HAL
open science

The Dangers of Post-hoc Interpretability: Unjustified Counterfactual Explanations

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, Marcin Detyniecki

To cite this version:

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, Marcin Detyniecki. The Dangers of Post-hoc Interpretability: Unjustified Counterfactual Explanations. Twenty-Eighth International Joint Conference on Artificial Intelligence IJCAI-19, Aug 2019, Macao, Macau SAR China. pp.2801-2807, 10.24963/ijcai.2019/388 . hal-02275308

HAL Id: hal-02275308

https://hal.sorbonne-universite.fr/hal-02275308

Submitted on 30 Aug 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The Dangers of Post-hoc Interpretability: Unjustified Counterfactual Explanations

Thibault Laugel[1*], **Marie-Jeanne Lesot**[1], **Christophe Marsala**[1], **Xavier Renard**[2] and **Marcin Detyniecki**[1,2,3]
[1]Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, LIP6, F-75005 Paris, France
[2]AXA, Paris, France
[3]Polish Academy of Science, IBS PAN, Warsaw, Poland
thibault.laugel@lip6.fr

Abstract

Post-hoc interpretability approaches have been proven to be powerful tools to generate explanations for the predictions made by a trained black-box model. However, they create the risk of having explanations that are a result of some artifacts learned by the model instead of actual knowledge from the data. This paper focuses on the case of counterfactual explanations and asks whether the generated instances can be *justified*, i.e. continuously connected to some ground-truth data. We evaluate the risk of generating unjustified counterfactual examples by investigating the local neighborhoods of instances whose predictions are to be explained and show that this risk is quite high for several datasets. Furthermore, we show that most state of the art approaches do not differentiate justified from unjustified counterfactual examples, leading to less useful explanations.

1 Introduction

Among the soaring number of methods proposed to generate explanations for a machine learning classifier, post-hoc interpretability aproaches [Guidotti *et al.*, 2018b] have been the subject of debates recently in the community [Rudin, 2018]. By generating explanations for the predictions of a trained predictive model without using any knowledge about it whatsoever (treating it as a *black-box*), these systems are inherently flexible enough to be used in any situation (model, task...) by any user, which makes them popular today in various industries. However, their main downside is that, under these assumptions, there is no guarantee that the built explanations are faithful to the original data that were used to train the model.

This is in particular the case for counterfactual example approaches that, based on counterfactual reasoning (see e.g. [Bottou *et al.*, 2013]), aim at answering the question: *given a classifier and an observation, how is the prediction altered when the observation changes?* In the context of classification, they identify the minimal perturbation required to change the predicted class of a given observation: a user is thus able to understand what features locally impact the prediction and therefore how it can be changed. Among interpretability methods, counterfactual examples have been shown to be useful solutions [Wachter *et al.*, 2018] that can be easily understood and thus directly facilitate decisions for a user.

However, without any guarantee on existing data, counterfactual examples in the post-hoc paradigm are vulnerable to issues raised by the robustness of the classifier, leading to explanations that are arguably not satisfying in the context of interpretability. More generally, this paper argues that a crucial property a counterfactual example should satisfy is that it should be connected to the training data of the classifier. Formally, we define this relation using the notion of connectedness and argue that there should be a continuous path between a counterfactual and an instance from the training data, as discussed and defined in Section 3.

This paper aims at showing that generating such post-hoc counterfactual explanations can be difficult. The contributions are the following:

- We propose an intuitive desideratum for more relevant counterfactual explanations, based on ground-truth labelled data, that helps generating better explanations.
- We design a test to highlight the risk of having undesirable counterfactual examples disturb the generation of counterfactual explanations.
- We apply this test to several datasets and classifiers and show that the risk of generating undesirable counterfactual examples is high.
- Additionally, we design a second test and show that state of the art post-hoc counterfactual approaches may generate unjustified explanations.

Section 2 is devoted to discussing the state of the art of post-hoc interpretability and counterfactual explanations, as well as presenting studies related to this work. Section 3 proposes a definition for ground-truth backed counterfactual explanations that we use in this paper. In Section 4, the central procedure of this paper is proposed, which aims at assessing the risk of generating unjustified counterfactual explanations, and experimental results are shown. Finally, the vulnerability of methods from the literature to this risk is evaluated in Section 5.

*Contact Author

2 Background

This section presents some existing interpretability methods and analyses that have been already conducted.

