mathbb{E}_{\mathrm{p}^{(k)}}\left[f \mid \mathrm{X} \in t_R\right]\right)^2\right] \\ &\geq 0\,, \end{aligned}$$

where (A2) is by the strong concavity in Definition 6. Thus, almost surely for all $k \in [d]$ and $\mathrm{x}_1, \mathrm{x}_2 \in \mathcal{X}$, once $\mathrm{x}_1^{(j\neq k)} = \mathrm{x}_2^{(j\neq k)}$, then one has $f(\mathrm{x}_1) = f(\mathrm{x}_2)$, implying that f is constant in every leaf. Therefore, we have the error $\hat{R}(T) = 0$. □

Appendix B: Proof of Lemma 5

We first show the proof of Lemma 4, which is the basis of Lemma 5.

Proof of Lemma 4

Lemma 4 *Suppose that* $\mathbb{L}_1$*-error of estimators* $\hat{\mathrm{p}}_{\mathrm{X}^{(k)}}$ *has upper bound* $\mathbb{E}\left[\|\hat{\mathrm{p}}_{\mathrm{X}^{(k)}} - \mathrm{p}_{\mathrm{X}^{(k)}}\|_1\right] \leq e_k$ *and Assumption 3 holds, then the following holds:*

$$\mathbb{E}\left[\left\|\prod_{k=1}^{d}\hat{\mathrm{p}}_{\mathrm{X}^{(k)}} - \mathrm{p}_{\mathrm{X}}\right\|_1\right] \leq \sum_{k=1}^{d} e_k\,.$$

Proof Suppose that we have $\left\|\hat{\mathrm{p}}_{\mathrm{X}^{(k)}} - \mathrm{p}_{\mathrm{X}^{(k)}}\right\|_1 \leq e_k$ for all k, then we have

$$
\begin{aligned}
\left\|\prod_{k=1}^{d}\hat{\mathrm{p}}_{\mathrm{X}^{(k)}}-\mathrm{p}_{\mathrm{X}}\right\|_1 &= \left\|\prod_{k=1}^{d}\hat{\mathrm{p}}_{\mathrm{X}^{(k)}}-\prod_{k=1}^{d}\mathrm{p}_{\mathrm{X}^{(k)}}\right\|_1 \\
&\le \left\|\prod_{k=1}^{d}\hat{\mathrm{p}}_{\mathrm{X}^{(k)}}-\mathrm{p}_{\mathrm{X}^{(1)}}\prod_{k=2}^{d}\hat{\mathrm{p}}_{\mathrm{X}^{(k)}}\right\|_1+\left\|\mathrm{p}_{\mathrm{X}^{(1)}}\prod_{k=2}^{d}\hat{\mathrm{p}}_{\mathrm{X}^{(k)}}-\prod_{k=1}^{d}\mathrm{p}_{\mathrm{X}^{(k)}}\right\|_1 \\
&= \left\|(\hat{\mathrm{p}}_{\mathrm{X}^{(1)}}-\mathrm{p}_{\mathrm{X}^{(1)}})\prod_{k=2}^{d}\hat{\mathrm{p}}_{\mathrm{X}^{(k)}}\right\|_1+\left\|\mathrm{p}_{\mathrm{X}^{(1)}}(\prod_{k=2}^{d}\hat{\mathrm{p}}_{\mathrm{X}^{(k)}}-\prod_{k=2}^{d}\mathrm{p}_{\mathrm{X}^{(k)}})\right\|_1 \\
&\le \left\|(\hat{\mathrm{p}}_{\mathrm{X}^{(1)}}-\mathrm{p}_{\mathrm{X}^{(1)}})\prod_{k=2}^{d}\hat{\mathrm{p}}_{\mathrm{X}^{(k)}}\right\|_1+\left\|\mathrm{p}_{\mathrm{X}^{(1)}}(\hat{\mathrm{p}}_{\mathrm{X}^{(2)}}-\mathrm{p}_{\mathrm{X}^{(2)}})\prod_{k=3}^{d}\hat{\mathrm{p}}_{\mathrm{X}^{(k)}}\right\|_1 \\
&\quad+\left\|\mathrm{p}_{\mathrm{X}^{(1)}}\mathrm{p}_{\mathrm{X}^{(2)}}(\prod_{k=3}^{d}\hat{\mathrm{p}}_{\mathrm{X}^{(k)}}-\prod_{k=3}^{d}\mathrm{p}_{\mathrm{X}^{(k)}})\right\|_1 \\
&\le \ldots \\
&\le \sum_{k=1}^{d}\left\|\prod_{i=1}^{k-1}\mathrm{p}_{\mathrm{X}^{(i)}}(\hat{\mathrm{p}}_{\mathrm{X}^{(k)}}-\mathrm{p}_{\mathrm{X}^{(k)}})\prod_{i=k+1}^{d}\hat{\mathrm{p}}_{\mathrm{X}^{(i)}}\right\|_1 .
\end{aligned}
\tag{B3}
$$

By the normalization of probability density, we have

$$
\begin{aligned}
&\left\|\prod_{i=1}^{k-1}\mathrm{p}_{\mathrm{X}^{(i)}}(\hat{\mathrm{p}}_{\mathrm{X}^{(k)}}-\mathrm{p}_{\mathrm{X}^{(k)}})\prod_{i=k+1}^{d}\hat{\mathrm{p}}_{\mathrm{X}^{(i)}}\right\|_1 \\
&\quad= \int\left|\prod_{i=1}^{k-1}\mathrm{p}_{\mathrm{X}^{(i)}}(\hat{\mathrm{p}}_{\mathrm{X}^{(k)}}-\mathrm{p}_{\mathrm{X}^{(k)}})\prod_{i=k+1}^{d}\hat{\mathrm{p}}_{\mathrm{X}^{(i)}}\right|\mathrm{dx}^{(1)}\mathrm{dx}^{(2)}\ldots\mathrm{dx}^{(d)} \\
&\quad= \left(\prod_{i=1}^{k-1}\int\mathrm{p}_{\mathrm{X}^{(i)}}\mathrm{dx}^{(i)}\right)\int\left|\hat{\mathrm{p}}_{\mathrm{X}^{(k)}}-\mathrm{p}_{\mathrm{X}^{(k)}}\right|\mathrm{dx}^{(k)}\left(\prod_{i=k+1}^{d}\int\hat{\mathrm{p}}_{\mathrm{X}^{(i)}}\mathrm{dx}^{(i)}\right) \\
&\quad\le 1\cdot e_k\cdot 1 \\
&\quad\le e_k .
\end{aligned}
\tag{B4}
$$

Combining (B3) and (B4), we then obtain the following bound

$$
\left\|\prod_{k=1}^{d}\hat{\mathrm{p}}_{\mathrm{X}^{(k)}}-\mathrm{p}_{\mathrm{X}}\right\|_1 \le \sum_{k=1}^{d}e_k ,
$$

which completes the proof. □

Proof of Lemma 5

With Lemma 4 in hand, it suffices to prove that $e_k \le \mathcal{O}\left(h_n + 1/\sqrt{nh_n}\right)$. We begin with Lemma 23.

Lemma 23 *(Wasserman (2006) Theorem 6.11) Suppose that* p' *is absolutely continuous and that* $\int(\mathrm{p}'(u))^2\mathrm{d}u < \infty$. *Then*

$$\mathbb{E}\left[\int \left[\hat{\mathrm{p}}(u) - \mathrm{p}(u)\right]^2 \mathrm{d}u\right] \leq \mathcal{O}\left(h_n^2 + \frac{1}{nh}\right).$$

Provided that $h_n = \Theta(n^{-1/3})$, *we have the optimal error bound*

$$\mathbb{E}\left[\int \left[\hat{\mathrm{p}}(u) - \mathrm{p}(u)\right]^2 \mathrm{d}u\right] \leq \mathcal{O}\left(n^{-2/3}\right).$$

We will then use Lemma 4 and Lemma 23 to prove Lemma 5.

Lemma 5 Under *Assumptions 2 and 3*, *the* $\mathbb{L}_1$*-error of histogram density estimator can be upper bounded by*

$$\mathbb{E}\left[\left\|\prod_{k=1}^{d}\hat{\mathrm{p}}_{\mathrm{X}^{(k)}} - \mathrm{p}_{\mathrm{X}}\right\|_1\right] \leq \mathcal{O}\left(d\left(h_n + \sqrt{\frac{1}{nh_n}}\right)\right).$$

Proof Taking the estimators into Lemma 23 we can bound the error at each dimension by

$$\mathbb{E}_{D_n}\left\|\hat{\mathrm{p}}_{\mathrm{X}^{(k)}} - \mathrm{p}_{\mathrm{X}^{(k)}}\right\|_2^2 \leq \mathcal{O}\left(h_n^2 + \frac{1}{nh_n}\right).$$

By Hölder's inequality $\|f \cdot 1\|_1 \leq \|f\|_2\|1\|_2$, and the fact that $\sqrt{a^2+b^2} \leq |a| + |b|$, we have

$$\mathbb{E}_{D_n}\left\|\hat{\mathrm{p}}_{\mathrm{X}^{(k)}} - \mathrm{p}_{\mathrm{X}^{(k)}}\right\|_1 \leq \mathcal{O}\left(h_n + \sqrt{\frac{1}{nh_n}}\right),$$

which completes the proof. □

Appendix C: Proof of Lemma 6

Lemma 24 *(Devroye et al. (1997)[Corollary 6.1]) If the classifier is defined by whether* $\hat{\eta}_n(\mathrm{x}) \leq 1/2$, *or equivalently as*

$$g_n(\mathrm{x}) = \begin{cases} 0, & \text{if } \hat{\eta}_n(\mathrm{x}) \leq 1/2\,, \\ 1, & \text{otherwise}\,. \end{cases}$$

Then the excess error of classifier g_n *satisfies*

$$R(g_n) - R^\star \le 2 \int |\eta(\mathrm{x}) - \hat{\eta}_n(\mathrm{x})|\, \mathrm{p}(\mathrm{x})\mathrm{dx}\,.$$

We will then prove Lemma 6 by using Lemma 24.

Lemma 6 *Under Assumption 1, suppose that the histogram rule satisfies* $h_n \to 0$ *and* $nh_n^d \to \infty$ *as* $n \to \infty$. *For any* $\mathrm{p}, \eta, n > 0$ *we have*

$$\mathbb{E}\left[R(g_n^{\mathrm{hist}})\right] - R^\star \le \mathcal{O}\left(h_n + \sqrt{\frac{1}{nh_n^d}}\right).$$

Exclusively provided that $h_n = \Theta(n^{-1/(d+2)})$, *we obtain a consistency rate of order* $\mathcal{O}(n^{-1/(d+2)})$ *for histogram classifier* g_n^{hist}.

Proof Then it suffices to provide the error bound of $|\eta(\mathrm{x}) - \hat{\eta}_n(\mathrm{x})|$. Our proof is based on Devroye et al. (1997)[Theorem 9.4], which provided only the consistency of histogram classifiers, whereas we here showed the consistency rate. Recall the definition of estimated conditional probability as follows:

$$\hat{\eta}_n(\mathrm{x}) \triangleq \frac{\sum_{i=1}^n Y_i \mathbb{I}\{\mathrm{X}_i \in \mathcal{C}(\mathrm{x})\}}{\sum_{i=1}^n \mathbb{I}\{\mathrm{X}_i \in \mathcal{C}(\mathrm{x})\}}\,.$$

By some simple calculation, we have

$$\begin{aligned}
\mathbb{E}_{D_n}\left[\hat{\eta}_n(\mathrm{x})\right] &= \mathbb{E}_{\mathrm{X}_1,\ldots,\mathrm{X}_n} \mathbb{E}_{Y_1,\ldots,Y_n|\mathrm{X}_1,\ldots,\mathrm{X}_n}\left[\hat{\eta}_n(\mathrm{x})\right] \\
&= \mathbb{E}_{\mathrm{X}_1,\ldots,\mathrm{X}_n} \frac{\sum_{i=1}^n \Pr\left[Y = 1 \mid \mathrm{X} \in \mathcal{C}(\mathrm{x})\right] \mathbb{I}\left[\mathrm{x}_i \in \mathcal{C}(\mathrm{x})\right]}{\sum_{i=1}^n \mathbb{I}\{\mathrm{x}_i \in \mathcal{C}(\mathrm{x})\}} \\
&= \Pr\left[Y = 1 \mid \mathrm{X} \in \mathcal{C}(\mathrm{x})\right] \\
&= \frac{1}{\Pr\left[\mathrm{X} \in \mathcal{C}(\mathrm{x})\right]} \int_{\mathcal{C}(\mathrm{x})} \eta(\mathrm{x})\mathrm{p}(\mathrm{x})\mathrm{dx}\,,
\end{aligned}$$

which means that the expectation of $\hat{\eta}_n(\mathrm{x})$ is a piecewise constant function. For some $\mathcal{C}$ that contains no samples we let $\hat{\eta}_n(\mathrm{x}) \triangleq 0$ for completeness. We then split $\int |\eta(\mathrm{x}) - \hat{\eta}_n(\mathrm{x})|\mathrm{p}(\mathrm{x})\mathrm{dx}$ into two terms

$$\begin{aligned}
&\int |\eta(\mathrm{x}) - \hat{\eta}_n(\mathrm{x})|\mathrm{p}(\mathrm{x})\mathrm{dx} \\
&\le \underbrace{\int \left|\eta(\mathrm{x}) - \mathbb{E}_{D_n}\left[\hat{\eta}_n(\mathrm{x})\right]\right|\mathrm{p}(\mathrm{x})\mathrm{dx}}_{\text{term A}} + \underbrace{\int \mathbb{E}_{D_n}\left[\left|\mathbb{E}_{D_n}\left[\hat{\eta}_n(\mathrm{x})\right] - \hat{\eta}_n(\mathrm{x})\right|\right]\mathrm{p}(\mathrm{x})\mathrm{dx}}_{\text{term B}}\,.
\end{aligned}$$

