Testing and Reconstruction via Decision Trees

Li-Yang Tan

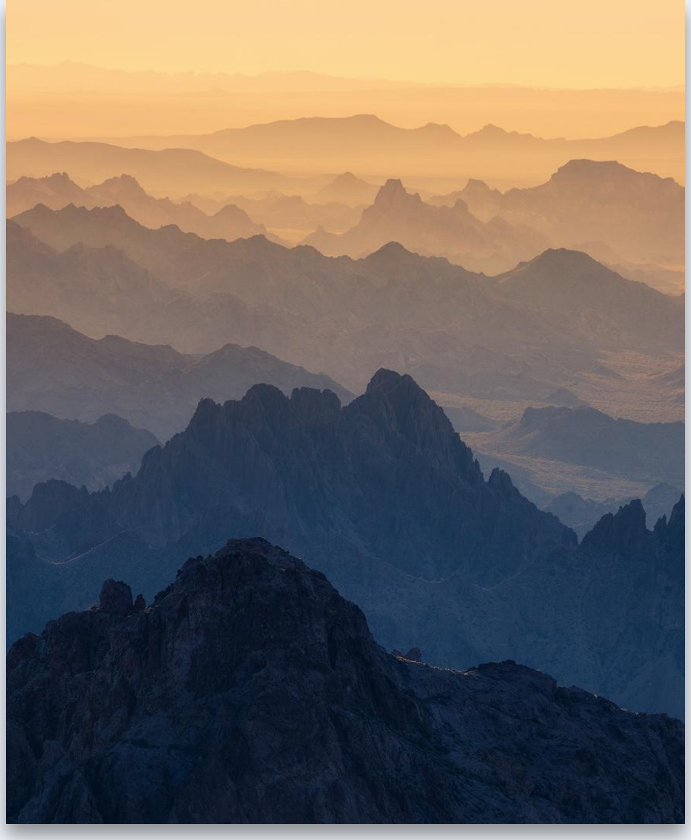
Joint work with:

Guy Blanc

Stanford

Jane Lange

MIT



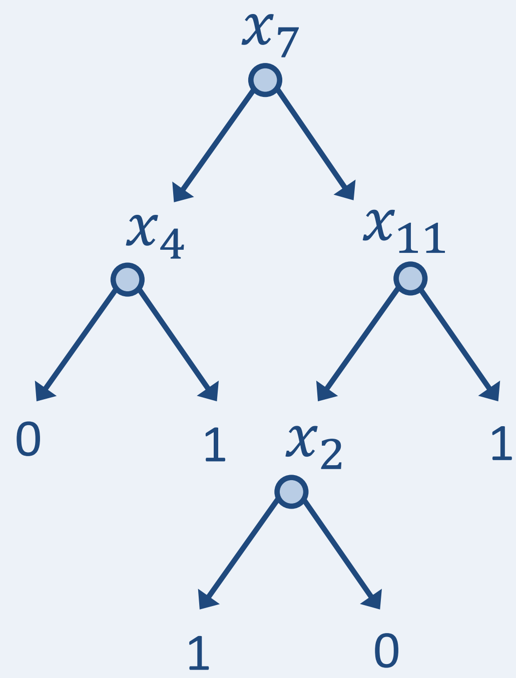
(Slides and preprint available on my webpage)

Decision tree learning

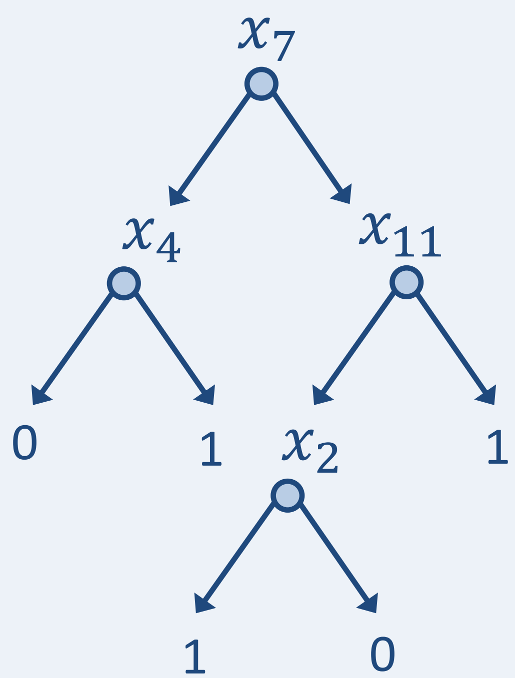
Labeled data

Example $x \in \{0,1\}^n$	Label $y \in \{0,1\}$
1 0 0 0 1 0 1 0 0 1 0 1	1
0 1 0 0 1 1 0 0 1 0 1 1	0
1 1 1 0 0 0 1 0 0 1 0 1	1
0 0 1 0 1 0 1 1 1 0 1 0	1
0 0 1 0 0 1 1 0 0 1 0 1	0
1 1 1 1 1 0 0 1 0 0 0 0	1
0 0 1 1 0 0 1 0 0 1 0 0	1
1 0 0 1 0 0 1 1 1 0 1 0	0
1 0 0 1 0 1 0 1 0 1 1 0	1

Decision tree representation



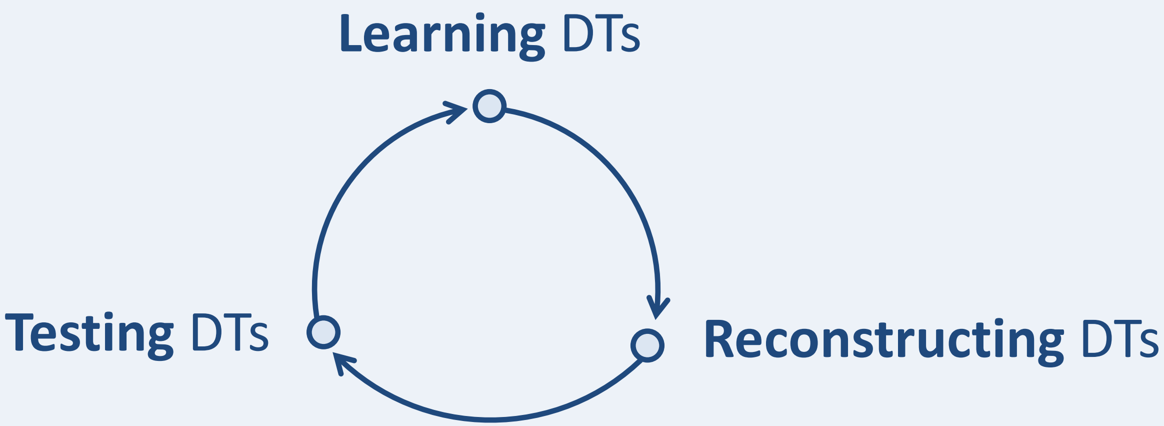
Decision trees: simple and effective



- Fast to evaluate
- Easy to understand,
  easy to explain predictions
- Algorithms widely employed,
  empirically successful

