Counterfactual explanations

Explainable and Trustworthy AI

Eliana Pastor

Introduction to Counterfactual Explanations

Counterfactual explanations involve **changing some aspects of an input to see how the output changes**, answering "What if...?"

Purpose. Provide insight into model decision-making by illustrating how small changes can lead to different outcomes.

Counterfactuals - Example

Age: 30
Income: 30K
Amount requested: 15K

Loan?

No

What if request = 12K?
Age: 30
Income: 30K
Amount requested: **12K**

Loan?

Yes

If the applicant's request was 12K instead of 15, the loan would be approved.

Counterfactual explanations

Given

- an instance to explain x and its prediction $\mathrm{y} = f(x)$ by model f
- A **predefined output** of interest
  - E.g., probability $y' \neq y$ or different predicted class

A **counterfactual explanation** of a prediction describes the **smallest change to the feature** values that **changes the prediction to a predefined output**.

A counterfactuals is an **example-based** explanations as it is a new instance.

- We have a new instance x’ that, starting from x, has some of the feature changed.

Why Counterfactual Explanations?

- **Interpretability**. Help users understand the decision boundary of the model, why a prediction is made.
  - Generally simple to understand as they involve the change of few features
- **Trust**. Build user trust by showing how decisions can be altered.
  - Provide insights also when users should contest the decision (e.g., to change outcome the user should change a sensitive and protected attributed)
- **Actionability**. Offer actionable insights on how to change outcomes.

Properties and desiderata of counterfactual explanations

- **Closeness to the predefined output.**
  - A counterfactual instance should produce the predefined prediction as closely as possible
- **Closeness to the input.**
  - The features of a counterfactual should be as similar as possible to the original instance
- **Sparsity.**
  - The counterfactual changes only few features.
- **Diversity and multiple explanations.**
  - We should generate multiple counterfactual explanations that are different from each other
    - So that we can identify which alterations are more suitable/actionable to get a different outcome
- **Feasibility and Actionability.**
  - A counterfactual instance should have feature values that are possible/likely
    - E.g., height 1.90 and weight 10 kgs
    - E.g., decreasing age is impossible, unactionable

Molnar, Christoph. *Interpretable machine learning*

Wachter et al.

Among first algorithms for generating counterfactual explanations

Target satisfying the two properties of **closeness to the predefined output** And **closeness to the input.**

Given:

- model f and the training set
- an instance x and an outcome y
- a desired outcome y'

The approach targets to find a counterfactual $\boldsymbol{x}'$ as **close to** the original instance $\boldsymbol{x}$ but **with** $\boldsymbol{f}(\boldsymbol{x}') = \boldsymbol{y}'$

Counterfactual explanations without opening the black box: Automated decisions and the GDPR (Wachter et al., 2017)

Wachter et al.

The approach identifyies x' by minimizing the following loss function

$$L(x, x', y', \lambda) = \lambda \cdot (f(x') - y')^2 + d(x, x')$$

closeness to the predefined output **closeness to the input**

where d is a distance function and λ is a regularization parameter that balances the distance in prediction against the distance in feature values.

Larger λ : prefer counterfactuals very close to y'.
Smaller λ: prefer counterfactuals very close to the original instance x

Counterfactual explanations without opening the black box: Automated decisions and the GDPR (Wachter et al., 2017)

Wachter et al.

Closeness to the predefined output.

$$(f(x') - y')^2$$

Quadratic distance between the model prediction for the counterfactual x' and the desired outcome y'

Closeness to the input.

Distance d between the instance x and the counterfactual x', with

$$d(x,x') = \sum_{j=1}^{p} \frac{|x_j - x'_j|}{MAD_j}$$

where

$MAD_k = median_{i\in\{1,\ldots,n\}}\left(\left|x_{i,k} - median_{i\in\{1,\ldots,n\}}\left(x_{l,k}\right)\right|\right)$ for feature k.

The feature-wise distance is scaled by the inverse of the median absolute deviation of feature j over the dataset

- Avoid to have different impacts for features with different variations (e.g., age and income)

Counterfactual explanations without opening the black box: Automated decisions and the GDPR (Wachter et al., 2017)

Wachter et al. - Definition of λ

Since λ may be difficult to select, the approach proposes instead to select a tolerance ϵ for how far from y' the prediction of the counterfactual x' is allowed to be:

$$|f(x') - y'| \leq \epsilon$$

The loss function is minimized for x' while increasing λ until a sufficiently close (i.e., respect to the tolerance ϵ) solution is found:

$$arg \min_{x'} \max_{\lambda} L(x, x', y', \lambda)$$

Wachter et al. - Algorithm

1. Given an instance x to be explained, the desired outcome y’, a tolerance ϵ and a (low) initial value for λ.
2. Sample a random instance as initial counterfactual.
3. Optimize the loss with the initially sampled counterfactual as starting point.
4. While $|f(x') - y'| > \epsilon$:
    1. Increase λ.
    2. Optimize the loss with the current counterfactual as starting point.
    3. Return the counterfactual that minimizes the loss.
5. Repeat steps 2-4 and return the list of counterfactuals or the one that minimizes the loss.

