A query-optimal algorithm for finding counterfactuals

Caleb Koch

Joint work with:

Guy Blanc

Stanford

Jane Lange

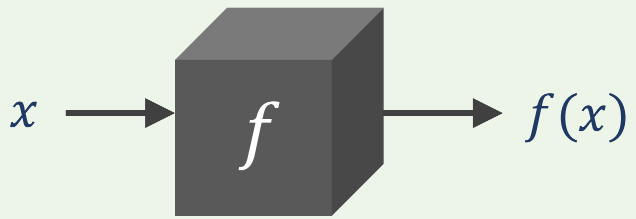
MIT

Li-Yang Tan

Stanford

Explaining the behavior of black boxes

Setup: Query access to an unknown function $f: \{0,1\}^d \to \{0,1\}$

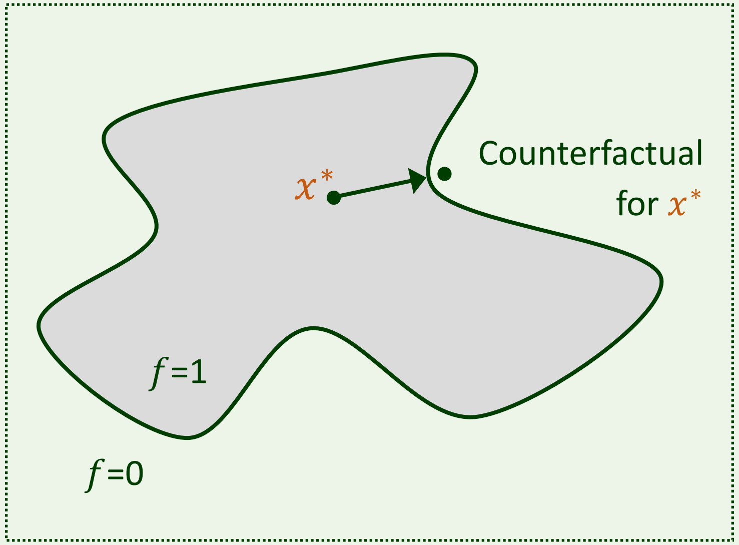


Goal: Given a specific input x^*, explain ***why*** f outputs $f(x^*)$ for x^*

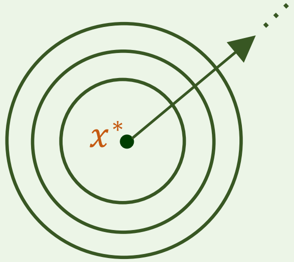
- Recent surge of interest from explainable ML
  - *Local* explanations: understand model's prediction for *specific* inputs
  - *Model agnostic* explanations: independent of the model's internal structure

Counterfactual explanations

A **counterfactual** for f's value at x^* is a close-by point y s.t. $f(x^*) \neq f(y)$



The natural algorithm: Local search



If optimal counterfactual at distance ℓ,
finds it with $\sim d^\ell$ queries.

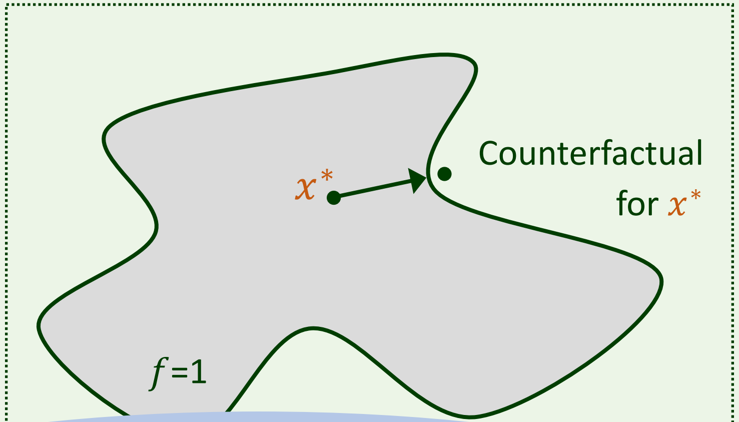
Our main theorem:

For monotone f's, can find optimal counterfactual with
$(\text{smoothness of } f)^\ell \cdot \log d$ queries

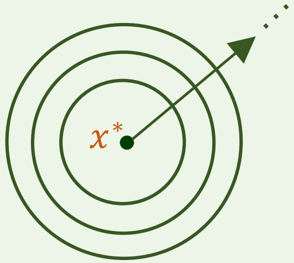
"Counterfactual explanations without opening the black box", Wachter, Mittelstadt, Russell, *Harvard Journal of Law & Technology* (2017)

Counterfactual explanations

A **counterfactual** for f's value at x^* is a close-by point y s.t. $f(x^*) \neq f(y)$



The natural algorithm: Local search



If optimal counterfactual at distance ℓ, finds it with $\sim d^\ell$ queries.

