

On the robustness of sparse counterfactual explanations to adverse perturbations

Marco Virgolin^{a,*}, Saverio Fracaros^b

^a Evolutionary Intelligence Group, Centrum Wiskunde & Informatica, Science Park 123, 1098 XG Amsterdam, the Netherlands

^b Department of Mathematics and Geosciences, University of Trieste, via Weiss 2, 34128 Trieste, Italy

ARTICLE INFO

Article history:

Received 8 April 2022

Received in revised form 25 November 2022

Accepted 12 December 2022

Available online 16 December 2022

Keywords:

Counterfactual explanation

Explainable machine learning

Explainable artificial intelligence

Robustness

Uncertainty

ABSTRACT

Counterfactual explanations (CEs) are a powerful means for understanding how decisions made by algorithms can be changed. Researchers have proposed a number of desiderata that CEs should meet to be practically useful, such as requiring minimal effort to enact, or complying with causal models. In this paper, we consider the interplay between the desiderata of *robustness* (i.e., that enacting CEs remains feasible and cost-effective even if adverse events take place) and *sparsity* (i.e., that CEs require only a subset of the features to be changed). In particular, we study the effect of addressing robustness separately for the features that are recommended to be changed and those that are not. We provide definitions of robustness for sparse CEs that are workable in that they can be incorporated as penalty terms in the loss functions that are used for discovering CEs. To carry out our experiments, we create and release code where five data sets (commonly used in the field of fair and explainable machine learning) have been enriched with feature-specific annotations that can be used to sample meaningful perturbations. Our experiments show that CEs are often not robust and, if adverse perturbations take place (even if not worst-case), the intervention they prescribe may require a much larger cost than anticipated, or even become impossible. However, accounting for robustness in the search process, which can be done rather easily, allows discovering robust CEs systematically. Robust CEs make additional intervention to contrast perturbations much less costly than non-robust CEs. We also find that robustness is easier to achieve for the features to change, posing an important point of consideration for the choice of what counterfactual explanation is best for the user. Our code is available at: <https://github.com/marcovirgolin/robust-counterfactuals>.

© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Modern Artificial Intelligence (AI) systems often rely on machine learning models such as ensembles of decision trees and deep neural networks [1–3], which contain from thousands to billions of parameters. These large models are appealing because, under proper training and regularization regimes, they are often unmatched by smaller models [4,5]. However, as large models perform myriads of computations, it can be very difficult to interpret and predict their behavior. Because of this, large models are often called *black-box models*, and ensuring that their use in high-stakes applications (e.g., of medicine and finance) is fair and responsible can be challenging [6,7].

* Corresponding author.

E-mail address: marco.virgolin@cwi.nl (M. Virgolin).

The field of eXplainable AI (XAI) studies methods to dissect and analyze black-box models [8,9] (as well as methods to generate interpretable models when possible [10]). Famous methods of XAI include feature relevance attribution [11, 12], explanation by analogy with prototypes [13,14], and, of focus in this work, *counterfactual explanations*. Counterfactual explanations enable to reason by contrast rather than by analogy, as they show in what ways the input given to a black-box model needs to be changed for the model to make a different decision [15,16]. A classic example of counterfactual explanation is: “Your loan request has been rejected. If your salary was 60 000\$ instead of 50 000\$ and your debt was 2500\$ instead of 5000\$, your request would have been approved.” A user who obtains an unfavorable decision can attempt to overturn it by intervening according to the counterfactual explanation.

Normally, the search of counterfactual explanations is formulated as an optimization problem (see Sec. 2.1 for a formal description). Given the feature values that describe the user as starting point, we seek the minimal changes to those feature values that result in a point for which the black-box model makes a different (and oftentimes, a specific favorable) decision. We wish the changes to be minimal for two reasons: one, to learn about the behavior of the black-box model for a neighborhood of data points, e.g., to assess its fairness (although this is not guaranteed in general, see e.g., [17]); two, in the hope that putting the counterfactual explanation into practice by means of real-life intervention will require minimal effort too. For counterfactual explanations to be most useful, more desiderata than requiring minimal feature changes may need to be taken into account (see Sec. 9) [18].

In this paper, we consider a desideratum that can be very important for the usability of counterfactual explanations: *robustness to adverse perturbations*. By adverse perturbations we mean changes in feature values that happen due to unforeseen circumstances beyond the user’s control, making reaching the desired outcome no longer possible, or requiring the user to put more effort than originally anticipated. These unforeseen circumstances can have various origins, e.g., time delays, measurement corrections, biological processes, and so on. For example, if a counterfactual explanation for improving a patient’s heart condition prescribes lowering the patient’s blood pressure, the chosen treatment may need to be employed for longer, or even turn out to be futile, if the patient has a genetic predisposition to resist that treatment (for more examples, see Sec. 5.1 and choices made in the coding of our experiments, in `robust_cfe/dataproc.py`).

We show that, if adverse perturbations might happen, one can and *should* seek counterfactual explanations that are robust to such perturbations. A particular novelty of our work is that we distinguish between whether perturbations impact the features that counterfactual explanations prescribe to *change* or *keep as they are* (note that some features may be irrelevant and can be changed differently than how prescribed by a counterfactual explanation, we address this in Sec. 2.3). This is because counterfactual explanations are normally required to be *sparse* in terms of the intervention they prescribe (i.e., only a subset of the features should be changed), for better usability (see Sec. 2.1). As it will be shown, making this discrimination allows to improve the effectiveness and efficiency with which robustness can be accounted for. Consequently, one might need to consider carefully which counterfactual explanation to pursue, based on whether they are robust to features to change or keep as they are.

In summary, this paper makes the following contributions:

1. We propose two workable definitions of robustness of counterfactual explanations that concern, respectively, the features prescribed to be changed and those to be kept as they are;
2. We release code to support further investigations, where five existing data sets are annotated with perturbations and plausibility constraints that are tailored to the features and type of user seeking recourse;
3. We provide experimental evidence that accounting for robustness is important to prevent adverse perturbations from making it very hard or impossible to achieve recourse through counterfactual explanations, when adverse perturbations are sampled from a distribution (i.e., they are not necessarily worst-case ones);
4. We show that robustness for the features to change is far more reliable and computationally efficient to account for than robustness for the features to keep as they are;
5. Additionally, we propose a simple but effective genetic algorithm that outperforms several existing gradient-free search algorithms for the discovery of counterfactual explanations. The algorithm supports plausibility constraints and implements the proposed definitions of robustness.

2. Preliminaries

In the following, we introduce preliminary concepts for reasoning about robustness of counterfactual explanations in a sparse sense. In particular, we (i) describe the problem statement of searching for counterfactual explanations, (ii) present the notions of perturbation and robustness in general terms, and (iii) introduce the definitions of \mathcal{C} and \mathcal{K} , which are sets that partition the features of a counterfactual explanation. The following Secs. 3 and 4 will then present the main contribution of this paper: notions of robustness that are tailored to sparse counterfactual explanations, i.e., specific to \mathcal{C} and \mathcal{K} .

2.1. Problem statement

Let us assume we are given a point $\mathbf{x} = (x_1, \dots, x_d)$, where d is the number of features. Each feature takes values either in (a subset of) \mathbb{R} , in which case we call it a *numerical* feature, or in (a subset of) \mathbb{N} , in which case we call it a *categorical*

feature. For categorical features, we use natural numbers as a convenient way to identify their categories, but disregard ordering. For example, for the categorical feature *gender*, 0 might mean *male*, 1 might mean *female*, and 2 might mean *non-binary*. Thus, $\mathbf{x} \in \mathbb{R}^{d_1} \times \mathbb{N}^{d_2}$, where $d_1 + d_2 = d$.

A *counterfactual example*¹ for a point \mathbf{x} is a point $\mathbf{z} \in \mathbb{R}^{d_1} \times \mathbb{N}^{d_2}$ such that, given a classification (black-box) machine learning model $f : \mathbb{R}^{d_1} \times \mathbb{N}^{d_2} \rightarrow \{c_1, c_2, \dots\}$ (c_i is a decision or *class*), $f(\mathbf{z}) \neq f(\mathbf{x})$. We wish \mathbf{z} to be *close* to \mathbf{x} under a meaningful distance function δ that is problem-specific and meets several desiderata (see Sec. 9). For example, commonly-used distances that are capable of handling both numerical and categorical features are variants of Gower's distance [19] (see Eq. (9) and, e.g., [20] for a variant thereof). Often, when dealing with more than two classes, we also impose $f(\mathbf{z}) = t$, i.e., the *target* class we desire \mathbf{z} to be. Other times, we wish to find a *set* of counterfactual examples $\{\mathbf{z}_1, \dots, \mathbf{z}_k\}$, possibly of different classes, to obtain multiple means of recourse or simply gain information on the decision boundary of f nearby \mathbf{x} (e.g., to explain f 's local behavior) [15,21,22].

For the sake of readability, we provide formal definitions only for the case when all features are numerical (i.e., $\mathbf{x} \in \mathbb{R}^d$, $d_1 = d$, $d_2 = 0$). For completeness, we include explanations of how to deal with categorical features in the running text. Furthermore, we assume feature independence. While this assumption is rarely entirely met in real-world practice, it is commonly done in literature due to the lack of causal models (e.g., only four works consider causality in Sec. 9), and allows us to greatly simplify the introduction of the concepts hereby presented. We discuss the limitations that arise from this assumption in Sec. 8.

A *counterfactual explanation* is represented by a description of how \mathbf{x} needs to be changed to obtain \mathbf{z} . In other words, a counterfactual explanation is a prescription on what interventions should be made to *reach* the respective counterfactual example. For example, under the assumption of independence and all-numerical features, the difference $\mathbf{z} - \mathbf{x}$ is typically considered the counterfactual explanation for how to reach \mathbf{z} from \mathbf{x} . What particular form counterfactual explanations take is not crucial to our discourse, and we will use $\mathbf{z} - \mathbf{x}$ for simplicity.

We proceed by considering the following traditional setting where, for simplicity of exposition and without loss of generality, we will assume that features are pre-processed so that a difference in one unit in terms of feature i is equivalent to a difference in one unit in feature j (i.e., the user's effort is commensurate across different features). Alternatively, one can account for this in the computation of the distance (see, e.g., Eq. (9)). We seek the (explanation relative to an) *optimal* \mathbf{z}^* with:

$$\begin{aligned} \mathbf{z}^* &\in \operatorname{argmin}_{\mathbf{z}} \delta(\mathbf{z}, \mathbf{x}) \\ \text{with } \delta(\mathbf{z}, \mathbf{x}) &:= \|\mathbf{z} - \mathbf{x}\|_1 + \lambda \|\mathbf{z} - \mathbf{x}\|_0 \\ \text{and subject to } f(\mathbf{z}) &= t \text{ and } \mathbf{z} - \mathbf{x} \in \mathcal{P}. \end{aligned} \quad (1)$$

In other words, δ is a linear combination, weighed by λ , of the sum of absolute distances between the feature values of \mathbf{x} and \mathbf{z} , and the count of feature values that are different between \mathbf{x} and \mathbf{z} . Note that \mathbf{z}^* needs not be unique, i.e., multiple optima may exist. Moreover, the difference $\mathbf{z} - \mathbf{x}$ must abide to some plausibility constraints specified in a collection \mathcal{P} . We model plausibility constraints as a set of specifications, each relative to a feature i , concerning whether $z_i - x_i$ is allowed to be > 0 , < 0 , and $\neq 0$, i.e., a feature can increase, decrease, or change at all (for categorical features, we only consider the latter). For example for a private individual who wishes to be granted a loan, one of such constraints may specify that they cannot reasonably intervene to change the value of a currency (such a feature is called *mutable but not actionable*), i.e., counterfactual explanations must have $z_i - x_i = 0$, for i representing currency value. Similarly, the individual's age may increase but not decrease, i.e., $z_i - x_i > 0$, for i representing age.

We particularly consider the $L1$ -norm (i.e., the term $\|\cdot\|_1$ of δ in Eq. (1)) because it is reasonable to think that, for independent features, the total *cost* of intervention (i.e., the effort the user must put) is the sum of the costs of intervention for each feature separately, and that these costs grow linearly. Some works (e.g., [20,23]) choose the $L2$ -norm ($\|\cdot\|_2$, also known as Euclidean norm) instead of the $L1$ -norm; the definitions of robustness given in this paper can be easily adapted for the $L2$ -norm. Regarding the $L0$ -norm (i.e., the term $\|\cdot\|_0$ of δ in Eq. (1)), this term explicitly promotes a form of sparsity, as it seeks to minimize how many features have a different value between \mathbf{z} and \mathbf{x} . This is desirable because, oftentimes, the user can only reasonably focus on, and intervene upon, a limited number of features (even if this amounts to a larger total cost in terms of $L1$ compared to intervention on *all* the features) [24].

2.2. Perturbations & robustness

Unforeseen circumstances (e.g., inflation) might lead to more or different intervention to be needed, compared to what was originally prescribed by a counterfactual explanation (e.g., increase savings by 1000 \$ to be granted credit access). Thus, instead of reaching \mathbf{z} as intended by the counterfactual explanation, a different point \mathbf{z}' is obtained. Note that while the effects of unforeseen circumstances can impact feature values, the circumstances themselves need not be encoded as feature values. In fact, we will only focus on the extent by which feature values may be perturbed by such circumstances.

¹ Many authors use \mathbf{x}' to represent a counterfactual example for \mathbf{x} , instead of \mathbf{z} . We chose \mathbf{z} not to overload the notation with superscripts later on in the manuscript, for readability.

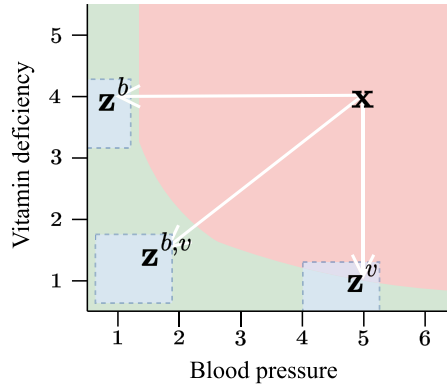


Fig. 1. Example of considering robustness to perturbations when seeking counterfactual examples. The red and green areas respectively represent *high risk* and *low risk* classifications of a cardiac condition according to a model f . The patient, represented by \mathbf{x} , is at high risk. Three possible counterfactual examples are shown for different interventions (white arrows): \mathbf{z}^b for treating blood pressure, \mathbf{z}^v for treating vitamin deficiency, and $\mathbf{z}^{b,v}$ for treating both. We assume to know the maximal extent of perturbation (under reasonable risk) for blood pressure and vitamin deficiency due to natural physiological events. This allows us to define the blue areas surrounding each counterfactual example. Perturbations \mathbf{w} to one of the counterfactual examples can lead to any other point in the blue area. \mathbf{z}^v is the best of the three in terms of proximity to \mathbf{x} but its blue area partly overlaps with the red area. This means that there exist \mathbf{w} such that $\mathbf{z}^v + \mathbf{w}$ leads to a point in the red area, invalidating the counterfactual explanation. In such cases, it is important to estimate if additional intervention is possible so that \mathbf{z}^v can still be reached, and at what cost. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Let us define the vector $\mathbf{w} = \mathbf{z} - \mathbf{z}'$ as a *perturbation* for the counterfactual example \mathbf{z} . We assume that perturbations that impact feature i are sampled from some distribution $P(w_i)$ and we are interested in controlling for, or *being robust to*, large magnitude perturbations that have reasonable risk. For example for normally-distributed perturbations, we might want to consider the values that can be sampled at the 95th or 99th percentile. We will therefore assume that we can define a vector $\mathbf{p} = (p_1^{(-)}, p_1^{(+)}, \dots, p_d^{(-)}, p_d^{(+)})$ where $p_i^{(-)} \leq 0$ and $p_i^{(+)} \geq 0$ represent, respectively, the smallest negative and largest positive perturbations that can reasonably happen to the i^{th} feature. For example, if the i^{th} feature represents the blood pressure of a patient, then $p_i^{(-)}$ tells by how much the blood pressure might lower at most (e.g., as a consequence of dehydration) and $p_i^{(+)}$ tells by how much the blood pressure might raise at most (e.g., as a consequence of anti-inflammatory drug intake). Clinicians may be able to define this information from their experience or retrieve it from medical literature. In general, the magnitudes of $p_i^{(-)}, p_i^{(+)}$ need not be the same, i.e., $|p_i^{(-)}| \neq |p_i^{(+)}|$. Note that for an i^{th} feature that is categorical, decreases or increases $p_i^{(-)}, p_i^{(+)}$ as explained for numerical features are no longer meaningful. For categorical features, we will assume that \mathbf{p} contains elements that represent what categorical perturbations are possible for that feature, i.e., p_i will be a set of indices that represent categories.

Under the problem setting we considered in Sec. 2.1, perturbations that may impact a counterfactual explanation define a *box* (hyper-rectangle) of all possible points \mathbf{z}' that can be reached from \mathbf{z} due to perturbations. An example is illustrated in Fig. 1. We define the concept of **p-neighborhood** of \mathbf{z} as follows:

Definition 1. (p-neighborhood** and **p-neighbors** of a counterfactual example)** Given a model f , a point \mathbf{x} , a respective counterfactual example \mathbf{z} , and a vector of possible perturbations \mathbf{p} , the **p-neighborhood** of \mathbf{z} is the set:

$$N := \left\{ \mathbf{z}' \mid \mathbf{z}' \in [\mathbf{z} + \mathbf{p}^{(-)}, \mathbf{z} + \mathbf{p}^{(+)}] \right\}. \quad (2)$$

A point $\mathbf{z}' \in N$ such that $\mathbf{z}' \neq \mathbf{z}$ is called a **p-neighbor** of \mathbf{z} .

