Popular decision tree algorithms are provably noise tolerant

Guy Blanc [*1] **Jane Lange** [*2] **Ali Malik** [*1] **Li-Yang Tan** [*1]

Abstract

Using the framework of boosting, we prove that all impurity-based decision tree learning algorithms, including the classic ID3, C4.5, and CART, are highly noise tolerant. Our guarantees hold under the strongest noise model of nasty noise, and we provide near-matching upper and lower bounds on the allowable noise rate. We further show that these algorithms, which are simple and have long been central to everyday machine learning, enjoy provable guarantees in the noisy setting that are unmatched by existing algorithms in the theoretical literature on decision tree learning. Taken together, our results add to an ongoing line of research that seeks to place the empirical success of these practical decision tree algorithms on firm theoretical footing.

1. Introduction

Decision trees have been central to machine learning since its early days. They give a simple way to represent a dataset in a hierarchical and logical manner, and they are perhaps the most canonical example of an intepretable model. They are also quick to evaluate, with evaluation time scaling with the depth of the tree, a quantity that is typically exponentially smaller than the overall number of nodes. Classic decision tree learning algorithms such as ID3 (Quinlan, 1986), C4.5 (Quinlan, 1993), and CART (Breiman et al., 1984), as well as tree-based ensemble methods such as random forests (Breiman, 2001) and XGBoost (Chen & Guestrin, 2016), are therefore standard techniques in the modern machine learning toolkit.

Impurity-based algorithms are a broad class that captures ID3, C4.5, CART, and indeed essentially all decision tree algorithms used in practice. These algorithms build a binary decision tree for a labeled dataset S in a greedy top-down fashion. Each algorithm $\mathcal{A}_\mathcal{G}$ is defined by an impurity function $\mathcal{G} : [0,1] \to [0,1]$ and a class $\mathcal{H}$ of allowable splitting functions. The root of the tree built by $\mathcal{A}_\mathcal{G}$ corresponds to the split of S into S_0 and S_1 by a function $h \in \mathcal{H}$ that maximizes the *purity gain with respect to* $\mathcal{G}$. The left and right subtrees are built by recursing on S_0 and S_1 respectively. We elaborate on this framework in the body of the paper, mentioning for now that the standard algorithms ID3, CART, and C4.5 can all be cast within this framework: ID3 and C4.5 use the binary entropy function $\mathcal{G}(p) = \mathrm{H}_2(p)$ and the associated purity gain is commonly called information gain, whereas CART uses the Gini impurity function $\mathcal{G}(p) = 4p(1-p)$.

Prior work on provable performance guarantees. Motivated by the empirical success of impurity-based decision tree algorithms, a fruitful and ongoing line of work has focused on establishing provable guarantees on their performance in a variety of models and settings (Kearns & Mansour, 1996; Kearns, 1996; Dietterich et al., 1996; Fiat & Pechyony, 2004; Lee, 2009; Blanc et al., 2020b;a; Brutzkus et al., 2019; 2020; Blanc et al., 2021a).

However, these existing results either only hold in the noiseless setting or require strong and stylized assumptions that limit their practical relevance. For example, the recent work of Blanc et al. (2020a) provides guarantees under the assumptions that examples are distributed according to a product distribution, that noise only affects the labels and not the features (i.e. agnostic noise (Haussler, 1992; Kearns et al., 1994)), and that the corrupted labels are monotone in the features.

1.1. Our contributions

Using the framework of boosting, we prove that all impurity-based decision tree algorithms are highly noise tolerant in a fully general setting: our results apply to arbitrary distributions over features and labels, and we consider the strongest noise model, allowing for adversarial corruptions of both features and labels (i.e. the nasty noise model of Bshouty et al. (2002), also known as strong contamination). Equivalently, we give the first formal guarantees on the robustness of impurity-based decision tree algorithms to

[*]Equal contribution [1]Department of Computer Science, Stanford University [2]Department of Computer Science, Massachusetts Institute of Technology. Correspondence to: Guy Blanc <gblanc@stanford.edu>, Jane Lange <jlange@mit.edu>, Ali Malik <malikali@cs.stanford.edu>, Li-Yang Tan <liyang@cs.stanford.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

distributional shifts; we elaborate on this perspective in the body of the paper. We further provide near-matching upper and lower bounds on the allowable noise rate, showing that our analysis is essentially optimal.

Lastly, specializing this result to the setting of product distributions over features, we show that these algorithms enjoy provable guarantees in the noisy setting that are unmatched by existing algorithms in the sizeable theoretical literature on decision tree learning. Most of these algorithms were developed after the invention of ID3, C4.5, and CART in the 1980s, and are more complicated and much less used in practice.

In more detail, our first result is the following:

Theorem 1.1 (Impurity-based algorithms are noise-tolerant boosting algorithms; see Theorem 3.1 for formal version)**.** *For all impurity functions $\mathcal{G}$ and distributions $\mathcal{D}$ over features and labels, w.h.p. over the draw of a sample $\boldsymbol{S}$ from $\mathcal{D}$, if $\mathcal{A}_{\mathcal{G}}$ is trained on an η-nasty-noise corruption $\widehat{\boldsymbol{S}}$ of $\boldsymbol{S}$ where $\eta \leq O(\varepsilon\gamma)$, as long as the internal nodes of the tree are γ-advantage hypotheses, growing the tree to size $\exp(O(1/\gamma^2\varepsilon^2))$ achieves error $\leq \varepsilon$.*

In one of the first papers to study impurity-based decision tree algorithms from a theoretical perspective, Kearns & Mansour (1996) showed that they can be viewed as boosting algorithms. Theorem 1.1 generalizes these results of Kearns & Mansour (1996) to the setting of adversarial noise and shows that these algorithms are in fact highly noise-tolerant boosting algorithms.

Optimality of Theorem 1.1. Next, we show that the quantitative parameters of Theorem 1.1 are essentially optimal. First, even in the noiseless setting it can be seen that there are target functions for which the tree needs to be grown to size $\exp(\Omega(1/\gamma^2))$ to achieve high accuracy; this was already observed in Kearns & Mansour (1996); Freund (1995). Second, we prove that the guarantees of Theorem 1.1 cannot hold for noise rates $\eta \geq \tilde{\Omega}(\varepsilon\gamma)$, even under strong feature and distributional assumptions:

Theorem 1.2 (Near-matching lower bound on noise rate; see Theorem 4.1 for formal version)**.** *Let the feature space be $\mathcal{X} = \{\pm 1\}^d$ and $\eta \geq \tilde{\Omega}(\varepsilon\gamma)$. There is a distribution $\mathcal{D}$ whose marginal over $\mathcal{X}$ is uniform, such that for all impurity functions $\mathcal{G}$, w.h.p. over the draw of a sample $\boldsymbol{S}$ from $\mathcal{D}$, there is an η-nasty-noise corruption $\widehat{\boldsymbol{S}}$ of $\boldsymbol{S}$ such that if $\mathcal{A}_{\mathcal{G}}$ is trained on $\widehat{\boldsymbol{S}}$, even if all the internal nodes of the tree are γ-advantage hypotheses, the tree has to be grown to size $2^{\Omega(d)}$ in order to achieve error $\leq \varepsilon$.*

A weaker lower bound of $\eta \geq \Omega(\sqrt{\gamma})$ for constant ε follows from the work of Dachman-Soled et al. (2015). Theorem 1.2 improves this to the near-optimal $\eta \geq \tilde{\Omega}(\varepsilon\gamma)$. A key ingredient in our proof is the celebrated Kahn-Kalai-Linial theorem (Kahn et al., 1988) from discrete Fourier analysis. We contrast the dimension-independent bound, $\exp(O(1/\gamma^2\varepsilon^2))$, on the size of tree in Theorem 1.1 with the $2^{\Omega(d)}$ lower bound in Theorem 1.2.

Improving on the theoretical state of the art. Specializing Theorem 1.1 to the setting of product distributions over binary features, we further show the following:

Theorem 1.3 (Learning monotone decision trees in the presence of nasty noise; see Theorem 5.4 for formal version)**.** *For any product distribution over $\{\pm 1\}^d$, impurity-based decision tree algorithms learn size-s monotone decision trees in the presence of nasty noise in $\mathrm{poly}(d) \cdot s^{O(\log s)}$ time.*

This problem of learning decision trees in the setting of product distributions has been intensively studied in learning theory (Hancock, 1993; Bshouty, 1993; Kushilevitz & Mansour, 1993; Blum et al., 1994; Jackson & Servedio, 2006; O'Donnell & Servedio, 2007; Gopalan et al., 2008; Lee, 2009; Kalai et al., 2009; Hazan et al., 2018; Chen & Moitra, 2019; Brutzkus et al., 2020; Blanc et al., 2020b; 2021b). Many real-world learning scenarios are naturally monotone in nature, and relatedly, monotonicity is a commonly studied assumption in learning theory.

Prior to our work, the only algorithms provably resilient to nasty noise run in time $d^{O(\log s)}$ (Linial et al., 1993; Kalai et al., 2008a; Blanc et al., 2021c), which is only $\mathrm{poly}(d)$ for constant s. These algorithms do not resemble the impurity-based algorithms used in practice. Theorem 1.3 therefore gives the first $\mathrm{poly}(d)$-time algorithm for any $s = \omega_d(1)$; our running time remains $\mathrm{poly}(d)$ for s as large as $2^{O(\sqrt{\log d})}$. For the weaker model of agnostic noise, recent work of (Blanc et al., 2020a) gives a $\mathrm{poly}(d) \cdot s^{O(\log s)}$ time algorithm.

We discuss other related work in Appendix A.

2. Preliminaries

Notation. We use **boldface** (e.g. $\boldsymbol{x} \sim \mathcal{D}$) to denote random variables. We consider the binary classification setting where we have a distribution $\mathcal{D}_X$ over an arbitrary domain $\mathcal{X}$ and a (randomised) classification function $\mathcal{D}_{Y=1|X} : \mathcal{X} \to [0,1]$. Together, these define a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$. The goal of a learning algorithm is to use i.i.d. samples $(\boldsymbol{x}, \boldsymbol{y}) \sim \mathcal{D}$ to construct a hypothesis $T : \mathcal{X} \to \{0,1\}$ that achieves low error on $\mathcal{D}$, where error is defined as:

$$\mathrm{error}_{\mathcal{D}}[T] \coloneqq \Pr_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}[T(\boldsymbol{x}) \neq \boldsymbol{y}]. \tag{1}$$

We also define $\mu(\mathcal{D}) \coloneqq \Pr_{\mathcal{D}}[\boldsymbol{y} = 1]$, the bias of $\mathcal{D}$ towards the 1-label, and $\varepsilon(\mathcal{D}) \coloneqq \min\{\mu(\mathcal{D}), 1-\mu(\mathcal{D})\}$ to be the error of predicting the majority label on $\mathcal{D}$.

Decision tree hypotheses. We consider binary decision trees $T : \mathcal{X} \to \{0,1\}$ whose internal nodes v are labeled by functions $h_v : \mathcal{X} \to \{0,1\}$ from a class $\mathcal{H}$ of allowable splitting functions. The most standard class of splitting functions is the set of thresholds of a single feature, $h(x) = \mathbb{1}[x_i \geq \theta]$, though our guarantees apply to any arbitrary class $\mathcal{H}$. Each instance $x \in \mathcal{X}$ follows a unique root-to-leaf path in T: at any internal node v, it follows either the left or right branch depending on the result of $h_v(x)$, until a leaf ℓ is reached. The set of leaves $\ell \in \text{leaves}(T)$ therefore form a partition of $\mathcal{X}$. We write $w_{\mathcal{D}}(\ell)$ to denote the probability that $\boldsymbol{x} \sim \mathcal{D}_X$ reaches ℓ and we write $\mathcal{D}_\ell$ to denote $\mathcal{D}$ conditioned on $\boldsymbol{x}$ reaching ℓ. We write $\boldsymbol{\ell} \sim (T, \mathcal{D})$ to denote the draw of a random leaf where each leaf $\ell \in \text{leaves}(T)$ is given weight $w_{\mathcal{D}}(\ell)$.

