Provable Guarantees for Decision Tree Induction: The Agnostic Setting

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

Abstract

We give strengthened provable guarantees on the performance of widely employed and empirically successful *top-down decision tree learning heuristics*. While prior works have focused on the realizable setting, we consider the more realistic and challenging *agnostic* setting. We show that for all monotone functions f and $s \in \mathbb{N}$, these heuristics construct a decision tree of size $s^{\tilde{O}((\log s)/\varepsilon^2)}$ that achieves error $\leq \mathsf{opt}_s + \varepsilon$, where opt_s denotes the error of the optimal size-s decision tree for f. Previously such a guarantee was not known to be achievable by any algorithm, even one that is not based on top-down heuristics. We complement our algorithmic guarantee with a near-matching $s^{\tilde{\Omega}(\log s)}$ lower bound.

1. Introduction

This paper is motivated by the goal of establishing strong provable guarantees for the class of popular and empirically successful *top-down decision tree learning heuristics*. This includes well-known instantiations such as ID3 (Quinlan, 1986), its successor C4.5 (Quinlan, 1993), and CART (Breiman, 2017), all widely employed in everyday machine learning applications. These simple heuristics are also at the heart of more sophisticated decision-tree-based algorithms such as random forests (Breiman, 2001) and XGBoost (Chen & Guestrin, 2016), which have quickly gained prominence in Kaggle and other data science competitions, and achieve state-of-the-art performance for diverse tasks.

We will soon formally describe the learning-theoretic framework within which we study these heuristics, mentioning for now that they build a decision tree T for a binary classifier $f : \mathbb{R}^n \to \{0,1\}$ in a *greedy*, *top-down* fashion:

1. Query $\mathbb{1}[x_i \geq \theta]$ at the root of T, where x_i and θ are chosen to maximize the *purity gain*

$$\mathscr{G}(\mathbb{E}[f]) - \Big(\Pr[\boldsymbol{x}_i \geq \theta] \cdot \mathscr{G}(\mathbb{E}[f_{x_i \geq \theta}]) + \Pr[\boldsymbol{x}_i < \theta] \cdot \mathscr{G}(\mathbb{E}[f_{x_i < \theta}])\Big)$$

with respect to an *impurity function* $\mathscr{G} : [0,1] \to [0,1]$. This carefully chosen function $\mathscr{G}$ encapsulates the *splitting criterion* of the heuristic.

2. Build the left and right subtrees of T by recursing on $f_{x_i \geq \theta}$ and $f_{x_i < \theta}$ respectively.

Different instantiations of this simple approach are distinguished by different impurity functions $\mathscr{G}$, which determine the *order* in which the recursive calls are made. For example, ID3 and C4.5 uses the binary entropy function $\mathscr{G}(p) = \mathrm{H}(p)$; CART uses the *Gini criterion* $\mathscr{G}(p) = 4p(1-p)$; Kearns and Mansour proposed and analyzed the function $\mathscr{G}(p) = 2\sqrt{p(1-p)}$ (Kearns & Mansour, 1999; Dietterich et al., 1996).

Even without specifying the impurity function $\mathscr{G}$, it is well known and easy to see that *any* such heuristic can fare poorly even for simple functions f, in the sense of building a decision tree that is much larger than the optimal one. Consider the most basic setting of binary features, the uniform distribution over inputs, and f being the parity of two variables $f(x) = x_1 \oplus x_2$. This function can be computed by a decision tree of size 4, but since $\mathbb{E}[f] = \mathbb{E}[f_{x_i \geq \theta}] = \mathbb{E}[f_{x_i < \theta}] = 0$ for all x_i and θ, *any* top-down heuristic—regardless of the impurity function $\mathscr{G}$—may build a tree of size $\Omega(2^n)$ before achieving any non-trivial accuracy.

Monotonicity and the conjectures of Fiat, Pechyony, and Lee. In light of such examples, a question suggests itself: can we identify natural and expressive classes of functions for which strong provable guarantees on the performance of these top-down heuristics *can* be obtained?

[*]Equal contribution [1]Stanford University. Correspondence to: Guy Blanc <gblanc@stanford.edu>, Jane Lange <jlange20@stanford.edu>, Li-Yang Tan <liyang@cs.stanford.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

The first results in this direction were given by Fiat and Pechyony (Fiat & Pechyony, 2004; Pechyony, 2004), who considered the case of binary features (i.e. $f : \{\pm 1\}^n \to \{0,1\}$) and showed a strong positive result for halfspaces and read-once DNF formulas. For such functions f, they showed that these heuristics build a decision tree of *optimal* size that compute f *exactly*. Furthermore, and most relevant to our work, they raised the intriguing possibility that *monotonicity* is the key property shared by these functions that enables such a guarantee.[1] They conjectured that for all monotone functions f, these heuristics build a decision tree of size "not far from minimal" that compute f exactly.

Subsequently, Lee (Lee, 2009) formulated a relaxation of Fiat and Pechyony's conjecture, allowing for an approximate representation rather than an exact representation. Lee conjectured that for all monotone functions $f : \{\pm 1\}^n \to \{0,1\}$ computable by size-s decision trees, these heuristics construct a decision tree of size $\mathrm{poly}(s, 1/\varepsilon)$ that achieves error ε with respect to the uniform distribution over inputs. (The author further remarked that even a bound that is $\mathrm{poly}(s)$ for constant values of ε "would be a huge advance.")

These conjectures of (Fiat & Pechyony, 2004) and (Lee, 2009) are especially appealing because monotonicity, beyond just being a natural assumption that excludes parity and "parity-like" functions, is an independently important and intensively-studied property in both the theory and practice of machine learning. Many real-world data sets are naturally monotone in their features. In learning theory, even restricting our attention just to uniform-distribution learning, there is a large body of work on learning monotone functions (Hancock & Mansour, 1991; Kearns & Valiant, 1994; Kearns et al., 1994; Bshouty, 1995; Bshouty & Tamon, 1996; Blum et al., 1998; Verbeurgt, 1998; Sakai & Maruoka, 2000; Servedio, 2004; O'Donnell & Servedio, 2007; Sellie, 2008; Dachman-Soled et al., 2009; Lee, 2009; Jackson et al., 2011; O'Donnell & Wimmer, 2013; Dachman-Soled et al., 2015). Partly responsible for this popularity is the fact that monotonicity allows one to sidestep the well-known *statistical query* lower bounds (Blum et al., 1994) that hold for many simple concept classes.

