FACE:
Feasible and Actionable Counterfactual Explanations

Rafael Poyiadzi
University of Bristol
rp13102@bristol.ac.uk

Kacper Sokol
University of Bristol
k.sokol@bristol.ac.uk

Raul Santos-Rodriguez
University of Bristol
enrsr@bristol.ac.uk

Tijl De Bie
Ghent University
tijl.debie@ugent.be

Peter Flach
University of Bristol
peter.flach@bristol.ac.uk

Abstract

Work in Counterfactual Explanations tends to focus on the principle of "the closest possible world" that identifies small changes leading to the desired outcome. In this paper we argue that while this approach might initially seem intuitively appealing it exhibits shortcomings not addressed in the current literature. First, a counterfactual example generated by the state-of-the-art systems is not necessarily representative of the underlying data distribution, and may therefore prescribe unachievable goals (e.g., an unsuccessful life insurance applicant with severe disability may be advised to do more sports). Secondly, the counterfactuals may not be based on a "feasible path" between the current state of the subject and the suggested one, making actionable recourse infeasible (e.g., low-skilled unsuccessful mortgage applicants may be told to double their salary, which may be hard without first increasing their skill level). These two shortcomings may render counterfactual explanations impractical and sometimes outright offensive. To address these two major flaws, first of all, we propose a new line of Counterfactual Explanations research aimed at providing actionable and feasible paths to transform a selected instance into one that meets a certain goal. Secondly, we propose **FACE**: an algorithmically sound way of uncovering these "feasible paths" based on the shortest path distances defined via density-weighted metrics. Our approach generates counterfactuals that are coherent with the underlying data distribution and supported by the "feasible paths" of change, which are achievable and can be tailored to the problem at hand.

1 Introduction

The widespread deployment of complex Machine Learning (ML) systems and their use for important decision-making have led to a rising interest in the fields of Interpretable and Explainable Machine Learning (IML and XML respectively). IML refers to developing models that not only perform well with respect to the usual measures of predictive performance, but which are also transparent. IML approaches typically aim to achieve this by choosing a model from a class of inherently interpretable models, for example, decision trees. However, this approach may come at a cost of accuracy, as the complexity of the used models is inherently limited. On the other hand, XML is mainly concerned with post-hoc approaches that aim at explaining an ML model, or its predictions, after it has been trained, often treating it as a black-box. It imposes no limitations on the complexity of the compatible models, nevertheless popular approaches in XML, such as "Global Surrogate Models" or "Local Surrogate Models" [12, 13, 11], add an extra layer of modelling complexity in exchange for transparency. In a third category – based on Counterfactual Explanations (CE) – one does not need to

Preprint. Under review.

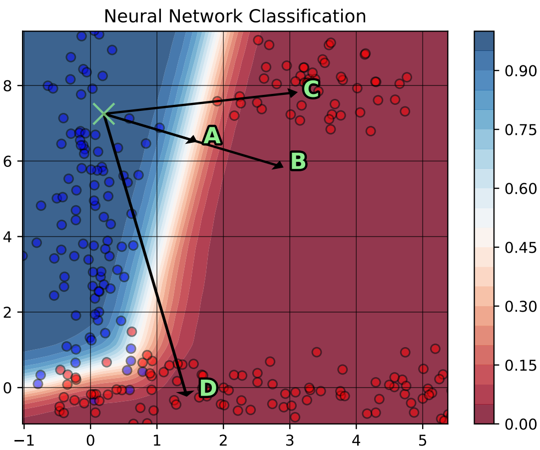


Figure 1: A, B, C and D are four viable counterfactuals of $\times$, all satisfying the condition of having a different predicted class to the selected instance. We argue that D is the best choice. A is the result of minimising the l_2-norm. B is a generic data point that has a large classification margin. Nevertheless, both A and B lie in a *low density region*. C and D do not share the shortcomings of A and B: they lie in high-density regions and have a relatively large classification margins. The major difference between C and D is the connection between $\times$ and D via a high-density path, indicating that it is feasible for the original instance to be transformed into D despite C being simply closer.

worry about such issues, as the objective is not to understand the inner workings of a model [20], but rather to provide a transformation of a particular instance leading to the desired prediction. In this paper we are concerned with Counterfactual Explanations (or, Contrastive Explanations [19]) that fall under the category of *Example-Based Reasoning*. While other approaches aim at answering: "Why has my loan been declined?", CE aim at answering a question of a different nature: "What do I need to do for my loan to be accepted?"