The prediction at a leaf ℓ is then taken to be the majority label of points that reach that leaf, i.e. $\mathbb{1}[\mu(\mathcal{D}_\ell) \geq \frac{1}{2}]$. The error of T on $\mathcal{D}$ is therefore given by the sum of the error at each leaf incurred by predicting the majority label, weighted by the probability of reaching that leaf:

$$\text{error}_{\mathcal{D}}[T] = \sum_{\ell \in \text{leaves}(T)} w_{\mathcal{D}}(\ell)\varepsilon(\mathcal{D}_\ell) = \mathop{\mathbb{E}}_{\boldsymbol{\ell} \sim (T,\mathcal{D})}[\varepsilon(\mathcal{D}_{\boldsymbol{\ell}})].$$

We will need the notion of *feature influences* in the context of binary features:

Definition 2.1 (Influence)**.** Let $f : \{\pm 1\}^d \to \{\pm 1\}$ be a function and $\mathcal{D}_X = \mathcal{D}_X^{(1)} \times \ldots \times \mathcal{D}_X^{(d)}$ be a production distribution over $\{\pm 1\}^d$. For $i \in [d]$, the *influence of feature i on f*, denoted as $\text{Inf}_i(f)$ is given by the quantity $2\Pr_{\boldsymbol{x} \sim \mathcal{D}_X, \boldsymbol{b} \sim \mathcal{D}_X^{(i)}}[f(\boldsymbol{x}) \neq f(\boldsymbol{x}_{i=\boldsymbol{b}})]$, where $\boldsymbol{x}_{i=\boldsymbol{b}}$ rerandomises the i-th bit of $\boldsymbol{x}$ with a random sample from $\mathcal{D}_X^{(i)}$.

2.1. Impurity-based decision tree learning algorithms

Essentially almost all decision tree learning algorithms used in practice, including the classic and popular ID3, C4.5, and CART, learn decision trees greedily in a top-down manner, using an *impurity function* as a measure of progress.

Definition 2.2 (Impurity function)**.** An impurity function $\mathcal{G} : [0,1] \to [0,1]$ is a concave function that is symmetric around $\frac{1}{2}$ and satisfies $\mathcal{G}(0) = \mathcal{G}(1) = 0$ and $\mathcal{G}(\frac{1}{2}) = 1$.

Definition 2.2 guarantees that $\mathcal{G}(p) \geq \min\{p, 1-p\}$ for all $p \in [0,1]$, allowing us to view $\mathcal{G}(\mu(\mathcal{D}_\ell))$ as an upper bound on $\varepsilon(\mathcal{D}_\ell)$. If we analogously define $\mathcal{G}_{\mathcal{D}}(T) := \mathbb{E}_{\boldsymbol{\ell} \sim \mathcal{D}}[\mathcal{G}(\mu(\mathcal{D}_{\boldsymbol{\ell}}))]$, we get that $\mathcal{G}_{\mathcal{D}}(T)$ is an upper bound on $\text{error}_{\mathcal{D}}[T]$. Common examples of impurity functions include $\mathcal{G}(p) = \text{H}_2(p)$ (binary cross-entropy, used by ID3 and C4.5), $\mathcal{G}(p) = 4p(1-p)$ (Gini impurity function, or simply variance, used by CART), or $\mathcal{G}(p) = 2\sqrt{p(1-p)}$ (introduced and analyzed in (Kearns & Mansour, 1996)). For this paper, we will focus on $\mathcal{G}(p) = 4p(1-p)$ for simplicity, but our results hold generally for all impurity functions that have a second derivative bounded away from 0 (see Remark C.1 in appendix).

Impurity-based decision tree learning algorithms are parameterized by an impurity function $\mathcal{G}$ and a class $\mathcal{H}$ of allowable splitting functions. For a tree T, leaf ℓ, and label function $h \in \mathcal{H}$, we denote $T_{\ell,h}$ to be the extension of T that replaces the leaf ℓ with an internal node that splits on h. At every step, the algorithm loops through all possible leaves and labelling functions $h \in \mathcal{H}$,[1] looking for a potential split that will result in the largest reduction $\mathcal{G}(T) - \mathcal{G}(T_{\ell,h})$ of the impurity function, and hence also (hopefully) the error of the new tree.

Definition 2.3 (Purity gain)**.** Let h be a splitting function, ℓ be a leaf of T, and $\mathcal{D}_\ell$ be $\mathcal{D}$ conditioned on $\boldsymbol{x}$ reaching ℓ. Let ℓ_0 be the leaf of $T_{\ell,h}$ corresponding to $h(\boldsymbol{x}) = 0$, ℓ_1 be the leaf corresponding to $h(\boldsymbol{x}) = 1$, and $\mathcal{D}_{\ell_0}$ and $\mathcal{D}_{\ell_1}$ be their respective conditional distributions.

We define the local drop in $\mathcal{G}$ at this leaf, after splitting with h, as:

$$\Delta_{\mathcal{D}_\ell}(h) := \mathcal{G}(\mu(\mathcal{D}_\ell)) - \Pr_{\mathcal{D}_\ell}[h(\boldsymbol{x}) = 0] \cdot \mathcal{G}(\mu(\mathcal{D}_{\ell_0})) - \Pr_{\mathcal{D}_\ell}[h(\boldsymbol{x}) = 1] \cdot \mathcal{G}(\mu(\mathcal{D}_{\ell_1})).$$

2.2. Impurity-based decision tree algorithms as boosting algorithms

Kearns & Mansour (1996) were the first to analyze impurity-based decision tree learning algorithms from the perspective of *boosting*. Their simple but key insight was that the splitting functions at the internal nodes of the tree can be viewed as *weak hypotheses*, and the decision tree construction as a process of creating a strong learner by combining these weak hypotheses. We recall that standard weak learning assumption from the literature on boosting:

Definition 2.4 (Standard distribution-independent weak learning assumption)**.** Let $f : \mathcal{X} \to \{0,1\}$ be a target function and $\mathcal{H}$ be a class of hypotheses from $\mathcal{X}$ to $\{0,1\}$. For $\gamma > 0$, we say that $\mathcal{H}$ *satisfies the distribution-independent γ-weak learning assumption w.r.t. f* if for all distributions $\mathcal{D}_X$ over $\mathcal{X}$, there exists $h \in \mathcal{H}$ such that $\Pr_{\boldsymbol{x} \sim \mathcal{D}_X}[h(\boldsymbol{x}) \neq f(\boldsymbol{x})] \leq \frac{1}{2} - \gamma$.

Definition 2.4 can be hard to satisfy because of the requirement that there exists a γ-advantage hypothesis *for every distribution $\mathcal{D}_X$* over $\mathcal{X}$. Our analysis will only rely on a milder *distribution-specific* weak learning assumption that

[1] In the context of practical decision tree algorithms, the class of splitting functions $\mathcal{H}$ is usually finite and small. Standard implementations of these popular algorithms, such as in scikit-learn, do a brute force search over hypotheses.

TopDownDT$_{\mathcal{H},\mathcal{G},\mathcal{D}}(t)$:

Given: Size bound t.

Output: Decision tree of size t approximating $\mathcal{D}$, with internal nodes chosen from $\mathcal{H}$.

1. Initialize T to be the empty tree.
2. While size$(T) < t$:
   (a) Let $(\ell^\star, h^\star)$ be set to
   $$\arg\max_{(\ell,h)\in \text{leaves}(T)\times\mathcal{H}} [w_{\mathcal{D}}(\ell)\cdot\Delta_{\mathcal{D}_\ell}(h)]$$
   breaking ties arbitrarily.
   (b) Set $T \leftarrow T_{\ell^*,h^*}$.
3. Label each leaf $\ell \in \text{leaves}(T)$ with value
   $$\mathbb{1}[\mu(\mathcal{D}_\ell) > \tfrac{1}{2}].$$
4. Return T.

Figure 1: The decision tree boosting algorithm for hypothesis class $\mathcal{H}$ and impurity function $\mathcal{G}$ over distribution $\mathcal{D}$. For simplicity, we assume that $\Delta_{\mathcal{D}_\ell}(h)$, $w_{\mathcal{D}}(\ell)$, and $\mu(\mathcal{D}_\ell)$ can be computed exactly. In practice, these quantities can be replaced with empirical estimates from a random sample of $\mathcal{D}$ (see Appendix D for details).

is significantly easier to satisfy. Looking ahead, the mildness of our weak learning assumption will be crucial for our proof of Theorem 1.3. We first need the notion of an induced distribution:

Definition 2.5 (Induced distributions)**.** Let $\mathcal{D}$ be a distribution over $\mathcal{X}\times\{0,1\}$ and $\mathcal{H}$ be a class of hypotheses from $\mathcal{X}$ to $\{0,1\}$. We say that $\mathcal{D}'$ is a distribution *induced by conditioning $\mathcal{D}$ on $\mathcal{H}$* if $\mathcal{D}'$ can be expressed as $\mathcal{D}$ conditioned on $\boldsymbol{x}\sim\mathcal{D}_X$ satisfying $h_1(\boldsymbol{x})\wedge\cdots\wedge h_k(\boldsymbol{x})$ where $h_i\in\mathcal{H}$.

Remark 2.6. Note that all the conditional distributions, $\mathcal{D}_\ell$, at the leaves of a decision tree are induced distributions of $\mathcal{D}$ conditioned on $\mathcal{H}$. Moreover, if $\mathcal{D}$ is such that $\mathcal{D}_X$ is a product distribution and if $\mathcal{H}$ is the class of hypotheses that threshold on a single feature (i.e. $h(x)=\mathbb{1}[x_i\geq\theta]$), then $\mathcal{D}'_X$ remains a product distribution for every distribution $\mathcal{D}'$ that is induced by conditioning $\mathcal{D}$ on $\mathcal{H}$.

Definition 2.7 (Our distribution-specific weak learning assumption)**.** Let $\mathcal{D}$ be a distribution over $\mathcal{X}\times\{0,1\}$ and $\mathcal{H}$ be a class of hypotheses from $\mathcal{X}$ to $\{0,1\}$. For $\gamma>0$, we say *$\mathcal{H}$ satisfies the γ-weak learning assumption w.r.t $\mathcal{D}$* if, for any distribution $\mathcal{D}'$ that is induced by conditioning $\mathcal{D}$ on $\mathcal{H}$, there exists an $h\in\mathcal{H}$ that satisfies:

$$|\text{Cov}_{\mathcal{D}'}[h(\boldsymbol{x}),\boldsymbol{y}]| \geq \gamma \,\text{Var}_{\mathcal{D}'}[\boldsymbol{y}]. \tag{2}$$

We call such an h a γ-advantage hypothesis with respect to $\mathcal{D}'$.

Remark 2.8. Our weak learning assumption in Definition 2.7 is a weaker assumption than the standard Definition 2.4. This is because Equation (2) is equivalent to having $\Pr_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}'_{\text{bal}}}[h(\boldsymbol{x})\neq\boldsymbol{y}]\leq 1/2-\gamma$, where $\mathcal{D}'_{\text{bal}}$ is the *balanced* version of $\mathcal{D}'$, so that the points in $\mathcal{X}$ are reweighted to make the probability of a positive or negative label equally likely. Since Definition 2.4 has to hold for *all* distributions over $\mathcal{X}$, it also has to hold in particular for the marginal distribution of $\mathcal{D}'_{\text{bal}}$ over $\mathcal{X}$. Therefore, Definition 2.4 implies Definition 2.7.

2.3. Learning with adversarial noise

Adversarial noise can take on many strengths and forms, depending on both what kind of corruptions are allowed and when these corruptions can be made. We focus on the strongest model of noise called *nasty noise* (Bshouty et al., 2002). In this setting, we wish to learn a binary classifier on a distribution $\mathcal{D}$ over $\mathcal{X}\times\{0,1\}$. However, instead of receiving a set of samples $S\sim\mathcal{D}^n$, an adversary is allowed to replace any η-fraction of points in S with arbitrary points to get a corrupted sample $\widehat{S}$. The algorithm then receives the corrupted sample $\widehat{S}$.

This noise model captures many other weaker forms of noise. For example, if the adversary can only change the *labels* of the η corrupted fraction, we recover agnostic noise. Similarly, if the adversary has to obliviously commit to a corruption strategy before seeing the sample S, this is equivalent to choosing a distribution $\widehat{\mathcal{D}}$ that is η-close to $\mathcal{D}$ in Total Variation (TV) distance, and drawing $\widehat{S}$ from that. This is often called the distributional shift setting.

3. Proof of Theorem 1.1: Impurity-based decision tree algorithms are noise-tolerant boosting algorithms

We can now state the formal version of Theorem 1.1, which states that impurity-based decision tree learning algorithms are boosting algorithms that are resilient to nasty noise.

We draw on a recent result by Blanc et al. (2022), which shows that learning in the presence of η-nasty-noise corruption is equivalent to learning in the presence of η distribution shift with respect to Total Variation distance (dist_{TV}), as long as the learner only interacts with its samples by computing expectations. For details on the formal relationship between Theorems 1.1 and 3.1, as well as the runtime analysis of TopDownDT see Appendix D.

