On the Consistency Rate of Decision Tree Learning Algorithms

Qin-Cheng Zheng, Shen-Huan Lyu, Shao-Qun Zhang, Yuan Jiang, Zhi-Hua Zhou
National Key Laboratory for Novel Software Technology, Nanjing University, China
{zhengqc, lvsh, zhangsq, jiangy, zhouzh}@lamda.nju.edu.cn

Abstract

Decision tree learning algorithms such as CART are generally based on heuristics that maximizes the purity gain greedily. Though these algorithms are practically successful, theoretical properties such as consistency are far from clear. In this paper, we discover that the most serious obstacle encumbering consistency analysis for decision tree learning algorithms lies in the fact that the worst-case purity gain, i.e., the core heuristics for tree splitting, can be zero. Based on this recognition, we present a new algorithm, named Grid Classification And Regression Tree (GridCART), with a provable consistency rate $\mathcal{O}(n^{-1/(d+2)})$, which is the first consistency rate proved for heuristic tree learning algorithms.

1 INTRODUCTION

Decision trees are among the most popular methods in machine learning and data mining. The widely used CART [Breiman et al., 1984] is a heuristic tree-based learning algorithm and is usually selected to be a base learner in ensemble learning such as the popular random forest [Breiman, 2001], gradient boosting decision tree [Chen and Guestrin, 2016; Ke et al., 2017] and deep forest [Zhou and Feng, 2019]. At each step, CART chooses a dimension and a cut point to maximize the purity gain, and then splits every leaf t into two children $t \cap \{\mathbf{x} \mid \mathbf{x}^{(k)} < s\}$ and $t \cap \{\mathbf{x} \mid \mathbf{x}^{(k)} \geq s\}$. The process will continue recursively until all the leaves contain samples with the same label. The algorithm is highly heuristic, but it succeeds in various types of tasks. Thus, it is crucial to understand the mysteries behind the great success.

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], and thus a successful algorithm should be provably consistent. Nevertheless, despite having been proposed for more than 30 years [Breiman et al., 1984], whether this type of heuristic algorithm for tree learning is generally consistent still remains mysterious. There have been a lot of efforts on this issue. Devroye et al. [1997], Biau et al. [2008], and Gao and Zhou [2020] showed the consistency of pure random trees, which splits every leaf independently on samples, whereas the label-dependent analysis of tree generating is instead the most challenging. Scornet et al. [2015] was the first to analyze the label-dependent node splitting and showed that CART is consistent under the assumptions of an additive target function and uniformly randomly distributing features. Klusowski [2021] eliminated all the assumptions on the feature distribution and presented that CART is consistent in the high dimension for additive target functions. However, we usually have no side information about the target function when one practically uses heuristic algorithms in real-world tasks. Thus, these assumptions are still far from mild.

In this paper, we revisit this issue by studying a new impurity measure, called Influence [Kahn et al., 1988]. Our study point is from some seminal works [Blanc et al., 2020b; Fiat and Pechyony, 2004], in which they showed the possibility that using Influence as the impurity measure contributes to the convergence of error, which is a key ingredient for consistency without any assumptions on the target function. Nevertheless, previous studies related to Influence cannot give consideration to both consistency and practiced effectiveness, since an Influence oracle is required for tree generating. We bridge this gap by proposing GridCART, which not only can run practically but also is consistent without assumptions on the target function. The contributions of this work are summarized as follows:

1. We develop an in-depth analysis for the consistency of the heuristic tree learning algorithms and disclose a serious obstacle that the worst-case gain in purity equals to zero when proving consistency.
2. We propose Grid Classification And Regression Tree (GridCART), whose gain in purity at each node splitting can be lower bounded nontrivially, making it fea-

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206. Copyright 2023 by the author(s).

sible for the consistency analysis.

3. We present a consistency rate $\mathcal{O}(n^{-1/(d+2)})$ for GridCART, which is the first consistency rate for heuristic tree learning algorithms, even under weaker assumptions for consistency in previous studies.

The rest of this paper is organized as follows. Section 2 reviews some previous works. Section 3 introduces some essential background knowledge and notations. Section 4 shows the difficulty in proving the convergence of CART. Section 5 presents the GridCART for binary classification, its consistency rate, and the time complexity. Section 6 extends GridCART to more general tasks. Section 7 conduct simulation experiments to verity the effectiveness of our theory. Section 8 concludes our work with prospects.

2 RELATED WORK

The history of decision trees dates back to the 1970s. Quinlan [1979, 1986] first proposed ID3 for classification in discrete feature space, one of the most popular tree learning algorithms till now. C4.5 for continuous features and CART for both classification and regression were then proposed by Quinlan [1993] and Breiman et al. [1984], respectively. Brodley and Utgoff [1995] thought about multivariate decision trees, which have no restriction on the orthogonality of the split. Mingers [1989] studied how the choice of splitting criterion impacted the generalization performance. Utgoff [1989] proposed ID5R, which enabled incremental learning for decision trees. Geurts et al. [2006] introduced the extremely randomized trees whose structures are independent of the labels of the learning samples. Tree-based ensemble algorithms such as random forest [Breiman, 2001] XGBoost [Chen and Guestrin, 2016], LightGBM [Ke et al., 2017] and deep forest [Zhou and Feng, 2019] are also popular and effective methods.

There are great efforts on the consistency of decision trees. Biau et al. [2008] investigated the connection between decision trees and tree ensemble methods, and they proved that purely randomized trees are consistent. Gao and Zhou [2020]; Gao et al. [2022] then presented the convergence rates of purely randomized trees and a simplified variant of Breiman's original CART [Breiman et al., 1984]. The growth of random trees they analyzed is label-independent, whereas heuristic algorithms for tree learning are usually label-dependent. Scornet et al. [2015] showed that CART is consistent under the assumptions of uniformly distributing features and additive target functions with Gaussian noise. Klusowski [2021] provided universal consistency of CART in the high dimensions assuming the target function is additive too. It is worth mentioning that Blanc et al. [2020b]; Fiat and Pechyony [2004] studied a variant of CART which uses the well-known Influence [Kahn et al., 1988; O'Donnell, 2014] as the impurity measure. Their works are mostly related to ours; they studied the training error of the tree for Boolean functions, while this work focuses on the consistency rate for real-valued feature spaces.

3 PRELIMINARY

Setting. Let $\mathcal{X} = [0,1]^d$ and $\mathcal{Y} = \{0,1\}$ be the feature space and label space, respectively. Let $\mathbf{p} : \mathcal{X} \to \mathbb{R}^+$ be an underlying probability density function on $\mathcal{X}$ and $\eta(\mathbf{x}) = \Pr[Y = 1 \mid \mathbf{X} = \mathbf{x}]$ is the conditional probability function. We observe data $D_n = \{(\mathbf{X}_1, Y_1), (\mathbf{X}_2, Y_2), ..., (\mathbf{X}_n, Y_n)\}$ drawn i.i.d. with $\mathbf{X}_i \sim \mathbf{p}$ and $Y_{i|\mathbf{X}_i} \sim \mathrm{Bernoulli}(\eta(\mathbf{X}_i))$, where $\mathbf{X}_i = (\mathbf{X}_i^{(1)}, \mathbf{X}_i^{(2)}, ..., \mathbf{X}_i^{(d)})$ is a d-dimension vector. Let $R(h) \triangleq \Pr[h(\mathbf{X}) \neq Y]$ denotes the generalization error of hypothesis h. We write the Bayes error and the target function (or Bayes optimal classifier) as $R^\star = \min_{h:\mathcal{X}\to\mathcal{Y}} R(h)$ and $f \in \arg\min_{h:\mathcal{X}\to\mathcal{Y}} R(h)$, respectively. As the underlying distribution $\mathbf{p}$ and the target function f are unknown, we run a learning algorithm and obtain estimators $\hat{\mathbf{p}}$ and $\hat{f}$. For simplicity, we write $\mathbb{E}_{\mathbf{X}\sim\mathbf{p}}$ and $\Pr_{\mathbf{X}\sim\mathbf{p}}$ as $\mathbb{E}_{\mathbf{p}}$ and $\Pr_{\mathbf{p}}$, respectively, so as to $\mathbb{E}_{\hat{\mathbf{p}}}$ and $\Pr_{\hat{\mathbf{p}}}$. We say a distribution $\mathbf{p}$ is a product distribution if $\mathbf{p}_{\mathbf{X}} \equiv \prod_{k=1}^d \mathbf{p}_{\mathbf{X}^{(k)}}$, i.e., $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(d)}$ are independent on each others. 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 $\{\mathbf{x} \mid \mathbf{x}^{(i)} < s_i\}$ or $\{\mathbf{x} \mid \mathbf{x}^{(j)} \geq s_j\}$, where $\mathbf{x}$ is a d-dimension vector in $\mathcal{X}$ and $\mathbf{x}^{(k)}$ is its k-th dimension. For all $\mathbf{x} \in t$, the prediction $T(\mathbf{x})$ is decided as follows:

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

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. The impurity function commonly used in famous heuristic algorithms comprises the entropy, Gini-index, etc. The impurity function provides an efficient and effective way to build a tree greedily. At each step, the learning algorithm chooses a dimension k and a cut point s splitting t into two children $t_L \triangleq t \cap \{\mathbf{x} \mid \mathbf{x}^{(k)} < s\}$ and $t_R \triangleq t \cap \{\mathbf{x} \mid \mathbf{x}^{(k)} \geq s\}$. The choices of k and s maximize the purity gain

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

where $p_t \triangleq \Pr_{\hat{\mathbf{p}}}[\mathbf{X} \in t]$ and $E_t \triangleq \mathbb{E}_{\hat{\mathbf{p}}}\big[\hat{f}(\mathbf{X}) \mid \mathbf{X} \in t\big]$. $w_{t_L} \triangleq p_t / p_{t_L}$ and $w_{t_R} \triangleq p_t / p_{t_R}$ denote the ratios of sample numbers falling into the two new leaf nodes.

