Convex Density Constraints for Computing Plausible Counterfactual Explanations*

André Artelt[1] and Barbara Hammer[1]

CITEC - Cognitive Interaction Technology
Bielefeld University, 33619 Bielefeld, Germany
{aartelt,bhammer}@techfak.uni-bielefeld.de

Abstract. The increasing deployment of machine learning as well as legal regulations such as EU's GDPR cause a need for user-friendly explanations of decisions proposed by machine learning models. Counterfactual explanations are considered as one of the most popular techniques to explain a specific decision of a model. While the computation of "arbitrary" counterfactual explanations is well studied, it is still an open research problem how to efficiently compute plausible and feasible counterfactual explanations. We build upon recent work and propose and study a formal definition of plausible counterfactual explanations. In particular, we investigate how to use density estimators for enforcing plausibility and feasibility of counterfactual explanations. For the purpose of efficient computations, we propose convex density constraints that ensure that the resulting counterfactual is located in a region of the data space of high density.

Keywords: XAI · Counterfactual Explanations · Transparency & Interpretability.

1 Introduction

As research on machine learning (ML) is making more and more progress and ML models constitute state-of-the-art approaches in domains such as machine translation, image and text classification, we observe an increased deployment of ML technology in practice [12,16,32]. At the same time, ML models are vulnerable to unexpected behavior such as adversarial attacks [29] and behavior which is regarded as unfair by humans [21], hence a large amount of the decision making process offered by ML is not fully understood by humans. As a consequence of this fact and due to legal regulations like EU's GDPR [23], transparency and interpretability of ML models becomes more and more relevant. Therefore, there is a need for tools that make ML models transparent in the sense that we can explain the decision making process of a model. Accordingly, we observe an increase of research in the area of explainable AI (XAI) [11,15,28,30].

* We gratefully acknowledge funding from the VW-Foundation for the project *IMPACT* funded in the frame of the funding line *AI and its Implications for Future Society.*

Over time, researchers developed a diverse set of methods for explaining ML models [15,22]: Model-agnostic methods [15,25] are not tailored to a particular model or representation, hence they are (in theory) applicable to any different types of ML models; in the extreme "truly" model-agnostic methods do not need access to the training data or model internals but they regard the model as a black-box. There exists a variety of different model-agnostic approaches, including feature interaction methods [13], feature importance methods [9], partial dependency plots [34] and local methods that approximates the model locally by an explainable model [14,26]. This group of technologies relies on feature importance ranking or similar to express decisions of a given model. A different class of explanations relies on examples that explain a prediction by a (set of) data points [2]. Prototypes & criticisms [17] and influential instances [18] are instances of such example-based explanations.

One popular instance of example-based explanations, often realized as black-box scheme, are counterfactual explanations [22,31]. A counterfactual explanation states a change to the original input that leads to a different prediction of a given ML model. This type of explanation is considered as particularly intuitive, because it tells the user what to do in order to achieve a desired goal [22,31]. Despite the huge variety of different - equally important - types of explanations, we limit ourselves to counterfactual explanations in this contribution. Counterfactual explanations can be phrased as a constrained optimization problem, aiming for minimizing the change which results in a different output. Depending on the specific setting, this optimization problem is solved by either gradient-based schemes or, in particular in agnostic settings, by black-box solvers. Thereby, approaches which rely on the specific form of the given classifier can lead to much more efficient computation schemes, as demonstrated in [6].

Yet, stated in its simplest form, counterfactuals are very similar to adversarial examples, since there are no guarantees that the resulting counterfactual is plausible and feasible in the data domain. As a consequence, the absence of such constraints often leads to counterfactual explanations that are not plausible [8,19,24] - an observation that we will also confirm in this work.

In this work, we aim for an extension of counterfactual explanation schemes which restricts possible explanations to plausible regions of the data space. More specifically, we propose and study a formal definition of plausible counterfactual explanations and propose a modeling framework, which phrases such constraints in convex form, such that they can efficiently be integrated into optimization schemes, preserving uniqueness of solutions or efficiency if this is valid for the constrained version.

2 Definition and Related Work

We briefly review existing work on enforcing plausibility of counterfactual explanations (Definition 1). In the context of ML models, counterfactual explanations are formalized as follows:

Definition 1 (Counterfactual explanation [31]). *Assume a prediction function h is given. Computing a counterfactual $\boldsymbol{x}' \in \mathbb{R}^d$ for a given input $\boldsymbol{x} \in \mathbb{R}^d$ is phrased as the following optimization problem:*

$$\underset{\boldsymbol{x}' \in \mathbb{R}^d}{\arg\min} \; \ell\left(h(\boldsymbol{x}'), y^c\right) + C \cdot \theta(\boldsymbol{x}', \boldsymbol{x}) \tag{1}$$

where $\ell(\cdot, \cdot)$ denotes a loss function, y^c the requested prediction, and $\theta(\cdot, \cdot)$ a penalty term for deviations of $\boldsymbol{x}'$ from the original input $\boldsymbol{x}$. $C > 0$ denotes the regularization strength.

