Decision Tree Heuristics Can Fail, Even in the Smoothed Setting

Guy Blanc
Stanford University, CA, USA

Jane Lange
Massachusetts Institute of Technology, Cambridge, MA, USA

Mingda Qiao
Stanford University, CA, USA

Li-Yang Tan
Stanford University, CA, USA

Abstract

Greedy decision tree learning heuristics are mainstays of machine learning practice, but theoretical justification for their empirical success remains elusive. In fact, it has long been known that there are simple target functions for which they fail badly (Kearns and Mansour, STOC 1996).

Recent work of Brutzkus, Daniely, and Malach (COLT 2020) considered the smoothed analysis model as a possible avenue towards resolving this disconnect. Within the smoothed setting and for targets f that are k-juntas, they showed that these heuristics successfully learn f with depth-k decision tree hypotheses. They conjectured that the same guarantee holds more generally for targets that are depth-k decision trees.

We provide a counterexample to this conjecture: we construct targets that are depth-k decision trees and show that even in the smoothed setting, these heuristics build trees of depth $2^{\Omega(k)}$ before achieving high accuracy. We also show that the guarantees of Brutzkus et al. cannot extend to the agnostic setting: there are targets that are very close to k-juntas, for which these heuristics build trees of depth $2^{\Omega(k)}$ before achieving high accuracy.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases decision trees, learning theory, smoothed analysis

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2021.45

Category RANDOM

Funding *Jane Lange*: NSF Award #CCF-2006664

Acknowledgements We thank the anonymous reviewers, whose suggestions have helped improved this paper.

1 Introduction

Greedy decision tree learning heuristics are among the earliest and most basic algorithms in machine learning. Well-known examples include ID3 [28], its successor C4.5 [29], and CART [6], all of which continue to be widely employed in everyday ML applications. These simple heuristics build a decision tree for labeled dataset S in a greedy, top-down fashion. They first identify a "good" attribute to query as the root of the tree. This induces a partition of S into S_0 and S_1, and the left and right subtrees are built recursively using S_0 and S_1 respectively.

In more detail, each heuristic is associated with an *impurity function* $\mathcal{G} : [0,1] \to [0,1]$ that is concave, symmetric around $\frac{1}{2}$, and satisfies $\mathcal{G}(0) = \mathcal{G}(1) = 0$ and $\mathcal{G}(\frac{1}{2}) = 1$. Examples include the binary entropy function $\mathcal{G}(p) = \mathrm{H}(p)$ that is used by ID3 and C4.5, and the Gini

impurity function $\mathcal{G}(p) = 4p(1-p)$ that is used by CART; Kearns and Mansour [21] proposed and analyzed the function $\mathcal{G}(p) = 2\sqrt{p(1-p)}$. For a target function $f : \mathbb{R}^n \to \{0,1\}$ and a distribution $\mathcal{D}$ over $\mathbb{R}^n$, these heuristics build a decision tree hypothesis for f as follows:

1. *Split:* Query $\mathbb{1}[x_i \geq \theta]$ as the root of the tree, where x_i and θ are chosen to (approximately) maximize the *purity gain with respect to* $\mathcal{G}$:

$$\mathcal{G}\text{-purity-gain}_{\mathcal{D}}(f, x_i) \coloneqq \mathcal{G}(\mathbb{E}[f]) - \big(\Pr[x_i \geq \theta] \cdot \mathcal{G}(\mathbb{E}[f_{x_i \geq \theta}]) + \Pr[x_i < \theta] \cdot \mathcal{G}(\mathbb{E}[f_{x_i < \theta}])\big),$$

   where the expectations and probabilities above are with respect to randomly drawn labeled examples $(x, f(x))$ where $x \sim \mathcal{D}$, and $f_{x_i \geq \theta}$ denotes the restriction of f to inputs satisfying $x_i \geq \theta$ (and similarly for $f_{x_i < \theta}$).
2. *Recurse:* Build the left and right subtrees by recursing on $f_{x_i \geq \theta}$ and $f_{x_i < \theta}$ respectively.
3. *Terminate:* The recursion terminates when the depth of the tree reaches a user-specified depth parameter. Each leaf ℓ of the tree is labeled by $\text{round}(\mathbb{E}[f_\ell])$, where we associate ℓ with the restriction corresponding to the root-to-ℓ path within the tree and $\text{round}(p) \coloneqq \mathbb{1}[p \geq \frac{1}{2}]$.

Given the popularity and empirical success of these heuristics[1], it is natural to seek theoretical guarantees on their performance:

> *Let $f : \mathbb{R}^n \to \{0,1\}$ be a target function and $\mathcal{D}$ be a distribution over $\mathbb{R}^n$. Can we obtain a high-accuracy hypothesis for f by growing a depth-k' tree using these heuristics, where k' is not too much larger than k, the optimal decision tree depth for f?* ($\diamondsuit$)

1.1 Background and prior work

A simple and well-known impossibility result

Unfortunately, it has long been known [21, 20] that no such guarantee is possible even under favorable feature and distributional assumptions. Consider the setting of binary features (i.e. $f : \{0,1\}^n \to \{0,1\}$) and the uniform distribution $\mathcal{U}$ over $\{0,1\}^n$, and suppose f is the parity of two unknown features $x_i \oplus x_j$ for $i, j \in [n]$. It can be easily verified that for all impurity functions $\mathcal{G}$, all features have the same purity gain: $\mathcal{G}\text{-purity-gain}_{\mathcal{U}}(f, x_\ell) = 0$ for all $\ell \in [n]$, regardless of whether $\ell \in \{i, j\}$. Therefore, these heuristics may build a tree of depth $\Omega(n)$, querying irrelevant variables x_ℓ where $\ell \notin \{i, j\}$, before achieving any nontrivial accuracy. This is therefore an example where the target f is computable by a decision tree of depth $k = 2$, and yet these heuristics may build a tree of depth $k' = \Omega(n)$ before achieving any nontrivial accuracy.

Smoothed analysis

In light of such impossibility results, a line of work has focused on establishing provable guarantees for restricted classes of target functions [13, 25, 7, 3, 2]; we give an overview of these results in Section 1.3.

---

[1] CART and C4.5 were named as two of the "Top 10 algorithms in data mining" by the International Conference on Data Mining (ICDM) community [33]; other algorithms on this list include k-means, k-nearest neighbors, Adaboost, and PageRank, all of whose theoretical properties are the subjects of intensive study. C4.5 has also been described as "probably the machine learning workhorse most widely used in practice to date" [32].

The focus of our work is instead on *smoothed analysis* as an alternative route towards evading these impossibility results, an approach that was recently considered by Brutzkus, Daniely, and Malach [8]. Smoothed analysis is by now a standard paradigm for going beyond worst-case analysis. Roughly speaking, positive results in this model show that "hard instances are pathological." Smoothed analysis has been especially influential in accounting for the empirical effectiveness of algorithms widely used in practice, a notable example being the simplex algorithm for linear programming [31]. The idea of analyzing greedy decision tree learning heuristics through the lens of smoothed analysis is therefore very natural.

A *smoothed product distribution* over $\{0,1\}^n$, a notion introduced by Kalai, Samrodnitsky, and Teng [19], is obtained by randomly and independently perturbing the bias of each marginal of a product distribution. For smoothed product distributions, Brutzkus et al. proved strong guarantees on the performance of greedy decision tree heuristics when run on targets that are *juntas*, functions that depend only on a small number of its features. For a given impurity function $\mathcal{G}$, let us write $\mathcal{A}_{\mathcal{G}}$ to denote the corresponding decision tree learning heuristic.