The work of (Blanc et al., 2020). Recent work of Blanc, Lange, and Tan established a weak version of (Lee, 2009)'s conjecture. For all monotone functions $f : \{\pm 1\}^n \to \{0,1\}$ that are computable by size-s decision trees, they showed that a close variant of these top-down heuristics constructs a decision tree of size $s^{O(\sqrt{\log s}/\varepsilon)}$ that achieves error ε with respect to the uniform distribution over inputs. (Blanc et al., 2020) also showed that the dependence on 's' *cannot* be made polynomial, thereby disproving (Lee, 2009)'s actual conjecture (and (Fiat & Pechyony, 2004)'s even stronger conjecture): for all sizes $s \leq 2^{\tilde{O}(n^{4/5})}$ and error parameters $\varepsilon \in (0, \frac{1}{2})$, they exhibited a monotone function f that is computable by a size-s decision tree, and showed that all top-down impurity-based heuristics have to build a decision tree of size $s^{\tilde{\Omega}(\sqrt[4]{\log s})}$ in order to achieve error ε.

Our contributions. We give strengthened provable guarantees on the performance of top-down decision tree learning heuristics. The three main contributions of our work are:

1. We consider the more realistic and challenging *agnostic* setting, where f is an *arbitrary* monotone function. Prior works focused on the realizable setting, and their results relied on assumptions about the computational complexity of f (i.e. that f is computable by a small decision tree, or a halfspace, or a read-once DNF formula, etc.).

2. We establish provable guarantees that apply to all top-down heuristics, including ID3, C4.5, and CART. (Blanc et al., 2020)'s analysis, on the other hand, dealt with a specific *variant* of these heuristics, one whose splitting criterion does not correspond to any impurity function $\mathscr{G}$. (As a secondary contribution, we further show that (Blanc et al., 2020)'s guarantees in the realizable setting also hold for all top-down heuristics.)

3. Our analysis extends to classifiers for *real-valued* features (i.e. $f : \mathbb{R}^n \to \{0,1\}$) and arbitrary product distributions over inputs, whereas prior works dealt with classifiers for binary features (i.e $f : \{\pm 1\}^n \to \{0,1\}$) and mostly focused on the uniform distribution over inputs. Trees for real-valued features branch on queries of the form $\mathbb{1}[x_i \geq \theta]$ for some $\theta \in \mathbb{R}$, whereas trees for binary features branch on queries of the form $\mathbb{1}[x_i = 1]$.

Our main result is as follows:

Theorem 1 (Our main result; informal)**.** *Let $f : \mathbb{R}^n \to \{0,1\}$ be a monotone function and $\mathcal{D}$ be a product distribution over $\mathbb{R}^n$. For $s \in \mathbb{N}$, let opt_s denote the error of the best size-s balanced decision tree for f with respect to $\mathcal{D}$.*[2] *Let $\mathscr{G}$ be any impurity function. For $t \leq s^{\tilde{O}((\log s)/\varepsilon^2)}$,*

[1] We consider a function to be monotone if it is either non-decreasing or non-increasing in every coordinate; we do not require the direction to be the same for all coordinates. (Such functions are sometimes also called "unate.") See Definition 5.

[2] A balanced decision tree of size s is one that has depth $O(\log s)$. This technical assumption is not necessary in the case of binary features and the uniform distribution over inputs (see Theorem 2); it is only necessary for the general setting of real-valued features and arbitrary product distributions.

the size-t decision tree for f constructed by the top-down heuristic with $\mathscr{G}$ as its splitting criterion achieves error $\leq \mathsf{opt}_s + \varepsilon$ with respect to $\mathcal{D}$.

Previously, such a guarantee was not known to be achievable by any algorithm, even one that is not based on top-down heuristics (i.e. Theorem 1 represents the first algorithm for properly learning decision trees in an agnostic setting).

We complement Theorem 1 with a near-matching lower bound. We show that for all $s \leq 2^{\tilde{O}(\sqrt{n})}$, there is a monotone function f such that $\mathsf{opt}_s \leq 0.01$, and yet any top-down heuristic has to grow a tree of size $s^{\tilde{\Omega}(\log s)}$ to even achieve error ≤ 0.49 with respect to f. Taken together with (Blanc et al., 2020)'s $s^{O(\sqrt{\log s})}$ *upper* bound in the realizable setting, this exhibits a separation between the realizable and agnostic settings.

1.1. Formal Statements of our Results

We define a *partial tree* to be a decision tree with unlabeled leaves, and write T° to denote such trees. We refer to any decision tree T obtained from T° by a labeling of its leaves as a *completion* of T°. Given a tree a partial tree T° and a function f, there is a canonical completion of T° that minimizes the approximation error with respect to f:

Definition 1 (f-completion of a partial tree)**.** *Let T° be a partial tree and $f : \mathbb{R}^n \to \{0,1\}$. Consider the following completion of T°: for every leaf ℓ in T°, label it $\mathrm{round}(\mathbb{E}[f_\ell])$, where f_ℓ denotes the restriction of f by the path leading to ℓ and $\mathrm{round}(p) = 1$ if $p \geq \frac{1}{2}$ and 0 otherwise. This completion minimizes the approximation error $\Pr[T(\boldsymbol{x}) \neq f(\boldsymbol{x})]$; we refer to it as the f-completion of T° and denote it as T°_f.*

Definition 2 (Impurity functions and strong concavity)**.** *An impurity function $\mathscr{G} : [0,1] \to [0,1]$ is a concave function that is symmetric around $\frac{1}{2}$, and satisfies $\mathscr{G}(0) = \mathscr{G}(1) = 0$ and $\mathscr{G}(\frac{1}{2}) = 1$. We say that $\mathscr{G}$ is κ-strongly concave if for all $a, b \in [0,1]$,*

$$\frac{\mathscr{G}(a) + \mathscr{G}(b)}{2} \leq \mathscr{G}\left(\frac{a+b}{2}\right) - \frac{\kappa}{2} \cdot (b-a)^2.$$

Remark 1 (κ values of common impurity functions)**.** ID3 and C4.5 use binary entropy as their impurity function, which is κ-strongly concave for $\kappa = 1/\ln(2)$, or ≈ 1.4. Gini impurity, which is used by CART, is strongly concave for $\kappa = 2$, and (Kearns & Mansour, 1999)'s impurity function is strongly concave for $\kappa = 1$.