Theorem 3.1 (Formal version of Theorem 1.1)**.** *Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \{0,1\}$ and $\mathcal{H}$ be a class of splitting functions from $\mathcal{X}$ to $\{0,1\}$ that satisfies the γ-weak learning assumption w.r.t. $\mathcal{D}$. For any noise rate $\eta \le O(\varepsilon\gamma)$ and any distribution $\widehat{\mathcal{D}}$ satisfying $\mathrm{dist}_{\mathrm{TV}}(\mathcal{D}, \widehat{\mathcal{D}}) \le \eta$, the decision tree hypothesis T constructed by $\textsc{TopDownDT}_{\mathcal{H},\mathcal{G},\widehat{\mathcal{D}}}(t)$ achieves $\mathrm{error}_{\mathcal{D}}[T] \le \varepsilon$ after $t \ge \exp\left(O(1/\gamma^2\varepsilon^2)\right)$.*

We now prove a few lemmas that will be useful for our proof of Theorem 3.1. We begin by quantifying how much the impurity function decreases when we split at this particular leaf with a splitting function $h : \mathcal{X} \to \{0,1\}$. The following lemma lower bounds this decrease, $\Delta_{\widehat{\mathcal{D}}_\ell}(h)$, in terms of the covariance between $h(\boldsymbol{x})$ and $\boldsymbol{y}$. We state the lemma generally for an arbitrary distribution $\mathcal{E}$ since we will be applying it to the distributions, $\widehat{\mathcal{D}}_\ell$, at each leaf.

Lemma 3.2 (Local drop in $\mathcal{G}$ in terms of covariance)**.** *Let $\mathcal{E}$ be any distribution over $\mathcal{X} \times \{0,1\}$ and $h : \mathcal{X} \to \{0,1\}$ be a splitting function. Then:*

$$\Delta_{\mathcal{E}}(h) \ge 16 \cdot \mathrm{Cov}_{\mathcal{E}}[h(\boldsymbol{x}), \boldsymbol{y}]^2. \tag{3}$$

Proof. This result follows almost directly from Kearns & Mansour (1996). See Appendix G for details. □

Our weak learning assumption (Definition 2.7) provides lower bounds on the covariance between $h(\boldsymbol{x})$ and $\boldsymbol{y}$ on the *uncorrupted* distribution $\mathcal{D}_\ell$. We want to relate this this to the covariance on an adversarially corrupted distribution $\widehat{\mathcal{D}}_\ell$ that is η_ℓ-close to $\mathcal{D}_\ell$.

We will need the following useful facts relating the variance and covariance of bounded functions on two distributions that are close in TV distance. We consider an arbitrary domain $\mathcal{V}$ and, without loss of generality, functions from $\mathcal{V}$ to $[0,1]$. We defer the proofs to Appendix B.

Lemma 3.3 (Moments and TV-distance)**.** *Let $\mathcal{E}, \widehat{\mathcal{E}}$ be two distributions over a domain $\mathcal{V}$ with $\mathrm{dist}_{\mathrm{TV}}(\mathcal{E}, \widehat{\mathcal{E}}) \le \eta$ and let $f, g : \mathcal{V} \to [0,1]$ be functions. Then*

$$|\operatorname{Var}_{\mathcal{E}}[f(\boldsymbol{x})] - \operatorname{Var}_{\widehat{\mathcal{E}}}[f(\boldsymbol{x})]| \le \eta \tag{4}$$
$$|\operatorname{Cov}_{\mathcal{E}}[f(\boldsymbol{x}), g(\boldsymbol{x})] - \operatorname{Cov}_{\widehat{\mathcal{E}}}[f(\boldsymbol{x}), g(\boldsymbol{x})]| \le 2\eta. \tag{5}$$

We now use Lemma 3.3 with our weak learning assumption to prove the existence of an h at that has high covariance on the adversarially corrupted distribution at a given leaf:

Lemma 3.4 (Covariance on the corrupted distribution)**.** *Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \{0,1\}$ and $\mathcal{H}$ be a class of splitting functions from $\mathcal{X}$ to $\{0,1\}$ that satisfies the γ-weak learning assumption w.r.t $\mathcal{D}$. For a decision tree T and leaf ℓ of T, let $\widehat{\mathcal{D}}_\ell$ be any distribution with $\mathrm{dist}_{\mathrm{TV}}(\mathcal{D}_\ell, \widehat{\mathcal{D}}_\ell) \le \eta_\ell$. Then there is an $h_\ell \in \mathcal{H}$ s.t.*

$$|\operatorname{Cov}_{\widehat{\mathcal{D}}_\ell}[h_\ell(\boldsymbol{x}), \boldsymbol{y}]| \ge \gamma \operatorname{Var}_{\widehat{\mathcal{D}}_\ell}[\boldsymbol{y}] - 3\eta_\ell.$$

Proof. Since $\mathcal{H}$ satisfies the weak learning assumption w.r.t. $\mathcal{D}$, and since $\mathcal{D}_\ell$ is an induced distribution of $\mathcal{D}$, there exists an $h_\ell \in \mathcal{H}$ s.t. $|\operatorname{Cov}_{\mathcal{D}_\ell}[h_\ell(\boldsymbol{x}), \boldsymbol{y}]| \ge \gamma \operatorname{Var}_{\mathcal{D}_\ell}[\boldsymbol{y}]$.

By Lemma 3.3, since $\mathrm{dist}_{\mathrm{TV}}(\mathcal{D}_\ell, \widehat{\mathcal{D}}_\ell) \le \eta_\ell$, we have:

$$\begin{aligned}
|\operatorname{Cov}_{\widehat{\mathcal{D}}_\ell}[h_\ell(\boldsymbol{x}), \boldsymbol{y}]| &\ge |\operatorname{Cov}_{\mathcal{D}_\ell}[h_\ell(\boldsymbol{x}), \boldsymbol{y}]| - 2\eta_\ell && \text{(Lemma 3.3)} \\
&\ge \gamma \operatorname{Var}_{\mathcal{D}_\ell}[\boldsymbol{y}] - 2\eta_\ell && \text{(Weak learning assumption)} \\
&\ge \gamma \operatorname{Var}_{\widehat{\mathcal{D}}_\ell}[\boldsymbol{y}] - 3\eta_\ell. && \text{(Lemma 3.3)}
\end{aligned}$$

□

We can now prove the theorem:

Proof of Theorem 3.1. If $\mathrm{error}_{\widehat{\mathcal{D}}}[T] < \frac{12\eta}{\gamma} + \varepsilon$, we are done since it follows that $\mathrm{error}_{\mathcal{D}}[T] < \eta + \frac{12\eta}{\gamma} + \varepsilon \le O(\varepsilon)$, by our assumption that $\eta \le O(\varepsilon\gamma)$.

Otherwise, if $\mathrm{error}_{\widehat{\mathcal{D}}}[T] \ge \frac{12\eta}{\gamma} + \varepsilon$, we prove the existence of a leaf $\ell^* \in T$ and splitting function $h_{\ell^*} \in \mathcal{H}$ such that splitting ℓ^* according to h_{ℓ^*} results in a substantial reduction in $\mathcal{G}_{\widehat{\mathcal{D}}}(T)$. In more detail, we consider the expected reduction in $\mathcal{G}_{\widehat{\mathcal{D}}}(T)$ if a *random* leaf $\boldsymbol{\ell} \sim (T, \widehat{\mathcal{D}})$ is split with the respective $h_{\boldsymbol{\ell}}$ from Lemma 3.4:

$$\begin{aligned}
\mathop{\mathbb{E}}_{\boldsymbol{\ell}\sim(T,\widehat{\mathcal{D}})}[\Delta_{\widehat{\mathcal{D}}_{\boldsymbol{\ell}}}(h_{\boldsymbol{\ell}})] &\ge 16 \cdot \mathop{\mathbb{E}}_{\boldsymbol{\ell}\sim(T,\widehat{\mathcal{D}})}\left[\operatorname{Cov}_{\widehat{\mathcal{D}}_{\boldsymbol{\ell}}}[h_{\boldsymbol{\ell}}(\boldsymbol{x}), \boldsymbol{y}]^2\right] && \text{(Lemma 3.2)} \\
&\ge 16 \cdot \mathbb{E}_{\boldsymbol{\ell}\sim(T,\widehat{\mathcal{D}})}\left[|\operatorname{Cov}_{\widehat{\mathcal{D}}_{\boldsymbol{\ell}}}[h_{\boldsymbol{\ell}}(\boldsymbol{x}), \boldsymbol{y}]|\right]^2 && \text{(Jensen's inequality)} \\
&\ge 16 \cdot \left(\mathbb{E}_{\boldsymbol{\ell}\sim(T,\widehat{\mathcal{D}})}\left[|\operatorname{Cov}_{\widehat{\mathcal{D}}_{\boldsymbol{\ell}}}[h_{\boldsymbol{\ell}}(\boldsymbol{x}), \boldsymbol{y}]|\right]\right)_+^2 && \text{(Since } x^2 \ge (x)_+^2\text{)} \\
&\ge 16 \cdot \left(\mathbb{E}_{\boldsymbol{\ell}\sim(T,\widehat{\mathcal{D}})}\left[\gamma \operatorname{Var}_{\widehat{\mathcal{D}}_{\boldsymbol{\ell}}}[\boldsymbol{y}] - 3\eta_{\boldsymbol{\ell}}\right]\right)_+^2 && \text{(Lemma 3.4)} \\
&= 16 \cdot \left(\gamma \mathbb{E}_{\widehat{\mathcal{D}}}[\operatorname{Var}_{\widehat{\mathcal{D}}_{\boldsymbol{\ell}}}[\boldsymbol{y}]] - 3\mathbb{E}_{\widehat{\mathcal{D}}}[\eta_{\boldsymbol{\ell}}]\right)_+^2, && \text{(Linearity of expectation)}
\end{aligned}$$

where we use the notation $(x)_+ \coloneqq \max\{0, x\}$.

Since $\operatorname{Var}_{\widehat{\mathcal{D}}_{\boldsymbol{\ell}}}[\boldsymbol{y}] = \mu(\widehat{\mathcal{D}}_{\boldsymbol{\ell}})(1 - \mu(\widehat{\mathcal{D}}_{\boldsymbol{\ell}}))$, it is clear that $\operatorname{Var}_{\widehat{\mathcal{D}}_{\boldsymbol{\ell}}}[\boldsymbol{y}] \ge 1/2 \cdot \varepsilon(\widehat{\mathcal{D}}_{\boldsymbol{\ell}})$. Therefore

$$\mathop{\mathbb{E}}_{\boldsymbol{\ell}\sim(T,\widehat{\mathcal{D}})}[\operatorname{Var}_{\widehat{\mathcal{D}}_{\boldsymbol{\ell}}}[\boldsymbol{y}]] \ge \mathrm{error}_{\widehat{\mathcal{D}}}[T]/2. \tag{6}$$

All together, we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{\boldsymbol{\ell}\sim(T,\widehat{\mathcal{D}})}[\Delta_{\widehat{\mathcal{D}}_{\boldsymbol{\ell}}}(h_{\boldsymbol{\ell}})] &\geq 16\cdot\left(\gamma\,\mathbb{E}_{\widehat{\mathcal{D}}}[\mathrm{Var}_{\widehat{\mathcal{D}}_{\boldsymbol{\ell}}}[\boldsymbol{y}]]-3\,\mathbb{E}_{\widehat{\mathcal{D}}}[\eta_{\boldsymbol{\ell}}]\right)_+^2\\
&\geq 16\cdot\left(\frac{\gamma}{2}\cdot\mathrm{error}_{\widehat{\mathcal{D}}}[T]-3\,\mathbb{E}_{\widehat{\mathcal{D}}}[\eta_{\boldsymbol{\ell}}]\right)_+^2 && \text{(Equation (6))}\\
&\geq 16\cdot\left(\frac{\gamma}{2}\cdot\mathrm{error}_{\widehat{\mathcal{D}}}[T]-6\eta\right)_+^2 && (\mathbb{E}_{\widehat{\mathcal{D}}}[\eta_{\boldsymbol{\ell}}]\leq 2\eta \text{ by Lemma B.4})\\
&\geq 16\cdot\left(\frac{\gamma\varepsilon}{2}+6\eta-6\eta\right)_+^2 && \text{(by assumption)}\\
&\geq 4\gamma^2\varepsilon^2.
\end{aligned}$$

Rewriting the expectation, we get

$$\sum_{\ell\in T} w_{\widehat{\mathcal{D}}}(\ell^*)\Delta_{\widehat{\mathcal{D}}_{\ell^*}}(h_{\ell^*})\geq 4\gamma^2\varepsilon^2.$$

If there are currently s leaves in T, there must exist a leaf $\ell^\star$ such that

$$w_{\widehat{\mathcal{D}}}(\ell^*)\Delta_{\widehat{\mathcal{D}}_{\ell^*}}(h_{\ell^*})\geq\frac{4\gamma^2\varepsilon^2}{s}.$$

Since TopDownDT greedily splits the leaf that results in the largest drop in $\mathcal{G}_{\widehat{\mathcal{D}}}(T)$, it follows that the drop in $\mathcal{G}$ at timestep s is at least least $4\gamma^2\varepsilon^2/s$. Therefore, after t steps, the total drop is at least:

$$4\gamma^2\varepsilon^2\left(1+\frac{1}{2}+\frac{1}{3}+\ldots+\frac{1}{t}\right)\geq 4\gamma^2\varepsilon^2\log t.$$

Since the range of $\mathcal{G}_{\widehat{\mathcal{D}}}$ is $[0,1]$, we conclude that after $t=\exp(O(1/\gamma^2\varepsilon^2))$ steps, we must be done i.e. $\mathrm{error}_{\widehat{\mathcal{D}}}[T]<\frac{12\eta}{\gamma}+\varepsilon$, and hence $\mathrm{error}_{\mathcal{D}}[T]\leq O(\varepsilon)$, since $\eta\leq O(\varepsilon\gamma)$. □

4. Proof of Theorem 1.2: Optimality of our parameters

Theorem 3.1 says that TopDownDT can grow a tree with error $\leq\varepsilon$ only when $\eta\leq O(\varepsilon\gamma)$, where η is the amount of corruption and γ is the weak learning advantage. Here, we show that this bound is tight: If we allow $\eta=\tilde{O}(\varepsilon\gamma)$, then we can design distributions $\widehat{\mathcal{D}}$ on which TopDownDT fails to achieve error $\leq\varepsilon$.