Not all perturbations are problematic. Our goal is to study robustness to *adverse* perturbations, i.e., those for which $f(\mathbf{z} + \mathbf{w}) = f(\mathbf{z}') \neq t$. In other words, we wish to seek counterfactual examples \mathbf{z} that have no (or the fewest possible) **p-neighbors** for which perturbations can cause the classification performed by f to be different from t . When that happens, we say that the counterfactual explanation has been *invalidated* by the perturbation. However, it may be the case that invalidation is not *permanent*: there may still exist intervention (i.e., a new counterfactual explanation) that adheres to the constraints in \mathcal{P} and allows us to overcome invalidation. Therefore, in this work, we will seek to discover counterfactual explanations that are *robust* in the sense that (i) if invalidated, additional intervention remains possible, (ii) the cost of additional intervention is small.

Unforeseenly, if f is assumed to be a general model (e.g., not necessarily a linear one), then the following argument holds.

Proposition 1. For a general f , information on the classification of a \mathbf{p} -neighbor (e.g., that $f(\mathbf{z}') = f(\mathbf{z})$ for \mathbf{z}' on the boundary of N) provides no information about the classification of another \mathbf{p} -neighbor (e.g., that $f(\mathbf{z}') \neq f(\mathbf{z}'')$ for \mathbf{z}'' in the interior of N).

Proof. We cannot preclude that the model f is, for example, a neural network. Under the universal approximation theorem [25], f may represent any function. Thus, f may represent a Swiss cheese-like function, where for example $f(\mathbf{z}') \neq f(\mathbf{z}'')$ with $\mathbf{z}'' := \mathbf{z}' + \mathbf{e}$ and $\mathbf{e} = (\varepsilon_1, \dots, \varepsilon_d)$ different from the zero-vector, however small $|\varepsilon_i|$, $\forall i$. \square

This proposition means that if no information on, e.g., regularity or smoothness of f is available, then we must check each and every \mathbf{p} -neighbor \mathbf{z}' of \mathbf{z} to assess whether some of them may invalidate the explanation, i.e., $\exists \mathbf{z}'$ such that $f(\mathbf{z}') \neq t$. Checking all neighbors is typically not feasible, e.g., as soon as some of the features are real valued. Thus, the best one can do is to take an approximate approach. For example, a Monte-Carlo sampling approach can be used where a batch of random points within N is considered, hoping that the batch is representative of all points in N . As we will show in the next sections, a better strategy can be designed if sparsity is considered.

We conclude this section by noting that perturbations, as described so far, are *absolute*, i.e., independent of the starting point \mathbf{x} , the counterfactual in consideration \mathbf{z} , or the intervention entailed by the counterfactual explanation $\mathbf{z} - \mathbf{x}$. Perturbations to feature i might however depend on x_i and z_i , i.e., be sampled from a distribution $P(w_i|x_i, z_i)$. For example, due to market fluctuations, a return on investment may be smaller than anticipated by 5% of the expected value. Such type of *relative* perturbations entail different \mathbf{p} -neighborhoods for different \mathbf{x} and \mathbf{z} . For simplicity and without loss of generality, we will proceed by assuming that perturbations can only be absolute. We explain how we also included *relative* perturbations in the annotations used for our experiments in Sec. 5.

2.3. Sparsity, features in \mathcal{C} and \mathcal{K}

We use the form of sparsity mentioned in Sec. 2.1 to partition the features into two sets. As mentioned before, sparsity is an important desideratum because it may not be reasonable to expect that the user can realistically intervene on, and keep track of, *all* the features to achieve recourse. Given a specific counterfactual explanation \mathbf{z} for the point \mathbf{x} , we call the set containing the (indices of the) features whose values should *change* $\mathcal{C} = \{i \in \{1, \dots, d\} \mid z_i \neq x_i\}$, and its complement, i.e., the set of the (indices of the) features whose values should be *kept as they are*, $\mathcal{K} = \{i \in \{1, \dots, d\} \mid z_i = x_i\}$. Typically, because a sufficiently large λ is used, or because of the plausibility constraints specified in \mathcal{P} , $\mathcal{K} \neq \emptyset$.

Note that the proposed partitioning between \mathcal{C} and \mathcal{K} implicitly assumes that all features are relevant to the counterfactual explanation. If certain features are irrelevant, perturbations to those features will have no effect on f 's decision, and thus those features need not be accounted for when assessing robustness. This means that accounting for irrelevant features makes assessing robustness more computationally expensive than needed. However, as f is considered a black-box, we cautiously assume that all features are relevant for assessing robustness.

We will proceed by accounting for perturbations and respective robustness separately for features in \mathcal{C} and \mathcal{K} . Accounting for robustness separately is important because, as we will show, assessing robustness for features in \mathcal{C} can be done far more efficiently and be more effective than for features in \mathcal{K} . Knowing this, if multiple counterfactual explanations can be found, the user may want to choose the counterfactual explanation that fits him/her best based on the robustness it exhibits in terms of \mathcal{C} and \mathcal{K} . In the next section, we present our first notion of robustness, which concerns \mathcal{C} .

3. Robustness for \mathcal{C}

We begin by focusing on the features that the counterfactual explanation instructs to change, i.e., the features (whose indices are) in \mathcal{C} . Recall that we assume that a vector of maximal perturbation magnitudes \mathbf{p} can be defined. This leads us to the following definition.

Definition 2. (\mathcal{C} -perturbation) Given a point \mathbf{x} , a respective counterfactual example \mathbf{z} , and the vector of maximal magnitude perturbations \mathbf{p} , a \mathcal{C} -perturbation for the counterfactual explanation $\mathbf{z} - \mathbf{x}$ is a vector

$$\mathbf{w}^{\mathcal{C}} = (w_1^{\mathcal{C}}, \dots, w_d^{\mathcal{C}}), \text{ where} \quad (3)$$

$$\begin{cases} p_i^{(-)} \leq w_i^{\mathcal{C}} \leq p_i^{(+)} & \text{if } i \in \mathcal{C}, \\ w_i^{\mathcal{C}} = 0 & \text{otherwise, i.e., } i \notin \mathcal{C} \end{cases} \quad (4)$$

and such that $\exists i : w_i^{\mathcal{C}} > 0$, i.e., $\mathbf{w}^{\mathcal{C}}$ is not the zero-vector.

In other words, a \mathcal{C} -perturbation is a perturbation that acts only on features in \mathcal{C} , and at least on one of such features. Next, we use the concept of \mathcal{C} -perturbations to introduce the one of \mathcal{C} -setbacks.

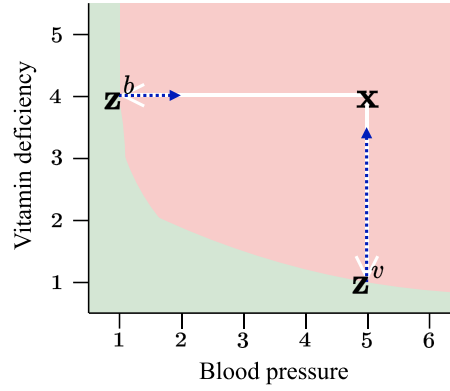


Fig. 2. Example of C -setbacks. The red and green areas represent *high risk* and *low risk* classifications of a cardiac condition according to a model f . The patient, represented by \mathbf{x} , is at high risk. Two treatments are possible but cannot be administered jointly due to drug incompatibility, hence sparsity of intervention is needed (only one of the two treatments can be pursued). The closest (and thus optimal) counterfactual example is \mathbf{z}^v and concerns treating vitamin deficiency (white arrow pointing down). Another counterfactual example is \mathbf{z}^b and concerns treating blood pressure (white arrow pointing left). Maximal C -setbacks are shown for both counterfactual examples (blue dashed segments). The setbacks can make both counterfactual examples invalid. However, the counterfactual examples can still be reached with additional intervention (treatment administration). Indeed, if one accounts for the possibility that maximal C -setbacks may take place, then the total cost (original intervention + additional intervention to remedy the perturbation, i.e., $3 + 2.5$) to reach \mathbf{z}^v becomes larger than the total cost to reach \mathbf{z}^b (i.e., $4 + 1$).

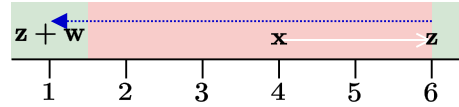


Fig. 3. One-dimensional example of how a C -setback can be advantageous if the magnitude of perturbation exceeds the magnitude of intervention.

Definition 3. (C -setback) A C -setback for the counterfactual explanation $\mathbf{z} - \mathbf{x}$ is a C -perturbation such that

$$\begin{cases} p_i^{(-)} \leq w_i^c \leq 0 & \text{if } z_i - x_i > 0 \\ 0 \leq w_i^c \leq p_i^{(+)} & \text{if } z_i - x_i < 0 \\ w_i^c = 0 & \text{otherwise, i.e., } i \notin \mathcal{C}. \end{cases} \quad (5)$$

We denote C -setbacks with $\mathbf{w}^{c,s}$.

In words, a C -setback is a C -perturbation where each and every element of the perturbation $w_i^{c,s}$ is of opposite sign to the counterfactual explanation $z_i - x_i$. We can interpret the meaning of C -setbacks $\mathbf{w}^{c,s}$ as vectors that *push the user away from \mathbf{z} and back towards \mathbf{x}* along the direction of intervention. Furthermore, we call a *maximal C -setback*, denoted by $\mathbf{w}_{\max}^{c,s}$, C -setback whose elements that correspond to features in \mathcal{C} have maximal magnitude, i.e., $w_i^{c,s} = p_i^-$ if $z_i - x_i > 0$ and $w_i^{c,s} = p_i^+$ if $z_i - x_i < 0$. An example is given in Fig. 2.

C -setbacks are arguably more interesting than C -perturbations because C -setbacks are the subset of these perturbations that plays against the user. In fact, certain C -perturbations might be advantageous, enabling to reach \mathbf{z} with less intervention than originally provisioned (i.e., when the sign of w_i^c and that of $z_i - x_i$ matches). To account for robustness, we are interested in understanding whether perturbations can prevent us to reach \mathbf{z} , hence we will proceed by focusing exclusively on C -setbacks.

It is important to note that even C -setbacks can be advantageous if one allows their perturbations to be of larger magnitude than intervention, i.e., if $|w_i^{c,s}| > |z_i - x_i|$ is allowed ([26] discuss this aspect in detail). In a nutshell, if $|w_i^{c,s}| > |z_i - x_i|$, then a C -setback can lead to a point that “precedes” \mathbf{x} in terms of the direction of intervention. For that point, the intervention may be less costly than the one that was originally planned or entirely not needed because the point is of the target class (see, e.g., Fig. 3). Advantageous situations are not interesting for robustness and counterfactual explanations that can be overturned by perturbations may well not be interesting to pursue. We therefore consider any C -setback to have elements capped by $|w_i^{c,s}| \leq |z_i - x_i|$.

Perhaps the most interesting scenario for considering C -setbacks is when dealing with \mathbf{z}^* , since a counterfactual example that is optimal (i.e., one minimizes Eq. (1)) is an ideal outcome. The following simple result holds for \mathbf{z}^* :

Proposition 2. For any C -setback $\mathbf{w}^{c,s}$ of \mathbf{z}^* (such that $|w_i^{c,s}| \leq |z_i^* - x_i|$ for all i), $f(\mathbf{z}^* + \mathbf{w}^{c,s}) \neq t$.

Proof. We use *reduction ad absurdum*. Let us assume the opposite of what was said in Proposition 2, i.e., there exists $\mathbf{w}^{c,s}$ such that $f(\mathbf{z}^* + \mathbf{w}^{c,s}) = t$. Let $\mathbf{z}' := \mathbf{z}^* + \mathbf{w}^{c,s}$, and so $f(\mathbf{z}') = t$. By construction of $\mathbf{w}^{c,s}$, $\delta(\mathbf{z}', \mathbf{x}) = \delta(\mathbf{z}^* + \mathbf{w}^{c,s}, \mathbf{x}) < \delta(\mathbf{z}^*, \mathbf{x})$. In other words, \mathbf{z}' is of the target class and is closer to \mathbf{x} than \mathbf{z}^* is. This contradicts the fact that \mathbf{z}^* is optimal. \square

Now, because of Proposition 2, we are *guaranteed* that if a \mathcal{C} -setback $\mathbf{w}^{c,s}$ happens to \mathbf{z}^* , the resulting point will no longer be classified as t . Intuitively, this is a natural consequence of the fact that optimal counterfactual examples lay on the border of the decision boundary as otherwise they would not be optimal. Also, since \mathbf{z}^* is optimal, the respective L_0 component for the distance between \mathbf{x} and \mathbf{z}^* is minimal, i.e., all features in \mathcal{C} and thus in $\mathbf{w}^{c,s}$ are relevant for the classification. Given the premises just made, it becomes important to understand whether invalidation to \mathbf{z}^* can be averted with additional intervention and, if so, whether the cost of such intervention can be minimized.

It is important to note that invalidation of a counterfactual explanation due to a \mathcal{C} -setback can always be averted, i.e., additional intervention to reach the intended z_i for all $i \in \mathcal{C}$ is always possible. To see this, consider the fact that the intervention entailed by the counterfactual explanation $\mathbf{z} - \mathbf{x}$ must adhere to the plausibility constraints specified in \mathcal{P} (else, $\mathbf{z} - \mathbf{x}$ would not be a possible counterfactual explanation). Since \mathcal{C} -setbacks are aligned with the direction of the original intervention, the point $\mathbf{z} + \mathbf{w}^{c,s} - \mathbf{x}$, which is in between \mathbf{x} and \mathbf{z} , must meet \mathcal{P} . It therefore suffices to apply additional intervention along the originally-intended direction to recover the desired counterfactual example. Under the L_1 -norm (as per the choice of δ in Eq. (1)), the cost associated with the additional intervention needed to overcome a \mathcal{C} -setback $\mathbf{w}^{c,s}$ is simply $\|\mathbf{w}^{c,s}\|_1$.

Since invalidation due to \mathcal{C} -setbacks can be dealt with additional intervention, and since one can reasonably assume that the user keeps track of how the value of x_i changes for $i \in \mathcal{C}$ over the course of intervention (otherwise, (s)he would not know when to stop the intervention), it follows that there is no necessity for counterfactual examples to be far from the decision boundary in terms of their features in \mathcal{C} . (Note that this is in contrast with prior work on aspects of robustness for counterfactuals, where the possibility of additional intervention is not considered and counterfactual examples are required to be far from the decision boundary in terms of all of their features; see Sec. 9.) Thus, rather than seeking counterfactual examples that are not invalidated by \mathcal{C} -setbacks, we seek counterfactual examples for which the additional intervention that is needed to contrast \mathcal{C} -setbacks is minimal. To this end, we can use Proposition 2 in order to seek counterfactual examples that are optimal (i.e., require minimum intervention cost) when the additional cost to contrast maximal \mathcal{C} -setbacks $\mathbf{w}_{\max}^{c,s}$ is factored in. In the following definition, to highlight that \mathcal{C} -setbacks depend on the specific \mathbf{z} and \mathbf{x} (as they determine \mathcal{C}) and avoid confusion, we use the function notation $W_{\max}^{c,s}(\mathbf{z}, \mathbf{x})$ in place of $\mathbf{w}_{\max}^{c,s}$.

Definition 4. (*Optimal counterfactual example under \mathcal{C} -setbacks*) Given a model f , a point \mathbf{x} , and a vector \mathbf{p} , we call a point $\mathbf{z}^{*,c}$ such that

$$\mathbf{z}^{*,c} - W_{\max}^{c,s}(\mathbf{z}^{*,c}, \mathbf{x}) \in \operatorname{argmin}_{(\mathbf{z} - W_{\max}^{c,s}(\mathbf{z}, \mathbf{x}))\delta} (\mathbf{z} - W_{\max}^{c,s}(\mathbf{z}, \mathbf{x})), \quad (6)$$

an *optimal counterfactual example under \mathcal{C} -setbacks*.

This definition gives us a way to seek a (multiple may exist) counterfactual explanation that entails minimal intervention cost when accounting for maximal \mathcal{C} -setbacks. Indeed, it suffices to equip a given search algorithm with Eq. (6), i.e., perform the following steps: (1) for any \mathbf{z} to be evaluated, compute the respective $\mathbf{w}_{\max}^{c,s}$, (2) instead of computing $\delta(\mathbf{z}, \mathbf{x})$, compute $\delta((\mathbf{z} - \mathbf{w}_{\max}^{c,s}), \mathbf{x})$, and (3) at the end of the search, return the point that minimizes such distance, i.e., $\mathbf{z}^{*,c}$.

Performing the computations just mentioned takes linear time in the number of features ($O(d)$) because we only need to build $\mathbf{w}_{\max}^{c,s}$ (step 1 above) and subtract it from \mathbf{z} prior to computing δ (step 2 above) for any given \mathbf{z} (f should still be evaluated on \mathbf{z}). This is relatively fast (as demonstrated in B.3.2), especially compared to the situation described in Sec. 2.2, where one would need to use f to predict the class of a number of neighbors of \mathbf{z} . Note also that in Eq. (6) setbacks are subtracted from counterfactual examples when computing δ , to account for the fact that the cost should increase (recall the construction of \mathcal{C} -setbacks in Definition 3).