Wachter et al. [20] propose three aims of (counterfactual) explanations with respect to their audience:

1. to inform and help the explainee understand why a particular decision was reached,
2. to provide grounds to contest adverse decisions, and
3. to understand what could be changed to receive a desired result in the future, based on the current decision-making model.

Counterfactual explanations achieve all three of these aims [20]. However, a naïve application of the last one – the principle of "the closest possible world" that prescribes small changes that lead to the desired outcome – may yield inadequate results. Firstly, a counterfactual generated by the state-of-the-art explainability system is not necessarily representative of the underlying data distribution, and therefore may prescribe unachievable goals. This shortcoming is illustrated in Figure 1, where points A and B – both close to the explained data point $\times$ with respect to the l_2-norm – achieve the desired prediction, nevertheless they lie in a low-density region. This last observation undermines the practical feasibility of points A and B since there are no precedents of any other similar instances in the data. Secondly, counterfactuals provided by current approaches may not allow for a *feasible path* between the initial instance and the suggested counterfactual making actionable recourse infeasible. This argument is illustrated with point D in Figure 1, which we argue is a more actionable counterfactual than C. Both these discoveries have prompted us to establish a new line of research for Counterfactual Explanations: providing *actionable* and *feasible* paths to transform a certain data point into one that meets certain goals (e.g., belong to a desirable class).

The contribution of this paper is twofold. We first critique the existing line of research on Counterfactual Explanations by pointing out the shortcomings of dismissing the inherent nature of the target counterfactual and its (real-life) context. We point out that existing research on counterfactual explanations is not aligned with real-world applications (e.g., offering a *useful* counterfactual advice to customers who have been denied loans). To overcome this challenge we identify two essential properties of counterfactual explanations: *feasibility* and *actionability*, which motivate a new line of research concerned with providing high-density paths of change. Secondly, we propose a novel, well-founded approach that provides feasible and actionable counterfactual explanations that respect the underlying data distribution and are connected via high-density paths (based on the shortest path distances defined via density-weighted metrics) to the explained instance. Our approach – which we call *Feasible and Actionable Counterfactual Explanations* (**FACE**) – mitigates all of the risks associated with the explanations produced by the current line of research. We support our claims by discussing how ignoring these premises could lead to "unachievable goals" with undesired consequences such as a loss of end user's trust. Furthermore, we show that our algorithmic contribution to generate feasible and actionable counterfactuals is non-trivial as the generated counterfactuals come from dense regions in addition to being connected through a high-density path with the original instance. Therefore, the explanations are coherent with the underlying data distribution and can be tailored to the user by customising the "feasible paths" of change. In Section 2 we establish the links and differences of our approach with current approaches in the literature. Section 3 introduces our methodology and Section 5 discusses related work. In Section 4 we present our experimental results and we conclude our work with a discussion in Section 6.

2 Counterfactual Thinking and Counterfactual Examples

The desiderata put forward by Wachter et al. [20] might at first seem sufficient to construct a helpful counterfactual for any task at hand. However, our experiments show that this is not necessarily the case, prompting the need for a new approach that ensures usefulness of counterfactual explanations in practice.

First, the nature of the target instance – the derived counterfactual example – is not taken into account. This may lead to a situation where the target instance is not representative of the underlying data distribution, i.e., it is located in a low density region, and thus essentially can be considered an outlier. Outliers are poor counterfactual explanations in practice as they would not naturally occur in the first place. In addition to being poor explanations, such counterfactuals are at risk of harming the explainee by suggesting a change of which the future outcome is highly uncertain, as classifiers tend to be less reliable in sparsely populated regions of the data space, especially close to a decision boundary. Points A and B shown in Figure 1 are prime examples of this major drawback. The uncertainty in a prediction, coming either from a low classification margin or due to the low density of a region, should be of utmost importance when generating a counterfactual.

