Properly Learning Decision Trees in Almost-Polynomial Time

Li-Yang Tan

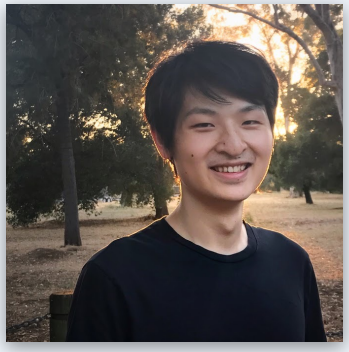
Joint work with:

Guy Blanc

Stanford

Jane Lange

MIT

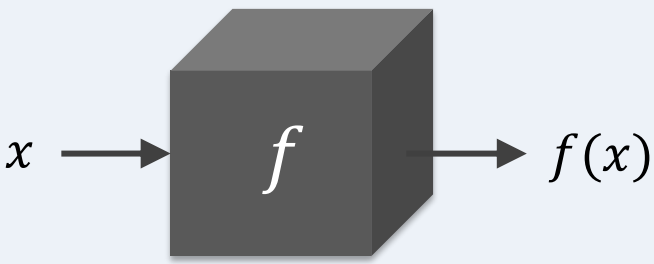


Mingda Qiao

Stanford

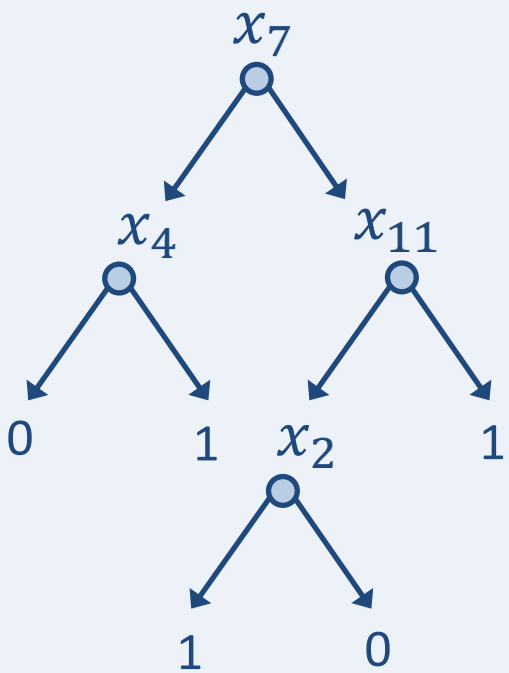
MIT Theory of Computation Colloquium, October 2021

This talk:
Turning black boxes into decision trees



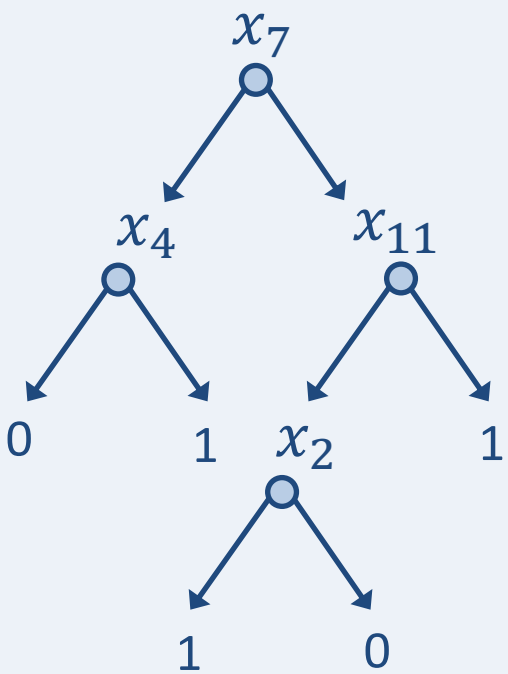
Given queries to a function

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



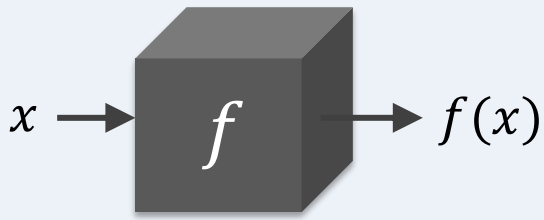
Output **decision tree** representation of f

Decision trees: simple and effective

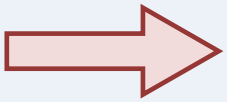


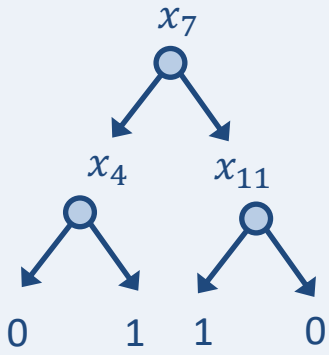
- Fast to evaluate
- Easy to understand its structure,
  Easy to explain its behavior

Want fast and accurate algorithms



Fast algorithm





Assumption about f:

Can be represented by a small decision tree

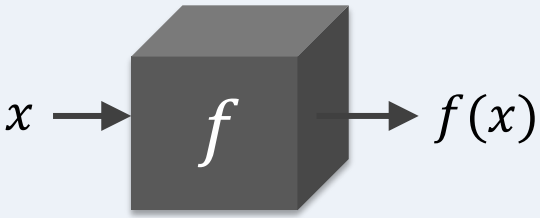
Accurate tree T:

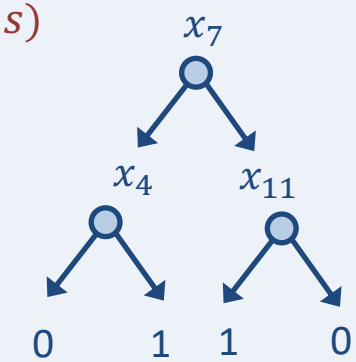
$$\Pr_{\text{uniform } \boldsymbol{x}}[T(\boldsymbol{x}) \neq f(\boldsymbol{x})] \leq \varepsilon$$