We remark that for technical convenience, in both this section and Section 5, we switch to functions outputting $\{\pm1\}$ rather than $\{0,1\}$. The two formulations are equivalent.

Theorem 4.1 (Formal version of Theorem 1.2)**.** *For any $\varepsilon,\gamma>0$ where $\gamma^{1/\gamma}\leq\varepsilon$, $d\in\mathbb{N}$, and $\eta\geq\Omega(\gamma\varepsilon\log(1/\varepsilon))$. There is a distribution $\mathcal{D}$ whose marginal over $\mathcal{X}:=\{\pm1\}^d$ is uniform, an η-nasty noise corruption $\widehat{\mathcal{D}}$ of $\mathcal{D}$, and a hypothesis class, $\mathcal{H}$, satisfying the γ-weak learning assumption w.r.t $\mathcal{D}$, such that for all impurity function $\mathcal{G}$, $\textsc{TopDownDT}_{\mathcal{H},\mathcal{G},\widehat{\mathcal{D}}}(t)$ fails to build an ε-error tree for $\mathcal{D}$ unless $t\geq 2^{d-O(\log(1/\gamma)/\gamma)}$.*

Proof sketch. We will design a function $f:\{\pm1\}^d\to\{\pm1\}$ that only depends on its first k features (meaning $f(x)=g(x_{[1:k]})$ for some function $g:\{\pm1\}^k\to\{\pm1\}$) for $k\ll d$ and set $\mathcal{D}$ to the distribution of $(\boldsymbol{x},f(\boldsymbol{x}))$ where $\boldsymbol{x}$ is uniform from $\{\pm1\}^d$. This function will be carefully designed so that there is an η-corruption $\widehat{\mathcal{D}}$ of $\mathcal{D}$ in which *every* hypothesis $h\in\mathcal{H}$ has a local drop in $\mathcal{G}$ of 0. As a result, TopDownDT cannot identify the "important" k hypotheses and is likely to pick one of the $d-k$ useless (because they are independent of the label) hypothesis. That continues until all $d-k$ useless hypotheses have been used, which requires the tree to have depth $d-k$ corresponding to a size of 2^{d-k}.

To formalize the above proof sketch, we will need to prove that the $\mathcal{D}$ we design satisfies the weak-learning hypothesis. To do so, we use the celebrated Kahn–Kalai–Linial inequality from the analysis of Boolean functions.

Fact 4.2 (KKL inequality, (Kahn et al., 1988))**.** *For any function $f:\{\pm1\}^k\to\{\pm1\}$ and $\mathcal{D}_X$ the uniform distribution over $\{\pm1\}^k$, there is a coordinate $i\in[k]$ for which*

$$\mathrm{Inf}_i(f)\geq\Omega\left(\frac{\log k}{k}\cdot\mathop{\mathrm{Var}}_{\boldsymbol{x}\sim\mathcal{D}_X}[f(\boldsymbol{x})]\right).$$

The KKL inequality will allow us to prove that a broad class of distributions satisfy the weak-learning assumption.

Definition 4.3 (Monotone functions)**.** We say that a Boolean function $f:\{\pm1\}^n\to\{\pm1\}$ is *monotone* if for any $x,y\in\{\pm1\}^n$ where $x_i\leq y_i$ for all $i\in[n]$, $f(x)\leq f(y)$.

Combining Fact 4.2 with Lemma 5.1 immediately gives the following.

Corollary 4.4. *For any monotone function $g:\{\pm1\}^k\to\{\pm1\}$, there is a coordinate $i\in[k]$ satisfying, for $\boldsymbol{x}$ uniform from $\{\pm1\}^k$*

$$\mathrm{Cov}\left[\boldsymbol{x}_i,g(\boldsymbol{x})\right]\geq\Omega\left(\frac{\log k}{k}\cdot\mathrm{Var}[g(\boldsymbol{x})]\right).$$

We apply the above corollary to prove a class of distributions satisfying the weak-learning assumption. Before doing so, we'll need the notion of a *restriction*:

Restrictions. Given some domain $\mathcal{X}:=\{\pm1\}^d$, a *restriction* of that domain is a defined by value for each coordinate,

$\rho \in \{-1, +1, \star\}^d$. An input $x \in \mathcal{X}$ is said to be *consistent* with a restriction ρ, if $x_i = \rho_i$ for all $i \in [d]$ where we define $\star$ to be equal to both $+1$ and -1. The coordinates i where $\rho_i \in \{\pm 1\}$ are said to be *specified*, and we define $|\rho|$ to be the number of coordinates specified. The number of inputs consistent with a restriction ρ is $2^{d-|\rho|}$, and there is a natural projection proj_ρ from $\{\pm 1\}^{d-|\rho|}$ to the subset of $\mathcal{X}$ consistent with ρ that uses the input to proj_ρ for the unspecified coordinates and fills in the specified coordinates according to ρ.

Proposition 4.5. *For any $k \le d$ and monotone function $g : \{\pm 1\}^k \to \{\pm 1\}$, let $f : \{\pm 1\}^d \to \{\pm 1\}$ be the function that computes $f(x) = g(x_{[1:k]})$ where $x_{[1:k]}$ is the first k-bits of x and $\mathcal{D}$ be the distribution over $(\boldsymbol{x}, f(\boldsymbol{x}))$ where $\boldsymbol{x}$ is uniform in $\mathcal{X} := \{\pm 1\}^d$. Then, the set of coordinate projections, $\mathcal{H} := \{\mathrm{proj}_i : i \in [d]\}$, satisfies the $(\gamma = O((\log k)/k))$-weak learning assumption (Definition 2.7) w.r.t. $\mathcal{D}$.*

Proof. Let $\mathcal{D}'$ be any induced distribution of $\mathcal{D}$ by $\mathcal{H}$. Our goal is to show that Equation (2) is satisfied, or equivalently, that there is some $i \in [d]$ for which

$$\mathrm{Cov}_{\boldsymbol{x} \sim \mathcal{D}'_X}[\boldsymbol{x}_i, f(\boldsymbol{x})] \ge \gamma \, \mathrm{Var}_{\boldsymbol{x} \sim \mathcal{D}'_X}[f(\boldsymbol{x})].$$

As $\mathcal{H}$ is the set of coordinate projections and $\mathcal{D}_X$ is uniform over $\{\pm 1\}^d$, every induced distribution $\mathcal{D}'_X$ corresponds to the uniform distribution over all elements of $\mathcal{X}$ consistent with some restriction ρ. Given that restriction ρ, we can define the function $f_\rho : \{\pm 1\}^{d-|\rho|} \to \{\pm 1\}$ defined $f_\rho(x) = f(\mathrm{proj}_\rho(x))$.

As f is a monotone function of the first k coordinates of its input, f_ρ is a monotone function of the first (up to k) coordinates of its input. Hence, there is some $k' \le k$ and monotone $g_\rho : \{\pm 1\}^{k'} \to \{\pm 1\}$ for which $f_\rho(x) = g_\rho(x_{[1:k']})$. Then,

$$\begin{aligned}
&\max_{i \in [d]} \left(\operatorname*{Cov}_{\boldsymbol{x} \sim \mathcal{D}'_X} [\boldsymbol{x}_i, f(\boldsymbol{x})] \right) \\
&\quad = \max_{i \in [d-|\rho|]} \left(\operatorname*{Cov}_{\boldsymbol{x} \sim \{\pm 1\}^{d-|\rho|}} [\boldsymbol{x}_i, f_\rho(\boldsymbol{x})] \right) \\
&\quad = \max_{i \in [k']} \left(\operatorname*{Cov}_{\boldsymbol{x} \sim \{\pm 1\}^{k'}} [\boldsymbol{x}_i, g_\rho(\boldsymbol{x})] \right) \\
&\quad \ge \Omega\left(\frac{\log k'}{k'} \cdot \operatorname*{Var}_{\boldsymbol{x} \sim \{\pm 1\}^{k'}} [g_\rho(\boldsymbol{x})] \right) \quad \text{(Corollary 4.4)} \\
&\quad \ge \gamma \, \mathrm{Var}_{\boldsymbol{x} \sim \mathcal{D}'_X}[f(\boldsymbol{x})]. \\
&\qquad\qquad \text{(since } k' \le k, \gamma = O((\log k)/k))
\end{aligned}$$

This means that $\mathcal{H}$ satisfies our weak learning assumption w.r.t. $\mathcal{D}$. □

To design the distribution $\mathcal{D}$ of Theorem 4.1, we'll use the following proposition.

Proposition 4.6. *For any $\varepsilon \in (0, 1/3]$ and $d \ge \log_2(1/\varepsilon)$, for some integer $k \le d$, there is a monotone function $f : \{\pm 1\}^k \to \{\pm 1\}$ where, for $\boldsymbol{x} \sim \{\pm 1\}^k$ chosen uniformly,*

$$\min_{b \in \{\pm 1\}} \Pr[f(\boldsymbol{x}) = b] \ge \varepsilon$$

and,

$$\mathbb{E}[\boldsymbol{x}_1 f(\boldsymbol{x})] = \cdots = \mathbb{E}[\boldsymbol{x}_k f(\boldsymbol{x})] = O\left(\varepsilon \log\left(\frac{1}{\varepsilon}\right) \cdot \frac{\log d}{d} \right). \tag{7}$$

The proof of Proposition 4.6 is given in Appendix E.

Finally, we prove the main result of this section.

Proof of Theorem 4.1. Let

$$\ell := \left\lceil O\left(\frac{\log(1/\gamma)}{\gamma} \right) \right\rceil.$$

Note as we assume that $\gamma^{1/\gamma} \le \varepsilon$, we have that $\ell \ge \log_2(1/\varepsilon)$. Therefore, by Proposition 4.6, we know for some $k \le \ell$, there exists a monotone $g : \{\pm 1\}^k \to \{\pm 1\}$ where for $\boldsymbol{x} \sim \{\pm 1\}^k$ chosen uniformly,

$$\min_{b \in \{\pm 1\}} \Pr[g(\boldsymbol{x}) = b] \ge 2\varepsilon \tag{8}$$

and,

$$\begin{aligned}
v &:= \mathbb{E}[\boldsymbol{x}_1 g(\boldsymbol{x})] = \cdots = \mathbb{E}[\boldsymbol{x}_k g(\boldsymbol{x})] \\
&= O\left(\varepsilon \log(1/\varepsilon) \cdot \frac{\log \ell}{\ell} \right) \\
&= O\left(\varepsilon \log(1/\varepsilon) \cdot \gamma \right).
\end{aligned} \tag{9}$$

Let $f : \{\pm 1\}^d \to \{\pm 1\}$ be the function that computes $f(x) = g(x_{[1:k]})$, and let $\mathcal{D}$ be the distribution over $(\boldsymbol{x}, f(\boldsymbol{x}))$ where $\boldsymbol{x}$ is uniform in $\mathcal{X} := \{\pm 1\}^d$. Then, by Proposition 4.5, the class of coordinate projection, $\mathcal{H} := \{\mathrm{proj}_i : i \in [d]\}$, satisfies the γ-weak learning assumption (Definition 2.7) w.r.t. $\mathcal{D}$.