Beyond feasibility and actionability, it is also important to consider the model's confidence of predictions as it may contribute to issues with a delayed impact [8]. For example, consider a person who had his loan application rejected and wants to know what changes to make for his application to be accepted next time. If this person is handed a counterfactual explanation and implements the proposed changes, his loan application will be accepted. However, if the *new state* of the subject (the proposed counterfactual) is in a region of high uncertainty, then there exists a high risk that this individual will default. This unintended consequence can be either attributed to the counterfactual data point lying on a decision boundary (caused by minimising a norm constraint only) or to its placement in a low-density region where the model had not seen enough data. In the process of trying to help, the system generating counterfactuals may actually hurt the explainees.

Furthermore, the desiderata presented by Wachter et al. [20] do not account for the extent to which the change – a transformation from the current state to the suggested counterfactual state – is feasible. "Counterfactual thinking" refers to the concept of hypothesising what would have happened had something been done differently [14], i.e., "Had I done X instead of Y, would the outcome still be Z?" However, when adapting this concept to ML applications (e.g., see [14]) the outcome is usually decided prior to finding a counterfactual cause. What has been overlooked by the XML community is that the aim of a counterfactual explanation is for the explainee to *actually try and make the change* given the actionable nature of the explanation. A customer whose loan application has been rejected would (probably) disregard a counterfactual explanation conditioned on him being 10 years younger.

In addition to the feasibility of a counterfactual explanation – the target instance has to be in a region of high density – it also has to be achievable in the real world, therefore "reachable" from the selected instance. This means that the explanation must not suggest altering attributes in ways that are particularly hard or even physically impossible in the real world, such as reducing one's age. This also implicitly implies the existence of a short, continuous and feasible path from the selected instance to the target instance for the counterfactual to be actionable. Specifically, paths crossing low-density regions are arguably less feasible as instances in these regions are rare and unlikely by definition.

To sum up, the current state-of-the-art solutions do not satisfy the three requirements proposed by Wachter et al., which we believe are critical for actionability and thus practical utility of counterfactual explanations. To remedy this situation we propose to following objectives for counterfactual explanations in addition to the inherent requirement of this instance belonging to the desired class:

1. feasibility of the counterfactual data point,
2. continuity and feasibility of the path linking it with the data point being explained, and
3. high density along this path and its relatively short length.

3 Feasible Counterfactuals

Before presenting **FACE** we introduce the necessary notation and background for completeness (see [1] and references therein for an in-depth presentation of this topic). We then show how different variants of our approach affect its performance and the quality of generated counterfactuals.

3.1 Background

Let $\mathscr{X} \subseteq \mathbb{R}^d$ denote the input space and let $\{x_i\}_{i=1}^N \in \mathscr{X}$ be an independent and identically distributed sample from a density p. Also, let f be a positive scalar function defined on $\mathscr{X}$ and let γ denote a path connecting x_i to x_j, then the f-length of the path is denoted by the *line integral* along γ with respect to f[1]:

$$\mathscr{D}_{f,\gamma} = \int_\gamma f(\gamma(t)) \cdot |\gamma'(t)| dt. \tag{1}$$

The path with the minimum f-length is called the f-geodesic, and its f-length is denoted by $\mathscr{D}_{f,\gamma^\star}$.

Consider a geometric graph $G = (V, E, W)$ with vertices V, edges E and (edge) weights W. The vertices correspond to the sampled instances (training data) and edges connect the ones that are close with respect to a chosen metric, which value (a measure of closeness) is encoded in the (edge) weights. We use the notation $i \sim j$ to indicate a presence of an edge connecting x_i and x_j, with the corresponding weight w_{ij}; and $i \nsim j$ to mark that x_i and x_j are not directly connected, in which case the weight is assumed to be $w_{ij} = 0$.

Let f depend on x through the density p with $f_p(x) := \tilde{f}(p(x))$. Then, the f-length of a curve $\gamma : [\alpha, \beta] \to \mathscr{X}$ can be approximated by a Riemann sum of a partition of $[\alpha, \beta]$ in sub-intervals $[t_{i-1}, t_i]$ (with $t_0 = \alpha$ and $t_N = \beta$):

$$\hat{\mathscr{D}}_{f,\gamma} = \sum_{i=1}^N f_p\Big(\frac{\gamma(t_{i-1}) + \gamma(t_i)}{2}\Big) \cdot \|\gamma(t_{i-1}) - \gamma(t_i)\|.$$

