Decision tree heuristics can fail, even in the smoothed setting

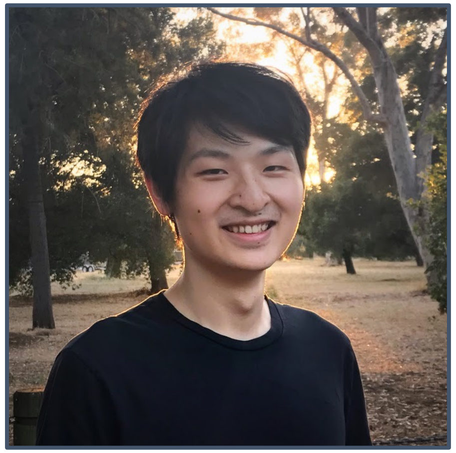
Guy Blanc
Stanford

Jane Lange
MIT

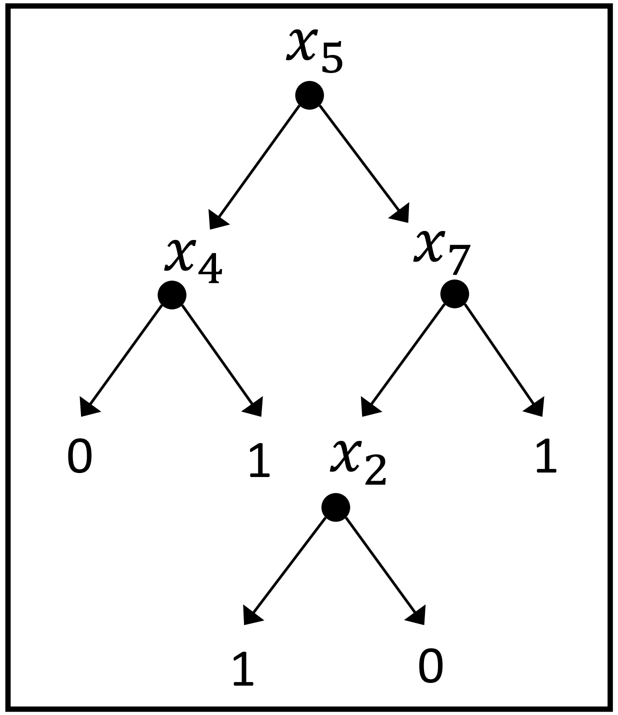
Mingda Qiao
Stanford

Li-Yang Tan
Stanford






Decision Tree Learning



Unknown decision tree
$f: \{0,1\}^n \to \{0,1\}$

Training data $\left(x_i, f(x_i)\right)$

Each x_i drawn from distribution $\mathcal{D}$

Learning Algorithm

Decision tree f'

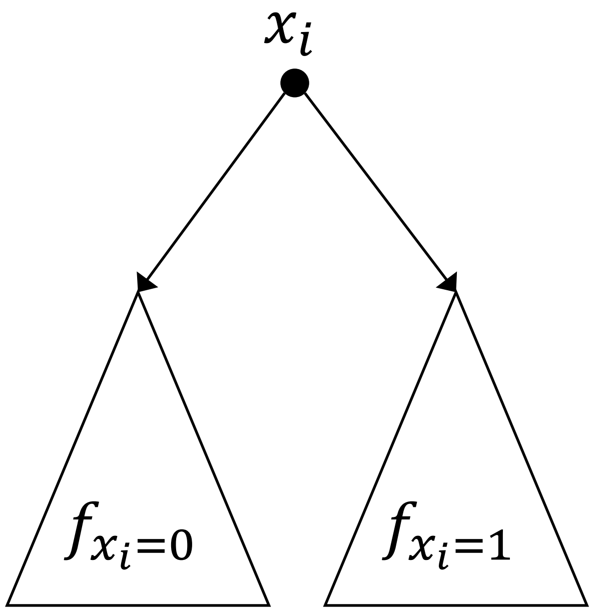
Goal: $f' \approx f$ w.r.t. $\mathcal{D}$, i.e., $\Pr_{x \sim \mathcal{D}}[f'(x) \neq f(x)] \leq \epsilon$