Our studies work with the following assumptions.

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

Assumption 2. *The probability density function is continuously differentiable, i.e., the derivative $\mathrm{d}\mathbf{p}_{\mathbf{X}^{(k)}}/\mathrm{d}\mathbf{x}^{(k)}$ is a continuous function for all k.*

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

It is worth mentioning that Assumption 1 and 2 are standard in consistency analysis [Audibert and Tsybakov, 2007]. Assumption 3 looks strong, but it is usually employed by seminal studies [Brutzkus et al., 2020; Kalai and Teng, 2008; Takimoto and Maruoka, 2003], which 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 Blanc et al. [2020a]; Klusowski [2021]; Scornet et al. [2015].

4 DIFFERENCE BETWEEN CART AND INFCART

In this section, we will demonstrate the distinguishing features of InfCART as compared to CART. 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:

$$C_K \triangleq \sum_{t \in \text{leaves}(T_K)} p_t \, \mathcal{G}(t) \ ,$$

where p_t follows the definition in Section 3, and $\mathcal{G}(t) \triangleq G\Big(\mathbb{E}_{\hat{\mathbf{p}}}\big[\hat{f}(\mathbf{X}) \mid \mathbf{X} \in t\big]\Big)$ is the impurity of leaf t. Here, we use $\hat{\mathbf{p}}$ instead of $\mathbf{p}$ because the true $\mathbf{p}$ is 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 $\hat{R}(T_K) \leq C_K$.*

One of the main ingredients of proving consistency is to show that the error converges to zero. Thus, it suffices to ensure a vanishing potential C_K from Proposition 1, which will be detailed in Section 4.2.

4.2 Conventional Impurity Measures Fail

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 $\mathbf{p}$ 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 shows that in the worst case, the potential gain Δ_K can always be zero, leading to a failure of proving a vanishing error, as discussed in Section 4.1. The proof of Proposition 2 is by simply giving a counterexample in which the maximal potential gain is zero but the error is not. More details will be shown in Appendix B.

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

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

where

$$\begin{aligned}\mathbb{E}_{\mathbf{p}^{(j \neq k)}}[\cdot] \triangleq \; & \mathbb{E}_{\mathbf{X}^{(1)} \sim \mathbf{p}^{(1)}} \ldots \mathbb{E}_{\mathbf{X}^{(k-1)} \sim \mathbf{p}^{(k-1)}} \\ & \mathbb{E}_{\mathbf{X}^{(k+1)} \sim \mathbf{p}^{(k+1)}} \ldots \mathbb{E}_{\mathbf{X}^{(d)} \sim \mathbf{p}^{(d)}}[\cdot] \ ,\end{aligned}$$

or equivalently taking expectation over all the coordinates except $\mathbf{X}^{(k)}$ over product distribution $\mathbf{p}$.

It is natural to define the conditional Influence as follows:

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

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

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 re-define $\mathcal{G}(t) \triangleq \mathrm{Inf}_k^G[f \mid t]$ for simplicity. Then, InfCART splits every leaf t by maximizing

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

We compare CART with InfCART in Table 1, 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.

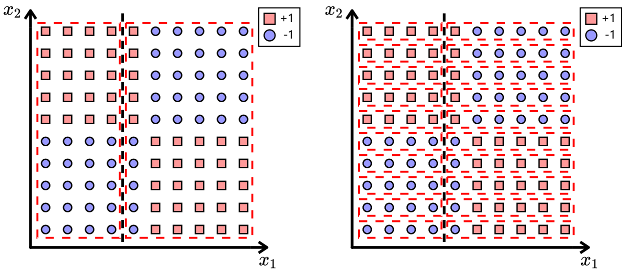
Table 1: Comparison of CART and InfCART

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

Proposition 3. *For any product distribution* $\mathbf{p}$ *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 never fails to obtain purity gain without any assumptions on the target functions [Klusowski, 2021; Scornet et al., 2015], while the latter can not.

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.



(a) CART

(b) InfCART

Figure 1: Comparison of the calculation mechanism of purity gain between CART and InfCART for solving the "XOR" problem. In contrast to the CART that 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.

Unfortunately, InfCART requires an Influence oracle to calculate the purity gain, as the expectations in Definition 2 are taken over the population, which is impractical for real-world tasks. The definition of Influence depends on Assumption 3 or features being independent on each others [Kahn et al., 1988; O'Donnell, 2014]. It is a strong assumption but is still weaker as discussed in Section 3.

Algorithm 1 Histogram Density Estimation (**HDE**)

```
Input: Training dataset $D$, grid size $h$
Output: An estimated probability density function $\hat{\mathbf{p}}_{\mathbf{X}}$
for $k \in [d]$ do
    $\mathcal{B} \leftarrow \{[0,h],[0,2h],\dots[0,1]\}$
    for $b$ in $\mathcal{B}$ do
        $\omega(b) \leftarrow 1/(nh)\sum_{i=1}^{n}\mathbb{I}\{\mathbf{X}_i^{(k)} \in b\}$
    $\hat{\mathbf{p}}_{\mathbf{X}^{(k)}}(x) \leftarrow \omega(b)$, for $x \in b, b \in \mathcal{B}$
$\hat{\mathbf{p}}_{\mathbf{X}} \leftarrow \prod_{k=1}^{d}\hat{\mathbf{p}}_{\mathbf{X}^{(k)}}$
return $\hat{\mathbf{p}}_{\mathbf{X}}$
```

5 GRIDCART: A REFINED CART WITH CONSISTENCY GUARANTEE

This section presents the GridCART, whose key idea is to estimate the underlying distribution so that Influence can be obtained from the estimated distribution. Therefore, GridCART can be a not only practical but also provably consistent heuristic algorithm. It consists of three steps: estimating the feature distribution, estimating the target function, and tree building, which are detailed in Algorithm 1-4.

5.1 Estimate the Feature Distribution

In this step, we venture to estimate a univariate distribution for each dimension, and the estimated density of the joint distribution is the multiplication of them. The key reason is that the notorious curse of dimensionality in non-parametric density estimation [Devroye et al., 1997] may be avoided, as shown in Lemma 4. The process of density estimation for each dimension can be handled by some well-known histogram methods [Devroye et al., 1997; Wasserman, 2006], which is detailed in Algorithm 1.

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

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

Generally, it is hard to estimate the distribution in high dimensions. However, Lemma 4 reduces the d-dimension case to many 1-dimension cases, making the estimation much easier. The total loss can be bounded by the summation of error in each dimension which is linear in d in the worst case but not depends on d exponentially in general cases [Tsybakov, 2009]. Combining it with the error bound of the histogram density estimation, we obtain Lemma 5.

Algorithm 2 Learning Histogram Classifier (**LHC**)

```
Input: Training dataset $D$, grid size $h$
Output: A histogram classifier $g_n^{\text{hist}}$
// Cartesian product of intervals
$\mathfrak{C} \leftarrow \left\{ \times_{k=1}^{d} [m_k h, (m_k+1)h) \mid m_k \in [1/h], \forall j \right\}$
for $\mathcal{C}$ in $\mathfrak{C}$ do
    $\hat{\eta}(\mathcal{C}) \leftarrow \frac{\sum_{i=1}^{n} \mathbb{I}\{\mathbf{X}_i \in \mathcal{C}\} Y_i}{\sum_{i=1}^{n} \mathbb{I}\{\mathbf{X}_i \in \mathcal{C}\}}$
$g_n^{\text{hist}}(x) \leftarrow \mathbb{I}\{\hat{\eta}(c) > 1/2\}$, for $x \in \mathcal{C}, \mathcal{C} \in \mathfrak{C}$
return $g_n^{\text{hist}}$
```

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

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

Lemma 5 provides an error bound for the density estimator in Algorithm 1. Based on this lemma, we have the expected $\mathbb{L}_1$-error has order $\mathcal{O}(n^{-1/3})$, by selecting the window 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 Estimate 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. 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 $\mathbf{1}_{1\times d}$ is neglected for simplicity. Let $\mathcal{C}(\mathbf{x})$ be the cube that contains $\mathbf{x}$ for any $\mathbf{x} \in \mathcal{X}$, then we can define

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

as the estimated conditional probability. If no samples are falling into the cube $\mathcal{C}(\mathbf{x})$, we choose a uniformly random label as the output. The histogram classifier g_n^{hist} can eventually be defined by

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

or equivalently predict the label by voting in every cube. Then we have the following conclusion.

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

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

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

Algorithm 3 Tree Building (**TB**)

```
Input: A histogram density estimator $\hat{\mathbf{p}}$, a histogram classifier $g_n^{\text{hist}}$, maximal tree depth $K_{\max}$, grid size $h$
Output: A learned tree $T_{K_{\max}} : \mathbb{R}^d \to \{0,1\}$
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
Leaves set $\mathcal{L} \leftarrow \{[0,1]^d\}$
for $K = 1 \dots K_{\max}$ do
    $\mathcal{L}' \leftarrow \mathcal{L}$ // Copy a leaves set
    for $t \in \mathcal{L}'$ do
        if All the samples in $t$ have the same label then
            continue
        else
            $(k_0, s_0) \leftarrow \arg\max_{k,s} \Delta^{\text{Inf}}(t, k, s)$
            $t_L \leftarrow t \cap \{\mathbf{x} \mid \mathbf{x}^{(k_0)} \leq s_0\}$
            $t_R \leftarrow t \cap \{\mathbf{x} \mid \mathbf{x}^{(k_0)} > s_0\}$
            $\mathcal{L} \leftarrow (\mathcal{L} - t) \cup t_L \cup t_R$
$T(\mathbf{x}) \leftarrow \arg\max_{y \in \{0,1\}} \sum_{i=1}^{n} \mathbb{I}\{\mathbf{X}_i \in t, Y_i = y\}$, for $\mathbf{x} \in t$
return $T$
```

Lemma 6 shows that the expected excess error of the histogram classifier can converge to zero in order related to n and h_n. Intuitively, as sample size $n \to \infty$ and $h_n \to 0$, the number of cubes $|\mathcal{J}| \to \infty$, which helps capture more detail in feature space; and the number of samples dropping into every cube $|\{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_n\} \cap \mathcal{C}| \to \infty$ in probability, which makes the voting more and more accurate.