Our result: A new algorithm for this task

Algorithm with runtime

$\mathrm{poly}(n) \cdot s^{O(\log\log s)}$





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

f has a size-s DT

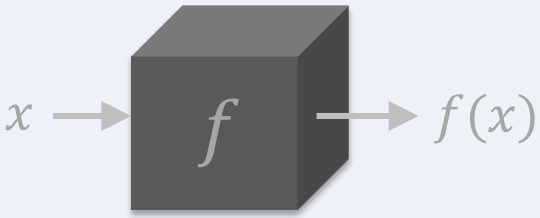
Size-s DT T such that:

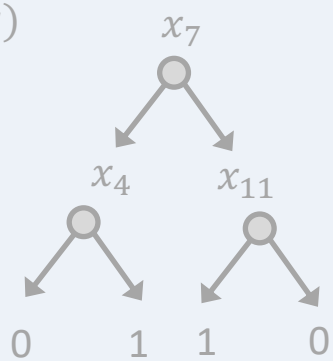
$$\Pr_{\text{uniform } \boldsymbol{x}}[T(\boldsymbol{x}) \neq f(\boldsymbol{x})] \leq \varepsilon$$

Our algorithm is furthermore agnostic

Algorithm with runtime

$\mathrm{poly}(n) \cdot s^{O(\log\log s)}$





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

f has a size-s DT

f is **opt-close** to size-s DT

Size-s DT T such that:

$\Pr_{\text{uniform } \boldsymbol{x}}[T(\boldsymbol{x}) \neq f(\boldsymbol{x})] \leq \varepsilon$

$\leq \mathrm{opt} + \varepsilon$

Equivalently, our algorithm is noise tolerant

Comparison with prior work

- For $s = \mathrm{poly}(n)$, our runtime is:

$$\mathrm{poly}(n) \cdot s^{O(\log\log s)} = n^{\mathrm{O}(\log\log n)}$$

- Previous fastest runtime $n^{\mathrm{O}(\log n)}$, even in the realizable setting
  - 3 different algorithms:

    [Ehrenfeucht–Haussler 89, Mehta–Raghavan 02, Blanc–Lange–T. 20]
  - Analyses of all 3 algorithms tight: $n^{\Omega(\log n)}$ lower bounds

Formal statement of our result

Given queries to $f: \{0,1\}^n \to \{0,1\}$ that is opt_s-close to size-s DT, our algorithm runs in time:

$$\tilde{O}(n^2) \cdot (s/\varepsilon)^{O(\log((\log s)/\varepsilon))}$$

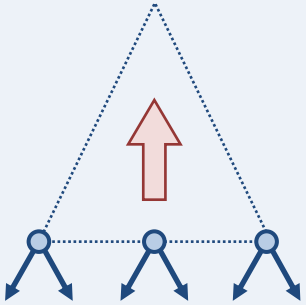
and outputs a size-s DT that is $(\mathrm{opt}_s + \varepsilon)$-close to f.

In the language of learning theory:

We give a new **query algorithm** for **properly** and **agnostically** learning decision trees under the **uniform distribution**

Two general approaches to learning DTs

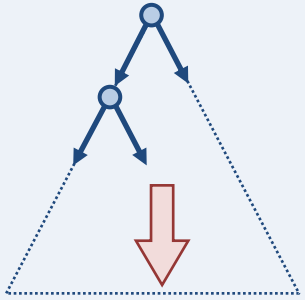
Bottom up



Ehrenfeucht–Haussler 89

Mehta–Raghavan 02

Top down



Blanc–Lange–T. 20

Our algorithm combines both approaches: “[BLT] meets [MR]”

Outline for the rest of this talk

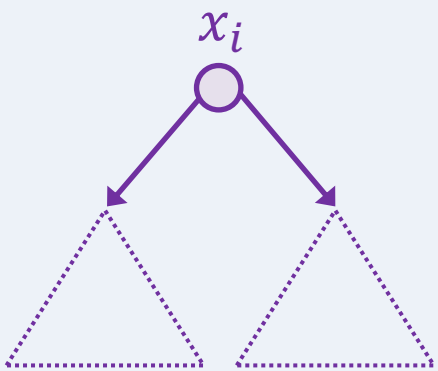
- [BLT]'s top-down algorithm and its limitations
- Our algorithm and its key ingredient: **decision tree pruning lemma**

Strengthens theorem of [O'Donnell, Saks, Schramm, Servedio 05], which was key ingredient in [BLT]

- Challenges in the agnostic setting and how we overcome them

Top-down algorithms for learning DTs

1. Choose a "good" variable x_i to query as root
2. Recurse on $f_{x_i=0}$ and $f_{x_i=1}$



- Definition of "good" = splitting criterion of algorithm
- Intuitively, want most "important"/"informative" variable

Influence as a splitting criterion

[BLT]'s splitting criterion:

Definition. The <u>influence</u> of a variable i on $f: \{0,1\}^n \to \{0,1\}$ is the quantity:

$$\mathrm{Inf}_i(f) \coloneqq \Pr_{\text{uniform } \boldsymbol{x}}\left[f(\boldsymbol{x}) \neq f(\boldsymbol{x}^{\oplus i})\right]$$

where $\boldsymbol{x}^{\oplus i}$ denotes $\boldsymbol{x}$ with its i-th coordinate flipped.

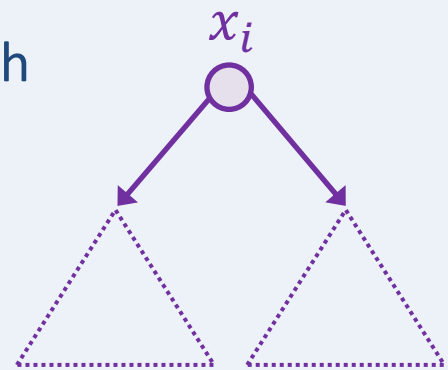
Splitting criteria used in practice: Gini impurity, information gain, etc.

