Counterfactual explanation

Chenyuan Zhao

Technische Universität München

Abstract. Used in classification problems. Counterfactual explanation treat the ML model as a black box, omitting the inner algorithm, instead, focus on the correlation between input and output. This explanation asks how the classification result will change if we alter the input.

Keywords: First keyword · Second keyword · Another keyword.

1 Introduction

People tend to ask themselves "what-if" questions to dream about another better results when unfortunate happens. These "what-if" arguments are usually a (slightly) different world in which a slight change in action may amend the current outcome. Counterfactual explanations (CF) are mostly applied for binary automated decision in social critical domains (such as disease diagnose, loan approval, school admission. . .), to audit possible bias and artifacts. [4] has pointed out three usages of a explanation of machine decisions: (a.) to understand the decision, (b.) to contest a decision, and (c.) to obtain instructions for better outcomes in the future. Counterfactual explanation, though ignoring the working principle of the model, satisfies all these demands.

Without "opening the black box", a CF algorithm generates a positive classified user-case based on a originally negative one (or vice versa). Providing that the algorithm suggests a higher GPA score or more deposit in bank account, the user understands the reason of the rejection, but also knows the measure to adopt for the desired outcome. However, if the algorithm suggests a change in the race or gender, which implies that the decision is made through a discrimination factor, the user is expected to contest the unfair decision.

In practical, there are still several features to taken into consideration. Sparsity, to ensure an actionable advice for the user, the generation should prefer as few changes in items as possible, changing one item vastly is better than changing several slightly. Diversity, to provide a list of choices, the algorithm should offer multiple CF cases varying in the changed item and in different extend. And finally, interpretability, the suggested CF case should be a possible case in the real life. Later, it will be leveraged that sparsity is contradictable with diversity and interpretability, optimizing sparsity usually causes downgrade in other 2 features.

2 Generation of a counterfactual explanation

2.1 by loss function

CF explanation doesn't require to understand the internal work principle of a model, instead focusing on which kind of minimal perturbation in the data subject will leads to a different classification result. Therefore, the search of a an appropriate CF example is boiled down to an optimizing problem with at least two requirements: (1.) classified as the counter class, and (2.) still similar to the original data with only necessary modifications. [4] has proposed the following formulation, which becomes the basis of the following research:

$$\mathbf{c} = \arg\min_{\mathbf{c}} trgtloss(f(\mathbf{c}), y) + d(\mathbf{x{,}c}) \tag{1}$$

where $\mathbf{x}$ is the original data, y is the target data class, $\mathbf{c}$ is the generated counterfactual data, and f is the model so that f($\mathbf{c}$) is the new prediction of the model, i.e. the new class. The first part of the formula encourages a different class, while the second part penalizes large distances away from the original data.

To facilitate equation 1 one need to define it in detail, namely how to define "loss" and "distance", which has a significant impact on final results. Another main contribution of [4] is to define distance as L_1 norm divided by MAD (median absolute deviation):

$$dist = \sum_{k=1}^{K} \frac{|\mathbf{x}\text{-}\mathbf{c}|}{MAD_k} \tag{2}$$

where MAD is defined for every feature k over the whole points set P:

$$MAD_k = median_{i \in P}(|X_{i,k} - median_{j \in P}(X_{j,k})|) \tag{3}$$

The L_1 norm in equation 2 prone to generate zero entries, which ensures a sparse optimizing result. Normalising the norm is important as well, otherwise big range data would have heavy weights. Here MAD turns out to outperform standard deviation, because it cooperates with L_1 norm better, and generates a even more sparse result.

Sometimes data may contain categorical features (e.g. occupation, gender...), it is counter-intuitive to define "distance" for these features. A simple matching distance is used here, where distance 1 is assigned to value changes and 0 if the value remains unchanged.

$$dist\_cat(\mathbf{c{,}x}) = \sum_{k=1}^{K_{cat}} \mathbb{I}(c^k \neq x^k) \tag{4}$$

Combining equation 2 and 4 weighted by the number of continuous and categorical features, the most common used distance metric is shown as following:

$$d(\mathbf{x{,}c}) = \frac{1}{K_{con}} \sum_{k=1}^{K_{con}} \frac{|\mathbf{x}\text{-}\mathbf{c}|}{MAD_k} + \frac{1}{K_{cat}} \sum_{k=1}^{K_{cat}} \mathbb{I}(\mathbf{c}^k \neq \mathbf{x}^k) \tag{5}$$