5.3 Tree Constructing

With estimator $\hat{\mathbf{p}}$ and g_n^{hist} at hand, we can build a tree by running Algorithm 3.

To show the convergence guarantee of the error, we begin with the following definition:

Definition 3 (N-piece function). *A function f is called an N-piece function if $f_k(z)$ is a piecewise constant function with at most N pieces for any $k \in [d]$, where $f_k(z) \triangleq f\left(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(k-1)}, z, \mathbf{x}^{(k+1)}, \dots, \mathbf{x}^{(d)}\right)$ with $\mathbf{x} \in \mathbb{R}^d$.*

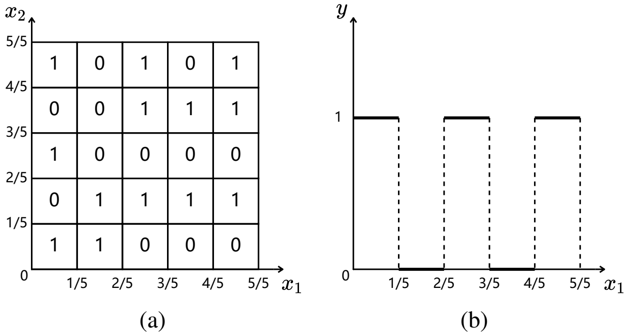
Definition 3 is an extension of the piecewise-constant function in univariate cases. An example of a 5-piece function is given in Figure 2, in which given either x_1 or x_2, the univariate function is piece-wise constant with 5 pieces. It is not difficult to show that the histogram classifier learned by Algorithm 2 is an $1/h_n$-piece function.

Theorem 7. *Suppose that the target function g is N-piece. Then, under Assumption 1-3, for the tree T generated by*

Algorithm 3 with depth K, we have

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

Theorem 7 shows that Algorithm 3 can fit any product distribution with N-piece function well. 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.



1	0	1	0	1
0	0	1	1	1
1	0	0	0	0
0	1	1	1	1
1	1	0	0	0

(a)

(b)

Figure 2: An example of a 5-piece function from $[0, 1]^2$ to $\{0, 1\}$. Figure 2a presents the graph of the function and Figure 2b shows its projection on $x_2 = 0.9$.

5.4 Consistency Rate of GridCART

Above all, we have introduced all the components of GridCART and the complete process is detailed in Algorithm 4. In this subsection, we will show the consistency rate of GridCART. We start with the following decomposition:

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

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:

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

where term (a) measures how the learned tree fits the histogram classifier, and term (b) measures the difference between the estimated distribution $\hat{\mathbf{p}}$ and the ground truth distribution $\mathbf{p}$. 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 Eq. (1) and Eq. (2), we obtain the consistency rate of GridCART in Theorem 8.

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

```
Input: Training dataset $D = \{(\mathbf{X}_1, Y_1), ..., (\mathbf{X}_n, Y_n)\}$, maximal tree depth $K_{\text{max}}$, grid size $h$
Output: A learned tree $T_{K_{\text{max}}} : \mathbb{R}^d \to \{0, 1\}$
// Estimating the distribution and the target function
for $k = 1 \dots d$ do
    // Learn the distribution $\mathbf{p}_{\mathbf{X}^{(k)}}$ by applying
      histogram density estimation with window size $h$
    $\hat{\mathbf{p}}_{\mathbf{X}^{(k)}} \leftarrow$ HDE$\left(\{\mathbf{X}_1^{(k)}, ..., \mathbf{X}_n^{(k)}\}, h\right)$
$\hat{\mathbf{p}}_{\mathbf{X}} \leftarrow \prod_{k=1}^{d} \hat{\mathbf{p}}_{\mathbf{X}^{(k)}}$

// Learn the histogram classifier with grid size $h$
$g_n^{\text{hist}} \leftarrow$ LHC$(D, h)$

// Build a tree to fit the histogram classifier
$T \leftarrow$ TB$(\hat{\mathbf{p}}_{\mathbf{X}}, g_n^{\text{hist}})$
return $T$
```

Theorem 8 (Consistency rate for binary classification)**.** *Under Assumption 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 $h_n = \Theta\left(n^{-1/(d+2)}\right)$, $K_n = \Omega(n^{4/(d+2)})$, we obtain a consistency rate of order $\mathcal{O}\left(n^{-1/(d+2)}\right)$.

The consistency rate in Theorem 8 is the first nontrivial consistency rate for heuristic tree learning algorithms. The previous consistency rate focused on the randomized tree, which generates independently on the labels, whereas previous works for label-dependent generating relied on strong assumptions and presented no consistency rate. More details for comparison are shown in Table 2. Notice that the parameter K_n in Theorem 8 controls the maximal depth the tree can grow with. Choosing $K_n = \Omega(n^{4/(d+2)})$ reaches the best convergence rate in the theorem, whereas this may be inefficient in running time. Here, we choose $K_n = \omega(\log^3(n))$ and $h_n = \Theta(1/\log n)$ to achieve both an efficient running time and the consistency.

5.5 Discussions about Time Complexity

In this section, we compare the time complexity of CART and our proposed GridCART. Note that the comparison focuses on the complexity of splitting at each leaf node, i.e., we neglect the comparison of the structures of trees, as the

Table 2: Comparison of our results with previous works

Algorithm	**Feature Distribution**	**Target Function**	**Result**
Pure Random Tree	any	any	$\mathcal{O}(n^{-1/(8d+2)})$ [Gao and Zhou, 2020]
Centered Random Tree	any	any	$\mathcal{O}((\ln n/n)^{1/(d+2)})$ [Gao and Zhou, 2020]
CART	uniformly random in $[0,1]^d$	additive model	$o(1)$ [Scornet et al., 2015]
CART	any	additive model	$\mathcal{O}(1/\log n)$ [Klusowski, 2021]
GridCART	product distribution	any	$\mathcal{O}(n^{-1/(d+2)})$ **(our result)**

total time spent in building a tree is a simple summation of the splitting complexity at all leaves.

CART. At each leaf node, CART traverses all the dimensions and sorts the samples falling into the leaf according to the dimension. After that, CART calculates the purity gains for all the possible cuts. The time complexity of traversing all possible cuts is dominated by the complexity of sorting. For each dimension, the sorting may spend a time of order $\mathcal{O}(n \log n)$, resulting in a total time of order $\mathcal{O}(dn \log n)$ at each leaf node.

GridCART. GridCART first estimates the underlying distribution using observed samples with time complexity of order $\mathcal{O}(n)$. When splitting a node, there are d/h_n possible cuts as there are $1/h_n$ possible cuts along each dimension. To calculate the purity gain of a specific cut, we only need one traverse over all the cubes, resulting in time complexity of order $\mathcal{O}(1/h_n^d)$. Therefore, the total complexity of one split is $\mathcal{O}\big(d/h_n^{(d+1)}\big)$. We can achieve $\mathcal{O}\big(n + dn^{(d+1)/(d+2)}\big)$, provided $h_n = \Theta(n^{-1/(d+2)})$ as shown in Theorem 8, being slightly faster than CART.

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

6 BEYOND BINARY CLASSIFICATION

In this section, we extend the proposed GridCART to multi-class classifications and regression settings.

6.1 Multiclass Classification

We consider the multi-class classification, e.g., $\mathcal{Y} = \{1, 2, \ldots, m\}$ for $m \in \mathbb{N}^+$. To begin with, we first define the conditional G-Influence for multi-class classification.

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

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

where we here abbreviate $\mathbb{E}_{\mathbf{p}^{(j \neq k)}} \Big[\mathbb{I}\{f(\mathbf{X}) = c\} \mid \mathbf{X} \in t \Big]$ as $\Pr_{\mathbf{p}^{(j \neq k)}} \big[f(\mathbf{X}) = c \mid t \big]$ for simplicity.

With the definition of Influence at hand, the extension from binary classification to multi-class classification is simple. We first 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 function G, we have

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

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

Similarly, the consistency rate for binary classification can be extended to multi-class classification cases. The key step is to pre-define the Bayes error and the target function, and then the extension is natural, which is detailed in Appendix F for anyone interested.

6.2 Regression

We consider the regression task, e.g., $\mathcal{Y} = [-M, M]$ for some $M > 0$. Similarly, we first formulate the conditional Influence for regression tasks.

Definition 5 (Conditional Influence for regression)**.** *Let $\mathbf{p}$ 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 $\mathbf{X}^{(k)}$ is defined by*

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

where we choose variance as the impurity measure, which is widely used in regression tasks [Breiman et al., 1984].