Next, we define the corrupted distribution $\widehat{\mathcal{D}}$. Let $\mathcal{E}$ be the distribution where, to sample $(\boldsymbol{x}, \boldsymbol{y}) \sim \mathcal{E}$, we first draw $\boldsymbol{y}$ uniformly in $\{\pm 1\}$ and $\boldsymbol{z}$ uniformly in $\{\pm 1\}^{d-k}$. Then, we set $\boldsymbol{x}$ to be

$$\boldsymbol{x} = (\underbrace{-\boldsymbol{y}, \ldots, -\boldsymbol{y}}_{k \text{ copies}}) \circ \boldsymbol{z}$$

where $\circ$ represents concatenation. Then, we set the corrupted distribution to be the mixture $\widehat{\mathcal{D}} := (1-\eta)\mathcal{D} + \eta\mathcal{E}$ where η is chosen as the unique solution of $(1-\eta)v - \eta = 0$ where v is as defined in Equation (9). Note that this solution satisfies $\eta \le v = O(\varepsilon \log(1/\varepsilon)\gamma)$. As $\widehat{\mathcal{D}}$ is a mixture with $(1-\eta)$ fraction coming from $\mathcal{D}$, it is an η-nasty noise corruption of $\mathcal{D}$. Furthermore, as the contribution of $\mathcal{E}$ is

chosen to exactly cancel out the correlations of $\mathcal{D}$, for all $i \in [d]$,

$$\mathbb{E}_{\widehat{\mathcal{D}}}[\boldsymbol{y}] = \mathbb{E}_{\widehat{\mathcal{D}}}[\boldsymbol{y} \mid \boldsymbol{x}_i = -1] = \mathbb{E}_{\widehat{\mathcal{D}}}[\boldsymbol{y} \mid \boldsymbol{x}_i = +1] = 0.$$

This means that for *any* impurity function $\mathcal{G}$, all hypotheses have a local drop in $\mathcal{G}$ of 0. Furthermore, all projections except for the first k are fully independent of one another and of the label. Therefore, if all internal nodes in the tree consists of projections for the last $d-k$ coordinates, then all hypotheses at every leaf will still have impurity gain.

As a result, TopDownDT will choose an arbitrary $(\ell^\star, h^\star)$ at each iteration. Unless $t \geq 2^{d-k}$, these arbitrary decisions can lead to the complete tree of depth $\log(t)$ being built, where all internal nodes have a hypothesis for one of the $d-k$ projection functions that are independent of $\boldsymbol{y}$.

In that case, by Equation (8) for every leaf ℓ, $\min_{b\in\{\pm1\}} \Pr_{\mathcal{D}_\ell}[g(\boldsymbol{x}) = b] \geq 2\varepsilon$. Therefore, regardless of how TopDownDT labels the leaves, the resulting tree will have error $\geq 2\varepsilon$. □

5. Proof of Theorem 1.3: Learning monotone decision trees in the presence of nasty noise

In this section we consider distributions $\mathcal{D}$ for which the marginal $\mathcal{D}_X$ is an arbitrary product distribution $\mathcal{D}_X = \mathcal{D}_X^{(1)} \times \ldots \times \mathcal{D}_X^{(d)}$ over $\{\pm1\}^d$ (i.e. each bit is independent) and the deterministic target function $\mathcal{D}_{Y|X}$ is monotone and representable by a size-s decision tree.

The following is a useful fact about the influence of features on *monotone* functions. See Appendix F for a proof.

Lemma 5.1 (Influence = covariance for monotone functions)**.** *Let $\mathcal{D}_X = \mathcal{D}_X^{(1)} \times \ldots \times \mathcal{D}_X^{(d)}$ be an arbitrary product distribution over $\{\pm1\}^d$. For a monotone function $f : \{\pm1\}^d \to \{\pm1\}$ and a feature $i \in [d]$, we have the identity $\mathrm{Inf}_i(f) = \mathrm{Cov}_{\mathcal{D}_X}[f(\boldsymbol{x}), \boldsymbol{x}_i]$.*

The key technical ingredient in our proof of Theorem 1.3 is a theorem of O'Donnell, Saks, Schramm, and Servedio from discrete Fourier analysis (O'Donnell et al., 2005):

Theorem 5.2 (OSSS inequality)**.** *Let $f : \{\pm1\}^d \to \{\pm1\}$ be function that is representable by a size-s decision tree and $\mathcal{D}_X$ be a product distribution over $\{\pm1\}^d$. Then*

$$\max_{i\in[d]}\{\mathrm{Inf}_i(f)\} \geq \frac{\mathrm{Var}[f]}{\log s}, \tag{10}$$

where $\mathrm{Inf}_i(f)$ and $\mathrm{Var}[f]$ are with respect to $\mathcal{D}_X$.

For the class of distributions described at the beginning of this section, Lemma 5.1 and Theorem 5.2 together imply that the weak learning assumption of Theorem 3.1 can be satisfied by the set $\mathcal{H} = \{\mathrm{proj}_i : i \in [d]\}$ of projection functions:

Lemma 5.3 (Projection functions satisfy weak learning assumption)**.** *Let $\mathcal{D}$ be a distribution for which the marginal $\mathcal{D}_X$ is a product distribution over $\mathcal{X} = \{\pm1\}^d$ and the target function $f := \mathcal{D}_{Y|X}$ is monotone and can represented as a size-s decision tree. The set $\mathcal{H} = \{\mathrm{proj}_i : i \in [d]\}$ of projection functions satisfies the γ-weak learning assumption w.r.t. $\mathcal{D}$ with $\gamma = 1/\log s$.*

Proof. We first note that there is an $h \in \mathcal{H}$ that is a γ-advantage hypothesis with respect to $\mathcal{D}$:

$$\mathrm{Cov}[f(\boldsymbol{x}), \boldsymbol{x}_i] = \mathrm{Inf}_i(f) \geq \frac{\mathrm{Var}[f]}{\log s},$$

where we have used Lemma 5.1 for the equality and Theorem 5.2 for the inequality. Since $\mathcal{H}$ is the set of projection functions, every distribution $\mathcal{D}'$ that is induced by conditioning $\mathcal{D}$ on $\mathcal{H}$ is such that $\mathcal{D}'_X$ is a product distribution over $\{\pm1\}^S$ for some $S \subseteq [d]$. Similarly, since f is monotone and representable by a size-s decision tree, the same remains true for any restriction of f by the projection functions in $\mathcal{H}$. Therefore, we can again apply Lemma 5.1 and Theorem 5.2 to infer the existence of a γ-advantage hypothesis $h \in \mathcal{H}$ with respect to $\mathcal{D}'$. □

Theorem 1.3 is now an immediate consequence of Theorem 3.1 and Lemma 5.3:

Theorem 5.4 (Formal version of Theorem 1.3)**.** *Let $\mathcal{D}$ be a distribution for which the marginal $\mathcal{D}_X$ is a product distribution over $\mathcal{X} = \{\pm1\}^d$ and the target function $\mathcal{D}_{Y|X}$ is monotone and can be represented as a size-s decision tree. For any impurity function $\mathcal{G}$, noise rate $\eta \leq O(\varepsilon/\log s)$, and distribution $\widehat{\mathcal{D}}$ such that $\mathrm{dist}_{\mathrm{TV}}(\widehat{\mathcal{D}}, \mathcal{D}) \leq \eta$, the algorithm $\mathrm{TopDownDT}_{\mathcal{H},\mathcal{G},\widehat{\mathcal{S}}}(t)$ where $t := s^{O((\log s)/\varepsilon^2)}$ runs in $\mathrm{poly}(d) \cdot s^{O((\log s)/\varepsilon^2)}$ time and constructs a size-t decision tree hypothesis T satisfying $\mathrm{error}_{\mathcal{D}}[T] \leq \varepsilon$.*

6. Conclusion

We have given the first noise tolerance guarantees for the class of impurity-based decision tree learning algorithms that hold in a fully general setting. Theorem 3.1 shows that they are noise-tolerant boosting algorithms that combine γ-advantage weak hypotheses into a strong hypothesis with error $\leq \varepsilon$, even in the presence nasty noise of rate as high as $\eta \leq O(\varepsilon\gamma)$. Theorem 4.1 provides a near-matching lower bound ruling out, in a strong sense, any such guarantee for noise rates $\eta \geq \tilde{\Omega}(\varepsilon\gamma)$. Finally, instantiating Theorem 3.1 in the setting of product distributions over binary features—a setting that is particularly well studied in the theoretical literature—we show that these classic and widely-used algorithms achieve guarantees that are better than those known for any existing theoretical algorithms. Taken as a whole, our work helps place the popularity and empirical success

of impurity-based decision tree learning algorithms on firm theoretical footing.

There are several immediate avenues for future research. First, a natural next step is to establish similar formal noise tolerance guarantees for tree-based ensemble methods such as random forests and XGBoost. Second, the focus of our work has been on understanding properties of impurity-based decision tree learning algorithms exactly as they are, to provide theoretical justification for their practical effectiveness. It would nevertheless be interesting to consider possible modifications of these algorithms that are even more resilient to adversarial noise—for example, are there such modifications that evade our lower bounds?

7. Acknowledgments

Guy and Li-Yang are supported by NSF CAREER Award 1942123. Jane is supported by NSF Award CCF-2006664. Ali is supported by a graduate fellowship award from Knight-Hennessy Scholars at Stanford University.

References

Blanc, G., Lange, J., and Tan, L.-Y. Provable guarantees for decision tree induction: the agnostic setting. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020a.

Blanc, G., Lange, J., and Tan, L.-Y. Top-down induction of decision trees: rigorous guarantees and inherent limitations. In *Proceedings of the 11th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 151, pp. 1–44, 2020b.

Blanc, G., Lange, J., Qiao, M., and Tan, L. Decision tree heuristics can fail, even in the smoothed setting. In Wootters, M. and Sanità, L. (eds.), *Proceedings of the 25th International Conference on Randomization and Computation (RANDOM)*, volume 207, pp. 45:1–45:16, 2021a.

Blanc, G., Lange, J., Qiao, M., and Tan, L. Properly learning decision trees in almost polynomial time. In *Proceedings of the 62nd IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, 2021b.

Blanc, G., Lange, J., and Tan, L.-Y. Learning Stochastic Decision Trees. In *Proceedings of the 48th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 198 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 30:1–30:16, 2021c. ISBN 978-3-95977-195-5.

Blanc, G., Lange, J., Malik, A., and Tan, L.-Y. On the power of adaptivity in statistical adversaries. In *Proceedings of the 35th Annual Conference on Computational Learning Theory (COLT)*, 2022.

Blum, A., Furst, M., Jackson, J., Kearns, M., Mansour, Y., and Rudich, S. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 253–262, 1994.

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

Breiman, L., Friedman, J., Stone, C., and Olshen, R. *Classification and regression trees*. Wadsworth International Group, 1984.

Brutzkus, A., Daniely, A., and Malach, E. On the Optimality of Trees Generated by ID3. *ArXiv*, abs/1907.05444, 2019.

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

Bshouty, N. Exact learning via the monotone theory. In *Proceedings of 34th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 302–311, 1993.

Bshouty, N. H., Eiron, N., and Kushilevitz, E. PAC learning with nasty noise. *Theoretical Computer Science*, 288(2): 255–275, 2002.

Chen, S. and Moitra, A. Beyond the low-degree algorithm: mixtures of subcubes and their applications. In *Proceedings of the 51st Annual ACM Symposium on Theory of Computing (STOC)*, pp. 869–880, 2019.

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

Dachman-Soled, D., Feldman, V., Tan, L.-Y., Wan, A., and Wimmer, K. Approximate resilience, monotonicity, and the complexity of agnostic learning. In *Proceedings of the 26th Annual Symposium on Discrete Algorithms (SODA)*, pp. 498–511, 2015.

Dietterich, T., Kearns, M., and Mansour, Y. Applying the weak learning framework to understand and improve C4.5. In *Proceedings of the 13th International Conference on Machine Learning (ICML)*, pp. 96–104, 1996.

Fiat, A. and Pechyony, D. Decision trees: More theoretical justification for practical algorithms. In *Proceedings of the 15th International Conference on Algorithmic Learning Theory (ALT)*, pp. 156–170, 2004.

Freund, Y. Boosting a weak learning algorithm by majority. *Information and computation*, 121(2):256–285, 1995.

Gopalan, P., Kalai, A., and Klivans, A. Agnostically learning decision trees. In *Proceedings of the 40th ACM Symposium on Theory of Computing (STOC)*, pp. 527–536, 2008.

Hancock, T. Learning $k\mu$ decision trees on the uniform distribution. In *Proceedings of the 6th Annual Conference on Computational Learning Theory (COT)*, pp. 352–360, 1993.