Choosing an appropriate loss term for the target class is of equal importance. Without loss of generality, in this article we choose 0 as the original class label, and 1 for the desired label. Note that the prediction of the model f($\mathbf{c}$) is a continuous value between 0 and 1 for binary classification. One of the choices of the target loss term is the L_2 norm $(f(\mathbf{c}) - y)^2$. [2] argues that a valid counterfactual suffices if it bypasses the classification threshold of the model (typically 0.5), but the L_2 norm always prefer the extreme value 1. The underlying issue is that counterfactuals, unless predicted exactly as 1, still receives penalty even if it is classified correctly. A better choice is the hinge-loss:

$$hinge\_trgtloss = \max(0, 1 - logit(f(\mathbf{c}))) \tag{6}$$

where *logit(f(* $\mathbf{c}$ *))* is the final activation before entering a sigmoid/softmax output

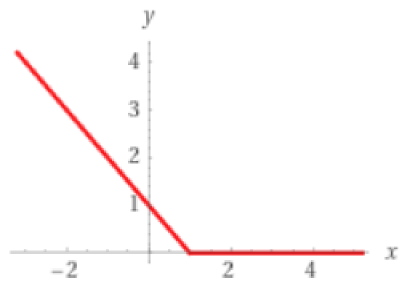

Fig. 1. the hinge loss penalize wrong classification heavily, correct one near the boundary slightly and has no effect above a certain threshold

layer. This loss function, however, requires internal logits of the model, therefore not available under a "black-box" situation.

So far the equations are derived for a binary classification problem, where a "why A not ¬A" or "why A not B" question is raised. For n-ary classification, where the prediction output is a n-length vector, [3] mentioned the following formula:

$$predloss = \max([f(\mathbf{c})]_{i=i_0} - \max_{i \neq i_0}[f(\mathbf{c})]_i), -\kappa) \tag{7}$$

where $i_0 = \arg\max f(\mathbf{x})$ is the original data label, and $[f(\mathbf{c})]_i$ is the i-th class prediction probability. This loss function gives a negative value if the predicted class is different than the original class, and $-\kappa < 0$ caps the negative value.

add a diversity term As mentioned in the introduction, providing a diverse list of counterfactuals to choose from could be more helpful rather than one single "most feasible" solution. A valuable CF algorithm is able to generate multiple counter instances that are distinct from each other, but in a moderate way. The counterfactuals should still proximate the original input, rather than changing numerous features, or changing too fiercely. This is realized by adding a specific diversity loss term to the loss function. [2] proposed to use the determinant of a kernel matrix as the diversity metric for n counterfactuals:

$$dpp_d iversity(\mathbf{c_1}, \ldots, \mathbf{c_n}) = \det(\mathbf{K}),\ \mathbf{K}_{i,j} = \frac{1}{1 + dist(\mathbf{c_i}, \mathbf{c_j})} \tag{8}$$

This matrix is named after DPP (determinantal point processes), which is a widely adopted method in diversity constraint problems (e.g. diverse videos recommendation). An interesting mathematical property of this metric is its symmetric form with an all-one diagonal.

interpretability term During the search of a CF example, it is equally important to demonstrate that the example is representable for the counter class, otherwise it is neither informative nor reasonable. For example, a user may want to change his occupation for better credits, but change the occupation to "professor" with educational background unchanged as "middle school" does not seem like a plausible answer. One trivial solution, as mentioned in the conclusion of [1], is to abandon data generation, and only choose a closest case with the opposite label from a data bank.

To handle with this issue, [3] proposed a prototype loss term, to measure the distance from the generated CF to other CF classes in latent layer. For each CF class i, the algorithm firstly picks out the N nearest neighbours of the input that are classified as i, then feeds them through an encoder and averages the output as the prototype of this CF class.

$$proto_i = \frac{1}{N} \sum_{n=1}^{N} \mathbf{ENC}(\mathbf{x_n^i}) \tag{9}$$

Among all prototypes, the closest one to the input is chosen as the "guidance" for optimization:

$$j = \arg\min_{i \neq f(\mathbf{x})} ||\mathbf{ENC}(\mathbf{x}) - proto_i||_2 \tag{10}$$

The prototype loss term is defined as following:

$$prtloss = ||\mathbf{ENC}(\mathbf{c}) - proto_j||_2^2 \tag{11}$$

With the guidance of prototype, the perturbation is oriented to one chosen CF class rather than random search. The encoder used here could be any external known model, therefore requiring no internal information of the "black-box" model in question.

References

1. Bertossi, L.: An asp-based approach to counterfactual explanations for classification. arXiv preprint arXiv:2004.13237 (2020)

2. Mothilal, R.K., Sharma, A., Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency. pp. 607–617 (2020)
3. Van Looveren, A., Klaise, J.: Interpretable counterfactual explanations guided by prototypes. arXiv preprint arXiv:1907.02584 (2019)
4. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: Automated decisions and the gdpr. Harv. JL & Tech. **31**, 841 (2017)