2.1 Post-hoc Interpretability

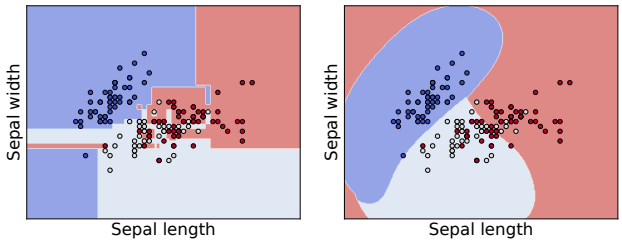
In order to build explanations for predictions made by a trained black-box model, post-hoc interpretability approaches generally rely on sampling instances and labelling them with the model to gather information about its behavior, either locally around a prediction [Ribeiro *et al.*, 2016] or globally [Craven and Shavlik, 1996]. These instances are then used to approximate the decision function of the black-box and build understandable explanations, either using a surrogate model (e.g. a linear model [Ribeiro *et al.*, 2016] or a decision tree [Hara and Hayashi, 2016]), or by computing meaningful coefficients [Turner, 2015; Lundberg and Lee, 2017]). Other methods rely on using specific instances to build explanations using comparison to relevant neighbors, such as prototype-based approaches [Kim *et al.*, 2014] and counterfactual explanation approaches.

Instead of simply identifying important features, the latter aim at finding the minimal perturbation required to alter the prediction of a given observation. A counterfactual explanation is thus a data instance, close to this observation, that is predicted to belong to a different class. This form of explanation provides a user tangible explanations that are directly understandable and actionable, which can be seen as an advantage. This can be opposed to other explanations using feature importance vectors, which are arguably harder to use and to understand for a non-expert user [Wachter *et al.*, 2018]. Several formalizations of the counterfactual problem can be found in the literature, depending on the formulation of the inverse problem and on the used distance. For instance, [Laugel *et al.*, 2018a] (resp. [Guidotti *et al.*, 2018a]) look for the L_2 (resp. L_0)-closest instance of an other class, while [Lash *et al.*, 2017] try to find the instance that has the highest probability of belonging to another class within a certain maximum distance. Another example is the formulation as a tradeoff between the L_1 closest instance and a certain classification score target [Wachter *et al.*, 2018].

2.2 Comparison of Post-hoc Interpretability Approaches

The post-hoc paradigm, and in particular the need for post-hoc approaches to use instances that were not used to train the model to build their explanations, raises questions about their relevance and usefulness. Several troublesome issues have been identified: [Baehrens *et al.*, 2010] e.g. notice that modeling the decision function of a black-box classifier with a surrogate model trained on generated instances can result in explanation vectors that point in the wrong directions in some areas of the feature space in a trivial problem. The stability of post-hoc explainer systems has been criticized as well, showing that some of these approaches are locally not stable enough [Alvarez Melis and Jaakkola, 2018] or on the contrary too stable and thus not locally accurate enough [Laugel *et al.*, 2018b].

However, there seems to have been few works [Lipton, 2017; Rudin, 2018] questioning the post-hoc paradigm itself, and wondering about the risks of generating explanations that are disconnected from the ground truth. The work of [Kabra *et al.*, 2015] can be related to this issue, as it attempts to identify which neighbors from the training data are the most influential for a given prediction made by a trained classifier. However, their approach relies on retraining the classifier to induce a notion of causality, and thus lies beyond the post-hoc context. In a similar fashion, [Jiang *et al.*, 2018] try to identify which predictions can be trusted in regard to ground truth instances based on their distance to training data.



(a) RF with 3 estimators (b) SVC with RBF kernel

Figure 1: Two classifiers have been trained on 80% of the dataset (a 2D version of the iris dataset) and have the same accuracy over the test set, 0.78. Left picture: because of its low robustness, the random forest classifier makes questionable generalizations (e.g. small red square in the dark blue region) Right picture: the support vector classifier makes questionable decisions in regions far away from the training data (red area in the top left corner).

3 Ground-truth Justification

This section presents motivations and formalization for the notion of ground-truth justification, central to this paper.

3.1 Motivations

The notion of ground-truth justification we seek aims at making a distinction between an explanation that has been generated because of some previous knowledge (such as training data) and one that would be a consequence of an artifact of the classifier. In the post-hoc paradigm, explainer systems do not have access to the training data; making this distinction is thus not trivial. In the case of counterfactual explanations that rely on specific instances, the risk would be having counterfactual instances (i.e. predictions from another class) caused by these artifacts. Artifacts can be created in particular because of a lack of robustness of the model, or because it is forced to make a prediction for an observation in an area that was not being covered by the training set. Fig. 1 shows an illustration of how easily such situations can arise in a trivial problem. In both cases, some regions can be found where the classifier makes questionable improvisations in areas he has no information about (no training data), either because it is not robust enough (left image) or because of its complexity (right image). Although not harmful in the context of prediction (a desirable property of a classifier remains its ability to generalize to new observations), having an explanation caused by an artifact such that it can not be associated to any existing knowledge by a human user may be undesirable

in the context of interpretability: for instance, a physician using a diagnostic tool and an explanation for a prediction that is not based on existing medical cases would be conceptually useless, if not very dangerous.

