Counterfactual Explanations for Oblique Decision Trees: Exact, Efficient Algorithms

Miguel Á. Carreira-Perpiñán and Suryabhan Singh Hada
{mcarreira-perpinan, shada}@ucmerced.edu

Dept. Computer Science and Engineering
University of California, Merced

AAAI, February 2021

Counterfactual explanations

- 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 ask: "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 y $(T(\mathbf{x}) = y)$.
- Here, we focus on decision tree classifier (axis-aligned and oblique).

Tree alternating optimization (TAO)

- Classification trees are important, particularly in applications where interpretability is desirable, such as business, law, and medicine.
- Traditionally, decision trees have been trained with a recursive partition procedure, such as CART and C4.5. However, this produces sub-optimal trees and does not work well with oblique trees.
- Tree alternating optimization (TAO) is a recently proposed algorithm that can achieve highly accurate oblique or axis-aligned trees.
- This makes it possible to train models that are both highly interpretable and highly accurate.

Basic formulation of the counterfactual explanation problem

Given an input instance $\overline{\mathbf{x}} \in \mathbb{R}^D$, classifier T, and target class y

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

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 depended equality and inequality constraints.

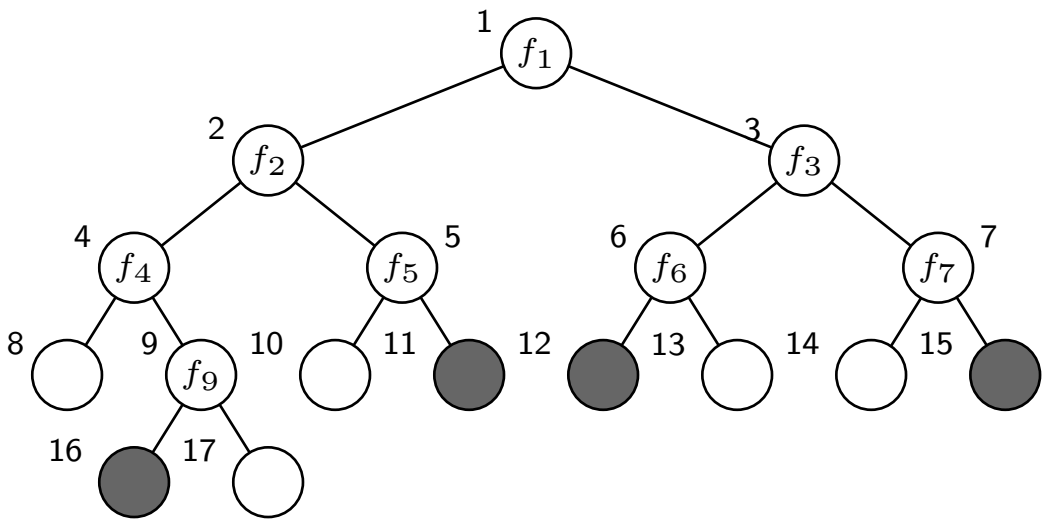
How to solve this optimization problem:

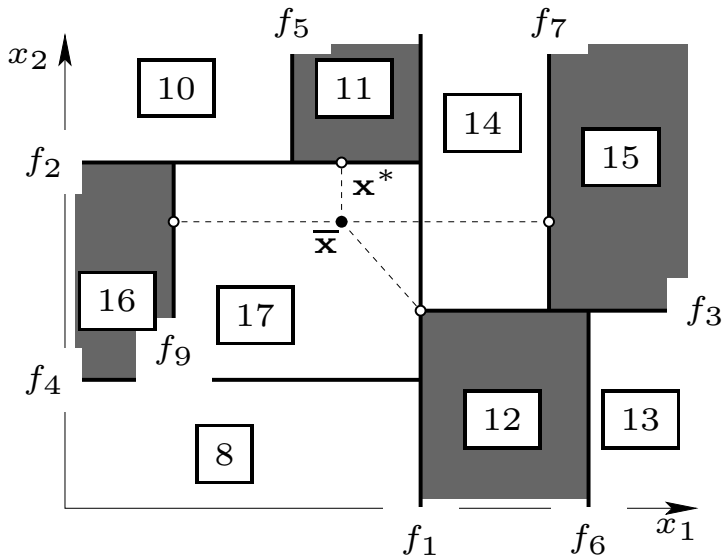
- If T is differentiable with respect to $\mathbf{x}$ the problem can be solved using gradient based methods, for example if T is a neural net.
- With decision trees T is not differentiable, this makes problem nondifferentiable and gradient based methods are not applicable. However, this problem can be solved exactly and efficiently.