Haussler, D. Decision theoretic generalizations of the pac model for neural net and other learning applications. *Information and computation*, 100(1):78–150, 1992.

Hazan, E., Klivans, A., and Yuan, Y. Hyperparameter optimization: A spectral approach. *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.

Jackson, J. C. and Servedio, R. A. On learning random dnf formulas under the uniform distribution. *Theory of Computing*, 2(8):147–172, 2006. doi: 10.4086/toc.2006.v002a008. URL http://www.theoryofcomputing.org/articles/v002a008.

Kahn, J., Kalai, G., and Linial, N. The influence of variables on boolean functions. In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 68–80, 1988.

Kalai, A. Learning monotonic linear functions. In *Proceedings of the 17th Annual International Conference on Computational Learning Theory (COLT)*, pp. 487–501. Springer, 2004.

Kalai, A., Klivans, A., Mansour, Y., and Servedio, R. A. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008a.

Kalai, A., Samorodnitsky, A., and Teng, S.-H. Learning and smoothed analysis. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 395–404, 2009.

Kalai, A. T. and Servedio, R. A. Boosting in the presence of noise. *Journal of Computer and System Sciences*, 71 (3):266–290, 2005.

Kalai, A. T., Mansour, Y., and Verbin, E. On agnostic boosting and parity learning. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 629–638, 2008b.

Kearns, M. Boosting theory towards practice: recent developments in decision tree induction and the weak learning framework (invited talk). In *Proceedings of the 13th National Conference on Artificial intelligence (AAAI)*, pp. 1337–1339, 1996.

Kearns, M. and Mansour, Y. On the boosting ability of top-down decision tree learning algorithms. In *Proceedings of the 28th Annual Symposium on the Theory of Computing (STOC)*, pp. 459–468, 1996.

Kearns, M., Schapire, R., and Sellie, L. Toward efficient agnostic learning. *Machine Learning*, 17(2/3):115–141, 1994.

Kushilevitz, E. and Mansour, Y. Learning decision trees using the fourier spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, December 1993.

Lee, H. *On the learnability of monotone functions*. PhD thesis, Columbia University, 2009.

Linial, N., Mansour, Y., and Nisan, N. Constant depth circuits, Fourier transform and learnability. *Journal of the ACM*, 40(3):607–620, 1993.

Long, P. and Servedio, R. Adaptive martingale boosting. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L. (eds.), *Proceedings of the 22nd Annucal Conference on Advances in Neural Information Processing Systems (NeurIPS)*, volume 21. Curran Associates, Inc., 2009. URL https://proceedings.neurips.cc/paper/2008/file/38b3eff8baf56627478ec76a704e9b52-Paper.pdf.

Long, P. M. and Servedio, R. A. Martingale boosting. In *Proceedings of the 18th Annual International Conference on Computational Learning Theory (COLT)*, pp. 79–94. Springer, 2005.

Mansour, Y. and McAllester, D. Boosting using branching programs. *Journal of Computer and System Sciences*, 64 (1):103–112, 2002.

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

O'Donnell, R. and Servedio, R. Learning monotone decision trees in polynomial time. *SIAM Journal on Computing*, 37(3):827–844, 2007.

O'Donnell, R., Saks, M., Schramm, O., and Servedio, R. Every decision tree has an influential variable. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 31–39, 2005.

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

Quinlan, R. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 1558602402.

A. Other related work

Boosting by branching programs. Kearns and Mansour (Kearns & Mansour, 1996) (see also (Kearns, 1996; Dietterich et al., 1996)) were the first to propose the perspective of viewing impurity-based decision tree algorithms as boosting algorithms. Their analysis assumes the noiseless setting. Subsequently, departing from (Kearns & Mansour, 1996)'s motivation of analyzing practical decision tree algorithms, Mansour and McAllester (Mansour & McAllester, 2002) initiated a line of work on boosting by *branching programs*, a variant of decision trees where the underlying graph is a DAG rather than a tree. While (Mansour & McAllester, 2002) assumes the noiseless setting, the followup works (Kalai, 2004; Long & Servedio, 2005; Kalai & Servedio, 2005; Long & Servedio, 2009) handle various types of random (i.e. non-adversarial) label noise, and the work of (Kalai et al., 2008b) handles agnostic noise.

Our work differs from this line of work in two ways: first and foremost, our results apply to impurity-based decision tree algorithms such as ID3, CART, and C4.5—the overarching goal of our work is to analyze and establish noise tolerance properties of these algorithms that are widely used in practice—whereas branching programs are much less commonly used. Second, we handle the strongest noise model of nasty noise, whereas these results only allow for corruptions of labels and not features.

Theoretical work on decision tree learning. Decision trees are one of the most intensively studied concept classes in learning theory. The literature on this problem is rich and vast, spanning over three decades, and it continues to grow. However, in most of these works, the algorithms analyzed not resemble practical impurity-based decision tree algorithms. Indeed, most of them are *improper* algorithms, in the sense that their hypotheses are not themselves decision trees. Quoting Kearns and Mansour (Kearns & Mansour, 1996), "In summary, it seems fair to say that despite their other successes, the models of computational learning theory have not yet provided significant insight into the apparent empirical success of programs like C4.5 and CART."

B. Bounds with Total Variation Distance

Lemma B.1 (Expectation and TV-distance)**.** *Let $\mathcal{E}, \widehat{\mathcal{E}}$ be two distributions over a domain $\mathcal{V}$ with $\mathrm{dist}_{\mathrm{TV}}(\mathcal{E}, \widehat{\mathcal{E}}) \leq \eta$ and let $f : \mathcal{V} \to [0,1]$. Then*

$$\left| \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{E}}[f(\boldsymbol{x})] - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \widehat{\mathcal{E}}}[f(\boldsymbol{x})] \right| \leq \eta. \tag{11}$$

Proof. The result follows immediately from the following definition of total variation distance:

$$\mathrm{dist}_{\mathrm{TV}}(\mathcal{E}, \widehat{\mathcal{E}}) = \sup_{T : \mathcal{V} \to [0,1]} \left(\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{E}}[T(\boldsymbol{x})] - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \widehat{\mathcal{E}}}[T(\boldsymbol{x})] \right). \qquad \square$$

Lemma B.2 (Variance and TV-distance)**.** *Let $\mathcal{E}, \widehat{\mathcal{E}}$ be two distributions over a domain $\mathcal{V}$ with $\mathrm{dist}_{\mathrm{TV}}(\mathcal{E}, \widehat{\mathcal{E}}) \leq \eta$ and let $f : \mathcal{V} \to [0,1]$. Then*

$$\left| \mathrm{Var}_{\mathcal{E}}[f(\boldsymbol{x})] - \mathrm{Var}_{\widehat{\mathcal{E}}}[f(\boldsymbol{x})] \right| \leq \eta.$$

Proof. We can write

$$\mathrm{Var}_{\mathcal{E}}[f(\boldsymbol{x})] = \mathop{\mathbb{E}}_{\substack{\boldsymbol{x} \sim \mathcal{E} \\ \boldsymbol{x}' \sim \mathcal{E}}} \left[\frac{(f(\boldsymbol{x}) - f(\boldsymbol{x}'))^2}{2} \right]. \tag{12}$$

If $\mathrm{dist}_{\mathrm{TV}}(\mathcal{E}, \widehat{\mathcal{E}}) \leq \eta$, then it is easy to see that $\mathrm{dist}_{\mathrm{TV}}(\mathcal{E}^2, \widehat{\mathcal{E}}^2) \leq 2\eta$, where $\mathcal{E}^2$ indicates the product distribution of two independent draws from $\mathcal{E}$, Moreover, since $(f(\boldsymbol{x}) - f(\boldsymbol{x}'))^2 \leq 1$, we can apply Lemma B.1 to Equation (12) with $\mathcal{E}^2$ and $\widehat{\mathcal{E}}^2$ to get $\left| \mathrm{Var}_{\mathcal{E}}[f(\boldsymbol{x})] - \mathrm{Var}_{\widehat{\mathcal{E}}}[f(\boldsymbol{x})] \right| \leq \eta$. $\square$

Lemma B.3 (Covariance and TV-distance)**.** *Let $\mathcal{E}, \widehat{\mathcal{E}}$ be two distributions over a domain $\mathcal{V}$ with $\mathrm{dist}_{\mathrm{TV}}(\mathcal{E}, \widehat{\mathcal{E}}) \leq \eta$ and let $f, g : \mathcal{V} \to [0,1]$ be two functions. Then*

$$\left|\mathrm{Cov}_{\mathcal{E}}[f(\boldsymbol{x}), g(\boldsymbol{x})] - \mathrm{Cov}_{\widehat{\mathcal{E}}}[f(\boldsymbol{x}), g(\boldsymbol{x})]\right| \leq 2\eta. \tag{13}$$

Proof. We can write

$$Cov_{\mathcal{E}}[f(\boldsymbol{x}), g(\boldsymbol{x})] = \mathop{\mathbb{E}}_{\substack{\boldsymbol{x}\sim\mathcal{E}\\ \boldsymbol{x}'\sim\mathcal{E}}}\left[\frac{(f(\boldsymbol{x}) - f(\boldsymbol{x}'))(g(\boldsymbol{x}) - g(\boldsymbol{x}'))}{2}\right]$$

If $\mathrm{dist}_{\mathrm{TV}}(\mathcal{E}, \widehat{\mathcal{E}}) \leq \eta$, then it is easy to see that $\mathrm{dist}_{\mathrm{TV}}(\mathcal{E}^2, \widehat{\mathcal{E}}^2) \leq 2\eta$, where $\mathcal{E}^2$ indicates the product distribution of two independent draws from $\mathcal{E}$, Moreover, since $f, g : \mathcal{V} \to \{0,1\}$, we have $((f(\boldsymbol{x}) - f(\boldsymbol{x}'))(g(\boldsymbol{x}) - g(\boldsymbol{x}'))) \in [-1,1]$, so we can apply Lemma B.1 to Equation (13) with $\mathcal{E}^2$ and $\widehat{\mathcal{E}}^2$ to get $\left|\mathrm{Cov}_{\mathcal{E}}[f(\boldsymbol{x}), g(\boldsymbol{x})] - \mathrm{Cov}_{\widehat{\mathcal{E}}}[f(\boldsymbol{x}), g(\boldsymbol{x})]\right| \leq 2\eta$. □

Lemma B.4. *For any finite set L, (possibly infinite) set X, and distributions $\mathcal{D}, \widehat{\mathcal{D}}$ each over the product domain $L \times X$,*

$$\mathop{\mathbb{E}}_{\boldsymbol{\ell}\sim\mathcal{D}_\ell}\left[\mathrm{dist}_{\mathrm{TV}}(\mathcal{D}_{x|\boldsymbol{\ell}}, \widehat{\mathcal{D}}_{x|\boldsymbol{\ell}})\right] \leq 2\,\mathrm{dist}_{\mathrm{TV}}(\mathcal{D}, \widehat{\mathcal{D}}) \tag{14}$$

where $\mathcal{D}_\ell$ is marginal distribution of $\boldsymbol{\ell}$ for $(\boldsymbol{x}, \boldsymbol{\ell}) \sim \mathcal{D}$ and $\mathcal{D}_{x|\ell}$ is the conditional distribution of $\boldsymbol{x}$ when $(\boldsymbol{\ell}, \boldsymbol{x}) \sim \mathcal{D}$ conditioned upon $\boldsymbol{\ell} = \ell$.

Intuitively, we will define a distribution $\mathcal{D}'$ that "mixes" $\mathcal{D}$ and $\widehat{\mathcal{D}}$: To sample $(\boldsymbol{\ell}, \boldsymbol{x}) \sim \mathcal{D}'$, we first draw $\boldsymbol{\ell} \sim \mathcal{D}_\ell$ and then $\boldsymbol{x} \sim \widehat{\mathcal{D}}_{x|\boldsymbol{\ell}}$. We'll be able to show that the l.h.s. of Equation (14) is equal to $\mathrm{dist}_{\mathrm{TV}}(\mathcal{D}, \mathcal{D}')$.

Then, we'll show that $\mathrm{dist}_{\mathrm{TV}}(\mathcal{D}', \widehat{\mathcal{D}}) = \mathrm{dist}_{\mathrm{TV}}(\mathcal{D}_\ell, \widehat{\mathcal{D}}_\ell) \leq \mathrm{dist}_{\mathrm{TV}}(\mathcal{D}, \widehat{\mathcal{D}})$. Finally, we can bound $\mathrm{dist}_{\mathrm{TV}}(\mathcal{D}, \mathcal{D}') \leq \mathrm{dist}_{\mathrm{TV}}(\mathcal{D}, \widehat{\mathcal{D}}) + \mathrm{dist}_{\mathrm{TV}}(\mathcal{D}', \widehat{\mathcal{D}})$ using triangle inequality.