3.2 Definitions

We propose to define this relation between an explanation and some existing knowledge (ground-truth data used to train the black-box model) using the topological notion of path, used when defining the path connectedness of a set. In order to be more easily understood and employed by a user, we argue that the counterfactual instance should be continuously connected to an observation from the training dataset. The idea of this *justification* is not to identify the instances that are *responsible* for a prediction (such as in the aforementioned work of [Kabra *et al.*, 2015]), but the ones that are correctly being predicted to belong to the same class for similar reasons.

Definition 1 (Justification)**.** *Given a classifier $f : \mathcal{X} \to \mathcal{Y}$ trained on a dataset X, a counterfactual example $e \in \mathcal{X}$ is* justified *by an instance $a \in X$ correctly predicted if $f(e) = f(a)$ and if there exists a continuous path h between e and a such that no decision boundary of f is met.*

Formally, e is justified *by $a \in X$ if: $\exists\, h : [0,1] \to \mathcal{X}$ such that: (i) h is continuous, (ii) $h(0) = a$, (iii) $h(1) = e$ and (iv) $\forall t \in [0,1],\ f(h(t)) = f(e)$.*

To adapt this continuous notion to the context of a black-box classifier, we replace the connectedness constraint with *ϵ-chainability*, with $\epsilon \in \mathbb{R}^+$: an *$\epsilon$-chain* between e and a is a finite sequence e_0, e_1, ... $e_N \in \mathcal{X}$ such that $e_0 = e$, $e_N = a$ and $\forall i < N$, $d(e_i, e_{i+1}) < \epsilon$, with d a distance metric.

Definition 2 (ϵ-justification)**.** *Given a trained classifier $f : \mathcal{X} \to \mathcal{Y}$ trained on a dataset X, a counterfactual example $e \in \mathcal{X}$ is* ϵ-justified *by an instance $a \in X$ correctly predicted if $f(e) = f(a)$ and if there exists an ϵ-chain $\{e_i\}_{i \leq N}$ between e and a such that $\forall i \leq N, f(e_i) = f(e)$.*

This definition is equivalent to approximating the aforementioned function h by a sequence $(e_i)_i$:, when ϵ decreases towards 0, this definition becomes a weak version of Definition 1. Consequently, we call a *justified* (resp. *unjustified*) counterfactual (written JCF, resp. UCF) a counterfactual example that does (resp. does not) satisfy Definition 2.

Setting parameter ϵ then allows to build an ϵ-graph of instances classified similarly. Using such a graph to approximate connectedness is also found in some manifold learning approaches (e.g. Isomap [Tenenbaum *et al.*, 2000]), where local neighborhoods help to approximate connectedness and thus reduce the dimension of the data.

This definition of connectedness can also be related to the one used in density-based clustering methods, such as the well-known DBSCAN [Ester *et al.*, 1996]: in DBSCAN, two parameters ϵ and $minPts$ control the resulting clusters (resp. outliers), built from the core points, which are instances that have at least (resp. less than) $minPts$ neighbors in an ϵ-ball. Thus, having an instance being ϵ-justified by another is equivalent to having them both belong to the same DBSCAN cluster, with parameters $minPts = 2$ and same ϵ.

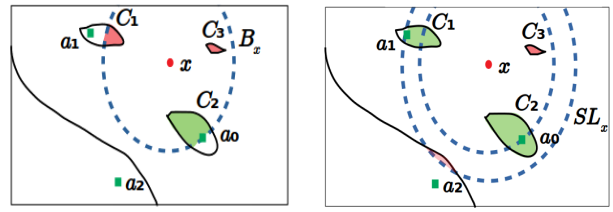


Figure 2: Illustration of the Local Risk Assessment procedure in the context of binary classification. Left: Definition and Initial Assessment steps; right: Iteration step.

4 Assessment of the Local Risk of Generating Unjustified Counterfactuals

The goal of this section is to design a test, the *Local Risk Assessment* (LRA) procedure, highlighting the existence of UCF in low-dimensional structured data.

