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 $f: \{0,1\}^n \rightarrow \{0,1\}$

Training data $(x_i, f(x_i))$

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

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

\( \mathcal{G} : [0,1] \rightarrow [0,1] \) is an **impurity function** if:

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

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

**Binary entropy:**

\[
\mathcal{G}(p) = -p \log_2 p - (1 - p) \log_2 (1 - p)
\]

**Normalized variance:**

\[
\mathcal{G}(p) = 4p(1 - p)
\]
Impurity-Based Heuristics

The “purity gain” of querying variable $x_i$:

$$
\text{PurityGain}(f, x_i) := \mathcal{G}(\mathbb{E}[f]) - \left[ \Pr[x_i = 0] \cdot \mathcal{G}(\mathbb{E}[f_{x_i=0}]) + \Pr[x_i = 1] \cdot \mathcal{G}(\mathbb{E}[f_{x_i=1}]) \right]
$$

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

- Find variable $x_i$ that (roughly) maximizes $\text{PurityGain}(f, x_i)$
- (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

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 data distributions are pathological*

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

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

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] \(\implies\) 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 $\epsilon$-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 \( \implies \) 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 \( f: \{0,1\}^k \times \{0,1\}^{2^k} \rightarrow \{0,1\} \)

- \( k \) “addressing bits” \( x_1, x_2, \ldots, x_k \)
- \( 2^k \) “memory bits” \( (y_a) \) indexed by \( a \in \{0,1\}^k \)
- Define \( f(x, y) := y_x \)

\[
f(x, y) = y_{101} = 0
\]
Hard Instance

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

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

**Hoped-for scenario:**

- The memory bits have higher purity gains than 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 \( f: \{0,1\}^{ck^2} \times \{0,1\}^{2^k} \rightarrow \{0,1\} \)

- \( ck^2 \) “addressing bits” \((x_{i,j})\) where \( i \in [k], j \in [ck]\)
- \( f(x, y) := y_{z(x)} \) where each \( z_i(x) \) is the XOR of \( x_{i,j} \) over \( j \in [ck] \)

\[
\begin{array}{ccc}
  x_{1,j} & x_{2,j} & x_{3,j} \\
  0 & 1 & 1 \\
  0 & 0 & 1 \\
  1 & 1 & 1 \\
  0 & 0 & 0 \\
\end{array}
\]

\( ck \) bits in each column

addressing bits

\( z(x) = 101 \)

\( y_{000} \ y_{001} \ y_{010} \ y_{011} \ y_{100} \ y_{101} \ y_{110} \ y_{111} \)

memory bits

\( f(x, y) = y_{101} = 0 \)
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:

\[ \text{PurityGain}(f, x_{i,j}) \ll \text{PurityGain}(f, y_a) \]

\[ \Rightarrow \text{Memory bits } (y_a) \text{ are queried first by impurity-based heuristics} \]

**Easy fact: purity gain \( \approx \) gap between means**

Under mild assumptions on impurity function \( \mathcal{G} \),

\[ \text{PurityGain}(f, x_i) = \Theta(1) \cdot \left( \mathbb{E}[f_{x_i=0}] - \mathbb{E}[f_{x_i=1}] \right)^2 \]
Purity Gain of Memory Bits

Claim: For each memory bit $y_a$:

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

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

By Lemma 1,

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

Thus,

$$\text{PurityGain}(f, y_a) \gtrapprox (\mathbb{E}[f_{y_a=0}] - \mathbb{E}[f_{y_a=1}])^2 \gtrapprox (1/2)^{2k}$$
Purity Gain of Addressing Bits

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

$$\left| \mathbb{E}[f_{x_{i,j}=0}] - \mathbb{E}[f_{x_{i,j}=1}] \right| \leq \text{TV}(P_0, P_1)$$

**Intuition:** $\mathbb{E}[f_{x_{i,j}=b}]$ is the expectation of a bounded function over $P_b$

Lemma 1 $\Rightarrow$ both $P_0$ and $P_1$ are $(2/5)^k$-close to uniform distribution

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

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

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

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

Repeating this argument $\implies$ 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