There are many (equivalent) definitions of total variation distance. To formalize the above intuition, we will use the formulation that, for two distributions $\mathcal{D}, \widehat{\mathcal{D}}$ over a domain Ω,

$$\mathrm{dist}_{\mathrm{TV}}(\mathcal{D}, \widehat{\mathcal{D}}) \coloneqq \sup_{A\subseteq\Omega}(\mathcal{D}(A) - \widehat{\mathcal{D}}(A)) \tag{15}$$

where $\mathcal{D}(A)$ is equal to $\Pr_{\boldsymbol{x}\sim\mathcal{D}}[\boldsymbol{x} \in A]$.

Proof of Lemma B.4. We compute,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\boldsymbol{\ell}\sim\mathcal{D}_\ell}\left[\mathrm{dist}_{\mathrm{TV}}(\mathcal{D}_{x|\boldsymbol{\ell}}, \widehat{\mathcal{D}}_{x|\boldsymbol{\ell}})\right] \\
&\quad= \mathop{\mathbb{E}}_{\boldsymbol{\ell}\sim\mathcal{D}_\ell}\left[\sup_{X_{\boldsymbol{\ell}}\subseteq X}\left(\mathcal{D}_{x|\boldsymbol{\ell}}(X_{\boldsymbol{\ell}}) - \widehat{\mathcal{D}}_{x|\boldsymbol{\ell}}(X_{\boldsymbol{\ell}})\right)\right] && \text{(Equation (15))} \\
&\quad= \sup_{X_{\boldsymbol{\ell}}\subseteq X \text{ for each } \ell\in L}\left(\mathop{\mathbb{E}}_{\boldsymbol{\ell}\sim\mathcal{D}_\ell}\left[\mathcal{D}_{x|\boldsymbol{\ell}}(X_{\boldsymbol{\ell}}) - \widehat{\mathcal{D}}_{x|\boldsymbol{\ell}}(X_{\boldsymbol{\ell}})\right]\right) && \text{(sup and } \mathbb{E} \text{ commute because } L \text{ is finite)} \\
&\quad= \sup_{A\subseteq L\times X}\left(\mathop{\mathbb{E}}_{\boldsymbol{\ell}\sim\mathcal{D}_\ell}\left[\mathcal{D}_{x|\boldsymbol{\ell}}(A_{\boldsymbol{\ell}}) - \widehat{\mathcal{D}}_{x|\boldsymbol{\ell}}(A_{\boldsymbol{\ell}})\right]\right) && \text{(defining } A_\ell \coloneqq \{x|(x,\ell)\in A\}) \\
&\quad= \sup_{A\subseteq L\times X}\left(\mathop{\mathbb{E}}_{\boldsymbol{\ell}\sim\mathcal{D}_\ell}\left[\mathcal{D}_{x|\boldsymbol{\ell}}(A_{\boldsymbol{\ell}})\right] - \mathop{\mathbb{E}}_{\widehat{\boldsymbol{\ell}}\sim\widehat{\mathcal{D}}_\ell}\left[\widehat{\mathcal{D}}_{x|\widehat{\boldsymbol{\ell}}}(A_{\widehat{\boldsymbol{\ell}}})\right]\right) + \sup_{A\subseteq L\times X}\left(\mathop{\mathbb{E}}_{\boldsymbol{\ell}\sim\mathcal{D}_\ell}\left[\widehat{\mathcal{D}}_{x|\boldsymbol{\ell}}(A_{\boldsymbol{\ell}})\right] - \mathop{\mathbb{E}}_{\widehat{\boldsymbol{\ell}}\sim\widehat{\mathcal{D}}_\ell}\left[\widehat{\mathcal{D}}_{x|\widehat{\boldsymbol{\ell}}}(A_{\widehat{\boldsymbol{\ell}}})\right]\right) \\
&&& \text{(triangle inequality)}
\end{aligned}$$

We analyze each term of the above two terms separately. For the first,

$$\begin{aligned}
&\sup_{A\subseteq L\times X}\left(\mathop{\mathbb{E}}_{\boldsymbol{\ell}\sim\mathcal{D}_\ell}\left[\mathcal{D}_{x|\boldsymbol{\ell}}(A_{\boldsymbol{\ell}})\right] - \mathop{\mathbb{E}}_{\widehat{\boldsymbol{\ell}}\sim\widehat{\mathcal{D}}_\ell}\left[\widehat{\mathcal{D}}_{x|\widehat{\boldsymbol{\ell}}}(A_{\widehat{\boldsymbol{\ell}}})\right]\right) \\
&\qquad= \sup_{A\subseteq L\times X}\left(\mathcal{D}(A) - \widehat{\mathcal{D}}(A)\right) = \mathrm{dist}_{\mathrm{TV}}(\mathcal{D}, \widehat{\mathcal{D}}).
\end{aligned}$$

Next, we'll bound the second term. Using $p_{\mathcal{D}}(\ell)$ as shorthand for $\Pr_{\boldsymbol{\ell}\sim\mathcal{D}_\ell}[\boldsymbol{\ell}=\ell]$,

$$\sup_{A\subseteq L\times X}\left(\mathop{\mathbb{E}}_{\boldsymbol{\ell}\sim\mathcal{D}_\ell}\left[\widehat{\mathcal{D}}_{x|\boldsymbol{\ell}}(A_{\boldsymbol{\ell}})\right]-\mathop{\mathbb{E}}_{\widehat{\boldsymbol{\ell}}\sim\widehat{\mathcal{D}}_\ell}\left[\widehat{\mathcal{D}}_{x|\widehat{\boldsymbol{\ell}}}(A_{\widehat{\boldsymbol{\ell}}})\right]\right)$$
$$=\sup_{A\subseteq L\times X}\sum_{\ell\in L}\left(p_{\mathcal{D}}(\ell)\cdot\widehat{\mathcal{D}}_{x|\ell}(A_\ell)-p_{\widehat{\mathcal{D}}}(\ell)\cdot\widehat{\mathcal{D}}_{x|\ell}(A_\ell)\right)$$

The above is maximized by setting $A_\ell=X$ whenever $p_{\mathcal{D}}(\ell)\geq p_{\widehat{\mathcal{D}}}(\ell)$ and $A_\ell=\emptyset$ otherwise. Therefore,

$$\sup_{A\subseteq L\times X}\left(\mathop{\mathbb{E}}_{\boldsymbol{\ell}\sim\mathcal{D}_\ell}\left[\widehat{\mathcal{D}}_{x|\boldsymbol{\ell}}(A_{\boldsymbol{\ell}})\right]-\mathop{\mathbb{E}}_{\widehat{\boldsymbol{\ell}}\sim\widehat{\mathcal{D}}_\ell}\left[\widehat{\mathcal{D}}_{x|\widehat{\boldsymbol{\ell}}}(A_{\widehat{\boldsymbol{\ell}}})\right]\right)$$
$$=\max\left(p_{\mathcal{D}}(\ell)-p_{\widehat{\mathcal{D}}}(\ell),0\right)$$
$$=\sup_{A'\subseteq L}\left(\mathcal{D}_\ell(A')-\widehat{\mathcal{D}}_\ell(A')\right)=\mathrm{dist}_{\mathrm{TV}}(\mathcal{D}_\ell,\widehat{\mathcal{D}}_\ell).$$

Finally, we note that $\mathrm{dist}_{\mathrm{TV}}(\mathcal{D}_\ell,\widehat{\mathcal{D}}_\ell)\leq\mathrm{dist}_{\mathrm{TV}}(\mathcal{D},\widehat{\mathcal{D}})$ as the TV distance of marginal distributions is at most the TV distance of the original distributions. Combining all of the above,

$$\mathop{\mathbb{E}}_{\boldsymbol{\ell}\sim\mathcal{D}_\ell}\left[\mathrm{dist}_{\mathrm{TV}}(\mathcal{D}_{x|\boldsymbol{\ell}},\widehat{\mathcal{D}}_{x|\boldsymbol{\ell}})\right]\leq\mathrm{dist}_{\mathrm{TV}}(\mathcal{D},\widehat{\mathcal{D}})+\mathrm{dist}_{\mathrm{TV}}(\mathcal{D},\widehat{\mathcal{D}})=2\,\mathrm{dist}_{\mathrm{TV}}(\mathcal{D},\widehat{\mathcal{D}}).\qquad\square$$

C. General Impurity Functions

Remark C.1 (Other impurity functions). Lemma 3.2 is the only part of the proof of Theorem 3.1 that depends on the particular impurity function $\mathcal{G}$. That Lemma goes through for any impurity function that, for any constant κ, satisfies $\mathcal{G}''(x)\leq-\kappa$ for all $x\in(0,1)$, though the 16 in Equation (3) is replaced with 2κ. This is because (using the fact that $\mu(\mathcal{D}_\ell)=\Pr_{\mathcal{D}_\ell}[h(\boldsymbol{x})=0]\cdot\mu(\mathcal{D}_{\ell_0})+\Pr_{\mathcal{D}_\ell}[h(\boldsymbol{x})=1]\cdot\mu(\mathcal{D}_{\ell_1})$), we can bound the local drop in $\mathcal{G}$ as

$$\Delta_{\mathcal{D}_\ell}(h)\geq\frac{\kappa}{2}\cdot\left(\Pr_{\mathcal{D}_\ell}[h(\boldsymbol{x})=0]\cdot(\mu(\mathcal{D}_{\ell_0})-\mu(\mathcal{D}_\ell))^2)+\Pr_{\mathcal{D}_\ell}[h(\boldsymbol{x})=1]\cdot(\mu(\mathcal{D}_{\ell_1})-\mu(\mathcal{D}_\ell))^2)\right).$$

The above holds with equality when $\mathcal{G}'(x)=4x(1-x)$ for $\kappa=8$. Therefore, the local drop in $\mathcal{G}$ will always be at least $\frac{\kappa}{8}$ as large as it is for $\mathcal{G}'$. The remainder of the proof of Theorem 3.1 goes through unmodified except for slight changes to the constants hidden by $O(\cdot)$.

D. Learning with samples versus exact expectations

In the pseudocode provided in Figure 1, we assume that we can exactly compute the drop in impurity and therefore can exactly compute expectations of the form $\mu(\mathcal{D}_\ell)=\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim\mathcal{D}_\ell}[\boldsymbol{y}]$. If instead, those expectations are estimated using random samples (as in the statements of Theorems 1.1 and 1.3 in the introduction), the algorithm only can estimate them to within some tolerance $\pm\tau$. It is straightforward to carry out the proofs of Theorems 3.1 and 5.4 accounting for this $\pm\tau$ error as long as $\tau\leq O(\frac{\gamma^2\varepsilon^2}{t})$.

As the TopDownDT only needs to compute at most $O(t^2|\mathcal{H}|)$ expectations, standard concentration inequalities can be used to show that a sample of size $\tilde{O}(1/\tau^2\cdot t^2\cdot|\mathcal{H}|)$ is sufficient to compute all expectations to accuracy τ with high probability. However, the situation with noise is slightly more nuanced, as the adversary gets to see the sample $\boldsymbol{S}\sim\mathcal{D}^n$ before deciding on η-nasty-noise corruption $\hat{\boldsymbol{S}}$. This means the adversary can choose how to modify empirical estimates after seeing the sample $\boldsymbol{S}$. Luckily, Theorem 5 of (Blanc et al., 2022) handles exactly this case. It says that as long as the sample has size $\tilde{O}(1/\tau^2\cdot t^2\cdot|\mathcal{H}|)$, with high probability, all empirical estimates computed on the corrupted sample $\hat{\boldsymbol{S}}$ will be within $\pm\tau$ of the true expectations of some $\hat{\mathcal{D}}$ that is η-close to $\mathcal{D}$. Therefore, by proving that TopDownDT succeeds on *every* $\hat{\mathcal{D}}$ that is η-close to $\mathcal{D}$, as we do in Theorem 3.1, we ensure that our algorithm succeeds even if an adversary gets to modify η-fraction of a sample after seeing it.

Runtime analysis. Before proving that TopDownDT build low error trees, we will briefly prove that it is efficient. Let ζ be the time it takes to compute $\mu(\mathcal{D}_\ell)$ and $\Pr_{\mathcal{D}_\ell}[h(x)=1]$ for some leaf ℓ of the tree. Typically, this will be proportional to the size of the data set. Then, as the number of leaves in each iteration will be at most t, the time required for an iteration of TopDownDT is at most $O(\zeta t\cdot|\mathcal{H}|)$. The algorithm runs for t iterations, so the total runtime is $O(\zeta t^2\cdot|\mathcal{H}|)$.