4. Robustness for \mathcal{K}

We now consider \mathcal{K} , i.e., the set concerning the features that should be kept to their current value. Mirroring the notion of \mathcal{C} -perturbation (Definition 2), we can define a \mathcal{K} -perturbation to be a vector \mathbf{w}^k such that $p_i^{(-)} \leq w_i^k \leq p_i^{(+)}$ if $i \in \mathcal{K}$ and $w_i^k = 0$ if $i \notin \mathcal{K}$. Similarly, we can cast the concept of neighborhood from Definition 1 to consider only \mathcal{K} -perturbations, leading to:

Definition 5. (*\mathcal{K} -neighborhood and \mathcal{K} -neighbors of a counterfactual example*) Given a model f , a point \mathbf{x} , a respective counterfactual example \mathbf{z} , and a vector of possible perturbations \mathbf{p} , the \mathcal{K} -neighborhood of \mathbf{z} under \mathbf{p} is the set:

$$K := \left\{ \mathbf{z}' \mid \begin{array}{l} z'_i \in [z_i + p_i^{(-)}, z_i + p_i^{(+)}], \text{ if } i \in \mathcal{K} \\ z'_i = z_i \text{ otherwise} \end{array} \right\}. \quad (7)$$

A point $\mathbf{z}' \in K$ such that $\mathbf{z}' \neq \mathbf{z}$ is called a \mathcal{K} -neighbor of \mathbf{z} .

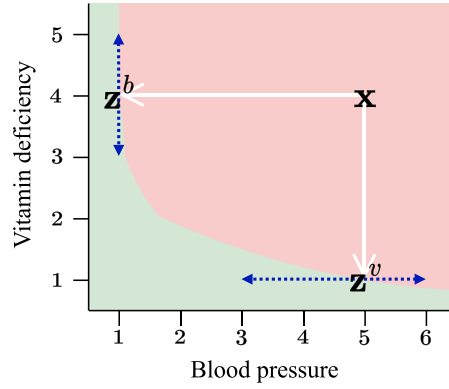


Fig. 4. Example of \mathcal{K} -perturbations. The counterfactual example \mathbf{z}' is vulnerable to \mathcal{K} -perturbations because these can lead to the red area; the same is not true for \mathbf{z}^b . If it is not plausible to reduce blood pressure, then \mathcal{K} -perturbations to \mathbf{z}' can lead to permanent invalidity. Else, they can be resolved with additional intervention, in terms of blood pressure.

For a categorical feature $i \in \mathcal{K}$, the neighborhood can be built by swapping z_i with one of the possibilities listed in $p_i \in \mathbf{p}$, where p_i will be a set containing categories perturbations can lead to.

Next, we use K to define the concept of vulnerability to \mathcal{K} -perturbations:

Definition 6. (*Vulnerability to \mathcal{K} -perturbations*) Given a model f , a point \mathbf{x} , and a vector \mathbf{p} , a counterfactual example \mathbf{z} is vulnerable to \mathcal{K} -perturbations if $\exists \mathbf{z}' \in N(\mathbf{z}, \mathbf{p})$ such that $f(\mathbf{z}') \neq f(\mathbf{z})$.

Informally, this definition says that \mathbf{z} is vulnerable to \mathcal{K} -perturbations if the decision boundary surrounding \mathbf{z} is not sufficiently loose with respect to the features in \mathcal{K} . Fig. 4 shows an example. The reason why vulnerability to \mathcal{K} -perturbations is particularly important is that, differently from the case of \mathcal{C} -perturbations, a \mathcal{K} -perturbation can invalidate the counterfactual explanation *permanently*. In fact, a \mathcal{K} -perturbation changes \mathbf{z} along a different direction than the one of intervention. Thus, a \mathcal{K} -perturbation can lead to a point \mathbf{z}' from which there exists no plausible intervention to reach the originally-intended \mathbf{z} from.

For example, consider the feature i to represent *inflation* as a mutable but not actionable feature, i.e., a feature that can be changed (e.g., by global market trends) but not by the user. \mathcal{P} will state that no (user) intervention can exist to change i , i.e., \mathcal{P} imposes $z_i - x_i = 0$. However, an unforeseen circumstance such as the financial crisis of 2008 may lead to a large inflation increase ($p_i^+ > 0$). Consequently, it may become impossible for the user to obtain the desired loan, e.g., because the bank does not hand out certain loans when the inflation is too high.

Now, recall that the reason why Definition 4 can be used for the case of \mathcal{C} -perturbations is that Proposition 2 holds, i.e., there cannot exist points of class t between \mathbf{x} and an optimal counterfactual example \mathbf{z}^* . The same does not hold for \mathcal{K} -perturbations, i.e., since the features in \mathcal{K} are orthogonal to the direction of intervention, it can happen that the maximal perturbation to a feature $i \in \mathcal{K}$ leads to a point \mathbf{z}' for which $f(\mathbf{z}') = t$, while a non-maximal perturbation to the same feature can lead to a point \mathbf{z}'' for which $f(\mathbf{z}'') \neq t$. Thus, checking for maximal perturbations is no longer sufficient: we must check instead for all points in the \mathcal{K} -neighborhood K .

As mentioned in Sec. 2.2, checking each and every point in a neighborhood may not be feasible. Thus, we propose to approximate the assessment of how \mathcal{K} -robust (i.e., non-vulnerable to perturbations in \mathcal{K}) counterfactual explanations can be, with Monte-Carlo sampling. Let $\mathbf{1}_{f(\mathbf{z})} : K \rightarrow \{0, 1\}$ be the indicator function that returns 1 for \mathcal{K} -neighbors that share the same class of \mathbf{z} (i.e., $f(\mathbf{z})$), and 0 for those that do not. Taken a random sample of m \mathcal{K} -neighbors, we define the following score:

$$\mathcal{K}\text{-robustness score}(\mathbf{z}, m) = \frac{1}{m} \sum_{i=1}^m \mathbf{1}_{f(\mathbf{z})}(\mathbf{z}'_i). \quad (8)$$

We remark that even if $\mathcal{K}\text{-robustness score}(\mathbf{z}, m) = 1$, we are not guaranteed that \mathbf{z} is \mathcal{K} -robust, because the score is an approximation. Still, this score can be used to determine which counterfactual examples are preferable to pursue in that they are associated with a smaller risk that adverse perturbations will invalidate them (permanently or not).

5. Experimental setup

In this section, we firstly describe the preparation of the data sets used in our experiments. Secondly, we describe the search algorithms considered for finding near-optimal counterfactual explanations. Lastly, we describe the loss function considered, as well as how to incorporate the proposed notions of robustness into it.

Table 1

Considered data sets, where n and d (resp., d_2) indicate the number of observations and features (only categorical) after pre-processing. The column t is the target class for the (simulated) user. Plausib. constr. reports the number of plausibility constraints that allow features to only increase (\geq), remain equal ($=$), and decrease (\leq). The column Perturb. reports the number of perturbations concerning numerical (N) and categorical (C) features. Finally, Acc._{rf} and Acc._{nn} report the average (across five folds) test accuracy of the random forest and neural network models.

Data set (abbrev.)	n	d	d_2	Classes	User	t	Plausib. constr.	Perturb.	Acc. _{rf}	Acc. _{nn}
Credit risk (Cre)	1000	20	6	High, low	Individual	Low	$\geq:3, =:8, \leq:0$	N:6, C:0	0.76	0.75
Income (Inc)	1883	12	7	High, low	Individual	High	$\geq:2, =:3, \leq:0$	N:4, C:4	0.83	0.82
House price (Hou)	506	13	1	High, low	Municipality	Low	$\geq:0, =:3, \leq:1$	N:11, C:0	0.93	0.93
Productivity (Pro)	1196	12	5	High, med., low	Company	High	$\geq:0, =:0, \leq:0$	N:5, C:2	0.79	0.70
Recidivism risk (Rec)	2000	10	6	High, low	Inmate	Low	$\geq:2, =:2, \leq:0$	N:3, C:2	0.80	0.78

Table 2

Examples of perturbations that we manually annotated on the considered data sets. We take relative perturbations (those with %) with respect to the value of the feature in the intended counterfactual example \mathbf{z} in consideration by the search algorithm.

D.set	Feature	Decrease	Increase	Note
		or Categories		
Cre	Savings	10%	10%	Might happen to save less or more relative to what intended.
Inc	Marital status	{single, married, widowed, ...}		Unforeseen change due to, e.g., proposal, divorce, death.
Hou	Crime rate	1%	5%	Relative, might increase more than decrease.
Pro	Overtime	3	3	Up to 3 more or less days of overtime might be needed.
Rec	Age	0	2	Judicial system delays for up to 2 years.

5.1. Data sets

Table 1 summarizes the data sets we consider. For each data set, we make an assumption on the type of user who seeks recourse, e.g., the user could be a private individual seeking to increase their income, or a company seeking to improve the productivity of its employees. Based on this, we manually define the target class t , the set of plausibility constraints \mathcal{P} on what interventions are reasonably plausible, and the collection \mathbf{p} of maximal magnitudes from which perturbations can be sampled (we will consider uniform and normal distributions). We named the data sets in Table 1 to represent their purpose. Originally, *Credit risk* (abbreviated to Cre) is known as *South German Credit Data* [27], which is a recent update that corrects inconsistencies in the popular *Statlog German Credit Data* [28]. *Income* (Inc) is often called *Adult* or *Census income* [29,30]. *Housing price* (Hou) is also known as *Boston housing* [31] and is often used for research on fairness and interpretability because one of its features raises ethical concerns [32]. *Productivity* (Pro) concerns the productivity levels of employees producing garments [33]. Lastly, *Recidivism* (Rec) is a data set collected by an investigation of ProPublica about possible racial bias in the commercial software *COMPAS*, which intends to estimate the risk that an inmate will re-offend [34]. Examples of recent works on fair and explainable machine learning that adopted (some of) these data sets are [20,35–40].

We pre-process the data sets similarly to how done often in the literature. This includes, e.g., removal of redundant features and of observations with missing values, and limiting the number of observations considered for Rec. Regarding our annotations for the perturbations, numerical features can have perturbations that increase or decrease the feature value, in absolute or relative terms; we compute relative perturbations with respect to \mathbf{z} . For example, for the numerical feature *capital-gain* of Inc, we assume that perturbations can happen that lead up to a relative 5% increase or 10% decrease of that feature, based on the value to achieve for that feature. For categorical features, we define only absolute perturbations, i.e., possible changes of category are not conditioned to the current category. The choices we made to build \mathbf{p} are subjective, we elaborate on this in Sec. 8. We sample the amount of perturbation using a uniform or normal distribution, as indicated in Sec. 7. Table 2 shows some examples of maximal perturbations we annotated. As mentioned before, we also define plausibility constraints \mathcal{P} for each data set. Each constraint is specific to a feature. For an i^{th} numerical feature, possible constraints are $z_i - x_i \geq 0$, $z_i - x_i \leq 0$, $z_i - x_i = 0$, and *none*. For an i^{th} categorical feature, possible constraints are $z_i = x_i$ and *none*. Full details about our pre-processing and definition of \mathbf{p} and \mathcal{P} are documented in the form of comments in our code, in `robust_cfe/dataproc.py`.

5.2. Black-box models

We consider random forest and neural networks (with standard multi-layer perceptron architecture) as black-box machine learning models f . We use Scikit-learn's implementations [41]. We assume that we can only access the predictions of f , and no other information such as model parameters or gradients. Our experiments are repeated across a stratified five-fold cross-validation, and each model is obtained by grid-search hyper-parameter tuning. Once trained, the models obtain test accuracy varying from 70% to more than 90% on average across the different data sets, i.e., meaningful decision boundaries are learned. See Appendix A for details on hyper-parameter tuning, and the accuracy of the models on the different data sets. For the discovery of counterfactual examples, we consider observations \mathbf{x} such that $f(\mathbf{x}) \neq t$, from the test sets of the cross-validation.

Table 3

Settings of the considered counterfactual search algorithms. For NeMe, we only set the maximum number of iterations to 100 to achieve commensurate runtimes to those of CoGS (other settings are default). For DiCE, we consider two configurations (“a” and “b”). The loss used (except for DiCE a) is Eq. (9).

CoGS		DiCE (a, b)	
Setting	Value	Setting	Value
Population size	1000	Method	Genetic
Num. generations	100	Total CEs	a : 20, b : 100
Tournament size	2	Max. iterations	a : 500, b : 100
Sm_{ut}	25%	Loss weights	a : Default, b : 0.5 prox., 0.5 spars., 0 div.

GrSp		LORE	
Setting	Value	Setting	Value
Num. in layer	2000	Population size	1000
First radius	0.1	Num. generations	10
Decrease radius	10	Discrete use probabilities	False
Sparse	True	Continuous function estim.	False

5.3. Counterfactual search algorithms

To provide experimental results concerning robustness (Sec. 7), we firstly seek a counterfactual search algorithm that performs best overall among several candidates. To that end, we consider and benchmark the following algorithms from the literature, that can operate upon black-box f : *Diverse Counterfactual Explanations* (DiCE) [22], *Growing Spheres* (GrSp) [23], *Local Rule-based Explanations* (LORE) [20,42], and the *Nelder-Mead method* (NeMe) [43,44]. Furthermore, we devise our own algorithm, a genetic algorithm that we name *Counterfactual Genetic Search* (CoGS).²

The settings used for the algorithms are reported in Table 3. We describe the algorithms below. Note that all of the algorithms are heuristics with no guarantee of discovering optimal (i.e., minimal distance) counterfactual examples, given the nature of the search problem (general, black-box f).

5.3.1. DiCE

DiCE is actually a library that includes three algorithms: random sampling, KD-tree search (i.e., a fast-retrieval data structure built upon the points in the training set), and a genetic algorithm. Of the three, we consider the latter because it performed substantially better in preliminary experiments (and simply refer to it by DiCE). DiCE is configured to return a collection of counterfactual examples rather than a single one. However, three of the other algorithms we consider return a single counterfactual example. Thus, to compare the algorithms on an equal footing, we set DiCE to return a single counterfactual example too. We achieve this by ranking each counterfactual example in the collection according to the loss function in consideration (explained below, see Sec. 5.4), and picking the best-ranking point. We will further consider two different configurations of DiCE:

- *Configuration “a”* uses the default settings except for allowing for a longer number of iteration, to match the same computational budget given to the other algorithms.
- *Configuration “b”* uses custom settings that are aligned to be similar to those used for CoGS, since both DiCE and CoGS are genetic algorithms.

5.3.2. GrSp

GrSp is a greedy algorithm that iteratively samples neighbors of the starting point \mathbf{x} within spheres (i.e., in an L_2 sense) that have increasing radius, until counterfactual examples are found. GrSp includes feature selection to promote sparsity. Unforeseenly, GrSp can only handle numerical features. To be able to use GrSp in our comparison, we let GrSp operate on categorical features as if they were numerical ones (categories are encoded as integers). At the end of the optimization, we transform numerical values back to categories by rounding. Note that this is sub-optimal because an artificial ordering is introduced between categories.

5.3.3. LORE

LORE works by generating a neighborhood around \mathbf{x} with random search or with a genetic algorithm, finding multiple counterfactual explanations at different distance. We consider the variant that adopts the genetic algorithm, because it performed substantially better in preliminary experiments. After the neighborhood is determined, LORE fits a decision tree

² <https://github.com/marcovirgolin/cogs>.

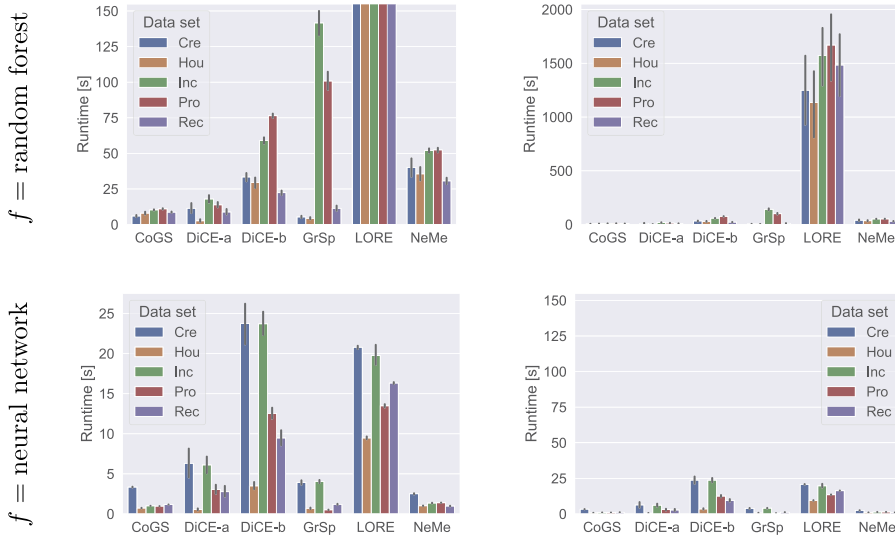


Fig. 5. Runtimes (means and 95% confidence intervals) of the counterfactual search algorithms for the considered data sets and black-boxes (random forest and neural network). The right plots are zoomed-out versions of the left ones.

upon it. Since each path from the root of the decision tree to a leaf represents a classification rule (e.g., “AGE ≥ 3.4 & SALARY_CATEGORY = HIGH $\rightarrow t$ ”), LORE essentially returns multiple counterfactual explanations expressed as rules. To be able to compare with the other algorithms (which return a single counterfactual example), we build one counterfactual example \mathbf{z} by taking the shortest rule returned by LORE, and applying the rule to the starting point \mathbf{x} (e.g., using the rule above, we set the \mathbf{x} 's age and salary to 3.4 and high, respectively).

We found (confirmed by a discussion with the authors) that applying LORE's rules may result in points that are not actually classified as t . When that happens, we perform up to 15 attempts at generating a counterfactual example from the (shortest returned) rule, by focusing on numerical features that are prescribed to be $>$, \geq (or $<$, \leq) than a certain value. In particular, in applying such part of the rule to \mathbf{x} , we add (or subtract) to the prescribed value a term ϵ , which is initially set to 10^{-3} and is doubled at every attempt. Moreover, since we found LORE to be computationally expensive to run (see Fig. 5), we used a fraction of the computation budget allowed for the other algorithms (see Table 3).