We now formally define the top-down heuristics that we study.

Definition 3 ($\mathscr{G}$-impurity of a partial tree)**.** *Let $\mathscr{G} : [0,1] \to [0,1]$ be an impurity function and $\mathcal{D}$ be a distribution over $\mathbb{R}^n$. For $f : \mathbb{R}^n \to \{0,1\}$ and a partial tree T°, the $\mathscr{G}$-impurity of T° with respect to f is defined to be*

$$\mathscr{G}\text{-impurity}_{f,\mathcal{D}}(T^\circ) := \sum_{\text{leaves } \ell \in T^\circ} \Pr_{\boldsymbol{x} \sim \mathcal{D}}[\,\boldsymbol{x} \text{ reaches } \ell\,] \cdot \mathscr{G}(\mathbb{E}[f_\ell]).$$

If T° is a partial tree and ℓ is a leaf of T°, we write $T^\circ_{\ell, \mathbb{1}[x_i \geq \theta]}$ to denote the extension of T° obtained by splitting ℓ with a query to $\mathbb{1}[x_i \geq \theta]$. The following algorithm captures the top-down decision tree learning heuristics that we study in this work:

> BUILDTOPDOWNDT$_{\mathscr{G},\mathcal{D}}(f,t)$:
>
> Initialize T° to be the empty tree.
>
> while (size$(T^\circ) < t$) {
>
> Grow T° by splitting leaf ℓ with a query to $\mathbb{1}[x_i \geq \theta]$, where ℓ and $\mathbb{1}[x_i \geq \theta]$ maximize:
>
> $$\mathscr{G}\text{-impurity}_{f,\mathcal{D}}(T^\circ) - \mathscr{G}\text{-impurity}_{f,\mathcal{D}}(T^\circ_{\ell, \mathbb{1}[x_i \geq \theta]}),$$
>
> the *purity gain* with respect to $\mathscr{G}$ and $\mathcal{D}$.
>
> }
>
> Output the f-completion of T°.

Figure 1. Top-down heuristic for building a size-t decision tree approximation of a function $f : \mathbb{R}^n \to \{0,1\}$, using the impurity function $\mathscr{G}$ as its splitting criterion.

We write TOPDOWNERROR$_{\mathscr{G},\mathcal{D}}(f,t)$ to denote the error $\Pr[T(\boldsymbol{x}) \neq f(\boldsymbol{x})]$, where T is the size-t tree constructed by BUILDTOPDOWNDT$_{\mathscr{G},\mathcal{D}}(f,t)$.

Definition 4 (opt_s)**.** *For a function $f : \mathbb{R}^n \to \{0,1\}$, a distribution $\mathcal{D}$ over $\mathbb{R}^n$, and an integer $s \in \mathbb{N}$, we write $\mathsf{opt}_{f,\mathcal{D},s} \in [0, \frac{1}{2}]$ to denote the error of the best size-s decision tree for f:*

$$\mathsf{opt}_{f,\mathcal{D},s} := \min\left\{ \Pr_{\boldsymbol{x} \sim \mathcal{D}}[T(\boldsymbol{x}) \neq f(\boldsymbol{x})] \colon T \text{ is a size-}s \text{ decision tree} \right\}.$$

When f and $\mathcal{D}$ are clear from context, we simply write opt_s.

Our main algorithmic guarantee. We first state and prove our results in the setting of binary features and the uniform distribution over inputs. As alluded to in the introduction, in Section 2.2 we will show how our results in this specialized setting extend to the more general setting of real-valued features and arbitrary product distributions over inputs.

Theorem 2 (Our main algorithmic guarantee for binary features and the uniform distribution)**.** *Let $f : \{\pm 1\}^n \to \{0,1\}$ be a monotone function, $\mathcal{U}$ be the uniform distribution over $\{\pm 1\}^n$, and $\mathscr{G} : [0,1] \to [0,1]$ be an κ-strongly concave impurity function. For all $s \in \mathbb{N}$ and $\varepsilon \in (0, \frac{1}{2})$,*

$$\textsc{TopDownError}_{\mathscr{G},\mathcal{U}}(f, s^{O(\log s)/\kappa\varepsilon^2}) \leq \mathsf{opt}_s + \varepsilon.$$

In words, Theorem 2 says that for $s \in \mathbb{N}$, the decision tree of size $s^{O(\log s)}$ constructed by BUILDTOPDOWNDT achieves error that nearly matches that of the best size-s decision tree for f.

A near-matching lower bound. We contrast Theorem 2 with (Blanc et al., 2020)'s result in the *realizable* setting: if f is a monotone function that is computable by a size-s decision tree (i.e. $\mathsf{opt}_s = 0$), then a variant of the top-down heuristics grows a tree of size $s^{O(\sqrt{\log s})}$ that achieves error ε.[3] With this in mind, it is natural to wonder if the parameters of Theorem 2 can be improved to $\textsc{TopDownError}_{\mathscr{G},\mathcal{U}}(f, s^{O_{\kappa,\varepsilon}(\sqrt{\log s})}) \leq \mathsf{opt}_s + \varepsilon$. We complement Theorem 2 with a lower bound that rules out such an improvement. We show that the dependence on 's' in Theorem 2 is in fact near-optimal:

Theorem 3 (Our main lower bound)**.** *For all $s \leq 2^{\tilde{O}(\sqrt{n})}$, there is a monotone function $f : \{\pm 1\}^n \to \{0,1\}$ such that $\mathsf{opt}_s \leq 0.01$ with respect to the uniform distribution over inputs, and $\textsc{TopDownError}_{\mathscr{G},\mathcal{U}}(f, s^{\tilde{\Omega}(\log s)}) \geq 0.49$ for any impurity function $\mathscr{G}$.*

Taken together with (Blanc et al., 2020)'s $s^{O(\sqrt{\log s})}$ upper bound in the realizable setting, Theorem 3 exhibits a separation between the realizable and agnostic settings.

