Learning Stochastic Decision Trees

Guy Blanc
Stanford University, CA, USA

Jane Lange
MIT, Cambridge, MA, USA

Li-Yang Tan
Stanford University, CA, USA

Abstract

We give a quasipolynomial-time algorithm for learning *stochastic decision trees* that is optimally resilient to adversarial noise. Given an η-corrupted set of uniform random samples labeled by a size-s stochastic decision tree, our algorithm runs in time $n^{O(\log(s/\varepsilon)/\varepsilon^2)}$ and returns a hypothesis with error within an additive $2\eta+\varepsilon$ of the Bayes optimal. An additive 2η is the information-theoretic minimum.

Previously no non-trivial algorithm with a guarantee of $O(\eta)+\varepsilon$ was known, even for weaker noise models. Our algorithm is furthermore proper, returning a hypothesis that is itself a decision tree; previously no such algorithm was known even in the noiseless setting.

2012 ACM Subject Classification Theory of computation → Boolean function learning

Keywords and phrases Learning theory, decision trees, proper learning algorithms, adversarial noise

Digital Object Identifier 10.4230/LIPIcs.ICALP.2021.30

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://arxiv.org/pdf/2105.03594.pdf`

1 Introduction

Decision trees are a touchstone class in learning theory. There is by now a rich and vast literature on the problem of learning decision trees, spanning three decades and studying it in a variety of models and from a variety of perspectives [10, 32, 5, 16, 7, 27, 4, 17, 23, 30, 20, 31, 15, 28, 26, 22, 19, 9, 2, 3, 1, 6].

We consider the problem of learning *stochastic decision trees*, a generalization of standard deterministic decision trees that allows for stochastic nodes. This generalization broadens the expressive power of decision trees, enabling them to represent not just deterministic functions but also stochastic functions. Figure 1 depicts a stochastic decision tree with two stochastic nodes, labeled "\$", one that branches on the outcome of a Bernoulli(0.8) random variable, and the other on the outcome of a Bernoulli(0.3) random variable.

Many real-world learning scenarios are inherently stochastic in nature, and relatedly, much of current research in learning theory focuses on the "probabilistic concept" generalization [24] of the standard PAC model of learning deterministic concepts (e.g. see [14, 13, 11, 12] for an ongoing line of work on learning neural networks in the probabilistic concept model). As discussed in [24], probabilistic concepts can also be viewed as latent variable models, where the uncertainty concerning latent variables is modeled as apparent probabilistic behavior.

Stochastic decision trees are a simple and natural way to represent stochastic functions. Despite compelling theoretical and practical motivations, there has thus far been considerably less attention on the problem learning stochastic decision trees as compared to deterministic decision trees. Many basic questions remain open; for example:

48th International Colloquium on Automata, Languages, and Programming (ICALP 2021).
Editors: Nikhil Bansal, Emanuela Merelli, and James Worrell; Article No. 30; pp. 30:1–30:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

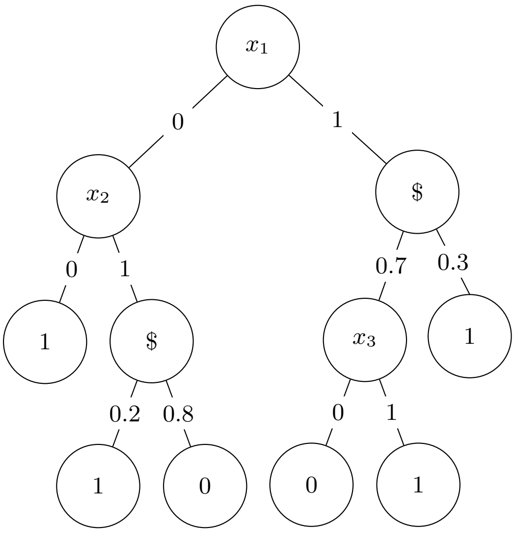

Figure 1 A stochastic decision tree with two stochastic nodes.

- Is there an algorithm for *properly* learning stochastic decision trees, one that returns a decision tree hypothesis?
- Is there an algorithm for learning stochastic decision trees that is resilient to *adversarial noise*?

These questions have been intensively studied in the case of deterministic decision trees, and the algorithms and techniques developed to answer them (e.g. [10, 21, 15]) have become foundational results in learning theory. A broad goal of our work is to help bring the state of our understanding of learning stochastic decision trees into closer alignment with that of deterministic decision trees.

1.1 Our results

We give new algorithms for learning stochastic decision trees under the uniform distribution. En route to our main result, we give the first algorithm for *properly* learning stochastic decision trees – our algorithm in fact returns a deterministic decision tree hypothesis:

▶ **Theorem 1** (Properly learning stochastic decision trees). *There is an algorithm $\mathcal{A}$ with the following guarantee. For all $\varepsilon \in (0,1)$ and $s \in \mathbb{N}$, given access to labeled samples $(\boldsymbol{x}, \boldsymbol{T}(\boldsymbol{x}))$ where $\boldsymbol{T} : \{0,1\}^n \to \{0,1\}$ is a size-s stochastic decision tree and $\boldsymbol{x}$ is uniform random, $\mathcal{A}$ runs in $n^{O(\log(s/\varepsilon)/\varepsilon^2)}$ time and with high probability outputs a deterministic decision tree h such that $\Pr[h(\boldsymbol{x}) \neq \boldsymbol{T}(\boldsymbol{x})] \leq \mathrm{opt} + \varepsilon$, where opt denotes the Bayes optimal error for $\boldsymbol{T}$.*

Theorem 1 is a special case of our main result, which gives a generalization of the algorithm $\mathcal{A}$ of Theorem 1 that is optimally resilient to adversarial noise.

▶ **Definition 2** (η-corrupted samples; "nasty noise" [8]). *Let $\boldsymbol{f} : \{0,1\}^n \to \{0,1\}$ be a stochastic function. We say that $\mathcal{S}$ is an η-corrupted set of uniform random samples labeled by $\boldsymbol{f}$ if it is formed in the following fashion: draw a set of labeled samples $(\boldsymbol{x}, \boldsymbol{f}(\boldsymbol{x}))$ where $\boldsymbol{x}$ is uniform random, and modify any η fraction to form $\mathcal{S}$.*

We allow for corruptions of both the example (i.e. changing $\boldsymbol{x}$ to a different $\boldsymbol{x}'$) and its label (i.e. flipping $\boldsymbol{f}(\boldsymbol{x})$), and note that the adversarial choice of which η fraction of samples to corrupt can be adaptive, depending arbitrarily on the original uncorrupted set of samples. This is regarded as the most challenging noise model for classification problems; weaker noise models include random classification noise, Massart noise, and agnostic noise.

Our main result is as follows:

▶ **Theorem 3** (Our main result: Properly learning stochastic decision trees in the presence of adversarial noise). *There is an algorithm $\mathcal{A}$ with the following guarantee. For all $\varepsilon, \eta \in (0,1)$ and $s \in \mathbb{N}$, given access to a sufficiently large η-corrupted set $\mathcal{S}$ of uniform random samples labeled by a size-s stochastic decision tree $\boldsymbol{T} : \{0,1\}^n \to \{0,1\}$, $\mathcal{A}$ runs in $n^{O(\log(s/\varepsilon)/\varepsilon^2)}$ time and with high probability outputs a decision tree hypothesis h such that $\Pr[h(\boldsymbol{x}) \neq \boldsymbol{T}(\boldsymbol{x})] \leq \mathrm{opt} + 2\eta + \varepsilon$, where opt denotes the Bayes optimal error for $\boldsymbol{T}$.*

An error of $\mathrm{opt} + 2\eta$ is the information-theoretic minimum (see e.g. [8]). Prior to our work there were (improper) algorithms that achieved either $\mathrm{opt} + O(\sqrt{\eta}) + \varepsilon$ or $2\,\mathrm{opt} + 2\eta + \varepsilon$, the low-degree algorithm of [29] and the L_1 polynomial regression algorithm of [21] respectively, but not the information-theoretically optimal $\mathrm{opt} + 2\eta + \varepsilon$. This was the case even for weaker noise models such as label-only noise (i.e. agnostic noise [18, 25]). In fact, the low-degree and L_1 polynomial regression algorithms are, in general, only known to be resilient to noise in the labels.

