Testing and Reconstruction via Decision Trees

Li-Yang Tan
Joint work with:

Guy Blanc
Stanford

Jane Lange
MIT

(Slides and preprint available on my webpage)
Decision tree learning

Labeled data

<table>
<thead>
<tr>
<th>Example $x \in {0,1}^n$</th>
<th>Label $y \in {0,1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100010100101</td>
<td>1</td>
</tr>
<tr>
<td>010011001011</td>
<td>0</td>
</tr>
<tr>
<td>111000100101</td>
<td>1</td>
</tr>
<tr>
<td>001010111010</td>
<td>1</td>
</tr>
<tr>
<td>001001100101</td>
<td>0</td>
</tr>
<tr>
<td>111110010000</td>
<td>1</td>
</tr>
<tr>
<td>001100100100</td>
<td>1</td>
</tr>
<tr>
<td>100100111010</td>
<td>0</td>
</tr>
<tr>
<td>100101010110</td>
<td>1</td>
</tr>
</tbody>
</table>

Decision tree representation
Decision trees: simple and effective

- Fast to evaluate
- Easy to understand, easy to explain predictions
- Algorithms widely employed, empirically successful
This talk: Testing and Reconstructing decision trees

- Both tasks easier than learning
  - We draw on recent techniques from learning DTs
  - Our results have new implications for learning DTs

This talk: Surprisingly rich web of connections for DTs
Reconstruction: On-the-fly learning

Given query access to $f$, promised to be close to small decision tree:

Traditional (Proper) Learning

Construct DT hypothesis:

```
\approx \text{opt}_s
```

```
\begin{array}{c}
\text{Size } S(s, \varepsilon) \\
\end{array}
```

Reconstruction: On-the-fly learning

Support queries to DT hypothesis:

```
\begin{array}{c}
x \\
T(x) \\
\end{array}
```

“Property-preserving data reconstruction” Ailon, Chazelle, Seshadhri, Liu, 2004

“On the testability and repair of hereditary hypergraph properties” Austin, Tao, 2008
Main result: Reconstruction algorithm for DTs

Given query access to $f: \{0,1\}^n \to \{0,1\}$, promised to be $\text{opt}_s$-close to size-$s$ DT. We support queries to a DT hypothesis $T$:

Every query answered efficiently:

- $x \xrightarrow{} T(x) \xleftarrow{} f$
- $\text{polylog}(s) \cdot \log n$ queries,
- $\text{polylog}(s) \cdot n \log n$ time
Traditional vs. On-the-fly learning of DTs

Both cases: Given query access to $f$, promised $\text{opt}_s$-close to size-$s$ DT

**Traditional (Proper) Learning**

Construct DT hypothesis:

$$f \approx \text{opt}_s \Downarrow = T$$

Size $S(s, \varepsilon)$

**Fact:** Need

$\Omega(s)$ queries to $f$

$\Omega(s) \cdot n$ time

**Reconstruction: On-the-fly learning**

Support queries to DT hypothesis:

$$x \Downarrow T(x)$$

Our result: Each query to $T$ answered with

$\text{polylog}(s) \cdot \log n$ queries to $f$

$\text{polylog}(s) \cdot n \log n$ time
Corollary: New tester for DTs

Given query access to unknown \( f : \{0,1\}^n \rightarrow \{0,1\} \) and \( s \in \mathbb{N} \),

- If \( f \) \( \varepsilon \)-close to size-\( s \) DT
- If \( f \) \( \Omega(\varepsilon) \)-far from size-\( s^{(\log s)^2} \) DTs

polylog\( (s) \cdot \log n \) queries
polylog\( (s) \cdot n \log n \) time

Adds to a long line of work on testing DTs:

[KR00, DLMORSW07, CGSM11, BBM12, Bsh20]
Comparison with prior work

Our tester:

- polylog(s) \cdot \log n \text{ queries}
- polylog(s) \cdot n \log n \text{ time}

\begin{align*}
\text{Yes} & \quad \text{If } f \epsilon\text{-close to size-s DT} \\
\text{No} & \quad \text{If } f \Omega(\epsilon)\text{-far from size-s}^{(\log s)^2} \text{ DTs}
\end{align*}

