Counterfactual Explanations for Oblique Decision Trees: Exact, Efficient Algorithms

Miguel Á. Carreira-Perpiñán and **Suryabhan Singh Hada**,
Dept. CSE, UC Merced

1 Motivation and summary

- A counterfactual explanation seeks the minimal change to a given feature vector that will change a classifier's decision in a prescribed way.
- Consider following example:
  - Loan application is denied by bank (classifier).
  - Applicant asks: "what should I change to get it approved"?
  - Bank replies: "If annual income had been \$45,000 instead of \$30,000, the loan would have been approved".
- Counterfactual explanation is important to interpret a black-box decision for a given instance.
- Mathematically, it has the same formulation as classifier inversion and adversarial examples: given a source instance $\overline{\mathbf{x}}$, target class y and a classifier T, find the closest instance $\mathbf{x}$ to $\overline{\mathbf{x}}$ such that $\mathbf{x}$ is classified as y ($T(\mathbf{x}) = y$) .
- Given an input instance $\overline{\mathbf{x}} \in \mathbb{R}^D$, classifier T, and target class y, the problem can be formulated as:

$$\min_{\mathbf{x}\in\mathbb{R}^D} E(\mathbf{x};\overline{\mathbf{x}}) \quad \text{s.t.} \quad T(\mathbf{x}) = y, \quad \mathbf{c}(\mathbf{x}) = \mathbf{0},\ \mathbf{d}(\mathbf{x}) \geq \mathbf{0} \tag{1}$$

  where $E(\mathbf{x};\overline{\mathbf{x}})$ is a cost of changing features of $\overline{\mathbf{x}}$, and $\mathbf{c}(\mathbf{x})$ and $\mathbf{d}(\mathbf{x})$ are problem-dependent equality and inequality constraints.
- Here, we consider as classifier T a decision tree.
- With decision tree T is not differentiable, this makes problem nondifferentiable and non-convex, and gradient-based methods are not applicable. However, this problem can be solved exactly and efficiently.

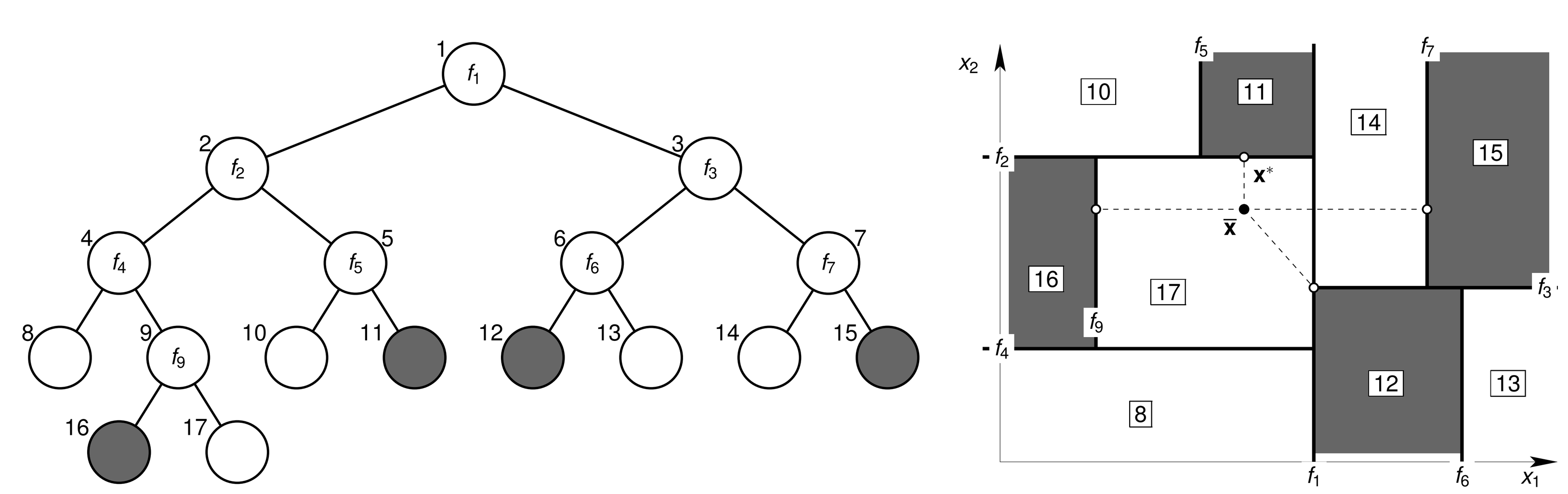
2 Counterfactual explanations in decision trees