Top-Down Heuristics

Step 1. Find variable x_i that "provides the most information" about $f(x)$

Step 2. (Split) Query variable x_i at the root of the decision tree

Step 3. (Recurse) Build trees for $f_{x_i=0}$ and $f_{x_i=1}$ recursively and use as subtrees



Terminate when depth reaches a specified budget

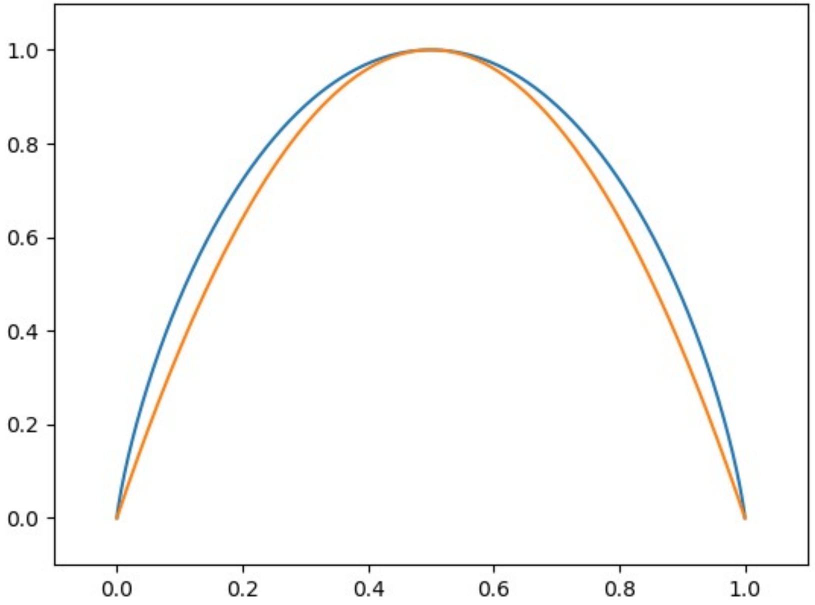
Label each leaf with the majority

Impurity Function

$\mathfrak{G}: [0,1] \to [0,1]$ is an ***impurity function*** if:

- $\mathfrak{G}$ is concave and symmetric w.r.t. $1/2$;
- $\mathfrak{G}(0) = \mathfrak{G}(1) = 0$ and $\mathfrak{G}(1/2) = 1$

$\mathfrak{G}(p) \approx$ amount of uncertainty if f evaluates to 1 on p-fraction of inputs



Binary entropy:
$\mathfrak{G}(p) = -p\log_2 p - (1-p)\log_2(1-p)$

Normalized variance:
$\mathfrak{G}(p) = 4p(1-p)$

Impurity-Based Heuristics

The "purity gain" of querying variable x_i:

$$\mathrm{PurityGain}(f, x_i) \coloneqq \underbrace{\mathfrak{G}(\mathbb{E}[f])}_{\text{uncertainty before querying } x_i} - \underbrace{\left[\Pr[x_i = 0] \cdot \mathfrak{G}\left(\mathbb{E}\left[f_{x_i=0}\right]\right) + \Pr[x_i = 1] \cdot \mathfrak{G}\left(\mathbb{E}\left[f_{x_i=1}\right]\right)\right]}_{\text{remaining uncertainty after seeing } x_i}$$

Top-down algorithm based on impurity function $\mathfrak{G}$:

- Find variable x_i that (roughly) maximizes $\mathrm{PurityGain}(f, x_i)$
- (Split) Query variable x_i at the root of the decision tree
- (Recurse) Build trees for $f_{x_i=0}$ and $f_{x_i=1}$ recursively and use them as subtrees

Impurity-Based Heuristics

Empirical success: ID3, C4.5, CART, …

Hoped-for theoretical guarantee:

For any depth-k DT f and distribution $\mathcal{D}$, these heuristics build a high-accuracy DT of depth k', where k' is not too much larger than k.