As the partition becomes finer, $\hat{\mathscr{D}}_{f,\gamma}$ converges to $\mathscr{D}_{f,\gamma}$ [3, Chapter 3]. This suggests using weights of the form:

$$\begin{aligned} &w_{ij} = f_p\Big(\frac{x_i + x_j}{2}\Big) \cdot \|x_i - x_j\|, \\ &\text{when} \quad \|x_i - x_j\| \le \varepsilon. \end{aligned}$$

The true density p is rarely known but Sajama and Orlitsky [17] show that using a *Kernel Density Estimator* (KDE) $\hat{p}$ instead will converge to the f-distance. Sajama and Orlitsky also show how to assign weights to edges while avoiding the need to perform density estimation altogether. Their

[1] We assume that $\mathscr{X}$ is endowed with a density function p with respect to the Lebesgue measure, where p is L-Lipschitz continuous with $L > 0$.

results apply to two graph constructions, namely, a k-NN graph and an ε-graph. In summary, for the three approaches the weights can be assigned as follows:

$$w_{ij} = f_{\hat{p}}\Big(\frac{x_i + x_j}{2}\Big) \cdot \|x_i - x_j\| \qquad \text{for } KDE; \tag{2}$$

$$w_{ij} = \tilde{f}\Big(\frac{r}{\|x_i - x_j\|}\Big) \cdot \|x_i - x_j\|, \qquad r = \frac{k}{N \cdot \eta_d} \qquad \text{for } k\text{-NN; and} \tag{3}$$

$$w_{ij} = \tilde{f}\Big(\frac{\varepsilon^d}{\|x_i - x_j\|}\Big) \cdot \|x_i - x_j\| \qquad \text{for } \varepsilon\text{-graph} \tag{4}$$

$$\text{when} \qquad \|x_i - x_j\| \leq \varepsilon$$

where η_d denotes the volume of a sphere with a unit radius in $\mathbb{R}^d$.

3.2 The FACE Algorithm: Feasible and Actionable Counterfactual Explanations

Building up on this background we introduce the **FACE** algorithm. It uses f-distance to quantify the trade-off between the path length and the density along this path, which can subsequently be minimised using a shortest path algorithm by approximating the f-distance by means of a finite graph over the data set. Moreover, **FACE** allows the user to impose additional feasibility and classifier confidence constraints in a natural and intuitive manner.

Firstly, a graph over the data points is constructed based on one of the three approaches: KDE, k-NN or ε-graph. The user then decides on the properties of the target instance (i.e., the counterfactual): the prediction threshold – a lower bound on prediction confidence outputted by the model, and the density (or its proxy) threshold. This part of the algorithm is described in Algorithm 1, which assumes access to a KDE.

To generate a counterfactual, **FACE** must be given its expected class. Optionally, the counterfactual can be additionally constrained by means of: a subjective prediction confidence threshold (t_p), a density threshold (t_d), a custom weight function (w), and a custom conditions function (c), which determines if a transition from a data point to its neighbour is feasible.[2] Subject to the new weight function and conditions function, if possible, the graph is updated by removing appropriate edges; otherwise a new graph is constructed.[3] The Shortest Path First Algorithm (SPFA) (Dijkstra's algorithm) [2] is executed on the resulting graph over all the candidate targets, i.e., the set I_{CT} of all the data points that meet the confidence and density requirements (see line 11 in Algorithm 1).

Complexity Execution of the Shortest Path First Algorithm between two instances can be optimised to have the worst case time complexity of $\mathcal{O}(|E| + |V|log|V|)$ where $|E|$ denotes the number of edges and $|V|$ the number of nodes in the graph. This complexity then scales accordingly with the number of candidate targets. The first term of the complexity – the number of edges – can be controlled by the user to a certain extent as it depends on the choice of the distance threshold parameter. The second term can also be controlled (and subsequently the first term as well) by reducing the number of instances to be considered, in which case the objective would be similar to the one of "Prototype Selection". A sub-sampling as simple as a random sampling of the data points, or more sophisticated alternatives such as Maximum Mean Discrepancy (MMD) [6, 5], can be used with a clear trade-off between the accuracy of the generated counterfactuals and the algorithm's speed. By defining the problem space in such a way our method is restricted to using paths that jump only in between existing data points with the generated counterfactuals also being part of the data set.