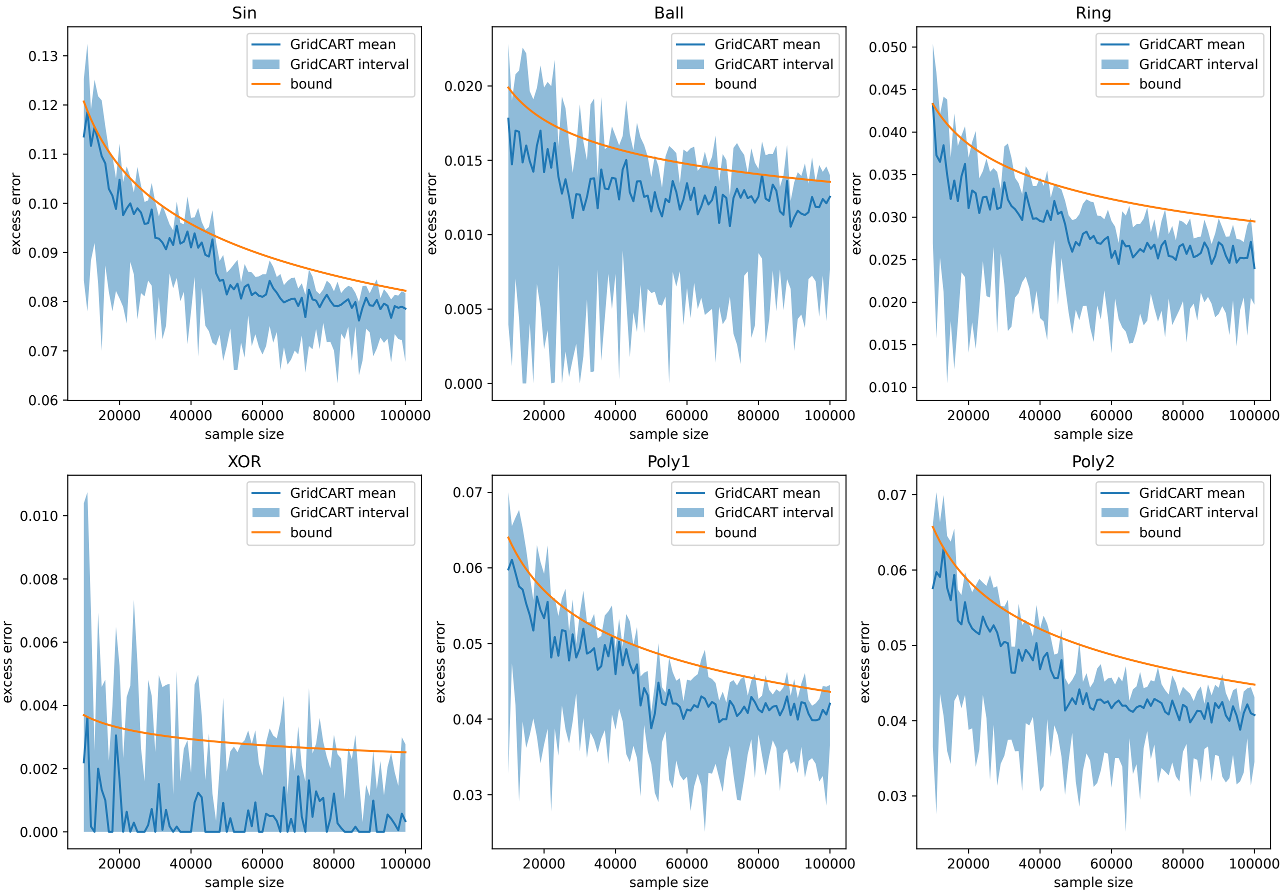


Figure 3: Comparison of excess errors for GridCART trained with increasing training sample sizes and our error bounds. The blue curves are the mean empirical excess errors (test error minus the Bayes error), and the blue regions are filled between the minimal and the maximal values among multiple repeats. 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.

For regression tasks, we replace the criterion in Algorithm 3 [Line 10] as follows:

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

where

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

Besides, the histogram method output $g_n^{\text{hist}} \leftarrow \hat{\eta}(c)$ in Algorithm 2 [Line 5] directly.

Theorem 9 (Consistency rate for regression)**.** *Suppose that Assumption 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:*

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

Theorem 9 shows that GridCART for regression is consistent with order $\mathcal{O}(Mn^{-1/(d+2)})$. In contrast to Theorem 8 for binary classification, Theorem 9 shows that an extra constant M would be suffered for regression tasks, which coincides with the fact that the regression task becomes more difficult if the range of label space $\mathcal{Y}$ is wider.

7 NUMERICAL SIMULATION

To corroborate our theoretical results, we present some numerical simulations. We choose many synthetic datasets and 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 3. 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 3, 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. In the problems of Ball and XOR, the bound curves can not match the blue lines well. This is because our error bound is obtained based on the worst-case analysis, which can be pessimistic in some cases. In other words, the two problems are not difficult for GridCART.

Besides, we also compare the proposed GridCART with the typical CART empirically. Due to space limitations, one can access these materials from Appendix G if interested.

8 CONCLUSIONS

In this work, we first investigated the difficulty in proving the consistency of CART, and then disclosed that no purity gain in the worst case leads to the hardness to obtain a nontrivial result. Motivated by this recognition, we proposed the GridCART, a slightly modified CART, and provided a consistency rate of order $\mathcal{O}(n^{-1/(d+2)})$, which is the first consistency rate for the heuristic tree learning algorithm. Our results shed light on a theoretical understanding of the success of decision tree learning algorithms.

ACKNOWLEDGEMENTS

This research was supported by National Science Foundation of China (62176117, 61921006). The authors would like to thank Tian Qin, Jia-Wei Shan, and Jin-Hui Wu for their constructive comments, and thank the anonymous reviewers for valuable suggestions.

References

Jean-Yves Audibert and Alexandre B. Tsybakov. Fast learning rates for plug-in classifiers. *Annals of Statistics*, 35(2):608 – 633, 2007.

Gérard Biau, Luc Devroye, and Gábor Lugosi. Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 9(66): 2015–2033, 2008.

Guy Blanc, Jane Lange, and Li-Yang Tan. Provable guarantees for decision tree induction: The agnostic setting. In *Proceedings of the 37th International Conference on Machine Learning*, pages 941–949, 2020a.

Guy Blanc, Jane Lange, and Li-Yang Tan. Top-down induction of decision trees: Rigorous guarantees and inherent limitations. In *Proceedings of the 11th Innovations in Theoretical Computer Science Conference*, pages 44:1–44:44, 2020b.

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

Leo Breiman, Jerome Harold Friedman, Richard Allen Oshlen, and Charles Joel Stone. *Classification and Regression Trees*. Chapman & Hall/CRC, 1984.

Carla E. Brodley and Paul E. Utgoff. Multivariate decision trees. *Machine Learning*, 19(1):45–77, 1995.

Alon Brutzkus, Amit Daniely, and Eran Malach. ID3 learns juntas for smoothed product distributions. In *Proceedings of the 33rd Conference on Learning Theory*, pages 902–915, 2020.

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

Luc Devroye, László Györfi, and Gábor Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1997.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

Amos Fiat and Dmitry Pechyony. Decision trees: More theoretical justification for practical algorithms. In *Proceedings of the 15th International Conference on Algorithmic Learning Theory*, pages 156–170, 2004.

Wei Gao and Zhi-Hua Zhou. Towards convergence rate analysis of random forests for classification. In *Advances in Neural Information Processing Systems 33*, pages 9300–9311, 2020.

Wei Gao, Fan Xu, and Zhi-Hua Zhou. Towards convergence rate analysis of random forests for classification. *Artificial Intelligence*, 313:103788, 2022.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006.

Jeff Kahn, Gil Kalai, and Nathan Linial. The influence of variables on boolean functions. In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science*, pages 68–80, 1988.

Adam Tauman Kalai and Shang-Hua Teng. Decision trees are PAC-learnable from most product distributions: A smoothed analysis. *ArXiv preprint*, arXiv:0812.0933, 2008.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems 30*, pages 3149–3157, 2017.

Nathan Keller. On the influences of variables on boolean functions in product spaces. *Combinatorics, Probability and Computing*, 20(1):83–102, 2011.

Jason M. Klusowski. Universal consistency of decision trees in high dimensions. *ArXiv preprint*, arXiv:2104.13881, 2021.

John Mingers. An empirical comparison of selection measures for decision-tree induction. *Machine Learning*, 3(4): 319–342, 1989.

Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

John R. Quinlan. Discovering rules by induction from large collections of examples. *Expert Systems in the Micro Electronics Age*, 1979.

John R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

John R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

Erwan Scornet, Gérard Biau, and Jean-Philippe Vert. Consistency of random forests. *Annals of Statistics*, 43 (4):1716–1741, 2015.

Eiji Takimoto and Akira Maruoka. Top-down decision tree learning as information based boosting. *Theoretical Computer Science*, 292(2):447–464, 2003.

Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009.

Paul E. Utgoff. Incremental induction of decision trees. *Machine Learning*, 4:161–186, 1989.

Larry Wasserman. *All of Nonparametric Statistics*. Springer, 2006.

Zhi-Hua Zhou and Ji Feng. Deep forest. *National Science Review*, 6(1):74–86, 2019.

Instructions for Paper Submissions to AISTATS 2023: Supplementary Materials

This appendix provides the supplementary materials for our AISTATS 2023 work **"On the Consistency Rate of Decision Tree Learning Algorithms"**, constructed according to the corresponding sections therein. The appendix is organized as follows. Appendix A reviews notations used in this paper. Appendix B-F shows the proofs of the main theorems and lemmas in this paper. Appendix G demonstrates experiments not presented in the main paper due to space limitations.

A NOTATIONS

Due to the large number of notations used in this paper, we review most of them in Table 3.

Table 3: Summary of notations

Notation	**Description**
$\mathbf{p}_\mathbf{X}$	The underlying probability density of feature distribution
$\hat{\mathbf{p}}$	The estimated probability density of feature distribution
$\hat{\mathbf{p}}^{(j\neq k)}$	The estimated probability density of all the coordinates except $\mathbf{X}^{(k)}$
$\mathbf{X}$	Random vector in $\mathcal{X} = [0,1]^d$, feature of a sample drawn from distribution $\mathcal{D}$
Y	Random variable in $\mathcal{Y} = \{0,1\}$, label of a sample drawn from distribution $\mathcal{D}$
D_n	Samples set containing n random pairs $(\mathbf{X}_i, Y_i)$
$\mathbf{X}^{(k)}$	The k-th dimension of random vector $\mathbf{X}$
$\mathbf{X}^{(j\neq k)}$	Vector consisting of all the dimensions of $\mathbf{X}$ except the k-th
$\mathbf{x}$	Vector in $\mathcal{X} = [0,1]^d$ without randomness
G	Impurity function $G : [0,1] \to [0,1]$
$\eta(\mathbf{x}), \hat{\eta}(\mathbf{x})$	True and estimated conditional probability function, respectively
$T(\mathbf{x})$	A function $T : [0,1]^d \to \{0,1\}$ which is a tree
leaves(T)	Leaves set of the tree T
t, t_L, t_R	Leaf of a tree, its left child, and its right child, respectively
$R(T)$	The error of a tree T evaluating on the distribution $\mathcal{D}$
h_n	Window size for histogram density estimation, or cube size for histogram classifier
g_n^{hist}	Histogram classifier learned from n i.i.d. samples, a function with randomness
T_K	A tree generated by our proposed GridCART after running for K steps, which is a depth-K tree

B PROOF OF PROPOSITION 2 AND PROPOSITION 3

In this section, we will prove the two propositions given in Section 4. We begin with the strict definition of the impurity function, which was not mentioned 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 mean 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 it reaches the maximal impurity.*
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.