▶ **Theorem 1** (Performance guarantee for targets that are k-juntas [8]). *For all impurity functions $\mathcal{G}$ and for all target functions $f : \{0,1\}^n \to \{0,1\}$ that are k-juntas, if $\mathcal{A}_{\mathcal{G}}$ is trained on examples drawn from a smoothed product distribution, it learns a decision tree hypothesis of depth k that achieves perfect accuracy.*

(Therefore Theorem 1 shows that the smoothed setting enables one to circumvent the impossibility result discussed above, which was based on targets that are 2-juntas.)

Every k-junta is computable by a depth-k decision tree, but a depth-k decision tree can depend on as many as 2^k variables. Brutzkus et al. left as an open problem of their paper a conjecture that the guarantees of Theorem 1 hold more generally for targets that are depth-k decision trees:

▶ **Conjecture 2** (Performance guarantee for targets that are depth-k decision trees). *For all impurity functions $\mathcal{G}$ and for all target functions $f : \{0,1\}^n \to \{0,1\}$ that are depth-k decision trees, if $\mathcal{A}_{\mathcal{G}}$ is trained on examples drawn from a smoothed product distribution, it learns a decision tree hypothesis of depth $O(k)$ that achieves high accuracy.*

In other words, Conjecture 2 states that for all targets $f : \{0,1\}^n \to \{0,1\}$, the sought-for guarantee ($\diamondsuit$) holds if the heuristics are trained on examples drawn from a smoothed product distribution.

1.2 This work: Lower bounds in the smoothed setting

Our main result is a counterexample to Conjecture 2. We construct targets that are depth-k decision trees for which all greedy impurity-based heuristics, even in the smoothed setting, may grow a tree of depth $2^{\Omega(k)}$ before achieving high accuracy. This lower bound is close to being maximally large since Theorem 1 implies an upper bound of $O(2^k)$. Our result is actually stronger than just a lower bound in the smoothed setting: our lower bound holds with respect to *any* product distribution that is balanced in the sense that its marginals are not too skewed.

▶ **Theorem 3** (Our main result: a counterexample to Conjecture 2; informal). *Conjecture 2 is false: For all $k = k(n)$, there are target functions $f : \{0,1\}^n \to \{0,1\}$ that are depth-k decision trees such that for all impurity functions $\mathcal{G}$, if $\mathcal{A}_{\mathcal{G}}$ is trained on examples drawn from any balanced product distribution, its decision tree hypothesis does not achieve high accuracy unless it has depth $2^{\Omega(k)}$.*

By building on our proof of Theorem 3, we also show that the guarantees of Brutzkus et al. for k-juntas cannot extend to the agnostic setting:

▶ **Theorem 4** (Theorem 1 does not extend to the agnostic setting; informal). *For all ε and $k = k(n)$, there are target functions $f : \{0,1\}^n \to \{0,1\}$ that are ε-close to a k-junta such that for all impurity functions $\mathcal{G}$, if $\mathcal{A}_{\mathcal{G}}$ is trained on examples drawn from any balanced product distribution, its decision tree hypothesis does not achieve high accuracy unless it has depth $\varepsilon \cdot 2^{\Omega(k)}$.*

In particular, there are targets that are $2^{-\Omega(k)}$-close to k-juntas, for which these heuristics have to construct a decision tree hypothesis of depth $2^{\Omega(k)}$ before achieving high accuracy. Taken together with the positive result of Brutzkus et al., Theorems 3 and 4 add to our understanding of the strength and limitations of greedy decision tree learning heuristics.

Our lower bounds are based on new generalizations of the *addressing function*. Since the addressing function is often a useful extremal example in a variety of settings, we are hopeful that these generalizations and our analysis of them will see further utility beyond the applications of this paper.

1.3 Related Work

As mentioned above, there has been a substantial line of work on establishing provable guarantees for greedy decision tree heuristics when run in restricted classes of target functions. Fiat and Pechyony [13] considered the class of read-once DNF formulas and halfspaces; the Ph.D. thesis of Lee [25] considered the class of monotone functions; Brutzkus, Daniely, and Malach [7] considered conjunctions and read-once DNF formulas; recent works of [3, 2] build on the work of Lee and further studied monotone target functions. (All these works focus on the case of binary features and product distributions over examples.)

Kearns and Mansour [21], in one of the first papers to study these heuristics from a theoretical perspective, showed that they can be viewed as boosting algorithms, with internal nodes of the decision tree hypothesis playing the role of weak learners. Their subsequent work with Dietterich [11] provide experimental results that complement the theoretical results of [21]; see also the survey of Kearns [20].

Finally, we mention that decision trees are one of the most intensively studied concept classes in learning theory. The literature on this problem is rich and vast (see e.g. [12, 30, 5, 15, 9, 24, 4, 16, 22, 26, 18, 27, 14, 23, 19, 19, 17, 10, 1]), studying it from a variety of perspectives and providing both positive and negative results. However, the algorithms developed in these works do not resemble the greedy heuristics used in practice, and indeed, most of them are not proper (in the sense of returning a hypothesis that is itself a decision tree).[2]

2 Preliminaries

Recall that an impurity function $\mathcal{G} : [0,1] \to [0,1]$ is concave, symmetric with respect to $\frac{1}{2}$, and satisfies $\mathcal{G}(0) = \mathcal{G}(1) = 0$ and $\mathcal{G}(\frac{1}{2}) = 1$. We further quantify the concavity and smoothness of $\mathcal{G}$ as follows:

[2] Quoting [21], "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."

▶ **Definition 5** (Impurity functions). *$\mathcal{G}$ is an (α, L)-impurity function if $\mathcal{G}$ is α-strongly concave and L-smooth, i.e., $\mathcal{G}$ is twice-differentiable and $\mathcal{G}''(x) \in [-L, -\alpha]$ for every $x \in [0,1]$.*

For a boolean function $f : \{0,1\}^n \to \{0,1\}$ and index $i \in [n]$, we write $f_{x_i=0}$ and $f_{x_i=1}$ to denote the restricted functions obtained by fixing the i-th input bit of f to either 0 or 1. Formally, each $f_{x_i=b}$ is a function over $\{0,1\}^n$ defined as $f_{x_i=b}(x) = f(x^{i\to b})$, where $x^{i\to b}$ denotes the string obtained by setting the i-th bit of x to b. More generally, a *restriction* π is a list of constraints of form "$x_i = b$" in which every index i appears at most once. For restriction $\pi = (x_{i_1} = b_1, x_{i_2} = b_2, \ldots)$, the restricted function $f_\pi : \{0,1\}^n \to \{0,1\}$ is similarly defined as $f_\pi(x) = f(x^{i_1\to b_1, i_2 \to b_2, \cdots})$.