We bound term A as follows:

$$
\begin{aligned}
&\int \left|\eta(\mathrm{x}) - \mathbb{E}_{D_n}\left[\hat{\eta}_n(\mathrm{x})\right]\right| \mathrm{p}(\mathrm{x})\mathrm{dx} \\
&= \sum_{j\in\mathcal{J}} \int_{\mathcal{C}_j} \left|\eta(\mathrm{x}) - \frac{1}{\Pr\left[\mathrm{X}\in\mathcal{C}_j\right]} \int_{\mathcal{C}_j} \eta(\mathrm{y})\mathrm{p}(\mathrm{y})\mathrm{dy}\right| \mathrm{p}(\mathrm{x})\mathrm{dx} \\
&= \sum_{j\in\mathcal{J}} \frac{1}{\Pr\left[\mathrm{X}\in\mathcal{C}_j\right]} \int_{\mathcal{C}_j} \left|\eta(\mathrm{x}) \int_{\mathcal{C}_j} \mathrm{p}(\mathrm{y})\mathrm{dy} - \int_{\mathcal{C}_j} \eta(\mathrm{y})\mathrm{p}(\mathrm{y})\mathrm{dy}\right| \mathrm{p}(\mathrm{x})\mathrm{dx} \\
&\le \sum_{j\in\mathcal{J}} \frac{1}{\Pr\left[\mathrm{X}\in\mathcal{C}_j\right]} \int_{\mathcal{C}_j}\int_{\mathcal{C}_j} |\eta(\mathrm{x}) - \eta(\mathrm{y})|\, \mathrm{p}(\mathrm{x})\mathrm{p}(\mathrm{y})\mathrm{dxdy} \\
&\le \sum_{j\in\mathcal{J}} \frac{1}{\Pr\left[\mathrm{X}\in\mathcal{C}_j\right]} \int_{\mathcal{C}_j}\int_{\mathcal{C}_j} L\mathrm{diam}(\mathcal{C}_j)\mathrm{p}(\mathrm{x})\mathrm{p}(\mathrm{y})\mathrm{dxdy}\,,
\end{aligned}
$$

where $\mathrm{diam}(\mathcal{C}_j) \triangleq \max_{\mathrm{x},\mathrm{x}'\in\mathcal{C}_j} \|\mathrm{x}-\mathrm{x}'\|_2$, and we use the fact that $\Pr\left[\mathrm{X}\in\mathcal{C}_j\right] = \int_{\mathcal{C}_j} \mathrm{p}(\mathrm{x})\mathrm{dx}$ and Lipschitz condition in Assumption 1. It is not difficult to prove that $\mathrm{diam}(\mathcal{C}_j) = h_n\sqrt{d}$. Consequently, we have

$$
\begin{aligned}
&\sum_{j\in\mathcal{J}} \frac{1}{\Pr\left[\mathrm{X}\in\mathcal{C}_j\right]} \int_{\mathcal{C}_j}\int_{\mathcal{C}_j} L\mathrm{diam}(\mathcal{C}_j)\mathrm{p}(\mathrm{x})\mathrm{p}(\mathrm{y})\mathrm{dxdy} \\
&\qquad = \sum_{j\in\mathcal{J}} \frac{1}{\Pr\left[\mathrm{X}\in\mathcal{C}_j\right]} Lh_n\sqrt{d}\Pr\left[\mathrm{X}\in\mathcal{C}_j\right]^2 \\
&\qquad = Lh_n\sqrt{d}\sum_{j\in\mathcal{J}} \Pr\left[\mathrm{X}\in\mathcal{C}_j\right] \\
&\qquad = Lh_n\sqrt{d}\,,
\end{aligned}
\tag{C5}
$$

Then we bound term B. At each cube $\mathcal{C}_j$ both $\hat{\eta}_n(\mathrm{x})$ and $\mathbb{E}_{D_n}\left[\hat{\eta}_n(\mathrm{x})\right]$ are constant function. Let $\nu_n(\mathcal{C}) = \frac{1}{n}\sum_{i=1}^n Y_i \mathbb{I}\{\mathrm{X}_i\in\mathcal{C}\}$. Then by Jensen's inequality, we have

$$
\begin{aligned}
\int \mathbb{E}_{D_n}\left[\left|\mathbb{E}_{D_n}\left[\hat{\eta}_n(\mathrm{x})\right] - \hat{\eta}_n(\mathrm{x})\right|\right]\mathrm{p}(\mathrm{x})\mathrm{dx} &= \mathbb{E}_{D_n}\left[\int \left|\mathbb{E}_{D_n}\left[\hat{\eta}_n(\mathrm{x})\right] - \hat{\eta}_n(\mathrm{x})\right|\mathrm{p}(\mathrm{x})\mathrm{dx}\right] \\
&= \sum_{j\in\mathcal{J}} \mathbb{E}_{D_n}\left[\left|\mathbb{E}_{D_n}\left[\nu_n(\mathcal{C}_j)\right] - \nu_n(\mathcal{C}_j)\right|\right] \\
&\le \sum_{j\in\mathcal{J}} \sqrt{\mathbb{E}_{D_n}\left[\left|\mathbb{E}_{D_n}\left[\nu_n(\mathcal{C}_j)\right] - \nu_n(\mathcal{C}_j)\right|^2\right]}\,.
\end{aligned}
$$

Note that $\mathbb{E}_{D_n}\left[\left|\mathbb{E}_{D_n}\left[\nu_n(\mathcal{C}_j)\right] - \nu_n(\mathcal{C}_j)\right|^2\right]$ equals to the variance of $\nu_n(\mathcal{C}_j)$. It can be trivially bounded by $\Pr[\mathrm{X}\in\mathcal{C}_j]$ and here we use Jensen's inequality again, and consequently we have

$$\begin{aligned}\int \mathbb{E}_{D_n}\left[\left|\mathbb{E}_{D_n}[\hat{\eta}_n(\mathrm{x})]-\hat{\eta}_n(\mathrm{x})\right|\right]\mathrm{p}(\mathrm{x})\mathrm{dx} &\le \sum_{j\in\mathcal{J}}\sqrt{\mathbb{E}_{D_n}\left[\left|\mathbb{E}_{D_n}[\nu_n(\mathcal{C}_j)]-\nu_n(\mathcal{C}_j)\right|^2\right]}\\ &\le \sum_{j\in\mathcal{J}}\sqrt{\frac{\Pr[\mathrm{X}\in\mathcal{C}_j]}{n}}\\ &\le |\mathcal{J}|\frac{1}{|\mathcal{J}|}\sum_{j\in\mathcal{J}}\sqrt{\frac{\Pr[\mathrm{X}\in\mathcal{C}_j]}{n}} \qquad \text{(C6)}\\ &\le |\mathcal{J}|\sqrt{\frac{1}{|\mathcal{J}|}\sum_{j\in\mathcal{J}}\frac{\Pr[\mathrm{X}\in\mathcal{C}_j]}{n}}\\ &= \sqrt{\frac{|\mathcal{J}|}{n}}.\end{aligned}$$

As $\mathcal{X}=[0,1]^d$, the cardinality of $\mathcal{J}$, is exactly $1/h_n^d$. Taking $|\mathcal{J}|=1/h_n^d$ into (C6) we have

$$\int \mathbb{E}_{D_n}\left[\left|\mathbb{E}_{D_n}[\hat{\eta}_n(\mathrm{x})]-\hat{\eta}_n(\mathrm{x})\right|\right]\mathrm{p}(\mathrm{x})\mathrm{dx} \le \sqrt{\frac{1}{nh_n^d}}\,. \qquad \text{(C7)}$$

Combining (C5), (C7) and Lemma 24, we obtain the following bound of the excess error

$$R(\mathrm{g}_n^{\mathrm{hist}})-R^\star \le 2\left(Lh_n\sqrt{d}+\sqrt{\frac{1}{nh_n^d}}\right).$$

This completes the proof. □

Appendix D: Proof of Theorem 7

We begin with the following lemmas, which will be used in the proof.

Lemma 25 *Let* $S_i \triangleq |\sum_{j=1}^{i} a_j|$ *we have* $\sum_{i=1}^{n} S_i \ge \frac{1}{2}\sum_{i=1}^{n}|a_i|$.

Proof It is equivalent to proving that

$$2\sum_{i=1}^{n} S_i \ge \sum_{i=1}^{n}|a_i|\,.$$

We first combine $|a_1+a_2+\cdots+a_k|$ and $|a_1+a_2+\cdots+a_k+a_{k+1}|$ together for all k as follows:

$$
\begin{aligned}
2\sum_{i=1}^{n} S_i &= \sum_{i=1}^{n} S_i + \sum_{i=1}^{n} S_i \\
&= |a_1| \\
&\quad + |a_1| + |a_1 + a_2| \\
&\quad + |a_1 + a_2| + |a_1 + a_2 + a_3| \\
&\quad + \cdots \\
&\quad + |a_1 + a_2 + \cdots + a_{n-1}| + |a_1 + a_2 + \cdots + a_{n-1} + a_n| \\
&\quad + |a_1 + a_2 + \cdots + a_{n-1} + a_n| \,.
\end{aligned}
$$

For every pair of $|a_1 + a_2 + \cdots + a_k| + |a_1 + a_2 + \cdots + a_k + a_{k+1}|$, we use the fact that $|a| + |a + b| \geq |b|$. Setting $a = a_1 + a_2 + \cdots + a_k$ and $b = a_{k+1}$, we have the following inequality holds.

$$
\begin{aligned}
2\sum_{i=1}^{n} S_i &\geq |a_1| + |a_2| + \cdots + |a_n| + |a_1 + a_2 + \cdots + a_{n-1} + a_n| \\
&\geq |a_1| + |a_2| + \cdots + |a_n| \,.
\end{aligned}
$$

This completes the proof. □

Lemma 26 *Let $C > 0$ be a constant such that for all $n \geq 1$, the conditions $a_1 \leq C$ and $a_n - a_{n+1} \geq \frac{a_n^2}{C}$ hold. Then, it follows that:*

$$
a_n \leq \frac{C}{n} \quad \forall n \geq 1 \,.
$$

Proof We prove this by applying mathematical induction on n

1. $n = 1$ holds as is given in the condition.
2. Suppose that the inequality holds for $n = k$. We then think about the case when $n = k + 1$
   (a) If $a_k \leq \frac{C}{k+1}$, then $a_{k+1} \leq a_k \leq \frac{C}{k+1}$ as $\{a_n\}_{n=1}^{\infty}$ is decreasing
   (b) Otherwise $\frac{C}{k} \geq a_k > \frac{C}{k+1}$, then we have

$$
a_{k+1} \leq a_k \left(1 - \frac{a_k}{C}\right) \leq \frac{C}{k}\left(1 - \frac{1}{k+1}\right) \leq \frac{C}{k+1} \,.
$$

This completes the proof. □

Proof of Theorem 7

We prove Theorem 7 by the following two steps. Firstly, we bound the purity gain of any 1-dimension toy problem. Then, we show that the d-dimension problem can be reduced to many of them, as a result of which, we can bound the potential after the tree grows to depth K.

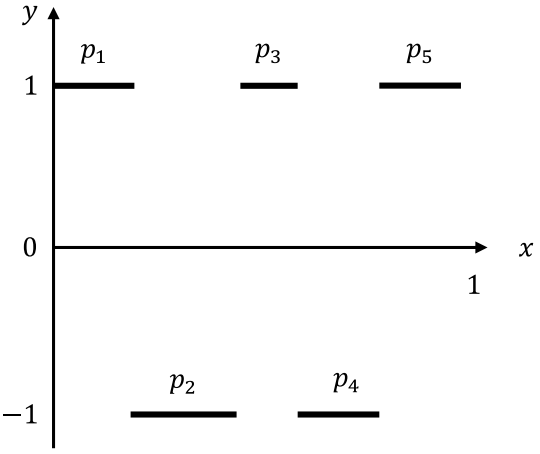


Fig. 6 Target function of the 1-dimension toy problem

Step I: Analysis of the 1-dimension Case

We initiate our discussion with the 1-dimensional scenario, depicted in Figure 6. Herein, we define the one-dimensional feature X as following a uniform distribution $U[0, 1]$. Specifically, $0 < p_i < 1$ for all $1 \le i \le N$, and $\sum_{i=1}^{N} p_i = 1$. Let P_i denote the cumulative sum of the first i probabilities, defined as $P_i = \sum_{j=1}^{i} p_j$. We then define the target function f as:

$$f(x) = \sum_{i=1}^{N} \mathbb{I}\{P_{i-1} \le x < P_i\}(-1)^{i-1} + \mathbb{I}\{x = P_N\}(-1)^{N-1},$$

which ranges from $\{-1, 1\}$. Note that we consider $\mathcal{Y} = \{-1, 1\}$ for simplicity, and it is not difficult to extend the result to $\mathcal{Y} = \{0, 1\}$. For $1 \le i \le N$, define the signed cumulative sum $Q_i = \sum_{j=1}^{i} (-1)^{j-1} p_j$. We also write

$$\mu = \mathbb{E}[f], \qquad \mu_L(s) = \mathbb{E}[f \mid 0 \le X < s], \qquad \mu_R(s) = \mathbb{E}[f \mid s \le X \le 1].$$

Let the purity gain at the split point s be

$$\Delta_G(s) \triangleq G(\mu) - sG(\mu_L(s)) - (1 - s)G(\mu_R(s)).$$

We can then lower bound the purity gain by

$$\begin{aligned} \max_{0<s<1} \Delta_G(s) &\ge \max_{0<s<1} \alpha s(1-s)\big(\mu_L(s) - \mu_R(s)\big)^2 \\ &\ge \max_{1\le k<N} P_k(1-P_k)\left(\frac{Q_k}{P_k} - \frac{Q_N - Q_k}{1-P_k}\right)^2. \end{aligned} \tag{D8}$$

The second inequality follows by restricting the candidate split points to $s = P_k$ for $1 \le k < N$, and $\alpha = 1$ for the Gini index. We will then bound (D8) in the following lemma.