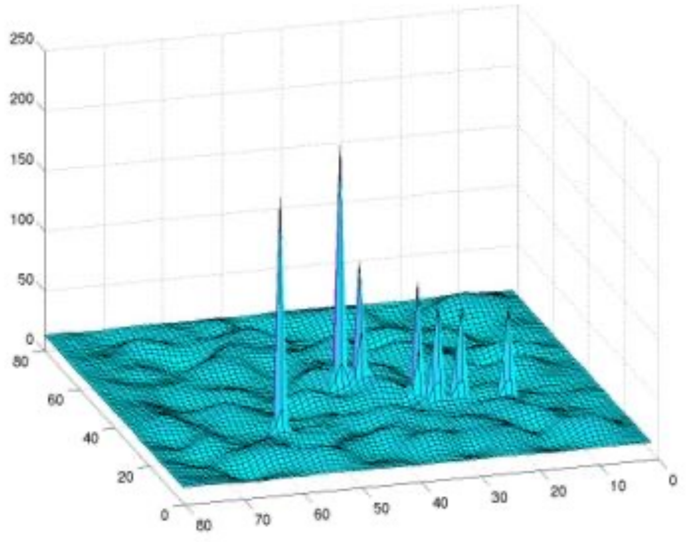
Unfortunately, such guarantee is known to be impossible:

- Even if distribution $\mathcal{D}$ is uniform over $\{0,1\}^n$
- Even if f is a DT of depth $k = 2$
- Need depth $k' = \Omega(n)$ to achieve nontrivial accuracy

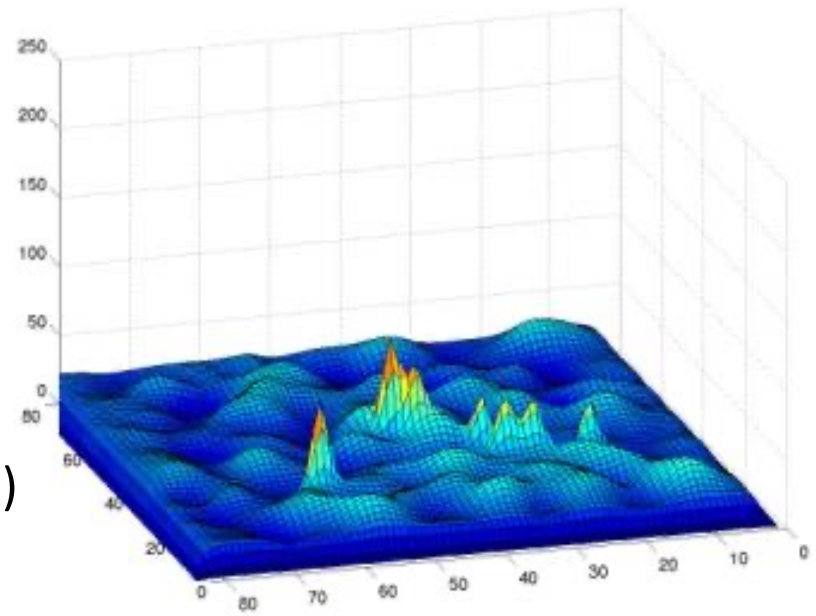
Smoothed Analysis

First developed by [Spielman-Teng’04] for the simplex algorithm

Hard instances are pathological



smoothening
(random perturbation)



Pictures produced by Daniel A. Spielman and Shang-Hua Teng:
https://www.cs.yale.edu/homes/spielman/SmoothedAnalysis/framework.html

Smoothed Learning

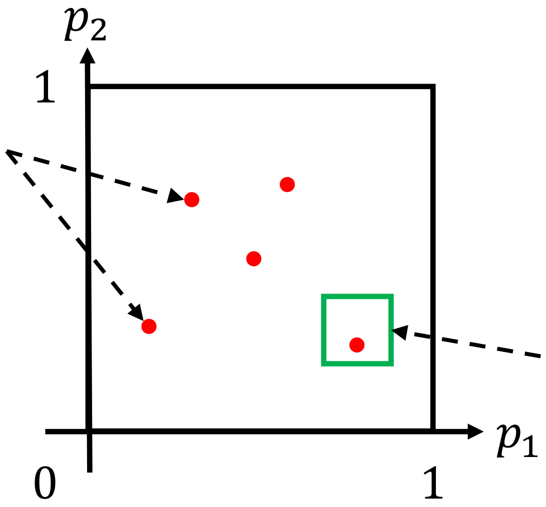
Smoothed learning setting of [Kalai-Samorodnitsky-Teng'09]:

Hard <u>data distributions</u> are pathological

Learning over a **smoothed product distribution** over $\{0,1\}^n$

- Biases $p_1, p_2, \dots, p_n$ are set as $p_i \leftarrow \hat{p}_i + \Delta_i$
- $\hat{p}_1, \dots, \hat{p}_n$ are fixed, whereas $\Delta_1, \dots, \Delta_n \sim \text{Uniform}([-c, c])$

Many hard distributions...