- Problem (1) is equivalent to:

$$\min_{i\in\mathcal{L}} \min_{\mathbf{x}\in\mathbb{R}^D} E(\mathbf{x};\overline{\mathbf{x}}) \quad \text{s.t.} \quad y_i = y, \quad \mathbf{h}_i(\mathbf{x}) \geq \mathbf{0},\ \mathbf{c}(\mathbf{x}) = \mathbf{0},\ \mathbf{d}(\mathbf{x}) \geq \mathbf{0}. \tag{2}$$

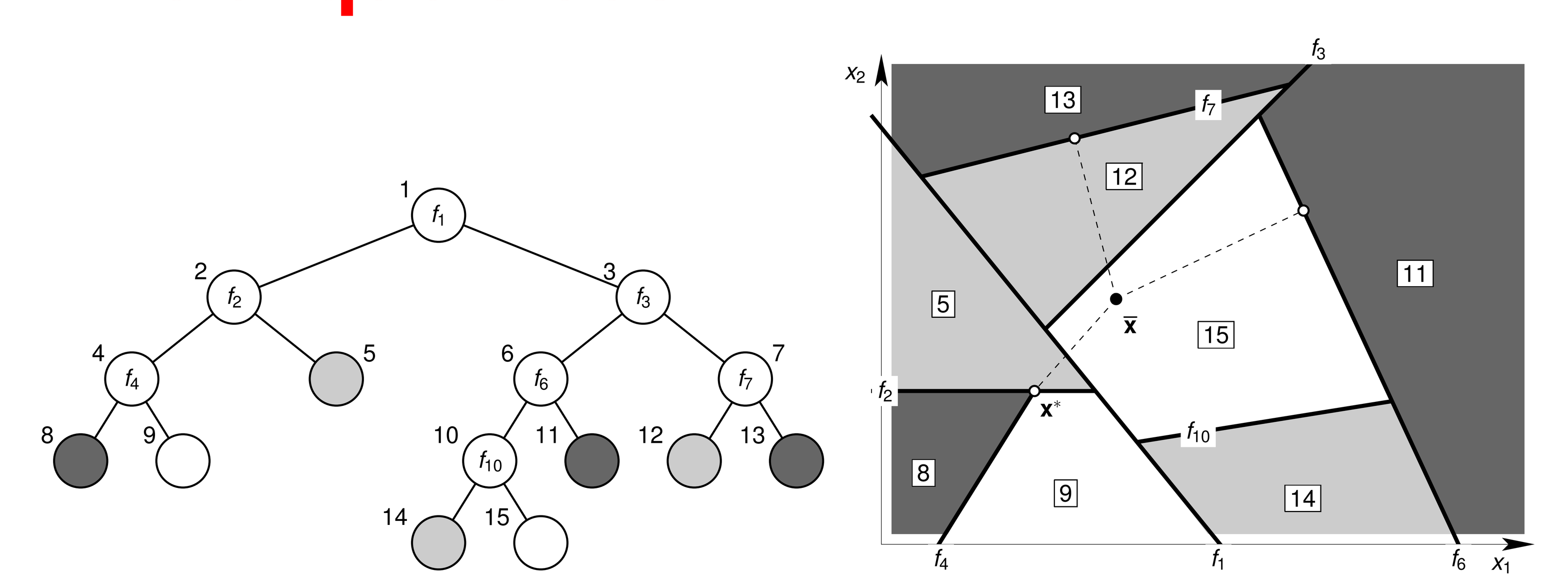
  where $\mathbf{h}_i(.)$ is the set of linear constraints that forms the region of leaf i, which is a polytope.
- Solving problem (1) is equivalent to solving it within each leaf's region and then picking the leaf with the best solution.
- Proposed approach can handle several useful distance functions and linear constraints (equality and inequality); and is applicable to both continuous and categorical variables.
- It can generate multiple different counterfactual explanations based on user need, rather than just giving the globally optimal one.
- Fast enough for interactive use: solving for counterfactual problem takes few milliseconds in all experiments.
- See experiment section in the paper with datasets of different dimensions and types of variables.

Axis-aligned trees



In each leaf region, the problem (2) can be solved for each variable x_d independently, by minimizing $E_d(x_d;\overline{x}_d)$ subject to the constraints on x_d. For each variable x_d the optimal value can be calculated as $\text{median}(\overline{x}_d, l_d, u_d)$, where l_d and u_d are the lower and upper bound on x_d respectively.

Oblique trees



In each leaf region, the problem (2) becomes an LP or QP, which can be solved very effectively.