5.3.4. NeMe

NeMe is a classic simplex-based algorithm for gradient-free optimization. Like GrSp, also NeMe cannot naturally handle categorical features. Thus, we use the same approximation used for GrSp, i.e., encode categories with integers, let NeMe treat categories as numerical values, and map such values back to integers (and thus categories) by rounding at the end. We use SciPy's implementation with default parameters [45].

5.3.5. CoGS

We design CoGS as a relatively standard genetic algorithm, adapted for the search of points neighboring \mathbf{x} (especially in terms of the L_0 -norm). CoGS operates as follows. First, an initial *population* of candidate solutions is generated by sampling feature values uniformly within an interval for numerical features, and from the possible categories for categorical features. These intervals can be specified or taken automatically from the training set. With probability of $2/d$ (d being the total number of features), the feature value of a candidate solution is *copied* from \mathbf{x} rather than sampled. Every iteration of the algorithm (in the jargon of evolutionary computation, *generation*), offspring solutions are produced from the current population by *crossover* and *mutation*. Following this, *survival of the fittest* is applied to form the population for the next generation.

Our version of crossover produces two offspring solutions by simply swapping the feature values of two random parents, uniformly at random. Our version of mutation produces one offspring solution from one parent solution by randomly altering its feature values. A feature value is altered with probability of $1/d$ (else, it is left untouched). If the feature to alter is categorical, then the category is swapped with another category, uniformly at random. If the feature to alter is numerical, firstly a random number r is sampled uniformly at random between $-s_{mut}/2$ and $+s_{mut}/2$, where $s_{mut} \in (0, 1]$ is a hyper-parameter that represents the maximal extent of allowed mutations; secondly, the original feature value is changed by adding $r \times (\max_i - \min_i)$, where \max_i and \min_i are, respectively, the maximum and minimum values that are possible for that feature.

After crossover and mutation, the quality (*fitness*) of offspring solutions is evaluated using the loss function (Eq. (9)) as fitness function (minimization is sought). Finally, we use tournament selection [66]) to form the population for the next generation.

We set CoGS to allow for plausibility constraints (\mathcal{P}) to be specified. If plausibility constraints are used, then mutation is restricted to plausible changes (e.g., the feature that represents age can only increase). If mutation makes a numerical feature obtain a value bigger than \max_i (resp., smaller than \min_i), then the value of that feature is set to \max_i (resp., \min_i).

CoGS is written in Python, and relies heavily on NumPy [67] for speeding up key computations. For example, the population is encoded as a NumPy matrix, and crossover and mutation are implemented with matrix operations.

5.4. Loss

We use the following loss to drive the search of counterfactual examples (where $f(\mathbf{z})$ and t are treated as integers):

$$\frac{1}{2}\gamma(\mathbf{z}, \mathbf{x}) + \frac{1}{2} \frac{\|\mathbf{z} - \mathbf{x}\|_0}{d} + \|f(\mathbf{z}) - t\|_0, \text{ where} \quad (9)$$

$$\gamma(\mathbf{z}, \mathbf{x}) = \frac{1}{d} \left(\sum_i^{d_1} \frac{|z_i - x_i|}{\max_i - \min_i} + \sum_j^{d_2} \|z_j - x_j\|_0 \right). \quad (10)$$

The function γ in the equation above is Gower's distance [19,46], where features indexed by i are numerical and those indexed by j are categorical (with values treated as integers); the maximal and minimal values of a numerical feature, \max_i and \min_i , can be taken from the (training) data set or, as done in our case, are provided as extra annotations of the data sets. The term $\|\mathbf{z} - \mathbf{x}\|_0/d$ promotes sparsity of intervention and, like Gower's distance, ranges from zero to one. The third and last term requires the execution of the machine learning model f , and simply returns zero when $f(\mathbf{z}) = t$ and one when $f(\mathbf{z}) \neq t$.

5.4.1. Incorporating robustness in the loss

To seek robust counterfactual examples, we make use of the notions described in Sec. 3 and Sec. 4. When optimizing for robustness to perturbations concerning \mathcal{C} , we use Definition 4, i.e., maximal \mathcal{C} -setbacks are computed on the fly for the candidate \mathbf{z} and their contribution is used to update the contribution of γ to the loss function. When optimizing for robustness to perturbations concerning \mathcal{K} , we compute the \mathcal{K} -robustness score with Eq. (8) and add $\frac{1}{2}(1 - \mathcal{K}\text{-robustness score})$ to the loss. In the results presented below, we use $m = 64$ to compute the \mathcal{K} -robustness score; an analysis on the impact of m is provided in B.3.

6. Preliminary results: choosing a suitable counterfactual search algorithm

This section reports on the benchmarking of the considered search algorithms, to identify an overall best. We repeat the execution of each algorithm five times and consider the best-found counterfactual example out of the five repetitions. We search for a counterfactual example for each \mathbf{x} in the test sets from the five cross-validation, for \mathbf{x} such that $f(\mathbf{x}) \neq t$. Since LORE takes much longer to execute than the other algorithms (see Fig. 5), we perform three repetitions instead of five, and consider only the first five \mathbf{x} in each test set of the five folds. Since only DiCE and CoGS support plausibility constraints, we do not use plausibility constraints in this comparison (a comparison between DiCE and COGS under plausibility constraints is provided in Appendix B.1).

6.1. Runtimes

Fig. 5 shows the runtime of the algorithms across the different data sets, irrespective of whether they succeed or fail to find a counterfactual example, i.e., a point for which f predicts t . The experiments were run on a cluster where the computing nodes can have slightly different CPUs, thus we invite to consider the order of magnitude of the runtimes rather than the exact numbers. The figure shows that, using random forest, CoGS and DiCE (configuration a) are the fastest algorithms (or, at least, have fastest implementations), but GrSp and NeMe are competitive. LORE is much slower to execute than the other algorithms. When using a neural network, inference times are generally faster, and CoGS, DiCE-a, GrSp and NeMe are competitive.

6.2. Success in discovering counterfactual examples

Table 4 shows the frequency with which the counterfactual search algorithms succeed in finding a counterfactual example, i.e., a point for which f predicts t . CoGS and the two variants of DiCE succeed systematically, whereas the other algorithms do not. GrSp performs third-best overall. In particular, GrSp always finds counterfactual examples on Hou, which is a data set with a single categorical feature. Since GrSp is intended to operate solely with numerical features, this result nicely supports the hypothesis that GrSp works well when (almost all) features are numerical. Although LORE supports both numerical and categorical features, it does not perform better than GrSp on most data sets; at least for the limited number of runs conducted with LORE due to excessive runtime, as explained before. Lastly, NeMe often performs substantially worse than all other algorithms.

Table 4

Mean \pm standard deviation across five cross-validation folds of the frequency with which the counterfactual search algorithms succeed in finding a counterfactual example. Plausibility constraints are not considered here because not all algorithms support them.

	Alg.	Cre	Inc	Hou	Pro	Rec
$f = \text{random forest}$	CoGS	1.00	1.00	1.00	1.00	1.00
	DiCE-a	1.00	1.00	1.00	1.00	1.00
	DiCE-b	1.00	1.00	1.00	1.00	1.00
	GrSp	0.46 ± 0.11	0.89 ± 0.06	1.00	0.86 ± 0.04	0.30 ± 0.15
	LORE	0.56 ± 0.20	0.20 ± 0.13	0.68 ± 0.20	0.24 ± 0.20	0.60 ± 0.38
	NeMe	0.08 ± 0.03	0.05 ± 0.02	0.04 ± 0.05	0.03 ± 0.01	0.14 ± 0.02
$f = \text{neural network}$	CoGS	1.00	1.00	1.00	1.00	1.00
	DiCE-a	1.00	1.00	1.00	1.00	1.00
	DiCE-b	1.00	1.00	1.00	1.00	1.00
	GrSp	0.87 ± 0.07	0.25 ± 0.04	1.00	0.51 ± 0.12	0.49 ± 0.10
	LORE	0.52 ± 0.20	0.28 ± 0.16	0.68 ± 0.10	0.76 ± 0.23	0.84 ± 0.23
	NeMe	0.14 ± 0.07	0.11 ± 0.03	0.09 ± 0.01	0.11 ± 0.04	0.51 ± 0.03

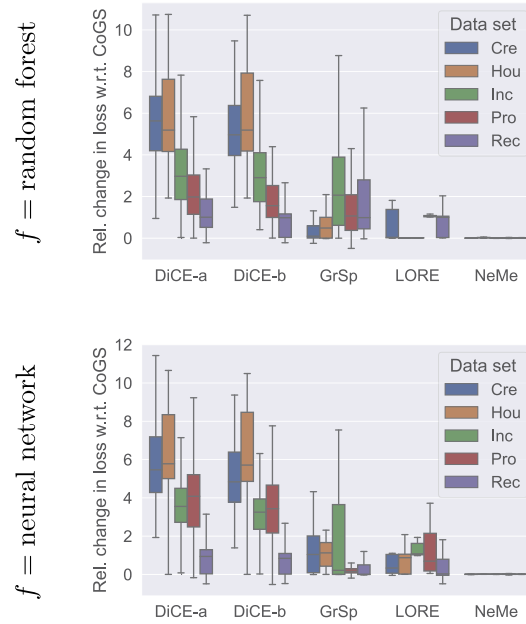


Fig. 6. Boxplots of relative change in loss with respect to CoGS for GrSp, LORE, and NeMe, on the different data sets and black-boxes, for success cases.

6.3. Quality of discovered counterfactual examples

As last part in our benchmarking effort, we consider what algorithm manages to produce near-optimal counterfactual examples (i.e., those with smallest loss). In particular, we report the relative change in loss for the best-found counterfactual example with respect to the loss obtained by CoGS, only for success cases. Since we consider only successes, the last term of the loss (Eq. (9)) is always null, i.e., $\|f(\mathbf{z}) - t\|_0 = 0$. The relative change in loss with respect to CoGS for another algorithm Alg is:

$$\frac{\mathcal{L}_{\text{Alg}}(\mathbf{z}) - \mathcal{L}_{\text{CoGS}}(\mathbf{z})}{\mathcal{L}_{\text{CoGS}}(\mathbf{z})}.$$

Fig. 6 shows the relative change in loss of DiCE, GrSp, LORE, and NeMe with respect to CoGS. DiCE, GrSp and LORE typically (but not always) find points that have larger loss than those found by CoGS. NeMe performs very similarly to CoGS, however NeMe seldom succeeds (cf., Table 4). This suggests that NeMe can explore a small neighborhood of \mathbf{x} particularly well, but fails if counterfactual examples are relatively distant from \mathbf{x} .

Table 5

Mean \pm standard deviation of the frequency with which the best-found (among five search repetitions) counterfactual example when not accounting for robustness is accidentally robust w.r.t. \mathcal{C} or \mathcal{K} . For numerical features, we consider them to match in value if they are within a tolerance level (Tol.) of 1%, 5% or 10% of the range for that feature.

	Robustness	Tol.	Cre	Inc	Hou	Pro	Rec
$f = \text{random forest}$	Only \mathcal{C}	1%	0.40 ± 0.06	0.02	0.76 ± 0.10	0.53 ± 0.05	0.27 ± 0.06
		5%	0.42 ± 0.07	0.04 ± 0.02	0.84 ± 0.09	0.57 ± 0.06	0.37 ± 0.07
		10%	0.43 ± 0.07	0.05 ± 0.02	0.85 ± 0.09	0.58 ± 0.06	0.40 ± 0.09
	Only \mathcal{K}	1%	0.37 ± 0.01	0.06 ± 0.02	0.33 ± 0.24	0.26 ± 0.05	0.04 ± 0.04
		5%	0.44 ± 0.03	0.40 ± 0.08	0.63 ± 0.17	0.37 ± 0.06	0.08 ± 0.03
		10%	0.46 ± 0.04	0.58 ± 0.07	0.67 ± 0.16	0.46 ± 0.07	0.12 ± 0.02
	Both \mathcal{C}, \mathcal{K}	1%	0.23 ± 0.04	0.00	0.21 ± 0.21	0.19 ± 0.06	0.03 ± 0.03
		5%	0.27 ± 0.03	0.00	0.54 ± 0.21	0.26 ± 0.05	0.06 ± 0.04
		10%	0.30 ± 0.05	0.00	0.60 ± 0.19	0.34 ± 0.06	0.08 ± 0.04
$f = \text{neural network}$	Only \mathcal{C}	1%	0.25 ± 0.12	0.01 ± 0.01	0.96 ± 0.02	0.87 ± 0.05	0.50 ± 0.08
		5%	0.27 ± 0.12	0.02 ± 0.01	0.97 ± 0.02	0.89 ± 0.05	0.56 ± 0.05
		10%	0.29 ± 0.11	0.02 ± 0.01	0.97 ± 0.02	0.89 ± 0.05	0.57 ± 0.04
	Only \mathcal{K}	1%	0.13 ± 0.07	0.35 ± 0.02	0.07 ± 0.05	0.08 ± 0.06	0.01
		5%	0.26 ± 0.08	0.52 ± 0.03	0.80 ± 0.12	0.42 ± 0.19	0.01 ± 0.01
		10%	0.39 ± 0.02	0.70 ± 0.04	0.93 ± 0.04	0.58 ± 0.14	0.02 ± 0.02
	Both \mathcal{C}, \mathcal{K}	1%	0.02 ± 0.02	0.00	0.07 ± 0.05	0.06 ± 0.06	0.00 ± 0.01
		5%	0.06 ± 0.03	0.00	0.69 ± 0.09	0.38 ± 0.18	0.01 ± 0.01
		10%	0.12 ± 0.08	0.00	0.93 ± 0.04	0.52 ± 0.14	0.01 ± 0.02

6.4. Conclusion of benchmarking

The results show that, overall, CoGS performs best. DiCE (in particular, DiCE-a) is the closest competitor in terms of speed and success rates, but the algorithm finds counterfactual examples that are substantially more distant from \mathbf{x} (i.e., have larger loss) than those found by CoGS. GrSp has good runtime and generally finds closer counterfactual examples (i.e., lower loss) than DiCE, but it remains inferior to CoGS, both in terms of distance (loss) and success rate. LORE has worse success rate than GrSp, and NeME worse of all. Therefore, we use CoGS for the following experiments on robustness.

We remark that DiCE, like CoGS, supports the specification of plausibility constraints. We show that CoGS performs better than DiCE also under plausibility constraints in Appendix B.1.

7. Experimental results: robustness

We proceed with presenting the experimental results regarding robustness to perturbations in \mathcal{C} , \mathcal{K} , and jointly. We focus on results that allow us to answer what we believe to be important research questions: (RQ1) *Do we need to account for robustness to discover robust counterfactual examples?* (RQ2) *Does a lack of robustness compromise the feasibility of correcting perturbations with additional intervention?* (RQ3) *Are robust counterfactual explanations advantageous in terms of additional intervention cost?* These questions are addressed, in order, in the next subsections. Because of space limitations, a number of additional results is reported in Appendix B, including runtime taken to account for robustness w.r.t. \mathcal{C} and \mathcal{K} , and the effect of varying m when computing the \mathcal{K} -robustness score. We now account for plausibility constraints \mathcal{P} in all of the following experiments. We remark that in all our experiments, CoGS always succeeded in discovering a counterfactual example for which f predicts t , except for having a mean success rate of 99% (st.dev. of 1%) on the Rec data set when f is implemented as a neural network.

7.1. (RQ1) Do we need to account for robustness to discover robust counterfactual examples?

Table 5 shows the frequency with which robust counterfactual examples are discovered accidentally. To realize this, we compare the best-found counterfactual example that is discovered by CoGS when robustness is *not* accounted for, and the one that is found when \mathcal{C} - or \mathcal{K} -robustness is accounted for (as indicated in Sec. 5.4.1). We take the frequency by which the two match as indication of whether robust counterfactual examples can be discovered by accident. Since numerical feature values may differ only slightly between two best-found counterfactual examples, we consider the values to match if they are sufficiently close to each other, according to a tolerance level of 1%, 5%, or 10% of the range of that feature. As reasonable to expect, the results show that the larger the tolerance level, the more a \mathbf{z}^* discovered when not accounting for robustness matches the respective one that is discovered when accounting for robustness. In general, the result depends on the data set in consideration, and also (albeit arguably less so) on whether random forest or a neural network is used as black-box model f .

For brevity, we now focus on the tolerance level of 5% and random forest. On Inc, best-found counterfactual examples rarely match with those discovered when accounting for \mathcal{C} -robustness (4% on average for the tolerance of 5%), while the

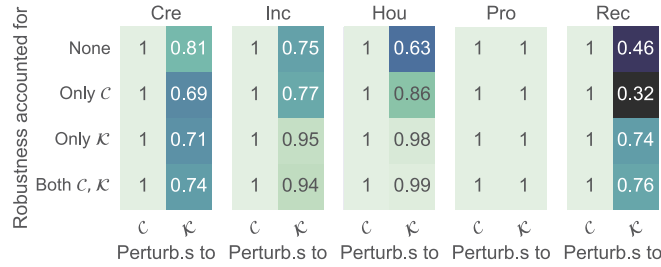
Uniformly-distributed perturbations, $f = \text{random forest}$ Normally-distributed perturbations, $f = \text{random forest}$ 

Fig. 7. Mean frequency with which a plausible additional intervention exists, to contrast the perturbations and reach the intended counterfactual example (uniformly-distributed categorical changes and normally- or uniformly-distributed numerical changes, $f = \text{random forest}$). Darker colors represent worse cases.

vice versa happens on Hou (84% on average for the same tolerance). For \mathcal{K} -robustness like for \mathcal{C} -robustness, the result depends on the data set. Importantly, the data sets where the frequencies are high for \mathcal{C} -robustness and \mathcal{K} -robustness are not necessarily the same. On Inc, best-found counterfactual examples are rarely optimal under \mathcal{C} -setbacks, but can often match with counterfactual examples discovered when penalizing low \mathcal{K} -robustness scores (40% on average for the tolerance of 5%). This should not be surprising because \mathcal{C} - and \mathcal{K} -robustness are orthogonal to each other under the assumption of feature independence. The last row shows how often best-found counterfactual examples happen to be both robust to perturbations to \mathcal{C} and \mathcal{K} . The frequencies are clearly always lower than for the previous triplets of rows. Hou is the only data set for which the frequency of discovering a counterfactual example that happens to be both robust w.r.t. \mathcal{C} and \mathcal{K} by chance is relatively large (e.g., above 50% for the tolerance of 5%).