B.1 Proof of Proposition 2

Proposition 2. *There exists a probability density $\mathbf{p}$ 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 $\Delta_K = 0$, even when the error is not zero. The feature distribution of the counterexample $\mathbf{p}_\mathbf{X}$ is a uniform distribution in $[0,1]^2$, and the target function of which is defined as follows:

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

or equivalently the exclusive OR of $\mathbf{x}^{(1)} > 0.5$ and $\mathbf{x}^{(2)} > 0.5$. We let the conditional probability function $\eta(\mathbf{x})$ exactly be f, i.e., the label is with probability one be the target function $f(\mathbf{x})$ for all $\mathbf{x}$. 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.

$$\begin{aligned} \Delta_K &= C_K - C_{K+1} \\ &= \sum_{t \in \text{leaves}(T_K)} p_t \mathcal{G}(t) - \sum_{t \in \text{leaves}(T_{K+1})} p_t \mathcal{G}(t) \\ &= \mathcal{G}\big([0,1]^2\big) - s\, \mathcal{G}\big(t_L\big) - (1-s)\, \mathcal{G}\big(t_R\big) \\ &= G\Big(\mathbb{E}[f \mid \mathbf{X} \in [0,1]^2]\Big) - s\, G\Big(\mathbb{E}\big[f \mid \mathbf{X} \in t_L\big]\Big) - (1-s)\, G\Big(\mathbb{E}\big[f \mid \mathbf{X} \in t_R\big]\Big) \\ &= G(1/2) - s\, G(1/2) - (1-s)\, G(1/2) \qquad (3) \\ &= 0\ , \end{aligned}$$

where we write $\{\mathbf{x} \mid \mathbf{x}^{(k)} < s\}$ and $\{\mathbf{x} \mid \mathbf{x}^{(k)} \geq s\}$ as t_L and t_R, respectively. Eq. (3) is because both the positive and the negative classes have the same ratio, no matter which cutpoint we choose. □

B.2 Proof of Proposition 3

Proposition 3. *For any product distribution $\mathbf{p}$ 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

$$\begin{aligned}
0 = & \max_{k,s} \ \mathrm{Inf}_k \left[[0,1]^2\right] - s\,\mathrm{Inf}_k\left[t_L\right] - (1-s)\,\mathrm{Inf}_k\left[t_R\right] \\
= & \max_{k,s} \ \mathbb{E}_{\mathbf{p}^{(j\neq k)}}\left[G\Big(\mathbb{E}_{\mathbf{p}^{(k)}}\left[f \mid \mathbf{X}\in t\right]\Big)\right] - w_{t_L}\mathbb{E}_{\mathbf{p}^{(j\neq k)}}\left[G\Big(\mathbb{E}_{\mathbf{p}^{(k)}}\left[f \mid \mathbf{X}\in t_L\right]\Big)\right] \\
& \qquad - w_{t_R}\mathbb{E}_{\mathbf{p}^{(j\neq k)}}\left[G\Big(\mathbb{E}_{\mathbf{p}^{(k)}}\left[f \mid \mathbf{X}\in t_R\right]\Big)\right] \\
\geq & \max_{k,s} \ \mathbb{E}_{\mathbf{p}^{(j\neq k)}}\left[\alpha w_{t_L} w_{t_R}\Big(\mathbb{E}_{\mathbf{p}^{(k)}}\left[f \mid \mathbf{X}\in t_L\right] - \mathbb{E}_{\mathbf{p}^{(k)}}\left[f \mid \mathbf{X}\in t_L\right]\Big)^2\right] \qquad (4) \\
\geq & \ 0\,,
\end{aligned}$$

where (4) is by the strong concavity in Definition 6. Then almost surely for all $k \in [d]$ and $\mathbf{x_1}, \mathbf{x_2} \in \mathcal{X}$, once $\mathbf{x}_1^{(j\neq k)} = \mathbf{x}_2^{(j\neq k)}$ then $f(\mathbf{x}_1) = f(\mathbf{x}_2)$, implying that f is constant in every leaf. Then, we have the error $\hat{R}(T) = 0$. □

C PROOF OF LEMMA 5

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

C.1 Proof of Lemma 4

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

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

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

$$\begin{aligned}
\left\|\prod_{k=1}^{d}\hat{\mathbf{p}}_{\mathbf{X}^{(k)}} - \mathbf{p}_{\mathbf{X}}\right\|_1 = & \left\|\prod_{k=1}^{d}\hat{\mathbf{p}}_{\mathbf{X}^{(k)}} - \prod_{k=1}^{d}\mathbf{p}_{\mathbf{X}^{(k)}}\right\|_1 \\
\leq & \left\|\prod_{k=1}^{d}\hat{\mathbf{p}}_{\mathbf{X}^{(k)}} - \mathbf{p}_{\mathbf{X}^{(1)}}\prod_{k=2}^{d}\hat{\mathbf{p}}_{\mathbf{X}^{(k)}}\right\|_1 + \left\|\mathbf{p}_{\mathbf{X}^{(1)}}\prod_{k=2}^{d}\hat{\mathbf{p}}_{\mathbf{X}^{(k)}} - \prod_{k=1}^{d}\mathbf{p}_{\mathbf{X}^{(k)}}\right\|_1 \\
= & \left\|(\hat{\mathbf{p}}_{\mathbf{X}^{(1)}} - \mathbf{p}_{\mathbf{X}^{(1)}})\prod_{k=2}^{d}\hat{\mathbf{p}}_{\mathbf{X}^{(k)}}\right\|_1 + \left\|\mathbf{p}_{\mathbf{X}^{(1)}}(\prod_{k=2}^{d}\hat{\mathbf{p}}_{\mathbf{X}^{(k)}} - \prod_{k=2}^{d}\mathbf{p}_{\mathbf{X}^{(k)}})\right\|_1 \\
\leq & \left\|(\hat{\mathbf{p}}_{\mathbf{X}^{(1)}} - \mathbf{p}_{\mathbf{X}^{(1)}})\prod_{k=2}^{d}\hat{\mathbf{p}}_{\mathbf{X}^{(k)}}\right\|_1 + \left\|\mathbf{p}_{\mathbf{X}^{(1)}}(\hat{\mathbf{p}}_{\mathbf{X}^{(2)}} - \mathbf{p}_{\mathbf{X}^{(2)}})\prod_{k=3}^{d}\hat{\mathbf{p}}_{\mathbf{X}^{(k)}}\right\|_1 \\
& + \left\|\mathbf{p}_{\mathbf{X}^{(1)}}\mathbf{p}_{\mathbf{X}^{(2)}}(\prod_{k=3}^{d}\hat{\mathbf{p}}_{\mathbf{X}^{(k)}} - \prod_{k=3}^{d}\mathbf{p}_{\mathbf{X}^{(k)}})\right\|_1 \\
\leq & \ \dots \\
\leq & \sum_{k=1}^{d}\left\|\prod_{i=1}^{k-1}\mathbf{p}_{\mathbf{X}^{(i)}}(\hat{\mathbf{p}}_{\mathbf{X}^{(k)}} - \mathbf{p}_{\mathbf{X}^{(k)}})\prod_{i=k+1}^{d}\hat{\mathbf{p}}_{\mathbf{X}^{(i)}}\right\|_1\ . \qquad (5)
\end{aligned}$$

By the normalization of probability density, we have

$$\begin{aligned}
&\left\| \prod_{i=1}^{k-1} \mathbf{p}_{\mathbf{X}^{(i)}} (\hat{\mathbf{p}}_{\mathbf{X}^{(k)}} - \mathbf{p}_{\mathbf{X}^{(k)}}) \prod_{i=k+1}^{d} \hat{\mathbf{p}}_{\mathbf{X}^{(i)}} \right\|_1 \\
&= \int \left| \prod_{i=1}^{k-1} \mathbf{p}_{\mathbf{X}^{(i)}} (\hat{\mathbf{p}}_{\mathbf{X}^{(k)}} - \mathbf{p}_{\mathbf{X}^{(k)}}) \prod_{i=k+1}^{d} \hat{\mathbf{p}}_{\mathbf{X}^{(i)}} \right| \mathrm{d}\mathbf{x}^{(1)} \mathrm{d}\mathbf{x}^{(2)} \dots \mathrm{d}\mathbf{x}^{(d)} \\
&= \left(\prod_{i=1}^{k-1} \int \mathbf{p}_{\mathbf{X}^{(i)}} \mathrm{d}\mathbf{x}^{(i)} \right) \int |\hat{\mathbf{p}}_{\mathbf{X}^{(k)}} - \mathbf{p}_{\mathbf{X}^{(k)}}| \, \mathrm{d}\mathbf{x}^{(k)} \left(\prod_{i=k+1}^{d} \int \hat{\mathbf{p}}_{\mathbf{X}^{(i)}} \mathrm{d}\mathbf{x}^{(i)} \right) \\
&\leq 1 \cdot e_k \cdot 1 \\
&\leq e_k \ .
\end{aligned} \tag{6}$$

Combining Eq. (5) and Eq. (6), we then obtain the following bound

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

which completes the proof. □

C.2 Proof of Lemma 5

With Lemma 4 in hand, it suffices to prove that $e_k \leq \mathcal{O}\left(h_n + 1/\sqrt{nh_n}\right)$. We begin with Lemma 10.