1.2. Related Work

Kearns and Mansour (Kearns, 1996; Kearns & Mansour, 1999) were the first to study top-down decision tree learning heuristics using the framework of learning theory. They showed that these heuristics can be viewed as *boosting algorithms*, where one views the functions queried at the internal nodes of the tree (single variables in our case) as weak hypotheses. As is standard in results on boosting, their results are *conditional* in nature: they assume the existence of weak hypotheses for filtered-and-rebalanced versions of the original distribution (what they call "The Weak Hypothesis Assmption"), and they show how these top-down heuristics build a decision tree that combines these weak hypotheses into a strong one. Dietterich, Kearns, and Mansour (Dietterich et al., 1996) gave an experimental comparison of the impurity functions used by ID3, C4.5, and CART, along with a new impurity function that (Kearns & Mansour, 1999) had proposed.

Fiat and Pechyony (Fiat & Pechyony, 2004; Pechyony, 2004) studied functions f that are computable by linear threshold functions or read-once DNF formulas, and showed that these heuristics build a decision tree of optimal size that compute f exactly. Recent work of Brutzkus, Daniely, and Malach (Brutzkus et al., 2019a) studies functions f that are conjunctions and read-once DNF formulas, and gives theoretical and empirical evidence showing that for such functions, the variant of ID3 proposed by (Kearns & Mansour, 1999), when run for t iterations, grows a tree that achieves accuracy that matches or nearly matches that of the best size-t tree for f. Concurrent work by the same authors (Brutzkus et al., 2019b) shows that ID3 learns $(\log n)$-juntas in the setting of smoothed analysis.

Learning algorithms not based on top-down heuristics: improper algorithms for learning decision trees. O'Donnell and Servedio (O'Donnell & Servedio, 2007) gave a $\mathrm{poly}(n, s^{1/\varepsilon^2})$-time uniform-distribution algorithm for learning monotone functions computable by size-s decision trees. This remains the fastest algorithm for the realizable setting. (O'Donnell & Servedio, 2007)'s algorithm is not based on the top-down heuristics that are the focus of our work (and the others discussed above); indeed, their algorithm does not output a decision tree as its hypothesis (i.e. it is not a proper learning algorithm). For the agnostic setting, the results of Kalai, Klivans, Mansour, and Servedio (Kalai et al., 2008) can be used to give a uniform-distribution algorithm that runs in time $n^{O(\log(s/\varepsilon))}$ and outputs a hypothesis that achieves error $\mathsf{opt}_s + \varepsilon$. Compared to Theorem 2, this algorithm does not require f to be monotone, but like (O'Donnell & Servedio, 2007)'s algorithm it is also improper. The work of (Gopalan et al., 2008) gives a uniform-distribution algorithm that runs in $\mathrm{poly}(n, s, 1/\varepsilon)$ time; however, their algorithm requires the use of *membership queries*, and is also improper. Furthermore, all the results discussed in this paragraph only hold in the setting of binary features and with respect to the uniform distribution over inputs.

1.3. Preliminaries

We use **boldface** (e.g. $\boldsymbol{x} \sim \mathbb{R}^n$) to denote random variables. Given two functions $f, g : \mathbb{R}^n \to \{0,1\}$, we write $\mathrm{dist}(f,g) := \Pr[f(\boldsymbol{x}) \neq g(\boldsymbol{x})]$ to denote the distance between f and g. We write $\mathrm{bias}(f) := \min\{\Pr[f(\boldsymbol{x}) = 0], \Pr[f(\boldsymbol{x}) = 1]\}$ to denote the distance of f to the closest constant function; equivalently, $\mathrm{bias}(f) = \Pr[f(\boldsymbol{x}) \neq \mathrm{round}(\mathbb{E}[f])]$, where $\mathrm{round}(p) = 1$ if $p \geq \frac{1}{2}$ and 0 other-

[3] The heuristic that (Blanc et al., 2020) analyzes does not correspond to any impurity function $\mathscr{G}$. As mentioned in the introduction, in Appendix C we show that (Blanc et al., 2020)'s guarantees on their variant of the top-down heuristics in fact hold for all top-down heuristics.

wise. If ℓ is a leaf in a decision tree, we write $|\ell|$ to denote the depth of ℓ within the tree.

Definition 5 (Monotone functions)**.** *We say that a function $f : \mathbb{R}^n \to \{0,1\}$ is monotone if for all coordinates $i \in [n]$, either*

- *f is non-decreasing in the i-th direction: $f(x) \le f(y)$ for all $x, y \in \mathbb{R}^n$ such that $x_i \le y_i$, or*
- *f is non-increasing in the i-th direction: $f(x) \ge f(y)$ for all $x, y \in \mathbb{R}^n$ such that $x_i \le y_i$.*

Organization of this paper. We give the complete proof of Theorem 2 in the next section. Due to space constraints, proofs of the extension of Theorem 2 to Theorem 1 (from the specific setting of binary features and the uniform distribution over inputs, to the general setting of real-valued features and arbitrary product distributions over inputs), and that of Theorem 3 (a lower bound showing that Theorem 1 is nearly optimal) are deferred to the appendix.

2. Our Main Algorithmic Result: Theorem 2

Recall that Theorem 2 is concerned with the special case of binary features and the uniform distribution over inputs. Therefore the trees that we reason about in the proof of Theorem 2 split on queries of the form $\mathbb{1}[x_i = 1]$ (rather than queries of the form $\mathbb{1}[x_i \ge \theta]$ as in the general setting of real-valued features). Also, all probabilities and expectations in this proof are with respect to the uniform distribution over inputs.

