Even-if Explanations: Formal Foundations, Priorities and Complexity

Gianvincenzo Alfano, Sergio Greco, Domenico Mandaglio, Francesco Parisi, Reza Shahbazian, Irina Trubitsyna

Department of Informatics, Modeling, Electronics and System Engineering, University of Calabria, ITALY

{g.alfano, greco, fparisi, i.trubitsyna, reza.shahbazian}@dimes.unical.it

Università della Calabria
DIPARTIMENTO DI INGEGNERIA INFORMATICA, MODELLISTICA, ELETTRONICA E SISTEMISTICA
DIMES

FAIR Future Artificial Intelligence Research

SERICS

Local Post-Hoc Explanations

- The term *local* refers to explaining the output of the system for a particular input;
- The term *post-hoc* refers to interpreting the system after it has been trained.

Classification Models

A (binary classification) model is a function:

$$\mathcal{M} : \{0,1\}^n \to \{0,1\}$$

An instance $\mathbf{x}$ is a vector in $\{0,1\}^n$ and represents a possible input for a model. We focused on 3 significant categories of ML models:

- *Free Binary Decision Diagram* (FBDD): BDD where no two nodes on any root-to-leaf path share the same label;
- *Multilayer perceptron (MLP)*: intuitively modeling feed-forward NN with hidden layers;
- *Perceptron*: an MLP with no hidden layers.

Complexity Classes

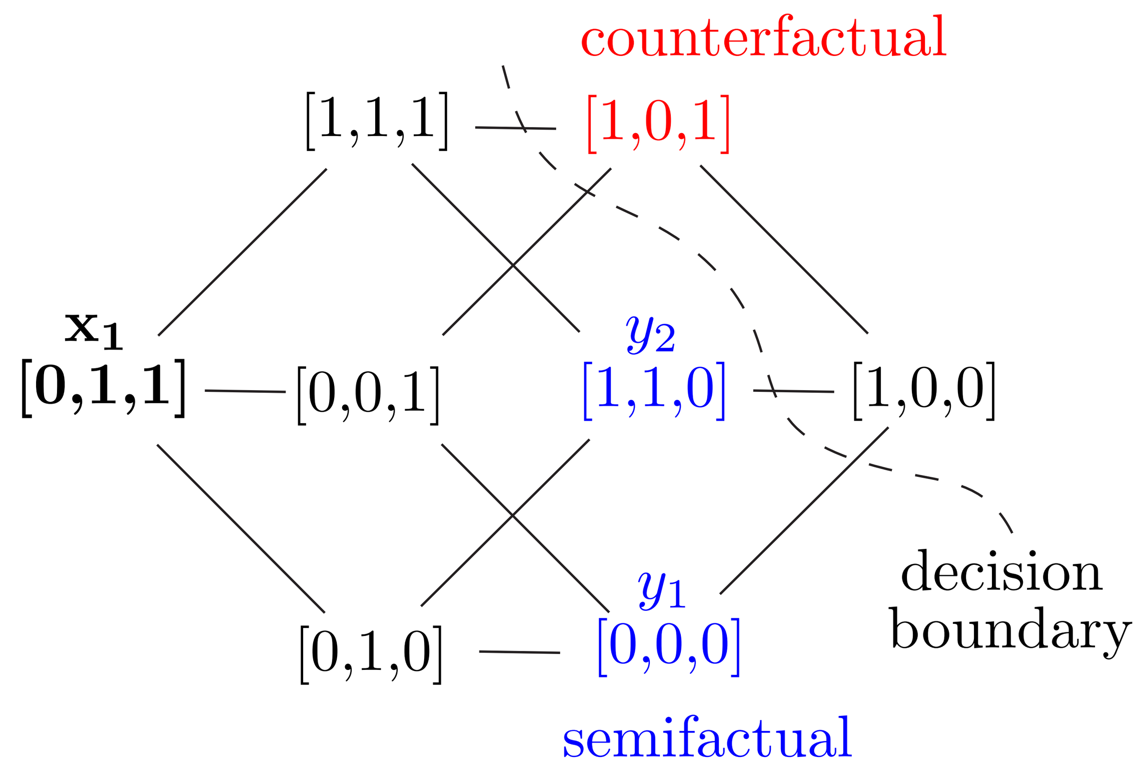
- Decision Problems: boolean functions mapping strings to strings with boolean output;
- (N)P contains the set of decision problems solvable in polynomial time by a (non)deterministic Turing machine;
- coNP is the complexity class containing the complements of problems in NP.