4.1 LRA Procedure

Given a black-box classifier $f : \mathcal{X} \to \mathcal{Y}$ trained on the dataset X of instances of $\mathcal{X}$ and an instance $x \in \mathcal{X}$ whose prediction $f(x)$ is to be explained, the aim is to assess the risk of generating unjustified counterfactual examples in a local neighborhood of x. To do this, we propose a generative approach that aims at finding which regions of this neighborhood are ϵ-connected to an instance of X (i.e. verify Definition 2). In the rest of the paper, given a subset of instances A of $\mathcal{X}$, we note $A^l = \{z \in A \mid f(z) = l\}$. The LRA procedure, commented below, is detailed in Algorithm 1 and illustrated in a two-dimensional setting in Fig. 2: the red dot represents x, the observation whose prediction is to be interpreted, while green squares (a_0, a_1 and a_2) are correctly classified instances from the training set X. The decision boundary of f, the considered binary classifier, is represented by the black lines.

For clarity purposes, the procedure is split in three steps: first, the studied area is defined in the Definition step. Then, an Initial Risk Assessment step is performed in this area. Finally, if needed, the procedure is repeated in the Iteration step.

Definition Step

We first define the studied local neighborhood, i.e. the region of x which we are trying to assess the risk of, as the ball with center x and radius defined by its distance to its closest neighbor from X correctly predicted to belong to another class: $\mathcal{B}(x, d(x, a_0))$, with $a_0 = \underset{z \in X^{l \neq f(x)}}{\arg\min}\, d(x, z)$, s.t. $f(a_0)$ is correct. The distance $d(x, a_0)$ represents an approximation of the minimal distance between x and the decision boundary of f, and is hence a reasonable distance to look for counterfactual examples: as it is ϵ-connected to itself, a_0 represents a "close" justified counterfactual explanation by itself. The limit of $\mathcal{B}(x, d(x, a_0))$ is illustrated in the left picture of Fig. 2 by the blue dashed circle.

Initial Assessment Step

A high number n of instances $B_x = \{x_i\}_{i \leq n}$ are then sampled in $\mathcal{B}(x, d(x, a_0))$ following a uniform distribution and

labelled using f.

Using Definition 2, the goal of LRA is to identify which instances of $B_x^{f(a_0)}$ are connected to X through an ϵ-chain. The process to set the value of ϵ is detailed in Section 4.4. As mentioned earlier, an easy way to implement this is to use the clustering algorithm DBSCAN on $B_x^{f(a_0)} \cup \{a_0\}$ with parameter values ϵ and $minPts = 2$. We note $\{C_t\}_t$ the resulting clusters and outliers: according to Def. 2, the instances belonging to the same cluster as a_0 are ϵ-justified.

This Initial Assessment step is written in lines 2 to 9 of Algorithm 1, and illustrated in the left image of Fig. 2: instances are generated in B_x (blue dashed circle; the generated instances are not shown in the illustration), and the ones belonging to $B_x^{f(a_0)}$ are assigned to clusters C_1 (unjustified at this step), C_2 (justified, since $a_0 \in C_2$) and C_3 (non-justified).

Iteration Step

However, at this step, there is no certainty that the other instances of $B^{f(a_0)}$ are unjustified, as they could simply be connected to other instances from $X^{f(a_0)}$ than a_0, or using an ϵ-path that can not be fully included in the explored area $\mathcal{B}(x, d(x, a_0))$ (see Fig. 2, instances belonging to cluster C_1 are actually justified by a_1 but are not connected within B_x). To answer this question, we define a_1 as the second closest instance of $X^{f(a_0)}$ to x correctly predicted and broaden the exploration region to the hyperspherical layer defined as:

$$\mathcal{SL}_1 = \{x \in \mathcal{X} \text{ s.t. } d(a_0, x_0) \leq d(x, x_0) \leq d(a_1, x_0)\}.$$

Instances SL_1 are generated uniformly in $\mathcal{SL}_1$, and the instances from $SL_1^{f(a_0)} \cup \{a_1\}$ can then be assigned either to previous clusters C_t or to a_1 using the same criteria, i.e. if their minimum distance to an instance of an existing cluster is less than ϵ (right image of Fig. 2: cluster C_1 can now be connected to a_1 through the instances generated in SL_1).

This step is repeated by generating hyperspherical layers defined by all instances from $X^{f(a_0)}$ until all the initial clusters C_t can either be justified or are not being updated by any new instances anymore (such as cluster C_3 of Fig. 2, which was not updated between the two iterations). This step is illustrated in the "while" loop of Algorithm 1, lines 10 to 19). If some non-connected clusters are still being updated when all instances from the training set have been explored (e.g. red region in the top left corner of the left picture of Fig. 1), they are considered as non-connected.

In the end, the LRA procedure returns n_J (resp. n_U) the number of JCF (resp. UCF) originally generated in $\mathcal{B}(x, d(x, a_0))$. If $n_U > 0$, there exists a risk of generating unjustified counterfactual examples in the studied area.

