Theoretical Basis and Computational Complexity of Semifactual Explanations

Gianvincenzo Alfano[1], Sergio Greco[1], Domenico Mandaglio[1], Francesco Parisi[1], Reza Shahbazian[2], Irina Trubitsyna[1]

[1]Department of Informatics, Modeling, Electronics and System Engineering (DIMES), University of Calabria, Italy
[2]Department of Humanities, University of Palermo, Italy

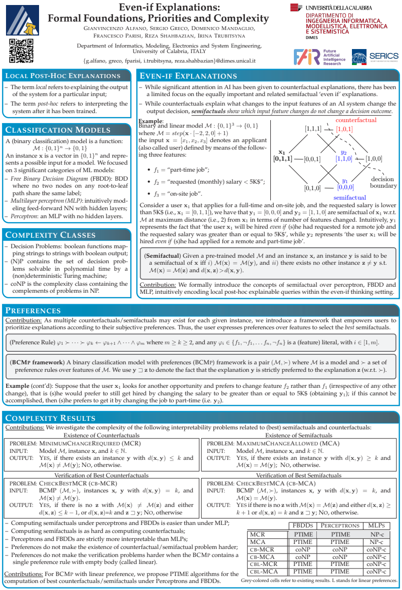
Ital-IA, Workshop AI Responsabile e Affidabile
June 23, 2025
Trieste, Italy

Finanziato dall'Unione europea NextGenerationEU | Ministero dell'Università e della Ricerca | Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA | FAIR Future Artificial Intelligence Research

Reference Publication

This talk is based on the following paper:

*'Even-if Explanations: Formal Foundations, Priorities and Complexity' (**AAAI'25**)*



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:
$\mathcal{M} : \{0,1\}^n \rightarrow \{0,1\}$
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 $\mathcal{M} : \{0,1\}^3 \rightarrow \{0,1\}$
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".

counterfactual
[1,1,1]
[1,0,1]
$\mathbf{x_1}$
[0,1,1]
[0,0,1]
y_2
[1,1,0]
[1,0,0]
y_1
[0,1,0]
[0,0,0]
decision boundary
semifactual

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, FBDD 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.

Outline

Motivation: EXplainable AI (XAI)

- ML models often operate as black boxes, lacking explainability and transparency while supporting decision-making *green-aware* processes
- Several (explanation) methods proposed so far:
  - **Factual** explanations: revealing only the why it can be not sufficient to understand how to change the outcome
  - **Counterfactual** explanations: suggest what should be different in the input instance to change the outcome of an AI system

Example (Counterfactual)

- Assume to have a ML model classifying products as either eco-friendly or non-eco-friendly based on various features (e.g., carbon footprint, recyclability, material sourcing)

if only

the product's packaging had been made from 30% more recycled material

then

the model would have classified it as 'eco-friendly' instead of 'non-eco-friendly

Semifactual

- Limited focus on the equally important semifactual ‘**even if**’ explanations

Example (Semifactual)

- Assume to have a ML model classifying products as either eco-friendly or non-eco-friendly based on various features (e.g., carbon footprint, recyclability, material sourcing)

even if
the product’s packaging had been made from 30% less recycled material
then
the model would have still classified it as ’eco-friendly’

Contributions

- We show that both linear and tree-based models are strictly more interpretable than neural networks under semifactual reasoning;
- We introduce a preference-based framework that enables users to personalize explanations based on their preferences;
- We explore the complexity of several interpretability problems in the proposed preference-based framework and provide algorithms for polynomial cases.

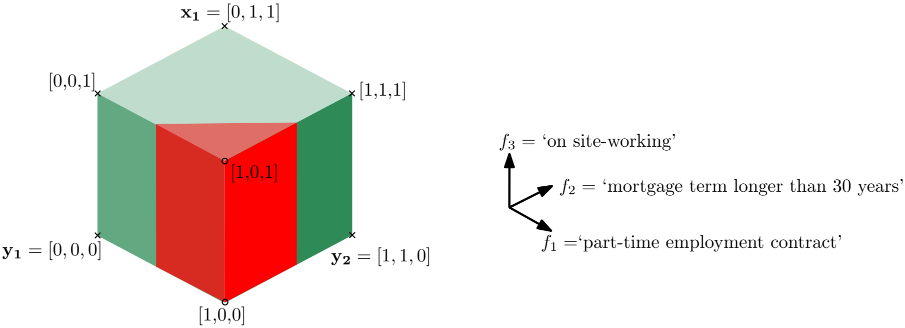
Classification Models

We consider binary classification models $\mathcal{M} : \{0,1\}^n \to \{0,1\}$

1. **Multi Layer Perceptrons** (MLP): feed-forward NN with relu activation functions in all intermediate layers, and the step function as last activation function.
2. **Perceptrons**: Special case of MLP with no hidden layers. Intuitively, it represents an SVM.
3. **Free Binary Decision Diagrams** (FBDD): A binary decision diagram where, for every path from the root to a leaf, no two nodes on that path have the same label.