Definition 6 (Influence)**.** *Given a function $f : \{\pm 1\}^n \to \{0,1\}$, the influence of coordinate $i \in [n]$ on f is defined to be*

$$\mathrm{Inf}_i(f) \coloneqq \Pr[f(\boldsymbol{x}) \neq f(\boldsymbol{x}^{\oplus i})],$$

where $\boldsymbol{x}^{\oplus i}$ denotes $\boldsymbol{x}$ with its i-coordinate flipped, and $\boldsymbol{x} \sim \{\pm 1\}$ is uniform random. The total influence of f is defined to be $\mathrm{Inf}(f) \coloneqq \sum_{i=1}^n \mathrm{Inf}_i(f)$.

A key technical ingredient in our proof will be an inequality of Jain and Zhang (Jain & Zhang, 2011), a robust version of the powerful O'Donnell–Saks–Schramm–Servedio inequality from the Fourier analysis of boolean functions (O'Donnell et al., 2005). The following is a special case of Corollary 1.4 of (Jain & Zhang, 2011):

Theorem 4 (Robust OSSS inequality)**.** *Let $f : \{\pm 1\}^n \to \{0,1\}$ be any function, and $g : \{\pm 1\}^n \to \{0,1\}$ be a size-s decision tree. Then*

$$\max_{i \in [n]} \{\mathrm{Inf}_i(f)\} \ge \frac{\mathrm{bias}(f) - \mathrm{dist}(f,g)}{\log s}.$$

Remark 2 (Context for Theorem 4)**.** The original OSSS inequality essentially corresponds to the special case of Theorem 4 where $f \equiv g$: if f is a size-s decision tree, then

$$\max_{i \in [n]} \{\mathrm{Inf}_i(f)\} \ge \frac{\mathrm{Var}(f)}{\log s}.$$

The OSSS inequality can be viewed as a variant of the famed Kahn–Kalai–Linial inequality (Kahn et al., 1988), one that takes into account the computational complexity of f. In (O'Donnell et al., 2005) the authors also gave a robust version of their inequality that is qualitative similar to Theorem 4 (see the discussion following Theorem 3.2 of (O'Donnell et al., 2005)): under the assumptions of Theorem 4,

$$\max_{i \in [n]} \{\mathrm{Inf}_i(f)\} \ge \frac{\mathrm{Var}(f) - 2 \cdot \mathrm{dist}(f,g)}{\log s}.$$

This inequality can be used in place of Theorem 4 to prove a statement that is qualitatively similar to Theorem 2, but with a weaker error bound of $O(\mathsf{opt}_s) + \varepsilon$ rather than the $\mathsf{opt}_s + \varepsilon$ of Theorem 2.

Fact 2.1 (Influence $\equiv$ correlation for monotone functions)**.** *Let $f : \{\pm 1\}^n \to \{0,1\}$ be a monotone function. Then $\mathrm{Inf}_i(f) = 2 \cdot |\mathbb{E}[f(\boldsymbol{x})\boldsymbol{x}_i]|$ for all $i \in [n]$.*

Proposition 2.2 (Influence and purity gain)**.** *For all κ-strongly concave impurity heuristics $\mathscr{G}$, all monotone functions $f : \{\pm 1\}^n \to \{0,1\}$, and all coordinates $i \in [n]$,*

$$\mathop{\mathbb{E}}_{\boldsymbol{b} \sim \{\pm 1\}}[\mathscr{G}(\mathbb{E}[f_{x_i = \boldsymbol{b}}])] \le \mathscr{G}(\mathbb{E}[f]) - \frac{\kappa}{32} \cdot \mathrm{Inf}_i(f)^2.$$

Proof. Note that:

$$\mathbb{E}[f(\boldsymbol{x})\boldsymbol{x}_i] = \tfrac{1}{2}(\mathbb{E}[f_{x_i=1}] - \mathbb{E}[f_{x_i=-1}])$$

and that:

$$\mathbb{E}[f] = \tfrac{1}{2}(\mathbb{E}[f_{x_i=1}] + \mathbb{E}[f_{x_i=-1}]).$$

The desired result therefore holds as a direct consequence of the κ-strong concavity of $\mathscr{G}$ and Fact 2.1. □

2.1. Proof of Theorem 2

Let $f : \{\pm 1\}^n \to \{0,1\}$ be a monotone function and g be a size-s decision tree for which $\mathrm{dist}(f,g) = \mathsf{opt}_s$. Fix some κ-strongly concave impurity heuristic $\mathscr{G}$. Given a partial tree T° (which we should think of as the approximator for f that is being built by BUILDTOPDOWNDT$_{\mathscr{G},\mathcal{U}}$), we define the potential function:

$$\mathscr{G}\text{-impurity}_f(T^\circ) \coloneqq \sum_{\text{leaves } \ell \in T^\circ} 2^{-|\ell|} \cdot \mathscr{G}(\mathbb{E}[f_\ell]).$$

The next lemma records a few useful properties of this potential function $\mathscr{G}$-impurity$_f$. (This lemma can be viewed

as our analogue of (Blanc et al., 2020)'s Lemma 5.1. (Blanc et al., 2020) worked with a potential function that is a variant of ours which has $\mathrm{Inf}(f_\ell)$ in place of $\mathscr{G}(\mathbb{E}[f_\ell])$.)

Definition 7 (Purity gain)**.** *Let $f : \{\pm 1\}^n \to \{0,1\}$ be a function and $\mathscr{G}$ be an impurity heuristic. For a partial tree T° and a leaf ℓ of T°, we write T°_{ℓ,x_i} to denote the extension of T° obtained by splitting ℓ with a query to x_i. The purity gain associated with splitting ℓ with a query to x_i is defined as*

$$\begin{aligned}&\mathscr{G}\text{-purity-gain}_f(T^\circ, \ell, x_i)\\ &:= \mathscr{G}\text{-impurity}_f(T^\circ) - \mathscr{G}\text{-impurity}_f(T^\circ_{\ell,x_i}).\end{aligned}$$

It is easy to verify that given a leaf ℓ, the variable associated with the largest purity gain is exactly one that is most correlated with f_ℓ:

Proposition 2.3 (Proposition 7.7 from (Blanc et al., 2020))**.** *For any leaf ℓ of T°, let $x_{i^\star}$ be the variable that maximizes $\mathscr{G}$-purity-gain$_{\mathscr{G}}(\ell, x_i)$ among all $i \in [n]$. Then,*

$$\mathbb{E}[f_\ell(\boldsymbol{x})\boldsymbol{x}_{i^\star}] \geq \mathbb{E}[f_\ell(\boldsymbol{x})\boldsymbol{x}_j] \quad \textit{for all } j \in [n].$$

Lemma 2.4 (Useful properties of $\mathscr{G}$-impurity$_f$)**.**

1. $\mathscr{G}$-impurity$_f$(*empty tree*) $= \mathscr{G}(\mathbb{E}[f]) \leq 1$.
2. $\mathrm{dist}(f, T^\circ_f) \leq \mathscr{G}$-impurity$_f(T^\circ)$.
3. *For any leaf ℓ of T° and variable $i \in [n]$,*

$$\mathscr{G}\text{-purity-gain}_f(T^\circ, \ell, x_i) \geq 2^{-|\ell|} \cdot \frac{\kappa}{32} \cdot \mathrm{Inf}_i(f_\ell)^2.$$

Proof. The first claim follows from the definition of $\mathscr{G}$-impurity$_f$. For the second claim, we have that

$$\begin{aligned}\mathrm{dist}(f, T^\circ_f) &= \sum_{\text{leaves } \ell \in T^\circ} \Pr[\,\boldsymbol{x} \text{ reaches } \ell\,] \cdot \mathrm{bias}(f_\ell)\\ &= \sum_{\text{leaves } \ell \in T^\circ} 2^{-|\ell|} \cdot \mathrm{bias}(f_\ell)\\ &\leq \sum_{\text{leaves } \ell \in T^\circ} 2^{-|\ell|} \cdot \mathscr{G}(\mathbb{E}[f_\ell]) \quad \text{(Definition 2)}\\ &= \mathscr{G}\text{-impurity}_f(T^\circ).\end{aligned}$$

As for the third claim, we have that

$$\begin{aligned}&\mathscr{G}\text{-purity-gain}_f(T^\circ, \ell, x_i)\\ &= 2^{-|\ell|}\left(\mathscr{G}(\mathbb{E}[f_\ell]) - \mathop{\mathbb{E}}_{\boldsymbol{b}\sim\{\pm 1\}}[\mathscr{G}(\mathbb{E}[(f_\ell)_{x_i=\boldsymbol{b}}])\right)\Bigg(\\ &\geq 2^{-|\ell|} \cdot \frac{\kappa}{32} \cdot \mathrm{Inf}_i(f_\ell)^2,\end{aligned}$$

where the final inequality is by Proposition 2.2. □

The following simple fact states that the error of the f-completion of a partial tree T° cannot increase with further splits:

Fact 2.5 (Splits cannot increase error)**.** *Let T° be a partial tree, and $\widetilde{T}^\circ$ be the partial tree that results from splitting a leaf of T°. Then* $\mathrm{dist}(f, \widetilde{T}^\circ_f) \leq \mathrm{dist}(f, T^\circ_f)$.

We are now ready to prove Theorem 2. By Fact 2.5, it suffices to show that error $\mathsf{opt}_s + \varepsilon$ is achieved after at most $s^{O(\log s)/\varepsilon^2}$ iterations/splits. We do so by lower bounding the score of the leaf that is split by BUILDTOPDOWNDT in each iteration *before* error $\mathsf{opt}_s + \varepsilon$ is achieved. Let T° be the size-$(j+1)$ partial tree that is built by BUILDTOPDOWNDT after j iterations. Suppose $\mathrm{dist}(f, T^\circ_f) > \mathsf{opt}_s + \varepsilon$, and let $\ell^\star$, $x_i^\star$ be the leaf and variable of T° that is split in the $(j+1)$-st iteration. We claim that

$$\mathscr{G}\text{-purity-gain}_f(T^\circ, \ell^\star, x_{i^\star}) > \frac{\kappa \cdot \varepsilon^2}{32 \cdot (j+1)(\log s)^2}. \quad (1)$$

Writing $x_{i(\ell)}$ to denote variable with the highest correlation with f_ℓ, we have that

$$\begin{aligned}&\sum_{\text{leaves } \ell \in T^\circ}\Bigg(\mathscr{G}\text{-purity-gain}_f(T^\circ, \ell, x_{i(\ell)})\\ &\geq \sum_{\text{leaves } \ell \in T^\circ}\Bigg(2^{-|\ell|} \cdot \frac{\kappa}{32} \cdot \mathrm{Inf}_{i(\ell)}(f_\ell)^2 \qquad \text{(Lemma 2.4)}\\ &\geq \sum_{\text{leaves } \ell \in T^\circ}\Bigg(2^{-|\ell|} \cdot \frac{\kappa}{32} \cdot \left(\frac{\mathrm{bias}(f_\ell) - \mathrm{dist}(f_\ell, g_\ell)}{\mathrm{size}(g_\ell)}\right)^2\\ &\qquad\qquad \text{(Theorem 4)}\\ &\geq \frac{\kappa}{32} \cdot \sum_{\text{leaves } \ell \in T^\circ}\Bigg(2^{-|\ell|} \cdot \left(\frac{\mathrm{bias}(f_\ell) - \mathrm{dist}(f_\ell, g_\ell)}{\log s}\right)^2\\ &\qquad\qquad (\mathrm{size}(g_\ell) \leq \mathrm{size}(g) = s)\\ &\geq \frac{\kappa}{32} \cdot \left[\sum_{\text{leaves } \ell \in T^\circ}\Bigg(2^{-|\ell|} \cdot \left(\frac{\mathrm{bias}(f_\ell) - \mathrm{dist}(f_\ell, g_\ell)}{\log s}\right)\right]^2\\ &\qquad\qquad \text{(Jensen's inequality)}\\ &= \frac{\kappa}{32} \cdot \frac{1}{(\log s)^2} \sum_{\text{leaves } \ell \in T^\circ} 2^{-|\ell|} \cdot \mathrm{bias}(f_\ell)\\ &\qquad\qquad - \sum_{\text{leaves } \ell \in T^\circ}\Bigg(2^{-|\ell|} \cdot \mathrm{dist}(f_\ell, g_\ell)\Bigg)^2\\ &= \frac{\kappa}{32} \cdot \frac{1}{(\log s)^2} \cdot \big(\mathrm{dist}(f, T^\circ_f) - \mathrm{dist}(f, g)\big)^2\\ &> \frac{\kappa}{32} \cdot \frac{1}{(\log s)^2} \cdot ((\mathsf{opt}_s + \varepsilon) - \mathsf{opt}_s)^2 \;=\; \frac{\kappa}{32} \cdot \left(\frac{\varepsilon}{\log s}\right)^2.\end{aligned}$$

It follows that there must be at least one leaf and variable with purity gain greater than $\kappa\varepsilon^2/32(j+1)(\log s)^2$. Since BUILDTOPDOWNDT splits the leaf and variable with the largest purity gain, this establishes Equation (1).