As our final contribution, we show that when applied in the context of decision tree learning, these algorithms are in fact resilient to noise in both the examples and their labels:

▶ **Theorem 4** (Noise-tolerant properties of the low-degree algorithm and L_1 polynomial regression). *For all $\varepsilon, \eta \in (0,1)$ and $s \in \mathbb{N}$, given access to a sufficiently large η-corrupted set $\mathcal{S}$ of uniform random samples labeled by a size-s stochastic decision tree $\boldsymbol{T} : \{0,1\}^n \to \{0,1\}$,*

- *the low-degree algorithm runs in time $n^{O(\log(s/\varepsilon))}$ and with high probability outputs a hypothesis h satisfying $\Pr[h(\boldsymbol{x}) \neq \boldsymbol{T}(\boldsymbol{x})] \leq \mathrm{opt} + O(\sqrt{\eta}) + \varepsilon$.*
- *the L_1 polynomial regression algorithm runs in time $n^{O(\log(s/\varepsilon))}$ and with high probability outputs a stochastic hypothesis $\boldsymbol{h}$ satisfying $\Pr[\boldsymbol{h}(\boldsymbol{x}) \neq \boldsymbol{T}(\boldsymbol{x})] \leq 2\,\mathrm{opt} + 2\eta + \varepsilon$.*

1.1.1 Summary and comparison with existing algorithms

The low-degree algorithm of Linial, Mansour, and Nisan [29] and a recent algorithm of Chen and Moitra [9] for learning mixtures of subcubes can both be used to learn stochastic decision trees as a special case of their main results. The algorithm of [29] runs in time $n^{O(\log(s/\varepsilon))}$, whereas the algorithm of [9] runs in time $O_s(1) \cdot n^{O(\log s)} \cdot \mathrm{poly}(1/\varepsilon)$. However, neither of these algorithms returns a decision tree hypothesis, and hence both are improper when applied in this context. The classic algorithm of Ehrenfeucht and Haussler [10, 5] properly learns deterministic decision trees in time $n^{O(\log s)} \cdot \mathrm{poly}(1/\varepsilon)$. However, being an Occam algorithm, its analysis seems fundamentally unable to accommodate stochasticity of the target concept.

Table 1 summarizes our contributions and places them in the context of prior work.

1.2 Our techniques

Our approach to Theorems 1 and 3 is simple and has two main conceptual parts: a structural lemma concerning stochastic decision trees and a noise-tolerant algorithm for learning a special type of stochastic decision tree.

Table 1 Performance guarantees of our algorithm and existing algorithms for learning stochastic decision trees in the presence of adversarial noise. Among these algorithms, ours is the only one that returns a decision tree hypothesis. Prior to our work, the error guarantees for the low-degree algorithm and L_1 polynomial regression were only known for label noise; we show in the context of decision tree learning, these guarantees can be strengthened to allow for noise in both the examples and labels.

Reference	Technique	Running time	Error guarantee
[29]	Low-degree algorithm	$n^{O(\log(s/\varepsilon))}$	$\mathrm{opt} + O(\sqrt{\eta}) + \varepsilon$ (**This work**)
[21]	L_1 polynomial regression	$n^{O(\log(s/\varepsilon))}$	$2\,\mathrm{opt} + 2\eta + \varepsilon$ (**This work**)
[9]	Learning mixtures of subcubes	$O_s(1) \cdot n^{O(\log s)} \cdot \mathrm{poly}(\frac{1}{\varepsilon})$	$\mathrm{opt} + \varepsilon$ Noiseless setting ($\eta = 0$)
This work	Approximation by stochastic-leaf DTs; Noise-tolerant learning of stochastic-leaf DTs	$n^{O(\log(s/\varepsilon)/\varepsilon^2)}$	$\mathrm{opt} + 2\eta + \varepsilon$

- *Structural lemma:* We show that every size-s stochastic decision tree can be ε-approximated by a "stochastic-leaf decision tree" of size $s^{O(1/\varepsilon^2)}$. A stochastic-leaf decision tree is a very specific type of stochastic decision tree, one whose stochastic nodes only occur at its leaves.

This lemma reduces the task of learning stochastic decision trees to that of learning stochastic-leaf decision trees, with a catch: due to the approximation error incurred, the algorithm for learning stochastic-leaf decision trees has to be *noise-tolerant*.

- *Noise-tolerant learning stochastic-leaf decision trees:* Mehta and Raghavan [30] gave an algorithm for properly learning deterministic decision trees in the noiseless setting. We show that their algorithm can be generalized to handle stochastic-leaf decision trees, and furthermore, we show that our generalization is optimally resilient to adversarial noise. This stands in contrast to the algorithm of Ehrenfeucht and Haussler [10], which as mentioned above seems fundamentally unable to accommodate either stochasticity or noise.

We are hopeful that each of these two parts will see further utility in problems involving stochastic decision trees, beyond the learning-theoretic setting that is the focus of this work.

As for Theorem 4, the low-degree algorithm and L_1 polynomial regression are versatile and powerful "meta-algorithms" in learning, but they are not generally known to handle the challenging nasty noise. Our key observation here is that the mean functions of stochastic decision trees are well-approximated by low-degree polynomials *with bounded outputs*. We then show that when run on such polynomials, the low-degree algorithm and L_1 polynomial regression are in fact resilient to nasty noise. Given the broad applicability of both algorithms, we are similarly hopeful that this fact will be of independent interest beyond decision trees.

1.3 Preliminaries

Let $\boldsymbol{f} : \{0,1\}^n \to \{0,1\}$ be a stochastic function. We associate $\boldsymbol{f}$ with its *mean function* $\mu_{\boldsymbol{f}} : \{0,1\}^n \to [0,1]$, $\mu_{\boldsymbol{f}}(x) := \Pr_{\boldsymbol{f}}[\boldsymbol{f}(x) = 1]$. The *Bayes optimal classifier* for $\boldsymbol{f}$ is the (deterministic) function $x \mapsto \mathrm{round}(\mu_{\boldsymbol{f}}(x))$, where $\mathrm{round}(t) := \mathbb{1}[t \geq \frac{1}{2}]$. Given two stochastic functions $\boldsymbol{f}, \boldsymbol{h} : \{0,1\}^n \to \{0,1\}$, we define

$$\mathrm{error}_{\boldsymbol{f}}(\boldsymbol{h}) \coloneqq \mathop{\mathbb{E}}_{\boldsymbol{x}}\left[\Pr_{\boldsymbol{f},\boldsymbol{h}}[\boldsymbol{f}(\boldsymbol{x}) \neq \boldsymbol{h}(\boldsymbol{x})]\right],$$

where here and throughout this paper, $\boldsymbol{x}$ denotes a uniform random input from $\{0,1\}^n$. We define $\mathrm{opt}_{\boldsymbol{f}} \coloneqq \mathrm{error}_{\boldsymbol{f}}(\mathrm{round}(\mu_{\boldsymbol{f}}))$, and when $\boldsymbol{f}$ is clear from context, we simply write opt.

▶ **Fact 5** (Bayes optimal classifier minimizes classification error)**.** *For all stochastic functions* $\boldsymbol{f}, \boldsymbol{h} : \{0,1\}^n \to \{0,1\}$*, we have* $\mathrm{error}_{\boldsymbol{f}}(\boldsymbol{h}) \geq \mathrm{opt}_{\boldsymbol{f}}$.

▶ **Fact 6** (L_1-error and Bayes optimality)**.** *Let* $\boldsymbol{f} : \{0,1\}^n \to \{0,1\}$ *be a stochastic function. For any* $h : \{0,1\}^n \to [0,1]$,

$$\Pr[\mathrm{round}(h(\boldsymbol{x})) \neq \boldsymbol{f}(\boldsymbol{x})] \leq \mathrm{opt}_{\boldsymbol{f}} + 2\,\mathbb{E}[|\mu_{\boldsymbol{f}}(\boldsymbol{x}) - h(\boldsymbol{x})|].$$

Fact 6 states that if we have a function close to $\mu_{\boldsymbol{f}}$, we can convert it to a classifier with error close to $\mathrm{opt}_{\boldsymbol{f}}$.