Lemma 10 (Wasserman [2006] Theorem 6.11)**.** *Suppose that* $\mathbf{p}'$ *is absolutely continuous and that* $\int (\mathbf{p}'(u))^2 du < \infty$. *Then*

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

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

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

We will then use Lemma 4 and Lemma 10 to prove Lemma 5.

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

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

Proof. Taking the estimators into Lemma 10 we can bound the error at each dimension by

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

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

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

which completes the proof. □

D PROOF OF LEMMA 6

Lemma 11 (Devroye et al. [1997][Corollary 6.1])**.** *If the classifier is defined by whether $\hat{\eta}_n(\mathbf{x}) \leq 1/2$, or equivalently as*

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

Then the excess error of classifier g_n satisfies

$$R(g_n) - R^\star \leq 2 \int |\eta(\mathbf{x}) - \hat{\eta}_n(\mathbf{x})| \, \mathbf{p}(\mathbf{x}) d\mathbf{x} .$$

We will then prove Lemma 6 by using Lemma 11.

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 $\mathbf{p}, \eta$, $n > 0$ we have*

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

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

Proof. Then it suffices to provide the error bound of $|\eta(\mathbf{x}) - \hat{\eta}_n(\mathbf{x})|$. 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:

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

By some simple calculation, we have

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

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

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

We bound term A as follows:

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

where $\mathrm{diam}(\mathcal{C}_j) \triangleq \max_{\mathbf{x},\mathbf{x}'\in\mathcal{C}_j} \|\mathbf{x}-\mathbf{x}'\|_2$, and we use the fact that $\Pr\left[\mathbf{X}\in\mathcal{C}_j\right] = \int_{\mathcal{C}_j}\mathbf{p}(\mathbf{x})\mathrm{d}\mathbf{x}$ 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

$$
\begin{aligned}
&\sum_{j\in\mathcal{J}} \frac{1}{\Pr\left[\mathbf{X}\in\mathcal{C}_j\right]} \int_{\mathcal{C}_j}\int_{\mathcal{C}_j} L\mathrm{diam}(\mathcal{C}_j)\mathbf{p}(\mathbf{x})\mathbf{p}(\mathbf{y})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y}\\
=&\sum_{j\in\mathcal{J}} \frac{1}{\Pr\left[\mathbf{X}\in\mathcal{C}_j\right]} Lh_n\sqrt{d}\Pr\left[\mathbf{X}\in\mathcal{C}_j\right]^2\\
=&Lh_n\sqrt{d}\sum_{j\in\mathcal{J}}\Pr\left[\mathbf{X}\in\mathcal{C}_j\right]\\
=&Lh_n\sqrt{d}\,,
\end{aligned}
\tag{7}
$$

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

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

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 $\Pr[\mathbf{X}\in\mathcal{C}_j]$ and here we use Jensen's inequality again, and consequently we have

$$
\begin{aligned}
\int \mathbb{E}_{D_n}\left[\left|\mathbb{E}_{D_n}\left[\hat{\eta}_n(\mathbf{x})\right]-\hat{\eta}_n(\mathbf{x})\right|\right]\mathbf{p}(\mathbf{x})\mathrm{d}\mathbf{x} &\le \sum_{j\in\mathcal{J}}\sqrt{\mathbb{E}_{D_n}\left[\left|\mathbb{E}_{D_n}\left[\nu_n(\mathcal{C}_j)\right]-\nu_n(\mathcal{C}_j)\right|^2\right]}\\
&\le \sum_{j\in\mathcal{J}}\sqrt{\frac{\Pr\left[\mathbf{X}\in\mathcal{C}_j\right]}{n}}\\
&\le |\mathcal{J}|\frac{1}{|\mathcal{J}|}\sum_{j\in\mathcal{J}}\sqrt{\frac{\Pr\left[\mathbf{X}\in\mathcal{C}_j\right]}{n}}\\
&\le |\mathcal{J}|\sqrt{\frac{1}{|\mathcal{J}|}\sum_{j\in\mathcal{J}}\frac{\Pr\left[\mathbf{X}\in\mathcal{C}_j\right]}{n}}\\
&= \sqrt{\frac{|\mathcal{J}|}{n}}.
\end{aligned}
\tag{8}
$$

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 Eq. (8) we have

$$
\int \mathbb{E}_{D_n}\left[\left|\mathbb{E}_{D_n}\left[\hat{\eta}_n(\mathbf{x})\right]-\hat{\eta}_n(\mathbf{x})\right|\right]\mathbf{p}(\mathbf{x})\mathrm{d}\mathbf{x} \le \sqrt{\frac{1}{nh_n^d}}. \tag{9}
$$

Combining Eq. (7), Eq. (9) and Lemma 11, we obtain the following bound of the excess error

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

□

E PROOF OF THEOREM 7

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

Lemma 12. *Let* $S_i \triangleq |\sum_{j=1}^{i} a_j|$ *we have* $\sum_{i=1}^{n} S_i \geq \frac{1}{2}\sum_{i=1}^{n} |a_i|$.

Proof. It is equivalent to proving that

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

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:

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

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 $b = a_1 + a_2 + \cdots a_k + a_{k+1}$, 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 13. *Suppose that there exists* $C > 0$ *such that* $a_1 \leq C$, $a_n - a_{n+1} \geq \frac{a_n^2}{C}, \forall n \geq 1$, *then we have*

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

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

$$\begin{aligned}
a_{k+1} & \leq a_k\left(1 - \frac{a_k}{C}\right) \\
& \leq \frac{C}{k}\left(1 - \frac{1}{k+1}\right) \\
& \leq \frac{C}{k+1} \ ,
\end{aligned}$$

   which completes the proof.

□

E.1 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.

E.1.1 Step I: Analysis of the 1-dimension Case

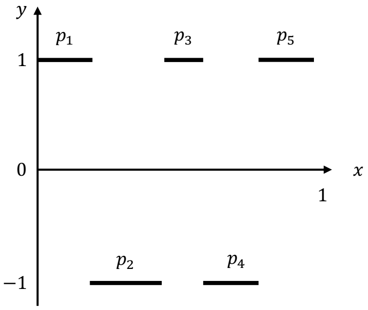


Figure 4: Target function of the 1-dimension toy problem

We begin with the 1-dimension case which is shown in Figure 4. In this case, we define that the 1-dimension feature $X \sim U[0,1]$, which is a uniform distribution in $[0,1]$. We have $0 < p_i < 1$ for all $1 \leq i \leq N$ and $\sum_{i=1}^{N} p_i = 1$. Let $P_i = \sum_{j=1}^{i} p_j$, we define the target function f as follows:

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

which ranges from $\{-1,1\}$. Note that we consider $\mathcal{Y} \in \{-1,1\}$ for simplicity, and it is not difficult to extend the result to $\mathcal{Y} = \{0,1\}$. We can lower bound the purity gain as follows:

$$\begin{aligned} &\max_s \; G\Big(\mathbb{E}\big[f \mid 0 \leq X \leq 1\big]\Big) - s\, G\Big(\mathbb{E}\big[f \mid 0 \leq X < s\big]\Big) - (1-s)\, G\Big(\mathbb{E}\big[f \mid s \leq X \leq 1\big]\Big) \\ \geq\; &\max_s \; \alpha\, s(1-s)\, \Big(\mathbb{E}\big[f \mid 0 \leq X < s\big] - \mathbb{E}\big[f \mid s \leq X \leq 1\big]\Big)^2 \\ \geq\; &\max_k \left(\sum_{i=1}^{k} p_i\right)\left(\sum_{i=k+1}^{N} p_i\right)\left(\frac{\sum_{i=1}^{k}(-1)^{i-1}p_i}{\sum_{i=1}^{k} p_i} - \frac{\sum_{i=k+1}^{N}(-1)^{i-1}p_i}{\sum_{i=k+1}^{N} p_i}\right)^2 . \end{aligned} \tag{10}$$

The second inequality is because the purity gains for splitting at $s \in \mathbb{R}$ is greater than splitting at $k \in \{1, 2, \ldots, N-1\}$, and $\alpha = 1$ for Gini-index. We will then bound Eq. (10) in the following lemma.

Lemma 14. *Solution of optimization problem Eq. (10) has lower bound*

$$\max_k \left(\sum_{i=1}^{k} p_i\right)\left(\sum_{i=k+1}^{N} p_i\right)\left(\frac{\sum_{i=1}^{k}(-1)^{i-1}p_i}{\sum_{i=1}^{k} p_i} - \frac{\sum_{i=k+1}^{N}(-1)^{i-1}p_i}{\sum_{i=k+1}^{N} p_i}\right)^2 \geq \frac{\mathrm{Var}^2[f]}{N(N+2)} .$$

Proof. Let $P_i \triangleq \sum_{i=1}^{s} p_i, Q_i \triangleq \sum_{i=1}^{s} (-1)^{i-1}p_i$, then we can rewrite the Eq. (10) as follows:

$$\begin{aligned}
&\max_k \left(\sum_{i=1}^{k} p_i\right)\left(\sum_{i=k+1}^{N} p_i\right)\left[\frac{\sum_{i=1}^{k}(-1)^{i-1}p_i}{\sum_{i=1}^{k} p_i} - \frac{\sum_{i=k+1}^{N}(-1)^{i-1}p_i}{\sum_{i=k+1}^{N} p_i}\right]^2 \\
=\ &\max_k \left[\frac{Q_k(1-P_k)-(Q_N-Q_k)P_k}{\sqrt{P_k(1-P_k)}}\right]^2 \\
\geq\ &\frac{\sum_{k=1}^{N-1} P_k(1-P_k)\left[\frac{Q_k(1-P_k)-(Q_N-Q_k)P_k}{\sqrt{P_k(1-P_k)}}\right]^2}{\sum_{k=1}^{N-1} P_k(1-P_k)} \\
=\ &\frac{\sum_{k=1}^{N-1}\left[Q_k(1-P_k)-(Q_N-Q_k)P_k\right]^2}{\sum_{k=1}^{N-1} P_k(1-P_k)} \\
=\ &\frac{\sum_{k=1}^{N-1}(Q_k-Q_NP_k)^2}{\sum_{k=1}^{N-1} P_k(1-P_k)} \,.
\end{aligned}$$

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

$$\begin{aligned}
Q_k - Q_NP_k &= \sum_{i=1}^{k}(-1)^{i-1}p_i - \left(\sum_{i=1}^{N}(-1)^{i-1}p_i\right)\sum_{i=1}^{k} p_i \\
&= \sum_{i=1}^{k} p_i\left[(-1)^{i-1} - \mathbb{E}[f]\right] \,. \qquad (11)
\end{aligned}$$

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

$$\sum_{k=1}^{N-1} P_k(1-P_k) \leq \frac{N-1}{4} \,. \tag{12}$$

Combining Eq.(11) and Eq. (12), we have

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

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

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

Using Lemma 12 as well as Jensen's inequality we have

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

Note that the equality is because $\sum_{i=1}^{N} p_i\left[(-1)^{i-1}-\mathbb{E}[f]\right] = 0$. 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.