Writing $\widetilde{T}^\circ$ to denote the partial tree that is obtained after BUILDTOPDOWNDT makes the single split with largest purity gain, we have that

$$\begin{aligned}&\mathscr{G}\text{-impurity}_f(\widetilde{T}^\circ)\\ &\quad= \mathscr{G}\text{-impurity}_f(T^\circ) - \mathscr{G}\text{-purity-gain}_f(T^\circ, \ell^\star, x_{i^\star})\\ &\quad\le \mathscr{G}\text{-impurity}_f(T^\circ) - \frac{\kappa\cdot\varepsilon^2}{32\cdot(j+1)(\log s)^2}.\end{aligned}$$

Combining this with the first and second claims of Lemma 2.4, we have the following: the value of the potential function starts off at at most 1 with T° being the empty tree, decreases by at least $\kappa\cdot\varepsilon^2/32j(\log s)^2$ with the j-th split, and error $\mathsf{opt}_s+\varepsilon$ is achieved once this value drops below $\mathsf{opt}_s+\varepsilon$. Therefore, we can bound the number of splits necessary to ensure error $\mathsf{opt}_s+\varepsilon$ by the smallest t that satisfies:

$$\sum_{j=1}^{t}\left(\frac{\kappa\varepsilon^2}{32j(\log s)^2}\right) \ge 1-(\mathsf{opt}_s+\varepsilon).$$

Since

$$\sum_{j=1}^{t}\left(\frac{\kappa\varepsilon^2}{32j(\log s)^2}\right) \ge \frac{\kappa\varepsilon^2\log t}{32(\log s)^2},$$

we conclude that $t\le s^{O(\log s)/\kappa\varepsilon^2}$ suffices. This completes the proof of Theorem 2.

2.2. Trees for Real-Valued Features

We will prove an extension of Theorem 2 that applies to functions of real-valued features and arbitrary product distributions over inputs.

In order to state our result, we need to slightly restrict our definition of opt so it only considers "balanced" trees.

Definition 8 (Balanced tree)**.** *A decision tree T of depth d and size s is balanced if $d = O(\log s)$.*

Definition 9 ($\mathsf{balanced\_opt}_s$)**.** *For a function $f:\mathbb{R}^n\to\{0,1\}$, product distribution $\mathcal{D}$ over $\mathbb{R}^n$, and an integer $s\in\mathbb{N}$, we write $\mathsf{balanced\_opt}_{f,\mathcal{D},s}\in[0,\frac{1}{2}]$ to denote the error of the best balanced size-s decision tree for f:*

$$\mathsf{balanced\_opt}_{f,\mathcal{D},s} \coloneqq \min\left\{\Pr_{\boldsymbol{x}\sim\mathcal{D}}[T(\boldsymbol{x})\neq f(\boldsymbol{x})] : \begin{array}{l}T \text{ is a balanced size-}s\text{ decision tree}\end{array}\right\},$$

When f and $\mathcal{D}$ are clear from context, we simply write $\mathsf{balanced\_opt}_s$.

We can now formally state our main theorem.

Theorem 1. *Let $f:\mathbb{R}^n\to\{0,1\}$ be a monotone function and $\mathcal{D}$ be any product distribution over $\mathbb{R}^n$. For any κ-strongly concave impurity heuristic $\mathscr{G}$ and $s\in\mathbb{N}$,*

$$\begin{aligned}&\textsc{TopDownError}_{\mathscr{G},\mathcal{D}}(f, s^{\tilde{O}((\log s)/\varepsilon^2)})\\ &\le \mathsf{balanced\_opt}_s+\varepsilon.\end{aligned}$$

The proof of Theorem 1 will follow the same overall structure as our proof of Theorem 2. One key new ingredient is a generalization of Theorem 4 to real-valued features; this extension could be of independent interest:

Theorem 5 (Extension of (Jain & Zhang, 2011) to trees for real-valued features)**.** *Let $f:\mathbb{R}^n\to\{0,1\}$ be a monotone function, $\mathcal{D}$ be an arbitrary product distribution over $\mathbb{R}^n$, and T be a size-s balanced decision tree. Then, there exist $i^\star\in[n]$ and $\theta^\star\in\mathbb{R}$ for which $\Pr_{\boldsymbol{x}\sim D}[\boldsymbol{x}_{i^\star}\ge\theta^\star]=\frac{1}{2}$ and*

$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D}}\left[f(\boldsymbol{x})\cdot\mathbb{1}[\boldsymbol{x}_{i^\star}\ge\theta^\star]\right] \ge \Omega\left(\frac{\varepsilon}{\log(s)\log\log(s/\varepsilon)}\right)$$
$$\text{where } \varepsilon \coloneqq \mathrm{bias}(f)-\mathrm{dist}(f,T).$$

where $\mathrm{bias}(f)$ and $\mathrm{dist}(f,T)$ are also measured with respect to $\mathcal{D}$.

To prove Theorem 1, we apply Theorem 5 in the same way Theorem 4 is used to prove Theorem 2. The full proof of Theorem 1 is deferred to the appendix.