When using the neural network instead of random forest, the trends mentioned before remain the same, but the specific magnitudes can differ. For example, the accidental discovery of robust counterfactual examples w.r.t. \mathcal{C} and/or \mathcal{K} is lower on Cre with the neural network compared to random forest, but the opposite holds for Hou (with some exceptions, e.g., the tolerance level of 1% when both \mathcal{C} - and \mathcal{K} -robustness are sought).

Overall, this result indicates that, except for lucky cases (e.g., Hou with f being the neural network), it is unlikely to discover robust counterfactual examples by chance. Hence, if one wishes to achieve robustness, the search must be explicitly instructed to that end. In the next sections, we investigate whether achieving robustness can actually be important.

7.2. (RQ2) Does a lack of robustness compromise the feasibility of correcting perturbations with additional intervention?

At this point, current works on the robustness of counterfactual explanations typically consider the extent by which robustness helps preventing the invalidation of counterfactual explanations (see Sec. 9). In other words, they consider whether the point \mathbf{z}' that is given by perturbing the best-found counterfactual example is still classified as t . For completeness, we report on this in B.2. Current works do not, however, consider whether an additional intervention that allows to correct the perturbation and obtain t might exist.

Figs. 7 and 8 show the frequency with which achieving the intended counterfactual explanation remains possible after random perturbations take place. The frequency is computed by applying, for each counterfactual explanation outcome of the search, 100 perturbations that are sampled uniformly at random from the categorical possibilities for categorical features, and normally (with st.dev. of 0.1) or uniformly within the numerical intervals for numerical features, as defined in p. We note that similar results are obtained between choosing random forest or a neural network as f .

As expected, it is always possible to contrast \mathcal{C} -setbacks, because these happen along the direction of intervention. Instead, perturbations concerning \mathcal{K} can lead to a \mathbf{z}' such that no further plausible intervention exists to reach the originally intended counterfactual example. We do not report a result for perturbations concerning both \mathcal{C} and \mathcal{K} at the same time

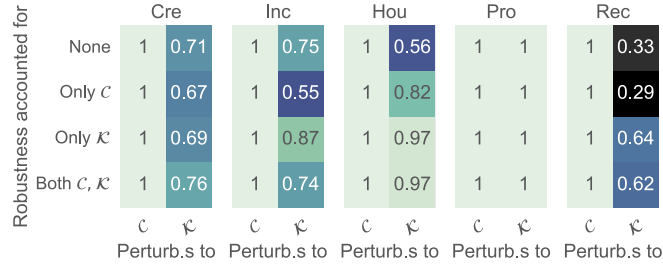
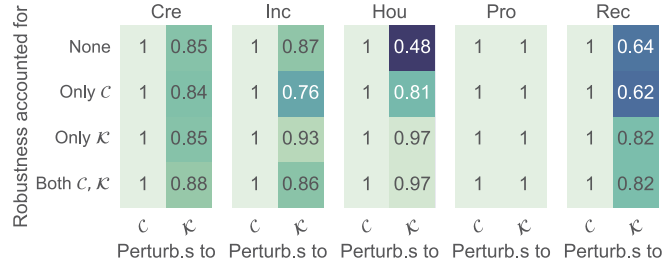
Uniformly-distributed perturbations, $f = \text{neural network}$ Normally-distributed perturbations, $f = \text{neural network}$ 

Fig. 8. Mean frequency with which a plausible additional intervention exists, to contrast the perturbations and reach the intended counterfactual example (uniformly-distributed categorical changes and normally- or uniformly-distributed numerical changes, $f = \text{neural network}$). Darker colors represent worse cases.

because, by construction, it is the same as the result for perturbations concerning only \mathcal{K} . Like for the results of Sec. 7.1, the extent by which perturbations to \mathcal{K} reduce the possibility for additional intervention depends on the data set. On Pro, all perturbations can be contrasted by an additional intervention because there are no plausibility constraints (see Table 1). Conversely, on Rec, perturbations to \mathcal{K} can often make it impossible to reach the originally-intended counterfactual example, unless \mathcal{K} -robustness is accounted for. In fact, accounting for \mathcal{K} -robustness generally improves the chances that additional intervention is possible, at times substantially (e.g., on Inc, Hou, and Rec). Cre represents the only exception to this, as accounting for \mathcal{K} -robustness performs similar (or sometimes worse) than accounting for none. This suggests that the decision boundary learned by f on this data set may not be very smooth, making the use of the \mathcal{K} -robustness score a too coarse approximation to be helpful. Generally, accounting for perturbations to \mathcal{C} alone does not help achieving substantial robustness to perturbations to \mathcal{K} , except for on Hou. This suggests that, on Hou, f learns decision boundaries that incorporate interesting interactions between certain features. Importantly, accounting for \mathcal{C} -robustness *together* with accounting for \mathcal{K} -robustness does not substantially compromise the gains obtained by accounting for \mathcal{K} -robustness alone, even though perturbations to \mathcal{C} always admit additional intervention. Overall, these results show that accounting for robustness can be crucial to ensure that, if perturbations happen, additional intervention to obtain t remains possible.

7.3. (RQ3) Are robust counterfactual explanations advantageous in terms of additional intervention cost?

We present the following results in terms of a *relative cost*, namely, the ratio between the cost of intervention to reach the intended \mathbf{z} when random perturbations take place (i.e., initial the cost of reaching \mathbf{z} from \mathbf{x} plus the cost of reaching \mathbf{z} from the perturbed \mathbf{z}'), and the *ideal cost*, i.e., the cost incurred in complete absence of perturbations (i.e., the cost of reaching \mathbf{z} from \mathbf{x}). We compute this relative cost when the notions of robustness are or are not accounted for. The ideal cost is computed when *not* accounting robustness. The cost is modeled by $\frac{1}{2}\gamma(\mathbf{z}, \mathbf{x}) + \frac{1}{2}\frac{\|\mathbf{z} - \mathbf{x}\|_0}{d}$ (i.e., the first part of Eq. (9)). Moreover, if $f(\mathbf{z}') = t$, we assume no additional intervention to be needed, and thus the additional cost is zero and the relative cost is 1.

Figs. 9 and 10 (for random forest and neural network, respectively) show that when no robustness is accounted for (the left-most triplets of boxes in each plot), the relative cost can become dramatically large. In other words, additional intervention to correct the perturbations can be extremely costly. Whether the relative cost increases mostly due to perturbations to \mathcal{C} (blue boxes) or to \mathcal{K} (orange boxes) depends on the data set. For example, perturbations to \mathcal{K} have the largest effect on Rec, while those to \mathcal{C} have the largest effect on Inc (by far), across types of distribution and types of f . For both the random forest and the neural network, the relative cost ranges from around $5\times$ or $10\times$ the ideal cost, up to over $100\times$ (Inc, perturbations to \mathcal{C}) when not accounting for robustness.

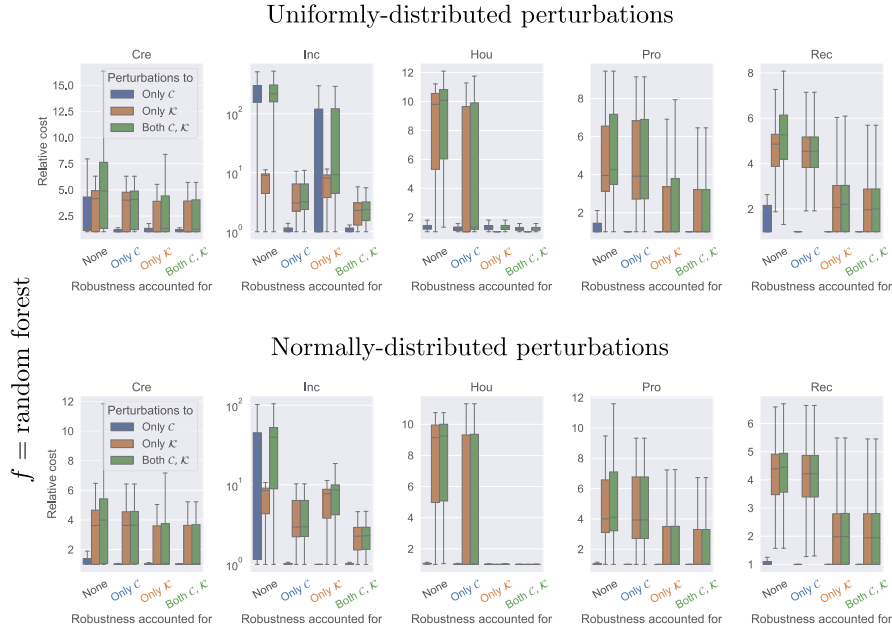


Fig. 9. Cost in terms of different configurations of accounting for robustness and under different perturbations, relative to the ideal cost (with random forest). Due to perturbations, the relative cost for when no notion of robustness is accounted for (label None) is typically much larger than the one for when the right notion of robustness is accounted for (matching color between box and label). The vertical axis for Inc is in logarithmic scale.

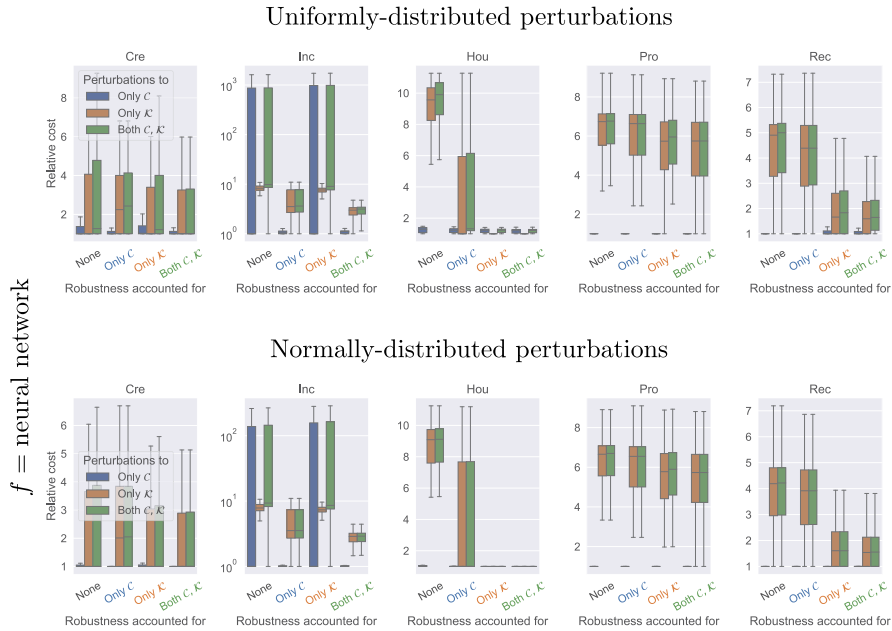


Fig. 10. Cost in terms of different configurations of accounting for robustness and under different perturbations, relative to the ideal cost (with neural network). Due to perturbations, the relative cost for when no notion of robustness is accounted for (label None) is typically much larger than the one for when the right notion of robustness is accounted for (matching color between box and label). The vertical axis for Inc is in logarithmic scale.

When one accounts for the notion of robustness that is meant to deal with the respective type of perturbation, the relative cost often decreases substantially. Accounting for \mathcal{C} -robustness (second blue box from the left in each plot) counters perturbations to \mathcal{C} very well on all the data sets. On Inc in particular, the relative cost improves by two orders of magnitude. As found in Sec. 7.2, accounting for perturbations to \mathcal{K} with the \mathcal{K} -robustness score can remain insufficient, as it can be observed on Cre and Inc for both types of f . Again, this is likely a limitation of using a simple heuristic such as the \mathcal{K} -robustness score to deal with \mathcal{K} -robustness. Accounting for robustness w.r.t. \mathcal{C} (resp., \mathcal{K}) does not, in general, lead to smaller relative cost under perturbations to \mathcal{K} (resp., \mathcal{C}). We confirm this general trend with statistical testing in Appendix C. Lastly,

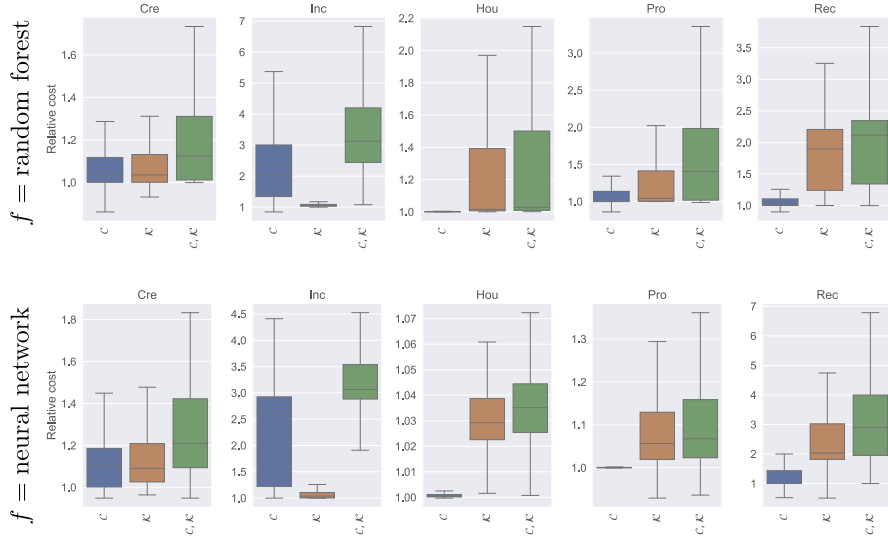


Fig. 11. Cost of accounting for robustness relative to not accounting for robustness (i.e., ideal cost) when no perturbations take place. Note that the (rare) relative costs smaller than 1 are due to a lack of optimality of the search algorithm.

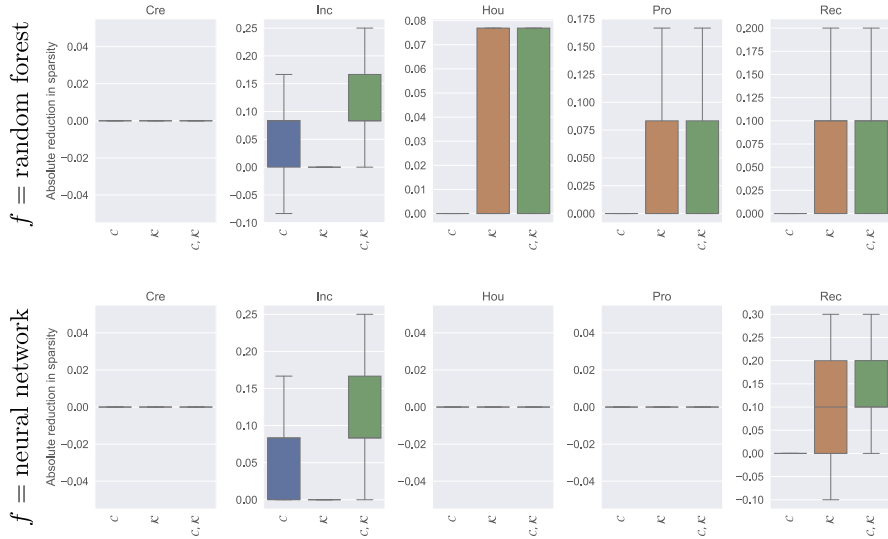


Fig. 12. Reduction in sparsity (relative to the number of features) caused by accounting for robustness. Note the different vertical axes.

accounting for both \mathcal{C} - and \mathcal{K} -robustness (right-most triplets of boxes in each plot) offers protection (lower relative cost) from situations in which both types of perturbations take place. In general across data sets and types of f , the distribution of relative costs for when perturbations to both \mathcal{C} and \mathcal{K} take place and both \mathcal{C} - and \mathcal{K} -robustness are accounted for (right-most green box in each plot) is better than the distribution for when the same perturbations take place but no notion of robustness is accounted for (left-most green box in each plot).

Since the ideal cost is computed when no notion of robustness is accounted for, part of the relative costs for when robustness is accounted for comes from the fact that robust counterfactual examples are generally farther away from \mathbf{x} than non-robust ones. Fig. 11 shows the cost increase that comes solely from accounting for robustness on the considered data sets and types of f , without any perturbation taking place. We remark that values smaller than 1 happen only because the discovered counterfactual examples can be suboptimal. Importantly, we find that the cost when accounting for robustness is between $1\times$ and $7\times$ the ideal cost, i.e., when not accounting for robustness. In general, this is significantly smaller than the increase incurred when perturbations take place and robustness is not accounted for, as reported before (generally between $5\times$ and $10\times$ the ideal cost, with up to $100\times$).