Lemma 27 *The maximum of optimization problem (D8) has a lower bound*

$$\max_{1\le k<N} P_k(1-P_k)\left(\frac{Q_k}{P_k}-\frac{Q_N-Q_k}{1-P_k}\right)^2 \ge \frac{\mathrm{Var}^2[f]}{(N-1)^2}\,.$$

Proof Using the definitions of P_k and Q_k, we can rewrite (D8) as

$$\begin{aligned}\max_{1\le k<N} P_k(1-P_k)\left(\frac{Q_k}{P_k}-\frac{Q_N-Q_k}{1-P_k}\right)^2 &= \max_{1\le k<N}\frac{(Q_k-Q_NP_k)^2}{P_k(1-P_k)}\\ &\ge \frac{\sum_{k=1}^{N-1}(Q_k-Q_NP_k)^2}{\sum_{k=1}^{N-1}P_k(1-P_k)}\,.\end{aligned}$$

The inequality is because the maximum is always greater than the weighted average. Note that $\mathbb{E}[f]=\sum_{i=1}^{N}(-1)^{i-1}p_i$ so we can rewrite the numerator

$$\begin{aligned}Q_k-Q_NP_k &= \sum_{i=1}^{k}(-1)^{i-1}p_i-\left(\sum_{i=1}^{N}(-1)^{i-1}p_i\right)\sum_{i=1}^{k}p_i\\ &= \sum_{i=1}^{k}p_i\big[(-1)^{i-1}-\mathbb{E}[f]\big]\,.\end{aligned} \tag{D9}$$

For all $0\le x\le 1$ we have $x(1-x)\le 1/4$, consequently we have

$$\sum_{k=1}^{N-1}P_k(1-P_k)\le\frac{N-1}{4}\,. \tag{D10}$$

Combining (D9) and (D10), we have

$$\frac{\sum_{k=1}^{N-1}(Q_k-Q_NP_k)^2}{\sum_{k=1}^{N-1}P_k(1-P_k)}\ge\frac{4\sum_{k=1}^{N-1}\left[\sum_{i=1}^{k}p_i\left[(-1)^{i-1}-\mathbb{E}[f]\right]\right]^2}{N-1}\,.$$

Using Cauchy-Schwarz inequality or the fact that $\sum_{i=1}^{n}a_i^2\ge\frac{1}{n}\big(\sum_{i=1}^{n}|a_i|\big)^2$ we have

$$\frac{4\sum_{k=1}^{N-1}\left[\sum_{i=1}^{k}p_i\left[(-1)^{i-1}-\mathbb{E}[f]\right]\right]^2}{N-1}\ge\frac{4}{(N-1)^2}\left(\sum_{k=1}^{N-1}\left|\sum_{i=1}^{k}p_i[(-1)^{i-1}-\mathbb{E}[f]]\right|\right)^2\,.$$

Using Lemma 25 as well as Jensen's inequality we have

$$\frac{4}{(N-1)^2}\left(\sum_{k=1}^{N-1}\left|\sum_{i=1}^{k} p_i\left[(-1)^{i-1}-\mathbb{E}[f]\right]\right|\right)^2$$
$$=\frac{4}{(N-1)^2}\left(\sum_{k=1}^{N}\left|\sum_{i=1}^{k} p_i\left[(-1)^{i-1}-\mathbb{E}[f]\right]\right|\right)^2 \geq \frac{\left(\sum_{i=1}^{N}\left|p_i\left[(-1)^{i-1}-\mathbb{E}[f]\right]\right|\right)^2}{(N-1)^2}.$$

Note that the equality is because $\sum_{i=1}^{N} p_i\left[(-1)^{i-1}-\mathbb{E}[f]\right]=0$. By using the fact that $\left(\sum_{i=1}^{N}|a_i|\right)^2 \geq \left(\sum_{i=1}^{N} a_i\right)^2$, we can finally obtain the following lower bound.

$$\begin{aligned}
\frac{\left(\sum_{i=1}^{N}\left|p_i\left[(-1)^{i-1}-\mathbb{E}[f]\right]\right|\right)^2}{(N-1)^2} &= \frac{\left(\sum_{i=1}^{N}\left|(-1)^{i-1}p_i\left[1-(-1)^{i-1}\mathbb{E}[f]\right]\right|\right)^2}{(N-1)^2} \\
&= \frac{\left(\sum_{i=1}^{N}\left|p_i\left[1-(-1)^{i-1}\mathbb{E}[f]\right]\right|\right)^2}{(N-1)^2} \\
&\geq \frac{\left(\sum_{i=1}^{N} p_i\left[1-(-1)^{i-1}\mathbb{E}[f]\right]\right)^2}{(N-1)^2} \\
&= \frac{\left(1-\mathbb{E}[f]^2\right)^2}{(N-1)^2} \\
&= \frac{\mathrm{Var}^2[f]}{(N-1)^2}.
\end{aligned}$$

The proof is complete. □

Step II: Reduction from *d* Dimensions to One Dimension

To reduce the general d-dimension cases to 1-dimension cases, we begin with the following conclusion.

Lemma 28 *(Keller (2011)[Proposition 2]) Let $h:[0,1]\to[0,1]$ be a concave function satisfying $h(t)\geq Ent(t)$ for all $0\leq t\leq 1$. There exists a constant $B'>0$ such that for any function $f:[0,1]^d\to\{0,1\}$ with $\mathbb{E}[f]=p$ we have*

$$p(1-p)\leq B'\sum_{k=1}^{d}\frac{\mathrm{Inf}_k^h[f]}{\log\frac{4}{3\mathrm{Inf}_k^h[f]}},$$

where $\mathrm{Inf}_k^h[f]=\mathbb{E}_{\mathrm{p}^{(j\neq k)}}\left[h\left(\mathbb{E}_{\mathrm{p}^{(k)}}[f]\right)\right]$ follows the definition in Definition 1. Then provided that $h(x)=4x(1-x)$ and $B=4B'/\log(4/3)$, we have

$$p(1-p)\leq B\sum_{k=1}^{d}\mathrm{Inf}_k^h[f].$$

With Lemma 27 and Lemma 28 at hand, we start proving Theorem 7.

Theorem 7 *Suppose that the target function g is grid-aligned N-piece and that the candidate cuts in Algorithm 3 include the corresponding cut-point sets. Then, under Assumptions 1-3, for the tree T generated by Algorithm 3 with depth K, we have*

$$\mathbb{E}\Big[\mathbb{I}\{T(\mathrm{X}) \neq g_n^{\mathrm{hist}}(\mathrm{X})\}\Big] \leq \mathcal{O}\left(N^3/K\right) .$$

Proof Algorithm 3 learns a tree to fit the histogram classifier g_n^{hist} in Algorithm 2 under the estimated distribution $\hat{\mathrm{p}}$. Let $\epsilon(T)$ be the error of tree T fitting the histogram classifier g_n^{hist}, i.e.,

$$\epsilon(T) = \mathbb{E}_{\mathrm{X}\sim\hat{\mathrm{p}}}\Big[\mathbb{I}\{T(\mathrm{X}) \neq g_n^{\mathrm{hist}}(\mathrm{X})\}\Big] .$$

Note that any x falling into leaf t has the same label, so the error is the minimum of $\Pr_{\mathrm{X}\sim\hat{\mathrm{p}}}[g_n^{\mathrm{hist}}(\mathrm{X}) = 1]$ and $\Pr_{\mathrm{X}\sim\hat{\mathrm{p}}}[g_n^{\mathrm{hist}}(\mathrm{X}) = 0]$. Let $p = \Pr_{\mathrm{X}\sim\hat{\mathrm{p}}}[g_n^{\mathrm{hist}}(\mathrm{X}) = 1]$ and $h \equiv \mathrm{Var}$, by Lemma 28, we have

$$\epsilon(T) = \min(p, 1-p) \leq 2p(1-p) \leq 2B \sum_{k=1}^{d} \mathrm{Inf}_k^{\mathrm{Var}}[g_n^{\mathrm{hist}}] , \tag{D11}$$

which implies that the error can be bounded by the impurity measure except for a constant. Note that results in Lemma 28 and (D11), which take expectations over $\mathrm{X} \in \mathcal{X}$, are easy to be extended to $\mathrm{X} \in t$. We define the error at leaf t as follows:

$$\epsilon(T \mid t) = \min\Big(\Pr\left[g_n^{\mathrm{hist}}(\mathrm{X}) = 1 \mid \mathrm{X} \in t\right], \Pr\left[g_n^{\mathrm{hist}}(\mathrm{X}) = 0 \mid \mathrm{X} \in t\right]\Big) .$$

Similarly, we recall the definition of conditional Influence, which has been introduced in Definition 2.

$$\mathrm{Inf}_k^{\mathrm{Var}}[f \mid t] = \mathbb{E}_{\mathrm{p}^{(j\neq k)}}\Big[\mathrm{Var}\left[\mathbb{E}_{\mathrm{p}^{(k)}}[f \mid \mathrm{X} \in t]\right]\Big] .$$

Then, it still holds that the error is bounded by Influence, i.e.,

$$\epsilon(T \mid t) \leq 2B \sum_{k=1}^{d} \mathrm{Inf}_k^{\mathrm{Var}}[f \mid t] .$$

We recall the definition of the potential of the learned tree T_K width depth K as follows:

$$\begin{aligned} C_K &\triangleq \sum_{t\in T_K} \Pr_{\hat{\mathrm{p}}}[\mathrm{X} \in t]\ \mathcal{G}(t) \\ &= \sum_{t\in T_K} \Pr_{\hat{\mathrm{p}}}[\mathrm{X} \in t]\ \frac{1}{d} \sum_{k=1}^{d} \mathrm{Inf}_k^{\mathrm{Var}}[f \mid t] , \end{aligned} \tag{D12}$$

where $\mathcal{G}(t) = \frac{1}{d}\sum_{k=1}^{d} \mathrm{Inf}_k^{\mathrm{Var}}[f \mid t]$ is the impurity at leaf t. By Lemma 28 we have

$$\begin{aligned} \epsilon(T_K) &= \sum_{t \in \mathrm{leaves}(T_K)} \Pr_{\hat{\mathrm{p}}}[\mathrm{X} \in t]\, \epsilon(T \mid t) \\ &\le 2BdC_K \,. \end{aligned} \tag{D13}$$

Then to control the potential C_K is sufficient to bound the error $\epsilon(T_K)$. We first show that the potential gain verifies

$$\Delta_K = C_K - C_{K+1} \ge \frac{4C_K^2}{(N-1)^3} \,.$$

Every leaf $t \in \mathrm{leaves}(T_K)$ is split into two new leaves t_L and t_R. We write its coordinate-wise conditional purity gain as

$$\delta(t,k,s) \triangleq \mathrm{Inf}_k^{\mathrm{Var}}[g_n^{\mathrm{hist}} \mid t] - w_{t_L}\mathrm{Inf}_k^{\mathrm{Var}}[g_n^{\mathrm{hist}} \mid t_L] - w_{t_R}\mathrm{Inf}_k^{\mathrm{Var}}[g_n^{\mathrm{hist}} \mid t_R] \,.$$

Thus, the previously defined Influence gain satisfies $\Delta^{\mathrm{Inf}}(t,k,s) = p_t\delta(t,k,s)$. The two children can be written as $t_L = t \cap \{\mathrm{x} \mid \mathrm{x}^{(k)} < s\}$ and $t_R = t \cap \{\mathrm{x} \mid \mathrm{x}^{(k)} \ge s\}$. At step K, the potential gain is the weighted sum of maximal conditional purity gains over all leaves.

$$C_K - C_{K+1} = \sum_{t \in \mathrm{leaves}(T_K)} p_t \max_{k,s} \delta(t,k,s) \,.$$