Perceptron

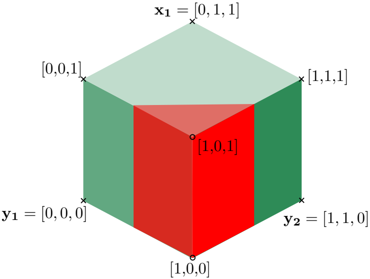
- Perceptron $\mathcal{M}$: $step(\mathbf{x} \cdot [-2, 2, 0] + 1)$ representing a mortgage scenario
- Input : user $x = [x_1, x_2, x_3]$ with features f_1, f_2, and f_3
- Crosses/Circles represent instances where the model outputs 1/0
- User $\mathbf{x_1} = [0, 1, 1]$ works full-time, on-site, and requests a mortgage with duration longer than 30 years, and the loan has been accepted ($\mathcal{M}(\mathbf{x_1}) = 1$)

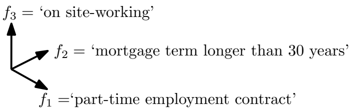


Counterfactual

Definition (Counterfactual [Barcelo et al., NIPS 2020])

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





- Consider again user $\mathbf{x}_1 = [0, 1, 1]$, $\mathbf{y}_3 = [1, 0, 1]$ is its only counterfactual ($d(\mathbf{x}_1, \mathbf{y}_3) = 2$)

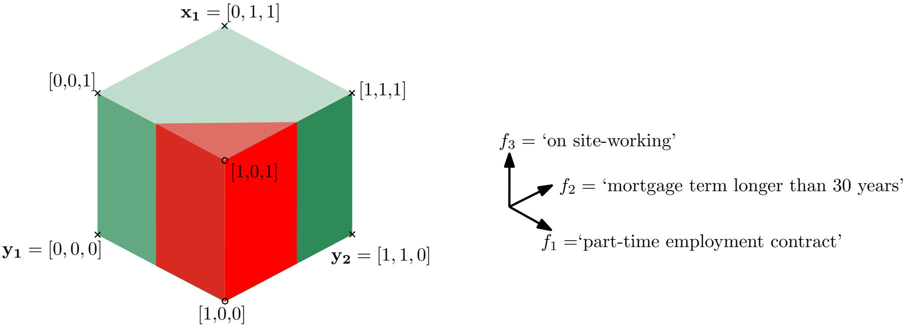
Outline

Formal Definition

Definition (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 $d(\mathbf{x},\mathbf{z})>d(\mathbf{x},\mathbf{y})$.

Formal Definition



- $\mathbf{y}_1 = [0, 0, 0]$ and $\mathbf{y}_2 = [1, 1, 0]$ are the only semifactuals of $\mathbf{x}_1 = [0, 1, 1]$

$\mathbf{y}_1$: **even if** 'the work would have been remote and the mortgage duration would have been < 30 years' **then** 'the user will obtain the mortgage'

Problem and Complexity

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.

Theorem

MCA *is i) in PTIME for FBDDs and perceptrons, and ii) NP-complete for MLPs.*

Take home message 1: Independently of the type of the model, computing semifactuals is as hard as computing counterfactuals.

Take home message 2: Perceptrons and FBDDs are strictly more interpretable than MLPs, in the sense that the complexity of answering explainability queries for models in the first two classes is lower than for those in the latter.

Outline

Preference Rules

- To express preferences over semifactuals and counterfactuals, we introduce a novel approach inspired to that proposed in [Brewka et al., 2003] for ASP
- Preference rules determine a preference ordering $\sqsupseteq$ on explanations, so that the *best* ones are selected.

Example

Consider the Binary Classification Model with Preferences (BCMP) $\langle \mathcal{M}, \kappa \rangle$ with

$$\kappa : f_1 \succ \neg\, f_2 \leftarrow \neg f_3$$

User prefers explanations where, whenever the work is not on-site (i.e., $\neg f_3$ holds), (s)he prefers explanations where the contract is part-time (i.e., f_1 holds) and, if this is not possible, (s)he prefers those where the mortgage duration is not longer than 30 years (i.e., $\neg f_2$)

Preference Rules

- We also focused on restricted version *Linear-BCMP*: a single preference rule with empty body.
- We investigate decision problems (analogous to MCA) in case of (linear) preferences

		FBDDs	Perceptrons	MLPs
(Counterfactual) [Barcelo et al., NIPS 2020]	MCR	PTIME	PTIME	NP-c
(Semifactual)	MCA	PTIME	PTIME	NP-c
(Counterfactual+Preferences)	CBMCR	coNP	coNP	coNP-c
(Semifactual+Preferences)	CBMCA	coNP	coNP	coNP-c
(Counterfactual+ Linear Pref.)	L-CBMCR	PTIME	PTIME	coNP-c
(Semifactual+ Linear Pref.)	L-CBMCA	PTIME	PTIME	coNP-c

Take home message: Linear preferences do not increase the complexity in the case of perceptrons and FBDDs, though they allow expressing user-specific desiderata among queries.