See [BLT] for detailed discussion.

"Top-down induction of decision trees: rigorous guarantees and inherent limitations" Blanc, Lange, Tan. ITCS 2020

[BLT]'s top-down algorithm

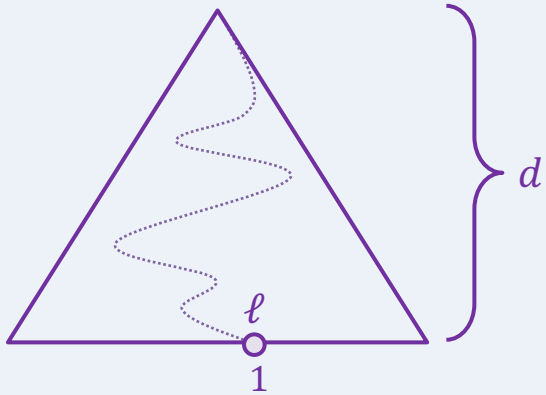
1. Using queries to f, identify the variable i with (approximately) the **largest influence** on f



2. Query x_i as the **root** of tree

3. Build left and right subtrees by **recursing** on $f_{x_i=0}$ and $f_{x_i=1}$

4. When tree reaches user-specified depth d, **label each leaf** ℓ with $\mathrm{round}(\mathbb{E}[f_\ell]) \in \{0,1\}$



Guarantees and limitations of this algorithm

Theorem [BLT]. Let $f: \{0,1\}^n \to \{0,1\}$ be a size-s DT.

Growing this "most-influential-at-the-root" tree to depth

$$d \leq O(\log(s/\varepsilon))^2$$

guarantees accuracy $\geq 1-\varepsilon$ with respect to f.

Remark: Depth $O(\log s)^2 \implies$ Size $s^{O(\log s)}$

Matching lower bound. There is a size-s DT f such that this tree must be grown to depth

$$d \geq \widetilde{\Omega}(\log s)^2$$

even just to achieve accuracy $\geq 51\%$.

Outline of this talk

- [BLT]'s top-down algorithm and its limitations
- Our algorithm and its key ingredient: **decision tree pruning lemma**

Strengthens theorem of [O'Donnell, Saks, Schramm, Servedio 05], which was key ingredient in [BLT]

- Challenges in the agnostic setting and how we overcome them

Being less greedy

[BLT]:

The most influential variable is a “somewhat good” root

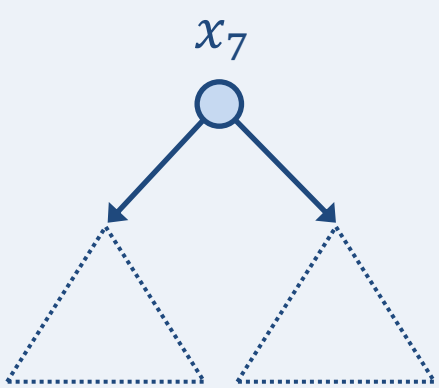
This work:

There’s an *even better root* among the k most influential variables

The difference in more detail

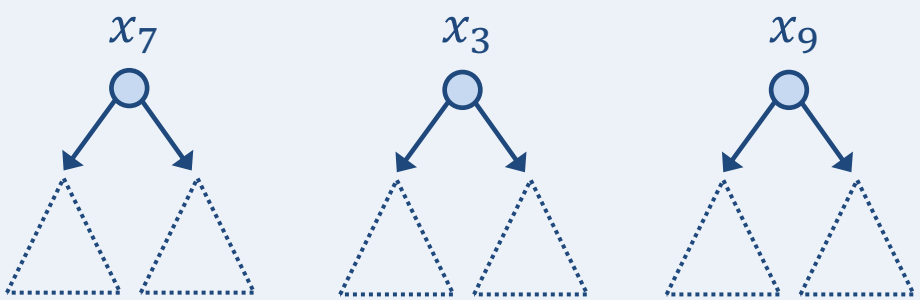
Let f be a size-s DT.

[BLT]: Recursively querying the <u>most influential</u> variable yields accurate tree for f if grown to depth $O(\log s)^2$.



[BLT] commits to the most influential variable as the root

Our Less-Greedy Thm: Exists variable x_i among the <u>polylog(s) most influential</u> ones s.t. there is accurate tree for f of depth $O(\log s)$ with x_i as its root.



We consider a small set of candidate roots

Queries to f

Greediness parameter k

Depth budget d

Less-Greedy(f, k, d):

1. If $d = 0$, return $\mathrm{round}(\mathbb{E}[f]) \in \{0,1\}$
2. Identify set $S \coloneqq \{\ k$ most influential variables of $f\ \}$
3. For each $i \in S$, construct tree $T_i \coloneqq$ x_i

Less-Greedy$(f_{x_i=0}, k, d\text{-}1)$ **Less-Greedy**$(f_{x_i=1}, k, d\text{-}1)$

4. Return tree T_i with minimal error w.r.t. f

Time complexity:
- **Less-Greedy**(f, k, d) makes $(2k)^d$ recursive calls
- S can be identified in $\mathrm{poly}(n)$ time

Done, assuming Less-Greedy Theorem

Less-Greedy Theorem: If f is a size-s DT, exists a variable x_i among the $\mathrm{polylog}(s)$ most influential ones s.t. there is an accurate tree for f of depth $O(\log s)$ with x_i as its root.