▶ **Definition 6** (Purity gain). *Let $\mathcal{D}$ be a distribution over $\{0,1\}^n$ and $p_i = \Pr_{x\sim\mathcal{D}}[x_i = 1]$. The $\mathcal{G}$-purity gain of querying variable x_i on boolean function f is defined as*

$$\mathcal{G}\text{-purity-gain}_{\mathcal{D}}(f, x_i) \coloneqq \mathcal{G}\left(\mathop{\mathbb{E}}_{x\sim\mathcal{D}}[f(x)]\right) - p_i\mathcal{G}\left(\mathop{\mathbb{E}}_{x\sim\mathcal{D}}[f_{x_i=1}(x)]\right) - (1-p_i)\mathcal{G}\left(\mathop{\mathbb{E}}_{x\sim\mathcal{D}}[f_{x_i=0}(x)]\right).$$

In a decision tree, each node v naturally corresponds to a restriction π_v formed by the variables queried by the ancestors of v (excluding v itself). We use f_v as a shorthand for f_{π_v}. We say that a decision tree learning algorithm is *impurity-based* if, in the tree returned by the algorithm, every internal node v queries a variable that maximizes the purity gain with respect to f_v.

▶ **Definition 7** (Impurity-based algorithms). *A decision tree learning algorithm is $\mathcal{G}$-impurity-based if the following holds for every $f : \{0,1\}^n \to \{0,1\}$ and distribution $\mathcal{D}$ over $\{0,1\}^n$: When learning f on $\mathcal{D}$, the algorithm outputs a decision tree such that for every internal node v, the variable x_i that is queried at v satisfies $\mathcal{G}$-purity-gain$_{\mathcal{D}}(f_v, x_i) \geq \mathcal{G}$-purity-gain$_{\mathcal{D}}(f_v, x_j)$ for every $j \in [n]$.*

The above definition assumes that the algorithm exactly maximizes the $\mathcal{G}$-purity gain at every split, while in reality, the purity gains can only be estimated from a finite dataset. We therefore consider an idealized setting that grants the learning algorithm with infinitely many training examples, which, intuitively, strengthens our lower bounds. (Our lower bounds show that in order for an algorithm to recover a good tree – a high-accuracy hypothesis whose depth is close to that of the target – it would need to query a variable that has *exponentially smaller* purity gain than that of the variable with the largest purity gain. Hence, if purity gains are estimated using finitely many random samples as is done in reality, the strength of our lower bounds imply that with extremely high probability, impurity-based heuristics will fail to build a good tree; see Remark 15 for a detailed discussion.)

When a decision tree queries variable x_i on function f, it naturally induces two restricted functions $f_{x_i=0}$ and $f_{x_i=1}$. The following lemma states that the purity gain of querying x_i is roughly the squared difference between the averages of the two functions, up to a factor that depends on the impurity function $\mathcal{G}$ and the data distribution $\mathcal{D}$. We say that a product distribution over $\{0,1\}^n$ is δ-*balanced* if the expectation of each of the n coordinates is in $[\delta, 1-\delta]$.

▶ **Lemma 8.** *For any $f : \{0,1\}^n \to \{0,1\}$, δ-balanced product distribution $\mathcal{D}$ over $\{0,1\}^n$ and (α, L)-impurity function $\mathcal{G}$, it holds for $\kappa = \max\left(\frac{2}{\alpha\delta(1-\delta)}, \frac{L}{8}\right)$ and every $i \in [n]$ that*

$$\frac{1}{\kappa} \leq \frac{\mathcal{G}\text{-purity-gain}_{\mathcal{D}}(f, x_i)}{\left[\mathbb{E}_{x\sim D}[f_{x_i=0}(x)] - \mathbb{E}_{x\sim D}[f_{x_i=1}(x)]\right]^2} \leq \kappa.$$

Proof of Lemma 8. Let $p_i = \Pr_{x\sim\mathcal{D}}[x_i = 1]$ and $\mu_b = \mathbb{E}_{x\sim D}[f_{x_i=b}(x)]$ respectively. Then, we have $\mathbb{E}_{x\sim\mathcal{D}}[f(x)] = p_i\mu_1 + (1-p_i)\mu_0$, and the purity gain can be written as

$$\mathcal{G}\text{-purity-gain}_{\mathcal{D}}(f, x_i) = \mathcal{G}(p_i\mu_1 + (1-p_i)\mu_0) - p_i\mathcal{G}(\mu_1) - (1-p_i)\mathcal{G}(\mu_0).$$

Since $\mathcal{G}$ is α-strongly concave and L-smooth, the above is bounded between $\frac{\alpha}{2}\cdot p_i(1-p_i)\cdot(\mu_0-\mu_1)^2$ and $\frac{L}{2}\cdot p_i(1-p_i)\cdot(\mu_0-\mu_1)^2$. Since $\mathcal{D}$ is δ-balanced, we have $\delta(1-\delta) \le p_i(1-p_i) \le \frac{1}{4}$. It follows that

$$\frac{\alpha}{2}\cdot\delta(1-\delta) \le \frac{\alpha}{2}\cdot p_i(1-p_i) \le \frac{\mathcal{G}\text{-purity-gain}_{\mathcal{D}}(f,x_i)}{(\mu_0-\mu_1)^2} \le \frac{L}{2}\cdot p_i(1-p_i) \le \frac{L}{8}.$$

Thus, the ratio is bounded between $1/\kappa$ and κ. ◀

Our lower bounds hold with respect to all δ-balanced product distributions. We compare this to the definition of a *c-smoothened* δ-balanced product distribution from [8].

▶ **Definition 9** (Smooth distributions). *A c-smoothened δ-balanced product distribution is a* random *product distribution over $\{0,1\}^n$ where the marginal for the i^{th} bit is* 1 *with probability $\widehat{p_i} + \Delta_i$ for fixed $\widehat{p_i} \in (\delta + c, 1 - \delta - c)$ and Δ_i drawn i.i.d. from* Uniform$([-c, c])$.

Since our lower bounds hold against all δ-balanced product distributions, it also holds against all *c-smoothened* δ-balanced product distributions.

3 Proof overview and formal statements of our results

Our goal is to construct a target function that can be computed by a depth-k decision tree, but on which impurity-based algorithms must build to depth $2^{\Omega(k)}$ or have large error. To do so, we construct a decision tree target T where the variables with *largest* purity gain are at the *bottom* layer of T (adjacent to its leaves). Intuitively, impurity-based algorithms will build their decision tree hypothesis for T by querying all the variables in the bottom layer of T before querying any of the variables higher up in T. Our construction will be such that until the higher up variables are queried, it is impossible to approximate the target with any nontrivial error. Summarizing informally, we show that impurity-based algorithms build its decision tree hypothesis for our target by querying variables in exactly the "wrong order".

The starting point of our construction is the well known *addressing function*. For $k \in \mathbb{N}$, the addressing function $f : \{0,1\}^{k+2^k} \to \{0,1\}$ is defined as follows: Given "addressing bits" $z \in \{0,1\}^k$ and "memory bits" $y \in \{0,1\}^{2^k}$, the output $f(y,z)$ is the z^{th} bit of y, where "z^{th} bit" is computed by interpreting z as a base-2 integer. Note that the addressing function is computable by a decision tree of depth $k+1$ that first queries the k addressing bits, followed by the appropriate memory bit.

For our lower bound, we would like the variables with the highest purity gain to be the memory bits. However, for smoothed product distributions, the addressing bits might have higher purity gain than the memory bits, and impurity-based algorithms might succeed in learning the addressing function. We therefore modify the addressing function by making each addressing bit the parity of multiple new bits. We show that by making each addressing bit the parity of sufficiently many new bits, we can drive the purity gain of these new bits down to the point where the memory bits have the highest purity gain as desired – in fact, larger than the addressing bits by a multiplicative factor of $e^{\Omega(k)}$. (Making each addressing bit the parity of multiple new bits increases the depth of the target, so this introduces technical challenges we have to overcome in order to achieve the strongest parameters.)