Counterfactual solution for decision tree

- A trained decision tree can be regarded as the partition of the input space into disjoint regions, where each region corresponds to one leaf.
- Therefore, finding the closest instance to the source instance having a desired target label can be done by finding closest instance in each leaf region, and picking the best among them.
- For each leaf, the region is defined by a polytope that acts as linear constraints. So, finding the counterfactual in a leaf becomes a quadratic/linear program that can be solved effectively.

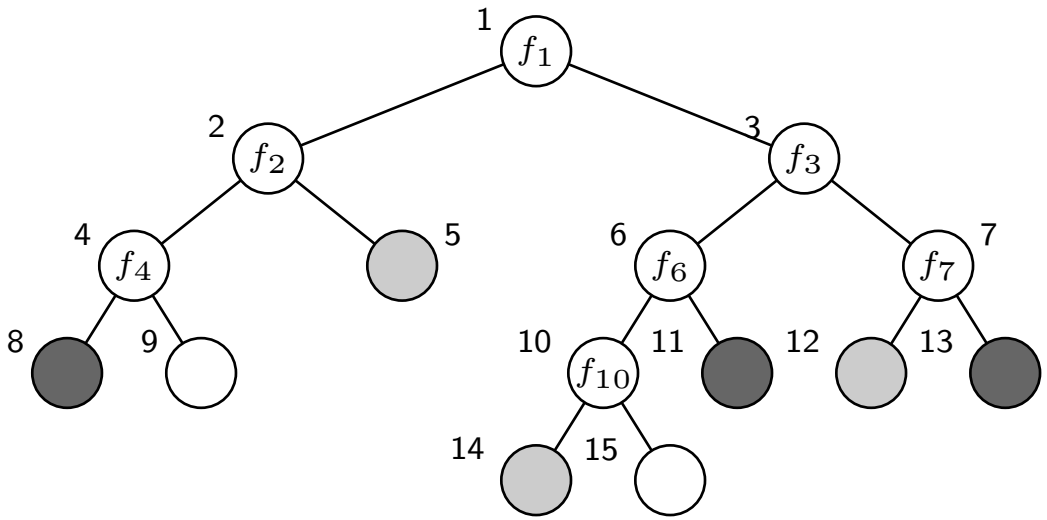
Counterfactual explanations in decision trees
Axis-aligned trees

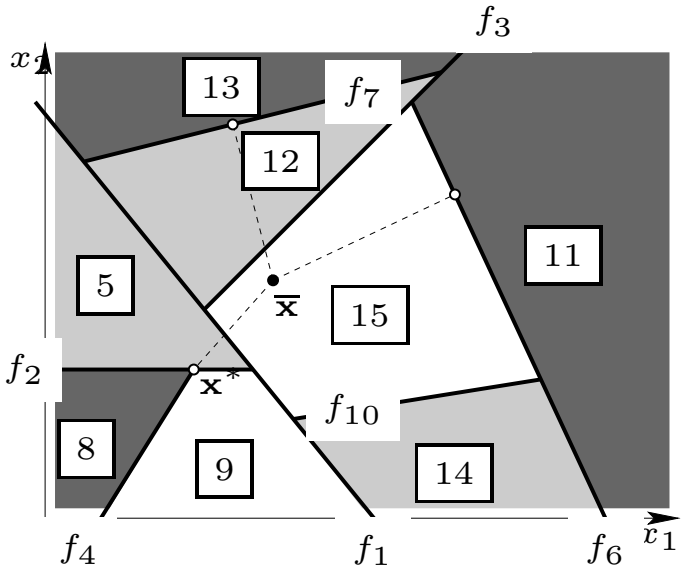




- The tree has L leaves, and it partitions the input space into L axis-aligned boxes.
- Each polytope is defined by the intersection of axis-aligned hyperplanes ("if $x_d \geq 0$ then go to right child, else go to left child") found in the path from root to a leaf.
- Source instance $\overline{\mathbf{x}}$ is in white class.
- The counterfactual instance subject to being in dark grey class is $\mathbf{x}^*$, which is closest to $\overline{\mathbf{x}}$.