4.2 Metrics

Using these results, we evaluate the proportion of generated UCF with the 2 criteria:

$$S_x = \mathbb{1}_{n_U > 0} \quad \text{and} \quad R_x = \frac{n_U}{n_U + n_J}\,.$$

We are mainly interested in S_x, which labels the studied instance x as being vulnerable to the risk (i.e. having a non-null risk). The risk itself, measured by R_x, describes the likelihood of having an unjustified counterfactual example in the studied area when looking for counterfactual examples.

Algorithm 1 Local risk assessment

Require: x, f, X
1: Sort correctly predicted instances from $X^{l \neq f(x)} = \{a_0, a_1, ...\}$ in increasing order of their distance to x
2: $B_x = \{x_i\}_{i \leq n} \sim \text{Uniform}(\mathcal{B}(x, a_0))$
3: $\epsilon = \max_{x_i \in B_x} \min_{x_j \in B_x} d(x_i, x_j)$
4: $B_x^{f(a_0)} = \{x_i \in B_x \,:\, f(x_i) = f(a_0)\}$
5: $\{C_t\}_t \leftarrow \text{DBSCAN}(B_x^{f(a_0)} \cup \{a_0\}, \epsilon, \text{minPts} = 2)$
6: $\mathcal{C}_J = \{C_{t_0}\}$ s.t. $a_0 \in C_{t_0}$; $n_J = |\mathcal{C}_J|$
7: $\mathcal{C}_{NC} = \{C_t\}_t \setminus C_{t_0}$; $n_{NC} = |\mathcal{C}_{NC}|$
8: $\mathcal{C}_U = \{\}$; $n_U = 0$
9: $k = 0$
10: **while** $n_{NC} > 0$ **do**
11: $\quad k = k + 1$
12: $\quad SL_k = \{x_i\}_i \sim \text{Uniform}(\mathcal{SL}_k)$
13: $\quad SL_k^{f(a_k)} = \{x_i \in SL_k \,:\, f(x_i) = f(a_k)\}$
14: $\quad \{C'_t\}_t \leftarrow \text{DBSCAN}(SL_k^{f(a_k)} \cup \{a_k\}, \epsilon, \text{minPts} = 2)$
15: $\quad$ Update $\mathcal{C}_J$ and $\mathcal{C}_{NC}$ with $\{C'_t\}_t$
16: $\quad$ Update $\mathcal{C}_U$ whith clusters from $\mathcal{C}_{NC}$ that are not growing anymore
17: $\quad$ Update n_J, n_U and n_{NC}
18: **end while**
19: **return** n_J, n_U

As these scores are calculated for a specific instance x with a random generation component, we are also interested in their average values $\bar{S}$ and $\bar{R}$ for 10 runs of the procedure for each x and over multiple instances.

4.3 Illustrative Results

For the purpose of giving insights about the procedure results, a toy dataset (half-moons dataset) is used. A classifier, deliberately chosen for its low complexity (random forest classifiers with only 3 trees), is trained on 70% of the data (98% accuracy on the rest of the dataset). The left image of Fig. 3 shows the result of the LRA procedure for a specific instance x (yellow instance), exploring the neighborhood $\mathcal{B}(x, d(x, a_0))$ (blue circle) delimited by its closest neighbor from opposite class a_0 (orange instance). The red and blue dots are the instances used to train the classifier, while the red and blue areas represent its decision function: x is predicted to belong to the blue class and a_0 to the red one. Within $\mathcal{B}(x, d(x, a_0))$, a red square "pocket" is detected as a group of unjustified counterfactual examples since there is no red instance in this pocket: $S_x = 1$.

This simple example illustrates the fact that the risk of generating unjustified counterfactual examples does exist. Full results of the LRA procedure run over multiple instances of several datasets are available in Section 4.5.

4.4 LRA Parameters: n and ϵ

The values of n and ϵ are obviously crucial since they define the notion of ϵ-justification and impact the average distance between the number of generated instances n. Broadly

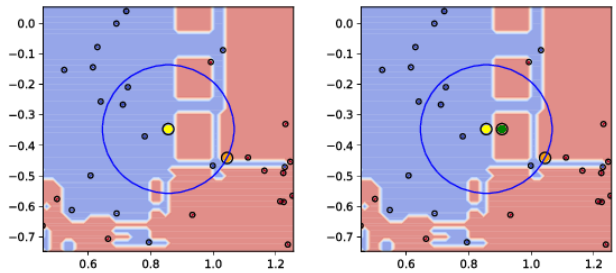


Figure 3: Illustrative result of the Local Risk Assessment procedure (left: $S_x = 1$) and the Vulnerability Evaluation test (right: $J_x = 1$) for an instance of the half-moons dataset.