Source: Molnar, Christoph. *Interpretable machine learning. Chapter 9.3*

DICE

- **Diverse Counterfactual Explanations (DiCE)**
- Extends Wachter et al. to consider also the properties of **Diversity** and **Feasibility**
- Goal to generate a set of counterfactual example $\{c_1, c_2, \ldots, c_k\}$ such that lead to a different decision than x, y'

Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations (Mothilal et al., 2019)

DICE – Terms in the loss function

Closeness to the input.

The set of counterfactual examples should be closed to the original instance

$$proximity = -\frac{1}{k}\sum_{i=1}^{k} dist(x_i', x)$$

DICE – Terms in the loss function

Closeness to the predefined output.

Minimize the distance between the counterfactual x' and the target outcome y'

$$\frac{1}{k}\sum_{i=1}^{k} yloss(f(x_i'), y')$$

DICE – Terms in the loss function

Diversity.

Via Determinantal Point Processes

$$dpp\_diversity = \det(K)$$

Where $K_{i,j} = \frac{1}{1+dist(x_i', x_j')}$ and $dist(x_i', x_j')$ = distance between two counterfactuals

We want to **penalize similar counterfactuals**

The determinant of a symmetric matrix with large values in [0,1] (i.e., similar counterfactual = small distance = large $K_{i,j}$) will be small (close to 0).

DICE – Addional constraints

Feasibility.

- The users can provide constraints on the feature manipulation
  - the feature X cannot increase beyond Y (e.g., income not beyond 1M)
  - specify the variables that can be changed (e.g., age)

DICE – Post-processing constrainsts

Sparsity.

This property considers the features to change to produce the counterfactuals.

The paper do not include this property in the loss function but operate on counterfactuals in a post-processing manner.

DICE – Final loss

The set of counterfactual is defined by minimizing the following loss function

$$X' = \underset{x'_1,\ldots,x'_k}{\mathrm{argmin}} \frac{1}{k}\sum_{i=1}^{k} yloss(f(x'_i), y') + \frac{\lambda_1}{k}\sum_{i=1}^{k} dist(x'_i, x) - \lambda_2\, dpp\_diversity(x'_1, \ldots, x'_k)$$

Where X' is the set of k counterfactual, and λ_1 and λ_2 are regularization terms

Counterfactual Generation for NLP

Polyjuice is a tool to generate counterfactuals for NLP

Purpose: explaining but also evaluating, and improving model

- **Diverse Counterfactual Generation**
  - It generates a set diverse of counterfactuals by making minimal changes to the original text.
  - Changes involve altering words, phrases, or even larger textual structures while preserving grammatical correctness and naturalness.

- **Multiple Types of Transformations**
  - Various textual transformations, including synonym replacement, paraphrasing, insertion, deletion

Wu, T., Ribeiro, M. T., Heer, J., & Weld, D. S. (2021). Polyjuice: Generating Counterfactuals for Explaining, Evaluating, and Improving Models. ACL

Polyjuice - Desiderata

It accounts for the following desiderata

- **Closeness to the input.**
- **Diversity and multiple explanations.**
    - **Multiple perturbation types**
- **Feasibility.**
    - **Fluency/naturalness**

+

Control perturbation

Polyjuice – Desiderata - How

- **Closeness to the input.**
  - Fine-tune GPT-2 on close sentence pairs
  - Original text as contenxt, perturbation of the context
    - e.g., it is great for kids, it is not great for children
- **Fluency & diversity**
  - Provided by GPT-2 itself
  - Fine-tuning of GPT-2 for multiple datasets and diverse perturbations

- **Control perturbation and generation process**
  - **Via prompting**

  Example of perturbations
  - Negation

  It is great for kids. <|perturb|> [negation] (pos) --> 'It is **not** great for **children**', 'It is great for **no one**.' (neg)
  - Replacing

  It is great for kids . <|perturb|> [lexical] (pos) --> 'It is **bad** for kids' (neg)

Evaluating counterfactuals

Counterfactual explanations

Evaluating counterfactuals

Validity.

The fraction of examples returned by a method that are actually counterfactuals.

It measures the fraction of counterfactuals that actually have the desider class label

$$CF - validity = \frac{|\{x_i' \in X' s.t. f(x_i') = y'\}|}{k}$$

Proximity.

Mean of feature-wise distances between the CF example and the original input.

$$CF - proximity = \frac{1}{k}\sum_{i=1}^{k} dist(x_i', x)$$

Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations (Mothilal et al., 2019)

Evaluating counterfactuals

Sparsity.