Proof. We need to upper bound $\mathrm{error}_{\boldsymbol{f}}(\mathrm{round} \circ h) - \mathrm{opt}_{\boldsymbol{f}}$ at $2\,\mathbb{E}[|\mu_{\boldsymbol{f}}(\boldsymbol{x}) - h(\boldsymbol{x})|]$. We rewrite that quantity as

$$\begin{aligned}
\mathrm{error}_{\boldsymbol{f}}(\mathrm{round} \circ h) - \mathrm{opt}_{\boldsymbol{f}} &= \Pr_{\boldsymbol{x} \sim \{0,1\}^n}[\boldsymbol{f}(\boldsymbol{x}) \neq \mathrm{round}(h(\boldsymbol{x}))] - \Pr_{\boldsymbol{x} \sim \{0,1\}^n}[\boldsymbol{f}(x) \neq \mathrm{round}(\mu_{\boldsymbol{f}}(\boldsymbol{x}))] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \{0,1\}^n}[|\, \mu_{\boldsymbol{f}}(\boldsymbol{x}) - \mathrm{round}(h(\boldsymbol{x}))\,| - |\, \mu_{\boldsymbol{f}}(\boldsymbol{x}) - \mathrm{round}(\mu_{\boldsymbol{f}}(\boldsymbol{x}))\,|] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \{0,1\}^n}\left[\mathbb{1}(\mathrm{round}(h(\boldsymbol{x}))) \neq \mathrm{round}(\mu_{\boldsymbol{f}}(\boldsymbol{x}))) \cdot 2 \cdot \left|\, \mu_{\boldsymbol{f}}(\boldsymbol{x}) - \tfrac{1}{2}\,\right|\right]
\end{aligned}$$

It is only possible that $\mathrm{round}(h(x)) \neq \mathrm{round}(\mu_{\boldsymbol{f}}(x))$ if $|\, f(x) - \mu_{\boldsymbol{f}}(x)\,| \geq |\, \mu_{\boldsymbol{f}}(x) - \frac{1}{2}\,|$. Therefore,

$$\begin{aligned}
\mathrm{error}_{\boldsymbol{f}}(\mathrm{round} \circ h) - \mathrm{opt}_{\boldsymbol{f}} &\leq \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \{0,1\}^n}\left[\mathbb{1}(|\, f(x) - \mu_{\boldsymbol{f}}(x)\,| \geq |\, \mu_{\boldsymbol{f}}(x) - \frac{1}{2}\,|) \cdot 2 \cdot \left|\, \mu_{\boldsymbol{f}}(\boldsymbol{x}) - \tfrac{1}{2}\,\right|\right] \\
&\leq 2 \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \{0,1\}^n}[|\, f(x) - \mu_{\boldsymbol{f}}(x)\,|]\,. \qquad ◀
\end{aligned}$$

2 Approximating stochastic DTs with stochastic-leaf DTs

▶ **Definition 7** (Stochastic-leaf DT)**.** *A stochastic-leaf DT is a stochastic DT for which all stochastic nodes have only leaves as their children.*

▶ **Lemma 8** (Approximating stochastic DTs with stochastic-leaf DTs)**.** *Let* $\boldsymbol{T}$ *be a size-*s *stochastic DT. For every* $\varepsilon \in (0, \frac{1}{2})$*, there is a size-*$S$ stochastic-leaf *DT* $\overline{\boldsymbol{T}}$ *such that* $S \leq s^{O(1/\varepsilon^2)}$ *and* $\mathbb{E}_{\boldsymbol{x}}[|\mu_{\boldsymbol{T}}(\boldsymbol{x}) - \mu_{\overline{\boldsymbol{T}}}(\boldsymbol{x})|] \leq \varepsilon$.

Proof. Let m denote the number of stochastic transitions in $\boldsymbol{T}$. For a fixed $r \in \{0,1\}^m$, let $\boldsymbol{T}(x, r)$ be the value of $\boldsymbol{T}$ evaluated on x with stochastic transitions determined by r. Suppose we pick random strings $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_c \sim \{0,1\}^m$ independently and uniformly at random. For each $x \in \{0,1\}^n$, consider the following random variable:

$$\mathbf{est}(x) \coloneqq \mathop{\mathbb{E}}_{\boldsymbol{i} \in [c]}[\boldsymbol{T}(x, \boldsymbol{r}_{\boldsymbol{i}})].$$

Note that

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{r}_1,\ldots,\boldsymbol{r}_c \in \{0,1\}^m}[\mathbf{est}(x)] = \mu_{\boldsymbol{T}}(x) &= \mathop{\mathbb{E}}_{\boldsymbol{r} \sim \{0,1\}^m}[\boldsymbol{T}(x, \boldsymbol{r})] \\
\mathrm{Var}[\mathbf{est}(x)] &= \tfrac{1}{c} \cdot \mathbf{Var}_{\boldsymbol{r} \sim \{0,1\}^m}[\boldsymbol{T}(x, \boldsymbol{r})],
\end{aligned}$$

where in both cases above, $\boldsymbol{r} \sim \{0,1\}^m$ on the RHS denotes $\boldsymbol{r}$ chosen uniformly at random from $\{0,1\}^m$. Since $\boldsymbol{T}$ is $\{0,1\}$-valued, it has variance at most $\frac{1}{4}$. Hence, the variance of $\mathbf{est}(x)$ is at most $\frac{1}{4c}$. If we take $c = 1/\varepsilon^2$, the following holds for any $x \in \{0,1\}^n$:

$$\mathop{\mathbb{E}}_{\boldsymbol{r}_1,\ldots,\boldsymbol{r}_c\sim\{0,1\}^m}\left[\left(\mathbf{est}(x) - \mu_{\boldsymbol{T}}(x)\right)^2\right] \leq \frac{\varepsilon^2}{4}, \quad \text{and therefore} \quad \mathop{\mathbb{E}}_{\boldsymbol{r}_1,\ldots,\boldsymbol{r}_c\sim\{0,1\}^m}[|\mathbf{est}(x) - \mu_{\boldsymbol{T}}(x)|] \leq \frac{\varepsilon}{2}.$$

Averaging over $\boldsymbol{x} \sim \{0,1\}^n$ and swapping expectations, we get:

$$\mathop{\mathbb{E}}_{\boldsymbol{r}_1,\ldots,\boldsymbol{r}_c\sim\{0,1\}^m}\left[\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\{0,1\}^n}\left[\left|\mathbf{est}(x) - \mu_{\boldsymbol{T}}(x)\right|\right]\right] \leq \frac{\varepsilon}{2}.$$

Therefore, there must exist outcomes $r_1^\star, \ldots, r_c^\star \in \{0,1\}^m$ of $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_c$ such that

$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\{0,1\}^n}\left[\left|\mathop{\mathbb{E}}_{\boldsymbol{i}\in[c]}[\boldsymbol{T}(\boldsymbol{x}, r_{\boldsymbol{i}^\star})] - \mu_{\boldsymbol{T}}(\boldsymbol{x})\right|\right] \leq \frac{\varepsilon}{2}. \tag{1}$$

For each $i \in [c]$, we define a size-s DT by fixing the stochastic nodes of $\boldsymbol{T}$ according to $r_i^\star \in \{0,1\}^m$. We define our stochastic-leaf DT $\overline{\boldsymbol{T}}$ by stacking these c many size-s DTs on top of one another: for each $i < n$, we replace each leaf of the i_{th} DT with a copy of the $(i+1)_{th}$ DT. Then for each leaf ℓ of this stacked tree, let x_ℓ be an input that is consistent with the root-to-ℓ path in $\overline{\boldsymbol{T}}$. We replace ℓ with a stochastic node which transitions to a 1-leaf with probability $p_\ell := \mathbb{E}_{\boldsymbol{i}\in[c]}[\boldsymbol{T}(x_\ell, r_{\boldsymbol{i}}^\star)]$, and to a 0-leaf with probability $1 - p_\ell$. Note that for each $i \in [c]$, the tree $\boldsymbol{T}(\cdot, r_i^\star)$ gives the same classification for all inputs reaching leaf ℓ of $\overline{\boldsymbol{T}}$, so p_ℓ does not depend on the choice of x_ℓ.