speaking, the higher n and the smaller ϵ are, the better the approximation of the topology of the local neighborhood is: if the value of n is too low, small "pockets" of unjustified counterfactual examples might be missed. Similarly, if the value of ϵ is too high, an ϵ-chain will not be a good approximation of a connected path between two instances. However, there is obviously a tradeoff between this accuracy and the computation time of the algorithm, the complexity of which can be evaluated to be at worst $O(pn^2)$, with p the size of $X^{f(a_0)}$. In practice, because the instances B_x are generated before running DBSCAN, we can set it to the maximum value of the distances of B to their closest neighbors: $\epsilon = \max_{x_i \in B_x} \min_{x_j \in B_x \setminus \{x_i\}} d(x_i, x_j)$. However, this requires the whole procedure to be run several time to make sure ϵ does not take an absurd value (e.g. in the case where one single instance would be generated far away from the others). Thus, the problem becomes of setting the value of n alone.

The role of parameter n is to make sure that the explored area is "saturated" enough and that no subtelty of the model's decion border, as well as potential unjustified counterfactual examples, are left undetected. Thus, for each observation x we assume there exists a threshold above which the complexity of the decision boundary of f would be fully "captured". When this value is met, increasing n has very little impact over the found clusters and therefore on the risk score R_x. We validate this intuition empirically by looking at the evolution of R_x depending on n for multiple instances, but due to space, do not include the results in the present paper. However, the obtained results and code to reproduce them are available in an online repository (https://github.com/thibaultlaugel/truce). Although seemingly high, the complexity of the LRA procedure is not seen as an issue as it is proposed as a diagnostic tool, rather than used to generate explanations by a user.

4.5 Quantitative Results

This section describes the datasets and approaches used in the experiments. For each considered dataset, a train-test split of the data is performed with 70%-30% proportion, and a binary classifier is trained. To mitigate the impact the choice of the classifier would make, we use the same classifier for every dataset, a random forest (RF) with 200 trees. However, we also train a Support Vector classifier (SVC) with Gaussian kernel on one of the Boston dataset (see below) to make sure the highlighted issue is not a characteristic feature of random forests.

Dataset	$\bar{S}$	$\bar{R}$ [%]
Iris	0.41	0.05 (0.04)
Half-moons	0.37	7.01 (17.1)
BostonRF	0.63	16.0 (25.2)
BostonSVC	0.29	6.1 (13.6)
Recidivism	0.81	26.6 (30.8)

Table 1: Proportion of instances being at risk (average S score) over the test sets, and risk (average percentage and standard deviation values of R) of generating an UCF for 5 datasets.

The test set is then used to run the experiments: for each instance it contains, Algorithm 1 is run and scores $\bar{R}$ and $\bar{S}$ are calculated. The code and data to reproduce all the experiments are available online in the aforementioned repository.

Datasets

The datasets considered for these experiments include 2 low-dimensional datasets (half-moons and iris) as well as 2 real datasets: Boston Housing [Harrison and Rubinfeld, 1978] and Propublica Recidivism [Larson *et al.*, 2016]. These structured datasets are commonly used in the interpretability (and fairness) literature.

Results

The results of the Local Risk Assessment procedure are shown in Table 1. Among all the considered datasets, between 29% and 81% of the studied instances have unjustified counterfactual examples in their neighborhood (S_x=1).

An important thing to note is that the $\bar{R}$ score varies greatly between the instances of a single dataset (high standard deviation), as well as between the datasets. This can be explained by the local complexity of both the used datasets and classifiers. Supposedly, an instance located far away from the decision boundary of the classifier will have greater chance to generate unjustified counterfactual examples than an instance located closer, since the neighborhood explored by the Local Risk Assessment procedure will be much wider. More generally, these results depend on assumptions such as characteristics of the classifier (SVC with Gaussian kernel seems to achieve better results than RF), of the data (dimensionality, density) as well as its labels (number of classes, classes separability), and the classifier's accuracy. These have not been studied here as they lie out of the frame of this work, but are somehow illustrated by the variability of $\bar{R}$ between datasets. The learning algorithm of the classifier itself may also have an impact, as models such as logistic regression or 1-NN classifier have, by design, no UCF ($\bar{S} = 0.0$). Thus, the link between connectedness and overfitting is not obvious and needs to be further explored.

These results confirm the intuition that the risk of generating UCF exists and is important. In this context, since they do not have the ability to distinguish a ground-truth backed decision from a classifier artifact, post-hoc counterfactual approaches are vulnerable to this issue. The next section studies the extent of this vulnerability.