3. Conclusion

We have given strengthened provable guarantees on the performance of widely employed and empirically successful top-down decision tree learning heuristics such as ID3, C4.5, and CART. Compared to previous works, our guarantees: (1) hold in the more realistic and challenging agnostic setting; (2) apply to all top-down heuristics and their associated impurity functions; (3) extend to the setting of real-valued features and arbitrary product distributions over the domain. Our main result shows that for all monotone functions $f:\mathbb{R}^n\to\{0,1\}$ and $s\in\mathbb{N}$, these top-down heuristics build a tree of size $s^{\tilde{O}((\log s)/\varepsilon^2)}$ that achieves error within ε of that of the optimal balanced size-s decision tree for f. We complement this with a near-matching lower bound. While our work was primarily motivated by the goal of understanding top-down heuristics, our results yield new guarantees that are not known to be achievable by any other algorithm, even ones that are not based on top-down heuristics.

There are several concrete avenues for future work:

1. Beyond monotonicity. As mentioned in the introduction, any top-down heuristic will fare badly on the parity functions f, in the sense of building a tree that is much larger than the optimal tree for f. Though broad and natural, the class of monotone functions is not the only class that excludes the parity function. Another fundamental property to consider is *noise stability* (see §2.4 of (O'Donnell, 2014)) — what guarantees can be made about the performance of these top-down heuristics when run on noise-stable functions?

2. Beyond product distributions. In this work our results hold for arbitrary product distributions over the domain, extending previous work that focuses on the uniform distribution. Could we establish provable distribution-independent guarantees, or failing that, perhaps provable guarantees for distributions with limited dependencies between coordinates?

3. Polynomial-size approximating trees. Our lower bound (Theorem 3) shows a monotone function such that any top-down heuristics has to build a tree of size $s^{\tilde{\Omega}(\log s)}$ in order to achieve error $\leq \mathsf{opt}_s + \varepsilon$. Results of (Blanc et al., 2020) show a similar lower bound of $s^{\tilde{\Omega}(\sqrt[4]{\log s})}$ in the realizable setting. Are there broad and natural *subclasses* of monotone functions that evade these lower bounds, and for which polynomial size upper bounds do exist?

Acknowledgements

We thank Michael Kim and the ICML reviewers for their helpful feedback and suggestions. LYT is supported by NSF grant CCF-192179 and NSF CAREER award CCF-1942123.

References

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)*, 2020. 1, 2, 1, 1.1, 1.1, 3, 2.1, 2.3, 3, A.1, C, 2, C.1, C, 6, C

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. 1

Blum, A., Burch, C., and Langford, J. On learning monotone boolean functions. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 408–415, 1998. 1

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

Breiman, L. *Classification and regression trees*. Routledge, 2017. 1

Brutzkus, A., Daniely, A., and Malach, E. On the Optimality of Trees Generated by ID3. *ArXiv*, abs/1907.05444, 2019a. 1.2

Brutzkus, A., Daniely, A., and Malach, E. ID3 Learns Juntas for Smoothed Product Distributions. *ArXiv*, abs/1906.08654, 2019b. 1.2

Bshouty, N. Exact learning via the monotone theory. *Information and Computation*, 123(1):146–153, 1995. 1

Bshouty, N. and Tamon, C. On the Fourier spectrum of monotone functions. *Journal of the ACM*, 43(4):747–770, 1996. 1

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*, pp. 785–794, 2016. 1

Dachman-Soled, D., Lee, H. K., Malkin, T., Servedio, R. A., Wan, A., and Wee, H. Optimal cryptographic hardness of learning monotone functions. *Theory of Computing*, 5(13):257–282, 2009. doi: 10.4086/toc.2009.v005a013. URL http://www.theoryofcomputing.org/articles/v005a013. 1

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. 1

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. 1, 1.2

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. 1, 1.2

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. 1.2

Hancock, T. and Mansour, Y. Learning monotone k-μ DNF formulas on product distributions. In *Proceedings of the 4th Annual Conference on Computational Learning Theory (COLT)*, pp. 179–193, 1991. 1

Jackson, J., Lee, H., Servedio, R., and Wan, A. Learning Random Monotone DNF. *Discrete Applied Mathematics*, 159(5):259–271, 2011. 1

Jain, R. and Zhang, S. The influence lower bound via query elimination. *Theory of Computing*, 7(1):147–153, 2011. 2, 5

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. 2

Kalai, A., Klivans, A., Mansour, Y., and Servedio, R. A. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008. 1.2

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. 1.2

Kearns, M. and Mansour, Y. On the boosting ability of top-down decision tree learning algorithms. *Journal of Computer and System Sciences*, 58(1):109–128, 1999. 1, 1, 1.2

Kearns, M. and Valiant, L. Cryptographic limitations on learning Boolean formulae and finite automata. *Journal of the ACM*, 41(1):67–95, 1994. 1

Kearns, M., Li, M., and Valiant, L. Learning Boolean formulas. *Journal of the ACM*, 41(6):1298–1328, 1994. 1

Lee, H. *On the learnability of monotone functions*. PhD thesis, Columbia University, 2009. 1

O'Donnell, R. *Analysis of Boolean Functions*. Cambridge University Press, 2014. Available at `http://analysisofbooleanfunctions.net/`. 1, A

O'Donnell, R. and Servedio, R. Learning monotone decision trees in polynomial time. *SIAM Journal on Computing*, 37(3):827–844, 2007. 1, 1.2, C

O'Donnell, R. and Wimmer, K. KKL, Kruskal–Katona, and Monotone Nets. *SIAM Journal on Computing*, 42(6): 2375–2399, 2013. 1

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. 2, 2

Pechyony, D. Decision trees: More theoretical justification for practical algorithms. Master's thesis, Tel Aviv University, 2004. 1, 1.2

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

Quinlan, R. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 1558602402. 1

Sakai, Y. and Maruoka, A. Learning monotone log-term DNF formulas under the uniform distribution. *Theory of Computing Systems*, 33:17–33, 2000. 1

Sellie, L. Learning random monotone DNF under the uniform distribution. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, pp. 181–192, 2008. 1

Servedio, R. On learning monotone DNF under product distributions. *Information and Computation*, 193(1):57–74, 2004. 1

Verbeurgt, K. Learning sub-classes of monotone DNF on the uniform distribution. In *Proceedings of the 9th Conference on Algorithmic Learning Theory (ALT)*, pp. 385–399, 1998. 1