$\overline{\boldsymbol{T}}$ is a stochastic-leaf DT that computes $x \mapsto \mathbb{E}_{\boldsymbol{i}\in[c]}[\boldsymbol{T}(\boldsymbol{x}, r_{\boldsymbol{i}}^\star)]$, which by Equation (1), has sufficiently small error. Since this DT has size $s^c = s^{O(1/\varepsilon^2)}$, the proof of Lemma 8 is complete. ◀

3 A simple backtracking algorithm for finding the optimal small-depth tree

The algorithmic core of Theorems 1 and 3 is a recursive backtracking procedure Find shown in Algorithm 1, which takes a labeled set of samples X and finds a depth-d decision tree that achieves minimal classification error. This algorithm is inspired by and simplifies the Find algorithm given by Mehta and Raghavan [30] for building a minimum-error decision tree from any "sat-countable representation" of a function.

▶ **Lemma 9** (Correctness of Find). *Consider any sample set X of labeled examples (x, y) and depth budget d. The algorithm Find(X, d) (see Algorithm 1) returns a depth-d DT $T^\star$ that minimizes $\Pr_{(\boldsymbol{x},\boldsymbol{y})\sim X}[T^\star(\boldsymbol{x}) \neq \boldsymbol{y}]$ among all depth-d DTs.*

Proof. We proceed by induction on d. If $d = 0$, then Find returns at Step 1 and is clearly correct. For the inductive step, suppose that $d \geq 1$. For any $i \in [n]$, we first claim that the tree T_i defined in Step 2 is a depth d DT that minimizes classification error with respect to X among those that query x_i at the root. Let $(T_i)_{\text{left}}$ and $(T_i)_{\text{right}}$ be its left and right subtrees respectively. By the inductive hypothesis, the left and right subtrees $(T_i)_{\text{left}}$ and $(T_i)_{\text{right}}$ are depth $d-1$ DTs that minimize error with respect to $X_{x_i=0}$ and $X_{x_i=1}$ respectively. Hence, T_i is a depth d DT that achieves minimal error with respect to X among those that query x_i at the root.

Since Find returns the $T_{i^\star}$ that minimizes $\Pr_{(\boldsymbol{x},\boldsymbol{y})\sim X}[T_i(\boldsymbol{x}) \neq \boldsymbol{y}]$ among all $i \in [n]$ in Step 3, and each T_i is a minimal-error depth-d DT among those that query x_i at the root, we conclude that Find returns a tree of minimal error with respect to X. ◀

Algorithm 1 A recursive backtracking algorithm for finding a depth-d DT of minimal classification error.

```
FIND(X, d):
    Input: Set $X$ of labeled examples $(x, y)$ and depth budget $d$.
    Output: A depth-$d$ DT $T^\star$ that minimizes $\Pr_{(\boldsymbol{x},\boldsymbol{y})\sim X}[T^\star(\boldsymbol{x}) \neq \boldsymbol{y}]$ among all depth-$d$ DTs.

1. If $d = 0$, return the constant $c \in \{0, 1\}$ that minimizes $\Pr_{(\boldsymbol{x},\boldsymbol{y})\sim X}[c \neq \boldsymbol{y}]$.
2. For every $i \in [n]$, let $T_i$ be the DT defined as follows:
     - $T_i$ queries $x_i$ at the root;
     - Has FIND($X_{x_i=0}, d-1$) as its left subtree;
     - Has FIND($X_{x_i=1}, d-1$) as its right subtree.
   Here $X_{x_i=b}$ denotes the subset of $X$ containing only examples where $x_i$ is set to $b$.
3. Return the tree $T_{i^\star}$ that minimizes $\Pr_{(\boldsymbol{x},\boldsymbol{y})\in X}[T_i(\boldsymbol{x}) \neq \boldsymbol{y}]$ among all $i \in [n]$.
```

▶ **Lemma 10** (Efficiency of Find). *Consider any sample set X of labeled examples and depth budget d. The algorithm* Find(X, d) *(see Algorithm 1) takes time $n^{O(d)} \cdot O(|X|)$.*

Proof. Let $T(d)$ denote the running time of Find when run with depth budget d. If $d = 0$ then the algorithm only executes Step 1, which can be done in $O(|X|)$ time by computing $\mathrm{round}(\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim X}[\boldsymbol{y}])$.

Next we consider the case of $d \geq 1$. In step 2, Find recurses $2n$ times, each with d decremented by one. Each time it also partitions X into $X_{x_i=0}$ and $X_{x_i=1}$. All of these recursive calls and partitioning takes total time $2n \cdot T(d-1) + n|X|$. In step 3, Find must compute $\Pr_{(\boldsymbol{x},\boldsymbol{y})\sim X}[T_i(\boldsymbol{x}) \neq \boldsymbol{y}]$ for up to n different coordinates i, where each T_i has depth at most d. This takes time $n \cdot d \cdot |X|$. We therefore have the recurrence relation:

$$T(d) \leq 2n \cdot T(d-1) + O(nd|X|).$$

Solving this recurrence relation gives us the bound $T(d) \leq (2n)^d \cdot O(nd|X|)$, which is $\leq n^{O(d)} \cdot O(|X|)$ as desired. ◀

4 Learning stochastic DTs: proofs of Theorems 1 and 3

4.1 Proof of Theorem 1

We recall Theorem 1, this time including the confidence parameter δ.

▶ **Theorem 1** (Properly learning stochastic decision trees). *There is an algorithm $\mathcal{A}$ with the following guarantee. For all $\varepsilon \in (0, 1)$ and $s \in \mathbb{N}$, given access to labeled samples $(\boldsymbol{x}, \boldsymbol{T}(\boldsymbol{x}))$ where $\boldsymbol{T} : \{0, 1\}^n \to \{0, 1\}$ is a size-s stochastic decision tree and $\boldsymbol{x}$ is uniform random, $\mathcal{A}$ runs in $n^{O(\log(s/\varepsilon)/\varepsilon^2)} \cdot \mathrm{poly}(\log(1/\delta))$ time and with probability $1 - \delta$ outputs a deterministic decision tree h such that $\Pr[h(\boldsymbol{x}) \neq \boldsymbol{T}(\boldsymbol{x})] \leq \mathrm{opt} + \varepsilon$, where* opt *denotes the Bayes optimal error for $\boldsymbol{T}$.*

Let $\boldsymbol{T}$ be a size-s stochastic decision tree. By Lemma 8, there is a stochastic-leaf decision tree $\overline{\boldsymbol{T}}$ of size $S \leq s^{O(1/\varepsilon^2)}$ such that $\mathbb{E}_{\boldsymbol{x}}[|\mu_{\boldsymbol{T}}(\boldsymbol{x}) - \mu_{\overline{\boldsymbol{T}}}(\boldsymbol{x})|] \leq \varepsilon$. Consider the Bayes optimal classifier $x \mapsto \mathrm{round}(\mu_{\overline{\boldsymbol{T}}}(x))$ for $\overline{\boldsymbol{T}}$. Since $\overline{\boldsymbol{T}}$ is a *stochastic-leaf* decision tree, we have that this function is computed by a size-S (deterministic) decision tree $T^\star$: to obtain $T^\star$ from $\overline{\boldsymbol{T}}$, simply replace every stochastic node in $\overline{\boldsymbol{T}}$, all of which occur at the leaves of $\overline{\boldsymbol{T}}$, with a 1-leaf if it branches on Bernoulli(p) where $p \geq \frac{1}{2}$, and a 0-leaf otherwise. Applying Fact 6, we get that

$$\Pr_{\boldsymbol{x},\boldsymbol{T}}[T^\star(\boldsymbol{x}) \neq \boldsymbol{T}(\boldsymbol{x})] \leq \mathrm{opt}_{\boldsymbol{T}} + 2\varepsilon.$$

Next, consider the decision tree $T^\star_{\mathrm{trunc}}$ obtained by truncating $T^\star$ to depth $\log(S/\varepsilon)$ (and replacing all truncated branches with a leaf with an arbitrary value, say a 1-leaf). $T^\star_{\mathrm{trunc}}$ and $T^\star$ can only differ on inputs that reach a leaf in $T^\star$ of depth at least $\log(S/\varepsilon)$, and there are at most S such leaves. Therefore,

$$\Pr_{\boldsymbol{x}}[T^\star_{\mathrm{trunc}}(\boldsymbol{x}) \neq T^\star(\boldsymbol{x})] \leq 2^{-\log(S/\varepsilon)} \cdot S = \varepsilon.$$