Our main theorem is formally restated as follows.

▶ **Theorem 10** (Formal version of Theorem 3). *Fix $L \geq \alpha > 0$ and $\delta \in (0, \frac{1}{2}]$. There are boolean functions $f_1, f_2, \ldots$ such that: (1) f_k is computable by a decision tree of depth $O(k/\delta)$; (2) For every δ-balanced product distribution $\mathcal{D}$ over the domain of f_k and every (α, L)-impurity function $\mathcal{G}$, any $\mathcal{G}$-impurity based decision tree heuristic, when learning f_k on $\mathcal{D}$, returns a tree that has either depth $\geq 2^k$ or an $\Omega(\delta)$ error.*

An extension of our construction and its analysis shows that the guarantees of Brutzkus et al. for targets that are k-juntas cannot extend to the agnostic setting. Roughly speaking, while our variant of the addressing function from Theorem 10 is far from all k-juntas, it can be made close to one by fixing most of the memory bits. We obtain our result by showing that our analysis continues to hold under such a restriction.

▶ **Theorem 11** (Formal version of Theorem 4). *Fix $L \geq \alpha > 0$, $\delta \in (0, \frac{1}{2}]$ and $\varepsilon \in (0, 1]$. There are boolean functions $f_1, f_2, \ldots$ such that for every δ-balanced product distribution $\mathcal{D}$ over the domain of f_k: (1) f_k is ε-close to an $O(k/\delta)$-junta with respect to $\mathcal{D}$; (2) For every (α, L)-impurity function $\mathcal{G}$, any $\mathcal{G}$-impurity based decision tree heuristic, when learning f_k on $\mathcal{D}$, returns a tree that has either a depth of $\Omega(\varepsilon \cdot 2^k)$ or an $\Omega(1)$ error.*

4 Warm-Up: A Weaker Lower Bound

We start by giving a simplified construction that proves a weaker version of Theorem 10, in which the $O(k/\delta)$ depth in condition (1) is relaxed to $O(k^2/\delta)$. For integers $c, k \geq 1$, we define a boolean function $f_{c,k} : \{0,1\}^{ck^2+2^k} \to \{0,1\}$ as follows. The input of $f_{c,k}$ is viewed as two parts: ck^2 *addressing bits* $x_{i,j}$ indexed by $i \in [k]$ and $j \in [ck]$, and 2^k *memory bits* y_a indexed by $a \in \{0,1\}^k$. The function value $f_{c,k}(x, y)$ is defined by first computing $z_i(x) = \bigoplus_{j=1}^{ck} x_{i,j}$ for every $i \in [k]$, and then assigning $f_{c,k}(x, y) = y_{z(x)}$.

In other words, $f_{c,k}$ is a disjoint composition of the k-bit addressing function and the parity function over ck bits. Given addressing bits x and memory bits y, the function first computes a k-bit address by taking the XOR of the addressing bits in each group of size ck, and then retrieves the memory bit with the corresponding address. Clearly, $f_{c,k}$ can be computed by a decision tree of depth $ck^2 + 1$ that first queries all the ck^2 addressing bits and then queries the relevant memory bit in the last layer.

4.1 Address is Almost Uniform

Drawing input (x, y) from a distribution $\mathcal{D}$ naturally defines a distribution over $\{0,1\}^k$ of the k-bit address $z(x) = (z_1(x), z_2(x), \ldots, z_k(x))$. The following lemma states that when $\mathcal{D}$ is a δ-balanced product distribution, the distribution of $z(x)$ is almost uniform in the ℓ_∞ sense. Furthermore, this almost uniformity holds even if one of the addressing bits $x_{i,j}$ is fixed.

▶ **Lemma 12.** *Suppose that $c \geq \frac{\ln 5}{\delta}$ and $\mathcal{D}$ is a δ-balanced product distribution over the domain of $f_{c,k}$. Then,*

$$\left| \Pr_{(x,y)\sim\mathcal{D}}[z(x) = a] - 2^{-k} \right| \leq 5^{-k}, \forall a \in \{0,1\}^k.$$

Furthermore, for every $i \in [k]$, $j \in [ck]$ and $b \in \{0,1\}$,

$$\left| \Pr_{(x,y)\sim\mathcal{D}}[z(x) = a | x_{i,j} = b] - 2^{-k} \right| \leq 5^{-k}, \forall a \in \{0,1\}^k.$$

The proof of Lemma 12 uses the following simple fact, which states that the XOR of independent biased random bits is exponentially close to an unbiased coin flip.

▶ **Lemma 13.** *Suppose that $x_1, x_2, \ldots, x_n$ are independent Bernoulli random variables, each with an expectation between δ and $1-\delta$. Then, $\left|\Pr\left[x_1 \oplus x_2 \oplus \cdots \oplus x_n = 1\right] - \frac{1}{2}\right| \leq \frac{1}{2}(1-2\delta)^n \leq \frac{1}{2}\exp(-2\delta n)$.*

Proof of Lemma 12. Since $z_i(x) = \bigoplus_{j=1}^{ck} x_{i,j}$ and $\mathcal{D}$ is δ-balanced, Lemma 13 gives

$$\left|\Pr_{(x,y)\sim\mathcal{D}}\left[z_i(x)=1\right] - \frac{1}{2}\right| \leq \frac{1}{2}\exp(-2\delta ck) \leq \frac{1}{2}\cdot 5^{-k}.$$

Note that the bits of $z(x)$ are independent, so $\Pr_{(x,y)\sim\mathcal{D}}\left[z(x)=a\right]$ is given by

$$\prod_{i=1}^{k}\Pr_{(x,y)\sim\mathcal{D}}\left[z_i(x)=a_i\right] \leq \left(\frac{1}{2}+\frac{1}{2}\cdot 5^{-k}\right)^k = 2^{-k}\cdot(1+5^{-k})^k \leq 2^{-k}\cdot(1+(2/5)^k) = 2^{-k}+5^{-k},$$

where the third step applies $(1+x)^k \leq 1+2^k x$ for $x\in[0,1]$ and integers $k\geq 1$. Similarly,

$$\Pr_{(x,y)\sim\mathcal{D}}\left[z(x)=a\right] \geq \left(\frac{1}{2}-\frac{1}{2}\cdot 5^{-k}\right)^k \geq 2^{-k}\cdot(1-k\cdot 5^{-k}) \geq 2^{-k}-5^{-k},$$

where the last two steps apply $(1-x)^k \geq 1-kx$ and $k\cdot 2^{-k}\leq 1$. This proves the first part.

The proof of the "furthermore" part is essentially the same, except that conditioning on $x_{i,j}=b$, $z_i(x)$ becomes the XOR of $ck-1$ independent bits and b. By Lemma 13, we have

$$\left|\Pr_{(x,y)\sim\mathcal{D}}\left[z_i(x)=1 | x_{i,j}=b\right] - \frac{1}{2}\right| \leq \frac{1}{2}\exp(-2\delta(ck-1)) \leq \frac{1}{2}\exp(-\delta ck) \leq \frac{1}{2}\cdot 5^{-k},$$

and the rest of the proof is the same. ◀

4.2 Memory Bits are Queried First

The following technical lemma states that the purity gain of $f_{c,k}$ is maximized by a memory bit, regardless of the impurity function and the data distribution. Therefore, when an impurity-based algorithm (in the sense of Definition 7) learns $f_{c,k}$, the root of the decision tree will always query a memory bit. Furthermore, this property also holds for restrictions of $f_{c,k}$ as long as the restriction only involves the memory bits.

