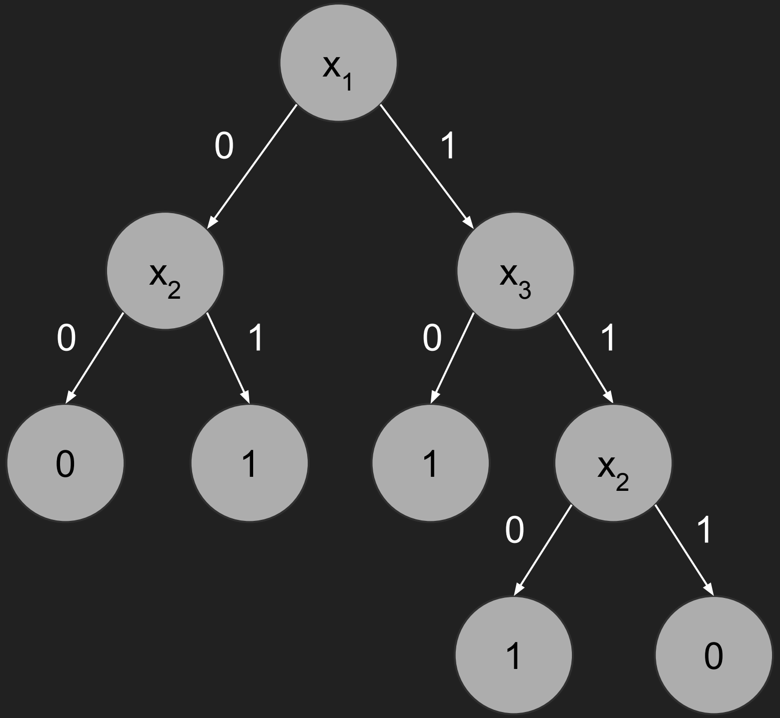
Top-down induction of decision trees: rigorous guarantees and inherent limitations

Guy Blanc, Jane Lange, Li-Yang Tan

Stanford University

This work: Learning decision trees from labeled data

x	f(x)
000010101	0
011011010	1
100100111	1
101001000	1
001010010	0



Induction of **decision trees** - **Quinlan** - Cited by 21867

C4. 5: programs for **machine learning** - Quinlan - Cited by 37060

Classification and **regression trees** - Breiman - Cited by 43990

“In experimental and applied machine learning work, it is hard to exaggerate the influence of top-down heuristics for building a decision tree from labeled sample data” - [Kearns and Mansour 96]

Decision trees also intensively studied in TCS

- Query model of computation
- Quantum complexity
- Derandomization
- ...
- **Learning theory**
  - [Ehrenfeucht-Haussler 89, Goldreich-Levin 89, Kushilevitz-Mansour 92, … MR02, OS07, GKK08, HKY18, CM19, …]

Theory vs. practice of learning decision trees: A disconnect

Practical heuristics work “top-down”

ID3, C4.5, CART

Theoretical algorithms work “bottom-up”

[EH89, MR02]

Our results (Part 1):
Rigorous guarantees and inherent limitations

Our results (Part 2):
Theoretical algorithms with improved guarantees

Theory vs. practice of learning decision trees: A disconnect

Practical heuristics work “top-down”

ID3, C4.5, CART

Our results (Part 1):
Rigorous guarantees and inherent limitations

Theoretical algorithms work “bottom-up”

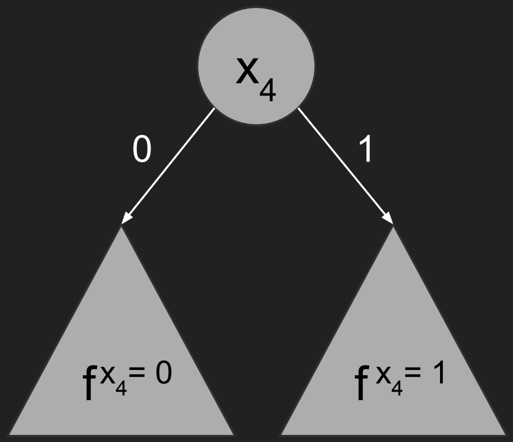
[EH89, MR02]