Counterfactual explanations in decision trees
Oblique trees





- The tree has L leaves, and it partitions the input space into L polytopes.
- Each polytope is defined by the intersection of arbitrary hyperplanes ("if $\mathbf{w}_i^T \mathbf{z} + b_i \geq 0$ then go to right child, else go to left child") found in the path from root to a leaf.
- Source instance $\overline{\mathbf{x}}$ is in white class.
- The counterfactual instance subject to being in dark grey class is $\mathbf{x}^*$, which is closest to $\overline{\mathbf{x}}$.

Counterfactual explanations in decision trees

Original problem:

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

Theorem

Problem (1) *is equivalent to:*

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

- Solving problem (1) is equivalent to solving it within each leaf's region and then picking the leaf with the best solution.

Counterfactual explanations in oblique trees

In an oblique decision tree each leaf region is an polytope defined by intersection of arbitrary hyperplanes found in the path from root to leaf.

- For each target leaf the problem becomes:

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

- $\mathbf{h}_i(\mathbf{x})$ is the set of hyperplanes that represents decision rule of the nodes in the path from root to leaf i.
- $\mathbf{h}_i(\mathbf{x})$ forms set of linear constraints.
- If E is ℓ_2 or ℓ_1 distance, then the problem becomes QP or LP, which can be solved very effectively.

Counterfactual explanations in axis-aligned trees

In an axis-aligned decision tree each leaf region is an axis-aligned boxes defined by intersection of axis-aligned hyperplanes found in the path from root to leaf.

Theorem

In problem (1)*, assume that each constraint depends on a single element of* $\mathbf{x}$ *(not necessarily the same) and that the objective function is separable, i.e.,* $E(\mathbf{x}; \overline{\mathbf{x}}) = \sum_{d=1}^{D} E_d(x_d; \overline{x}_d)$*. Then the problem separates over the variables* $x_1, \ldots, x_D$*.*

- This applies to axis-aligned trees because each of the constraints $\mathbf{h}_i(\mathbf{x}) \geq \mathbf{0}$ in the path from the root to leaf i involve a single feature of $\mathbf{x}$ (they are bound constraints).
- This means that, in axis-aligned tree within each leaf, we can solve for each x_d independently, by minimizing $E_d(x_d; \overline{x}_d)$ subject to the constraints on x_d.

Counterfactual explanations in axis-aligned trees cont.

Theorem

Consider the scalar constrained optimization problem, where the bounds can take the values $l_d = -\infty$ and $u_d = \infty$:

$$\min_{x_d \in \mathbb{R}} E_d(x_d; \overline{x}_d) \quad \text{s.t.} \quad l_d \leq x_d \leq u_d.$$

Assume E_d is convex on x_d and satisfies $E_d(\overline{x}_d; \overline{x}_d) = 0$ and $E_d(x_d; \overline{x}_d) \geq 0$ $\forall x_d \in \mathbb{R}$. Then x_d^*, defined as the median of $\overline{x}_d$, l_d and u_d, is a global minimizer of the problem:

$$x_d^* = \text{median}(\overline{x}_d, l_d, u_d) = \begin{cases} l_d, & \overline{x}_d < l_d \\ u_d, & \overline{x}_d > u_d \\ \overline{x}_d, & \text{otherwise} \end{cases}.$$

- This makes solving the counterfactual explanation problem exceedingly fast for axis-aligned trees.

Some useful distance functions

Different distance functions are useful for different applications:

- ℓ_2 distance: $E(\mathbf{x}; \overline{\mathbf{x}}) = \|\mathbf{x} - \overline{\mathbf{x}}\|_2^2$.
- ℓ_1 distance: $E(\mathbf{x}; \overline{\mathbf{x}}) = \|\mathbf{x} - \overline{\mathbf{x}}\|_1$.
- General quadratic distance:
  $E(\mathbf{x}; \overline{\mathbf{x}}) = \boldsymbol{\delta}^T \mathbf{Q} \boldsymbol{\delta} = \sum_{d,e=1}^{D} q_{de}\, \delta_d\, \delta_e$,
  where $\boldsymbol{\delta} = \mathbf{x} - \overline{\mathbf{x}}$ and $\mathbf{Q}$ is the cost matrix.
- Combinations of all the above, such as:
  $E(\mathbf{x}; \overline{\mathbf{x}}) = \|\mathbf{x} - \overline{\mathbf{x}}\|_1 + \lambda \|\mathbf{x} - \overline{\mathbf{x}}\|_2^2$.

The ℓ_1 or ℓ_2 distances should be weighted (or equivalently each feature should be normalized).