▶ **Lemma 14.** *Fix $L \geq \alpha > 0$ and $\delta \in (0, \frac{1}{2}]$. Let $c_0 = \frac{\ln 5}{\delta}$ and $k_0 = \frac{\ln(2\kappa)}{\ln(5/4)} + 1$, where κ is chosen as in Lemma 8. The following holds for every function $f_{c,k}$ with $c \geq c_0$ and $k \geq k_0$: For any (α, L)-impurity function $\mathcal{G}$, δ-balanced product distribution $\mathcal{D}$ and restriction π of size $< 2^k$ that only contains the memory bits of $f_{c,k}$, the purity gain $\mathcal{G}$-purity-gain$_{\mathcal{D}}((f_{c,k})_\pi, \cdot)$ is maximized by a memory bit.*

Proof of Lemma 14. Fix $c \geq c_0$ and $k \geq k_0$ and shorthand f for $f_{c,k}$. We will prove a stronger claim: with respect to f_π, every memory bit (that is not in π) gives a much higher purity gain than every addressing bit does.

Purity gain of the memory bits

Fix a memory bit y_a ($a \in \{0,1\}^k$) that does not appear in restriction π. Let $\mu_b = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[f_{\pi, y_a=b}(x,y)\right]$ for $b\in\{0,1\}$. By the law of total expectation,

$$\begin{aligned}\mu_b &= \Pr_{(x,y)\sim\mathcal{D}}\left[z(x)=a\right]\cdot \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[f_{\pi,y_a=b}(x,y)|z(x)=a\right] \\ &\quad + \Pr_{(x,y)\sim\mathcal{D}}\left[z(x)\neq a\right]\cdot \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[f_{\pi,y_a=b}(x,y)|z(x)\neq a\right] \\ &= \Pr_{(x,y)\sim\mathcal{D}}\left[z(x)=a\right]\cdot b + \Pr_{(x,y)\sim\mathcal{D}}\left[z(x)\neq a\right]\cdot \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[f_\pi(x,y)|z(x)\neq a\right].\end{aligned}$$

Here the second step holds since $f_{\pi,y_a=b}(x,y)$ evaluates to b when the address $z(x)$ equals a, and $f_{\pi,y_a=b}$ agrees with f_π when $z(x) \neq a$. Since only the first term above depends on b, we have

$$|\mu_0 - \mu_1| = \Pr_{(x,y)\sim\mathcal{D}}[z(x) = a] \geq 2^{-k} - 5^{-k} \geq \frac{1}{2} \cdot 2^{-k},$$

where the second step follows from $c \geq c_0$ and Lemma 12. Finally, by Lemma 8, $\mathcal{G}$-purity-gain$_\mathcal{D}(f_\pi, y_a) \geq \frac{1}{\kappa}(\mu_0 - \mu_1)^2 \geq \frac{1}{4\kappa} \cdot 2^{-2k}$.

Purity gain of the addressing bits

Similarly, we fix an addressing bit $x_{i,j}$ and define the average $\mu_b = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[f_{\pi,x_{i,j}=b}(x,y)\right]$. Since $\mathcal{D}$ is a product distribution, μ_b is equal to the conditional expectation $\mathbb{E}_{(x,y)\sim\mathcal{D}}[f_\pi(x,y)|x_{i,j} = b]$. Then, by the law of total expectation, we can write μ_b as

$$\begin{aligned}\mu_b &= \sum_{a\in\{0,1\}^k} \Pr_{(x,y)\sim\mathcal{D}}[z(x) = a|x_{i,j} = b] \cdot \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[f_\pi(x,y)|z(x) = a, x_{i,j} = b] \\ &= \sum_{a\in\{0,1\}^k} \Pr_{(x,y)\sim\mathcal{D}}[z(x) = a|x_{i,j} = b] \cdot \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[f_\pi(x,y)|z(x) = a]\,.\end{aligned}$$

Here the second step holds since $f_\pi(x,y)$ and $x_{i,j}$ are independent conditioning on the address $z(x)$; in other words, once we know the value of $z(x)$, it doesn't matter how x is set in determining the output of f.

Let c_a denote $\mathbb{E}_{(x,y)\sim\mathcal{D}}[f_\pi(x,y)|z(x) = a]$, and let $\mathcal{P}_b$ be the distribution of $z(x)$ conditioning on $x_{i,j} = b$. Then, μ_b is exactly given by $\mathbb{E}_{a\sim\mathcal{P}_b}[c_a]$. Since each c_a is in $[0,1]$, $|\mu_0 - \mu_1|$ is upper bounded by the total variation distance between $\mathcal{P}_0$ and $\mathcal{P}_1$:

$$\begin{aligned}|\mu_0 - \mu_1| &\leq \frac{1}{2} \sum_{a\in\{0,1\}^k} |\mathcal{P}_0(a) - \mathcal{P}_1(a)| \\ &\leq \frac{1}{2} \sum_{a\in\{0,1\}^k} \left(|\mathcal{P}_0(a) - 2^{-k}| + |\mathcal{P}_1(a) - 2^{-k}|\right) \\ &\leq \frac{1}{2} \cdot 2^k \cdot 2 \cdot 5^{-k} = (2/5)^k. \qquad \text{(Lemma 12)}\end{aligned}$$

Finally, applying Lemma 8 shows that $\mathcal{G}$-purity-gain$_\mathcal{D}(f_\pi, x_{i,j}) \leq \kappa(\mu_0 - \mu_1)^2 \leq \kappa \cdot (2/5)^{2k}$.

Recall that $k \geq k_0 > \frac{\ln(2\kappa)}{\ln(5/4)}$, so we have $\kappa \cdot (2/5)^{2k} < \frac{1}{4\kappa} \cdot 2^{-2k}$. Therefore, the purity gain of every memory bit outside the restriction is strictly larger than that of any addressing bit, and the lemma follows immediately. ◀

▶ Remark 15. The proof above bounds the purity gain of each memory bit and each addressing bit by $\Omega((1/2)^{2k})$ and $O((2/5)^{2k})$ respectively. For Lemma 14 to hold when the purity gains are estimated from a finite dataset, it suffices to argue that each estimate is accurate up to an $O((2/5)^{2k})$ additive error. By a standard concentration argument, to estimate the purity gains for all restriction π of size $\leq h$, $2^{O(h+k)}$ training examples are sufficient. When applied later in the proof of Theorem 10, this finite-sample version of Lemma 14 would imply that impurity-based algorithms need to build a tree of depth h as soon as the sample size reaches $2^{\Omega(h+k)}$.

4.3 Proof of the Weaker Version

Now we are ready to prove the weaker version of Theorem 10. We will apply Lemma 14 to argue that the tree returned by an impurity-based algorithm never queries an addressing bit (unless all the 2^k memory bits have been queried), and then show that every such decision tree must have an error of $\Omega(\delta)$.