In practice a base graph can be generated and stored with the most generic conditions imposed, e.g., if the data represent people, edges between people of different sex would be removed. When

[2]Domain knowledge of this form (e.g., immutable features such as sex or conditionally immutable changes such as age, which is only allowed to change in one direction) are incorporated within the *conditions function* $c(\cdot,\cdot)$. This knowledge is *essential* if the desired counterfactual is to be useful.

[3]If the explainee wants to provide a custom cost function for the feature value changes, e.g., the cost of changing a job is twice that of change a marital status, a new graph has to be built from scratch. If, on the other hand, the cost function stays fixed and only new constraints (inconsistent with the current graph) are introduced, e.g., the counterfactuals should not be conditioned on a job change, the existing graph can be modified by removing some of its edges.

Algorithm 1: FACE Counterfactual Generator

input : Data ($X \in \mathbb{R}^d$), density estimator ($\hat{p} : \mathscr{X} \to [0,1]$), probabilistic predictor ($clf : \mathscr{X} \to [0,1]$), distance function ($d : \mathscr{X} \times \mathscr{X} \to \mathbb{R}_{/geq0}$), distance threshold ($\varepsilon > 0$), weight function ($w : \mathscr{X} \times \mathscr{X} \to \mathbb{R}_{>=0}$), and conditions function ($c : \mathscr{X} \times \mathscr{X} \to \{True, False\}$).

output : Graph (V, E, W) and candidate targets (I_{CT}).

/* Construct a graph. */

for *every pair* (x_i, x_j) *in* X **do**

    if $d(x_i, x_j) > \varepsilon$ *and* $c(x_i, x_j)$ *is True* **then**

        $i \nsim j$

        $w_{ij} = 0$

    else

        $i \sim j$

        /* In this case we use Equation 2 (KDE). This should be adjusted for k-NN and ε-graph constructions by using Equation 3 and 4 respectively. */

        $w_{ij} = w(\hat{p}(\frac{x_i + x_j}{2})) \cdot d(x_i, x_j)$

/* Construct a set of candidate targets. */

$I_{CT} = \{\}$

for x_i *in* X **do**

    if $clf(x_i) \geq t_p$ *and* $\hat{p}(x_i) \geq t_d$ **then**

        $I_{CT} = I_{CT} \cup i$

an explainee requests a counterfactual, he can impose further restrictions (by removing edges) to create a personalised graph, e.g., this individual is not willing to get divorced. On the other hand, if personalised cost function is required, entirely new graph needs to be generated. While the theory presented here only holds for continuous distributions, which satisfy the requirements discussed earlier, the approach can still be used with discrete features.

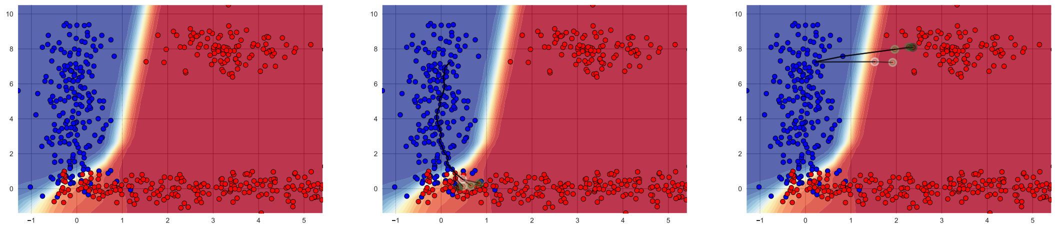
4 Experiments

To empirically demonstrate the utility of **FACE** we present results of its execution on two distinct data sets. First, we show the behaviour of our algorithm on a toy data set and compare the three graph construction approaches introduced in Section 3. Following our discussion and critique of existing approaches for Counterfactual Examples generation, e.g., their suboptimality, we refer the reader to [20] and present one of their counterfactual examples to illustrate its inferiority with respect to considering just a norm as the cost. Secondly, we apply our algorithm to the MNIST data set [7] and show how it can be used to derive meaningful digit transformations based on the calculated path.