Existing testers: [DLMORSW07, CGSM11, Bsh20]

- \tilde{O}(s) \text{ queries}
- s^{O(s)} \cdot n \text{ time}

\begin{align*}
\text{Yes} & \quad \text{If } f \text{ exactly size-s DT} \\
\text{No} & \quad \text{If } f \Omega(\epsilon)\text{-far from size-s DTs}
\end{align*}
Comparison with prior work

**Our tester:**

- If $f$ $\varepsilon$-close to size-$s$ DT:
  - Yes
- If $f$ $\Omega(\varepsilon)$-far from size-$s^{(\log s)^2}$ DTs:
  - No

- $\text{polylog}(s) \cdot \log n$ queries
- $\text{polylog}(s) \cdot n \log n$ time

**Existing testers:** [DLMORSW07, CGSM11, Bsh20]

- If $f$ exactly size-$s$ DT:
  - Yes
- If $f$ $\Omega(\varepsilon)$-far from size-$s$ DTs:
  - No

- $\tilde{O}(s)$ queries
- $s^{O(s)} \cdot n$ time

Strengthen to size-$s$?
Suppose our tester can be improved to:

\[ f \]

- If \( f \) \( \varepsilon \)-close to size-\( s \) DT
- If \( f \) \( \Omega(\varepsilon) \)-far from size-\( \frac{s}{(\log s)^2} \) DTs

Then exists \( \text{poly}(n, s, 1/\varepsilon) \)-time algorithm for proper learning of size-\( s \) DTs.

- Runtime of current best algorithm: \( \text{poly}(n^{\log s}, 1/\varepsilon) \) [EH89]
- “Proper Learning \( \Rightarrow \) Testing” standard and long known [GGR98]
- This gives an example of “Testing \( \Rightarrow \) Proper Learning”
Reconstructors and testers for other properties

DT complexity closely related to many other measures:
- Fourier degree
- Approximate degree
- Randomized query complexity
- Quantum query complexity
- Sensitivity
- ...

Our results for DTs

⇒ Reconstructors and testers for these properties
Example: Reconstructor for Fourier degree

**Fact:** For all $f$, we have $\deg(f) \leq \text{DT depth}(f) \leq \deg(f)^3$.

**DT Reconstructor + Fact** $\uparrow$: Given query access to $f: \{0,1\}^n \rightarrow \{0,1\}$, promised to be $\text{opt}_d$-close to a degree-$d$ polynomial $p: \{0,1\}^n \rightarrow \{0,1\}$. We support queries to a degree-$O(d^7)$ polynomial $q: \{0,1\}^n \rightarrow \{0,1\}$.

Every query answered efficiently:

\[
p \leftarrow \text{opt}_d \rightarrow f \leftarrow \leq O(\text{opt}_d + \varepsilon) \rightarrow q
\]

poly($d$) \cdot \log n \text{ queries, poly($d$) \cdot n \log n \text{ time}
Outline of the rest of this talk

- Overview of our results
- **Key structural result and its proof**
- Our reconstruction algorithm
- Avenues for future work
Suppose $f$ is $\text{opt}_s$-close to a size-$s$ decision tree

No idea what this tree looks like

Very specific structure,
Many enjoyable properties

Size $s$ \((\log s)/\varepsilon)^2\)
Structure and properties of this tree

1. \( f \leq O(\text{opt}_s + \varepsilon) \)

2. Size \( \leq s((\log s)/\varepsilon)^2 \)

3. For all paths \( \pi \), variable queried is the one with the largest "noisy influence" on \( f_\pi \)

4. All leaves \( \ell \) labeled round \( (\mathbb{E}[f_\ell]) \in \{0,1\} \)
No noise sensitivity and noisy influence

**Def.** Noise sensitivity of $f : \{0,1\}^n \to \{0,1\}$ at noise rate $\delta$ is the quantity:

$$\text{NS}_\delta(f) := \mathbb{P} [ f(x) \neq f(y) ]$$

**Def.** The noisy influence of $i \in [n]$ on $f$ is the quantity:

$$\text{NS}_\delta(f) - \mathbb{E} \left[ \text{NS}_\delta(f_{x_i=b}) \right]$$
Real-world heuristics (e.g. ID3, C4.5, CART) split on $x_i$ with largest correlation with $f_\pi$

Noisy influence = higher-order generalization of correlation

(Structure theorem false for correlation.)
Our structural result, restated

Let $f$ be $\text{opt}_s$-close to a size-$s$ DT.