E. Lower bounds

In this section, we prove Proposition 4.6, restated for convenience.

Proposition E.1. *For any $\varepsilon \in (0, 1/3]$ and $d \geq \log_2(1/\varepsilon)$, for some integer $k \leq d$, there is a monotone function $f : \{\pm 1\}^k \to \{\pm 1\}$ where, for $\boldsymbol{x} \sim \{\pm 1\}^k$ chosen uniformly,*

$$\min_{b \in \{\pm 1\}} \Pr[f(\boldsymbol{x}) = b] \geq \varepsilon$$

and,

$$\mathbb{E}[\boldsymbol{x}_1 f(\boldsymbol{x})] = \cdots = \mathbb{E}[\boldsymbol{x}_k f(\boldsymbol{x})] = O\left(\varepsilon \log(1/\varepsilon) \cdot \frac{\log d}{d}\right) \tag{16}$$

The function f will be based on the TRIBES function.

Definition E.2 (TRIBES)**.** For any $s, w \in \mathbb{N}$, the function $\text{TRIBES}_{w,s} : \{\pm 1\}^{ws} \to \{\pm 1\}$ is defined to be the function computed by the read-once disjunctive normal form with s terms (over disjoint sets of variables) of width exactly w:

$$\text{TRIBES}_{w,s}(x) = (x_{1,1} \wedge \cdots \wedge x_{1,w}) \vee \cdots \vee (x_{s,1} \wedge \cdots \wedge x_{s,w})$$

and where we adopt the convention that -1 represents logical FALSE and 1 represents logical TRUE.

We'll use the following easy to verify facts (see Chapter §4.2 of (O'Donnell, 2014)) about TRIBES.

Fact E.3 (Properties of TRIBES)**.** *For any $s, w \in \mathbb{N}$ and $\boldsymbol{x}$ uniform in $\{\pm 1\}^{ws}$,*

$$\Pr[\text{TRIBES}_{w,s}(\boldsymbol{x}) = -1] = (1 - 2^{-w})^s,$$

and, for each $i \in [sw]$,

$$\mathbb{E}[x_i \cdot \text{TRIBES}_{w,s}(\boldsymbol{x})] = \frac{1}{2^w - 1} \cdot \Pr[\text{TRIBES}_{w,s}(\boldsymbol{x}) = -1].$$

Proof of Proposition E.1. For any $w \in \mathbb{N}$, let s_w be the largest integer s such that $(1 - 2^{-w})^s \geq \varepsilon$, and let $w^\star$ be the largest integer for which $ws_w \leq d$. As $d \geq \log(1/\varepsilon)$, $w^\star \geq 1$. We will prove that $\text{TRIBES}_{w^\star, s_{w^\star}}$ meets all of the criteria of Proposition 4.6. Before doing so, we will need to bound s_w. Using the Taylor approximation of $\log(1 - x)$,

$$(1 - 2^{-w})^s = \exp(-s(2^{-w} + o(2^{-w}))).$$

As a result, we have that

$$s_w = \left\lfloor \frac{\ln(1/\varepsilon)}{2^{-w} + o(2^{-w})} \right\rfloor = \ln(1/\varepsilon) 2^w \cdot (1 + o_w(1)).$$

Let $k_w = ws_w$. Then $k_{w+1} = k_w \cdot (2 + o_w(1))$. Therefore, the value of $k = w^\star s_{w^\star}$ selected satisfies $k = \Theta(d)$.

By Fact E.3, $\Pr[f(\boldsymbol{x}) = -1] \geq \varepsilon$. Furthermore, as $(1 - 2^{-w^\star})^{s_{w^\star}+1} < \varepsilon$,

$$\begin{aligned}
\Pr[f(\boldsymbol{x}) = 1] &= (1 - \Pr[f(\boldsymbol{x}) = -1]) \\
&= 1 - (1 - 2^{-w^\star})^s_{w^\star} \\
&= 1 - \frac{(1 - 2^{-w^\star})^{s_{w^\star}+1}}{1 - 2^{-w^\star}} \\
&> 1 - \frac{\varepsilon}{1/2} \\
&\geq \frac{1}{3} \geq \varepsilon.
\end{aligned}$$

Lastly, we verify Equation (16). As $k = \Theta(d)$,

$$\begin{aligned}\frac{\log d}{d} &= \Theta\left(\frac{\log k}{k}\right)\\ &= \Theta\left(\frac{\log(\log(1/\varepsilon)\cdot w^\star \cdot 2^{w^\star})}{\log(1/\varepsilon)\cdot w^\star\cdot 2^{w^\star}}\right)\\ &\geq \Omega\left(\frac{\log(2^{w^\star})}{\log(1/\varepsilon)\cdot w^\star\cdot 2^{w^\star}}\right)\\ &= \Omega\left(\frac{1}{\log(1/\varepsilon)2^{w^\star}}\right)\end{aligned}$$

Applying the above, Fact E.3, and that $\Pr[f(\boldsymbol{x}) = -1] \leq 2\varepsilon$

$$\begin{aligned}\mathbb{E}[\boldsymbol{x}_1 f(\boldsymbol{x})] = \cdots = \mathbb{E}[\boldsymbol{x}_k f(\boldsymbol{x})] &= \frac{1}{2^{w^\star}-1}\cdot \Pr[f(\boldsymbol{x}) = -1]\\ &= O\left(\varepsilon\log(1/\varepsilon)\cdot\frac{\log d}{d}\right).\end{aligned}$$

□

F. Influence for monotone functions

Lemma F.1 (Influence = covariance for monotone functions)**.** *Let $\mathcal{D}_X = \mathcal{D}_X^{(1)} \times \ldots \times \mathcal{D}_X^{(d)}$ be an arbitrary product distribution over $\{\pm 1\}^d$. For a monotone function $f : \{\pm 1\}^d \to \{\pm 1\}$ and a feature $i \in [d]$, we have the identity $\mathrm{Inf}_i(f) = \mathrm{Cov}_{\mathcal{D}_X}[f(\boldsymbol{x}), \boldsymbol{x}_i]$.*

Proof. Let us denote $p_i = \Pr_{\mathcal{D}_X^{(i)}}[\boldsymbol{x}_i = 1]$ and $q_i = 1 - p_i$. We can further define $\alpha = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_X}[f(\boldsymbol{x}) \mid \boldsymbol{x}_i = 1]$ and $\beta = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_X}[f(\boldsymbol{x}) \mid \boldsymbol{x}_i = -1]$. We note that $\mathbb{E}_{\mathcal{D}_X}[f(\boldsymbol{x})\boldsymbol{x}_i] = p_i\alpha - q_i\beta$, and $\mathbb{E}_{\mathcal{D}_X}[f(\boldsymbol{x})] = p_i\alpha + q_i\beta$, and $\mathbb{E}_{\mathcal{D}_X}[\boldsymbol{x}_i] = p_i - q_i$.

We first expand the definition of covariance:

$$\begin{aligned}\mathrm{Cov}_{\mathcal{D}_X}[f(\boldsymbol{x}), \boldsymbol{x}_i] &= \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D}_X}[f(\boldsymbol{x})\boldsymbol{x}_i] - \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D}_X}[f(\boldsymbol{x})]\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D}_X}[\boldsymbol{x}_i] && \text{(definition of covariance)}\\ &= p_i\alpha - q_i\beta - (p_i\alpha + q_i\beta)(p_i - q_i) && \text{(shorthand)}\\ &= \alpha(p_i - p_i(p_i - q_i)) - \beta(q_i + q_i(p_i - q_i))\\ &= \alpha p_i(1 - p_i + q_i) - \beta q_i(1 - q_i + p_i)\\ &= 2p_iq_i(\alpha - \beta). && \text{(simplification)}\end{aligned}$$

We finish by showing that $\mathrm{Inf}_i(f)$ is equal to this last line:

$$\begin{aligned}\mathrm{Inf}_i(f) &= 2\cdot\Pr_{\substack{\boldsymbol{x}\sim\mathcal{D}_X\\ \boldsymbol{b}\sim\mathcal{D}_X^{(i)}}}[f(\boldsymbol{x}) \neq f(\boldsymbol{x}_{i=b})]\\ &= \mathop{\mathbb{E}}_{\substack{\boldsymbol{x}\sim\mathcal{D}_X\\ \boldsymbol{b}\sim\mathcal{D}_X^{(i)}}}[|f(\boldsymbol{x}) - f(\boldsymbol{x}_{i=b})|]\\ &= \Pr_{\substack{\boldsymbol{x}\sim\mathcal{D}_X\\ \boldsymbol{b}\sim\mathcal{D}_X^{(i)}}}[\boldsymbol{b} \neq \boldsymbol{x}_i]\cdot\mathop{\mathbb{E}}_{\substack{\boldsymbol{x}\sim\mathcal{D}_X\\ \boldsymbol{b}\sim\mathcal{D}_X^{(i)}}}[|f(\boldsymbol{x}) - f(\boldsymbol{x}_{i=\boldsymbol{b}})| \mid \boldsymbol{b} \neq \boldsymbol{x}_i]\\ &= 2p_i(1-p_i)\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D}_X}[f(\boldsymbol{x}_{i=1}) - f(\boldsymbol{x}_{i=-1})] && (f \text{ is monotone})\\ &= 2p_i(1-p_i)\left(\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D}_X}[f(\boldsymbol{x}) \mid \boldsymbol{x}_i = 1] - \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D}_X}[f(\boldsymbol{x}) \mid \boldsymbol{x}_i = -1]\right)\\ &= 2p_iq_i(\alpha - \beta).\end{aligned}$$

□

G. Local drop in $\mathcal{G}$ and covariance

Lemma G.1 (Local drop in $\mathcal{G}$ in terms of covariance)**.** *Let $\mathcal{E}$ be any distribution over $\mathcal{X} \times \{0,1\}$ and $h : \mathcal{X} \to \{0,1\}$ be a splitting function. Then:*

$$\Delta_{\mathcal{E}}(h) \geq 16 \cdot \mathrm{Cov}_{\mathcal{E}}[h(\boldsymbol{x}), \boldsymbol{y}]^2 \tag{17}$$

Proof. Let $\tau = \mathbb{E}_{\mathcal{E}}[h(\boldsymbol{x})]$ and $\delta = \mathbb{E}_{\mathcal{E}}[\boldsymbol{y} \mid h(\boldsymbol{x}) = 1] - \mathbb{E}_{\mathcal{E}}[\boldsymbol{y} \mid h(\boldsymbol{x}) = 0]$. Equation 20 of (Kearns & Mansour, 1996) states that $\Delta_{\mathcal{E}}(h) \geq 4\tau(1-\tau)\delta^2$. Then to prove this lemma, it suffices to show that $16\,\mathrm{Cov}_{\mathcal{E}}[h(\boldsymbol{x}), \boldsymbol{y}]^2 \leq 4\tau(1-\tau)\delta^2$. We expand the definition of covariance:

$$\begin{aligned}
&\mathrm{Cov}[h(\boldsymbol{x}), \boldsymbol{y}] = \mathbb{E}[h(\boldsymbol{x})\boldsymbol{y}] - \mathbb{E}[h(\boldsymbol{x})]\,\mathbb{E}[\boldsymbol{y}] \\
&\quad = \mathbb{E}[\boldsymbol{y} \mid h(\boldsymbol{x}) = 1]\,\mathbb{E}[h(\boldsymbol{x})] - \mathbb{E}[h(\boldsymbol{x})]\,\mathbb{E}[\boldsymbol{y}] \\
&\quad = \mathbb{E}[h(\boldsymbol{x})](\mathbb{E}[\boldsymbol{y} \mid h(\boldsymbol{x}) = 1] - \mathbb{E}[\boldsymbol{y}]) \\
&\quad = \mathbb{E}[h(\boldsymbol{x})](\mathbb{E}[\boldsymbol{y} \mid h(\boldsymbol{x}) = 1] - \mathbb{E}[\boldsymbol{y} \mid h(\boldsymbol{x}) = 1]\,\mathbb{E}[h(\boldsymbol{x})] \\
&\qquad - \mathbb{E}[\boldsymbol{y} \mid h(\boldsymbol{x}) = 0](1 - \mathbb{E}[h(\boldsymbol{x})])) \\
&\quad = \mathbb{E}[h(\boldsymbol{x})](1 - \mathbb{E}[h(\boldsymbol{x})]) \\
&\qquad (\mathbb{E}[\boldsymbol{y} \mid h(\boldsymbol{x}) = 1] - \mathbb{E}[\boldsymbol{y} \mid h(\boldsymbol{x}) = 0]) \\
&\quad = \tau(1-\tau)\delta.
\end{aligned}$$

The lemma follows from the fact that $\tau(1-\tau) \leq 1/4$. □