Synthetic Data Set To this end, we trained a Neural Network, which architecture is based on two hidden layers of length 10 with ReLU activation functions. The toy data set (see, e.g., Figure 2) consists of three parts:

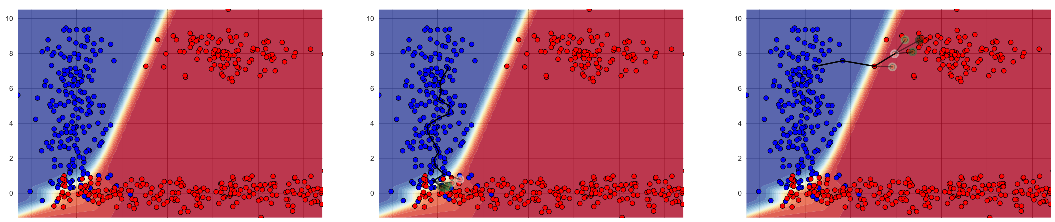
1. horizontal cloud of blue points to the left of the figure – 200 points distributed uniformly at random across the y-axis and sampled from a mean-zero Gaussian with 0.4 standard deviation on the x-axis,
2. vertical cloud of red points to the bottom of the figure – 200 points distributed uniformly at random across the x-axis and sampled from a mean-zero Gaussian with 0.5 standard deviation on the y-axis, and
3. vertical cloud of red points to the top-right of the figure – 100 points sampled from a Gaussian distribution with $(3.5, 8.0)$ mean and 0.5 standard deviation.

FACE was initialised with $w(z) = -log(z)$ as the weight function and the l_2-norm as the distance function. Figures 2, 3 and 4 show the results of applying **FACE** on the toy data set when used with *KDE*, *e*-graph and *k*-NN respectively. In each, the triplet follows a similar pattern: (a) no



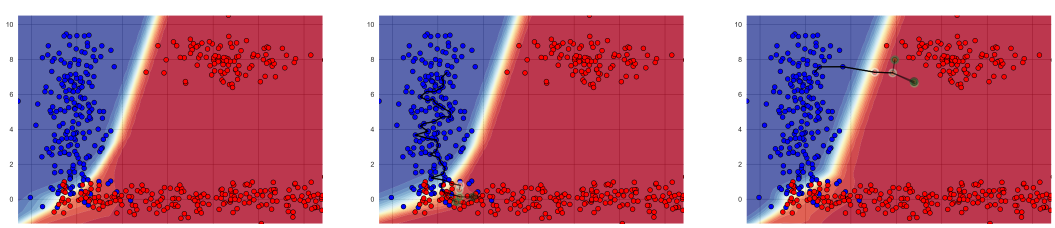
(a) $\varepsilon = 0.25$ distance threshold. (b) $\varepsilon = 0.50$ distance threshold. (c) $\varepsilon = 2$ distance threshold.

Figure 2: The five shortest paths from a starting data point to a target (counterfactual) data point generated from a graph, which edge weights were computed using the *KDE* approach. The targets are restricted by: i) $t_p \geq 0.75$ prediction threshold, ii) $t_d \geq 0.001$ density threshold and the distance threshold set to: (a) $\varepsilon = 0.25$, (b) $\varepsilon = 0.50$ and (c) $\varepsilon = 1$.



(a) $\varepsilon = 0.25$ distance threshold. (b) $\varepsilon = 0.50$ distance threshold. (c) $\varepsilon = 1$ distance threshold.

Figure 3: The five shortest paths from a starting data point to a target (counterfactual) data point generated from a graph, which edge weights were computed using the *e*-graph approach. The targets are restricted by $t_p \geq 0.75$ prediction threshold with the distance threshold set to: (a) $\varepsilon = 0.25$, (b) $\varepsilon = 0.50$ and (c) $\varepsilon = 1$.



(a) $k = 2$ neighbours and $\varepsilon = 0.25$ distance threshold. (b) $k = 4$ neighbours and $\varepsilon = 0.35$ distance threshold. (c) $k = 10$ neighbours and $\varepsilon = 0.80$ distance threshold.

Figure 4: The five shortest paths from a starting data point to a target (counterfactual) data point generated from a graph, which edge weights were computed using the k-NN graph approach. The targets are restricted by $t_p \geq 0.75$ prediction threshold with the ε distance threshold and k neighbours set to: (a) $k = 2$ and $\varepsilon = 0.25$; (b) $k = 4$ and $\varepsilon = 0.35$; and (c) $k = 10$ and $\varepsilon = 0.80$.