... but they can be rare

Smoothed Learning of Decision Trees

Conjecture of Brutzkus, Daniely and Malach (COLT'20):

Conjecture: For any depth-k decision tree f, any impurity-based heuristic builds a high-accuracy DT of depth $O(k)$ given samples from a smoothed product distribution.

Evidence: Provable guarantee for learning k-juntas

Theorem [BDM20]: For any k-junta f, any impurity-based heuristic builds a high-accuracy DT of depth k given samples from a smoothed product distribution.

Our Results

Counterexample to the conjecture of [BDM20]:

Theorem 1. There is a depth-k decision tree f such that: ***any*** impurity-based heuristic must build a DT of depth $2^{\Omega(k)}$ given samples from ***any*** balanced product distribution $\mathcal{D}$.

$\mathcal{D}$ is balanced if $\Pr_{x \sim \mathcal{D}}[x_i = 1] \in [0.01, 0.99]$ for every coordinate $i \in [n]$

This $2^{\Omega(k)}$ depth is almost as bad as it can get:

- Every depth-k decision tree is a 2^k-junta
- Result of [BDM20] $\Longrightarrow$ heuristics build trees of depth $\leq 2^k$

Our Results

Counterexample to the conjecture of [BDM20]:

Theorem 1. There is a depth-k decision tree f such that: ***any*** impurity-based heuristic must build a DT of depth $2^{\Omega(k)}$ given samples from ***any*** balanced product distribution $\mathcal{D}$.

Theorem 1 is stronger than what is needed:

- The same function f is simultaneously hard for all heuristics
- f is hard over all product distributions and, in particular, over a smoothed product distribution

Our Results

Moreover, the guarantee for juntas does not extend to agnostic setting:

Theorem 2. There is a function f that is ϵ-close to k-juntas such that: ***any*** impurity-based heuristic must build a DT of depth $\epsilon \cdot 2^{\Omega(k)}$ given samples from ***any*** balanced product distribution $\mathcal{D}$.

Corollary: There exists function f s.t.

- f is $2^{-\Omega(k)}$-close to a k-junta
- DT heuristics build trees of depth $2^{\Omega(k)}$ when learning f from a smoothed product distribution

Hard Instance

Recall from [BDM20]'s positive result for juntas:

f depends on N variables $\Longrightarrow$ DT heuristics build trees of depth $\leq N$

To prove the $2^{\Omega(k)}$ lower bound in Theorem 1, we need f to:

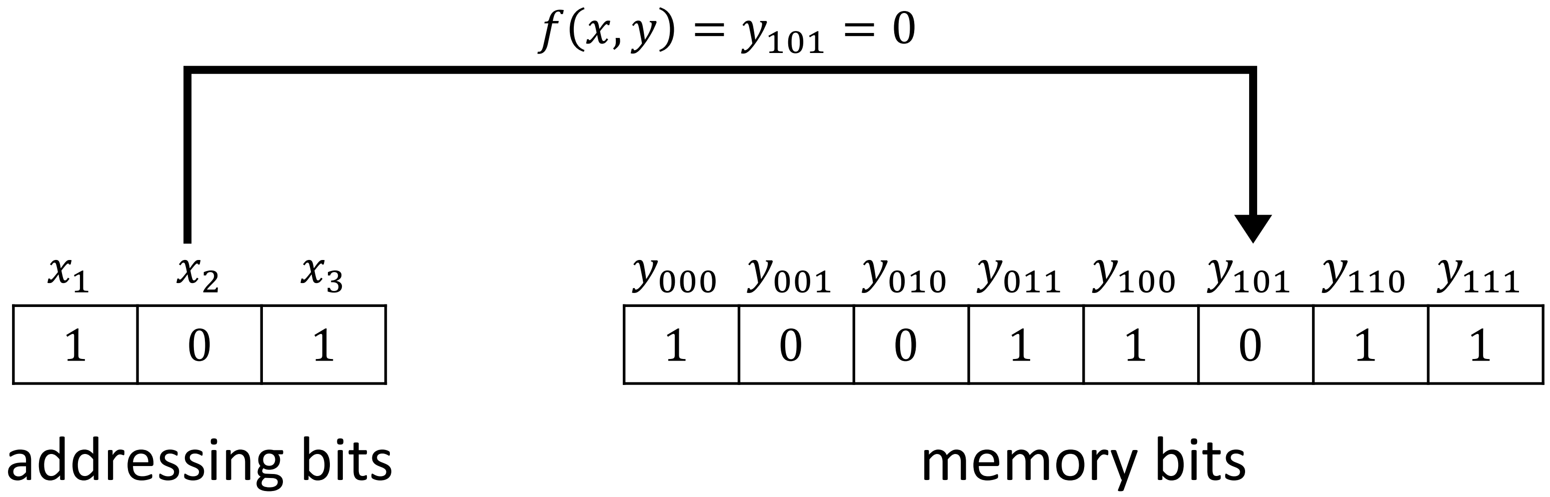
- Be computable by a depth-k decision tree
- Have $2^{\Omega(k)}$ relevant variables