Even-if Explanations

- While significant attention in AI has been given to counterfactual explanations, there has been a limited focus on the equally important and related semifactual 'even if' explanations.
- While counterfactuals explain what changes to the input features of an AI system change the output decision, ***semifactuals*** *show which input feature changes do not change a decision outcome*.

Example:
Binary and linear model $\mathcal{M} : \{0,1\}^3 \to \{0,1\}$ where $\mathcal{M} = step(\mathbf{x} \cdot [-2, 2, 0] + 1)$ the input $\mathbf{x} = [x_1, x_2, x_3]$ denotes an applicant (also called user) defined by means of the following three features:

- f_1 = "part-time job";
- f_2 = "requested (monthly) salary < 5K\$";
- f_3 = "on-site job".



Consider a user $\mathbf{x}_1$ that applies for a full-time and on-site job, and the requested salary is lower than 5K\$ (i.e., $\mathbf{x}_1 = [0,1,1]$), we have that $\mathbf{y}_1 = [0,0,0]$ and $\mathbf{y}_2 = [1,1,0]$ are semifactual of $\mathbf{x}_1$ w.r.t. $\mathcal{M}$ at maximum distance (i.e., 2) from $\mathbf{x}_1$ in terms of number of features changed. Intuitively, $\mathbf{y}_1$ represents the fact that 'the user $\mathbf{x}_1$ will be hired *even if* (s)he had requested for a remote job and the requested salary was greater than or equal to 5K\$', while $\mathbf{y}_2$ represents 'the user $\mathbf{x}_1$ will be hired *even if* (s)he had applied for a remote and part-time job'.

> (**Semifactual**) Given a pre-trained model $\mathcal{M}$ and an instance $\mathbf{x}$, an instance $\mathbf{y}$ is said to be a semifactual of $\mathbf{x}$ iff *i*) $\mathcal{M}(\mathbf{x}) = \mathcal{M}(\mathbf{y})$, and *ii*) there exists no other instance $\mathbf{z} \neq \mathbf{y}$ s.t. $\mathcal{M}(\mathbf{x}) = \mathcal{M}(\mathbf{z})$ and $d(\mathbf{x},\mathbf{z}) > d(\mathbf{x},\mathbf{y})$.

Contribution: We formally introduce the concepts of semifactual over perceptron, FBBD and MLP, intuitively encoding local post-hoc explainable queries within the even-if thinking setting.

Preferences

Contribution: As multiple counterfactuals/semifactuals may exist for each given instance, we introduce a framework that empowers users to prioritize explanations according to their subjective preferences. Thus, the user expresses preferences over features to select the *best* semifactuals.

> (Preference Rule) $\varphi_1 \succ \cdots \succ \varphi_k \leftarrow \varphi_{k+1} \wedge \cdots \wedge \varphi_m$ where $m \geq k \geq 2$, and any $\varphi_i \in \{f_1, \neg f_1, \dots f_n, \neg f_n\}$ is a (feature) literal, with $i \in [1, m]$.

> (**BCMP framework**) A binary classification model with preferences (BCMP) framework is a pair $(\mathcal{M}, \succ)$ where $\mathcal{M}$ is a model and $\succ$ a set of preference rules over features of $\mathcal{M}$. We use $\mathbf{y} \sqsupset \mathbf{z}$ to denote the fact that the explanation $\mathbf{y}$ is strictly preferred to the explanation $\mathbf{z}$ (w.r.t. $\succ$).