Some useful constraints

- Limit the upper and lower value of individual features: $x_d \geq$ something or $x_d \leq$ something.
- Keep some features unchanged: $x_d = \overline{x}_d$.
- We can make counterfactual constrained to be within a set of instances, for example the training set: $\mathbf{x} \in \mathbb{R}^D \Rightarrow \mathbf{x} \in$ training set.

Categorical variables

- For training:
  - Convert each categorical variable with C categories as one hot encoding of size C.
- For counterfactual problem formulation:
  - Add following constraints: $\mathbf{1}^T\mathbf{x} = 1$, where $x_1, \ldots, x_C \in \{0, 1\}$.
- Counterfactual problem formulation then becomes a mixed integer optimization problem. This can be solved exactly (for small C) or approximately (for large C) using CPLEX or Gurobi.
  In practice categorical variables and number of categories are small.

Extension of the basic problem

Basic problem:

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

- Instead of optimal return a diverse set of solutions.
  - Return the solutions of all the leaves with the target class label sorted in increasing cost E.
- Target classes need not be a single class, but a set of classes.
  - Replace y with a (nonempty) subset of classes $\mathcal{Y} \subset \{1,\dots,K\} \setminus \overline{y}$.
- Instead of a hard constraint $T(\mathbf{x}) = y$, use per class cost.
  - Change original cost function to $\min_{\mathbf{x}} E(\mathbf{x};\overline{\mathbf{x}}) + L(T(\mathbf{x}))$, where $L(y) \geq 0$ is the cost for class $y \in \{1,\dots,K\}$.
- Keep counterfactual away from the boundary
  - Replace $\mathbf{h}_i(\mathbf{x}) \geq \mathbf{0}$ with $\mathbf{h}_i(\mathbf{x}) \geq \epsilon$, where $\epsilon \geq 0$.

All of these can be solved with the same approach.

Illustrative example

Binary classification { <\$50k , ≥\$50k }. $\overline{\mathbf{x}}$ is classified as <\$50k (using a pre-trained tree), and target class is ≥\$50k.

Feature	$\overline{x}$, source instance			
age	25			
workclass	Private			
education	11th			
marital -status	Never -married			
occupation	Machine-op -inspect			
relationship	Own-child			
race	Black			
sex	Male			
capital-gain	0			
capital-loss	0			
hours- per-week	40			
native -country	United-States			
income	<\$50k			

Illustrative example

Binary classification { <\$50k , ≥\$50k }. $\overline{\mathbf{x}}$ is classified as <\$50k (using a pre-trained tree), and target class is ≥\$50k.

Feature	$\overline{x}$, source instance	$\mathbf{x}_1^*$, no constraints		
age	25	=		
workclass	Private	=		
education	11th	=		
marital -status	Never -married	=		
occupation	Machine-op -inspect	=		
relationship	Own-child	=		
race	Black	Asian-Pac -Islander		
sex	Male	=		
capital-gain	0	=		
capital-loss	0	1		
hours- per-week	40	=		
native -country	United-States	Peru		
income	<\$50k	≥\$50K		

Illustrative example

Binary classification { $<$\$50k , $\geq$\$50k }. $\overline{\mathbf{x}}$ is classified as $<$\$50k (using a pre-trained tree), and target class is $\geq$\$50k.

Feature	$\overline{x}$, source instance	$\mathbf{x}_1^*$, no constraints	$\mathbf{x}_2^*$, some constraints
age	25	=	=
workclass	Private	=	=
education	11th	=	Assoc-voc
marital -status	Never -married	=	Married-AF-spouse
occupation	Machine-op -inspect	=	=
relationship	Own-child	=	=
race	Black	Asian-Pac -Islander	=
sex	Male	=	=
capital-gain	0	=	=
capital-loss	0	1	=
hours-per-week	40	=	39
native -country	United-States	Peru	=
income	$<$\$50k	$\geq$\$50K	$\geq$\$50K

Illustrative example

Binary classification { <\$50k , ≥\$50k }. $\overline{\mathbf{x}}$ is classified as <\$50k (using a pre-trained tree), and target class is ≥\$50k.