One such extremal example: the "addressing function"

Hard Instance

Addressing function $f\colon \{0,1\}^k \times \{0,1\}^{2^k} \to \{0,1\}$

- k "addressing bits" $x_1, x_2, \dots, x_k$
- 2^k "memory bits" (y_a) indexed by $a \in \{0,1\}^k$
- Define $f(x,y) \coloneqq y_x$



Hard Instance

Addressing function f is computable by a DT of depth $k+1$

- First query the addressing bits $x_1, x_2, \ldots, x_k$
- Query the relevant memory bit y_x, and label the leaf accordingly

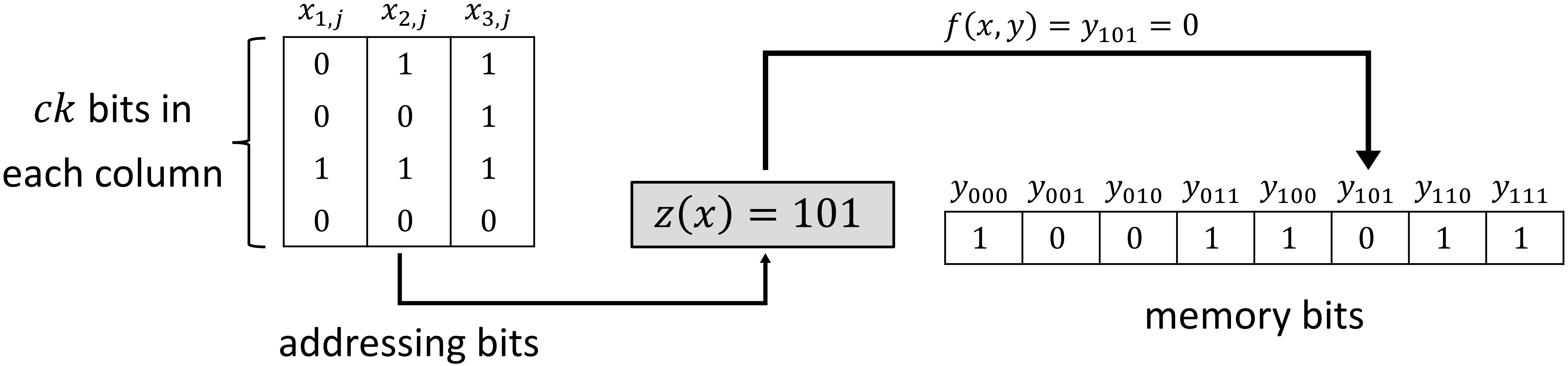
Hoped-for scenario:

- The memory bits have higher purity gains that addressing bits
- DT heuristic builds a tree that queries the variables in the wrong order, i.e., the 2^k memory bits are queried first

Hard Instance

Actual hard instance $f: \{0,1\}^{ck^2} \times \{0,1\}^{2^k} \to \{0,1\}$

- ck^2 "addressing bits" $\left(x_{i,j}\right)$ where $i \in [k], j \in [ck]$
- $f(x,y) \coloneqq y_{z(x)}$ where each $z_i(x)$ is the XOR of $x_{i,j}$ over $j \in [ck]$



Address $z(x)$ is Almost Uniform

Benefit of XOR: when input (x, y) is drawn randomly, the address $z(x)$ is almost uniformly distributed over $\{0,1\}^k$:

Lemma 1. For sufficiently large c and balanced product distribution $\mathcal{D}$,

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

Furthermore, this holds after conditioning on a single bit $x_{i,j}$.