Consider the tree $T$ of size $s^{((\log s)/\varepsilon)^2}$ defined as follows:

For all paths $\pi$, variable queried is the one with the largest "noisy influence" on $f_\pi$.

All leaves $\ell$ labeled $\text{round}(\mathbb{E}[f_\ell]) \in \{0,1\}$

This tree is $O(\text{opt}_s + \varepsilon)$-close to $f$. 
Proof overview

Let $f$ be $\text{opt}_S$-close to a size-$s$ decision tree.

Claim: $\text{dist}(T, f) \leq O(\text{opt}_S + \varepsilon)$

Proof strategy: Define potential function $\Phi : \text{Trees} \rightarrow [0,1]$

Argue that for all $k$, either:

- Already done: $\text{dist}(T_k, f) \leq O(\text{opt}_S + \varepsilon)$
- $\downarrow$ in potential: $\Phi(T_{k+1}) \leq \Phi(T_k) - \varepsilon^2 / (\log s)^3$
The potential function

\[ \Phi: \text{Trees} \rightarrow [0,1], \quad \Phi(T) = \text{Noise sensitivity of } f \text{ with respect to } T \]

\[ \Phi(T) := \mathbb{E} [\text{NS}(f_\ell)] \]

leaves \( \ell \)

Observations:

- \( \Phi(\text{empty tree}) = \text{NS}(f) \leq 1 \)
- \( \Phi(T) \geq 0 \) for all trees \( T \)

"A regularity lemma for low noisy-influences" O’Donnell, Servedio, Tan, Wan, 2010

“A noisy-influence regularity lemma for Boolean functions” Jones, 2016
Our potential function and our splitting criterion

\[ \Phi(T_k) - \Phi(T_{k+1}) \]

\[ = \mathbb{E}_{\ell \sim T_k} \left[ \text{NS}(f_{\ell}) \right] - \mathbb{E}_{\ell^* \sim T_{k+1}} \left[ \text{NS}(f_{\ell^*}) \right] \]

\[ = \mathbb{E}_{\ell \sim T_k} \left[ \text{NoisyInf}_{i_{\ell}}(f_{\ell}) \right] \]

Our splitting criterion greedily drives down our potential function

\[ \text{NS}(f_{\ell}) = \text{NoisyInf}_{i_{\ell}}(f_{\ell}) + \mathbb{E}_{b \sim \{0,1\}} \left[ \text{NS}(f_{\ell}, x_{i_{\ell}} = b) \right] \]
Let $f$ be $\text{opt}_s$-close to a size-$s$ DT. Then:

$$\max_{i \in [n]} \{\text{NoisyInf}_i(f)\} \geq \frac{\text{Var}(\tilde{f}) - \text{opt}_s}{(\log s)^2}$$

$\tilde{f} = \text{smoothened version of } f$

Hides dependence on noise rate

- Variant of the "OSSS inequality" from analysis of Boolean functions

- Applying this lemma:
  - $\text{Var}(\tilde{f}) > \text{opt}_s + \varepsilon \implies \text{RHS} > \varepsilon/(\log s)^2$
  - $\text{Var}(\tilde{f}) \leq \text{opt}_s + \varepsilon \implies f$ is $O(\text{opt}_s + \varepsilon)$-close to constant

"Every decision tree has an influential variable" O'Donnell, Saks, Schramm, Servedio. FOCS 2005

"The influence of variables on Boolean functions", Kahn, Kalai, Linial. FOCS 1988
Outline of this talk

- Overview of our results
- Key structural result and its proof
- Our reconstruction algorithm
- Avenues for future work
Recall our reconstruction algorithm for DTs