Proof of Theorem 10 (weaker version). Fix integer $c \geq \frac{\ln 5}{\delta}$ and consider the functions $f_{c,1}, f_{c,2}, \ldots$. Since each $f_{c,k}$ is represented by a decision tree of depth $ck^2 + 1 = O(k^2/\delta)$, it remains to show that impurity-based algorithms fail to learn $f_{c,k}$. Fix integer $k \geq k_0$ (where k_0 is chosen as in Lemma 14) and δ-balanced product distribution $\mathcal{D}$ over the domain of $f_{c,k}$. In the following, we use shorthand f for $f_{c,k}$.

Small trees never query addressing bits

Let T be the decision tree returned by a $\mathcal{G}$-impurity-based algorithm when learning f on $\mathcal{D}$. If T has depth $> 2^k$, we are done, so we assume that T has depth at most 2^k. We claim that T never queries the addressing bits of f. Suppose otherwise, that an addressing bit is queried at node v in T, and no addressing bits are queried by the ancestors of v. Then, the restriction π_v associated with node v only contains the memory bits of f. Since T has depth $\leq 2^k$, the size of π_v is strictly less than 2^k. Then, by Lemma 14, the $\mathcal{G}$-purity gain with respect to f_v is maximized by a memory bit. This contradicts the assumption that the algorithm is $\mathcal{G}$-impurity-based.

Trivial accuracy if no addressing bits are queried

We have shown that T only queries the memory bits of f. We may further assume that T queries *all* the 2^k memory bits before reaching any of its leaves, i.e., T is a full binary tree of depth 2^k. This assumption is without loss of generality because we can add dummy queries on the memory bits to the leaves of depth $< 2^k$, and label all the resulting leaves with the same bit. This change does not modify the function represented by T.

Assuming that T is full, every leaf ℓ of T is labeled by 2^k bits $(c_a)_{a\in\{0,1\}^k}$, meaning that each memory bit y_a is fixed to c_a on the root-to-ℓ path. The expectation of the restricted function f_ℓ is then given by $\mu_\ell \coloneqq \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[c_{z(x)}\right]$. Clearly, the error of T is minimized when each leaf ℓ is labeled with $\mathbb{1}\left[\mu_\ell \geq \frac{1}{2}\right]$, and the conditional error when reaching leaf ℓ is $\min(\mu_\ell, 1-\mu_\ell)$.

It remains to show that for a large fraction of leaves ℓ, μ_ℓ is bounded away from 0 and 1, so that $\min(\mu_\ell, 1-\mu_\ell)$ is large. When leaf ℓ is randomly chosen according to distribution $\mathcal{D}$, the corresponding μ_ℓ is given by

$$\mu_\ell = \sum_{a\in\{0,1\}^k} \Pr_{(x,y)\sim\mathcal{D}}\left[z(x) = a\right] \cdot c_a, \tag{1}$$

where $(c_a)_{a\in\{0,1\}^k}$ are 2^k independent Bernoulli random variables with means in $[\delta, 1-\delta]$.

By Lemma 12 and our choice of $c \geq c_0$, $\Pr_{(x,y)\sim\mathcal{D}}\left[z(x) = a\right] \leq 2 \cdot 2^{-k}$ holds for every $a \in \{0,1\}^k$. Thus, each term in (1) is bounded between 0 and $2 \cdot 2^{-k}$. Furthermore, since each c_a has expectation at least δ, $\mathbb{E}\left[\mu_\ell\right] \geq \delta$. Then, Hoeffding's inequality guarantees that over the random choice of $(c_a)_{a\in\{0,1\}^k}$, $\mu_\ell \geq \delta/2$ holds with probability at least $1 - \exp\left(-\frac{2\cdot(\delta/2)^2}{2^k\cdot(2\cdot 2^{-k})^2}\right) = 1 - \exp(-2^k\delta^2/8)$, which is lower bounded by $2/3$ for all sufficiently large k. By a symmetric argument, $\mu_\ell \leq 1-\delta/2$ also holds with probability $\geq 2/3$. Therefore,

with probability $\geq 1/3$ over the choice of leaf ℓ, $\mu_\ell \in [\delta/2, 1-\delta/2]$ holds and thus the conditional error on leaf ℓ is at least $\delta/2$. This shows that the error of T over distribution $\mathcal{D}$ is lower bounded by $\delta/6$, which completes the proof. ◀

5 Proof of Theorem 10

When proving the weaker version of Theorem 10, each hard instance $f_{c,k}$ has $\Theta(k^2)$ addressing bits grouped into k disjoint subsets, and the k-bit address is defined by the XOR of bits in each subset. We will prove Theorem 10 using a slightly different construction that computes address from k overlapping subsets of only $O(k)$ addressing bits.

For integers $c, k \geq 1$ and a list of k sets $S = (S_1, S_2, \ldots, S_k)$ where each $S_i \subseteq [ck]$, we define a boolean function $f_{c,k,S} : \{0,1\}^{ck+2^k} \to \{0,1\}$ as follows. The input of $f_{c,k,S}$ is again divided into two parts: ck addressing bits $x_1, x_2, \ldots, x_{ck}$ and 2^k memory bits y_a indexed by a k-bit address a. The function value $f(x,y)$ is computed by taking $z_i(x) = \bigoplus_{j \in S_i} x_j$ and then $f(x,y) = y_{z(x)}$. Clearly, $f_{c,k,S}$ can be computed by a decision tree of depth $ck+1$ that first queries all the ck addressing bits $x_1, x_2, \ldots, x_{ck}$, and then queries the relevant memory bit $y_{z(x)}$.

Let $\triangle_{i=1}^{k} S_i$ denote the k-ary symmetric difference of sets S_1 through S_k, i.e., the set of elements that appear in an odd number of sets. We say that a list of sets $S = (S_1, S_2, \ldots, S_k)$ has *distance* d, if any non-empty collection of sets has a symmetric difference of size at least d, i.e., $|\triangle_{i \in I} S_i| \geq d$ for every non-empty $I \subseteq [k]$. In the following, we prove analogs of Lemmas 12 and 14 for function $f_{c,k,S}$ assuming that S has a large distance; Theorem 10 would then follow immediately.

▶ **Lemma 16.** *Suppose that $\mathcal{D}$ is a δ-balanced product distribution over the domain of $f_{c,k,S}$ and S has distance $d \geq \frac{\ln 5}{\delta} \cdot k$. Then,*

$$\left| \Pr_{(x,y)\sim\mathcal{D}} [z(x) = a] - 2^{-k} \right| \leq 5^{-k}, \forall a \in \{0,1\}^k.$$

Furthermore, for every $i \in [ck]$ and $b \in \{0,1\}$,

$$\left| \Pr_{(x,y)\sim\mathcal{D}} [z(x) = a | x_i = b] - 2^{-k} \right| \leq 5^{-k}, \forall a \in \{0,1\}^k.$$

We prove Lemma 16 by noting that the distribution of $z(x)$ has exponentially small Fourier coefficients (except the degree-0 one) under the assumptions, and is thus close to the uniform distribution over $\{0,1\}^k$. More concretely, our goal is to show that, for every $I \subseteq [k]$ the quantity $\bigoplus_{i \in I} z_i(x)$ is 1 with probability nearly exactly $\frac{1}{2}$. Afterwards, we will show this is sufficient to guarantee that the distribution of $z(x)$ is close to the uniform distribution.