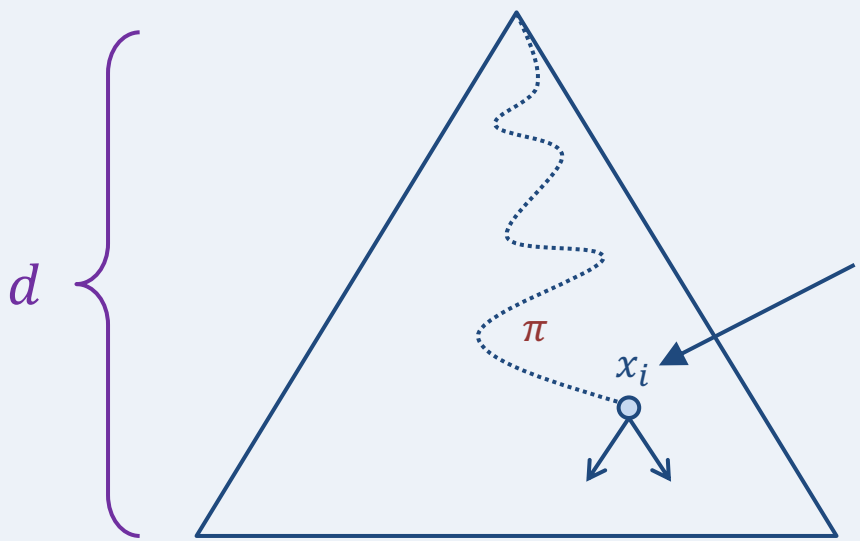
<u>Equivalently</u>:

Less-Greedy(f, k, d) with $k \coloneqq \mathrm{polylog}(s)$ and $d \coloneqq O(\log s)$ returns an accurate tree for f

- # recursive calls = $(2k)^d$ = $s^{O(\log\log s)}$
- Overall runtime = $\mathrm{poly}(n) \cdot s^{O(\log\log s)}$ $\square$

Search space of Less-Greedy

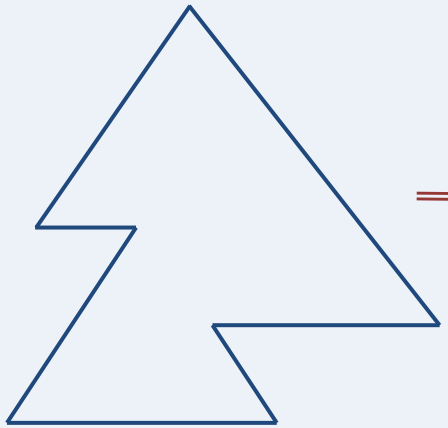
Which trees does **Less-Greedy**(f, k, d) search over?



For all paths π, variable queried is one of the k most influential ones of f_π

Less-Greedy Theorem via Pruning

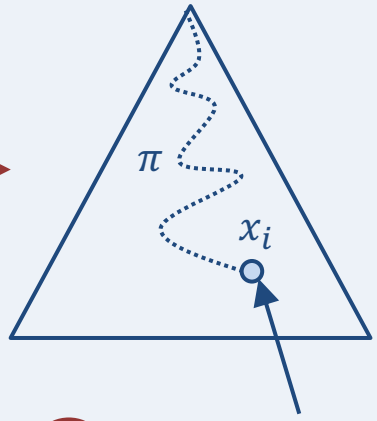
Size-s tree T for f



Prune

1 $\leq \varepsilon$ error

Pruned tree T'



2 $d \leq \log(s/\varepsilon)$

3 $\forall \pi\colon\ \mathrm{Inf}_i(f_\pi) \geq \tau := \frac{\varepsilon}{\log(s/\varepsilon)}$

This proves the **Less-Greedy Theorem** because

1: T' is an accurate tree for T

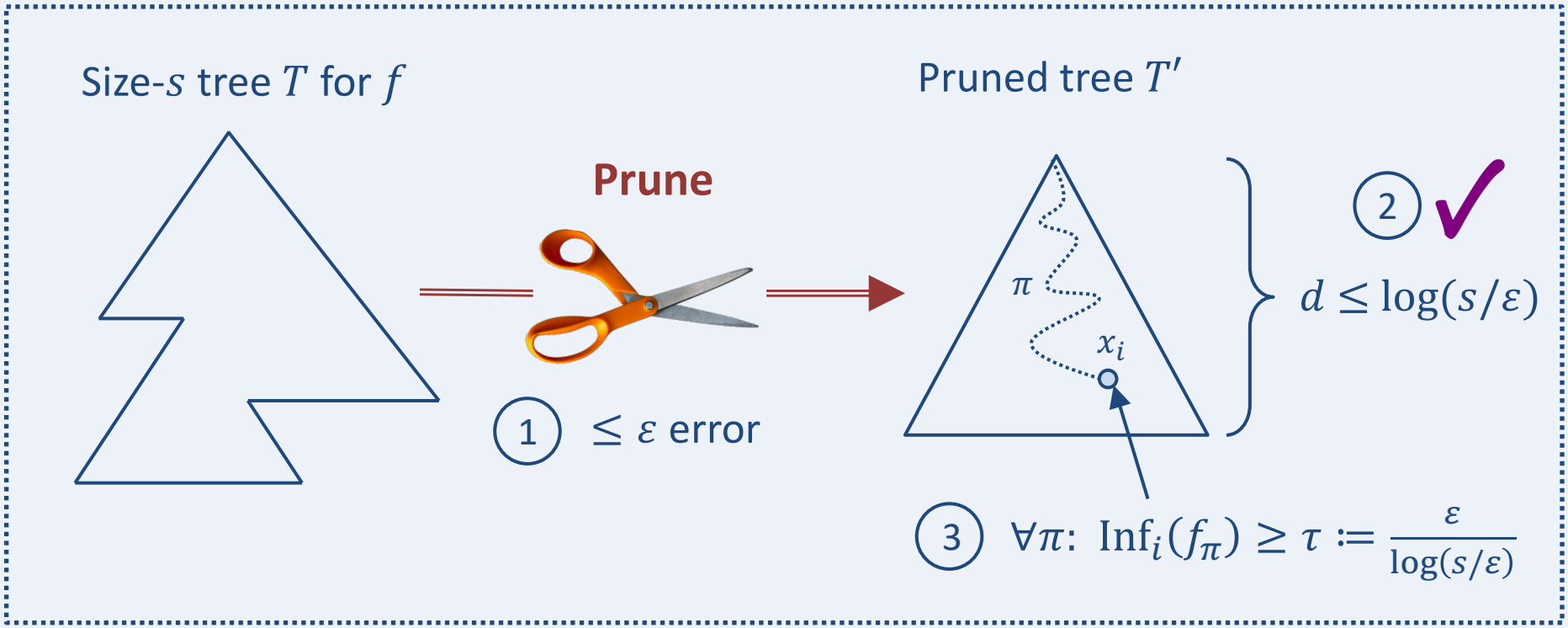
2+3: T' is in search space of **Less-Greedy**(f, k, d)