For any leaf t, we choose a dimension k and cut point s to maximize $\delta(t,k,s)$, which gives

$$\begin{aligned} \max_{k,s} \delta(t,k,s) &\ge \frac{1}{d}\sum_{k=1}^{d} \max_{s} \delta(t,k,s) \\ &\ge \frac{1}{d}\sum_{k=1}^{d} \frac{1}{|\mathcal{S}_k|} \sum_{s \in \mathcal{S}_k} \delta(t,k,s) \,, \end{aligned}$$

where we frequently use the fact that $\max_{a \in \mathcal{A}} f(a) \ge \frac{1}{|\mathcal{A}|}\sum_{a \in \mathcal{A}} f(a)$, *i.e.*, the maximum is no lower than the mean. As the tree grows to fit the histogram classifier, we have $\mathcal{S}_k = \{h_n, 2h_n, \ldots, (1/h_n - 1)h_n\}$, which implies that $|\mathcal{S}_k| = 1/h_n - 1$. For compactness, define

$$\begin{aligned} \mu_{k,L}(s) &\triangleq \mathbb{E}_{\hat{\mathrm{p}}^{(k)}}[g_n^{\mathrm{hist}} \mid \mathrm{X} \in t_L], \\ \mu_{k,R}(s) &\triangleq \mathbb{E}_{\hat{\mathrm{p}}^{(k)}}[g_n^{\mathrm{hist}} \mid \mathrm{X} \in t_R], \\ \delta_{\mathrm{sc}}(t,k,s) &\triangleq \mathbb{E}_{\hat{\mathrm{p}}^{(j \ne k)}}\left[w_{t_L}w_{t_R}\big(\mu_{k,L}(s) - \mu_{k,R}(s)\big)^2\right] \,. \end{aligned}$$

The strong concavity of variance gives

$$\delta(t,k,s) \ge \delta_{\mathrm{sc}}(t,k,s) \,.$$

Here, $\mathcal{S}_k$ is the set of all possible cuts for leaf t along dimension k. As the sum is no smaller than its largest summand, we have

$$\sum_{s\in\mathcal{S}_k} \delta(t,k,s) \geq \max_{s\in\mathcal{S}_k} \delta_{\text{sc}}(t,k,s)\,,$$

which has the same form as (D8) and can be bounded using Lemma 27. To apply the lemma, let $z = 2g_n^{\text{hist}} - 1$. The conditional mean difference for z is twice that for g_n^{hist}, and $\operatorname{Var}[z \mid \mathrm{X} \in t] = 4\operatorname{Var}[g_n^{\text{hist}} \mid \mathrm{X} \in t]$. Consequently, we have

$$\begin{aligned}\max_{k,s} \delta(t,k,s) &\geq \frac{1}{d(N-1)} \sum_{k=1}^{d} \max_{s\in\mathcal{S}_k} \delta_{\text{sc}}(t,k,s) \\ &\geq \frac{4}{d(N-1)^3} \sum_{k=1}^{d} \mathbb{E}_{\hat{\mathrm{p}}^{(j\neq k)}} \left[\operatorname{Var}^2_{\hat{\mathrm{p}}^{(k)}}[g_n^{\text{hist}} \mid \mathrm{X} \in t] \right] .\end{aligned}$$

By Jensen's inequality and the definition of impurity measure $\mathcal{G}$, we have

$$\begin{aligned}&\frac{4}{N-1}\frac{1}{d}\sum_{k=1}^{d} \mathbb{E}_{\hat{\mathrm{p}}^{(j\neq k)}} \left[\frac{\operatorname{Var}^2_{\hat{\mathrm{p}}^{(k)}}[g_n^{\text{hist}} \mid \mathrm{X} \in t]}{(N-1)^2} \right] \\ &\quad= \frac{4}{(N-1)^3}\frac{1}{d}\sum_{k=1}^{d} \mathbb{E}_{\hat{\mathrm{p}}^{(j\neq k)}} \left[\operatorname{Var}^2_{\hat{\mathrm{p}}^{(k)}}[g_n^{\text{hist}} \mid \mathrm{X} \in t] \right] \\ &\quad\geq \frac{4}{(N-1)^3} \left(\frac{1}{d}\sum_{k=1}^{d} \mathbb{E}_{\hat{\mathrm{p}}^{(j\neq k)}} \left[\operatorname{Var}_{\hat{\mathrm{p}}^{(k)}}[g_n^{\text{hist}} \mid \mathrm{X} \in t] \right] \right)^2 \\ &\quad= \frac{4}{(N-1)^3} \mathcal{G}(t)^2 .\end{aligned}$$

Then the potential gain $C_K - C_{K+1}$ at step K is the weighted average of purity gains over all the leaves:

$$\begin{aligned}C_K - C_{K+1} &= \sum_{t\in\text{leaves}(T_K)} p_t \max_{k,s} \delta(t,k,s) \\ &\geq \frac{4}{(N-1)^3} \sum_{t\in\text{leaves}(T_K)} p_t \mathcal{G}(t)^2 \\ &\geq \frac{4}{(N-1)^3} \left(\sum_{t\in\text{leaves}(T_K)} p_t \mathcal{G}(t) \right)^2 \\ &= \frac{4}{(N-1)^3} C_K^2 .\end{aligned} \tag{D14}$$

By Lemma 26 and (D14) we can upper bound the potential C_K

$$C_K \leq \frac{(N-1)^3}{4K} . \tag{D15}$$

Finally, by combining the upper bounds of the error (D13) and the potential (D15), we have

$$\epsilon(T_K) \leq \frac{Bd(N-1)^3}{2K} = \mathcal{O}(N^3/K)\,,$$

which completes the proof. □

Appendix E: Beyond Binary Classification

This section proves the consistency guarantees for multiclass classification and regression.

Consistency Rate for Multi-class Classification

Let $\mathcal{Y} = \{1, \ldots, m\}$, let $\eta(\mathrm{x}, c) = \Pr[Y = c \mid \mathrm{X} = \mathrm{x}]$, and let

$$f^\star(\mathrm{x}) \in \arg\max_{c \in \mathcal{Y}} \eta(\mathrm{x}, c)$$

be a Bayes classifier. The Bayes error is $R^\star = R(f^\star)$. We define the estimated class probabilities and the histogram classifier by

$$\hat{\eta}_n(\mathrm{x}, c) \triangleq \frac{\sum_{i=1}^n \mathbb{I}\{Y_i = c, \mathrm{X}_i \in \mathcal{C}(\mathrm{x})\}}{\sum_{i=1}^n \mathbb{I}\{\mathrm{X}_i \in \mathcal{C}(\mathrm{x})\}} \quad \text{and} \quad \mathrm{g}_n^{\mathrm{hist}}(\mathrm{x}) \in \arg\max_{c \in \mathcal{Y}} \hat{\eta}_n(\mathrm{x}, c)\,. \tag{E16}$$

A fixed tie-breaking rule is used, and an arbitrary probability vector is assigned to an empty cell.

Corollary 29 *(Extension of Devroye et al. (1997)[Corollary 6.1]) If* $\mathrm{g}_n(\mathrm{x}) \in \arg\max_{c \in \mathcal{Y}} \hat{\eta}_n(\mathrm{x}, c)$ *, then*

$$R(\mathrm{g}_n) - R^\star \leq \sum_{c \in \mathcal{Y}} \int |\eta(\mathrm{x}, c) - \hat{\eta}_n(\mathrm{x}, c)|\, \mathrm{p}(\mathrm{x}) \mathrm{dx}\,.$$

Proof Fix x, write $c^\star = f^\star(\mathrm{x})$ and $\hat{c} = \mathrm{g}_n(\mathrm{x})$, and use $\hat{\eta}_n(\mathrm{x}, \hat{c}) \geq \hat{\eta}_n(\mathrm{x}, c^\star)$. The conditional excess error satisfies

$$\begin{aligned} \eta(\mathrm{x}, c^\star) - \eta(\mathrm{x}, \hat{c}) &\leq |\eta(\mathrm{x}, c^\star) - \hat{\eta}_n(\mathrm{x}, c^\star)| + |\hat{\eta}_n(\mathrm{x}, \hat{c}) - \eta(\mathrm{x}, \hat{c})| \\ &\leq \sum_{c \in \mathcal{Y}} |\eta(\mathrm{x}, c) - \hat{\eta}_n(\mathrm{x}, c)|\,. \end{aligned}$$

Integration with respect to p proves the claim. □

Corollary 30 *Suppose that* $\mathcal{Y} = \{1, 2, \ldots, m\}$*. Suppose that every* $\eta(\cdot, c)$ *is L-Lipschitz and that* $h_n \to 0$ *and* $nh_n^d \to \infty$*. The histogram classifier in (E16) satisfies*

$$\mathbb{E}_{D_n}\left[R(g_n^{\text{hist}})\right] - R^\star \le 2m\left(Lh_n\sqrt{d} + \sqrt{\frac{1}{nh_n^d}}\right).$$

In particular, $h_n = \Theta(n^{-1/(d+2)})$ *gives the rate* $\mathcal{O}(mn^{-1/(d+2)})$.

Proof For each $c \in \mathcal{Y}$, apply the argument for terms A and B in the proof of Lemma 6 to the binary variables $Z_{i,c} = \mathbb{I}\{Y_i = c\}$. The conditional mean of $Z_{i,c}$ is $\eta(\mathrm{x}, c)$, so (C5) and (C7) give

$$\mathbb{E}_{D_n}\int |\eta(\mathrm{x}, c) - \hat{\eta}_n(\mathrm{x}, c)|\, \mathrm{p}(\mathrm{x})\mathrm{dx} \le 2\left(Lh_n\sqrt{d} + \sqrt{\frac{1}{nh_n^d}}\right).$$

Summing this inequality over c and applying Corollary 29 proves the result. □

Lemma 31 *(Multiclass tree-fitting bound) Let* $g : \mathcal{X} \to \{1, \dots, m\}$ *be an N-piece function. A depth-K tree built with* $G = \mathrm{Var}$ *in the multiclass impurity of Section 7 satisfies*

$$\Pr_{\hat{\mathrm{p}}}[T(\mathrm{X}) \ne g(\mathrm{X})] \le \mathcal{O}\left(\frac{mN^3}{K}\right).$$

Proof Define $g_c(\mathrm{x}) = \mathbb{I}\{g(\mathrm{x}) = c\}$ and

$$I_{k,c}(t) = \mathrm{Inf}_k^{\mathrm{Var}}[g_c \mid t], \qquad \mathcal{G}(t) = \frac{1}{dm}\sum_{k=1}^{d}\sum_{c=1}^{m} I_{k,c}(t).$$

If $q_c(t) = \Pr_{\hat{\mathrm{p}}}[g(\mathrm{X}) = c \mid \mathrm{X} \in t]$, the conditional majority-vote error obeys

$$\begin{aligned} 1 - \max_c q_c(t) &\le \sum_{c=1}^{m} q_c(t)(1 - q_c(t)) \\ &\le B_0 \sum_{c=1}^{m}\sum_{k=1}^{d} I_{k,c}(t) = B_0 dm\, \mathcal{G}(t), \end{aligned} \tag{E17}$$

where the second inequality applies Lemma 28 conditionally on t to every binary function g_c. The constant B_0 is independent of N, K, and m.

Let $C_K = \sum_{t \in \mathrm{leaves}(T_K)} \Pr_{\hat{\mathrm{p}}}[\mathrm{X} \in t]\mathcal{G}(t)$. For a cut s on coordinate k, let $\delta_{k,c}(t, s)$ be the decrease in $I_{k,c}(t)$. Every classwise decrease is nonnegative by concavity of variance, and hence

$$\Delta_{\mathrm{mc}}(t, k, s) \ge \frac{1}{dm}\sum_{c=1}^{m} \delta_{k,c}(t, s).$$

For every pair (k, c), g_c is an N-piece binary function. The one-dimensional calculation in Lemma 27, in the form used to obtain (D14), gives

$$\sum_{s \in \mathcal{S}_k} \delta_{k,c}(t,s) \geq \frac{4}{(N-1)^2} I_{k,c}(t)^2.$$

Averaging the multiclass gain over all $d(N-1)$ coordinate-cut pairs and applying Jensen's inequality now gives

$$\begin{aligned}
\max_{k,s} \Delta_{\mathrm{mc}}(t,k,s) &\geq \frac{1}{d(N-1)} \sum_{k=1}^{d} \sum_{s \in \mathcal{S}_k} \Delta_{\mathrm{mc}}(t,k,s) \\
&\geq \frac{4}{d^2 m (N-1)^3} \sum_{k=1}^{d} \sum_{c=1}^{m} I_{k,c}(t)^2 \\
&\geq \frac{4}{d(N-1)^3} \mathcal{G}(t)^2.
\end{aligned}$$

Consequently,

$$\begin{aligned}
C_K - C_{K+1} &\geq \frac{4}{d(N-1)^3} \sum_{t} \Pr_{\hat{\mathrm{p}}}[\mathrm{X} \in t] \mathcal{G}(t)^2 \\
&\geq \frac{4}{d(N-1)^3} C_K^2.
\end{aligned}$$

Lemma 26 gives $C_K = \mathcal{O}(N^3/K)$. Averaging (E17) over the leaves then gives the stated $\mathcal{O}(mN^3/K)$ fitting error. □

Proof of Theorem 21 For any two classifiers, the difference of their risks is at most their disagreement probability. Therefore,

$$R(T_{K_n}) - R(\mathrm{g}_n^{\mathrm{hist}}) \leq \Pr_{\hat{\mathrm{p}}}[T_{K_n}(\mathrm{X}) \neq \mathrm{g}_n^{\mathrm{hist}}(\mathrm{X})] + \|\hat{\mathrm{p}} - \mathrm{p}\|_1.$$

The histogram classifier is an N-piece function with $N = 1/h_n$. Lemma 31, Lemma 5, and Corollary 30 thus give

$$\mathbb{E}_{D_n}[R(T_{K_n})] - R^\star \leq \mathcal{O}\left(\frac{m}{K_n h_n^3} + d\left(h_n + \sqrt{\frac{1}{n h_n}} \right) + m\left(h_n + \sqrt{\frac{1}{n h_n^d}} \right) \right).$$

Since $m \geq 1$, d is fixed, and $h_n \leq 1$, the middle term is absorbed by the last term. Substitution of the choices stated in the theorem completes the proof. □

Consistency Rate for Regression

Let $\eta(\mathrm{x}) = \mathbb{E}[Y \mid \mathrm{X} = \mathrm{x}]$ and $R(h) = \mathbb{E}[(Y - h(\mathrm{X}))^2]$. The histogram regressor is

$$\hat{\eta}_n(\mathrm{x}) \triangleq \frac{\sum_{i=1}^{n} Y_i \, \mathbb{I}\{\mathrm{X}_i \in \mathcal{C}(\mathrm{x})\}}{\sum_{i=1}^{n} \mathbb{I}\{\mathrm{X}_i \in \mathcal{C}(\mathrm{x})\}},$$

with an arbitrary value in $[-M, M]$ on an empty cell.