This talk:
Testing and Reconstructing decision trees

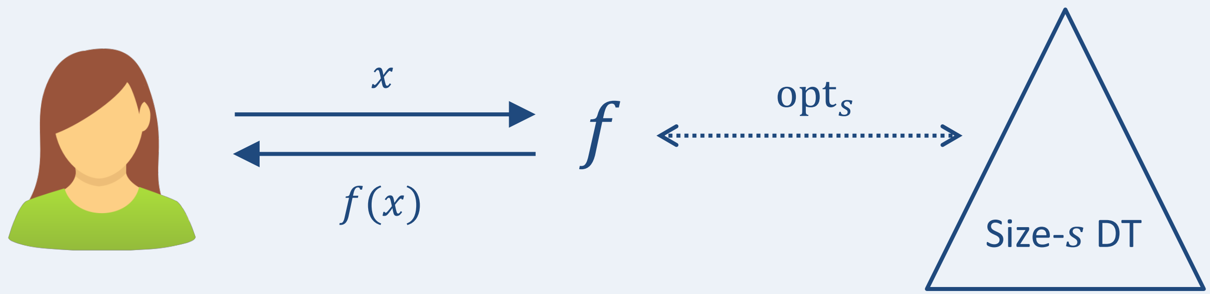
- Both tasks easier than learning
  - We draw on recent techniques from learning DTs
  - Our results have new implications for learning DTs



This talk: Surprisingly rich web of connections for DTs

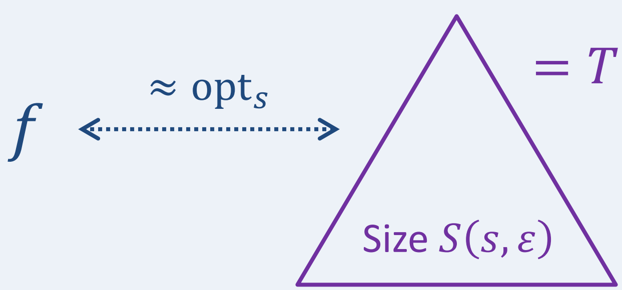
Reconstruction: On-the-fly learning

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



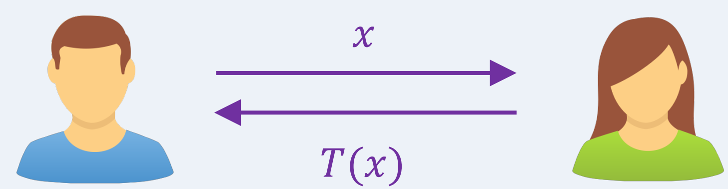
Traditional (Proper) Learning

Construct DT hypothesis:



Reconstruction: On-the-fly learning

Support queries to DT hypothesis:

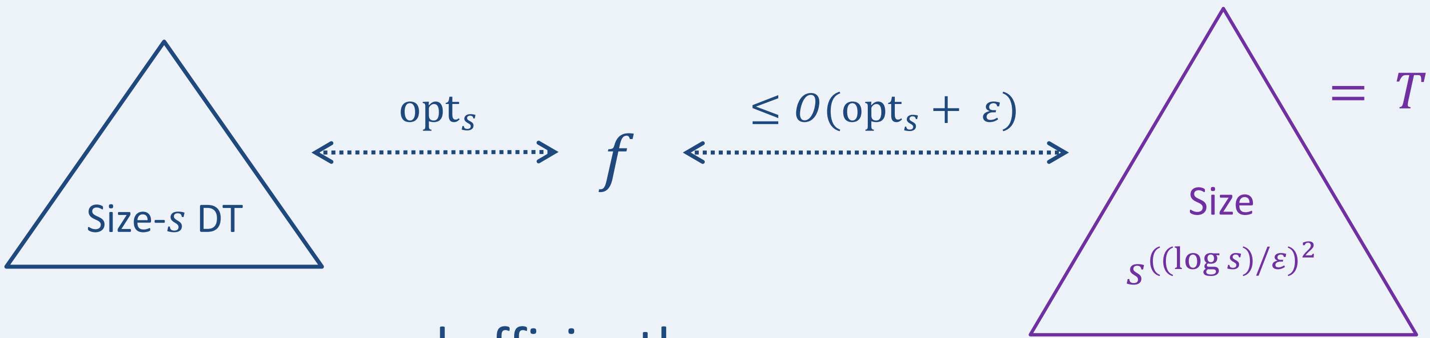


"Property-preserving data reconstruction" Ailon, Chazelle, Seshadhri, Liu, 2004

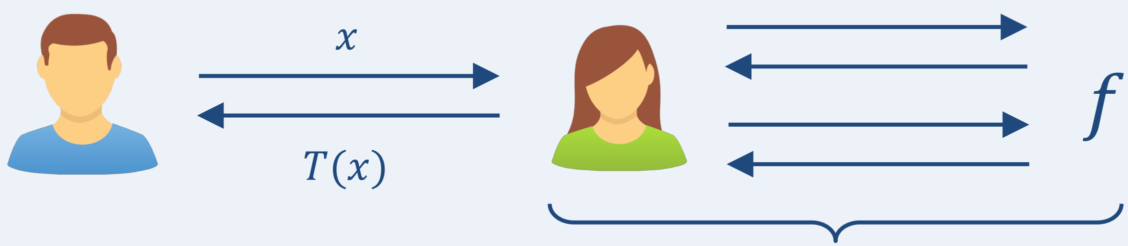
"On the testability and repair of hereditary hypergraph properties" Austin, Tao, 2008

Main result: Reconstruction algorithm for DTs

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



Every query answered efficiently:



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

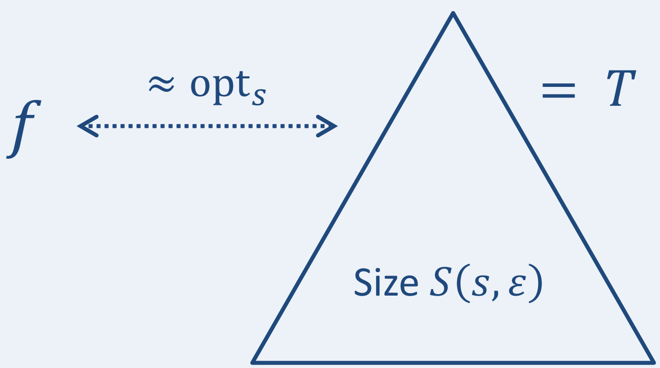
$\text{polylog}(s) \cdot n \log n$ time

Traditional vs. On-the-fly learning of DTs

Both cases: Given query access to f, promised opt_s-close to size-s DT

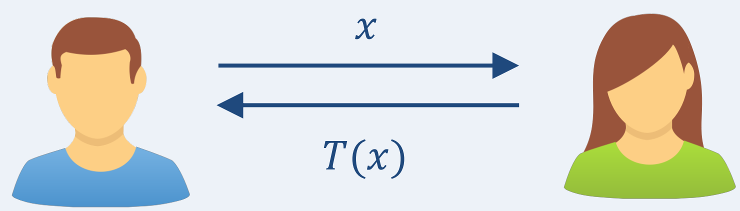
Traditional (Proper) Learning

Construct DT hypothesis:



Reconstruction: On-the-fly learning

Support queries to DT hypothesis:



Fact: Need

$\Omega(s)$ queries to f

$\Omega(s) \cdot n$ time

Our result: Each query to T answered with

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

$\mathrm{polylog}(s) \cdot n \log n$ time

Corollary: New tester for DTs

Given query access to unknown $f\colon\{0,1\}^n \to \{0,1\}$ and $s \in \mathbb{N}$,

$\underbrace{\hspace{6cm}}$

$\mathrm{polylog}(s) \cdot \log n$ queries

$\mathrm{polylog}(s) \cdot n \log n$ time

Yes — If f ε-close to size-s DT

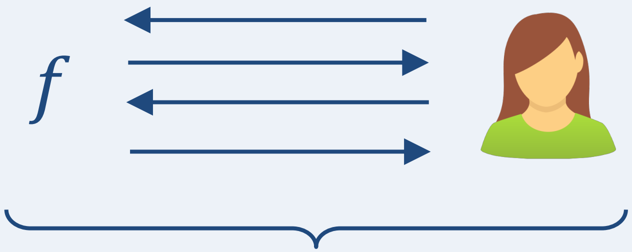
No — If f $\Omega(\varepsilon)$-far from size-$s^{(\log s)^2}$ DTs

Adds to a long line of work on testing DTs:

[KR00, DLMORSW07, CGSM11, BBM12, Bsh20]

Comparison with prior work

Our tester:



$\mathrm{polylog}(s) \cdot \log n$ queries

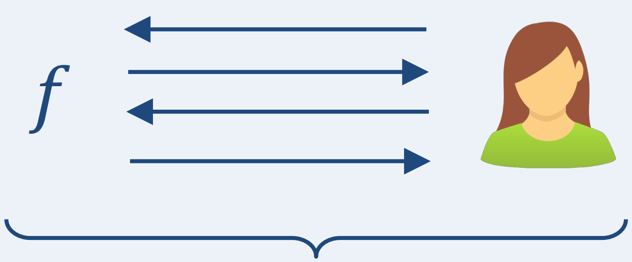
$\mathrm{polylog}(s) \cdot n \log n$ time

Yes — If f ε-close to size-s DT

No — If f $\Omega(\varepsilon)$-far from size-$s^{(\log s)^2}$ DTs

---

Existing testers: [DLMORSW07, CGSM11, Bsh20]



$\tilde{O}(s)$ queries

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

Yes — If f **exactly** size-s DT

No — If f $\Omega(\varepsilon)$-far from **size-s** DTs

Comparison with prior work

Our tester:

Yes — If f ε-close to size-s DT

No — If f $\Omega(\varepsilon)$-far from size-$s^{(\log s)^2}$ DTs

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

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

Strengthen to size-s?

Existing testers: [DLMORSW07, CGSM11, Bsh20]

Yes — If f **exactly** size-s DT

No — If f $\Omega(\varepsilon)$-far from **size-s** DTs

$\tilde{O}(s)$ queries

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

Improved tester $\Longrightarrow$ New learning algorithm

Suppose our **tester** can be improved to:

Yes — If f ε-close to size-s DT

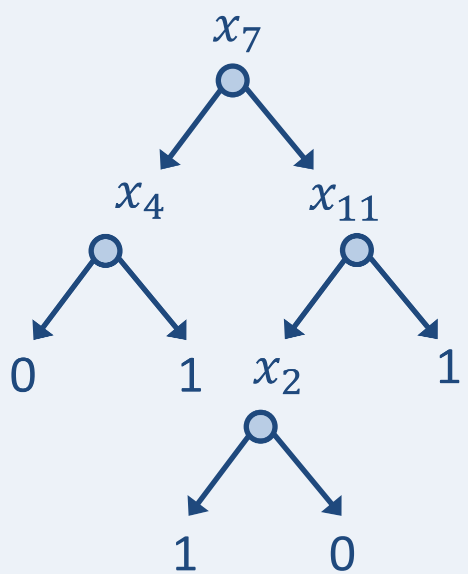
No — If f $\Omega(\varepsilon)$-far from ~~size-$s^{(\log s)^2}$~~ size-s DTs

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

$\implies$ Reconstructors and testers for these properties

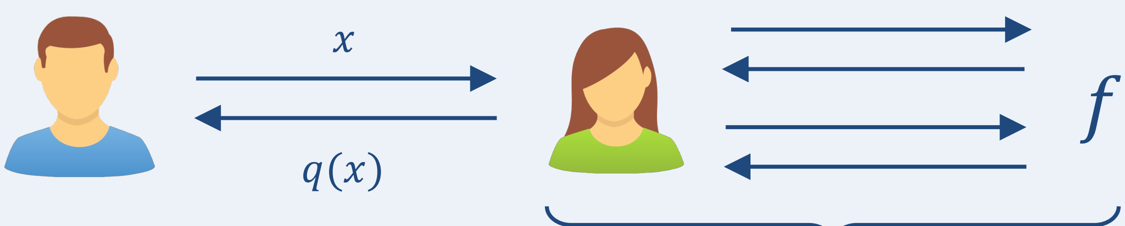
Example: Reconstructor for Fourier degree

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

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

$$p \overset{\mathrm{opt}_d}{\longleftrightarrow} f \overset{\leq O(\mathrm{opt}_d + \varepsilon)}{\longleftrightarrow} q$$

Every query answered efficiently:



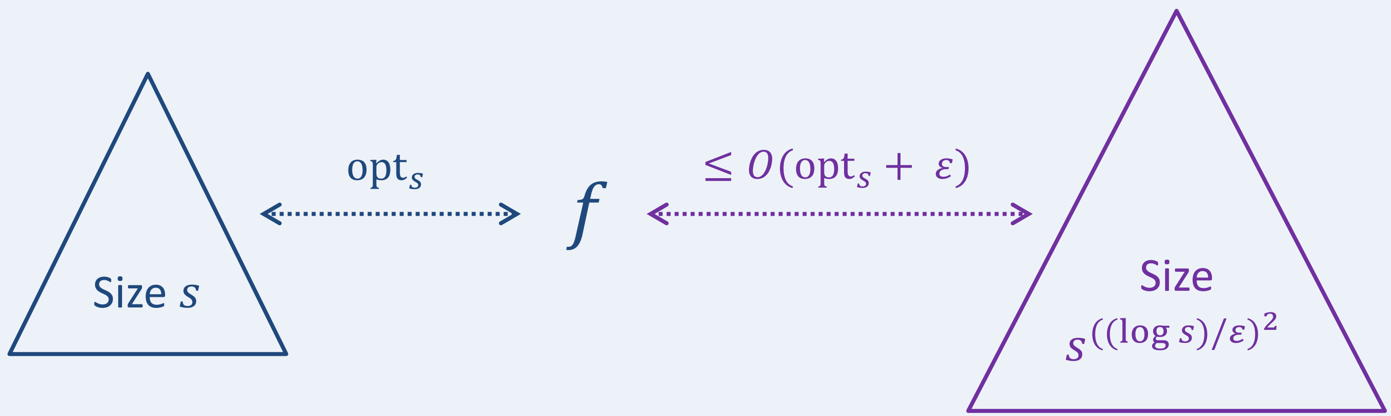
$\mathrm{poly}(d) \cdot \log n$ queries,
$\mathrm{poly}(d) \cdot n \log n$ time