Nearly optimal!

$$\Omega\left((\text{smoothness of } f)^\ell + \log d\right)$$

queries needed in the worst-case

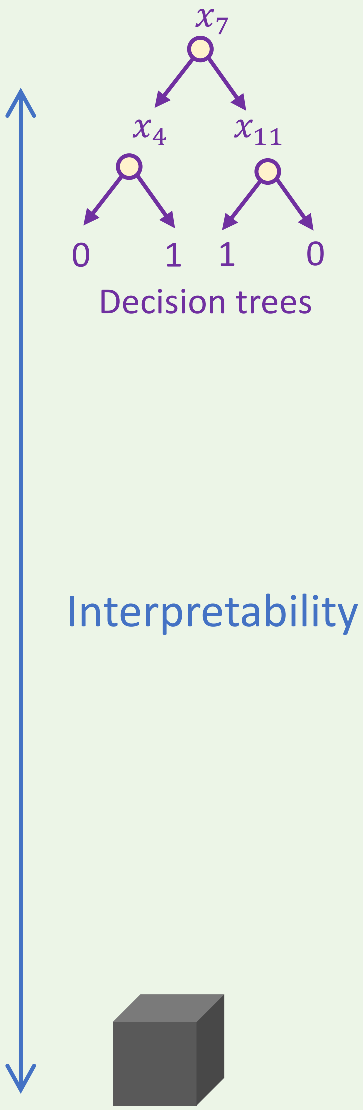
Our main theorem:

For monotone f's, can find optimal counterfactual with

$$(\text{smoothness of } f)^\ell \cdot \log d \ \text{ queries}$$

"Counterfactual explanations without opening the black box", Wachter, Mittelstadt, Russell, *Harvard Journal of Law & Technology* (2017)

Implicit decision trees

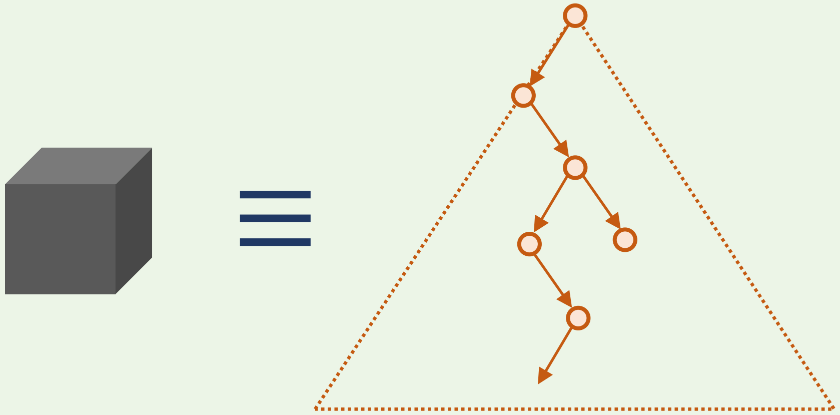


Decision trees

Interpretability

Black boxes

- Popular method for explaining black boxes: convert it into a decision tree
- ... but most models require intractably large DTs
- ***Implicit* decision trees:** efficiently navigate a DT for a black box without building it in full

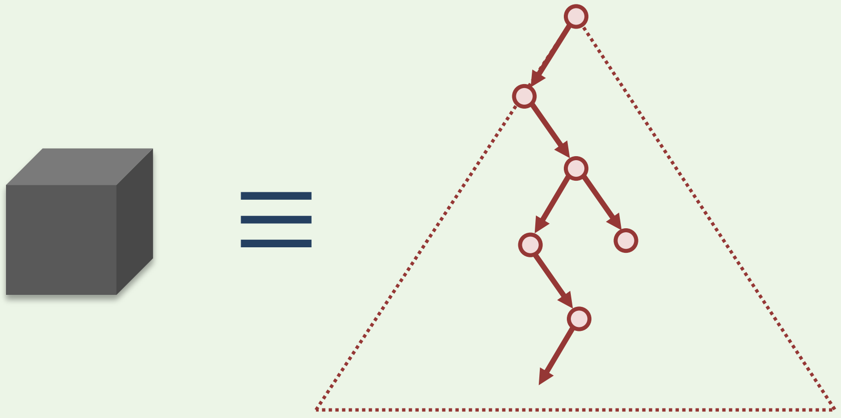


- Motivation: Can often glean useful information about black box from just a tiny portion of the tree

Future directions

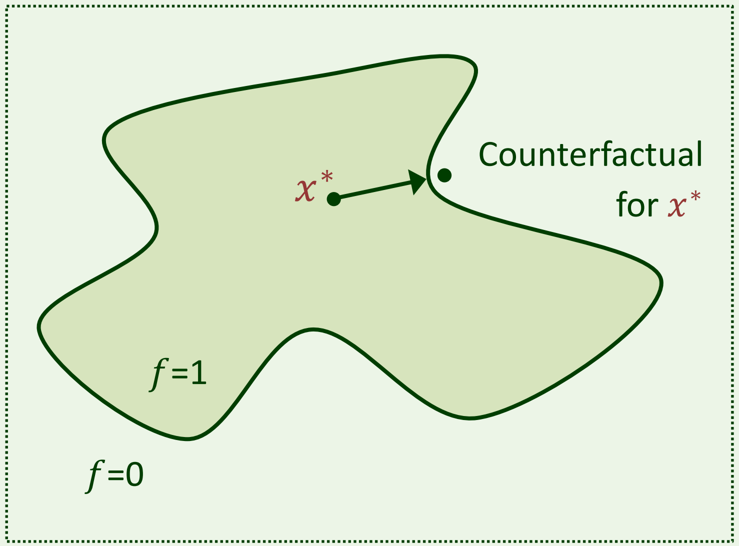
Implicit decision trees are **versatile** tools for understanding black boxes

- Other explanations?
- Other algorithmic applications?



This work uses **sparsity** as the notion of distance

- Other distance metrics?



Thank you for listening.