Lemma 32 *(Histogram regression bound) If $|Y| \le M$ and η is L-Lipschitz, then*

$$\mathbb{E}_{D_n} \int (\hat{\eta}_n(\mathrm{x}) - \eta(\mathrm{x}))^2 \, \mathrm{p}(\mathrm{x}) \mathrm{d}\mathrm{x} \le \mathcal{O}\left(L^2 d h_n^2 + \frac{M^2}{n h_n^d}\right).$$

Proof For a grid cell $\mathcal{C}$, let

$$\bar{\eta}_{\mathcal{C}} = \mathbb{E}[Y \mid \mathrm{X} \in \mathcal{C}].$$

The Lipschitz assumption and $\mathrm{diam}(\mathcal{C}) = h_n\sqrt{d}$ imply

$$|\eta(\mathrm{x}) - \bar{\eta}_{\mathcal{C}}| \le L h_n \sqrt{d} \qquad \text{for every } \mathrm{x} \in \mathcal{C}.$$

Hence the integrated squared bias is at most $L^2 d h_n^2$. Conditional on the number $N_{\mathcal{C}}$ of observations in $\mathcal{C}$, the sample mean has variance at most $M^2/(N_{\mathcal{C}} \vee 1)$. The standard binomial bound

$$\mathbb{E}\left[\frac{\mathbb{I}\{N_{\mathcal{C}} > 0\}}{N_{\mathcal{C}}}\right] \le \frac{2}{(n+1)\Pr[\mathrm{X} \in \mathcal{C}]}$$

together with the empty-cell contribution gives

$$\sum_{\mathcal{C}} \Pr[\mathrm{X} \in \mathcal{C}] \mathbb{E}_{D_n}\left[(\hat{\eta}_n(\mathcal{C}) - \bar{\eta}_{\mathcal{C}})^2\right] \le \mathcal{O}\left(\frac{M^2 |\mathcal{J}|}{n}\right).$$

Since $|\mathcal{J}| = h_n^{-d}$, the bias–variance decomposition proves the claim. □

Proof of Theorem 22 The square-loss excess-risk identity gives

$$R_{\mathrm{reg}}(T_{K_n}) - R^{\star}_{\mathrm{reg}} = \int (T_{K_n}(\mathrm{x}) - \eta(\mathrm{x}))^2 \mathrm{p}(\mathrm{x}) \mathrm{d}\mathrm{x}.$$

Insert $\hat{\eta}_n$ and use $(a+b)^2 \le 2a^2 + 2b^2$. Both T_{K_n} and $\hat{\eta}_n$ take values in $[-M, M]$, so replacing p by $\hat{\mathrm{p}}$ in the tree-fitting term costs at most $4M^2 \|\hat{\mathrm{p}} - \mathrm{p}\|_1$. It follows that

$$\begin{aligned} R_{\mathrm{reg}}(T_{K_n}) - R^{\star}_{\mathrm{reg}} \le{}& 2\mathbb{E}_{\hat{\mathrm{p}}}[(T_{K_n}(\mathrm{X}) - \hat{\eta}_n(\mathrm{X}))^2] + 8M^2 \|\hat{\mathrm{p}} - \mathrm{p}\|_1 \\ &+ 2 \int (\hat{\eta}_n(\mathrm{x}) - \eta(\mathrm{x}))^2 \mathrm{p}(\mathrm{x}) \mathrm{d}\mathrm{x}. \end{aligned}$$

The target $\hat{\eta}_n$ is an N-piece function with $N = 1/h_n$. Applying Theorem 16, Lemma 5, and Lemma 32, and then taking expectation over D_n, gives

$$\mathbb{E}_{D_n}[R_{\text{reg}}(T_{K_n})] - R^\star_{\text{reg}} \le \mathcal{O}\left(\frac{M^2}{K_n h_n^3} + M^2 d\left(h_n + \sqrt{\frac{1}{nh_n}}\right) + L^2 d h_n^2 + \frac{M^2}{nh_n^d}\right).$$

For $h_n \le 1$ and $nh_n^d \ge 1$, we have $h_n^2 \le h_n$, $(nh_n^d)^{-1} \le (nh_n^d)^{-1/2}$, and $(nh_n)^{-1/2} \le (nh_n^d)^{-1/2}$. The displayed bound therefore implies the bound in Theorem 22. Substituting the stated h_n and K_n proves the rate. □

Appendix F: Proofs of the GridForest Randomization Lemmas

Proof of Lemma 11 Use the potential C_K defined in (D12). For a leaf t, define the best gain along coordinate k by

$$\Delta_k^\star(t) \triangleq \max_s \delta(t, k, s),$$

where $\delta(t, k, s)$ is the conditional purity gain used in Appendix D. Conditional on the tree constructed through depth K, choose

$$k_t^\star \in \arg\max_{k \in [d]} \Delta_k^\star(t).$$

Since $\mathcal{S}_t$ is sampled uniformly from all subsets of $[d]$ with size m,

$$\Pr(k_t^\star \in \mathcal{S}_t) = \frac{m}{d}.$$

Every candidate purity gain is nonnegative. Therefore,

$$\begin{aligned}\mathbb{E}_{\mathcal{S}_t}\left[\max_{k \in \mathcal{S}_t} \Delta_k^\star(t)\right] &\ge \Pr(k_t^\star \in \mathcal{S}_t)\Delta_{k_t^\star}^\star(t) \\ &= \frac{m}{d}\max_{k \in [d]} \Delta_k^\star(t).\end{aligned} \tag{F18}$$

Let $\mathcal{F}_K$ contain the training data and all random choices made through depth K. Applying (F18) independently at every leaf and then using the full-feature potential decrease in (D14) gives

$$\mathbb{E}[C_K - C_{K+1} \mid \mathcal{F}_K] \ge \frac{m}{d}\frac{4}{(N-1)^3} C_K^2.$$

Write $u_K = \mathbb{E}[C_K]$. The law of total expectation and Jensen's inequality yield

$$u_{K+1} \le u_K - \frac{4m}{d(N-1)^3} u_K^2.$$

Applying Lemma 26 to this recursion gives

$$u_K \leq \mathcal{O}\left(\frac{d}{m}\frac{(N-1)^3}{K}\right).$$

Finally, the error-to-potential inequality (D13) holds for every realization of the randomized tree. Taking its expectation and substituting the preceding potential bound proves

$$\mathbb{E}_{\boldsymbol{\Theta}}\mathbb{E}_{\hat{\mathrm{p}}}\left[\mathbb{I}\{T_{K,\boldsymbol{\Theta}}(\mathrm{X}) \neq g(\mathrm{X})\}\right] \leq \mathcal{O}\left(\frac{d}{m}\frac{N^3}{K}\right).$$

□

Proof of Lemma 12 Let $f^\star$ be a Bayes classifier, with ties resolved by a fixed rule. The binary excess-risk identity gives

$$R(g) - R^\star = \mathbb{E}_{\mathrm{X}}\left[|2\eta(\mathrm{X}) - 1|\mathbb{I}\{g(\mathrm{X}) \neq f^\star(\mathrm{X})\}\right] \tag{F19}$$

for every binary classifier g. If $F_M(\mathrm{x}) \neq f^\star(\mathrm{x})$, at least half of the base classifiers disagree with $f^\star(\mathrm{x})$. Hence, pointwise in x,

$$\mathbb{I}\{F_M(\mathrm{x}) \neq f^\star(\mathrm{x})\} \leq \frac{2}{M}\sum_{b=1}^{M}\mathbb{I}\{T_b(\mathrm{x}) \neq f^\star(\mathrm{x})\}.$$

Multiplying this inequality by $|2\eta(\mathrm{x}) - 1|$, taking expectation over X, and applying (F19) to every classifier proves the claim. No independence among the base classifiers is required.

Appendix G: Proof of Proposition 13

Proposition 13 *Under Assumptions 2 and 3, the density* p *is bounded on* $[0,1]^d$. *Define*

$$\bar{\mathrm{p}} \triangleq \sup_{\mathrm{x}\in[0,1]^d} \mathrm{p}(\mathrm{x}) < \infty.$$

For $0 \leq \delta_{\max} \leq 1/2$, *we have*

$$\Pr\left[\exists k \in [d] \quad \text{s.t.} \quad \mathrm{X}^{(k)} \in [0,\delta_{\max}] \cup [1-\delta_{\max}, 1]\right] \leq 2d\bar{\mathrm{p}}\delta_{\max} = \mathcal{O}(d\delta_{\max}).$$

Proof Under Assumption 2, every marginal density is continuous and bounded on the compact interval [0, 1]. Assumption 3 then implies that p is continuous and bounded on $[0,1]^d$. For each $k \in [d]$, let

$$A_k = \left\{\mathrm{X}^{(k)} \in [0,\delta_{\max}] \cup [1-\delta_{\max}, 1]\right\}.$$

The subset of $[0,1]^d$ corresponding to A_k has Lebesgue measure $2\delta_{\max}$. By the union bound,

$$\Pr\left[\bigcup_{k=1}^{d} A_k\right] \le \sum_{k=1}^{d} \Pr(A_k) \le 2d\bar{\mathrm{p}}\delta_{\max} = \mathcal{O}(d\delta_{\max}).$$

where the implicit constant is independent of $\delta_{\max}$. □

Appendix H: Proof of Corollary 15

Corollary 15 (Margin fast rate for GridForest) *Suppose Assumptions* 1–3 *and* 4 *hold. Suppose also that there exist constants* $0 < p_{\min} \le p_{\max} < \infty$ *such that*

$$p_{\min} \le \mathrm{p}(\mathrm{x}) \le p_{\max}, \qquad \mathrm{x} \in [0,1]^d .$$

If $0 \le \delta_{\max,n} \le h_n$, *then for some constant* $c > 0$ *the expected excess error of GridForest satisfies*

$$\mathbb{E}_{D_n,\boldsymbol{\Theta},\boldsymbol{\delta}}\big[R(f(\boldsymbol{\Theta},\boldsymbol{\delta}))\big] - R^\star \le \mathcal{O}\left(\frac{d}{m}\frac{1}{K_n h_n^3} + h_n^{\alpha+1} + \left(\frac{1}{a_n h_n^d}\right)^{(\alpha+1)/2} + d\delta_{\max,n} + \frac{d}{h_n} e^{-c a_n h_n}\right).$$

Choosing $a_n = \Theta(n)$, $h_n = \Theta(n^{-1/(d+2)})$,

$$K_n = \Omega\left(\frac{d}{m} n^{(\alpha+4)/(d+2)}\right) \quad \text{and} \quad \delta_{\max,n} = \mathcal{O}\left(n^{-(\alpha+1)/(d+2)}\right)$$

yields

$$\mathbb{E}_{D_n,\boldsymbol{\Theta},\boldsymbol{\delta}}\big[R(f(\boldsymbol{\Theta},\boldsymbol{\delta}))\big] - R^\star = \mathcal{O}\left(n^{-(\alpha+1)/(d+2)}\right) .$$

Proof We first restrict attention to the interior grid cubes whose side lengths equal h_n. Proposition 13 bounds the probability of their complement by $\mathcal{O}(d\delta_{\max,n})$.

The strong density bounds imply that every interior cube $\mathcal{C}$ has probability at least $p_{\min} h_n^d$. The conditional mean of the labels in $\mathcal{C}$ differs from $\eta(\mathrm{x})$ by at most $L h_n \sqrt{d}$ for every $\mathrm{x} \in \mathcal{C}$. Hoeffding's inequality, together with a binomial concentration bound for the number of observations in $\mathcal{C}$, therefore gives constants $c_1, c_2 > 0$ such that

$$\Pr_{D_{a_n}}\left[|\hat{\eta}_{a_n}(\mathrm{x}) - \eta(\mathrm{x})| \ge L h_n \sqrt{d} + u\right] \le c_1 \exp(-c_2 a_n h_n^d u^2) \tag{H20}$$

for every $u > 0$ and every x in an interior cube. Applying the standard plug-in argument under Assumption 4 to (H20) yields

$$\mathbb{E}_{D_{a_n}}\left[R(g_{a_n}^{\text{hist}})\right] - R^\star \;\le\; \mathcal{O}\left(h_n^{\alpha+1} + \left(\frac{1}{a_n h_n^d}\right)^{(\alpha+1)/2} + d\delta_{\max,n}\right). \tag{H21}$$

This is the usual fast-rate conversion for plug-in classifiers (Audibert and Tsybakov, 2007, Lemma 3.1).