Outline of the rest of this talk

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

Key structural result in a nutshell

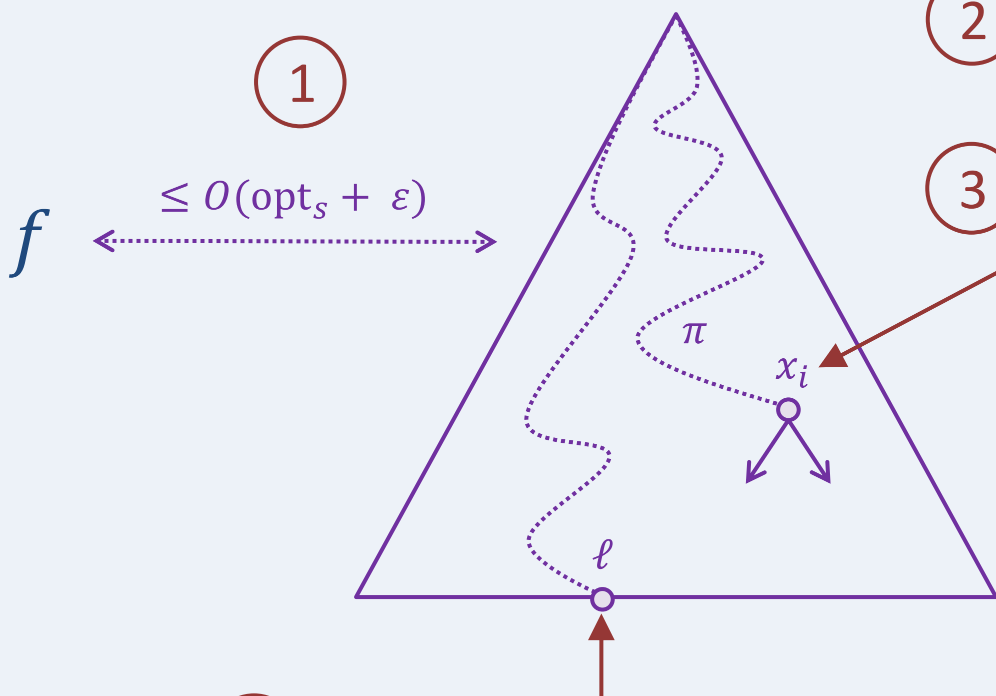
Suppose f is opt_s-close to a size-s decision tree



No idea what this tree looks like

Very specific structure,
Many enjoyable properties

Structure and properties of this tree



2 Size $\leq s^{((\log s)/\varepsilon)^2}$

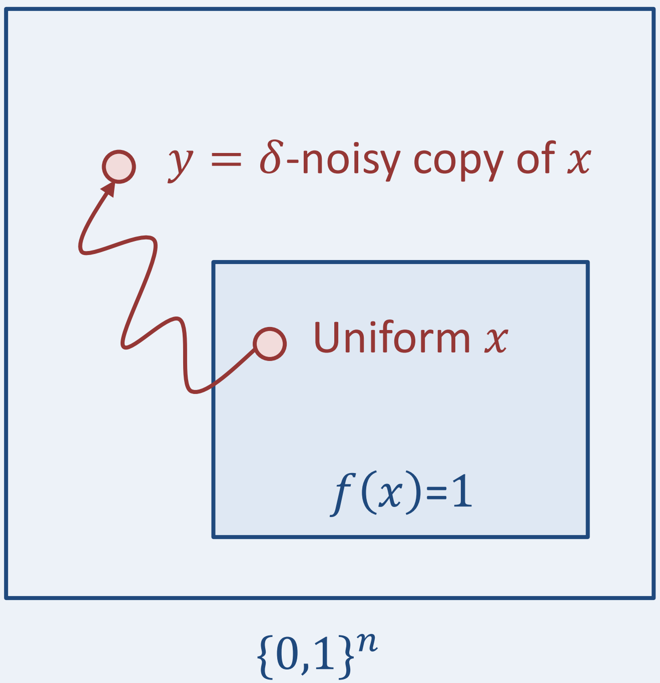
3 For all paths π, variable queried is the one with the largest “noisy influence” on f_π

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

Noise sensitivity and noisy influence

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

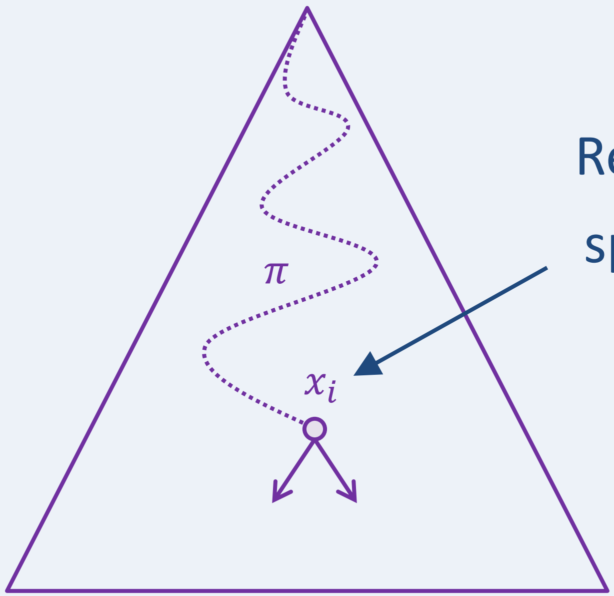
$$\mathrm{NS}_\delta(f) := \mathop{\mathbb{P}}_{y \underset{\delta}{\sim} x}[f(x) \neq f(y)]$$



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

$$\mathrm{NS}_\delta(f) - \mathop{\mathbb{E}}_{b \sim \{0,1\}}\left[\mathrm{NS}_\delta\left(f_{x_i=b}\right)\right]$$

Context: Splitting criteria of DT learning heuristics

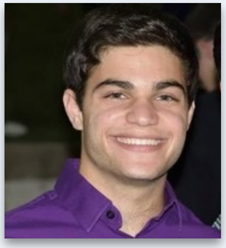


Real-world heuristics (e.g. ID3, C4.5, CART)

split on x_i with largest **correlation** with f_π

Noisy influence = higher-order generalization of correlation

(Structure theorem false for correlation.)

Blanc, Lange, Tan. ICML 2020

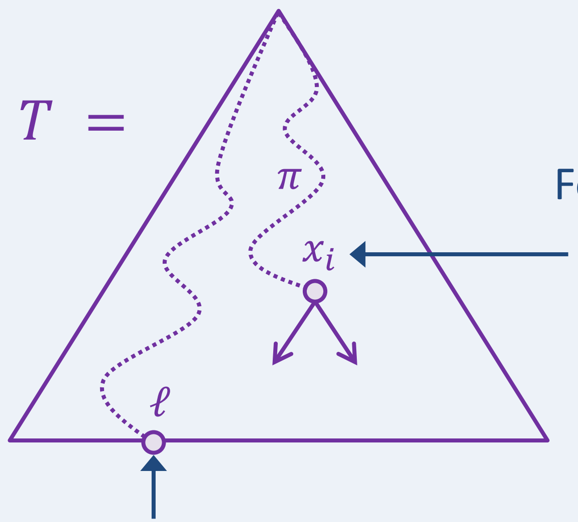
Blanc, Gupta, Lange, Tan. NeurIPS 2020

Our structural result, restated

Let f be opt_s-close to a size-s DT.