Since f_π is a size-s tree, at most

$$k := (\log s)/\tau = O(\log s)^2$$

such variables.

Small depth is easy to achieve

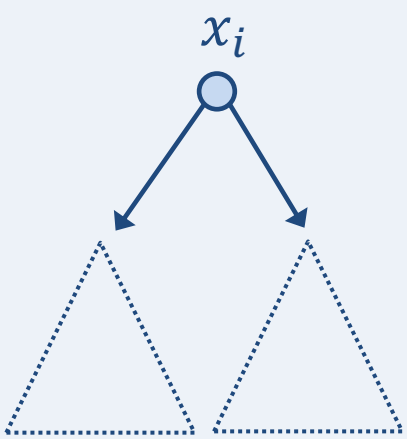


Easy fact: Truncating size-s tree at depth $\log(s/\varepsilon)$ incurs $\leq \varepsilon$ error.

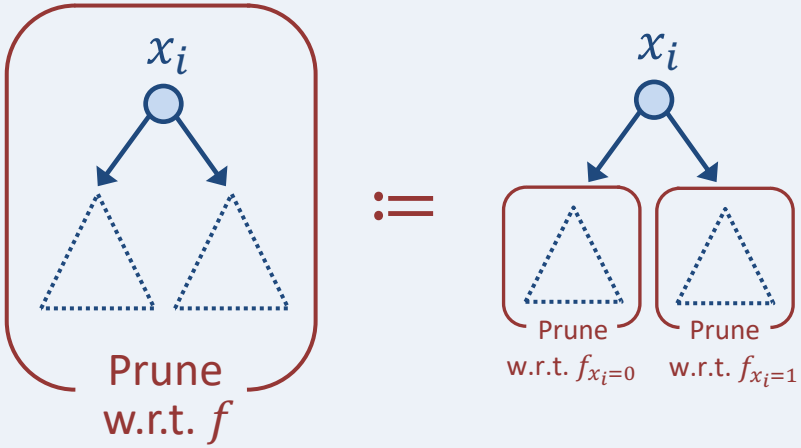
Achieving high influence at all nodes

We enforce high influence recursively, starting at the root

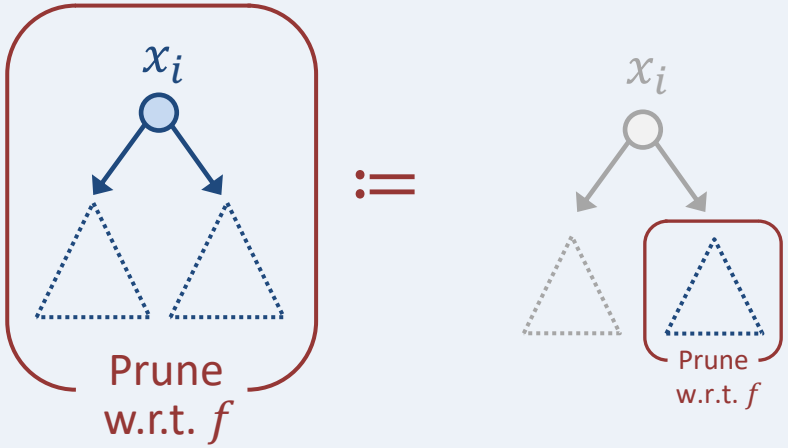
Q: $\mathrm{Inf}_i(f) \geq \tau$?



If so: keep root; prune subtrees recursively



If not: Discard root + subtree that's further from f

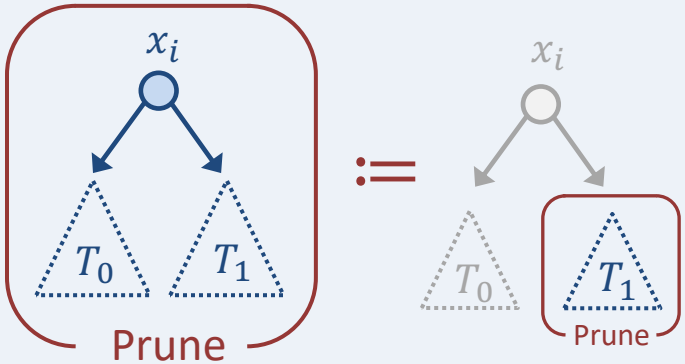


Pruning Lemma and its proof

Lemma: Let f be a function and T be a depth-d DT. Then:

$$\mathrm{dist}\big(f, \mathrm{Prune}_{f,\tau}(T)\big) \le \mathrm{dist}(f, T) + d \cdot \tau.$$

Proof: Induction on d. Only do harder case where $\mathrm{Inf}_i(f) \le \tau$.