Proof of Lemma 16. Since $z_i(x) = \bigoplus_{j \in S_i} x_j$, we have $\bigoplus_{i \in I} z_i(x) = \bigoplus_{j \in S_I} x_j$ for every $I \subseteq [k]$, where $S_I = \triangle_{i \in I} S_i$ is the symmetric difference of the corresponding sets. Since S has distance d, $|S_I| \geq d$ for every non-empty $I \subseteq [k]$ and thus $\bigoplus_{i \in I} z_i(x)$ is the XOR of at least d independent bits. Note that $1 - 2\bigoplus_{i \in I} z_i(x) = \prod_{i \in I}(1 - 2z_i(x))$. By Lemma 13 and $d \geq \frac{\ln 5}{\delta} \cdot k$,

$$\left| \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[\prod_{i \in I} (1 - 2z_i(x)) \right] \right| = 2 \cdot \left| \Pr_{(x,y)\sim\mathcal{D}} \left[\bigoplus_{i \in I} z_i(x) = 1 \right] - \frac{1}{2} \right| \leq \exp(-2\delta d) \leq 5^{-k}. \quad (2)$$

Note that for $b_1, b_2 \in \{0,1\}$, we have $\mathbb{1}[b_1 = b_2] = \frac{(1-2b_1)(1-2b_2)+1}{2}$. Therefore, for every $a \in \{0,1\}^k$,

$$\begin{aligned}
\left| \Pr_{(x,y)\sim\mathcal{D}}[z(x) = a] - 2^{-k} \right| &= \left| \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[\prod_{i=1}^{k} \frac{(1-2a_i)(1-2z_i(x))+1}{2} \right] - 2^{-k} \right| \\
&= 2^{-k} \left| \sum_{I\subseteq[k]} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[\prod_{i\in I} (1-2a_i)(1-2z_i(x)) \right] - 1 \right| \\
&\qquad \text{(expansion of product and linearity)} \\
&= 2^{-k} \left| \sum_{I\subseteq[k]:I\neq\emptyset} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[\prod_{i\in I} (1-2a_i)(1-2z_i(x)) \right] \right| \\
&\qquad \text{(empty product equals 1)} \\
&\leq 2^{-k} \sum_{I\subseteq[k]:I\neq\emptyset} \left| \prod_{i\in I} (1-2a_i) \right| \cdot \left| \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[\prod_{i\in I} (1-2z_i(x)) \right] \right| \\
&\qquad \text{(triangle inequality and linearity)} \\
&= 2^{-k} \sum_{I\subseteq[k]:I\neq\emptyset} \left| \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[\prod_{i\in I} (1-2z_i(x)) \right] \right| \qquad (|1-2a_i| = 1) \\
&\leq 2^{-k} \cdot (2^k - 1) \cdot 5^{-k} < 5^{-k}. \qquad \text{(Inequality (2))}
\end{aligned}$$

The proof of the "furthermore" part is the same, except that after conditioning on $x_i = b$, each $\bigoplus_{j\in I} z_j(x)$ is now the XOR of at least $d-1$ independent bits, and the remaining proof goes through. ◀

We note that the proof of Lemma 14 depends on the definition of $z(x)$ only through the application of Lemma 12. Thus, Lemma 16 directly implies the following analog of Lemma 14:

▶ **Lemma 17.** *Fix $L \geq \alpha > 0$ and $\delta \in (0, \frac{1}{2}]$. Let $c_0 = \frac{\ln 5}{\delta}$ and $k_0 = \frac{\ln(2\kappa)}{\ln(5/4)} + 1$, where κ is chosen as in Lemma 8. The following holds for every function $f_{c,k,S}$ such that $k \geq k_0$ and S has distance $c_0 k$: For any (α, L)-impurity function $\mathcal{G}$, δ-balanced product distribution $\mathcal{D}$ and restriction π of size $< 2^k$ that only contains the memory bits of $f_{c,k,S}$, the purity gain $\mathcal{G}$-purity-gain$_{\mathcal{D}}((f_{c,k,S})_\pi, \cdot)$ is maximized by a memory bit.*

Finally, we prove Theorem 10 by showing the existence of a set family S with a good distance.

Proof of Theorem 10. Fix $\delta \in (0, \frac{1}{2}]$. The Gilbert–Varshamov bound for binary linear codes implies that for some $c = \Theta(1/\delta)$, there exists a binary linear code with rate $\frac{1}{c}$ and relative distance $\frac{\ln 5}{\delta c}$. It follows that for every sufficiently large k, there exists $S^{(k)} = (S_1^{(k)}, S_2^{(k)}, \ldots, S_k^{(k)})$ such that each $S_i^{(k)} \subseteq [ck]$ and $S^{(k)}$ has distance $\frac{\ln 5}{\delta} \cdot k$. This can be done by using the i-th basis of the linear code as the indicator vector of subset $S_i^{(k)}$ for each $i \in [k]$.

We prove Theorem 10 using functions $f_{c,1,S^{(1)}}, f_{c,2,S^{(2)}}, \ldots$. Since each $f_{c,k,S^{(k)}}$ can be represented by a decision tree of depth $ck + 1 = O(k/\delta)$, it remains to prove that impurity-based algorithms fail to learn $f_{c,k,S^{(k)}}$. Lemma 17 guarantees that the tree returned by such algorithms either has depth $> 2^k$, or never queries any addressing bits. In the latter case, by the same calculation as in the proof of the weaker version, the decision tree must have an $\Omega(\delta)$ error on distribution $\mathcal{D}$. ◀

6 Proof of Theorem 11

We prove Theorem 11 using the construction of $f_{c,k,S}$ in Section 5, where $S = (S_1, S_2, \ldots, S_k)$ is a list of k subsets of $[ck]$ and each S_i specifies how the i-th bit of the address, $z_i(x)$, is computed from the addressing bits x_1 through x_{ck}. Note that $f_{c,k,S}$ itself depends on $\Omega(2^k)$ input bits and is thus not an $O(k)$-junta. Nevertheless, we will show that, after we fix most of the memory bits of $f_{c,k,S}$, the function is indeed close to a (ck)-junta with relevant inputs being the ck addressing bits. Then, as in the proof of Theorem 10, we will argue that impurity-based heuristics still query the (unfixed) memory bits before querying any of the addressing bits, resulting in a tree that is either exponentially deep or far from the target function.

Proof of Theorem 11. As in the proof of Theorem 10, we can find functions $f_{c,1,S^{(1)}}, f_{c,2,S^{(2)}}, \ldots$ for some $c = \Theta(1/\delta)$ such that each $S^{(k)}$ has distance $\geq \frac{\ln 5}{\delta} \cdot k$. We fix a sufficiently large integer k and shorthand f for $f_{c,k,S^{(k)}}$ in the following.