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

$T =$



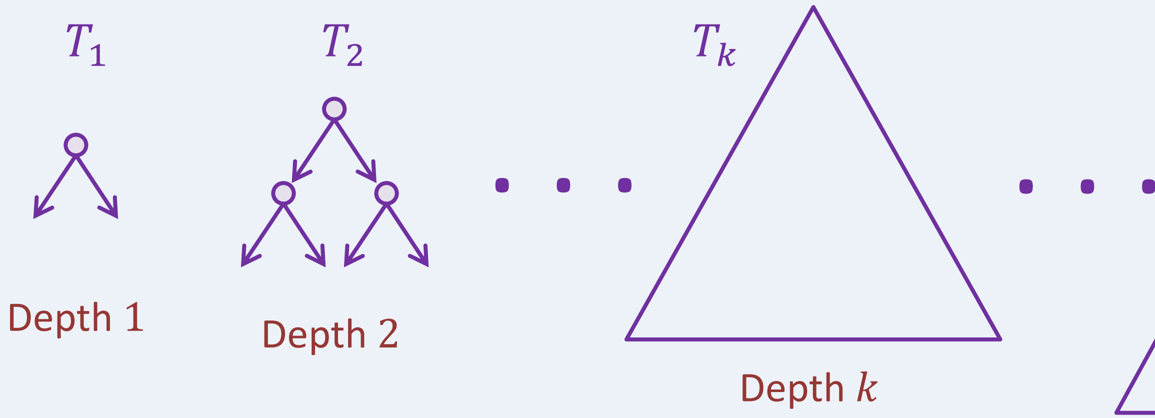
For all paths π, variable queried is the one with the largest "noisy influence" on f_π

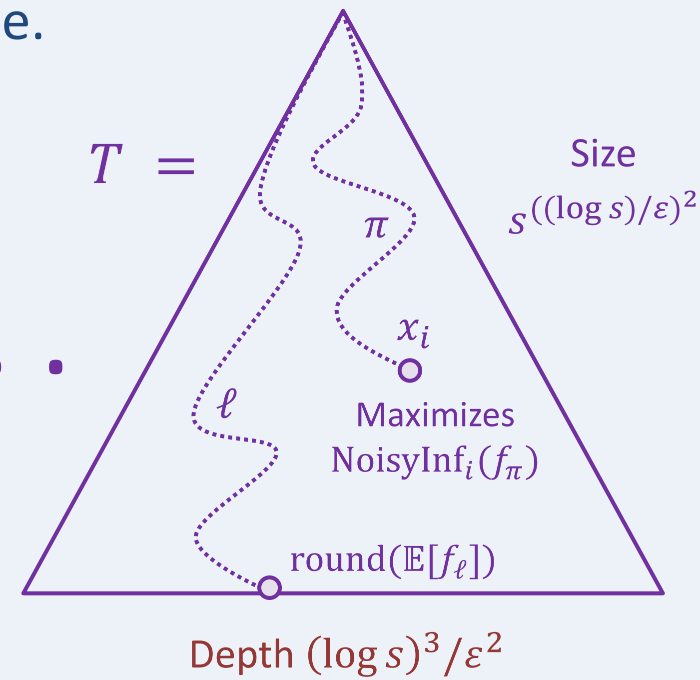
All leaves ℓ labeled $\mathrm{round}(\mathbb{E}[f_\ell]) \in \{0,1\}$

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

Proof overview

Let f be 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 Φ : Trees $\to [0,1]$

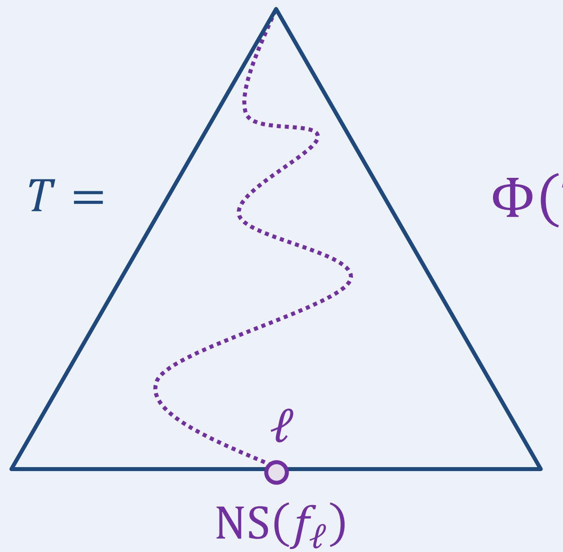
Argue that for all k, either:

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

The potential function

Φ: Trees $\rightarrow [0,1]$, $\Phi(T)$ = Noise sensitivity of f **with respect to T**

$T =$



$$\Phi(T) \coloneqq \mathop{\mathbb{E}}_{\text{leaves } \ell}[\mathrm{NS}(f_\ell)]$$

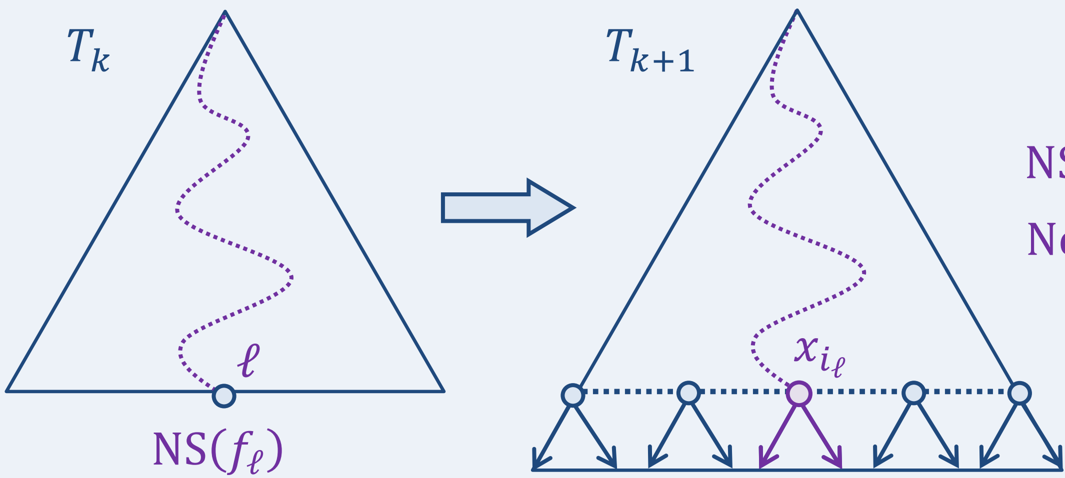
Observations:

- $\Phi(\text{empty tree}) = \mathrm{NS}(f) \leq 1$
- $\Phi(T) \geq 0$ 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



$$\mathrm{NS}(f_\ell) = \mathrm{NoisyInf}_{i_\ell}(f_\ell) + \mathop{\mathbb{E}}_{b\sim\{0,1\}}\left[\mathrm{NS}\left(f_{\ell,\, x_{i_\ell}=b}\right)\right]$$

$$\left.\begin{aligned} &\Phi(T_k) - \Phi(T_{k+1}) \\ &= \mathop{\mathbb{E}}_{\ell\sim T_k}[\mathrm{NS}(f_\ell)] - \mathop{\mathbb{E}}_{\ell^*\sim T_{k+1}}[\mathrm{NS}(f_{\ell^*})] \\ &= \mathop{\mathbb{E}}_{\ell\sim T_k}[\mathrm{NoisyInf}_{i_\ell}(f_\ell)] \end{aligned}\right\}$$

Our **splitting criterion** greedily drives down our **potential function**

Key lemma: Lower bound on noisy influence

Let f be opt_s-close to a size-s DT. Then:

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

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

Hides dependence on noise rate

- Variant of the "OSSS inequality" from analysis of Boolean functions
- Applying this lemma:
  - $\mathrm{Var}(\tilde{f}) > \mathrm{opt}_s + \varepsilon \implies$ RHS $> \varepsilon/(\log s)^2$
  - $\mathrm{Var}(\tilde{f}) \le \mathrm{opt}_s + \varepsilon \implies f$ is $O(\mathrm{opt}_s + \varepsilon)$-close to constant

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

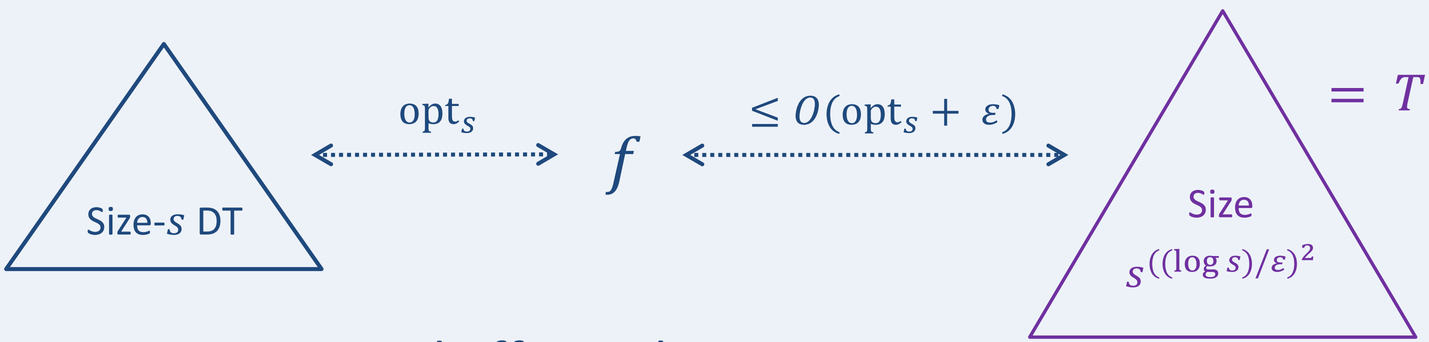
"The influence of variables on Boolean functions", Kahn, Kalai, Linial. FOCS 1988

Outline of this talk

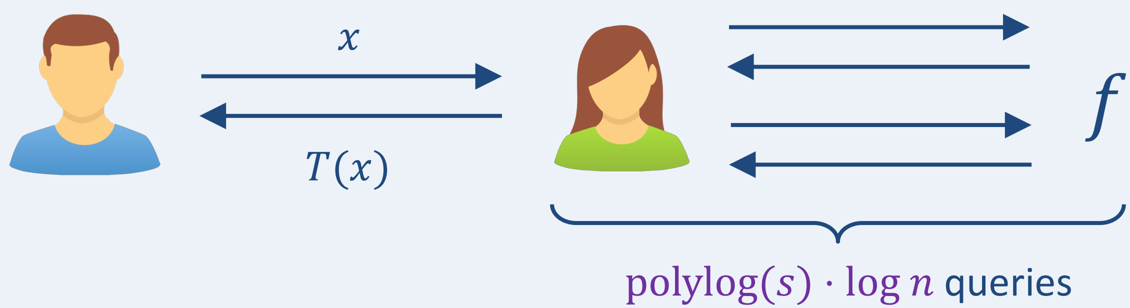
- Overview of our results
- Key structural result and its proof
- **Our reconstruction algorithm**
- Avenues for future work

Recall our reconstruction algorithm for DTs

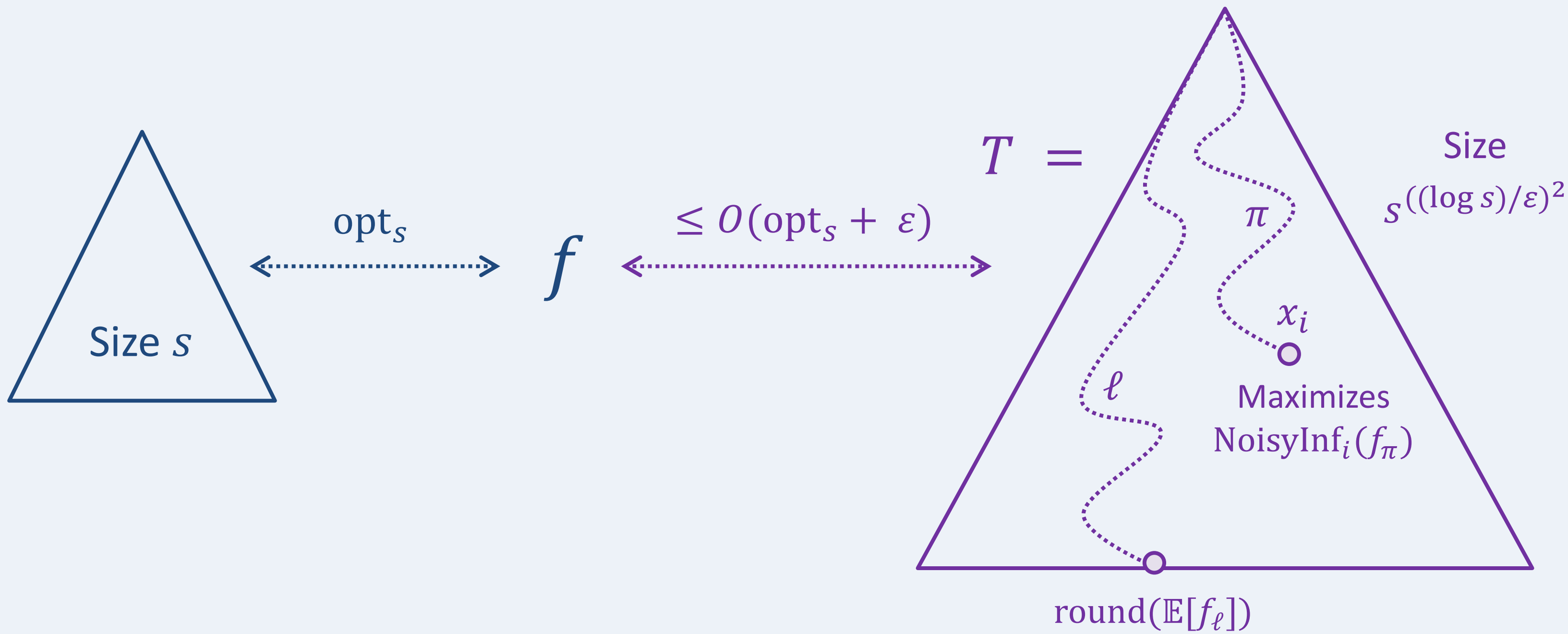
Given query access to $f: \{0,1\}^n \to \{0,1\}$, promised to be opt_s-close to size-s DT. We support queries to a DT hypothesis T:



Every query answered efficiently:



Algorithmic features of our structural result



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

$$\mathrm{NoisyInf}_i(f_\pi) \coloneqq \underbrace{\mathrm{NS}(f_\pi)}_{\underset{y \sim_\delta x}{\mathbb{P}}\left[f_\pi(x) \neq f_\pi(y)\right]} - \mathop{\mathbb{E}}_{b}\left[\mathrm{NS}\left(f_{\pi,\, x_i = b}\right)\right]$$

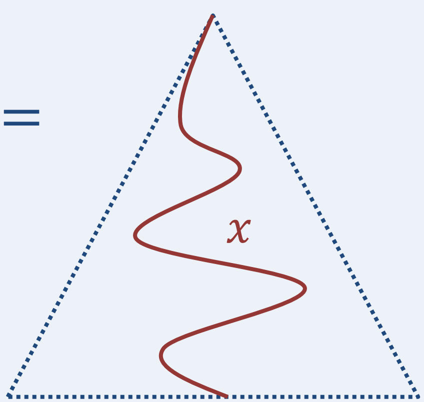
Evaluating T on a specific input x

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

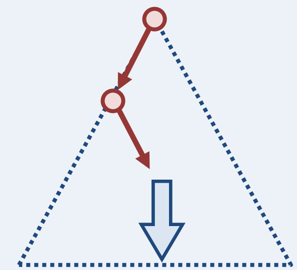
In fact, given query access to f and an input x, can compute $T(x)$ without constructing T in full.

Build only the path in T that x follows:

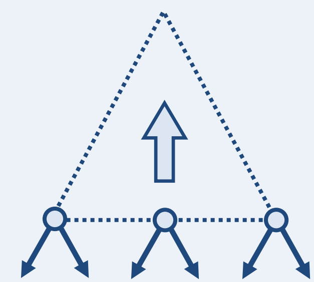
$T =$



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]

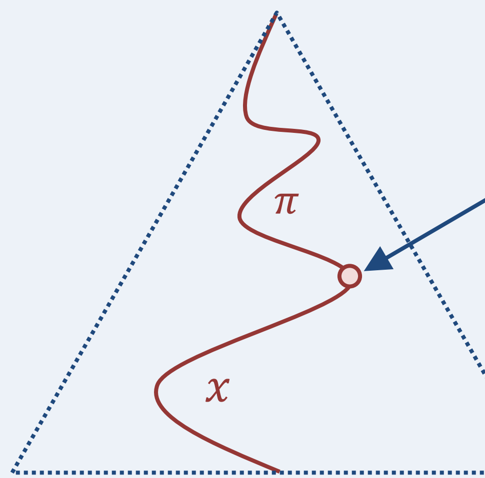
Query complexity of our reconstructor

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

Structural Lemma

$\Rightarrow$ $\mathrm{polylog}(s)$ depth

✓



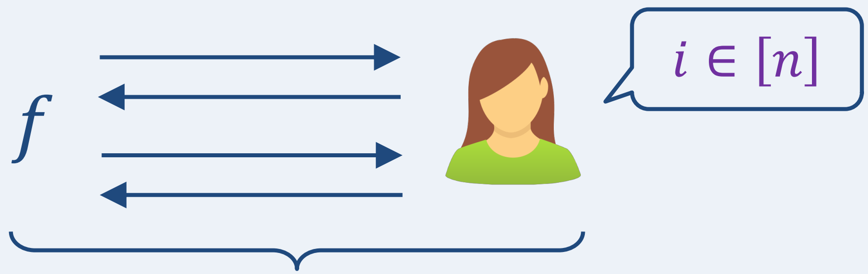
Query complexity of figuring out which variable has the largest noisy influence on f_π?

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



As few queries as possible

With high probability,

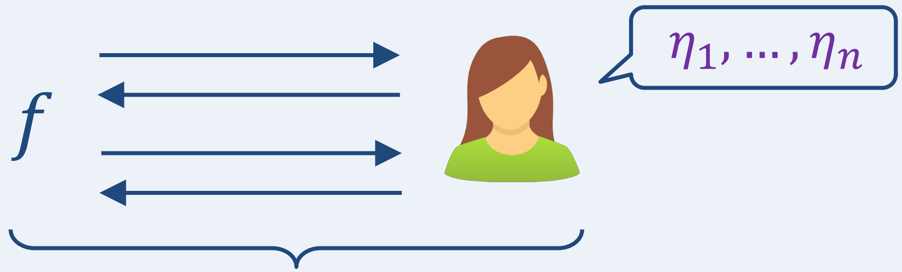
$$\mathrm{NoisyInf}_i(f) \geq \mathrm{NoisyInf}_j(f) \quad \text{for all } j \in [n].$$

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



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

With high probability,

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

Crux of proof: 2-query unbiased estimator

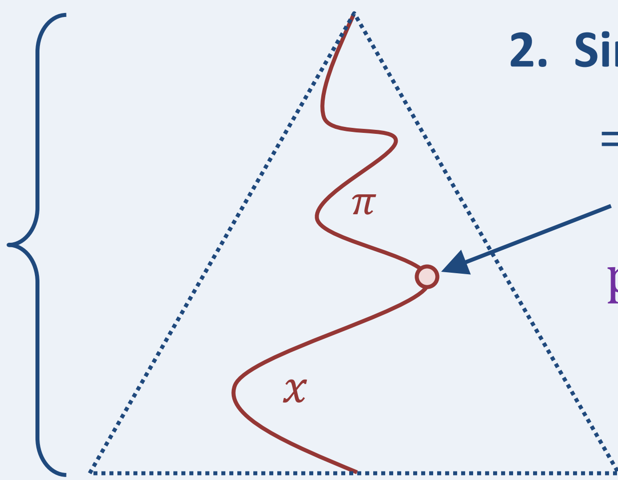
Zooming out: Two main components of our proof