$$\mathrm{dist}\big(f, \mathrm{Prune}_{f,\tau}(T)\big)$$

(min ≤ avg) $$\le \mathop{\mathbb{E}}_{\boldsymbol{b}}\big[\mathrm{dist}\big(f, \mathrm{Prune}_{f,\tau}(T_{\boldsymbol{b}})\big)\big]$$

(IH) $$\le \mathbb{E}[\mathrm{dist}(f, T_{\boldsymbol{b}}) + (d-1)\cdot\tau]$$

(Δ-ineq.) $$\le \mathbb{E}\big[\mathrm{dist}(f, f_{x_i=\boldsymbol{b}})\big] + \mathbb{E}\big[\mathrm{dist}(f_{x_i=\boldsymbol{b}}, T_{\boldsymbol{b}})\big] + (d-1)\cdot\tau$$

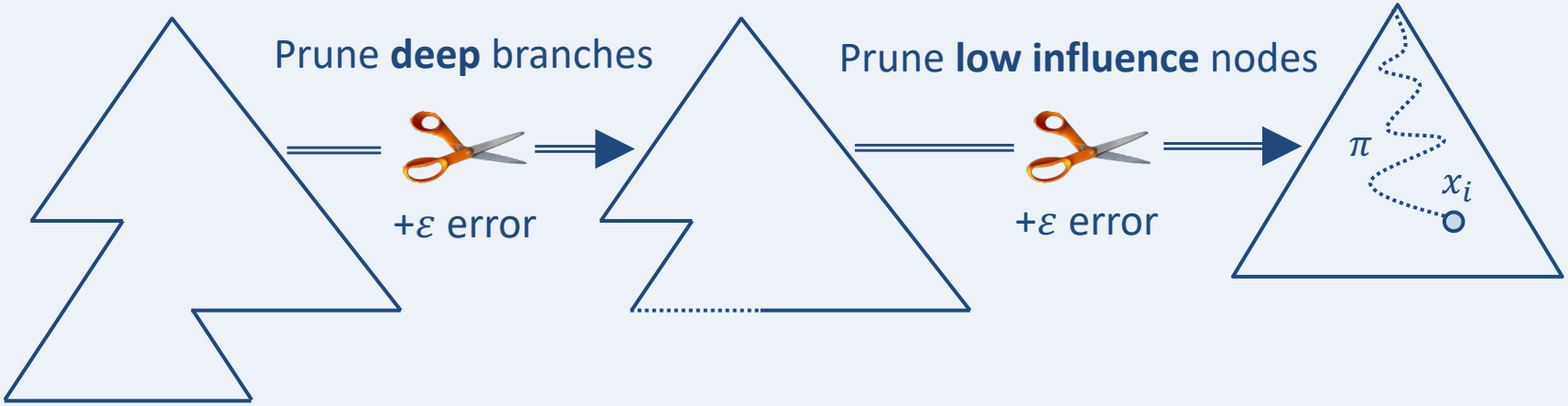
$$= \mathrm{Inf}_i(f) + \mathrm{dist}(f, T) + (d-1)\cdot\tau$$

($\mathrm{Inf}_i(f) \le \tau$) $$\le \mathrm{dist}(f, T) + d\cdot\tau.$$

□

Putting everything together

Size-s tree T for f Depth $d \leq \log(s/\varepsilon)$ $\forall \pi\colon\ \mathrm{Inf}_i(f_\pi) \geq \tau := \frac{\varepsilon}{d}$



- Final tree falls into search space of **Less-Greedy**(f, k, d) where:

$$k \leq (\log s)/\tau = O(\log s)^2, \quad d \leq O(\log s)$$

- **Less-Greedy** finds it in time $\mathrm{poly}(n) \cdot (2k)^d = \mathrm{poly}(n) \cdot s^{O(\log\log s)}$ $\square$

Zooming out:
Our algorithm vs. previous ones

Our algorithm: Less-Greedy(f, k=polylog(s), d=$\log s$)

Runtime: $(2k)^d = s^{O(\log\log s)}$

[BLT]: Less-Greedy(f, k=1, d=$(\log s)^2$)

Runtime: $(2k)^d = s^{O(\log s)}$

[MR]: Less-Greedy(f, k=n, d=$\log s$)

Runtime: $(2k)^d = n^{O(\log s)}$

"Decision tree approximations of Boolean functions" Mehta and Raghavan. COLT 2000