Our results (Part 2):
Theoretical algorithms with improved guarantees

Top-down induction of decision trees

1) Determine "good" variable to query as root

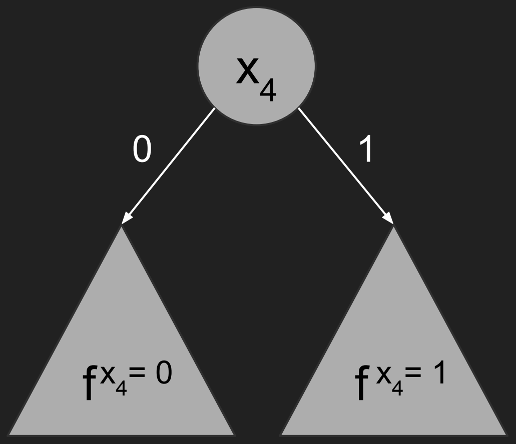
2) Recurse on both subtrees



Top-down induction of decision trees

1) Determine "good" variable to query as root

2) Recurse on both subtrees



"Good" variable = one that is very "relevant," "important," "influential"

Our splitting criterion: Influence

$$\mathrm{Inf}_i(f) := \Pr_{\boldsymbol{x}\sim\{0,1\}^n}[f(\boldsymbol{x}) \neq f(\boldsymbol{x}^{\oplus i})]$$

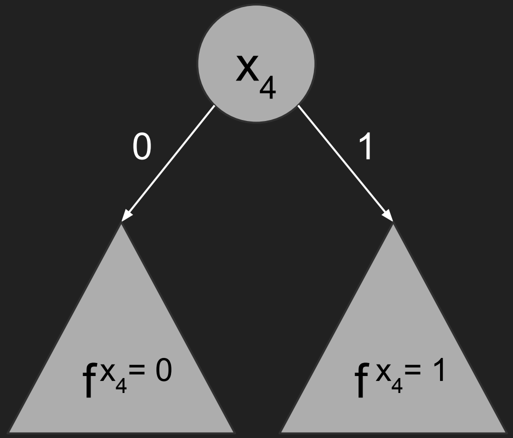
x with the i^{th} bit flipped

Basic and well-studied notion with applications throughout TCS

Our algorithm: TopDown

1) Query the **most influential variable** of f at the root

2) Recurse on both subtrees



Our results: Provable guarantees and inherent limitations of TopDown

A guarantee for all functions

Theorem: Let f be a size-s decision tree. TopDown builds a tree of size at most $s^{O(\log(s/\varepsilon)\log(1/\varepsilon))}$ that ε-approximates f

A matching lower bound

Theorem: For any s and ε, there is a size-s decision tree f such that the size of TopDown(f, ε) is $s^{\tilde{\Omega}(\log s)}$

A guarantee for monotone functions

Theorem: Let f be a monotone size-s decision tree. TopDown builds a tree of size at most $s^{O(\sqrt{\log s}/\varepsilon)}$ that ε-approximates f.

A near-matching lower bound

Theorem: For any s and ε, there is a monotone size-s decision tree f such that the size of TopDown(f, ε) is $s^{\tilde{\Omega}(\sqrt[4]{\log s})}$.

A bound of poly(s) had been conjectured by [FP04].

Algorithmic consequences

- Properly learn decision trees in time $s^{O(\log(s/\varepsilon)\log(1/\varepsilon))}$
  - Runtime compares favorably with best algorithm with provable guarantee [EH89]
  - Downside: requires query access to the function

- For monotone functions, properly learn decision trees in time $s^{O(\sqrt{\log s}/\varepsilon)}$ using only random examples
  - For monotone functions, influence = splitting criteria used in practical heuristics (ID3, C4.5, and CART)
  - Provable guarantees on these heuristics for a broad and natural class of data sets

Theory vs. practice of learning decision trees: A disconnect

Practical heuristics work “top-down”

ID3, C4.5, CART

Theoretical algorithms work “bottom-up”

[EH89, MR02]

Our results (Part 1):
Rigorous guarantees and inherent limitations

Our results (Part 2):
Theoretical algorithms with improved guarantees

Improving Ehrenfeucht-Haussler (1989)

Theorem [EH89]: There is a quasi-polynomial time algorithm for properly learning decision trees.

Theorem (Our work): There is a quasi-polynomial time algorithm for properly learning decision trees with **polynomial** memory and sample complexity.

Thank you!

Practical heuristics work “top-down”

ID3, C4.5, CART

Theoretical algorithms work “bottom-up”

[EH89, MR02]

Our results (Part 1):
Rigorous guarantees and inherent limitations

Our results (Part 2):
Theoretical algorithms with improved guarantees