Note that the depth of $T^\star_{\mathrm{trunc}}$ is $\leq \log(s/\varepsilon)/\varepsilon^2$. We have shown the following corollary of Lemma 8:

▶ **Corollary 11** (Approximating stochastic DTs with deterministic ones). *Let $\boldsymbol{T} : \{0,1\}^n \to \{0,1\}$ be a size-s stochastic DT. For every $\varepsilon \in (0, \frac{1}{2})$, there is a deterministic DT $T^\star_{\mathrm{trunc}} : \{0,1\}^n \to \{0,1\}$ such that*

1. $\mathrm{depth}(T^\star_{\mathrm{trunc}}) \leq \log(S/\varepsilon) \leq \log(s/\varepsilon)/\varepsilon^2$ *and*
2. $\Pr_{\boldsymbol{x},\boldsymbol{T}}[T^\star_{\mathrm{trunc}}(\boldsymbol{x}) \neq \boldsymbol{T}(\boldsymbol{x})] \leq \mathrm{opt}_{\boldsymbol{T}} + 3\varepsilon.$

To show that Find returns a tree of small error with respect to $\boldsymbol{T}$, we need the following generalization bound from [30]:

▶ **Lemma 12** (Generalization). *Let $\boldsymbol{T}$ be a stochastic tree of size s. For $S = s^{O(1/\varepsilon^2)}$ and a sample size of*

$$m := \mathrm{poly}\left(n^{\log(S/\varepsilon)}, \frac{1}{\varepsilon}, \log\left(\frac{1}{\delta}\right)\right),$$

let X be a dataset of m i.i.d points of the form $(\boldsymbol{x}, \boldsymbol{T}(\boldsymbol{x}))$. Then Find$(X, \log(S/\varepsilon))$ outputs $T^\star$ such that

$$\Pr_{\textit{draw of } \boldsymbol{X}}\Big[\Pr_{\boldsymbol{x}\sim\{0,1\}^n}[T^\star(\boldsymbol{x}) \neq \boldsymbol{T}(\boldsymbol{x})] \leq \mathrm{opt}_{\boldsymbol{T}} + 3\varepsilon\Big] \geq 1 - \delta.$$

Proof. The proof is given in the proof of Theorem 2 in [30]. Lemma 9 gives us that Find outputs a tree of minimal error with respect to X. They apply Chernoff bounds to bound the probability that a fixed tree T' of depth $\log(S/\varepsilon)$ and error $> \mathrm{opt}_{\boldsymbol{T}} + 3\varepsilon$ with respect to $\boldsymbol{T}$ has smaller error with respect to X than $T^\star_{\mathrm{trunc}}$ as described in Corollary 11. More specifically, the probability over draws of X that $\Pr_{(\boldsymbol{x},\boldsymbol{y})\sim X}[T^\star_{\mathrm{trunc}}(\boldsymbol{x}) \neq \boldsymbol{y}] > \mathrm{opt}_{\boldsymbol{T}} + 3\varepsilon$ or $\Pr_{(\boldsymbol{x},\boldsymbol{y})\sim X}[T'(\boldsymbol{x}) \neq \boldsymbol{y}] \leq \mathrm{opt}_{\boldsymbol{T}} + 3\varepsilon$ is exponentially small in $|X|$. This is a bound on the probability that Find outputs a particular tree of error greater than $\mathrm{opt}_{\boldsymbol{T}} + 3\varepsilon$; the lemma follows from a union bound over all trees of depth at most $\log(S/\varepsilon)$. ◀

Lemma 10 gives us that Find$(X, \log(S/\varepsilon))$ runs in time $n^{O(\log(S/\varepsilon))} \cdot O(|X|) = n^{O(\log(s/\varepsilon)/\varepsilon^2)} \cdot O(|X|)$. For confidence parameter δ, $|X|$ is polynomial in $n^{\log(S/\varepsilon)}$, $\log(1/\varepsilon)$, and $\log(1/\delta)$. Thus, the total runtime of Find is $n^{O(\log(s/\varepsilon)/\varepsilon^2)} \cdot \mathrm{poly}\log(1/\delta))$. The desired result holds by renaming $\varepsilon' = \varepsilon/3$. ◀

4.2 Proof of Theorem 3

We recall Theorem 3, this time including the confidence parameter δ.

▶ **Theorem 3** (Our main result). *There is an algorithm $\mathcal{A}$ with the following guarantee. For all $\varepsilon, \eta \in (0,1)$ and $s \in \mathbb{N}$, given access to a sufficiently large η-corrupted set $\mathcal{S}$ of uniform random samples labeled by a size-s stochastic decision tree $\boldsymbol{T} : \{0,1\}^n \to \{0,1\}$, $\mathcal{A}$ runs in $n^{O(\log(s/\varepsilon)/\varepsilon^2)} \cdot \mathrm{poly}(\log(1/\delta))$ time and with probability $1-\delta$ outputs a decision tree hypothesis h such that $\Pr[h(\boldsymbol{x}) \neq \boldsymbol{T}(\boldsymbol{x})] \le \mathrm{opt} + 2\eta + \varepsilon$, where opt denotes the Bayes optimal error for $\boldsymbol{T}$.*

The proof requires the following fact.

▶ **Fact 13** (Error from sample corruption). *For any bounded function $p : \{0,1\}^n \to [0,1]$ and sample $\mathcal{S}^\circ$ of points $(x_1, y_1), \dots, (x_m, y_m)$ with $0 \le y_i \le 1$. Let $\mathcal{S}$ be a corrupted sample formed by picking an arbitrary η-fraction of points $\mathcal{S}^\circ$ and replacing each with an arbitrary (also bounded) point. Then for any $\mathrm{err} : [0,1] \times [0,1] \to [0,1]$*

$$\left| \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y}) \sim \mathcal{S}^\circ} [\mathrm{err}(p(\boldsymbol{x}), \boldsymbol{y})] - \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y}) \sim \mathcal{S}} [\mathrm{err}(p(\boldsymbol{x}), \boldsymbol{y})] \right| < \eta.$$

Recall $T^\star_{\mathrm{trunc}}$ as described in Corollary 11, which has error $\le \mathrm{opt}_{\boldsymbol{T}} + O(\varepsilon)$ with respect to $\boldsymbol{T}$. Let $\mathcal{S}^\circ$ be the uncorrupted set of examples of $\boldsymbol{T}$, and $\mathcal{S}$ be an η-corruption of $\mathcal{S}^\circ$. Then with probability $1-\delta$ over draws of $\mathcal{S}^\circ$,

$$\Pr_{(\boldsymbol{x},\boldsymbol{y}) \sim \mathcal{S}^\circ} [T^\star_{\mathrm{trunc}}(\boldsymbol{x}) \neq \boldsymbol{y}] \le \mathrm{opt}_{\boldsymbol{T}} + 3\varepsilon \qquad \text{(Lemma 12)}$$

$$\Pr_{(\boldsymbol{x},\boldsymbol{y}) \sim \mathcal{S}} [T^\star_{\mathrm{trunc}}(\boldsymbol{x}) \neq \boldsymbol{y}] \le \mathrm{opt}_{\boldsymbol{T}} + \eta + 3\varepsilon. \qquad \text{(Fact 13)}$$

Let $T^\star$ be the output of $\textsc{Find}(\mathcal{S}, \log(S/\varepsilon))$. Then,

$$\Pr_{(\boldsymbol{x},\boldsymbol{y}) \sim \mathcal{S}} [T^\star(\boldsymbol{x}) \neq \boldsymbol{y}] \le \mathrm{opt}_{\boldsymbol{T}} + \eta + 3\varepsilon \qquad \text{(Lemma 9)}$$

$$\Pr_{(\boldsymbol{x},\boldsymbol{y}) \sim \mathcal{S}^\circ} [T^\star(\boldsymbol{x}) \neq \boldsymbol{y}] \le \mathrm{opt}_{\boldsymbol{T}} + 2\eta + 3\varepsilon. \qquad \text{(Fact 13)}$$

$$\Pr_{\text{draw of } \mathcal{S}^\circ} \left[\Pr_{x \sim \{0,1\}^n} [T^\star(x) \neq \boldsymbol{T}(x)] \le \mathrm{opt}_{\boldsymbol{T}} + 2\eta + 3\varepsilon \right] > 1 - \delta \qquad \text{(Lemma 12)}$$

The desired result holds by renaming $\varepsilon' = \varepsilon/3$. ◀

5 Noise-tolerant properties of L_1 and L_2 regression

In this section, we prove Theorem 4, showing that the low-degree algorithm of [29] (also known as L_2 regression) and L_1 regression algorithm of [21] both learn stochastic-leaf DTs with adversarial corruption, albeit with worse parameters than our method. Throughout this section, we use the following function.