Example (cont'd): Suppose that the user $\mathbf{x}_1$ looks for another opportunity and prefers to change feature f_2 rather than f_1 (irrespective of any other change), that is (s)he would prefer to still get hired by changing the salary to be greater than or equal to 5K\$ (obtaining $\mathbf{y}_1$); if this cannot be accomplished, then (s)he prefers to get it by changing the job to part-time (i.e. $\mathbf{y}_2$).

Complexity Results

Contributions: We investigate the complexity of the following interpretability problems related to (best) semifactuals and counterfactuals:

Existence of Counterfactuals

PROBLEM:	MinimumChangeRequired (MCR)
INPUT:	Model $\mathcal{M}$, instance $\mathbf{x}$, and $k \in \mathbb{N}$.
OUTPUT:	Yes, if there exists an instance $\mathbf{y}$ with $d(\mathbf{x},\mathbf{y}) \leq k$ and $\mathcal{M}(\mathbf{x}) \neq \mathcal{M}(\mathbf{y})$; No, otherwise.

Existence of Semifactuals

PROBLEM:	MaximumChangeAllowed (MCA)
INPUT:	Model $\mathcal{M}$, instance $\mathbf{x}$, and $k \in \mathbb{N}$.
OUTPUT:	Yes, if there exists an instance $\mathbf{y}$ with $d(\mathbf{x},\mathbf{y}) \geq k$ and $\mathcal{M}(\mathbf{x}) = \mathcal{M}(\mathbf{y})$; No, otherwise.

Verification of Best Counterfactuals

PROBLEM:	CheckBestMCR (CB-MCR)
INPUT:	BCMP $(\mathcal{M}, \succ)$, instances $\mathbf{x}$, $\mathbf{y}$ with $d(\mathbf{x},\mathbf{y}) = k$, and $\mathcal{M}(\mathbf{x}) \neq \mathcal{M}(\mathbf{y})$.
OUTPUT:	Yes, if there is no $\mathbf{z}$ with $\mathcal{M}(\mathbf{x}) \neq \mathcal{M}(\mathbf{z})$ and either $d(\mathbf{x},\mathbf{z}) \leq k-1$, or $d(\mathbf{x},\mathbf{z})=k$ and $\mathbf{z} \sqsupset \mathbf{y}$; No, otherwise

Verification of Best Semifactuals

PROBLEM:	CheckBestMCA (CB-MCA)
INPUT:	BCMP $(\mathcal{M}, \succ)$, instances $\mathbf{x}$, $\mathbf{y}$ with $d(\mathbf{x},\mathbf{y}) = k$, and $\mathcal{M}(\mathbf{x}) = \mathcal{M}(\mathbf{y})$.
OUTPUT:	Yes if there is no $\mathbf{z}$ with $\mathcal{M}(\mathbf{x}) = \mathcal{M}(\mathbf{z})$ and either $d(\mathbf{x},\mathbf{z}) \geq k+1$ or $d(\mathbf{x},\mathbf{z}) = k$ and $\mathbf{z} \sqsupset \mathbf{y}$; No, otherwise.

- Computing semifactuals under perceptrons and FBDDs is easier than under MLP;
- Computing semifactuals is as hard as computing counterfactuals;
- Perceptrons and FBDDs are strictly more interpretable than MLPs;
- Preferences do not make the existence of counterfactual/semifactual problem harder;
- Preferences do not make the verification problems harder when the BCMP contains a single preference rule with empty body (called linear).

	FBDDs	Perceptrons	MLPs
MCR	PTIME	PTIME	NP-c
MCA	PTIME	PTIME	NP-c
CB-MCR	coNP	coNP	coNP-c
CB-MCA	coNP	coNP	coNP-c
CBL-MCR	PTIME	PTIME	coNP-c
CBL-MCA	PTIME	PTIME	coNP-c

Grey-colored cells refer to existing results. L stands for linear preferences.

Contributions: For BCMP with linear preference, we propose PTIME algorithms for the computation of best counterfactuals/semifactuals under Perceptrons and FBDDs.