Proof Idea: Each $z_i(x)$ is the XOR of ck independent random bits, each with an expectation in $[0.01,0.99]$

For each $i \in [k]$, $\Pr[z_i(x) = 1]$ is $2^{-\Omega(k)}$-close to $1/2$

Proof Overview

Need to argue:

$$\mathrm{PurityGain}(f, x_{i,j}) \ll \mathrm{PurityGain}(f, y_a)$$

$\Longrightarrow$ Memory bits (y_a) are queried first by impurity-based heuristics

Easy fact: purity gain $\approx$ gap between means

Under mild assumptions on impurity function $\mathfrak{G}$,

$$\mathrm{PurityGain}(f, x_i) = \Theta(1) \cdot \left(\mathbb{E}\left[f_{x_i=0}\right] - \mathbb{E}\left[f_{x_i=1}\right]\right)^2$$

Purity Gain of Memory Bits

Claim: For each memory bit y_a:

$$\left|\mathbb{E}\left[f_{y_a=0}\right] - \mathbb{E}\left[f_{y_a=1}\right]\right| = \Pr[z(x) = a]$$

Intuition: Flipping y_a changes $f(x, y)$ iff the relevant address $z(x)$ is a

By Lemma 1,

$$\Pr[z(x) = a] \geq 2^{-k} - 5^{-k} = \Omega\left(2^{-k}\right)$$

Thus,

$$\mathrm{PurityGain}(f, y_a) \gtrsim \left(\mathbb{E}\left[f_{y_a=0}\right] - \mathbb{E}\left[f_{y_a=1}\right]\right)^2 \gtrsim (1/2)^{2k}$$

Purity Gain of Addressing Bits

Claim: For each addressing bit $x_{i,j}$, let $\mathcal{P}_b$ be the distribution of $z(x)$ conditioning on $x_{i,j} = b$. Then,

$$\left|\mathbb{E}\left[f_{x_{i,j}=0}\right] - \mathbb{E}\left[f_{x_{i,j}=1}\right]\right| \leq \mathrm{TV}(\mathcal{P}_0, \mathcal{P}_1)$$

Intuition: $\mathbb{E}\left[f_{x_{i,j}=b}\right]$ is the expectation of a bounded function over $\mathcal{P}_b$

Lemma 1 $\Longrightarrow$ both $\mathcal{P}_0$ and $\mathcal{P}_1$ are $(2/5)^k$-close to uniform distribution

$$\mathrm{PurityGain}\left(f, x_{i,j}\right) \lesssim \left(\mathbb{E}\left[f_{x_{i,j}=0}\right] - \mathbb{E}\left[f_{x_{i,j}=1}\right]\right)^2 \lesssim (2/5)^{2k}$$

Putting Things Together

For any memory bit y_a and addressing bit $x_{i,j}$,

$$\text{PurityGain}(f, y_a) \gtrsim (1/2)^{2k} \gg (2/5)^{2k} \gtrsim \text{PurityGain}\big(f, x_{i,j}\big)$$

Thus, an impurity-based heuristic always builds a tree that queries an addressing bit at the root

Repeating this argument $\Longrightarrow$ all the 2^k memory bits need to be queried before any addressing bit is queried

Recap & Open Problem

Prior work: Smoothed analysis was conjectured to be a promising route towards theoretical guarantees of DT heuristics

Our negative results: These heuristics may still fail badly in the smoothed setting

Recap & Open Problem

Open question: Stronger guarantees for restricted classes of functions via smoothed analysis?

- E.g., [Blanc-Lange-Tan’20] focused on monotone functions
- The hard instances in this work are highly non-monotone