It remains to transfer the tree-fitting guarantee from $\hat{\mathrm{p}}$ to p. The joint density bounds imply the same bounds for every marginal density. For each coordinate and each interior interval b of length h_n, a multiplicative Chernoff bound gives

$$\Pr\left[\hat{\mathrm{p}}_{X^{(k)}}(x) < p_{\min}/2 \text{ for } x \in b\right] \;\le\; \exp(-c_3 a_n h_n)$$

for a constant $c_3 > 0$. A union bound over at most d/h_n marginal intervals shows that all estimated marginal densities are at least $p_{\min}/2$ on the interior with probability at least

$$1 - \frac{d}{h_n}\exp(-c_3 a_n h_n)\,.$$

On this event, Assumption 3 gives

$$\mathrm{p}(\mathrm{x}) \;\le\; p_{\max}\left(\frac{2}{p_{\min}}\right)^d \hat{\mathrm{p}}(\mathrm{x})$$

throughout the interior. The population disagreement probability between a fitted tree and its histogram target is consequently bounded by a constant multiple of the disagreement probability under $\hat{\mathrm{p}}$, together with the boundary and concentration-failure probabilities. Lemma 11 and Theorem 7 then give the tree-fitting contribution

$$\mathcal{O}\left(\frac{d}{m}\frac{1}{K_n h_n^3} + d\delta_{\max,n} + \frac{d}{h_n}\exp(-c_3 a_n h_n)\right).$$

Combining this display with (H21) proves the first claim.

Finally, $a_n = \Theta(n)$ and $h_n = \Theta(n^{-1/(d+2)})$ make both histogram terms of order $n^{-(\alpha+1)/(d+2)}$. The stated choices of K_n and $\delta_{\max,n}$ give the same order for the tree-fitting and boundary terms. The remaining exponential term is asymptotically smaller than every polynomial in n. This proves the claimed fast rate. □

Appendix I: Proof of the Consistency Rate of GridGBDT

This appendix proves the results of Section 6.2.2, namely Theorem 16, Lemma 17, Lemma 18, and Theorem 19, together with the auxiliary split-gain bound of Lemma 34 that Section 6.2.2 states only informally. We first record the Efron–Stein inequality, which sub-

stitutes for the Bourgain-type inequality of Lemma 28 when the target is real-valued rather than $\{0,1\}$-valued.

Lemma 33 *(Efron–Stein inequality, Boucheron et al. (2013)[Theorem 3.1]) Let* $\mathrm{X}^{(1)},\dots,\mathrm{X}^{(d)}$ *be independent random variables and let* $f : \mathcal{X} \to \mathbb{R}$ *satisfy* $\mathbb{E}[f(\mathrm{X})^2] < \infty$. *Then*

$$\mathrm{Var}(f(\mathrm{X})) \;\le\; \sum_{k=1}^{d} \mathbb{E}_{\mathrm{X}^{(-k)}}\Big[\mathrm{Var}_{\mathrm{X}^{(k)}}\big(f(\mathrm{X}) \mid \mathrm{X}^{(-k)}\big)\Big] \;=\; \sum_{k=1}^{d} \mathrm{Inf}_k^{\mathrm{Var}}[f]\,.$$

A Split-Gain Lower Bound for Real-Valued Targets

Lemma 27 is not merely stated for a special case but bounds the specific optimization problem (D8), whose objective is written in terms of the alternating partial sums $\sum_{i\le k}(-1)^{i-1}p_i$, since the classification target it was designed for takes the values ± 1 on successive pieces. The residual histogram $\bar{\rho}_t$ takes arbitrary values in $[-B,B]$, for which these partial sums are undefined, so the split gain of (D8) is no longer a function of the masses p_i alone and the bound cannot be recovered by rescaling. The argument must instead be reorganized to keep the piece values explicit. The following lemma does so, and its proof uses the same sum-of-absolute-values argument as Lemma 25 without requiring the piece values to be ± 1. Setting $B = 1$ recovers the order of Lemma 27.

Lemma 34 *Let* $X \sim U[0,1]$ *and let* $f : [0,1] \to [-B,B]$ *be an* N*-piece function taking value* v_i *on its* i*-th piece, of probability mass* p_i, *for* $i = 1,\dots,N$. *For* $1 \le k < N$, *write*

$$P_k \triangleq \sum_{i\le k} p_i, \qquad D_k \triangleq \mathbb{E}[f \mid X < P_k] - \mathbb{E}[f \mid X \ge P_k]\,.$$

Then

$$\max_{1\le k<N} P_k(1-P_k)D_k^2 \ge \frac{\mathrm{Var}^2[f]}{4B^2(N-1)^2}\,.$$

Proof Let $\mu = \mathbb{E}[f] = \sum_i p_i v_i$ and $u_i = v_i - \mu$, so that $\sum_i p_i u_i = 0$, $\sum_i p_i u_i^2 = \mathrm{Var}[f] =: \sigma^2$, and $|u_i| \le 2B$ for every i, since $v_i \in [-B,B]$. Write $Q_k = \sum_{i\le k} p_i u_i$. Since $\sum_{i=1}^{N} p_i u_i = 0$, we have $\sum_{i>k} p_i u_i = -Q_k$, so

$$D_k = \frac{Q_k}{P_k} + \frac{Q_k}{1-P_k} = \frac{Q_k}{P_k(1-P_k)}\,.$$

Using $P_k(1-P_k) \le 1/4$,

$$\begin{aligned}\max_{1\le k<N} P_k(1-P_k)D_k^2 &= \max_{1\le k<N} \frac{Q_k^2}{P_k(1-P_k)} \\ &\ge 4 \max_{1\le k<N} Q_k^2\,.\end{aligned}$$

Since the maximum is no less than the average, and by Cauchy–Schwarz ($\sum_{k=1}^{n} a_k^2 \geq n^{-1}(\sum_k |a_k|)^2$),

$$4 \max_{1\leq k\leq N-1} Q_k^2 \;\geq\; \frac{4}{N-1}\sum_{k=1}^{N-1} Q_k^2 \;\geq\; \frac{4}{(N-1)^2}\left(\sum_{k=1}^{N-1}|Q_k|\right)^2 .$$

Applying Lemma 25 with $a_i = p_i u_i$ (so $S_i = |Q_i|$, and $Q_N = 0$ so $\sum_{k=1}^{N-1}|Q_k| = \sum_{k=1}^{N}|Q_k|$),

$$\sum_{k=1}^{N-1}|Q_k| \;\geq\; \frac{1}{2}\sum_{i=1}^{N} p_i|u_i| .$$

Since $|u_i| \leq 2B$, we have $u_i^2 \leq 2B|u_i|$, i.e., $|u_i| \geq u_i^2/(2B)$, so $\sum_i p_i|u_i| \geq \sigma^2/(2B)$. Combining the last three displays,

$$\max_{1\leq k<N} P_k(1-P_k)D_k^2 \geq \frac{4}{(N-1)^2}\left(\frac{\sigma^2}{4B}\right)^2 = \frac{\sigma^4}{4B^2(N-1)^2} ,$$

which completes the proof. □

Proof of Theorem 16

Theorem 16 *Suppose the target $g : \mathcal{X} \to [-B, B]$ is grid-aligned N-piece (Definition 3) and that the candidate cuts include the corresponding cut-point sets. Then, under Assumptions 1–3, the regression tree T generated by Algorithm 3 with the impurity* Var *in place of G (Section 7) and depth K satisfies*

$$\mathbb{E}_{\hat{\mathrm{p}}}\big[(T(\mathrm{X}) - g(\mathrm{X}))^2\big] \;\leq\; \mathcal{O}\left(\frac{B^2N^3}{K}\right) .$$

Proof We follow the two-step structure of the proof of Theorem 7 in Appendix D, replacing Lemma 28 by the Efron–Stein inequality (Lemma 33) and Lemma 27 by Lemma 34.

Let $\epsilon(T) \triangleq \mathbb{E}_{\hat{\mathrm{p}}}\big[(T(\mathrm{X}) - g(\mathrm{X}))^2\big]$ denote the fitting error of a regression tree T against the target g, and for a leaf t let $\epsilon(T \mid t) \triangleq \mathrm{Var}_{\hat{\mathrm{p}}}(g \mid \mathrm{X} \in t)$, the mean squared error of predicting the constant $\mathbb{E}_{\hat{\mathrm{p}}}[g \mid \mathrm{X} \in t]$ on t, which is exactly the value Algorithm 3 predicts at leaf t when run with the Var impurity. Because t is an axis-aligned box and $\hat{\mathrm{p}}$ is a product distribution (Assumption 3), $\hat{\mathrm{p}}$ restricted to t is again a product distribution, so Lemma 33 applies conditionally on t:

$$\epsilon(T \mid t) \;=\; \mathrm{Var}_{\hat{\mathrm{p}}}(g \mid \mathrm{X} \in t) \;\leq\; \sum_{k=1}^{d} \mathrm{Inf}_k^{\mathrm{Var}}[g \mid t] \;=\; d\,\mathcal{G}(t) ,$$

where $\mathcal{G}(t) = \frac{1}{d}\sum_{k=1}^{d} \mathrm{Inf}_k^{\mathrm{Var}}[g \mid t]$, as in (D12). Summing over the leaves of the depth-K tree T_K,

$$\epsilon(T_K) \;=\; \sum_{t\in\text{leaves}(T_K)} p_t\,\epsilon(T_K \mid t) \;\le\; d\,C_K\,, \tag{I22}$$

with $C_K = \sum_t p_t \mathcal{G}(t)$ as in Section 4.1.

For compactness, define

$$\mu^g_{k,L}(s) \triangleq \mathbb{E}_{\hat{\mathrm{p}}^{(k)}}[g \mid t_L], \quad \mu^g_{k,R}(s) \triangleq \mathbb{E}_{\hat{\mathrm{p}}^{(k)}}[g \mid t_R],$$

$$V_k(t) \triangleq \mathrm{Var}_{\hat{\mathrm{p}}^{(k)}}[g \mid t], \quad c_N \triangleq \frac{1}{4B^2(N-1)^3}\,.$$

For the purity gain, the law of total variance gives the following exact identity for any leaf t split at (k, s) into t_L and t_R.

$$\delta_g(t,k,s) \triangleq \mathbb{E}_{\hat{\mathrm{p}}^{(j\neq k)}}\left[w_{t_L} w_{t_R}\big(\mu^g_{k,L}(s) - \mu^g_{k,R}(s)\big)^2\right].$$

Restricting cuts to the grid $\mathcal{S}_k = \{h_n, \ldots, (1/h_n - 1)h_n\}$, $|\mathcal{S}_k| = N - 1$ with $N = 1/h_n$, exactly as in Appendix D, and applying Lemma 34 coordinate-wise (conditionally on $\mathrm{X}^{(-k)}$, with range bound B, then taking $\mathbb{E}_{\hat{\mathrm{p}}^{(j\neq k)}}$ of both sides) in place of Lemma 27, the same chain of inequalities as in Appendix D gives

$$\begin{aligned}
\max_{k,s} \delta_g(t,k,s) &\ge \frac{c_N}{d}\sum_{k=1}^{d} \mathbb{E}_{\hat{\mathrm{p}}^{(j\neq k)}}[V_k(t)^2] \\
&\ge \frac{c_N}{d}\sum_{k=1}^{d} \big(\mathrm{Inf}_k^{\mathrm{Var}}[g \mid t]\big)^2 \\
&\ge c_N \left(\frac{1}{d}\sum_{k=1}^{d} \mathrm{Inf}_k^{\mathrm{Var}}[g \mid t]\right)^2 \\
&= c_N\,\mathcal{G}(t)^2\,,
\end{aligned}$$

The last two inequalities follow from Jensen's inequality, since $\mathbb{E}[X^2] \ge (\mathbb{E}X)^2$ and $d^{-1}\sum_k a_k^2 \ge (d^{-1}\sum_k a_k)^2$. Summing over leaves and using Jensen's inequality once more gives

$$\begin{aligned}
C_K - C_{K+1} &= \sum_{t\in\text{leaves}(T_K)} p_t \max_{k,s} \delta_g(t,k,s) \\
&\ge c_N \sum_t p_t\,\mathcal{G}(t)^2 \\
&\ge c_N C_K^2\,.
\end{aligned}$$

By Lemma 26 with $C = 4B^2(N-1)^3$ (valid since $C_1 \le B^2 \le C$),

$$C_K \;\le\; \frac{4B^2(N-1)^3}{K}\,.$$

Combining with (I22),

$$\epsilon(T_K) \;\le\; d\,C_K \;\le\; \frac{4dB^2(N-1)^3}{K} \;=\; \mathcal{O}\!\left(\frac{B^2N^3}{K}\right),$$

which completes the proof. □

Proof of Lemma 17

Lemma 17 *Let F_{t-1} be an iterate generated by Algorithm 6 with $\delta_{\max} = 0$. Then F_{t-1} is constant on every unshifted grid cube $\mathcal{C} \in \mathcal{J}$, and the residual histogram satisfies $\bar{\rho}_t(\mathcal{C}) = \hat{\eta}_n(\mathcal{C}) - F_{t-1}(\mathcal{C})$ for every $\mathcal{C} \in \mathcal{J}$.*

Proof We argue by induction on t. The base case $F_0 \equiv n^{-1}\sum_i Y_i$ is constant everywhere, in particular on every grid cube. Suppose F_{t-2} is constant on every cube of $\mathcal{J}$ (with $F_{-1} \triangleq F_0$ interpreted trivially for $t = 1$). Because Algorithm 3 (Section 5.3) restricts every candidate cut to $\{h_n, 2h_n, \dots\}$, every leaf of T_{t-1} is a union of elementary grid cubes, so T_{t-1} is constant on every $\mathcal{C} \in \mathcal{J}$. Since $\mathrm{clip}_{[0,1]}$ is applied pointwise to the (now constant-on-$\mathcal{C}$) sum $F_{t-2}(\mathrm{x}) + \nu T_{t-1}(\mathrm{x})$, $F_{t-1} = \mathrm{clip}_{[0,1]}(F_{t-2} + \nu T_{t-1})$ is likewise constant on every $\mathcal{C} \in \mathcal{J}$, completing the induction.