5 Vulnerability of Post-hoc CF Approaches

Once the risk of generating UCF has been established, we analyze how troublesome it is for existing counterfactual approaches. This section presents a second procedure, called *Vulnerability Evaluation*, which aims at assessing how a post-hoc interpretability method behaves in the presence of UCF.

5.1 Vulnerability Evaluation Procedure: VE

The goal of the VE procedure is to assess the risk for state of the art methods to generate UCF in vulnerable regions. Given an instance $x \in X$, we use the LRA procedure to assess the risk R_x and focus on the instances where this risk is "significant" (imposing $R_x > 0.25$). Using the method to be evaluated, a counterfactual explanation $E(x) \in \mathcal{X}$ is generated.

To check whether $E(x)$ is justified or not, a similar procedure as LRA is used: instances $B_{E(x)}$ are generated uniformly in a local region defined as the hyperball with center $E(x)$ and radius $d(E(x), b_0)$, where b_0 denotes the closest instance to $E(x)$ correctly predicted to belong to the same class, i.e. $b_0 = \underset{x_i \in X^{f(E(x))}}{\text{argmin}}\ d(E(x), x_i)$. The DBSCAN algorithm is also used with the same parameters as previously. If $E(x)$ is assigned to the same cluster as the closest instance to b_0, then there exists an ϵ-chain linking $E(x)$ and b_0, meaning $E(x)$ is a JCF according to Definition 2.

If not, similarly as previously, the explored area is expanded to the hyperspherical layer defined by the distance to b_1, the second closest instance from $X^{f(E(x)}$ correctly predicted. This step is repeated by widening the studied area as many times as necessary: if no instance from $X^{f(E(x))}$ can be connected, then $E(x)$ is labelled as being unjustified.

As for state of the art methods, we focus on the approaches: HCLS [Lash *et al.*, 2017], Growing Spheres [Laugel *et al.*, 2018a] and LORE [Guidotti *et al.*, 2018a] (modified to return counterfactual examples rather than rules).

5.2 Metrics

The goal being to check wether a counterfactual example $E(x)$ is justified or not, we simply define the justification score $J_{E(x)}$ as a binary score that equals 1 if $E(x)$ is justified, 0 in the other case. Again, we also measure the average value $\bar{J}$ of $J_{E(x)}$ over multiple instances and multiple runs.

5.3 Illustrative Results

The considered post-hoc counterfactual approaches are applied to a toy dataset to illustrate how they can behave, in the aforementioned setup (see Section 4.3), see Fig. 3 (right image). As this instance has a risk score $R_x \geq 0.25$, we apply VE. A counterfactual example is generated using HCLS [Lash *et al.*, 2017] with budget $B = d(x, b_0)$. The output $E(x)$ is represented by the green instance. In this situation, $E(x)$ lies in the unjustified area, meaning HCLS generated an unjustified counterfactual example ($J_{E(x)} = 0$). This example illustrates the fact that, when there is a consequent risk R_x, post-hoc approaches are vulnerable to it.

5.4 Quantitative Results

The results of the VE procedure are shown in Table 2. As expected, when confronted to situations with a local risk of generating unjustified counterfactual examples ($R_x \geq 0.25$), the considered approaches fail to generate JCF: the proportion of generated JCF with state of the art approaches can fall as low as 63 %. This confirms the assumption that post-hoc counterfactual approaches are, by construction, vulnerable to the studied issue. Among the considered approaches, HCLS seems to achieve slightly better performance than GS. This could be explained by the fact that HCLS tries to maximize the classification confidence, leading to counterfactuals that may be located further from x than GS and hence with a higher chance of avoiding unjustified classification regions. The same observation and conclusion can be made for LORE, which minimizes a L_0 norm. Hence, connectedness seems to be able to be favored at the expense of the distance.

Dataset	HCLS	GS	LORE
Iris	0.63	0.63	0.70
Half-moons	0.83	0.67	0.83
BostonSVC	0.95	0.93	1.0
BostonRF	0.86	0.84	1.0
Recidivism	0.94	0.94	0.98

Table 2: Proportion of generated counterfactuals that are justified ($\bar{J}$) for vulnerable instances ($R_x \geq 0.25$)

Again, a major variability in $\bar{J}$ can be observed between instances. This confirms the previously made assumption that the connectedness of the classes predicted by f heavily depends on the data and considered classifier.

6 Conclusions

In this work, we propose a justification definition that counterfactual explanations should satisfy, motivated by an intuitive requirement based on path connectedness. We propose a procedure to assess the risk of generating unjustified counterfactual examples in the post-hoc context, and additionally show that post-hoc counterfactual explanations are indeed vulnerable to such risk.