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

$\Rightarrow$ $\text{polylog}(s)$ depth

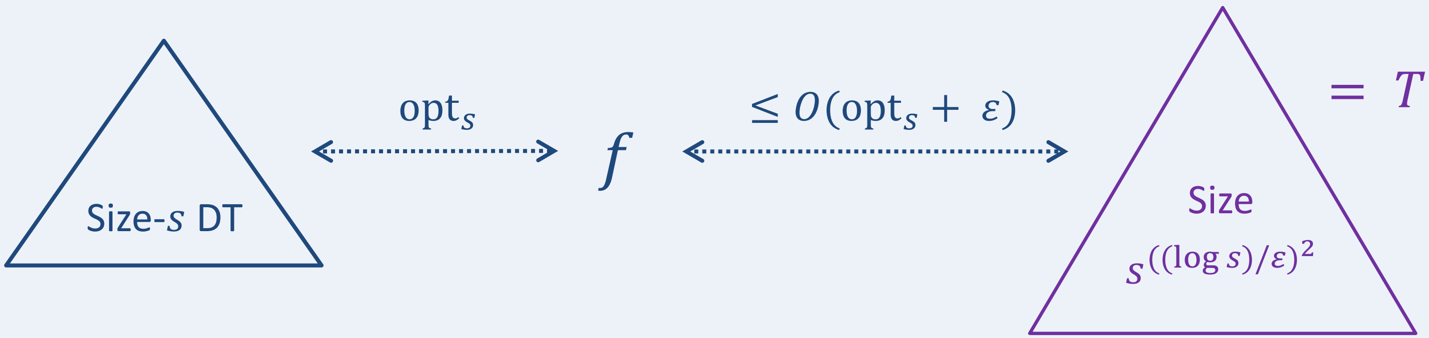


2. Simultaneous estimation algorithm

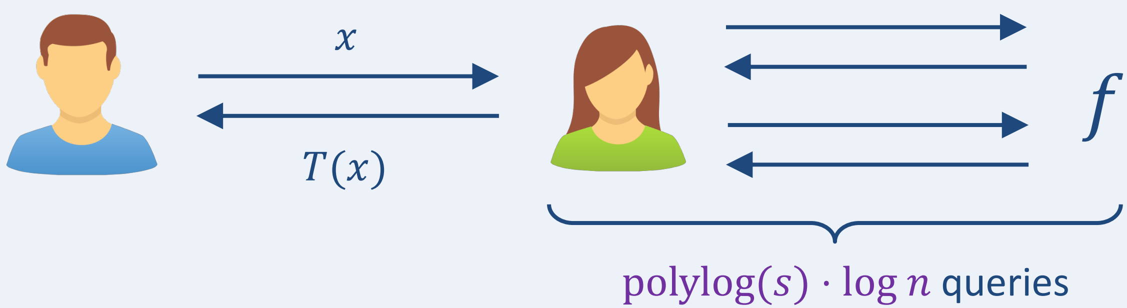
$\Rightarrow$ Identify variable with largest noisy influence on f_π with $\text{polylog}(s) \cdot \log n$ queries to f

Wrapping up: Reconstruction algorithm for DTs

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



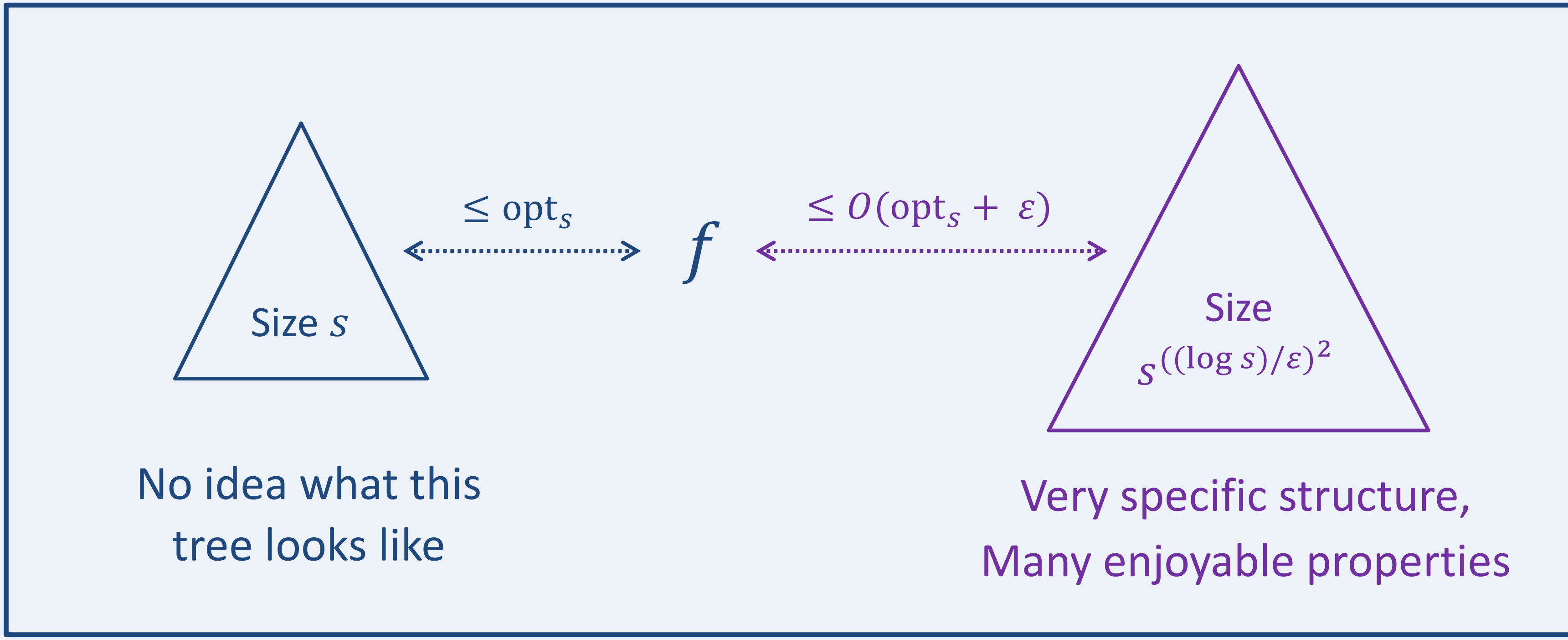
Every query answered efficiently:



Outline of this talk

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

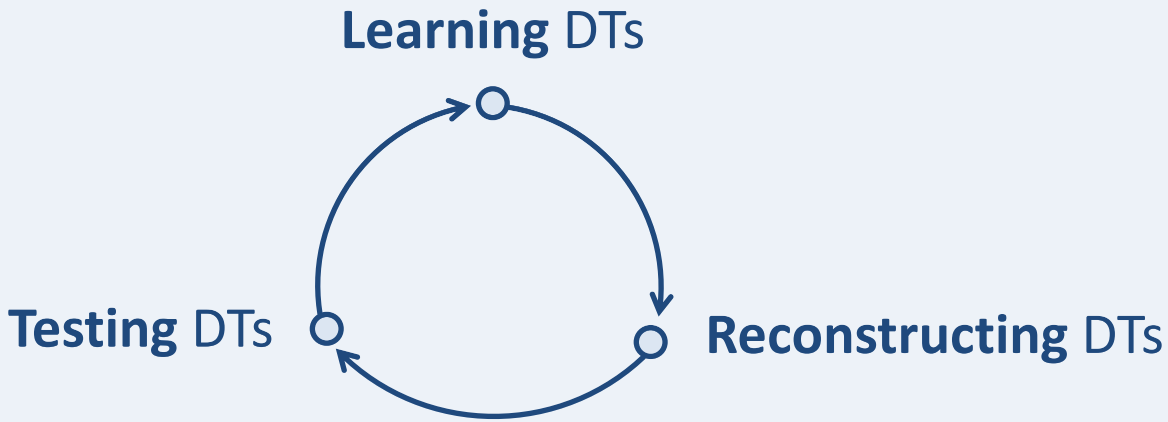
Further applications of our structural result?



- 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.