Given query access to \( f : \{0,1\}^n \rightarrow \{0,1\} \), promised to be \( \text{opt}_s \)-close to size-\( s \) DT. We support queries to a DT hypothesis \( T \):

\[
\text{Size } s \left( \frac{\log s}{\varepsilon^2} \right) = T
\]

Every query answered efficiently:

\[
\text{polylog}(s) \cdot \log n \text{ queries}
\]
Algorithmic features of our structural result

Observation: Given query access to $f$, can construct $T$ efficiently.

$$\text{NoisyInf}_i(f_\pi) := \text{NS}(f_\pi) - \mathbb{E}_b \left[ \text{NS}(f_\pi, x_i=b) \right]$$

$$\mathbb{P}_{y \sim x} \left[ f_\pi(x) \neq f_\pi(y) \right]$$
Evaluating $T$ on a specific input $x$

Previous slide: Given query access to $f$, can construct $T$ — in full.

In fact, given query access to $f$ and an input $x$, can compute $T(x)$ without constructing $T$ in full.

Build only the path in $T$ that $x$ follows:

$T = \ldots$ 

Key enabling feature of $T$:

*top-down*, inductive definition

Cf. *bottom-up*, backtracking

DT learning algorithms

(e.g. Ehrenfeucht–Haussler 89)

The spirit of Local Computation Algorithms [Rubinfeld et al. 11]
Query complexity of our reconstructor

**Claim:** For any input $x$, can compute $T(x)$ using $\text{polylog}(s) \cdot \log n$ queries to $f$

**Challenge:** There are $n$ variables. Estimating $n$ noisy influences $\Rightarrow \Omega(n)$ query complexity?
Finding the variable with largest noisy influence

**Task:** Given query access to \( f: \{0,1\}^n \rightarrow \{0,1\} \),

As few queries as possible

With high probability,

\[
\text{NoisyInf}_i(f) \geq \text{NoisyInf}_j(f) \quad \text{for all } j \in [n].
\]

**Challenge:** There are \( n \) variables.

Estimating \( n \) noisy influences \( \Rightarrow \Omega(n) \) query complexity?
Query-efficient **simultaneous** estimation of noisy influences

**Lemma:** Given query access to $f : \{0,1\}^n \to \{0,1\}$,

$$f \leftarrow \eta_1, \ldots, \eta_n$$

$(1/\tau)^2 \cdot \log n$ queries

With high probability,

$$\eta_i = \text{NoisyInf}_i(f) \pm \tau \quad \text{for all } i \in [n].$$

**Crux of proof:** 2-query unbiased estimator
Zooming out: Two main components of our proof

**Claim:** For any input $x$, can compute $T(x)$ using $\text{polylog}(s) \cdot \log n$ queries to $f$

**Proof:** Build only the path in $T$ that $x$ follows:

1. **Structural lemma**
   $\Rightarrow$ polylog($s$) depth

2. **Simultaneous estimation algorithm**
   $\Rightarrow$ Identify variable with largest noisy influence on $f_\pi$ with polylog($s$) $\cdot$ log $n$ queries to $f$
Wrapping up: Reconstruction algorithm for DTs

Given query access to $f: \{0,1\}^n \rightarrow \{0,1\}$, promised to be $\text{opt}_s$-close to size-$s$ DT. We support queries to a DT hypothesis $T$:

Every query answered efficiently:

\[\text{polylog}(s) \cdot \log n \text{ queries}\]
Outline of this talk

- Overview of our results
- Key structural result and its proof
- Our reconstruction algorithm
- **Avenues for future work**
Further applications of our structural result?

- Inspired by real-world DT learning heuristics
- This talk: Applications to reconstruction and testing
Learning, Testing, and Reconstruction

- Generic algorithmic tasks, well-studied for many classes
- Surprisingly rich web of connections for the class of DTs

Much more to be understood, quantitatively and qualitatively
Understanding practical DT learning heuristics

- Rigorous guarantees and inherent limitations?
- Theory of splitting criteria?
- Random forests and boosted DTs?

“In summary, 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 successes of programs like C4.5 and CART.”

– Kearns and Mansour

On the boosting ability of top-down decision tree learning algorithms, STOC 1996
Thank you for listening.