▶ **Definition 14** (The trunc function). *The function, $\mathrm{trunc} : \mathbb{R} \to [0,1]$, is defined as*

$$\mathrm{trunc}(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x > 1 \\ x & \text{otherwise.} \end{cases}$$

The basis of the results in this section is Proposition 15, that if $\boldsymbol{T}$ is a size-s stochastic DT, there is a degree $\log(s/\varepsilon)$ bounded polynomial $p : \{0,1\}^n \to [0,1]$ which is ε close to $\mu_{\boldsymbol{T}}$:

▶ **Proposition 15** ($\mu_{\boldsymbol{T}}$ is morally low degree). *Let $\boldsymbol{T}$ be a size-s stochastic DT. There is a polynomial $p : \{0,1\}^n \to [0,1]$ such that $\Pr_{\boldsymbol{x}}[p(\boldsymbol{x}) \neq \mu_{\boldsymbol{T}}(\boldsymbol{x})] \leq \varepsilon$, where $\deg(p) \leq \log(s/\varepsilon)$.*

In order to handle our challenging noise model, it is important that we can guarantee the p in Proposition 15 is bounded. Without that guarantee, L_1 and L_2 regression are not known to handle noise in both the examples and the labels.

Proof. For any leaf ℓ of $\boldsymbol{T}$, let $\text{depth}(\ell)$ be the number of *deterministic* nodes on the root-to-leaf path to ℓ, not counting ℓ itself. The fraction of inputs in $\{0,1\}^n$ that have a nonzero chance of reaching ℓ is $2^{-\text{depth}(\ell)}$. Now, let $\boldsymbol{T}'$ be the stochastic decision tree that is nearly equivalent to $\boldsymbol{T}$ except if an input reaches a leaf with deterministic depth more than $\log(s/\varepsilon)$, $\boldsymbol{T}'$ returns 0. We claim that $p := \mu_{\boldsymbol{T}'}$ satisfies Proposition 15. For that, we need to verify three things about $\mu_{\boldsymbol{T}'}$:

1. $\mu_{\boldsymbol{T}'}$ and $\mu_{\boldsymbol{T}}$ are close: $\Pr_{\boldsymbol{x}}[\mu_{\boldsymbol{T}}(\boldsymbol{x}) \neq \mu_{\boldsymbol{T}'}(\boldsymbol{x})] \leq \varepsilon$. This is true because $\boldsymbol{T}$ and $\boldsymbol{T}'$ can differ only on inputs which reach a leaf with deterministic depth at least $\log(s/\varepsilon)$. At most $2^{-\log(s/\varepsilon)} = \varepsilon/s$ fraction of inputs reach each such leaf, and there are at most s of them.
2. $\mu_{\boldsymbol{T}'}$ is a degree $\log(s/\varepsilon)$ polynomial. We can write $\mu_{\boldsymbol{T}'}(x)$ as

$$\begin{aligned}\mu_{\boldsymbol{T}'}(x) &= \sum_{\text{leaves } \ell \in \boldsymbol{T}'} \Pr[x \text{ reaches } \ell] \cdot (\text{label of } \ell)\\ &= \sum_{\text{leaves } \ell \in \boldsymbol{T}} \Pr[x \text{ reaches } \ell] \cdot \mathbb{1}[\text{depth}(\ell) \leq \log(s/\varepsilon)] \cdot (\text{label of } \ell).\end{aligned}$$

   The expression $\Pr[x \text{ reaches } \ell]$ is a degree $\text{depth}(\ell)$ polynomial. Therefore, $\mu_{\boldsymbol{T}'}(x)$ is a degree $\log(s/\varepsilon)$ polynomial.
3. The output of $\mu_{\boldsymbol{T}'}$ is bounded on $[0,1]$. This is true since $\boldsymbol{T}'$ always returns a value in $\{0,1\}$. ◀

5.1 L_2 Regression

Given corrupted samples from some stochastic DT $\boldsymbol{T}$, we will apply Lemma 16, given below, to show that L_2 regression can find a function f that is close to $\mu_{\boldsymbol{T}}$. Then, we will apply Fact 6 to generate a hypothesis with error close to the Bayes optimal error.

▶ **Lemma 16** (L_2 error to mean error). *Fix any stochastic DT $\boldsymbol{T} : \{0,1\}^n \to \{0,1\}$, degree $d \in \mathbb{N}$, and $\varepsilon, \delta > 0$. For a sample size of*

$$m := \text{poly}\left(n^d, 1/\varepsilon, \log(1/\delta)\right),$$

let $\mathcal{S}^\circ$ be a dataset of m i.i.d points of the form $(\boldsymbol{x}, \boldsymbol{T}(\boldsymbol{x}))$. With probability at least $1-\delta$, there exists a constant $C \in \mathbb{R}$ for which the following holds for all degree d polynomials $p : \{0,1\}^n \to \mathbb{R}$.

$$\left| \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{S}^\circ}\left[(\text{trunc}(p(\boldsymbol{x})) - \boldsymbol{y})^2\right] - \left(\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\{0,1\}^n}\left[(\text{trunc}(p(\boldsymbol{x})) - \mu_{\boldsymbol{T}}(\boldsymbol{x}))^2\right] + C\right)\right| \leq \varepsilon. \tag{2}$$

Proof. We prove Lemma 16 in two steps: First, we argue that there is a C for which Equation (2) holds for any fixed polynomial with extremely high probability. Then, we discretize the set of all truncated degree d polynomials into a finite set $\mathcal{P}$. By union bound, we can show that Equation (2) applies to all functions in $\mathcal{P}$, and since every truncated degree d polynomial is sufficiently close to a function in $\mathcal{P}$, this is enough to guarantee that Equation (2) applies to all degree d polynomials.

We use the following identity: For any constant $a \in \mathbb{R}$ and random variable $\boldsymbol{z} \in \mathbb{R}$,

$$\mathop{\mathbb{E}}_{\boldsymbol{z}}\left[(a-\boldsymbol{z})^2\right] = (a-\mathbb{E}[\boldsymbol{z}])^2 + \mathrm{Var}[\boldsymbol{z}].$$

Fix any $p : \{0,1\}^n \to \mathbb{R}$. For any $x \in \{0,1\}^n$, $\boldsymbol{T}(x)$ is a random variable with mean $\mu_{\boldsymbol{T}}(x)$. Therefore,

$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\{0,1\}^n}\left[(\mathrm{trunc}(p(\boldsymbol{x})) - \boldsymbol{T}(x))^2\right] = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\{0,1\}^n}\left[(\mathrm{trunc}(p(\boldsymbol{x})) - \mu_{\boldsymbol{T}}(x))^2\right] + \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\{0,1\}^n}[\mathrm{Var}[\boldsymbol{T}(x)]]$$

For $C = \mathbb{E}_{\boldsymbol{x}\sim\{0,1\}^n}[\mathrm{Var}[\boldsymbol{T}(x)]]$, Equation (2) holds in expectation over $\mathcal{S}$ with $\varepsilon = 0$. Since $(\mathrm{trunc}(p(x)) - y)^2$ is bounded on $[0,1]$, we can apply Hoeffdings inequality: For any fixed p, Equation (2) holds with probability at least $1 - 2\exp_e(-2m^2\varepsilon^2)$.