Partition $\{0,1\}^k$ into three sets A^0, A^1 and A^{free} such that $|A^0| = |A^1|$ and $\varepsilon \cdot 2^{k-2} \leq |A^{\text{free}}| \leq \varepsilon \cdot 2^{k-1}$. Consider the restriction π of function f such that the memory bit y_a is fixed to be 0 for every $a \in A^0$ and fixed to be 1 for every $a \in A^1$; the memory bits with addresses in A^{free} are left as "free" variables. We will prove the theorem using f_π as the k-th function in the family.

f_π is close to a junta

Consider the function $g : \{0,1\}^{ck+2^k} \to \{0,1\}$ defined as $g(x,y) = \mathbb{1}\left[z(x) \in A^1\right]$, where $z(x)$ denotes $(z_1(x), z_2(x), \ldots, z_k(x))$ and each $z_i(x) = \bigoplus_{j \in S_i^{(k)}} x_j$. Clearly, $g(x,y)$ only depends on $x \in \{0,1\}^{ck}$ and is thus a (ck)-junta. Furthermore, for every input (x,y) such that $z(x) \in A^0$ (resp. $z(x) \in A^1$), both f_π and g evaluate to 0 (resp. 1). Thus, f_π and g may disagree only if $z(x) \in A^{\text{free}}$. It follows that for every δ-balanced product distribution $\mathcal{D}$,

$$\begin{aligned}
\Pr_{(x,y)\sim\mathcal{D}}\left[f_\pi(x,y) \neq g(x,y)\right] &\leq \Pr_{(x,y)\sim\mathcal{D}}\left[z(x) \in A^{\text{free}}\right] \\
&\leq |A^{\text{free}}| \cdot (2^{-k} + 5^{-k}) && \text{(Lemma 16)} \\
&\leq \varepsilon \cdot 2^{k-1} \cdot (2^{-k} + 5^{-k}) < \varepsilon. && (|A^{\text{free}}| \leq \varepsilon \cdot 2^{k-1})
\end{aligned}$$

Therefore, f_π is ε-close to an $O(k/\delta)$-junta (namely, g) with respect to distribution $\mathcal{D}$.

Impurity-based algorithms fail to learn f_π

Let T be the decision tree returned by an $\mathcal{G}$-impurity based algorithm when learning f_π on distribution $\mathcal{D}$. By Lemma 17, T must query all the free memory bits with addresses in A^{free} before querying any of the addressing bits. Thus, either T has depth $> |A^{\text{free}}| = \Omega(\varepsilon \cdot 2^k)$, or T only queries the free memory bits of f_π.

In the latter case, we may again assume without loss of generality that T queries all the free memory bits $(y_a)_{a \in A^{\text{free}}}$ before reaching any of its leaves, i.e., T is a full binary tree of depth $|A^{\text{free}}|$. Then, every leaf ℓ naturally specifies 2^k bits $(c_a)_{a \in \{0,1\}^k}$ defined as

$$c_a = \begin{cases} 0, & a \in A^0, \\ 1, & a \in A^1, \\ b, & a \in A^{\text{free}}, y_a \text{ is fixed to } b \text{ on the root-to-}\ell \text{ path.} \end{cases}$$

Let $\mu_\ell \coloneqq \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[c_{z(x)}\right]$. Again, the minimum possible error conditioning on reaching leaf ℓ is $\min(\mu_\ell, 1-\mu_\ell)$, achieved by labeling ℓ with $\mathbb{1}\left[\mu_\ell \geq \frac{1}{2}\right]$. On the other hand, we have

$$\begin{aligned}
\mu_\ell &\geq \Pr_{(x,y)\sim\mathcal{D}}\left[z(x) \in A^1\right] \\
&\geq |A^1| \cdot (2^{-k} - 5^{-k}) && \text{(Lemma 16)} \\
&\geq \frac{2^k - |A^{\text{free}}|}{2} \cdot 2^{-(k+1)} && (2|A^1| + |A^{\text{free}}| = 2^k) \\
&\geq \frac{2^k - 2^{k-1}}{2} \cdot 2^{-(k+1)} = \frac{1}{8}, && (|A^{\text{free}}| \leq \varepsilon \cdot 2^{k-1} \leq 2^{k-1})
\end{aligned}$$

and a similar calculation shows $\mu_\ell \leq \frac{7}{8}$. We conclude that the error of the decision tree T over distribution $\mathcal{D}$ is at least $\frac{1}{8} = \Omega(1)$. ◀

7 Conclusion

We have constructed target functions for which greedy decision tree learning heuristics fail badly, even in the smoothed setting. Our lower bounds complement and strengthen the parity-of-two-features example discussed in the introduction, which showed that these heuristics fail badly in the non-smoothed setting.

It can be reasonably argued that real-world data sets do not resemble the target functions considered in this paper or the parity-of-two-features example. Perhaps the sought-for guarantee ($\diamondsuit$), while false for certain target functions even in the smoothed setting, is nonetheless true for broad and natural classes of targets? It would be interesting to reexamine, through the lens of smoothed analysis, provable guarantees for restricted classes of functions that have been established. For example, can the guarantees of [3, 2] for monotone target functions and product distributions be further strengthened in the smoothed setting? The target functions considered in this paper, as well as the parity-of-two-features example, are non-monotone.

References

1 Guy Blanc, Neha Gupta, Jane Lange, and Li-Yang Tan. Universal guarantees for decision tree induction via a higher-order splitting criterion. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

2 Guy Blanc, Jane Lange, and Li-Yang Tan. Provable guarantees for decision tree induction: the agnostic setting. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020. Available at `https://arxiv.org/abs/2006.00743`.

3 Guy Blanc, Jane Lange, and Li-Yang Tan. Top-down induction of decision trees: rigorous guarantees and inherent limitations. In *Proceedings of the 11th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 151, pages 1–44, 2020.

4 Avirm Blum, Merrick Furst, Jeffrey Jackson, Michael Kearns, Yishay Mansour, and Steven Rudich. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing (STOC)*, pages 253–262, 1994.

5 Avrim Blum. Rank-r decision trees are a subclass of r-decision lists. *Inform. Process. Lett.*, 42(4):183–185, 1992. `doi:10.1016/0020-0190(92)90237-P`.

6 Leo Breiman, Jerome Friedman, Charles Stone, and Richard Olshen. *Classification and regression trees*. Wadsworth International Group, 1984.

7 Alon Brutzkus, Amit Daniely, and Eran Malach. On the Optimality of Trees Generated by ID3. *ArXiv*, abs/1907.05444, 2019.

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

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

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

11 Tom Dietterich, Michael Kearns, and Yishay Mansour. Applying the weak learning framework to understand and improve C4.5. In *Proceedings of the 13th International Conference on Machine Learning (ICML)*, pages 96–104, 1996.

12 Andrzej Ehrenfeucht and David Haussler. Learning decision trees from random examples. *Information and Computation*, 82(3):231–246, 1989.

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

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

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

16 Thomas Hancock, Tao Jiang, Ming Li, and John Tromp. Lower bounds on learning decision lists and trees. *Information and Computation*, 126(2):114–122, 1996.

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

18 Jeffrey C. Jackson and Rocco A. Servedio. On learning random dnf formulas under the uniform distribution. *Theory of Computing*, 2(8):147–172, 2006. doi:10.4086/toc.2006.v002a008.

19 Adam Kalai, Alex Samorodnitsky, and Shang-Hua Teng. Learning and smoothed analysis. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 395–404, 2009.

20 Michael Kearns. 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)*, pages 1337–1339, 1996.

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

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

23 Adam Klivans and Rocco Servedio. Toward attribute efficient learning of decision lists and parities. *Journal of Machine Learning Research*, 7(Apr):587–602, 2006.

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

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

26 Dinesh Mehta and Vijay Raghavan. Decision tree approximations of boolean functions. *Theoretical Computer Science*, 270(1-2):609–623, 2002.

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

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

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

30 Ronald Rivest. Learning decision lists. *Machine learning*, 2(3):229–246, 1987.

31 Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51(3):385–463, 2004.

32 Ian Witten, Eibe Frank, Mark Hall, and Christopher Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

33 Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, S Yu Philip, et al. Top 10 algorithms in data mining. *Knowledge and information systems*, 14(1):1–37, 2008.