Lastly, Fig. 12 shows what part of the cost increase comes from counterfactuals becoming less sparse. For certain data sets (e.g., Cre for both random forest and neural network), robust counterfactual explanations are as sparse as non-robust ones. In general, however, robust counterfactual explanations tend to be less sparse, depending on the choice of f and the

type of robustness that is accounted for. The reduction in sparsity can be moderate or substantial. For example, less than 10% more of the features need to change to account for \mathcal{K} perturbations on Hou with random forest (i.e., approximately one feature). Instead, up to 30% more of the features need to change to account for \mathcal{K} perturbations on Rec with the neural network (i.e., approximately three features). The fact that sparsity decreases when seeking robust counterfactuals is a natural consequence of adopting a linearization of the objectives (see Sec. 5.4). Thus, the reduction in sparsity can be tackled by tuning the weight attributed to the L_0 -norm in Eq. (9).

These results confirm that even though robust counterfactual explanations are, in principle, more costly to pursue than non-robust ones, if random perturbations take place, robust counterfactual explanations require much less additional intervention than non-robust ones.

8. Discussion

Our experimental results provide a positive answer to all three research questions. In general, counterfactual explanations are *not* robust, be it in terms of the features whose value is prescribed to be changed (\mathcal{C} -robustness), or those whose value is prescribed to be kept as is (\mathcal{K} -robustness). Moreover, non-robust counterfactual explanations are more susceptible to make it impossible for the user to remedy perturbations by additional intervention, and the cost of additional intervention is larger for non-robust counterfactual explanations than for robust ones. Ultimately, it is clear that accounting for robustness is important.

Our experimental results suggest that accounting for robustness for features in \mathcal{C} tempers perturbations to \mathcal{C} , and similarly, accounting for robustness for features in \mathcal{K} tempers perturbations to \mathcal{K} . Moreover, even though f can learn non-linear feature interactions, accounting for \mathcal{C} (or \mathcal{K}) has limited effect on contrasting perturbations to \mathcal{K} (resp., \mathcal{C}). Only in some cases (e.g., on Hou), robustness w.r.t. \mathcal{C} has substantial repercussions on the effect of perturbations to \mathcal{K} or vice versa.

In addition to this, even if a counterfactual search algorithm does not guarantee that the discovered counterfactual example will be optimal, we experimentally see that incorporating our Definition 4 into the loss (Sec. 5.4.1) produces a strong resilience to additional cost (Sec. 7.3) for perturbations to the features in \mathcal{C} . Besides being effective, implementation of Definition 4 is also efficient (see B.3.2).

What our results also show is that seeking robustness with respect to features in \mathcal{K} is problematic. This is because of Proposition 1 and the fact that features in \mathcal{K} are not aligned with the direction of intervention. Thus, we proposed to control for \mathcal{K} -robustness using an approximation, i.e., the \mathcal{K} -robustness score. We found that seeking counterfactual examples that maximize the \mathcal{K} -robustness score are often but not always sufficient to obtain a good resilience to perturbations to the features in \mathcal{K} . Moreover, the \mathcal{K} -robustness score requires to sample (and evaluate with f) multiple points, which is far more expensive than computing Definition 4. Therefore, future work should consider whether a better method can be used than the \mathcal{K} -robustness score. For example, if information on f is available, that information may be used to provide guarantees on the neighborhood of \mathbf{z} (see, e.g., Theorem 2 in [40] for linear f).

The assumption that features are independent is simplistic but often made in literature, because only a small number of works assume a causal model is available (e.g., [47,48]). Under the assumption of feature independence, as done here, one models the neighborhood of a counterfactual example with a box (under L_1) or a hyper-sphere (under L_2). However, if certain features have a causal dependency on other features, this neighborhood morphs into other, possibly very complex shapes (e.g., when this dependency is not linear). Importantly, if feature i depends on j , then one cannot change j without having that i implicitly changes too. Similarly, a perturbation happening to j would implicitly alter i . As our framework currently assumes independence, it is important to study to what extent separation between \mathcal{C} and \mathcal{K} remains possible and meaningful. For many real-world problems, it is reasonable to expect that there exist groups of features that are truly independent from other groups of features. Thus, the study of robustness for \mathcal{C} and \mathcal{K} could be carried out at a higher level, i.e., of feature groups in future work.

There is a number of further aspects worth mentioning when one wishes to implement a research work like on counterfactual explanations into practice, including this work. For example, we use the L_1 -norm within Gower's distance to measure intervention cost. In fact, literature works typically choose one distance measure (e.g., ours, or Gower's with L_2 -norm instead, or other variants, see Sec. 9). Of course, a realistic implementation of intervention cost may need to be more refined, e.g., by mixing different types of norms. Similarly, one might wish to use different distributions to sample meaningful perturbations (as opposed to only uniform or only normal as done in our synthetic experiments), and different functions to define the maximal extent of perturbation, which may e.g., account for the distribution of feature values. For example, for denser areas of feature i , p_i^+ and p_i^- should be smaller than for less dense areas. Other desiderata may need to be included when seeking counterfactuals in practice (see, e.g., [49,50]), including accounting for multiple types of robustness of the same time, such as those related to uncertainties of f [51,52].

Lastly, we made subjective choices to define perturbations (\mathbf{p}) and plausibility constraints (\mathcal{P}) in the data sets. We made these choices as best as we could, based on reading the meta-information in web sources and the papers that describe the data sets. We have no doubt that domain experts would make much better choices than ours. Nevertheless, we argue that this is not an important limitation because, as long as the community agrees that our choices are reasonable, they suffice to provide a sensible test bed for benchmarking robustness. Hopefully, other researchers will find our annotations to be useful for future experiments on the robustness of counterfactual explanations. Similarly, we hope that other researchers will find CoGS to be an interesting algorithm to benchmark against.

9. Related work

A number of works in literature propose several new desiderata that are largely orthogonal to our notions of robustness but can be important to enhance the practical usability of counterfactual explanations. For example, Dandl et al. [49] consider, besides proximity of \mathbf{z} to \mathbf{x} according to different distances, whether other training points \mathbf{x}' are sufficiently close to \mathbf{z} for it to reasonably belong to the training data distribution. A similar desideratum is considered in [21] and [53]; the latter work employs neural autoencoders to that end. [54] remarks the importance of sparsity for explanations, with the concepts of *pertinent negatives* (the minimal features that should be different to (more) confidently predict the given class) and *pertinent positives* (the minimal features that help correctly identifying the class). Laugel et al. [36,50] require that \mathbf{z} can always be reached from a training point \mathbf{x}' without having to cross the decision boundary of f , for \mathbf{z} not to be the result of an artifact in the decision boundary of f . In [47] and [22], counterfactual explanations are studied through the lens of causality. For recent surveys on counterfactual explanations, the reader is referred to [55,16,56].

We now focus on works that deal with some notion of robustness and/or perturbations explicitly. Artelt et al. [57] present theoretical results on the effect of perturbations (e.g., under linear f), evaluate the effect of different type of perturbations (Gaussian, uniform, masking) with three classifiers, and find that counterfactual explanations that obey plausibility constraints are more robust than counterfactual explanations that do not. Differently from us, Artelt et al. do not consider sparsity and do not optimize for robustness. The work by Karimi et al. [48] extends [47] to consider possible uncertainties in causal modeling. In [17], it is shown that a malicious actor can, in principle, jointly optimize small perturbations and the model f such that, when applying the perturbations to points of a specific group (e.g., white males), the respective counterfactual explanations are much less costly than normal (in fact, counterfactual explanations are conceptually similar to adversarial examples, see, e.g., [58–60]). Some works consider forms of robustness of counterfactual explanations with respect to changes of f (e.g., whether \mathbf{z} is still classified as t if f' is used instead of f) [51,61] or updates to f (e.g., after data distribution shift of temporal or geospatial nature) [62,52]. In [63], robustness of counterfactual explanations is studied in the context of differentially-private support vector machines. Dominguez et al. [40] consider whether counterfactual explanations remain valid in presence of uncertainty on \mathbf{x} , and also account for causality. We also note that Dominguez et al. consider a neighborhood of uncertainty around \mathbf{x} which is akin to Definition 1; in fact, such sort of neighborhoods is common tools in post-hoc explanation methods, e.g., the Anchor explainer by [64] seeks representative points for a class by assessing that the prediction of f for the points in their neighborhood is the same. Zhang et al. [65] propose a counterfactual search method based on linear programming that works for neural networks with ReLU activations; this work can be seen through the lens of robustness in that the method produces *regions* of points that share the desired class. Finally, contemporary to our work, Fokkema et al. [26] provide important theoretical results that counterfactual explanations (and other XAI methods such as feature attribution ones) can be dramatically different when small perturbations are applied to the starting point \mathbf{x} (or, in general, point to explain).

To the best of our knowledge, there exists no other work prior to ours that attempts to exploit sparsity when assessing robustness, although sparsity is an important property for counterfactual explanations. Moreover, existing works typically consider whether robustness helps preventing counterfactual explanations from becoming invalid, while we further consider that additional intervention may be possible, and assess the associated cost.

10. Conclusion

Counterfactual explanations can help us understand how black-box AI systems reach certain decisions, as well as what intervention is possible to alter such decisions. For counterfactual explanations to be most useful in practice, we studied how they can be made *robust* to adverse perturbations that may naturally happen due to unforeseen circumstances, to ensure that the intervention they prescribe remains valid, and potential additional intervention cost that may be needed remains limited. We presented novel notions of robustness, which concern adverse perturbations to the features that a counterfactual explanation prescribes to change (\mathcal{C} -robustness) and to keep as they are (\mathcal{K} -robustness), respectively. We have annotated five existing data sets with reasonable perturbations and plausibility constraints and developed a competitive counterfactual search algorithm to search for (robust) counterfactual explanations. Our experimental results show that, most often than not, counterfactual explanations do not happen to be robust by accident. Consequently, if adverse perturbations take place, counterfactual explanations may require a much larger cost to be realized than anticipated, or even make it impossible for the user to achieve recourse. Our definitions of robustness can be incorporated in the search process, and robust counterfactual explanations can be discovered. We have shown that \mathcal{C} -robustness can be accounted for efficiently and effectively, while the same is not always true for \mathcal{K} -robustness. This aspect should be taken into account when choosing what counterfactual explanation is best for the user. Overall, robust counterfactual explanations are resilient against invalidation and require much smaller additional intervention to contrast perturbations.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Table A.1

Hyper-parameter settings considered for tuning random forest.

Name	Options
No. trees	{50, 500}
Min. samples split	{2, 8}
Max. features	{ \sqrt{d} , d }

Table A.2

Hyper-parameter settings considered for tuning the neural network.

Name	Options
Learning rate	{0.0001, 0.01}
Max. iterations	{200, 1000}
Solver	{Adam, SGD}

Table A.3Test accuracy of hyper-parameter-tuned random forests acting as black-box models f for the considered data sets across five-fold cross-validation.

Fold	Cre	Inc	Hou	Pro	Rec
0	0.71	0.86	0.93	0.79	0.80
1	0.78	0.82	0.90	0.77	0.82
2	0.78	0.79	0.91	0.78	0.78
3	0.74	0.82	0.91	0.82	0.77
4	0.76	0.83	0.97	0.78	0.80
Avg.	0.76	0.83	0.93	0.79	0.80

Table A.4Test accuracy of hyper-parameter-tuned neural networks acting as black-box models f for the considered data sets across five-fold cross-validation.

Fold	Cre	Inc	Hou	Pro	Rec
0	0.74	0.83	0.94	0.62	0.78
1	0.78	0.82	0.93	0.70	0.79
2	0.73	0.80	0.91	0.69	0.75
3	0.78	0.82	0.91	0.77	0.78
4	0.74	0.82	0.96	0.72	0.79
Avg.	0.75	0.82	0.93	0.70	0.78

Data availability

The data is available at the github repository linked in the abstract.

Acknowledgements

We thank dr. Stef C. Maree for insightful early discussions. This work made use of the Dutch national e-infrastructure with the support of the SURF Cooperative using grant no. EINF-2512. Funding: This publication is part of the project Robust Counterfactual Explanations (with project number EINF-2512) of the research program Computing Time on National Computer Facilities which is (partly) financed by the Dutch Research Council (NWO).

Appendix A. Hyper-parameter optimization of random forest and neural network

To obtain a black-box model f for a given cross-validation fold, we train a random forest model or a neural network (a multi layer perceptron for classification) optimized with grid-search hyper-parameter tuning (with five-fold cross-validation on the training set). The hyper-parameter settings we considered are listed in Tables A.1 and A.2, all other being Scikit-learn's default (v. 1.0.1). For random forest, we one-hot encode categorical features when training and querying the random forest model (see the code `robust_cfe/blackbox_with_preproc.py`). For the neural network, we additionally scale numerical features to have mean of zero and standard deviation of one.

The performance of tuned random forest on all folds is shown in Table A.3, the respective one for the neural network is shown in Table A.4.

Table B.1

Mean \pm standard deviation across five cross-validation folds of the frequency with which CoGS and the two variants of DiCE succeed in finding a counterfactual example under plausibility constraints.

	Alg.	Cre	Inc	Hou	Pro	Rec
$f = \text{r.f.}$	CoGS	1.00	1.00	1.00	1.00	1.00
	DiCE-a	1.00	1.00	1.00	1.00	0.99 ± 0.06
	DiCE-b	1.00	1.00	1.00	0.20 ± 4.00	1.00 ± 0.05
$f = \text{n.n.}$	CoGS	1.00	1.00	1.00	1.00	1.00 ± 0.04
	DiCE-a	1.00	1.00 ± 0.04	1.00	1.00	1.00
	DiCE-b	1.00	1.00	1.00	1.00	0.80 ± 4.00

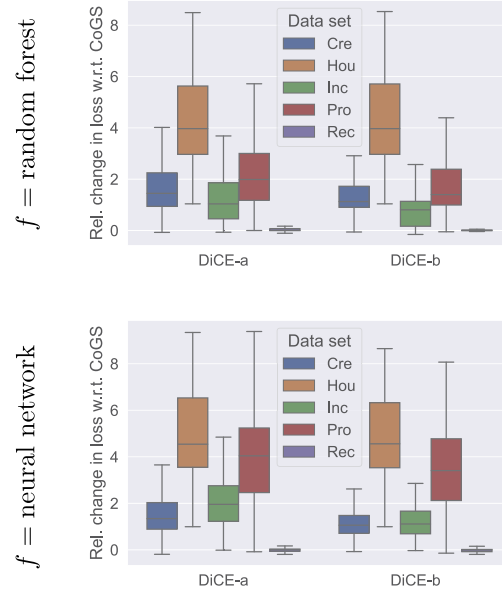


Fig. B.1. Boxplots of relative change in loss with respect to CoGS for the two variants of DiCE when using plausibility constraints, on the different data sets and black-boxes, for success cases.

Appendix B. Additional results

We provide additional results. These are (i) a further comparison between DiCE and CoGS when plausibility constraints are enforced, (ii) the (possibly non-permanent) invalidity caused by perturbations, as typically done in the literature of robustness, (iii) and effect of increasing m for the computation of the \mathcal{K} -robustness score.

B.1. DiCE vs. CoGS with plausibility constraints

Table B.1 shows the average and the standard deviation of the success rate (how many times a counterfactual of the desired class is found) for the two variants of DiCE and of CoGS when plausibility constraints are active. The algorithms are comparable in that they succeed in most cases. However, DiCE-b can perform substantially worse in two cases, i.e., data set Pro when using the random forest, and data set Rec when using the neural network.

Lastly, Fig. B.1 shows the relative change in loss obtained by the algorithms. As it can be seen, DiCE comes close to the performance of CoGS only on Rec. Thus, overall, CoGS remains superior to DiCE also when plausibility constraints are enforced.

We attribute this to the fact that, differently from CoGS, DiCE is inherently designed to discover a diverse set of counterfactuals instead of a single and closest-possible counterfactual.