Given that F_{t-1} is constant on $\mathcal{C}$, write $F_{t-1}(\mathcal{C})$ for its value there. For every sample i with $\mathrm{X}_i \in \mathcal{C}$, $Y_i - F_{t-1}(\mathrm{X}_i) = Y_i - F_{t-1}(\mathcal{C})$, so averaging over the samples falling in $\mathcal{C}$,

$$\bar{\rho}_t(\mathcal{C}) \;=\; \frac{1}{n_{\mathcal{C}}}\sum_{i:\mathrm{X}_i\in\mathcal{C}}\big(Y_i - F_{t-1}(\mathcal{C})\big) \;=\; \left(\frac{1}{n_{\mathcal{C}}}\sum_{i:\mathrm{X}_i\in\mathcal{C}} Y_i\right) - F_{t-1}(\mathcal{C}) \;=\; \hat{\eta}_n(\mathcal{C}) - F_{t-1}(\mathcal{C}),$$

by the definition of $\hat{\eta}_n(\mathcal{C})$ in Section 5.2, which completes the proof. □

Proof of Lemma 18

Lemma 18 *Suppose every round of Algorithm 6 satisfies the guarantee of Theorem 16 with $B = 1$, $N = 1/h_n$, i.e., $\mathbb{E}_{\hat{\mathrm{p}}}\big[(T_t(\mathrm{X}) - \bar{\rho}_t(\mathrm{X}))^2\big] \le \gamma$ for $\gamma = \mathcal{O}\big(1/(Kh_n^3)\big)$. Then*

$$\sqrt{e_t} \;\le\; (1-\nu)\sqrt{e_{t-1}} + \nu\sqrt{\gamma}\,, \qquad t = 1, \dots, M\,,$$

and consequently $\sqrt{e_M} \le \sqrt{\gamma} + (1-\nu)^M\sqrt{e_0}$, with $e_0 \le 1/4$.

Proof By Lemma 17, $\bar{\rho}_t = \hat{\eta}_n - F_{t-1}$ as functions on $\mathcal{J}$. Let $\tilde{F}_t \triangleq F_{t-1} + \nu T_t$ denote the unclipped update, so $F_t = \mathrm{clip}_{[0,1]}(\tilde{F}_t)$. Since $\hat{\eta}_n(\mathcal{C}) \in [0,1]$ for every $\mathcal{C}$ and clipping to $[0, 1]$ is a 1-Lipschitz projection onto a convex set containing $\hat{\eta}_n(\mathcal{C})$, we have $|\hat{\eta}_n(\mathcal{C}) - F_t(\mathcal{C})| \le |\hat{\eta}_n(\mathcal{C}) - \tilde{F}_t(\mathcal{C})|$ for every $\mathcal{C}$, so $e_t \le \tilde{e}_t \triangleq \sum_{\mathcal{C}} \Pr_{\hat{\mathrm{p}}}[\mathrm{X} \in \mathcal{C}](\hat{\eta}_n(\mathcal{C}) - \tilde{F}_t(\mathcal{C}))^2$.

Write $g_t \triangleq \bar{\rho}_t - T_t$, the round's own fitting residual, so that $\sum_{\mathcal{C}} \Pr_{\hat{\mathrm{p}}}[\mathrm{X} \in \mathcal{C}] g_t(\mathcal{C})^2 = \mathbb{E}_{\hat{\mathrm{p}}}\big[(T_t(\mathrm{X}) - \bar{\rho}_t(\mathrm{X}))^2\big] \le \gamma$ by hypothesis. Since $\hat{\eta}_n - \bar{F}_t = \bar{\rho}_t - \nu T_t = (1-\nu)\bar{\rho}_t + \nu g_t$,

$$\begin{aligned}
\tilde{e}_t &= \sum_{\mathcal{C}} \Pr_{\hat{\mathrm{p}}}[\mathrm{X} \in \mathcal{C}] \Big[(1-\nu)\bar{\rho}_t(\mathcal{C}) + \nu g_t(\mathcal{C})\Big]^2 \\
&= (1-\nu)^2 e_{t-1} + 2\nu(1-\nu) \sum_{\mathcal{C}} \Pr_{\hat{\mathrm{p}}}[\mathrm{X} \in \mathcal{C}]\, \bar{\rho}_t(\mathcal{C}) g_t(\mathcal{C}) + \nu^2 \sum_{\mathcal{C}} \Pr_{\hat{\mathrm{p}}}[\mathrm{X} \in \mathcal{C}]\, g_t(\mathcal{C})^2 \\
&\le (1-\nu)^2 e_{t-1} + 2\nu(1-\nu)\sqrt{e_{t-1}\gamma} + \nu^2\gamma \;=\; \Big[(1-\nu)\sqrt{e_{t-1}} + \nu\sqrt{\gamma}\Big]^2 ,
\end{aligned}$$

where the inequality uses $\sum_{\mathcal{C}} \Pr_{\hat{\mathrm{p}}}[\mathrm{X} \in \mathcal{C}] \bar{\rho}_t(\mathcal{C})^2 = e_{t-1}$, Cauchy–Schwarz for the cross term, and $\sum_{\mathcal{C}} \Pr_{\hat{\mathrm{p}}}[\mathrm{X} \in \mathcal{C}] g_t(\mathcal{C})^2 \le \gamma$. Taking square roots and combining with $e_t \le \tilde{e}_t$ gives $\sqrt{e_t} \le (1-\nu)\sqrt{e_{t-1}} + \nu\sqrt{\gamma}$. Unrolling,

$$\sqrt{e_t} - \sqrt{\gamma} \;\le\; (1-\nu)\big(\sqrt{e_{t-1}} - \sqrt{\gamma}\big) \;\le\; \cdots \;\le\; (1-\nu)^t\big(\sqrt{e_0} - \sqrt{\gamma}\big) \;\le\; (1-\nu)^t \sqrt{e_0}\,,$$

so $\sqrt{e_M} \le \sqrt{\gamma} + (1-\nu)^M \sqrt{e_0}$. Finally, since $F_0 = n^{-1} \sum_i Y_i$ equals the $\hat{\mathrm{p}}$-weighted average of $\hat{\eta}_n$ over $\mathcal{J}$, $e_0 = \sum_{\mathcal{C}} \Pr_{\hat{\mathrm{p}}}[\mathrm{X} \in \mathcal{C}](\hat{\eta}_n(\mathcal{C}) - F_0)^2 = \mathrm{Var}_{\hat{\mathrm{p}}}(\hat{\eta}_n)$, which is at most 1/4 because $\hat{\eta}_n$ takes values in [0, 1] (any [0, 1]-valued random variable has variance at most 1/4). This completes the proof. □

Proof of Theorem 19

Theorem 19 (Consistency rate of GridGBDT for binary classification) *Under Assumptions 1–3, fix any shrinkage $\nu \in (0,1)$ and let $\hat{f}_M$ be the output of Algorithm 6 with $\delta_{\max} = 0$, so that all boosting rounds share the same unshifted grid. The expected excess error satisfies*

$$\mathbb{E}_{D_n}\big[R(\hat{f}_M)\big] - R^\star \;\le\; \mathcal{O}\left(\sqrt{\frac{1}{K h_n^3}} + (1-\nu)^M + h_n + \sqrt{\frac{1}{n h_n^d}}\right).$$

Choosing $h_n = \Theta\big(n^{-1/(d+2)}\big)$, $K = K_n = \Omega\big(n^{5/(d+2)}\big)$, and $M = M_n = \Omega(\log n)$, we obtain a consistency rate of order $\mathcal{O}\big(n^{-1/(d+2)}\big)$ for GridGBDT, matching the rate of GridCART (Theorem 8) under the same assumptions.

Proof Since every iterate is clipped to $[0,1]$, we have $F_M : \mathcal{X} \to [0,1]$, and $\hat{f}_M = \mathbb{I}\{F_M > 1/2\}$ is the plug-in classifier induced by F_M. Lemma 24 applies to any such [0, 1]-valued function, since its proof is pointwise and uses no property of the estimator; applying it with F_M in place of $\hat{\eta}_n$ gives

$$R(\hat{f}_M) - R^\star \;\le\; 2 \int |\eta(\mathrm{x}) - F_M(\mathrm{x})|\, \mathrm{p}(\mathrm{x})\mathrm{dx} \;\le\; 2 \underbrace{\int |\eta - \hat{\eta}_n| \mathrm{p}\mathrm{dx}}_{\text{term (c)}} + 2 \underbrace{\int |\hat{\eta}_n - F_M| \mathrm{p}\mathrm{dx}}_{\text{term (a)}},$$

by inserting $\hat{\eta}_n$ and applying the triangle inequality. Term (c) is $\mathcal{O}(h_n\sqrt{d} + \sqrt{1/(n h_n^d)})$ in expectation over D_n: it is exactly the quantity bounded by terms A and B in the proof of

Lemma 6 (Appendix C), and is unaffected by boosting. For term (a), since $|\hat{\eta}_n - F_M| \le 1$ pointwise, replacing p by the estimate $\hat{\mathrm{p}}$ costs at most a total-variation term:

$$\int |\hat{\eta}_n - F_M| \mathrm{pdx} \;\le\; \mathbb{E}_{\hat{\mathrm{p}}}\big[|\hat{\eta}_n(\mathrm{X}) - F_M(\mathrm{X})|\big] + \underbrace{\|\hat{\mathrm{p}}_{\mathrm{X}} - \mathrm{p}_{\mathrm{X}}\|_1}_{\text{term (b)}},$$

and term (b) is $\mathcal{O}(d(h_n + \sqrt{1/(nh_n)}))$ in expectation by Lemma 5, as in (2).

It remains to bound $\mathbb{E}_{\hat{\mathrm{p}}}|\hat{\eta}_n - F_M|$. By Cauchy–Schwarz and the definition of the potential,

$$\mathbb{E}_{\hat{\mathrm{p}}}\big[|\hat{\eta}_n(\mathrm{X}) - F_M(\mathrm{X})|\big] \;\le\; \sqrt{\mathbb{E}_{\hat{\mathrm{p}}}\big[(\hat{\eta}_n(\mathrm{X}) - F_M(\mathrm{X}))^2\big]} \;=\; \sqrt{e_M},$$

where the equality holds because both $\hat{\eta}_n$ and F_M are constant on every grid cube (Lemma 17). By Lemma 18,

$$\sqrt{e_M} \;\le\; \sqrt{\gamma} + (1-\nu)^M \sqrt{e_0} \;\le\; \sqrt{\gamma} + \frac{(1-\nu)^M}{2}, \qquad \gamma = \mathcal{O}\left(\frac{1}{Kh_n^3}\right),$$

by Theorem 16 with $B = 1$ and $N = 1/h_n$. Hence term (a) is $\mathcal{O}\big(\sqrt{1/(Kh_n^3)} + (1-\nu)^M\big)$, and combining terms (a), (b), (c) gives the stated bound.

For the rate, $h_n = \Theta(n^{-1/(d+2)})$ makes terms (b) and (c) both $\Theta(n^{-1/(d+2)})$, as in the proof of Theorem 8. Since $(1-\nu)^M$ decays exponentially in M for any fixed $\nu \in (0,1)$, there is $M_n = \Omega(\log n)$ (with constant depending on ν, d) such that $(1-\nu)^{M_n} \le n^{-1/(d+2)}$. For the depth, with $K_n = \Omega\big(n^{5/(d+2)}\big)$ and $h_n = \Theta(n^{-1/(d+2)})$,

$$\sqrt{\frac{1}{K_n h_n^3}} \;=\; \mathcal{O}\left(\sqrt{n^{-5/(d+2)+3/(d+2)}}\right) \;=\; \mathcal{O}\left(n^{-1/(d+2)}\right).$$

All terms of the bound are therefore $\mathcal{O}(n^{-1/(d+2)})$ under these choices, completing the proof. □

Proof of Proposition 20

Proposition 20 (Refinement under a margin condition) *Suppose Assumption 5 also holds with exponent $\beta \ge 0$. In the setting of Theorem 19, the combined tree-fitting and optimization contribution may then be bounded by*

$$\mathcal{O}\left(\left(\frac{1}{Kh_n^3} + (1-\nu)^{2M}\right)^{\beta/(\beta+2)}\right).$$

For $\beta > 0$, the choice $K_n = \Omega\big(n^{(4\beta+2)/(\beta(d+2))}\big)$ suffices to retain the rate $\mathcal{O}(n^{-1/(d+2)})$.