We next discretize the set of all truncated degree d polynomials. Let $\mathcal{P}$ be the following finite set of functions,

$$\mathcal{P} \coloneqq \{\mathrm{trunc} \circ p \,|\, p \text{ is degree-}d \text{ polynomial with coefficients that are all a multiple of } \varepsilon/n^d\}$$

Degree d polynomials have at most n^d coefficients. Therefore,

$$\log(|\mathcal{P}|) \leq \log\left(\left(\frac{n^d}{\varepsilon}\right)^{n^d}\right) = \mathrm{poly}\left(n^{O(d)}, \log(1/\varepsilon)\right).$$

This means that for the sample size in Lemma 16, Equation (2) holds for all functions in $\mathcal{P}$ with probability at least $1-\delta$. We show that Equation (2) holding for function in $\mathcal{P}$ implies the desired result.

Every degree d truncated polynomial is *pointwise* close to a function in $\mathcal{P}$: Fix any degree d polynomial p. There is some $f \in \mathcal{P}$, for which

$$|\mathrm{trunc}(p(x)) - f(x)| < \varepsilon \quad \text{for all } x \in \{0,1\}^n.$$

This f is easy to specify: It's the truncation of p', where p' is p with all of its coefficients rounded to the nearest ε/n^d. In order to expand Equation (2) to p, we use the following inequality for all $a, \varepsilon \in [0,1]$:

$$|\,(a+\varepsilon)^2 - a^2\,| = |2a\varepsilon| + \varepsilon^2 \leq 3|\varepsilon|.$$

Therefore,

$$\begin{aligned}\left|\mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{S}}\left[(\mathrm{trunc}(p(\boldsymbol{x})) - \boldsymbol{y})^2\right] - \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{S}}\left[f(\boldsymbol{x}) - \boldsymbol{y})^2\right]\right| &\leq 3 \max_{x\in\{0,1\}^n} |\,\mathrm{trunc}(p(x)) - f(x)\,| \\ &\leq 3\,\varepsilon.\end{aligned}$$

Similarly, $\mathbb{E}_{\boldsymbol{x}\sim\{0,1\}^n}\left[(\mathrm{trunc}(p(\boldsymbol{x})) - \mu_{\boldsymbol{T}}(\boldsymbol{x}))^2\right]$ and $\mathbb{E}_{\boldsymbol{x}\sim\{0,1\}^n}\left[(f(\boldsymbol{x}) - \mu_{\boldsymbol{T}}(\boldsymbol{x}))^2\right]$ are within 3ε of one another. Finally, by triangle inequality,

$$\left|\mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{S}}\left[(\mathrm{trunc}(p(\boldsymbol{x})) - \boldsymbol{y})^2\right] - \left(\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\{0,1\}^n}\left[(\mathrm{trunc}(p(\boldsymbol{x})) - \mu_{\boldsymbol{T}}(\boldsymbol{x}))^2\right] + C\right)\right| \leq 7\varepsilon.$$

The desired result holds if we rename $\varepsilon' = \frac{\varepsilon}{7}$. ◀

We are now ready to prove the low-degree algorithm (i.e. L_2 regression) part of Theorem 4.

▶ **Lemma 17** (L_2 regression part of Theorem 4)**.** *Choose any $\varepsilon, \eta, \delta \in (0,1)$, $s \in \mathbb{N}$, and size-s stochastic decision tree $\boldsymbol{T} : \{0,1\}^n \to \{0,1\}$. For a sample of size*

$$m := \mathrm{poly}\left(n^{O(d)}, \frac{1}{\varepsilon}, \log\left(\frac{1}{\delta}\right)\right),$$

let $\mathcal{S}$ be an η-corrupted set of m uniform random samples from $\boldsymbol{T}$. If

$$p^* = \underset{\textit{Degree } \log(s/\varepsilon) \textit{ polynomials } p}{\arg\min} \left(\mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{S}} \left[(p(\boldsymbol{x}) - \boldsymbol{y})^2\right] \right),$$

and $h : \{0,1\}^n \to \{0,1\}$ is the hypothesis $h(x) = \mathrm{round}(\mathrm{trunc}(p^*(x)))$. Then with probability at least $1 - \delta$ over the randomness of the sample,

$$\mathrm{error}_{\boldsymbol{T}}(h) \le \mathrm{opt} + O(\sqrt{\eta}) + \varepsilon.$$

Proof. Let $\mathcal{S}^\circ$ be the original uncorrupted (i.i.d) set of samples, from which $\mathcal{S}$ differs on at most η fraction of points. By Lemma 16, Equation (2) holds, with respect to $\mathcal{S}^\circ$, for all degree $\log(s/\varepsilon)$ polynomials with probability at least $1 - \delta$. We show that if it holds, then $\mathrm{error}_{\boldsymbol{T}}(h) \le \mathrm{opt} + O(\sqrt{\eta}) + \varepsilon$.

Proposition 15 guarantees there exists $p : \{0,1\}^n \to [0,1]$, a degree $\log(s/\varepsilon)$ bounded polynomial, satisfying

$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\{0,1\}^n} \left[(p(\boldsymbol{x}) - \mu_{\boldsymbol{T}}(\boldsymbol{x}))^2\right] \le \varepsilon.$$

Fix C as in Lemma 16. Combining Equation (2) and Fact 13, we have that

$$\mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{S}} \left[(p(\boldsymbol{x}) - \boldsymbol{y})^2\right] \le C + 2\varepsilon + \eta.$$

Since p^* has the minimum L_2 error of all degree $\log(s/\varepsilon)$ polynomials on $\mathcal{S}$,

$$\mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{S}} \left[(p^*(\boldsymbol{x}) - \boldsymbol{y})^2\right] \le C + 2\varepsilon + \eta.$$

Truncating p^* can only decrease its L_2 error. Combining that with a second application of Fact 13,

$$\mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{S}^\circ} \left[(\mathrm{trunc}(p^*(\boldsymbol{x})) - \boldsymbol{y})^2\right] \le C + 2\varepsilon + 2\eta.$$

Then, by Equation (2),

$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\{0,1\}^n} \left[(\mathrm{trunc}(p^*(\boldsymbol{x})) - \mu_{\boldsymbol{T}}(\boldsymbol{x}))^2\right] \le 3\varepsilon + 2\eta. \tag{3}$$

Finally,

$$\begin{aligned}
\mathrm{error}_{\boldsymbol{T}}(h) &\le \mathrm{opt}_{\boldsymbol{T}} + 2 \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\{0,1\}^n}[|\, \mu_{\boldsymbol{T}}(\boldsymbol{x}) - \mathrm{trunc}(p^*(\boldsymbol{x}))\, |] && \text{Fact 6} \\
&\le \mathrm{opt}_{\boldsymbol{T}} + 2\sqrt{\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\{0,1\}^n}[(\mu_{\boldsymbol{T}}(\boldsymbol{x}) - \mathrm{trunc}(p^*(\boldsymbol{x})))^2]} && \text{Jensen's inequality} \\
&\le \mathrm{opt}_{\boldsymbol{T}} + 2\sqrt{3\varepsilon + 2\eta} && \text{Equation (3)} \\
&\le \mathrm{opt}_{\boldsymbol{T}} + O(\sqrt{\varepsilon}) + O(\sqrt{\eta}).
\end{aligned}$$

The desired result then holds by renaming $\varepsilon' = \Omega(\varepsilon^2)$. ◀

5.2 L_1 regression

We will need the following generalization bound:

▶ **Lemma 18** (L_1 error generalization)**.** *Fix any stochastic DT $\boldsymbol{T} : \{0,1\}^n \to \{0,1\}$, degree $d \in \mathbb{N}$, and $\varepsilon, \delta > 0$. For a sample size of*

$$m := \mathrm{poly}\left(n^{O(d)}, \frac{1}{\varepsilon}, \log\left(\frac{1}{\delta}\right)\right),$$

let $\mathcal{S}^\circ$ be a dataset of m i.i.d points of the form $(\boldsymbol{x}, \boldsymbol{T}(\boldsymbol{x}))$. With probability at least $1-\delta$, the following holds for all degree d polynomials $p : \{0,1\}^n \to \mathbb{R}$.

$$\left| \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{S}^\circ}\left[|\mathrm{trunc}(p(\boldsymbol{x})) - \boldsymbol{y}|^2\right] - \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\{0,1\}^n, \boldsymbol{T}}\left[|\mathrm{trunc}(p(\boldsymbol{x})) - \boldsymbol{T}(\boldsymbol{x})|\right] \right| \le \varepsilon. \tag{4}$$

Lemma 18 can be proven using the same discretization argument as Lemma 16. We omit the proof for brevity.