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

□

E.1.2 Step II: Reduce from d-dimension to 1-dimension

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

Lemma 15 (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*

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

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

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

With Lemma 14 and Lemma 15 at hand, we start proving Theorem 7.

Theorem 7. *Suppose that the target function g is N-piece. Then, under Assumption 1-3, for the tree T generated by Algorithm 3 with depth K, we have*

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

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

$$\epsilon(T) = \mathbb{E}_{\mathbf{X}\sim\hat{\mathbf{p}}}\left[\mathbb{I}\left\{T(\mathbf{X}) \neq g_n^{\text{hist}}(\mathbf{X})\right\}\right] \, .$$

Note that any $\mathbf{x}$ falling into leaf t has the same label, so the error is the minimum of $\Pr_{\mathbf{X}\sim\hat{\mathbf{p}}}[g_n^{\text{hist}}(\mathbf{X}) = 1]$ and $\Pr_{\mathbf{X}\sim\hat{\mathbf{p}}}[g_n^{\text{hist}}(\mathbf{X}) = 0]$. Let $p = \Pr_{\mathbf{X}\sim\hat{\mathbf{p}}}[g_n^{\text{hist}}(\mathbf{X}) = 1]$ and $h \equiv \mathrm{Var}$, by Lemma 15, we have

$$\epsilon(T) = \min(p, 1-p) \leq 2p(1-p) \leq 2B \sum_{k=1}^{d} \mathrm{Inf}_k^{\mathrm{Var}}[g_n^{\text{hist}}] \, , \tag{13}$$

which implies that the error can be bounded by the impurity measure except for a constant. Note that results in Lemma 15 and Eq. (13), which take expectations over $\mathbf{X} \in \mathcal{X}$, are easy to be extended to $\mathbf{X} \in t$. We define the error at leaf t as follows:

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

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

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

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:

$$\begin{aligned} C_K &\triangleq \sum_{t \in T_K} \Pr_{\hat{\mathbf{p}}}[\mathbf{X} \in t] \, \mathcal{G}(t) \\ &= \sum_{t \in T_K} \Pr_{\hat{\mathbf{p}}}[\mathbf{X} \in t] \, \frac{1}{d} \sum_{k=1}^{d} \mathrm{Inf}_k^{\mathrm{Var}}[f \mid t] , \end{aligned} \tag{14}$$

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 15 we have

$$\begin{aligned} \epsilon(T_K) &= \sum_{t \in \text{leaves}(T_K)} \Pr_{\hat{\mathbf{p}}}[\mathbf{X} \in t] \, \epsilon(T \mid t) \\ &\leq 2BdC_K . \end{aligned} \tag{15}$$

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

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

Every leaf $t \in \text{leaves}(T_K)$ is split into two new leaves t_L and t_R the purity gain $\Delta(t,k,s)$ has the following form

$$\Delta(t,k,s) = \Pr_{\hat{\mathbf{p}}}[\mathbf{X} \in t] \, \mathcal{G}(t) - \Pr_{\hat{\mathbf{p}}}[\mathbf{X} \in t_L] \, \mathcal{G}(t_L) - \Pr_{\hat{\mathbf{p}}}[\mathbf{X} \in t_R] \, \mathcal{G}(t_R) ,$$

where $\Pr[\mathbf{X} \in t_L] + \Pr[\mathbf{X} \in t_R] = \Pr[\mathbf{X} \in t]$. The two children of leaf t can be written as $t_L = t \cap \{\mathbf{x} \mid \mathbf{x}^{(k)} < s\}$ and $t_R = t \cap \{\mathbf{x} \mid \mathbf{x}^{(k)} \geq s\}$. At step K, we have a depth-K tree with leaves leaves(T_K). Then, the potential gain is the weighted average summation of maximal purity gains over all the leaves, i.e.,

$$C_K - C_{K+1} = \sum_{t \in \text{leaves}(T_K)} \Pr_{\hat{\mathbf{p}}}[\mathbf{X} \in t] \max_{k,s} \Delta(t,k,s) .$$

For any leaf t we choose a dimension k and cut point s to maximize $\Delta(t,k,s)$, then we have

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

where we frequently use the fact that $\max_{a \in \mathcal{A}} f(a) \geq \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} f(a)$, *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 $|\mathcal{S}_k| = 1/h_n - 1$. Let t_L and t_R be $\{\mathbf{x} \mid \mathbf{x}^{(k)} < s\}$ and $\{\mathbf{x} \mid \mathbf{x}^{(k)} \geq s\}$, respectively, by the strong concavity of variance we have

$$\begin{aligned} \Delta(t,k,s) = &\ \mathrm{Inf}_k^{\mathrm{Var}} \left[g_n^{\text{hist}} \mid t \right] - w_{t_L} \mathrm{Inf}_k^{\mathrm{Var}} \left[g_n^{\text{hist}} \mid t_L \right] - w_{t_R} \mathrm{Inf}_k^{\mathrm{Var}} \left[g_n^{\text{hist}} \mid t_R \right] \\ &\geq \mathbb{E}_{\hat{\mathbf{p}}^{(j \neq k)}} \left[w_{t_L} w_{t_R} \Big(\mathbb{E}_{\hat{\mathbf{p}}^{(k)}} \left[g_n^{\text{hist}} \mid \mathbf{X} \in t_L \right] - \mathbb{E}_{\hat{\mathbf{p}}^{(k)}} \left[g_n^{\text{hist}} \mid \mathbf{X} \in t_R \right] \Big)^2 \right] , \end{aligned}$$

where $p_t = \Pr_{\hat{\mathbf{p}}}[\mathbf{X} \in t]$ equals to the ratio of samples dropping into leaf t, and $\mathcal{S}_k$ is the set of all possible cuts for leaf t at dimension k. As the summation is greater than the maximal, we have

$$\sum_{s\in\mathcal{S}_k} \Delta(t,k,s) \geq \max_{s\in\mathcal{S}_k} \mathbb{E}_{\hat{\mathbf{p}}^{(j\neq k)}} \left[w_{t_L} w_{t_R} \left(\mathbb{E}_{\hat{\mathbf{p}}^{(k)}} \left[g_n^{\text{hist}} \mid \mathbf{X} \in t_L \right] - \mathbb{E}_{\hat{\mathbf{p}}^{(k)}} \left[g_n^{\text{hist}} \mid \mathbf{X} \in t_R \right] \right)^2 \right] ,$$

which has the same form as Eq. (10) and can be bounded using Lemma 14. Specifically, we have

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

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

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

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

$$\begin{aligned} C_K - C_{K+1} &= \sum_{t\in\text{leaves}(T_K)} p_t \max_{k,s} \Delta(t,k,s) \\ &\geq \sum_{t\in\text{leaves}(T_K)} p_t \mathcal{G}(t)^2 \\ &\geq \frac{4}{(N-1)^3} \left(\sum_{t\in\text{leaves}(T_K)} p_t \mathcal{G}(t) \right)^2 \\ &= \frac{4}{(N-1)^3} C_K^2 . \end{aligned} \tag{16}$$

By Lemma 13 and Eq. (16) we can upper bound the potential C_K

$$C_K \leq \frac{(N-1)^3}{4K} . \tag{17}$$

Finally, we combine the upper bounds of the error Eq. (15) and the potential Eq. (17), we have

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

which completes the proof. □

F BEYOND BINARY CLASSIFICATION

In this section, we show more consistency rates for GridCART beyond binary classification.

F.1 Consistency Rate for Multi-class Classification

For multi-class classification, for example $\mathcal{Y} = \{1, 2, \dots, m\}$. To begin with, we redefine Bayes error $R^\star \triangleq \min_{h:\mathcal{X}\to\mathcal{Y}} R(h)$ and Bayes optimal classifier $f \in \underset{h:\mathcal{X}\to\mathcal{Y}}{\arg\max} R(h)$. Then, we need also redefine the conditional probability function $\eta(\mathbf{x}, c) = \Pr[Y = c \mid \mathbf{X} = \mathbf{x}], c \in \mathcal{Y}$, which can recovers the definition $\mathbb{E}[Y \mid \mathbf{X} = \mathbf{x}]$ when $m = 2$. Naturally, we can define the estimated conditional probability function and histogram classifier as follows:

$$\hat{\eta}(\mathbf{x}, c) \triangleq \frac{\sum_{i=1}^n \mathbb{I}\{Y_i = c, \mathbf{X}_i \in \mathcal{C}(\mathbf{x})\}}{\sum_{i=1}^n \mathbb{I}\{\mathbf{X}_i \in \mathcal{C}(\mathbf{x})\}}, \quad g_n^{\text{hist}}(\mathbf{x}) \triangleq \underset{c\in\mathcal{Y}}{\arg\max}\ \hat{\eta}(\mathbf{x}, c)\ . \tag{18}$$

To derive a consistency rate of GridCART for multi-class classification, we begin with the extension of Lemma 11.