Proof Under Assumption 5 one may bound term (a) by comparing $\hat{f}_M$ with $\mathrm{g}_n^{\mathrm{hist}} = \mathbb{I}\{\hat{\eta}_n > 1/2\}$ instead of routing through Lemma 24. Writing

$R(\hat{f}_M) - R^\star = (R(\hat{f}_M) - R(g_n^{\text{hist}})) + (R(g_n^{\text{hist}}) - R^\star)$, bounding the first difference by the disagreement probability under p, and replacing p by $\hat{\text{p}}$ at the cost of term (b) as in (2), the residual quantity to control is $\Pr_{\hat{\text{p}}}[\hat{f}_M \neq g_n^{\text{hist}}]$, while $R(g_n^{\text{hist}}) - R^\star$ is again term (c), bounded by Lemma 6.

If $\hat{f}_M(\text{x}) \neq g_n^{\text{hist}}(\text{x})$, then $F_M(\text{x})$ and $\hat{\eta}_n(\text{x})$ lie on opposite sides of 1/2, since both classifiers threshold at 1/2. Hence, for any $\tau > 0$: either $|\hat{\eta}_n(\text{x}) - 1/2| < \tau$, or $|\hat{\eta}_n(\text{x}) - 1/2| \geq \tau$, in which case 1/2 lying strictly between $F_M(\text{x})$ and $\hat{\eta}_n(\text{x})$ forces $|\hat{\eta}_n(\text{x}) - F_M(\text{x})| \geq |\hat{\eta}_n(\text{x}) - 1/2| \geq \tau$. Pointwise, therefore,

$$\mathbb{I}\{\hat{f}_M(\text{x}) \neq g_n^{\text{hist}}(\text{x})\} \;\leq\; \mathbb{I}\{|\hat{\eta}_n(\text{x}) - 1/2| < \tau\} + \mathbb{I}\{|\hat{\eta}_n(\text{x}) - F_M(\text{x})| \geq \tau\} \,.$$

Taking expectation under $\hat{\text{p}}$, using Assumption 5 for the first term and Markov's inequality with $\mathbb{E}_{\hat{\text{p}}}[(\hat{\eta}_n - F_M)^2] = e_M$ for the second,

$$\Pr_{\hat{\text{p}}}[\hat{f}_M \neq g_n^{\text{hist}}] \;\leq\; C_0 \tau^\beta + \frac{e_M}{\tau^2} \,, \qquad \forall \tau > 0 \,.$$

Choosing $\tau = \Theta(e_M^{1/(\beta+2)})$ balances the two terms and gives $\mathcal{O}(e_M^{\beta/(\beta+2)})$. By Lemma 18, $e_M \leq \mathcal{O}\big(1/(Kh_n^3) + (1-\nu)^{2M}\big)$, which is the stated replacement for term (a). For the depth, with $K_n = \Omega\big(n^{(4\beta+2)/(\beta(d+2))}\big)$ and $h_n = \Theta(n^{-1/(d+2)})$,

$$\frac{1}{K_n h_n^3} \;=\; \mathcal{O}\left(n^{-(4\beta+2)/(\beta(d+2))+3/(d+2)}\right) \;=\; \mathcal{O}\left(n^{-(\beta+2)/(\beta(d+2))}\right) \,,$$

using $-\frac{4\beta+2}{\beta(d+2)} + \frac{3}{d+2} = -\frac{\beta+2}{\beta(d+2)}$; raising to the power $\beta/(\beta+2)$ gives $\mathcal{O}(n^{-1/(d+2)})$, and $M_n = \Omega(\log n)$ makes $(1-\nu)^{2M_n}$ negligible as before. This completes the proof. □

Acknowledgements This research was supported by National Natural Science Foundation of China (Collaborative Innovation Center of Novel Software Technology and Industrialization. The authors would like to thank Tian Qin, Jia-Wei Shan, and Jin-Hui Wu for their constructive comments and discussions.

Author Contributions Qin-Cheng Zheng initiated and developed the approach, carried out the experiments, and prepared the initial manuscript. Shao-Qun Zhang and Shen-Huan Lyu helped refine the method and analysis, and contributed to manuscript revision. Yuan Jiang and Zhi-Hua Zhou conceived the study and provided overall guidance on its direction and planning. All authors discussed the findings, contributed to the final manuscript, and approved the submitted version.

Funding This research was supported by the National Natural Science Foundation of China (62576165).

Data Availability The real-world datasets used in this work are publicly available from OpenML.

Code Availability The code will be released after publication.

Materials Availability The datasets and experimental results used in this work are available with the accompanying code.

References

Audibert, J., & Tsybakov, A. B. (2007). Fast learning rates for plug-in classifiers. *The Annals of Statistics, 35*(2), 608–633. https://doi.org/10.1214/009053606000001217

Biau, G., Devroye, L., & Lugosi, G. (2008). Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research, 9*(66), 2015–2033.

Bischl, B., Casalicchio, G., & Feurer, M. (2021). Openml benchmarking suites. Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks

Blanc, G., Lange, J., & Tan, L. (2020a) Provable guarantees for decision tree induction: The agnostic setting. In *Proceedings of the 37th International Conference on Machine Learning*, pp 941–949

Blanc, G., Lange, J., & Tan, L. (2020b) Top-down induction of decision trees: Rigorous guarantees and inherent limitations. In *Proceedings of the 11th Innovations in Theoretical Computer Science Conference*, pp 44:1–44:44, https://doi.org/10.4230/LIPIcs.ITCS.2020.44

Boucheron, S., Lugosi, G., & Massart, P. (2013). Concentration inequalities: A nonasymptotic theory of independence. *Oxford University Press*. https://doi.org/10.1093/acprof:oso/9780199535255.001.0001

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Breiman, L., Friedman, J. H., Olshen, R. A., et al. (1984). *Classification and Regression Trees*. Belmont: Wadsworth.

Brodley, C. E., & Utgoff, P. E. (1995). Multivariate decision trees. *Machine Learning, 19*(1), 45–77. https://doi.org/10.1023/A:1022607123649

Brutzkus, A., Daniely, A., & Malach, E. (2020). ID3 learns juntas for smoothed product distributions. *Proceedings of the 33rd Conference on Learning Theory* (pp. 902–915)

Bühlmann, P., & Yu, B. (2003). Boosting with the L_2 loss: Regression and classification. *Journal of the American Statistical Association, 98*(462), 324–339.

Chen, T., & Guestrin, C. (2016) XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp 785–794, https://doi.org/10.1145/2939672.2939785

Devroye, L., Györfi, L., & Lugosi, G. (1997). A Probabilistic Theory of Pattern Recognition. *Springer*. https://doi.org/10.1007/978-1-4612-0711-5

Fiat, A., & Pechyony, D. (2004) Decision trees: More theoretical justification for practical algorithms. In *Proceedings of the 15th International Conference on Algorithmic Learning Theory*, pp 156–170, https://doi.org/10.1007/978-3-540-30215-5_13

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics, 29*(5), 1189–1232.

Gao, W., & Zhou, Z. (2013). On the doubt about margin explanation of boosting. *Artificial Intelligence, 203*, 1–18. https://doi.org/10.1016/j.artint.2013.07.002

Gao, W., & Zhou, Z. (2020). Towards convergence rate analysis of random forests for classification. *Advances in Neural Information Processing Systems* (pp. 9300–9311)

Gao, W., Xu, F., & Zhou, Z. (2022). Towards convergence rate analysis of random forests for classification. *Artificial Intelligence, 313*, 103788. https://doi.org/10.1016/j.artint.2022.103788

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning, 63*(1), 3–42. https://doi.org/10.1007/s10994-006-6226-1

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). The Elements of Statistical Learning. *Springer*. https://doi.org/10.1007/978-0-387-84858-7

Kahn, J., Kalai, G., & Linial, N. (1988) The influence of variables on boolean functions. In: *Proceedings of the 29th Annual Symposium on Foundations of Computer Science* (pp. 68–80), https://doi.org/10.1109/SFCS.1988.21923

Kalai, A.T., & Teng, S. (2008) Decision trees are PAC-learnable from most product distributions: A smoothed analysis. arXiv preprint arXiv:0812.0933. https://doi.org/10.48550/arXiv.0812.0933

Ke, G., Meng, Q., & Finley, T. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems* (pp. 3149–3157)

Keller, N. (2011). On the influences of variables on boolean functions in product spaces. *Combinatorics, Probability and Computing, 20*(1), 83–102. https://doi.org/10.1017/S0963548310000234

Klusowski, J. M., & Tian, P. M. (2024). Large scale prediction with decision trees. *Journal of the American Statistical Association, 119*(545), 525–537. https://doi.org/10.1080/01621459.2022.2126782

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

Mazumder, R., & Wang, H. (2023). On the convergence of CART under sufficient impurity decrease condition. *Advances in Neural Information Processing Systems* (pp. 57754–57782)

Mingers, J. (1989). An empirical comparison of selection measures for decision-tree induction. *Machine Learning, 3*(4), 319–342. https://doi.org/10.1007/BF00116837
O'Donnell, R. (2014). Analysis of Boolean Functions. *Cambridge University Press*. https://doi.org/10.1017/CBO9781139814782
Prokhorenkova, L. O., Gusev, G., & Vorobev, A. (2018). CatBoost: Unbiased boosting with categorical features. *Advances in Neural Information Processing Systems* (pp. 6639–6649)
Quinlan, J. R. (1979). Discovering rules by induction from large collections of examples. In D. Michie (Ed.), *Expert Systems in the Micro-Electronic Age.* Edinburgh: Edinburgh University Press.
Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*(1), 81–106. https://doi.org/10.1007/BF00116251
Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann.
Scornet, E., Biau, G., & Vert, J. (2015). Consistency of random forests. *The Annals of Statistics, 43*(4), 1716–1741. https://doi.org/10.1214/15-AOS1321
Stone, C. J. (1982). Optimal global rates of convergence for nonparametric regression. *The Annals of Statistics, 10*(4), 1040–1053. https://doi.org/10.1214/aos/1176345969
Takimoto, E., & Maruoka, A. (2003). Top-down decision tree learning as information based boosting. *Theoretical Computer Science, 292*(2), 447–464. https://doi.org/10.1016/S0304-3975(02)00181-0
Tan, Y. S., Agarwal, A., & Yu, B. (2022). A cautionary tale on fitting decision trees to data from additive models: Generalization lower bounds. *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics* (pp. 9663–9685)
Tsybakov, A. B. (2009). Introduction to Nonparametric Estimation. *Springer*. https://doi.org/10.1007/b13794
Utgoff, P. E. (1989). Incremental induction of decision trees. *Machine Learning, 4*, 161–186. https://doi.org/10.1023/A:1022699900025
Vanschoren, J., van Rijn, J. N., Bischl, B., et al. (2013). OpenML: Networked science in machine learning. *ACM SIGKDD Explorations Newsletter, 15*(2), 49–60. https://doi.org/10.1145/2641190.2641198
Wang, L., Sugiyama, M., & Yang, C. (2008). On the margin explanation of boosting algorithms. *Proceedings of the 21st Annual Conference on Learning Theory* (pp. 479–490)
Wasserman, L. (2006). All of Nonparametric Statistics. *Springer*. https://doi.org/10.1007/0-387-30623-4
Yang, Y. (1999). Minimax nonparametric classification-part I: Rates of convergence. *IEEE Transactions on Information Theory, 45*(7), 2271–2284. https://doi.org/10.1109/18.796368
Zhang, T., & Yu, B. (2005). Boosting with early stopping: Convergence and consistency. *The Annals of Statistics, 33*, 1538–1579. https://doi.org/10.1214/009053605000000255
Zheng, Q., Lyu, S., & Zhang, S. (2023). On the consistency rate of decision tree learning algorithms. *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics* (pp. 7824–7848)
Zhou, Z. (2012). Ensemble Methods: Foundations and Algorithms. *CRC Press, Boca Raton, FL,*. https://doi.org/10.1201/b12207
Zhou, Z., & Feng, J. (2019). Deep forest. *National Science Review, 6*(1), 74–86. https://doi.org/10.1093/nsr/nwy108

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Qin-Cheng Zheng[1,2] · Shao-Qun Zhang[1,3] · Shen-Huan Lyu[4,5] · Yuan Jiang[1,2] · Zhi-Hua Zhou[1,2]

✉ Zhi-Hua Zhou
zhouzh@lamda.nju.edu.cn

Qin-Cheng Zheng
zhengqc@lamda.nju.edu.cn

Shao-Qun Zhang
zhangsq@lamda.nju.edu.cn

Shen-Huan Lyu
lvsh@hhu.edu.cn

Yuan Jiang
jiangy@lamda.nju.edu.cn

[1] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China

[2] School of Artificial Intelligence, Nanjing University, Nanjing 210023, China

[3] School of Intelligence Science and Technology, Nanjing University, Suzhou 215163, China

[4] College of Computer Science and Software Engineering, Hohai University, Nanjing 211100, China

[5] Key Laboratory of Water Big Data Technology of Ministry of Water Resources, Hohai University, Nanjing 211100, China