▶ **Lemma 19** (L_1 regression part of Theorem 4)**.** *Choose any $\varepsilon, \eta, \delta \in (0,1)$, $s \in \mathbb{N}$, and size-s stochastic decision tree $\boldsymbol{T} : \{0,1\}^n \to \{0,1\}$. For a sample of size*

$$m := \mathrm{poly}\left(n^{O(d)}, \frac{1}{\varepsilon}, \log\left(\frac{1}{\delta}\right)\right),$$

let $\mathcal{S}$ be an η-corrupted set of m uniform random samples from $\boldsymbol{T}$. If

$$p^* = \mathop{\arg\min}_{\text{Degree } \log(s/\varepsilon) \text{ polynomials } p} \left(\mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{S}}[|\, p(\boldsymbol{x}) - \boldsymbol{y})\, |] \right),$$

and $\boldsymbol{h} : \{0,1\}^n \to \{0,1\}$ is the randomized hypothesis where $\boldsymbol{h}(x)$ is 1 with probability $\mathrm{trunc}(p^*(x))$ *and 0 otherwise. Then with probability at least $1-\delta$ over the randomness of the sample,*

$$\mathrm{error}_{\boldsymbol{T}}(\boldsymbol{h}) \le 2\mathrm{opt} + 2\eta + \varepsilon.$$

Proof. Let $\mathcal{S}^\circ$ be the original uncorrupted (i.i.d) set of samples from which $\mathcal{S}$ differs on at most η fraction of points. By Lemma 18, Equation (4) holds, with respect to $\mathcal{S}^\circ$ for all degree $\log(s/\varepsilon)$ polynomials with probability at least $1-\delta$. We show that if it holds, then $\mathrm{error}_{\boldsymbol{T}}(h) \le 2\mathrm{opt} + 2\eta + \varepsilon$.

Proposition 15 guarantees there exists $p : \{0,1\}^n \to [0,1]$, a degree $\log(s/\varepsilon)$ polynomial, satisfying

$$\mathop{\mathbb{E}}_{\boldsymbol{x}}[|p(\boldsymbol{x}) - \mu_{\boldsymbol{T}}(\boldsymbol{x})|] \le \varepsilon.$$

We first bound the expected error of $\mu_{\boldsymbol{T}}(\boldsymbol{x})$ relative to $\boldsymbol{T}(\boldsymbol{x})$.

$$\begin{aligned}
\mathop{\mathbb{E}}_{\boldsymbol{x}}[|\mu_{\boldsymbol{T}}(\boldsymbol{x}) - \boldsymbol{T}(\boldsymbol{x})|] &= \mathop{\mathbb{E}}_{\boldsymbol{x}}[\Pr[\boldsymbol{T}(\boldsymbol{x}) = 1](1 - \mu_{\boldsymbol{T}}(\boldsymbol{x})) + \Pr[\boldsymbol{T}(\boldsymbol{x}) = 0](\mu_{\boldsymbol{T}}(\boldsymbol{x}))] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{x}}[2\mu_{\boldsymbol{T}}(\boldsymbol{x})(1 - \mu_{\boldsymbol{T}}(\boldsymbol{x}))] \\
&\le 2 \cdot \mathop{\mathbb{E}}_{\boldsymbol{x}}[\min(\mu_{\boldsymbol{T}}(\boldsymbol{x}), 1 - \mu_{\boldsymbol{T}}(\boldsymbol{x}))] \\
&= 2 \cdot \mathrm{opt}_{\boldsymbol{T}}
\end{aligned}$$

By triangle inequality, we have that $\mathbb{E}_{\boldsymbol{x}}[|p(\boldsymbol{x}) - \boldsymbol{T}(\boldsymbol{x})|] \leq 2 \cdot \mathrm{opt}_{\boldsymbol{T}} + \varepsilon$. By Equation (4)

$$\mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y} \sim \mathcal{S}^\circ}[|p(\boldsymbol{x}) - \boldsymbol{y}|] \leq 2 \cdot \mathrm{opt}_{\boldsymbol{T}} + 2\varepsilon.$$

By Fact 13 initialized with $\mathrm{err}(x, y) = |x - y|$,

$$\mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y} \sim \mathcal{S}}[|p(\boldsymbol{x}) - \boldsymbol{y}|] \leq 2 \cdot \mathrm{opt}_{\boldsymbol{T}} + 2\varepsilon + \eta.$$

Since p^* has minimum L_1 error among all degree $\log(s/\varepsilon)$ polynomials,

$$\mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y} \sim \mathcal{S}}[|p^*(\boldsymbol{x}) - \boldsymbol{y}|] \leq 2 \cdot \mathrm{opt}_{\boldsymbol{T}} + 2\varepsilon + \eta.$$

Reapplying Fact 13, combined with the fact that truncating p^* can only decrease its error,

$$\begin{aligned}\mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y} \sim \mathcal{S}^\circ}[|\mathrm{trunc}(p^*(\boldsymbol{x})) - \boldsymbol{y}|] &\leq \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y} \sim \mathcal{S}}[|\mathrm{trunc}(p^*(\boldsymbol{x})) - \boldsymbol{y}|] + \eta \\ &\leq 2 \cdot \mathrm{opt}_{\boldsymbol{T}} + 2\varepsilon + 2\eta.\end{aligned}$$

Applying Equation (4) again.

$$\begin{aligned}\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \{0,1\}^n}[|p(\boldsymbol{x}) - \boldsymbol{T}(\boldsymbol{x})|] &\leq \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y} \sim \mathcal{S}^\circ}[|p(\boldsymbol{x}) - \boldsymbol{T}(\boldsymbol{x})|] + \varepsilon \\ &\leq 2 \cdot \mathrm{opt}_{\boldsymbol{T}} + 3\varepsilon + 2\eta.\end{aligned}$$

Finally, since $\boldsymbol{h}(x)$ returns 1 with probability $\mathrm{trunc}(p^*(x))$, and $\boldsymbol{T}(x)$ is always in $\{0, 1\}$,

$$\begin{aligned}\Pr_{\boldsymbol{x},\boldsymbol{h},\boldsymbol{T}}[\boldsymbol{h}(\boldsymbol{x}) \neq \boldsymbol{T}(\boldsymbol{x})] &= \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \{0,1\}^n}[|p(\boldsymbol{x}) - \boldsymbol{T}(\boldsymbol{x})|] \\ &\leq 2 \cdot \mathrm{opt}_{\boldsymbol{T}} + 3\varepsilon + 2\eta.\end{aligned}$$

The desired result holds with the renaming $\varepsilon' = \frac{\varepsilon}{3}$. ◀

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 arXiv: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 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.

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

8 Nader H Bshouty, Nadav Eiron, and Eyal Kushilevitz. Pac learning with nasty noise. *Theoretical Computer Science*, 288(2):255–275, 2002.

9 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.

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

11 Surbhi Goel, Aravind Gollakota, Zhihan Jin, Sushrut Karmalkar, and Adam Klivans. Superpolynomial lower bounds for learning one-layer neural networks using gradient descent. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119, pages 3587–3596, 2020.

12 Surbhi Goel, Aravind Gollakota, and Adam R. Klivans. Statistical-query lower bounds via functional gradients. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

13 Surbhi Goel and Adam Klivans. Learning neural networks with two nonlinear layers in polynomial time. In *Proceedings of the 32nd Conference on Learning Theory (COLT)*, volume 99, pages 1470–1499, 2019.

14 Surbhi Goel, Adam Klivans, and Raghu Meka. Learning one convolutional layer with overlapping patches. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80, pages 1783–1791, 2018.

15 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.

16 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.

17 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.

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

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

20 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`.

21 Adam Kalai, Adam Klivans, Yishay Mansour, and Rocco A. Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008.

22 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.

23 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.

24 Michael Kearns and Robert Schapire. Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and System Sciences*, 48(3):464–497, 1994.

25 Michael Kearns, Robert Schapire, and Linda Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2/3):115–141, 1994.

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

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

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

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

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

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

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