Count the number of features that are different, i.e., the number of changes between the original input and a generated counterfactual.

$$CF - sparisity \ = \frac{1}{k}\sum_{i=1}^{k}\sum_{l=1}^{d} 1_{[x_i'^l \neq x^l]}$$

Where d is the numbe of features

Diversity.

feature-wise distances between each pair of CF examples. Compute as mean of the distances

$$\frac{1}{\#pairs}\sum_{i=1}^{k-1}\sum_{j=i+1k} dist\left(x_i', x_j'\right)$$

Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations (Mothilal et al., 2019)

Cognitive metrics: Intuitiveness, comprehensibility

- Evaluated with user study

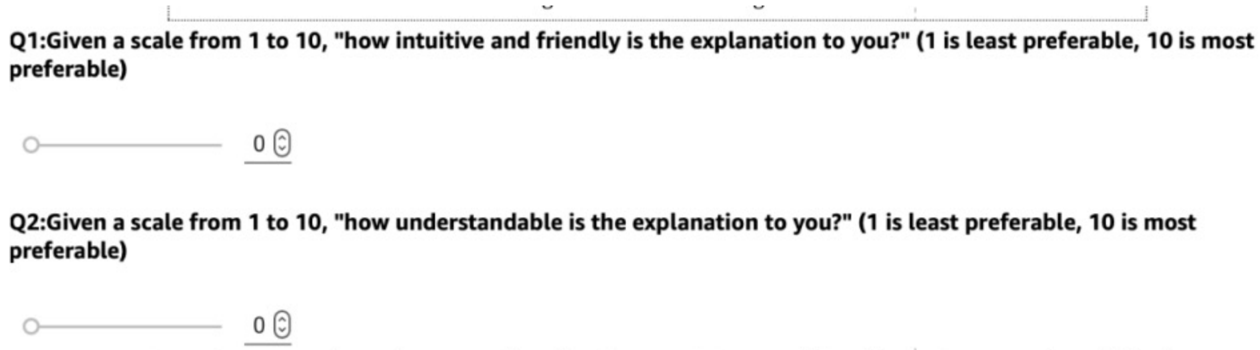
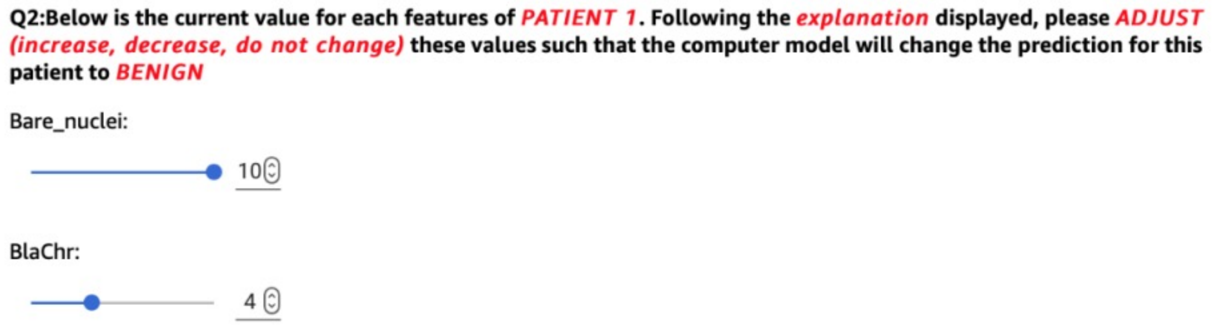


Figure Sup2: Interface of User-study 1 (GRACE: Intuitiveness, friendliness & comprehensibility).



T. Le, S. Wang and D. Lee, GRACE: Generating Concise and Informative Contrastive Sample to Explain Neural Network Model's Prediction, KDD, 2020

Advantages

- Easy to interpret
  - Changing the feature would change the prediction
- Form of explanations
  - Explanation by example
  - Minimal change in the features
- Depending on the generation method, we do not require accessing the training data
- Generally easy to implement as often it is a minimization process of a loss function

Disadvantages

- Feasibility
  - Unrealistic Changes. Counterfactual explanations might suggest changes that are not feasible or realistic, e.g., change age
  - Actionability. Suggested changes might not be actionable for the individual, e.g., increase salary
- Ambiguity
  - Multiple Possible Explanations. There can be many possible counterfactual explanations for a given decision. Which one is the best?
- Local Validity.
  - Counterfactual explanations are local and specific to the individual instance.
  - Lack of Generalizability. Changes suggested by counterfactuals for one instance might not be applicable to other
- Some users may prefer other form of explanations

References

- Molnar, Christoph. *Interpretable machine learning* https://christophm.github.io/interpretable-ml-book/. Chapter 9.3 [Recommended]
- Counterfactual Explanations in Explainable AI: A Tutorial - https://sites.google.com/view/kdd-2021-counterfactual - KDD 2021 Tutorial [Recommended]
- T. Le, S. Wang and D. Lee, GRACE: Generating Concise and Informative Contrastive Sample to Explain Neural Network Model's Prediction, KDD, 2020
- Wu, T., Ribeiro, M. T., Heer, J., & Weld, D. S. (2021). Polyjuice: Generating Counterfactuals for Explaining, Evaluating, and Improving Models. ACL