Feature	$\overline{x}$, source instance	$\mathbf{x}_1^*$, no constraints	$\mathbf{x}_2^*$, some constraints	$\mathbf{x}_3^*$, more constraints
age	25	=	=	=
workclass	Private	=	=	Federal -gov
education	11th	=	Assoc-voc	=
marital -status	Never -married	=	Married-AF-spouse	=
occupation	Machine-op -inspect	=	=	Armed -Forces
relationship	Own-child	=	=	=
race	Black	Asian-Pac -Islander	=	=
sex	Male	=	=	=
capital-gain	0	=	=	1
capital-loss	0	1	=	3
hours-per-week	40	=	39	=
native -country	United-States	Peru	=	=
income	<\$50k	≥\$50K	≥\$50K	≥\$50K

Experiment setup

- Axis-aligned trees are trained using CART implemented in Scikit-learn.
- Oblique trees are trained using the Tree Alternating Optimization (TAO) algorithm, implemented in Python.
- Datasets used:

Dataset	D	Feature type	K
MNIST	784	Continuous	10
Adult	102	Continuous and categorical	2
Breast-Cancer	9	Continuous	2
Spambase	57	Continuous	2
Letter	16	Continuous	26
German Credit	61	Continuous and categorical	2

Experimental results in oblique trees

	% c	Our algorithm, $\mathbf{x} \in \mathbb{R}^D$				Our algorithm, $\mathbf{x} \in$ training & test set		
		Runtime (ms)	ℓ_2	ms	ℓ_1	Runtime (ms)	ℓ_2	% feas
MNIST	0	40	0.63±0.48	580	2.85±1.40	40	53.50±17.24	100
	9	40	0.63±0.48	580	2.85±1.40	40	53.50±17.24	100
	47	40	9.28±6.70	550	12.80±7.89	—	—	0
Adult	0	710	2.40±0.83	600	2.40±0.83	70	3.4e4±1.2e5	100
	7	810	2.45±0.86	590	2.40±0.83	4300	1.6e7±5.2e7	100
	14	780	2.49±0.97	570	2.50±0.97	2700	1.9e7±5.8e7	100
Spambase	0	7	0.002±0.01	12	0.060±0.06	0	0.060±0.06	100
	17	6	0.002±0.01	12	0.060±0.06	10	0.070±0.09	32
	53	7	0.003±0.01	12	0.070±0.06	20	0.020±0.01	17
Letter	0	18	0.02±0.01	20	0.31±0.12	0	0.19±0.09	100
	25	17	0.03±0.02	20	0.31±0.12	0	0.36±0.19	19
	62	16	0.20±0.84	190	0.55±0.47	—	—	0
Credit	0	50	3.97±3.50	80	2.70±1.20	0	3.0e3±7.3e3	100
	7	50	4.00±3.50	80	2.70±1.20	—	—	0
	16	50	4.25±5.80	80	2.80±1.30	—	—	0

Experimental results in axis-aligned trees

	% c	Our algorithm, $\mathbf{x} \in \mathbb{R}^D$			
		Runtime (ms)	ℓ_2	ms	ℓ_1
MNIST	0	110	0.04±0.11	110	0.16±0.26
	9	110	0.04±0.11	110	0.16±0.26
	47	120	5.83±3.27	120	13.16±3.65
Adult	0	130	2.05 ±0.31	130	2.05±0.31
	7	130	2.07±0.34	130	2.07±0.34
	14	140	26.10±14.97	140	2.85±4.54
Spambase	0	30	1.7e—5±0.0	30	0.003±0.002
	17	30	1.7e—5±0.0	30	0.004±0.002
	53	40	4.5e—5±0.0	40	0.005±0.004
Letter	0	50	0.014±0.01	50	0.16±0.08
	25	40	0.016±0.02	50	0.17±0.09
	62	40	0.058±0.05	60	0.28±0.02
Credit	0	40	2.60±1.01	40	2.60±1.01
	7	40	2.60±1.01	40	2.60±1.01
	16	40	26.50±50.2	40	4.87±4.64

Solving problem in axis-aligned trees is fast, but in some cases it is slow as the axis-aligned trees are big and contains large number of target leaves compare to oblique trees.

Conclusion

- Classification trees are important, particularly in applications where interpretability is desirable, such as business, law, and medicine.
- Oblique decision trees, trained by the TAO algorithm, can be surprisingly accurate.
- The counterfactual explanation problem for classification trees (axis-aligned and oblique) is nonconvex and nondifferentiable but can be solved exactly and efficiently.
- Proposed approach can handle several useful distance function and linear constraints (equality and inequality); and is applicable to both continuous and categorical variables.
- Fast enough for interactive use.

Thank You !