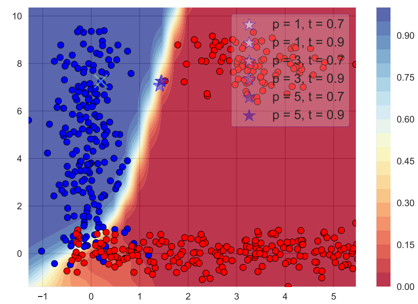


Figure 5: Counterfactuals generated using the method proposed by Wachter et al. [20]. p denotes the penalty parameter and t the classification threshold. These counterfactuals clearly do not comply with the desired properties described in Section 2.

counterfactual is generated, (b) a "good" counterfactual is generated, and (c) a "bad" counterfactual is generated.[4] Our experimental setup adheres to a real-life use case where **FACE** is originally applied with a fairly "restrictive" configuration, which is subsequently being relaxed until a counterfactual is found. Figure 5 shows the counterfactuals found by optimising Equation 5 proposed by Wachter et al. [20], which can be compared against the onse achived with **FACE** on the same data set (cf. Figures 2, 3 and 4).

MNIST Data Set To this end, we applied **FACE** (based on the k-NN construction algorithm with $k = 50$) to two images of the zero digit taken from the MNIST data set [7] with the target counterfactual class set to the digit eight. The underlying predictive model is a Neural Network trained on the whole MNIST data set. Figure 6 depicts the full path from the starting instance (left) to the final counterfactual (right). The resulting path shows a smooth transformation through the zeros until an eight is reached.

Figure 6: The "transformation" path achieved by applying **FACE** to compute a counterfactual example for two different images of a zero.

5 Related Work

Counterfactual explanations of Machine Learning models and predictions have been studied extensively in the recent years [15, 16, 14, 19]. Their popularity in Machine Learning is mainly attributed to the use of counterfactual explanations in everyday human life to explain phenomena that surround us [9], therefore they do not require the explainee to be familiar with Artificial Intelligence concepts to understand them. Despite this recent surge in popularity of counterfactual explanations in ML literature, they have been extensively studied in [10] social sciences, hence are well grounded as an explanatory technique. Furthermore, they have been deemed to satisfy the "Right to Explanation" requirement [20] introduced by the European Union's General Data Protection Act (GDPR), therefore making them a viable solution for many businesses applying predictive modelling to human matters.

To produce this type of explanations Wachter et al. [20] adapted the standard machinery used in the *Adversarial Examples* literature [4]:

$$\arg\min_{x'}\max_{\lambda}(f_w(x') - y')^2 + \lambda \cdot d(x, x'), \tag{5}$$

[4] "Good" and "bad" are with respect to the desired properties of counterfactuals discussed earlier in the paper.

where x and x' denote respectively the current state of the subject and the counterfactual, y' the desired outcome, $d(\cdot,\cdot)$ a distance function that measures the difficulty of moving from x to x' and f_w a classifier parametrised by w. The problem is optimised by iteratively solving for x' and increasing λ until a sufficiently close solution is found. Wachter et al. emphasise the importance of the distance function choice and suggest using the l_1-norm penalty on the counterfactual, which encourages sparse solutions, weighted by the *Median Absolute Deviation* (MAD) that for a feature k is given by:

$$MAD_k := median_{x\in\mathscr{X}}(|x_k - median_{x\in\mathscr{X}}(x)|,$$

which leads to the following formulation of the distance function to be used for Equation 5:

$$d(x,x') = \sum_{k=1}^{K} \frac{|x_k - x'_k|}{MAD_k}.$$

Wachter et al. deal with discrete variables by doing a separate execution of the optimisation problem, one for each unique value of every feature, and then choosing a counterfactual with the shortest distance. While it is possible to adopt techniques from the Adversarial Learning literature [20] and keep discarding the unactionable counterfactuals until an actionable one is found such an approach is clearly sub-optimal.