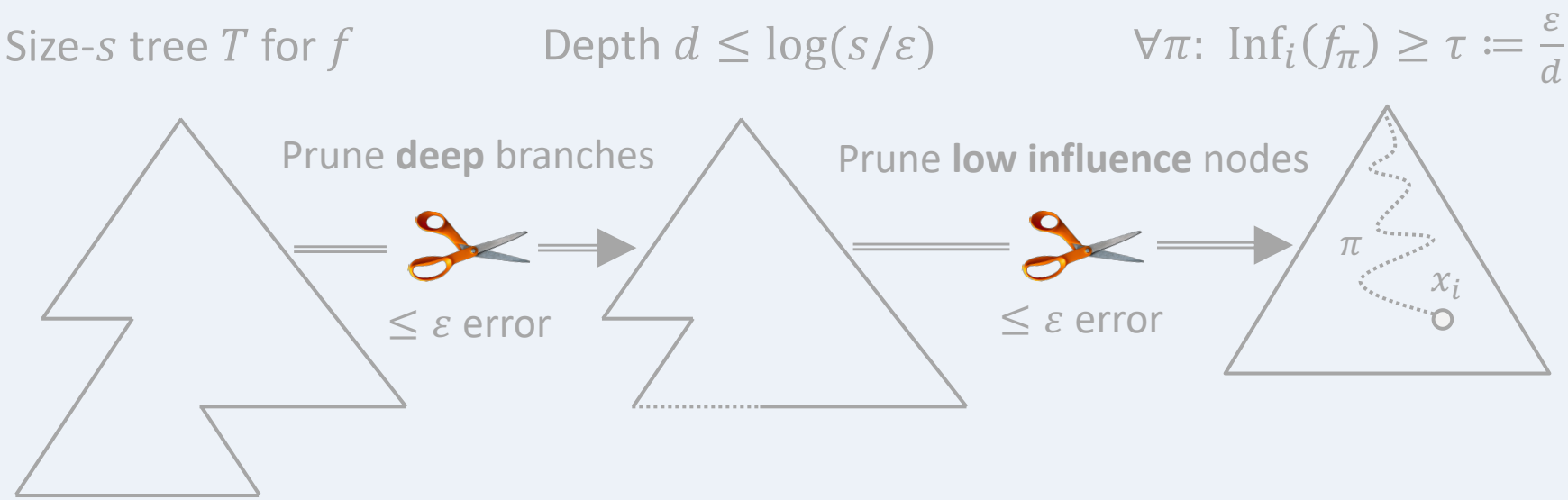
Outline of this talk

- [BLT]'s top-down algorithm and its limitations
- Our algorithm and its key ingredient: **decision tree pruning lemma**

Strengthens theorem of [O'Donnell, Saks, Schramm, Servedio 05], which was key ingredient in [BLT]

- Challenges in the agnostic setting and how we overcome them

Pruning lemma and the OSSS theorem



Pruning lemma easily implies:

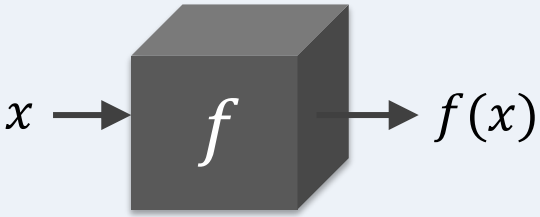
Theorem [OSSS]. Every size-s DT T is either ε-close to constant, or

$$\exists\, i \in [n] \text{ s.t. } \mathrm{Inf}_i(T) \geq \varepsilon/\log(s).$$

Proof. If T has no influential variables, our pruning algorithm prunes everything away, resulting in a constant that's ε-close to T. $\square$

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

Agnostic Learning

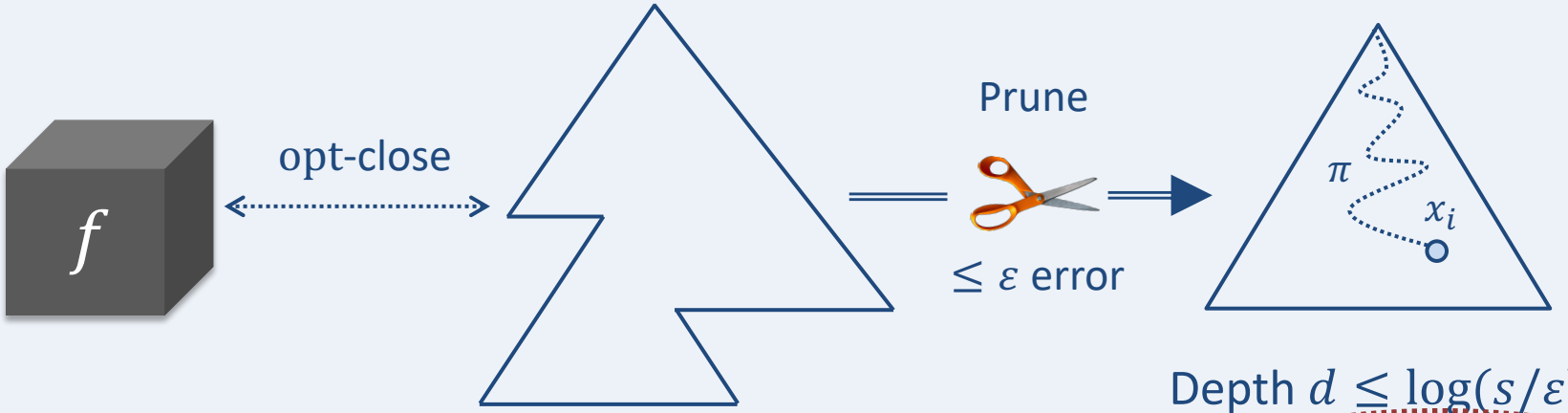


Assumption about f:

Realizable setting: f is a size-s DT

Agnostic setting: f is opt-close to a size-s DT

Why our current algorithm fails in the agnostic setting



Depth $d \leq \log(s/\varepsilon)$

$\forall \pi$: $\mathrm{Inf}_i(f_\pi) \geq \tau := \frac{\varepsilon}{d}$

Realizable setting: Since f_π is size-s DT, $\leq (\log s)/\tau = O(\log s)^2$ variables with influence $\geq \tau$.

Proof. Every size-s DT has total influence $\leq \log s$.

False if f_π is merely *close to* size-s DT

Solution: Smoothing things out

<u>Definition</u>: For a function $f: \{0,1\}^n \to \{0,1\}$, its **$\boldsymbol{\delta}$-smoothed version** is the function $f_\delta: \{0,1\}^n \to [0,1]$,