In this context, we argue that there is no existing satisfying way to provide post-hoc explanations that are both faithful to the classifier and to ground-truth data and further research is necessary. In the mean time, using the train instances, although not always possible in the utilisation context of post-hoc methods, seems to remain necessary.

In light of this study, further works include adapting the current procedure to high-dimensional data and making a more in-depth study of the role of the classifier's complexity in justification. Prospective works also include studying the connectedness of adversarial examples, as ongoing experiments suggest that they verify the proposed justification definition.

Aknowledgements

This work has been done as part of the Joint Research Initiative (JRI) project "Interpretability for human-friendly machine learning models" funded by the AXA Research Fund. We would like to thank Arthur Guillon for stimulating discussions and useful comments.

References

[Alvarez Melis and Jaakkola, 2018] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in Neural Information Processing Systems*, 31:7786–7795, 2018.

[Baehrens *et al.*, 2010] David Baehrens, Timon Schroeter, Stefan Harmeling, Katja Hansen, and Klaus-Robert Muller. How to Explain Individual Classification Decisions Motoaki Kawanabe. *Journal of Machine Learning Research*, 11:1803–1831, 2010.

[Bottou *et al.*, 2013] Léon Bottou, Jonas Peters, Joaquin Quiñonero Candela, Denis X. Charles, D. Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research*, 14:3207–3260, 2013.

[Craven and Shavlik, 1996] Mark W. Craven and Jude W. Shavlik. Extracting tree-structured representations of trained neural networks. *Advances in Neural Information Processing Systems*, 8:24–30, 1996.

[Ester *et al.*, 1996] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of the 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD'96)*, pages 226–231, 1996.

[Guidotti *et al.*, 2018a] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. Local rule-based explanations of black box decision systems. *arXiv preprint 1805.10820*, 2018.

[Guidotti *et al.*, 2018b] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Computing Surveys (CSUR)*, 51(5):93, 2018.

[Hara and Hayashi, 2016] Satoshi Hara and Kohei Hayashi. Making tree ensembles interpretable. *ICML Workshop on Human Interpretability in Machine Learning (WHI 2016)*, 2016.

[Harrison and Rubinfeld, 1978] David Harrison and Daniel Rubinfeld. Hedonic prices and the demand for clean air. *Environment Economics and Management*, 5:81–102, 1978.

[Jiang *et al.*, 2018] Heinrich Jiang, Been Kim, Melody Guan, and Maya Gupta. To trust or not to trust a classifier. *Advances in Neural Information Processing Systems*, 31:5541–5552, 2018.

[Kabra *et al.*, 2015] Mayank Kabra, Alice Robie, and Kristin Branson. Understanding classifier errors by examining influential neighbors. *2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3917–3925, 2015.

[Kim *et al.*, 2014] Been Kim, Cynthia Rudin, and Julie A Shah. The Bayesian Case Model: A generative approach for case-based reasoning and prototype classification. *Advances in Neural Information Processing Systems*, 27:1952–1960, 2014.

[Larson *et al.*, 2016] Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. How we analyzed the compas recidivism algorithm. *ProPublica*, 2016.

[Lash *et al.*, 2017] Michael Lash, Qihang Lin, Nick Street, Jennifer Robinson, and Jeffrey Ohlmann. Generalized inverse classification. In *Proc. of the 2017 SIAM Int. Conf. on Data Mining*, pages 162–170, 2017.

[Laugel *et al.*, 2018a] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. Comparison-based inverse classification for interpretability in machine learning. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 100–111, 2018.

[Laugel *et al.*, 2018b] Thibault Laugel, Xavier Renard, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. Defining locality for surrogates in post-hoc interpretablity. *ICML Workshop on Human Interpretability in Machine Learning (WHI 2018)*, 2018.

[Lipton, 2017] Zachary C. Lipton. The mythos of model interpretability. *ICML Workshop on Human Interpretability in Machine Learning (WHI 2017)*, 2017.

[Lundberg and Lee, 2017] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30:4765–4774, 2017.

[Ribeiro *et al.*, 2016] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should I trust you?": Explaining the predictions of any classifier. In *Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, KDD '16, pages 1135–1144, 2016.

[Rudin, 2018] Cynthia Rudin. Please stop explaining black box models for high stakes decisions. *NeurIPS Workshop on Critiquing and Correcting Trends in Machine Learning*, 2018.

[Tenenbaum *et al.*, 2000] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[Turner, 2015] Ryan Turner. A model explanation system. *NIPS Workshop on Black Box Learning and Inference*, 2015.

[Wachter *et al.*, 2018] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box; automated decisions and the GDPR. *Harvard Journal of Law & Technology*, 31(2):841–887, 2018.