Ustun et al. [18] present an Integer Programming (IP) toolkit for linear models intended to be used by practitioners to analyse actionability and difficulty of recourse in a given population as well as generate actionable changes (counterfactuals) to subjects. Russell [16] propose a Mixed Integer Programming (MIP) formulation to handle mixed data types to offer counterfactual explanations for linear classifiers that respect the original data structure. This formulation is guaranteed to find coherent solutions (avoiding nonsense states) by only searching within the “mixed-polytope” structure defined by a suitable choice of linear constraints. Russell chose an iterative approach to providing diverse collection of counterfactuals. Given one solution, the user can add extra constraints to the MIP that will restrict previous alterations. The list of counterfactuals is then ranked according to their l_1-norm distance to the original instance.

van der Waa et al. [19] propose a counterfactual generation method for decision trees. Their approach uses locally trained one-vs-the-rest decision trees to establish a set of disjoint rules that cause the chosen instance to be classified as the target class.

FACE improves over all of the aforementioned counterfactual generation schemata in a number of ways:

- Contrarily to [20] and similarly to [18, 16, 19] it can handle discrete features and their restrictions in a more principled manner. For example, it natively supports features that cannot change, features which values can only change within a specified range and user preferences on subjective distance measures.
- Contrarily to [18, 16, 19] and similarly to [20] it is *model-agnostic* (not restricted to linear models or decision trees), hence it can handle any predictive model.
- Contrarily to [20, 18, 16, 19] it produces counterfactual explanations that are both feasible and actionable.

6 Summary and Future Work

In this paper we have highlighted the shortcomings of popular Counterfactual Explanation approaches in the Machine Learning literature and proposed a new method, called **FACE**, that aims at resolving them. Our approach accounts for both the nature of the target instance (the counterfactual) and the degree to which the proposed change is feasible and actionable. Our research has led us to uncover the dangers of ignoring this information when explaining automated predictions and the possible adverse impact this may have on both involved parties. We will continue this line of research by evaluating the performance of our approach on real-world data sets of dynamic nature and exploring the degree to which our suggested counterfactuals match the *true* change. Furthermore, we are interested in exploring the added value and the usefulness of the path itself for the explainee. The **FACE** algorithm has made great strides in addressing actionability and feasibility issues of currently available Counterfactual Explanation methods, nevertheless we believe that further exploration of alternative approaches is a fruitful area for further investigation.

References

[1] Morteza Alamgir and Ulrike Von Luxburg. Shortest path distance in random k-nearest neighbor graphs. *arXiv preprint arXiv:1206.6381*, 2012.

[2] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. 2009.

[3] Theodore Gamelin. *Complex analysis*. Springer Science and Business Media, 2003.

[4] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *stat*, 1050:20, 2015.

[5] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Scholkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

[6] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in Neural Information Processing Systems*, pages 2280–2288, 2016.

[7] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist/`.

[8] Lydia T Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. Delayed impact of fair machine learning. *arXiv preprint arXiv:1803.04383*, 2018.

[9] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 2018.

[10] Tim Miller, Piers Howe, and Liz Sonenberg. Explainable ai: Beware of inmates running the asylum or: How i learnt to stop worrying and love the social and behavioural sciences. *IJCAI 2017 Workshop on Explainable Artificial Intelligence (XAI)*, 2017.

[11] Gregory Plumb, Denali Molitor, and Ameet S Talwalkar. Model agnostic supervised local explanations. In *Advances in Neural Information Processing Systems*, pages 2515–2524, 2018.

[12] Marco Tulio Ribeiro, Sameer Singh, and Carlos 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*, pages 1135–1144. ACM, 2016.

[13] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[14] Marcel Jurriaan Robeer. Contrastive explanation for machine learning. Master's thesis.

[15] Cynthia Rudin. Please stop explaining black box models for high stakes decisions. *arXiv preprint arXiv:1811.10154*, 2018.

[16] Chris Russell. Efficient search for diverse coherent explanations. *arXiv preprint arXiv:1901.04909v1*, 2019.

[17] Sajama and Alon Orlitsky. Estimating and computing density based distance metrics. In *Proceedings of the 22nd international conference on Machine learning*, pages 760–767. ACM, 2005.

[18] Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. *arXiv preprint arXiv:1809.06514*, 2018.

[19] Jasper van der Waa, Marcel Robeer, Jurriaan van Diggelen, Matthieu Brinkhuis, and Mark Neerincx. Contrastive explanations with local foil trees. *arXiv preprint arXiv:1806.07470*, 2018.

[20] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. 2017.