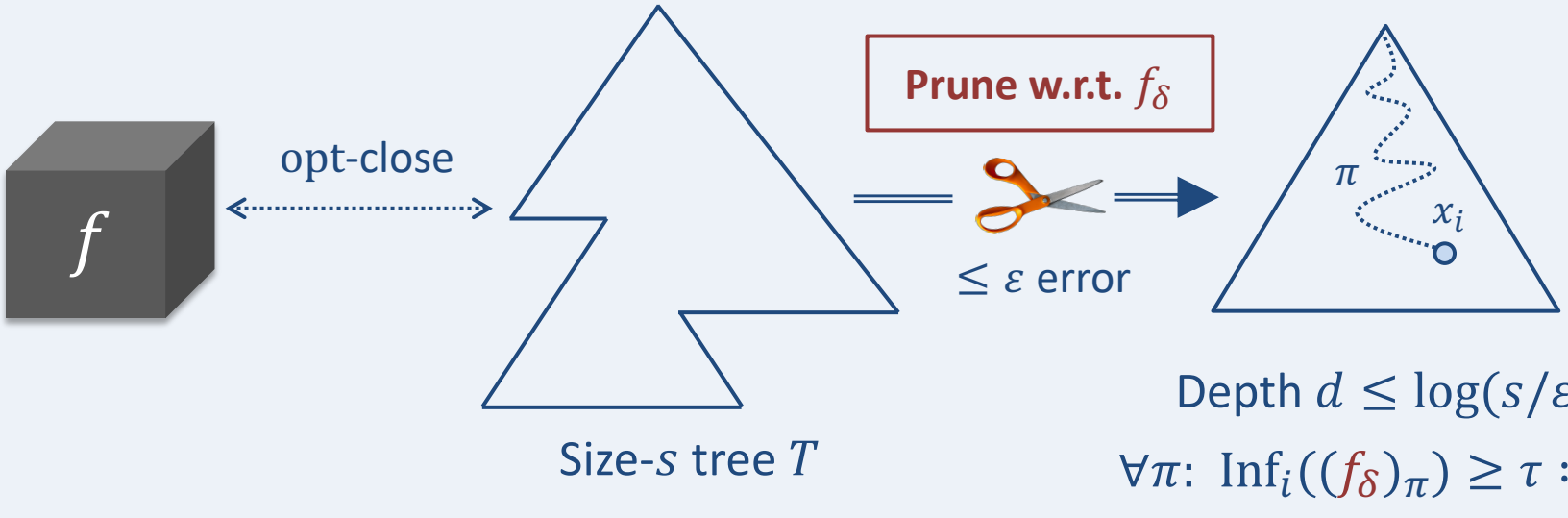
$$f_\delta(x) := \mathbb{E}[f(\boldsymbol{y})]$$

where $\boldsymbol{y} = x$ with each bit flipped w.p. δ.

Why is this useful for us?

Standard Fourier fact: For all functions f, total influence of f_δ is $\leq O(1/\delta)$.

Our algorithm for the agnostic setting



- Key difference: now can bound # variables with $\mathrm{Inf}_i((f_\delta)_\pi) \geq \tau$ by $1/(\delta\tau)$.
- **Less-Greedy**(f_δ, k, d) finds this tree, which we prove is $(\mathrm{opt} + \varepsilon)$-close to f

Recap and future directions

Proper learning in polynomial time?

Our main result:

Given queries to $f: \{0,1\}^n \to \{0,1\}$ that is opt_s-close to size-s DT, our algorithm runs in time:

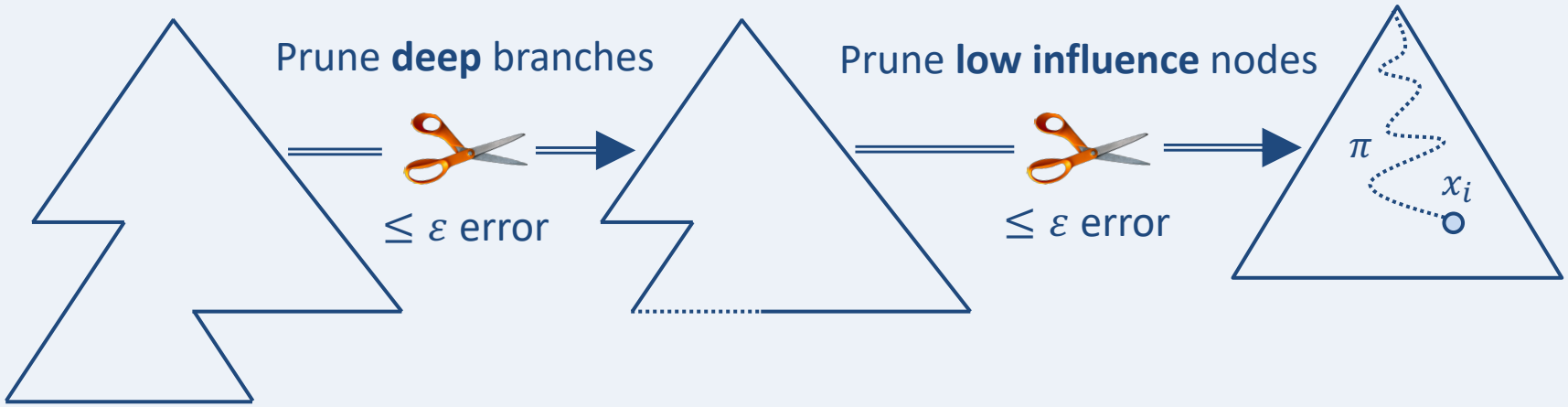
$$\tilde{O}(n^2) \cdot (s/\varepsilon)^{O(\log((\log s)/\varepsilon))}$$

and outputs a size-s DT that is $(\mathrm{opt}_s + \varepsilon)$-close to f.

- A $\mathrm{poly}(n, s)$-time algorithm would be great, even just in the realizable setting
- ... via a new splitting criterion?

Further applications of our pruning lemma?

Size-s tree T for f Depth $d \leq \log(s/\varepsilon)$ $\forall \pi\colon\ \mathrm{Inf}_i(f_\pi) \geq \tau := \frac{\varepsilon}{d}$



Applications of the O'Donnell–Saks–Schramm–Servedio theorem:

- Original application: Evasiveness of **graph properties**
- Subsequent applications in **percolation theory**

Mathematics > Probability

[Submitted on 6 May 2020]

The OSSS Method in Percolation Theory

Julian Kern

Thank you for listening.