Algorithms

- PTIME algorithms for computing a best semifactual/counterfactual explanation for perceptrons and FBDDs.

Algorithm 1 Computing a (best) semifactual for perceptrons

Input: Perceptron $\mathcal{M} = (\mathbf{W}, b)$, instance $\mathbf{x} \in \{0,1\}^n$, and linear preference $\kappa = f_{p_1} \succ \cdots \succ f_{p_l}$.
Output: A best semifactual $\mathbf{y}$ for $\mathbf{x}$ w.r.t. $\mathcal{M}$ and κ.

Let $\mathbf{s} = [f_1/s_1, \ldots, f_n/s_n]$ where $\forall i \in [1, n]$, $s_i = 2x_i w_i - w_i$ if $\mathcal{M}(\mathbf{x}) = 1$, $w_i - 2x_i w_i$ otherwise;
Let $\mathbf{s}' = [f_{q_1}/s_{q_1}, \ldots, f_{q_n}/s_{q_n}]$ be the sorted version of $\mathbf{s}$ in ascending order of s_i;
$k = \mathsf{max}(\{i \in [0, n] \mid \mathcal{M}(\mathsf{flip}(\mathbf{x}, \mathsf{pos}(\mathbf{s}', i)) = \mathcal{M}(\mathbf{x})\})$;
if $k = 0$ **return** $\mathbf{x}$;
if $k = n$ **return** $[1 - x_1, \ldots, 1 - x_n]$;
$\mathbf{y} = \mathsf{flip}(\mathbf{x}, \mathsf{pos}(\mathbf{s}', k))$;
$\delta = \mathsf{min}(\{i \in [1, l] \mid y_{p_i} = 1\} \cup \{l + 1\})$;
for $i \in [1, \ldots, \delta - 1]$ **do**
  if $y_{p_i} = 1$ **return** $\mathbf{y}$;
  Let $j = q_1$ if $x_{p_i} = y_{p_i}$, $j = q_{k+1}$ otherwise;
  $\mathbf{z} = \mathsf{flip}(\mathbf{y}, \{p_i, j\})$;
  if $\mathcal{M}(\mathbf{x}) = \mathcal{M}(\mathbf{z})$ **return** $\mathbf{z}$;
return $\mathbf{y}$;

Algorithm 2 Computing a (best) semifactual for FBDDs

Input: FBDD $\mathcal{M} = (V, E, \lambda_V, \lambda_E)$ with root t, instance $\mathbf{x} \in \{0,1\}^n$, and linear preference $\kappa = f_{p_1} \succ \cdots \succ f_{p_l}$.
Output: A best semifactual $\mathbf{y}$ for $\mathbf{x}$ w.r.t. $\mathcal{M}$ and κ.

Let $\mathcal{M}' = (V' = V, E' = E, \lambda_{V'} = \lambda_V, \lambda_{E'})$ be a copy of $\mathcal{M}$, where $\lambda_{E'}(u, v) = 1$ if$(x_{\lambda_V(u)} = \lambda_E(u, v))$, 0 otherwise;
Let $\mathcal{N} = \mathsf{subgraph}(\mathcal{M}', \mathcal{M}(\mathbf{x}))$;
Let Π be the set of paths in $\mathcal{N}$ from t to leaf nodes;
for $f_{p_i} \in \{f_{p_1}, \ldots, f_{p_l}\}$ **do**
  if $\exists \pi \in \Pi$ with $\mathbf{y} = \mathsf{build}(\mathbf{x}, \pi)$ and $y_{p_i} = 1$
    return $\mathbf{y}$;
Let π be a path of Π taken non-deterministically;
return $\mathbf{y} = \mathsf{build}(\mathbf{x}, \pi)$;

Outline

Conclusions and Future Work

- We analyzed the complexity of local post-hoc interpretability queries related to semifactuals across three model classes, and introduces a preference-based framework for personalizing semifactual and counterfactual explanations.

Future Works:

- Investigating models dealing with real-number inputs and non-binary discrete features
- Investigating interpretability queries for other ML models (e.g., Graph Neural Networks)

Thank you!
Questions?