Corollary 16 (Extension of Devroye et al. [1997][Corollary 6.1])**.** *Suppose that $\mathcal{Y} = \{1, 2, \dots, m\}$. If the classifier is defined as $g_n(\mathbf{x}) = \underset{c\in\mathcal{Y}}{\arg\max}\ \hat{\eta}(\mathbf{x}, c)$, or equivalently as*

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

Then the excess error of classifier g_n satisfies

$$R(g_n) - R^\star \le \sum_{c\in\mathcal{Y}} \int |\eta(\mathbf{x}, c) - \hat{\eta}_n(\mathbf{x}, c)|\, \mathbf{p}(\mathbf{x}) d\mathbf{x}\ .$$

Corollary 16 bridges the gap between the error for classification and the error for the estimation of the conditional probability function. Note that Corollary 16 recovers Lemma 11 as $\eta(\mathbf{x}, 0) = 1 - \eta(\mathbf{x}, 1)$ when $m = 2$. With the corollary at hand, we then have the following bound of excess error.

Corollary 17. *Suppose that $\mathcal{Y} = \{1, 2, \dots, m\}$. Under Assumption 1, suppose that the histogram rule satisfies $h_n \to 0$ and $nh_n^d \to \infty$ as $n \to \infty$. Then, for any $\mathbf{p}, \eta$, $n > 0$ and histogram classifier defined by Eq. (18), we have*

$$\mathbb{E}\left[R(g_n^{\textit{hist}})\right] - R^\star \le \mathcal{O}\left(m\, n^{-1/(d+2)}\right),$$

provided that $h_n = \Theta(n^{-1/(d+2)})$.

Compared with Lemma 6, an extra constant m would be suffered for Corollary 17, implying that the task is harder when the number of class m is larger.

F.2 Consistency Rate for Regression

Similarly, we define the Bayes error $R^\star \triangleq \underset{h:\mathcal{X}\to\mathcal{Y}}{\min} R(h)$ and the ground-truth regression function $\eta(\mathbf{x}) \triangleq \mathbb{E}[Y \mid \mathbf{X} = \mathbf{x}]$ for regression tasks, where the error $R(h) \triangleq \mathbb{E}\big[Y - h(\mathbf{X})\big]^2$ is the mean square error. Then we can define the regressor

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

which directly outputs $\hat{\eta}$ without taking $\arg\max$ comparing with Eq. (18). Theorem 9 can be proved by following the proofs of Lemma 5, Lemma 6, and Theorem 7, which is omitted here.

G MORE EXPERIMENTS

This section introduces more details about experiments in Section 7 and presents more experiments to verify the effectiveness and efficiency of our proposed GridCART.

G.1 Datasets Introduction

This subsection introduces more details, including descriptions, scales, and so on, about both synthetic and real-world datasets we use in this paper.

G.1.1 Synthetic Datasets

Experiments for synthetic datasets focus on the approximating ability for different target functions, therefore, all of the synthetic datasets consist of 10000 samples with 4-dimension uniformly distributing features in $[0,1]^4$. As we think about the noisy labels setting, we add 5% noise to the labels, *i.e.*, all the labels may flip with probability 0.05. Formally, we have

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

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

Table 4: Formulation of synthetic datasets used in our experiments

Target	**Forumulation**
Sin	$f(\mathbf{x}) = \mathbb{I}\{\mathbf{x}^{(4)} \leq \sin\left[5(\mathbf{x}^{(1)} + \mathbf{x}^{(2)} + \mathbf{x}^{(3)})\right]\}$
Ball	$f(\mathbf{x}) = \mathbb{I}\{(\mathbf{x}^{(1)})^2 + (\mathbf{x}^{(2)})^2 + (\mathbf{x}^{(3)})^2 + (\mathbf{x}^{(4)})^2 \leq 1/2\}$
Ring	$f(\mathbf{x}) = \mathbb{I}\{1/3 \leq (\mathbf{x}^{(1)})^2 + (\mathbf{x}^{(2)})^2 + (\mathbf{x}^{(3)})^2 + (\mathbf{x}^{(4)})^2 \leq 2/3\}$
XOR	$f(\mathbf{x}) = \mathbb{I}\{\mathbf{x}^{(1)} < 0.5\} \otimes \mathbb{I}\{\mathbf{x}^{(2)} < 0.5\} \otimes \mathbb{I}\{\mathbf{x}^{(3)} < 0.5\} \otimes \mathbb{I}\{\mathbf{x}^{(4)} < 0.5\}$
Poly1	$f(\mathbf{x}) = \mathbb{I}\{4(\mathbf{x}^{(1)})^1 + 3(\mathbf{x}^{(2)})^2 + 2(\mathbf{x}^{(3)})^3 + (\mathbf{x}^{(4)})^4 \leq 4\}$
Poly2	$f(\mathbf{x}) = \mathbb{I}\{(\mathbf{x}^{(1)})^4 + 2(\mathbf{x}^{(2)})^3 + 3(\mathbf{x}^{(3)})^2 + 4(\mathbf{x}^{(4)})^1 \leq 4\}$

G.1.2 Real-World Datasets

The real-world datasets we use include Iris[1], Abalone[2], Transfusion[3], Faults[4], Magic[5], and Accelerometer[6]. We rescale the space to $[0,1]^d$ and unify all the tasks to binary classification.

G.2 More Algorithm Details

Histogram density estimation The histogram density estimator and the histogram classifier can be calculated simultaneously for efficiency. As the assumption of product distribution is violated for real-world datasets, we exploit 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 $\{1/N \mid N \in \mathbb{N}^+\}$, *i.e.*, the interval $[0,1]$ is splitted to equal N parts. As the scale of datasets is not large, choosing $N-1$, N, or $N+1$ may have an impact on the performance. Therefore, we choose the best N from $\{N-1, N, N+1\}$ by cross-validation.

G.3 Results

To show the effectiveness of our proposed GridCART, we compare three basic performance measures of the tree learned by CART and GridCART on both synthetic and real-world datasets. The performance measures we choose include the test accuracy, the maximal depth, and the number of leaves. Features for the synthetic dataset are the uniform distribution on $[0,1]^4$, and the labels are given by an underlying target function. CART's max depth is chosen by cross-validation, while the grid size of GridCART is chosen using the result $\Theta(n^{-1/(d+2)})$ in Theorem 8. For real-world datasets, GridCART

[1] https://archive.ics.uci.edu/ml/datasets/Iris

[2] https://archive.ics.uci.edu/ml/datasets/Abalone

[3] https://archive.ics.uci.edu/ml/datasets/Blood+Transfusion+Service+Center

[4] https://archive.ics.uci.edu/ml/datasets/Steel+Plates+Faults

[5] https://archive.ics.uci.edu/ml/datasets/MAGIC+Gamma+Telescope

[6] https://archive.ics.uci.edu/ml/datasets/Accelerometer

directly applies multivariate histogram density estimation as the assumption of product distribution may not hold. We run every combination of algorithm and dataset twenty times and then present the mean and the standard deviation of them, which are shown in Table 5 and Table 6.

It is observed that GridCART can always learn a tree with a smaller depth and number of leaves. Surprisingly, its generalization performance is still highly competitive with CART. It is worth mentioning that CART's hyperparameter, the maximal depth of the tree, is selected by cross-validation, but GirdCART requires only the default one. Besides, GridCART has proven to be consistent under certain conditions, but CART's consistency is still unclear. For real-world data, the assumption of product distribution may not hold, but somehow, it does work. This motivates us to relax the assumption for the analysis of our proposed GridCART in the future.

Table 5: Comparison of GridCART with CART on synthetic datasets. Bold font indicates the algorithm outperforms the other one (higher accuracy, lower depth, or smaller number of leaves)

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

Table 6: Comparison of GridCART with CART on UCI datasets [Dua and Graff, 2017]. Bold font indicates the algorithm outperforms the other one (higher accuracy, lower depth, or smaller number of leaves)

Dataset	Test Accuracy		Depth		Number of Leaves		Time (seconds)	
	CART	GridCART	CART	GridCART	CART	GridCART	CART	GridCART
Iris	**100.0** ± **0.0**	99.7 ± 1.4	**1.0** ± **0.0**	3.6 ± 1.1	**2.0** ± **0.0**	7.8 ± 2.5	**0.01** ± **0.01**	0.03 ± 0.01
Abal.	71.6 ± 1.8	**74.5** ± **1.9**	23.6 ± 3.0	**12.8** ± **1.1**	567.1 ± 13.7	**203.9** ± **20.0**	14.0 ± 3.6	**4.1** ± **1.0**
Tran.	70.9 ± 5.1	**74.7** ± **3.9**	18.2 ± 3.1	**3.5** ± **0.7**	161.1 ± 6.7	**5.8** ± **1.5**	0.40 ± 0.06	**0.10** ± **0.02**
Faul.	**100.0** ± **0.0**	**100.0** ± **0.0**	**6.0** ± **0.0**	**6.0** ± **0.0**	**7.0** ± **0.0**	**7.0** ± **0.0**	11.3 ± 2.9	**5.6** ± **1.4**
Magi.	**81.4** ± **1.1**	78.5 ± 0.8	34.8 ± 2.3	**11.9** ± **1.5**	1588.7 ± 24.9	**118.8** ± **8.1**	646.1 ± 156.5	**18.8** ± **4.7**
Acce.	**87.6** ± **0.2**	70.3 ± 0.5	40.2 ± 1.9	**7.9** ± **0.3**	15561.5 ± 65.0	**13.6** ± **2.1**	261.1 ± 90.8	**21.7** ± **5.9**