B.2. Invalidity of counterfactual explanations

We now show whether the fact that best-found counterfactual explanations are typically not robust is associated with a greater chance that perturbations can make them invalid, i.e., such that $f(\mathbf{z}') \neq t$ where \mathbf{z}' is the point to which \mathbf{z}^* is shifted by the perturbation. Here, we do not consider whether additional intervention may or may not be possible. Fig. B.2 shows the average frequency with which perturbations cause invalidity. The frequencies are computed by applying, to each

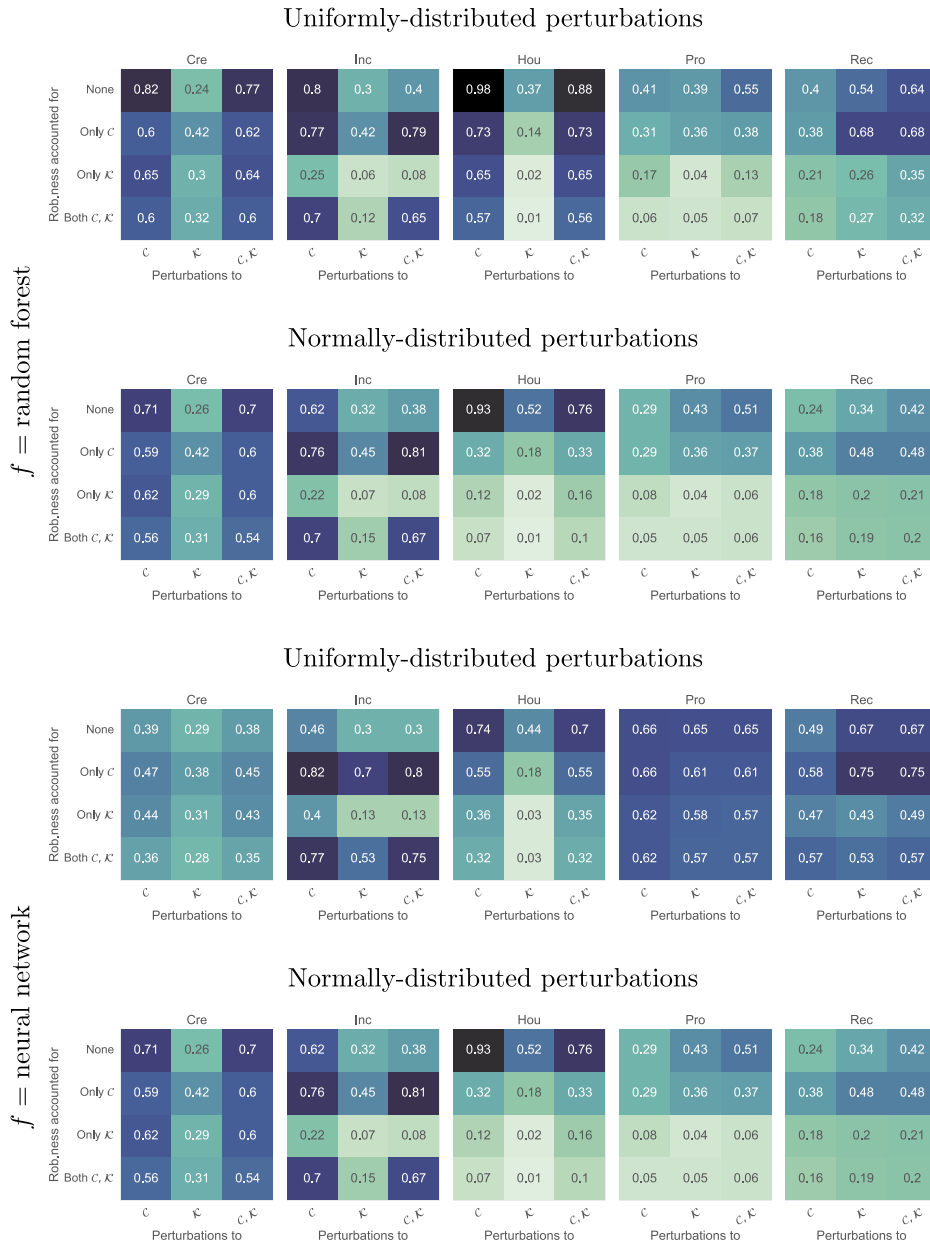


Fig. B.2. Mean frequency of invalidity of counterfactual explanations under different types of perturbations and when accounting for different types of robustness. Darker colors represent worse scenarios, i.e., larger average invalidity.

discovered counterfactual example, 100 perturbations that are sampled uniformly at random for categorical features (from the categorical possibilities) and uniformly or normally (with st.dev. of 0.1) for numerical features (within the numerical intervals). The figure shows that when no notion of robustness is accounted for, perturbations generally have a larger chance of causing invalidity of the counterfactual explanation.

Regarding perturbations to (features in) \mathcal{C} (i.e., \mathcal{C} -setbacks), recall that accounting for the respective notion of robustness is intended to provide counterfactual explanations with minimal additional intervention cost, the maximal \mathcal{C} -setback were to happen. Ideally, the returned counterfactual example should still be optimal, i.e., as near to \mathbf{x} as possible, which means that the example is on the border of the decision boundary of f . Thus, under optimality guarantees, $f(\mathbf{z}^* + \mathbf{w}^{\mathcal{C},s}) \neq t$ (Proposition 2); This means that any \mathcal{C} -setback should result in invalidity (i.e., all entries for perturbations to \mathcal{C} should report 1). This does not always happen in Fig. B.2 because CoGS does not guarantee to discover optimal counterfactual examples and, thus, in many cases the returned example is not on the boundary, and the \mathcal{C} -setback is too small to cross the boundary. The frequency of this phenomenon depends on the data set. Also, while accounting for robustness w.r.t. \mathcal{C} should not, in theory, decrease invalidity rate but only make additional intervention less costly, as confirmed in Sec. 7.3),

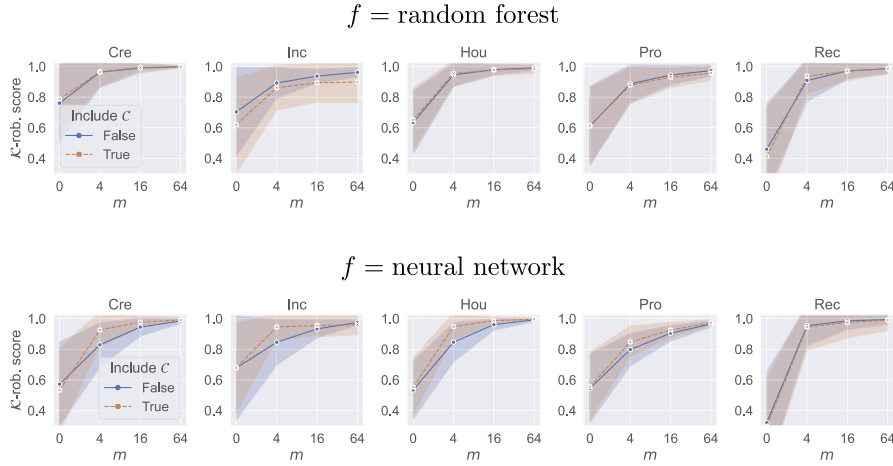


Fig. B.3. Approximated ground-truth \mathcal{K} -robustness scores (using 1000 samples) for increasing m , to determine what value of m is needed for good robustness to perturbations to \mathcal{K} . Shaded areas represent standard deviations.

we find that accounting for robustness w.r.t. \mathcal{C} lowers invalidity rate on Hou (e.g., most evident for both types of f with normally-distributed perturbations).

When \mathcal{K} -robustness is accounted for, the best-found counterfactual explanation is supposed to be in a region such that the decision boundary is relatively loose with respect to the features in \mathcal{K} . Consequently, accounting for \mathcal{K} -robustness *should*, in fact, counter invalidity, as we do not wish risking that it becomes impossible to carry out additional intervention due to the plausibility constraints. The figure shows that, in general, there can be a substantial gain in lowering invalidity by accounting for \mathcal{K} -robustness. At times, accounting for \mathcal{K} -robustness allows to reach almost zero invalidity, see the cell that corresponds to robustness for \mathcal{K} and perturbations to \mathcal{K} , on Inc, Hou, and Pro, for both types of f and sampling distributions. However, it is not always the case that \mathcal{K} -robustness helps, due to the heuristic nature of the \mathcal{K} -robustness score: see, e.g., Cre.

Lastly, we observe that the frequency of invalidity can raise when both notions of robustness are accounted for at the same time (e.g., on Inc for uniformly-distributed perturbations when using the neural network). Note that this is not necessarily a problem because invalidity from perturbations to \mathcal{C} is expected to be high, as the goal of robustness w.r.t. \mathcal{C} is to be able to minimize additional intervention cost.

B.3. Setting m for \mathcal{K} -robustness

We report results on setting the hyper-parameter m for computing \mathcal{K} -robustness scores (see Eq. (8)). In particular, we run CoGS accounting for \mathcal{K} -robustness in the loss function, for $m \in \{0, 4, 16, 64\}$. Note that using $m = 0$ corresponds to *not* accounting for \mathcal{K} -robustness.

B.3.1. Achieved \mathcal{K} -robustness

We consider how increasing m improves \mathcal{K} -robustness, using an approximated *ground-truth*. We approximate the ground-truth of the true \mathcal{K} -robustness by calculating the \mathcal{K} -robustness score over 1000 samples over the counterfactual example discovered using a specific m .

Fig. B.3 shows the results obtained for this experiment. We also consider the case in which \mathcal{C} -robustness is accounted for. If \mathcal{K} -robustness is not accounted for ($m = 0$), then the (approximated ground-truth) \mathcal{K} -robustness of the discovered counterfactual examples can be quite low, see, e.g., Rec for random forest (score approximately of 0.4) and neural network (score below 0.4). As soon as a few samples are considered ($m = 4$), the \mathcal{K} -robustness increases substantially (see, e.g., Cre). Further increasing m has diminishing returns (note that m is increased exponentially). Accounting for \mathcal{C} -robustness is largely orthogonal, meaning, it has no effect in terms of \mathcal{K} -robustness.

B.3.2. Additional required runtime

Fig. B.4 shows the additional runtime incurred between runs of CoGS that account for some notion of robustness and runs that do not account for it, in particular for increasing m in the calculation of the \mathcal{K} -robustness score. The figure shows that accounting for \mathcal{C} -robustness comes at no significant extra cost in runtime. This follows from the fact that we can use Definition 4 and thus only need to compute the maximal \mathcal{C} -setback. Conversely, accounting for \mathcal{K} -robustness can come at a relatively large additional cost in runtime, which appears to be linear in m (note that m grows exponentially in the plots). Fortunately, the experimental results of B.3 suggest that small values of m are often sufficient to obtain good \mathcal{K} -robustness scores.

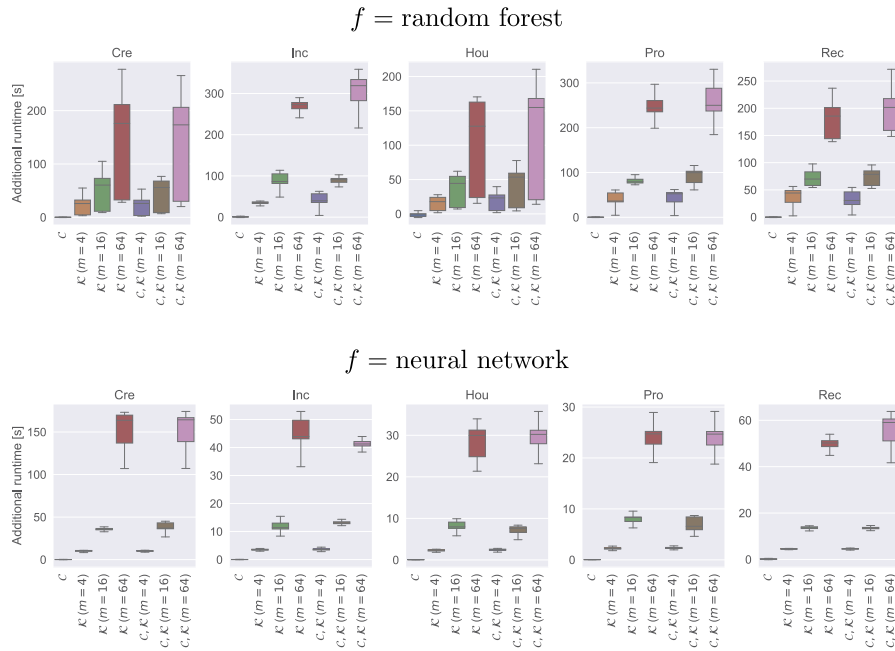


Fig. B.4. Additional runtime of CoGS for different configurations of accounting for robustness with respect to the runtime when not accounting for robustness.

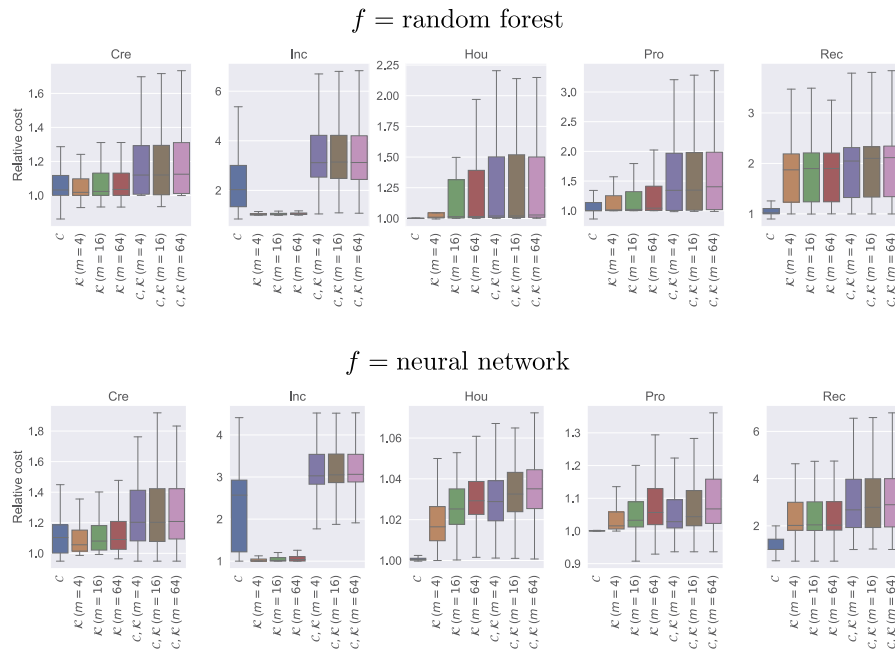


Fig. B.5. Cost of accounting for robustness relative to not accounting for robustness (i.e., ideal cost) when no perturbations take place for different values of m . Note that the (rare) relative costs smaller than 1 are due to a lack of optimality of the search algorithm.

B.3.3. Additional cost from accounting for robustness

Fig. B.5 expands on the results reported in Fig. 11 by including different values of m . We do not find major differences based on the setting of m for computing the \mathcal{K} -robustness score, except for the tails of the respective distributions on Hou, and slightly less so on Pro (for both types of f). Accounting for C - and \mathcal{K} -robustness at the same time leads to larger costs than accounting for only one of the two, as it is reasonable to expect. On average, the cost that comes from accounting for robustness alone is limited (up to $6.5\times$ the ideal cost, see Inc), especially in light of the results found for

Table C.1

Result of pairwise comparison on the effect of accounting for different types of robustness for data set Cre under different perturbations (both random forest and neural network, both uniform and normal sampling distributions). The displayed p -values are obtained with the Mann-Whitney U test under Holm-Bonferroni correction and post Kruskal-Wallis test rejecting the null hypothesis with p -value $\ll 0.01$.

Cre, perturbations to \mathcal{C}				
Robustness	None	Only \mathcal{C}	Only \mathcal{K}	Both \mathcal{C}, \mathcal{K}
None	1.000	0.000	0.000	0.000
Only \mathcal{C}	0.000	1.000	0.000	0.125
Only \mathcal{K}	0.000	0.000	1.000	0.000
Both \mathcal{C}, \mathcal{K}	0.000	0.125	0.000	1.000

Cre, perturbations to \mathcal{K}				
Robustness	None	Only \mathcal{C}	Only \mathcal{K}	Both \mathcal{C}, \mathcal{K}
None	1.000	0.520	0.000	0.000
Only \mathcal{C}	0.520	1.000	0.000	0.000
Only \mathcal{K}	0.000	0.000	1.000	0.912
Both \mathcal{C}, \mathcal{K}	0.000	0.000	0.912	1.000

Cre, perturbations to \mathcal{C} and \mathcal{K}				
Robustness	None	Only \mathcal{C}	Only \mathcal{K}	Both \mathcal{C}, \mathcal{K}
None	1.000	0.000	0.000	0.000
Only \mathcal{C}	0.000	1.000	0.000	0.000
Only \mathcal{K}	0.000	0.000	1.000	0.003
Both \mathcal{C}, \mathcal{K}	0.000	0.000	0.003	1.000

Table C.2

Result of pairwise comparison on the effect of accounting for different types of robustness for data set Inc under different perturbations (both random forest and neural network, both uniform and normal sampling distributions). The displayed p -values are obtained with the Mann-Whitney U test under Holm-Bonferroni correction and post Kruskal-Wallis test rejecting the null hypothesis with p -value $\ll 0.01$.

Inc, perturbations to \mathcal{C}				
Robustness	None	Only \mathcal{C}	Only \mathcal{K}	Both \mathcal{C}, \mathcal{K}
None	1.000	0.000	0.000	0.000
Only \mathcal{C}	0.000	1.000	0.000	0.000
Only \mathcal{K}	0.000	0.000	1.000	0.000
Both \mathcal{C}, \mathcal{K}	0.000	0.000	0.000	1.000

Inc, perturbations to \mathcal{K}				
Robustness	None	Only \mathcal{C}	Only \mathcal{K}	Both \mathcal{C}, \mathcal{K}
None	1.000	0.000	0.000	0.000
Only \mathcal{C}	0.000	1.000	0.000	0.000
Only \mathcal{K}	0.000	0.000	1.000	0.000
Both \mathcal{C}, \mathcal{K}	0.000	0.000	0.000	1.000

Inc, perturbations to \mathcal{C} and \mathcal{K}				
Robustness	None	Only \mathcal{C}	Only \mathcal{K}	Both \mathcal{C}, \mathcal{K}
None	1.000	0.000	0.000	0.000
Only \mathcal{C}	0.000	1.000	0.000	0.000
Only \mathcal{K}	0.000	0.000	1.000	0.000
Both \mathcal{C}, \mathcal{K}	0.000	0.000	0.000	1.000

when perturbations take place, described in Sec. 7.3 (additional intervention due to perturbations can lead to $100\times$ times larger costs for non-robust counterfactual explanations, see Inc on Fig. 9).

Appendix C. Statistical significance

We report the statistical significance for the results displayed in Sec. 7.3. For each data set and type of perturbation, we perform the Kruskal-Wallis tests (since we cannot assume normality) to determine whether significant differences are present between the relative cost induced by applying the different notion of robustness. In all cases, the outcome of the test is that significant differences are present (p -value $\ll 0.01$). Next, we perform post-hoc pairwise comparisons with the Mann-Whitney-U test to assess whether one notion of robustness protects from the considered perturbation significantly differently than another. The result of the pairwise comparison analysis is shown in Tables C.1 to C.5.

Table C.3

Result of pairwise comparison on the effect of accounting for different types of robustness for data set Hou under different perturbations (both random forest and neural network, both uniform and normal sampling distributions). The displayed p -values are obtained with the Mann-Whitney U test under Holm-Bonferroni correction and post Kruskal-Wallis test rejecting the null hypothesis with p -value $\ll 0.01$.

Hou, perturbations to \mathcal{C}				
Robustness	None	Only \mathcal{C}	Only \mathcal{K}	Both \mathcal{C}, \mathcal{K}
None	1.000	0.000	0.000	0.000
Only \mathcal{C}	0.000	1.000	0.261	0.006
Only \mathcal{K}	0.000	0.261	1.000	0.083
Both \mathcal{C}, \mathcal{K}	0.000	0.006	0.083	1.000

Hou, perturbations to \mathcal{K}				
Robustness	None	Only \mathcal{C}	Only \mathcal{K}	Both \mathcal{C}, \mathcal{K}
None	1.000	0.000	0.000	0.000
Only \mathcal{C}	0.000	1.000	0.000	0.000
Only \mathcal{K}	0.000	0.000	1.000	0.070
Both \mathcal{C}, \mathcal{K}	0.000	0.000	0.070	1.000

Hou, perturbations to \mathcal{C} and \mathcal{K}				
Robustness	None	Only \mathcal{C}	Only \mathcal{K}	Both \mathcal{C}, \mathcal{K}
None	1.000	0.000	0.000	0.000
Only \mathcal{C}	0.000	1.000	0.000	0.000
Only \mathcal{K}	0.000	0.000	1.000	0.008
Both \mathcal{C}, \mathcal{K}	0.000	0.000	0.008	1.000

Table C.4

Result of pairwise comparison on the effect of accounting for different types of robustness for data set Pro under different perturbations (both random forest and neural network, both uniform and normal sampling distributions). The displayed p -values are obtained with the Mann-Whitney U test under Holm-Bonferroni correction and post Kruskal-Wallis test rejecting the null hypothesis with p -value $\ll 0.01$.

Pro, perturbations to \mathcal{C}				
Robustness	None	Only \mathcal{C}	Only \mathcal{K}	Both \mathcal{C}, \mathcal{K}
None	1.000	0.000	0.000	0.000
Only \mathcal{C}	0.000	1.000	0.000	0.008
Only \mathcal{K}	0.000	0.000	1.000	0.000
Both \mathcal{C}, \mathcal{K}	0.000	0.008	0.000	1.000

Pro, perturbations to \mathcal{K}				
Robustness	None	Only \mathcal{C}	Only \mathcal{K}	Both \mathcal{C}, \mathcal{K}
None	1.000	0.000	0.000	0.000
Only \mathcal{C}	0.000	1.000	0.000	0.000
Only \mathcal{K}	0.000	0.000	1.000	0.760
Both \mathcal{C}, \mathcal{K}	0.000	0.000	0.760	1.000

Pro, perturbations to \mathcal{C} and \mathcal{K}				
Robustness	None	Only \mathcal{C}	Only \mathcal{K}	Both \mathcal{C}, \mathcal{K}
None	1.000	0.000	0.000	0.000
Only \mathcal{C}	0.000	1.000	0.000	0.000
Only \mathcal{K}	0.000	0.000	1.000	0.014
Both \mathcal{C}, \mathcal{K}	0.000	0.000	0.014	1.000

On Cre under \mathcal{K} -perturbations (middle part of Table C.1), accounting for robustness w.r.t. \mathcal{C} is not significantly different than not accounting for any notion of robustness (p -value = 0.52 > 0.01); similarly, accounting for robustness w.r.t. \mathcal{K} induces the same relative cost as accounting for robustness w.r.t. both \mathcal{C} and \mathcal{K} . On Hou under \mathcal{C} -perturbations (top part of Table C.3), accounting for \mathcal{C} is not significantly different than accounting for \mathcal{K} (p -value = 0.52 > 0.01), while accounting for \mathcal{K} is not significantly different than accounting for both \mathcal{C} and \mathcal{K} (p -value = 0.083 > 0.01). When perturbations happen to both \mathcal{C} and \mathcal{K} on Pro and Rec (bottom part of respective tables), accounting for \mathcal{K} is not significantly different than accounting for both \mathcal{C} and \mathcal{K} (p -value = 0.014 > 0.01 and p -value = 0.271 > 0.01, respectively). In general, the results match what can be seen in Figs. 9 and 10. Also, we note that in the majority of the cases, accounting for one notion of robustness is significantly different than accounting for another (or for none).

Table C.5

Result of pairwise comparison on the effect of accounting for different types of robustness for data set Rec under different perturbations (both random forest and neural network, both uniform and normal sampling distributions). The displayed p -values are obtained with the Mann-Whitney U test under Holm-Bonferroni correction and post Kruskal-Wallis test rejecting the null hypothesis with p -value $\ll 0.01$.

Rec, perturbations to \mathcal{C}				
Robustness	None	Only \mathcal{C}	Only \mathcal{K}	Both \mathcal{C}, \mathcal{K}
None	1.000	0.000	0.000	0.000
Only \mathcal{C}	0.000	1.000	0.000	0.000
Only \mathcal{K}	0.000	0.000	1.000	0.115
Both \mathcal{C}, \mathcal{K}	0.000	0.000	0.115	1.000

Rec, perturbations to \mathcal{K}				
Robustness	None	Only \mathcal{C}	Only \mathcal{K}	Both \mathcal{C}, \mathcal{K}
None	1.000	0.000	0.000	0.000
Only \mathcal{C}	0.000	1.000	0.000	0.000
Only \mathcal{K}	0.000	0.000	1.000	0.651
Both \mathcal{C}, \mathcal{K}	0.000	0.000	0.651	1.000

Rec, perturbations to \mathcal{C} and \mathcal{K}				
Robustness	None	Only \mathcal{C}	Only \mathcal{K}	Both \mathcal{C}, \mathcal{K}
None	1.000	0.000	0.000	0.000
Only \mathcal{C}	0.000	1.000	0.000	0.000
Only \mathcal{K}	0.000	0.000	1.000	0.271
Both \mathcal{C}, \mathcal{K}	0.000	0.000	0.271	1.000

References

- [1] J.H. Friedman, Greedy function approximation: a gradient boosting machine, *Ann. Stat.* 29 (2001) 1189–1232.
- [2] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, LightGBM: a highly efficient gradient boosting decision tree, *Adv. Neural Inf. Process. Syst.* 30 (2017) 3146–3154.
- [3] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature* 521 (7553) (2015) 436–444.
- [4] M. Belkin, D. Hsu, S. Ma, S. Mandal, Reconciling modern machine-learning practice and the classical bias–variance trade-off, *Proc. Natl. Acad. Sci.* 116 (32) (2019) 15849–15854.
- [5] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, I. Sutskever, Deep double descent: where bigger models and more data hurt, *J. Stat. Mech. Theory Exp.* 2021 (12) (2021) 124003.
- [6] B. Goodman, S. Flaxman, European Union regulations on algorithmic decision-making and a “right to explanation”, *AI Mag.* 38 (3) (2017) 50–57.
- [7] A. Jobin, M. Ienca, E. Vayena, The global landscape of AI ethics guidelines, *Nat. Mach. Intell.* 1 (9) (2019) 389–399.
- [8] A. Adadi, M. Berrada, Peeking inside the black-box: a survey on eXplainable Artificial Intelligence (XAI), *IEEE Access* 6 (2018) 52138–52160.
- [9] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, *ACM Comput. Surv.* 51 (5) (2018) 1–42.
- [10] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, *Nat. Mach. Intell.* 1 (5) (2019) 206–215.
- [11] M.T. Ribeiro, S. Singh, C. Guestrin, “Why should I trust you?” explaining the predictions of any classifier, in: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1135–1144.
- [12] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: *Advances in Neural Information Processing Systems*, 2017, pp. 4768–4777.
- [13] B. Kim, R. Khanna, O.O. Koyejo, Examples are not enough, learn to criticize! Criticism for interpretability, in: *Advances in Neural Information Processing Systems*, vol. 29.
- [14] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, J.K. Su, This looks like that: deep learning for interpretable image recognition, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [15] S. Wachter, B. Mittelstadt, C. Russell, Counterfactual explanations without opening the black box: automated decisions and the GDPR, *Harv. J. Law Technol.* 31 (2017) 841.
- [16] I. Stepin, J.M. Alonso, A. Catala, M. Pereira-Fariña, A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence, *IEEE Access* 9 (2021) 11974–12001.
- [17] D. Slack, S. Hilgard, H. Lakkaraju, S. Singh, Counterfactual explanations can be manipulated, *arXiv preprint*, arXiv:2106.02666.
- [18] S. Barocas, A.D. Selbst, M. Raghavan, The hidden assumptions behind counterfactual explanations and principal reasons, in: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020, pp. 80–89.
- [19] J.C. Gower, A general coefficient of similarity and some of its properties, *Biometrics* (1971) 857–871.
- [20] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, F. Giannotti, Local rule-based explanations of black box decision systems, *arXiv preprint*, arXiv:1805.10820.
- [21] S. Sharma, J. Henderson, J. Ghosh, CERTIFAI: a common framework to provide explanations and analyse the fairness and robustness of black-box models, in: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 2020, pp. 166–172.
- [22] R.K. Mothilal, A. Sharma, C. Tan, Explaining machine learning classifiers through diverse counterfactual explanations, in: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020, pp. 607–617.

- [23] T. Laugel, M.-J. Lesot, C. Marsala, X. Renard, M. Detyniecki, Comparison-based inverse classification for interpretability in machine learning, in: *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Springer, 2018, pp. 100–111.
- [24] M.T. Keane, B. Smyth, Good counterfactuals and where to find them: a case-based technique for generating counterfactuals for explainable AI (XAI), in: *International Conference on Case-Based Reasoning*, Springer, 2020, pp. 163–178.
- [25] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, *Neural Netw.* 2 (5) (1989) 359–366.
- [26] H. Fokkema, R. de Heide, T. van Erven, Attribution-based explanations that provide recourse cannot be robust, arXiv preprint, arXiv:2205.15834.
- [27] U. Grömping, South German credit data: correcting a widely used data set, report 4/2019, Reports in Mathematics, Physics and Chemistry, Department II, Beuth University of Applied Sciences Berlin, 2019, <https://archive.ics.uci.edu/ml/datasets/South+German+Credit+%28UPDATE%29>.
- [28] H. Hofmann, Statlog German credit data, [https://archive.ics.uci.edu/ml/datasets/Statlog+\(German+Credit+Data\)](https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)), 1994.
- [29] R. Kohavi, B. Becker, Census income, <https://archive.ics.uci.edu/ml/datasets/adult>, 1996.
- [30] R. Kohavi, Scaling up the accuracy of naive-Bayes classifiers: a decision-tree hybrid, in: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, vol. 96, 1996, pp. 202–207.
- [31] D. Harrison Jr., D.L. Rubinfeld, Hedonic housing prices and the demand for clean air, *J. Environ. Econ. Manag.* 5 (1) (1978) 81–102.
- [32] M. Carlisle, Racist data destruction?, <https://medium.com/@docintangible/racist-data-destruction-113e3eff54a8>, 2019.
- [33] A.A. Imran, M.S. Rahim, T. Ahmed, Mining the productivity data of the garment industry, *Int. J. Bus. Intell. Data Min.* 19 (3) (2021) 319–342.
- [34] J. Larson, S. Mattu, L. Kirchner, J. Angwin, How we analyzed the COMPAS recidivism algorithm, <https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm>, 2016.
- [35] M. Kearns, S. Neel, A. Roth, Z.S. Wu, Preventing fairness gerrymandering: auditing and learning for subgroup fairness, in: *International Conference on Machine Learning*, PMLR, 2018, pp. 2564–2572.
- [36] T. Laugel, M.-J. Lesot, C. Marsala, X. Renard, M. Detyniecki, The dangers of post-hoc interpretability: unjustified counterfactual explanations, arXiv preprint, arXiv:1907.09294.
- [37] F. Ding, M. Hardt, J. Miller, L. Schmidt, Retiring adult: new datasets for fair machine learning, in: *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.
- [38] W. La Cava, J.H. Moore, Genetic programming approaches to learning fair classifiers, in: *Proceedings of the 2020 Genetic and Evolutionary Computation Conference, GECCO '20, Association for Computing Machinery*, New York, NY, USA, 2020, pp. 967–975.
- [39] M. Virgolin, A. De Lorenzo, F. Randone, E. Medvet, M. Wahde, Model learning with personalized interpretability estimation (ml-pie), in: *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '21, Association for Computing Machinery*, New York, NY, USA, 2021, pp. 1355–1364.
- [40] R. Dominguez-Olmedo, A.H. Karimi, B. Schölkopf, On the adversarial robustness of causal algorithmic recourse, in: *Proceedings of the 39th International Conference on Machine Learning*, vol. 162, PMLR, 2022, pp. 5324–5342.
- [41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in Python, *J. Mach. Learn. Res.* 12 (2011) 2825–2830.
- [42] R. Guidotti, A. Monreale, F. Giannotti, D. Pedreschi, S. Ruggieri, F. Turini, Factual and counterfactual explanations for black box decision making, *IEEE Intell. Syst.* 34 (6) (2019) 14–23.
- [43] J.A. Nelder, R. Mead, A simplex method for function minimization, *Comput. J.* 7 (4) (1965) 308–313.
- [44] F. Gao, L. Han, Implementing the Nelder-Mead simplex algorithm with adaptive parameters, *Comput. Optim. Appl.* 51 (1) (2012) 259–277.
- [45] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, Í. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: fundamental algorithms for scientific computing in Python, *Nat. Methods* 17 (2020) 261–272.
- [46] M. D’Orazio, Distances with mixed type variables some modified Gower’s coefficients, arXiv preprint, arXiv:2101.02481.
- [47] A.-H. Karimi, J. Von Kügelgen, B. Schölkopf, I. Valera, Algorithmic recourse under imperfect causal knowledge: a probabilistic approach, arXiv preprint, arXiv:2006.06831.
- [48] A.-H. Karimi, B. Schölkopf, I. Valera, Algorithmic recourse: from counterfactual explanations to interventions, in: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21, Association for Computing Machinery*, New York, NY, USA, 2021, pp. 353–362.
- [49] S. Dandl, C. Molnar, M. Binder, B. Bischl, Multi-objective counterfactual explanations, in: *International Conference on Parallel Problem Solving from Nature*, Springer, 2020, pp. 448–469.
- [50] T. Laugel, M.-J. Lesot, C. Marsala, X. Renard, M. Detyniecki, Unjustified classification regions and counterfactual explanations in machine learning, in: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2019, pp. 37–54.
- [51] M. Pawelczyk, K. Broelemann, G. Kasneci, On counterfactual explanations under predictive multiplicity, in: *Conference on Uncertainty in Artificial Intelligence*, PMLR, 2020, pp. 809–818.
- [52] K. Rawal, E. Kamar, H. Lakkaraju, Algorithmic recourse in the wild: understanding the impact of data and model shifts, arXiv preprint, arXiv:2012.11788.
- [53] A. Van Looveren, J. Klaise, Interpretable counterfactual explanations guided by prototypes, arXiv preprint, arXiv:1907.02584.
- [54] A. Dhurandhar, P.-Y. Chen, R. Luss, C.-C. Tu, P. Ting, K. Shanmugam, P. Das, Explanations based on the missing: towards contrastive explanations with pertinent negatives, in: *Advances in Neural Information Processing Systems*, vol. 31.
- [55] S. Verma, J. Dickerson, K. Hines, Counterfactual explanations for machine learning: a review, arXiv preprint, arXiv:2010.10596.
- [56] A.-H. Karimi, G. Barthe, B. Schölkopf, I. Valera, A survey of algorithmic recourse: contrastive explanations and consequential recommendations, *ACM Computing Surveys (CSUR)*.
- [57] A. Artelt, V. Vaquet, R. Velioglu, F. Hinder, J. Brinkrolf, M. Schilling, B. Hammer, Evaluating robustness of counterfactual explanations, in: *IEEE Symposium Series on Computational Intelligence*, IEEE, 2021, pp. 01–09.
- [58] M. Pawelczyk, C. Agarwal, S. Joshi, S. Upadhyay, H. Lakkaraju, Exploring counterfactual explanations through the lens of adversarial examples: a theoretical and empirical analysis, arXiv preprint, arXiv:2106.09992.
- [59] V. Ballet, X. Renard, J. Aigrain, T. Laugel, P. Frossard, M. Detyniecki, Imperceptible adversarial attacks on tabular data, arXiv preprint, arXiv:1911.03274.
- [60] T. Freiesleben, The intriguing relation between counterfactual explanations and adversarial examples, *Minds Mach.* (2021) 1–33.
- [61] A. Ferrario, M. Loi, The robustness of counterfactual explanations over time, *IEEE Access*.
- [62] A. Ferrario, M. Loi, A series of unfortunate counterfactual events: the role of time in counterfactual explanations, arXiv preprint, arXiv:2010.04687.
- [63] R. Mochaourab, S. Sinha, S. Greenstein, P. Papapetrou, Robust counterfactual explanations for privacy-preserving SVM, in: *International Conference on Machine Learning (ICML 2021)*, Workshop on Socially Responsible Machine Learning, 2021.
- [64] M.T. Ribeiro, S. Singh, C. Guestrin, Anchors, High-precision model-agnostic explanations, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.
- [65] X. Zhang, A. Solar-Lezama, R. Singh, Interpreting neural network judgments via minimal, stable, and symbolic corrections, in: *Advances in Neural Information Processing Systems*, vol. 31.

Appendix references

- [66] B.L. Miller, D.E. Goldberg, Genetic algorithms, tournament selection, and the effects of noise, *Complex Syst.* 9 (3) (1995) 193–212.
- [67] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane, J.F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T.E. Oliphant, Array programming with NumPy, *Nature* 585 (7825) (2020) 357–362.