Two common regularizations are the weighted Manhattan distance and the Mahalanobis distance. The weighted Manhattan distance is defined as:

$$\theta(\boldsymbol{x}', \boldsymbol{x}) = \sum_j \alpha_j \cdot |(\boldsymbol{x})_j - (\boldsymbol{x}')_j| \tag{2}$$

where $\alpha_j > 0$ denote feature-wise weights. The Mahalanobis distance is defined as:

$$\theta(\boldsymbol{x}', \boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x}')^\top \boldsymbol{\Omega} (\boldsymbol{x} - \boldsymbol{x}') \tag{3}$$

where $\boldsymbol{\Omega}$ denotes a s.psd. matrix.

In general, $\boldsymbol{x}'$ is arbitrary, hence possibly implausible. A variety of approaches aims for a restriction of the domain to plausible patterns only. The authors of [24] propose to compute a path of intermediate counterfactuals that lead to the final counterfactual. The idea is to provide the user with a set of intermediate goals that finally lead to the desired goal - it might be easier to "go into the direction" of the final goal step by step instead of accomplishing it in a single step. In order to compute such a path of intermediate counterfactuals, different strategies for constructing a graph on the training data set are proposed - including the query point. In this graph, two samples are connected by a weighted edge if they are "sufficiently close to each other" - e.g. based on density estimation. The path of intermediate counterfactuals is then computed as the shortest path between the query point and a point that has the requested label - this data point is the final counterfactual. Therefore, the final counterfactual as well as all intermediate counterfactuals are elements of the training data set, which ensures that all counterfactuals are plausible and feasible. However, the limitation to samples from the training set can be seen as a major drawback of this method, in particular for sparsely populated data spaces.

A slightly modified version of Eq. (1) was proposed in [19]. The authors suggest that the original formalization Eq. (1) does not take into account that the counterfactual should lie on the data manifold which would enforce plausibility. Therefore, they propose to add two additional terms to the original objective in Eq. (1), which should be simultaneously optimized:

1. The distance between the counterfactual $\boldsymbol{x}'$ and the reconstructed version of it that has been computed by using a pretrained autoencoder.
2. The distance between the encoding of the counterfactual $\boldsymbol{x}'$ and the mean encoding of training samples that belong to the requested class y^c.

The first term is supposed to ensure that the counterfactual $\boldsymbol{x}'$ lies on the data manifold and thus is a plausible data instance. The second term is supposed to accelerate the solver for computing the solution of the final optimization problem. We think that this is a very promising approach - However, the objective itself still behaves like "a heuristic" because, like the original Eq. (1), there are no guarantees that the resulting counterfactual is plausible/feasible or even valid at all - one would have to do an extensive hyperparameter tuning of the objective. Furthermore, the need of a working autoencoder can be considered as another bottleneck because building high quality and stable autoencoders can be quite challenging if only very little data are available - in particular if the autoencoder is modeled by deep neural networks. Lastly, due to the non-convexity of the autoencoder and the model itself, the resulting optimization problem is highly non-convex and thus difficult to solve.

Somehow similar to [19], the authors of [20] propose to use GANs and VAEs for creating realistic images. Although they do not talk explicitly about counterfactuals - they want to compute contrastive explanations[1] [8] which are similar to counterfactuals in the sense that in both cases we want to find a minimal change that leads to a specific prediction (although we have a second objective in constrastive explanations).

The authors of [5] propose a convex modeling framework for efficiently computing counterfactual explanations of different ML models. They propose to turn the optimization problem Eq. (1) into a constraint optimization problem:

$$\underset{\boldsymbol{x}' \in \mathbb{R}^d}{\arg\min}\ \theta(\boldsymbol{x}', \boldsymbol{x}) \quad \text{s.t. } h(\boldsymbol{x}') = y^c \tag{4}$$

By exploiting model specific structures, they are able to turn Eq. (4) into a convex program for many different ML models. The benefits of this modeling are that convex programs can be solved very efficiently [7], additional convex constraints can be added without changing the complexity of the problem, feasibility - does a solution (counterfactual), under a given set of constraints, exist? - can be verified easily. By adding additional constraints we can ensure that the counterfactual is plausible/feasible in the specific data domain. However, manually constructing plausibility constraints can be very time consuming and requires solid domain knowledge which might not be available. These approaches yield promising approaches, yet their greatest disadvantage is the potentially high computational load of the induced optimization problem. Here, we will take a different avenue by phrasing the condition of plausibility as a convex constraint.

Our contribution builds on our prior work [5], which phrases counterfactual computation in terms of efficient constrained optimization problems for many

[1] A contrastive explanations states a minimal amount of (present and absent) features (including their values) that are responsible for a specific prediction. Such an explanation is computed by finding a minimal perturbation to the input that yields the same (present features) or different (absent features) prediction. In order to stay close to the data manifold - enforce that the results are plausible - they propose to use an autoencoder.

popular classifiers. Besides a formal definition of plausible counterfactuals, we propose convex density constraints that can be built from a given data set automatically and efficiently. These constraints ensure that the density of the resulting counterfactual is lower bounded by a predefined/requested threshold. Note that all proofs and derivations can be found in the appendix A.

3 Plausible Counterfactual Explanations

3.1 Computation of Plausible Counterfactual Explanations

For the purpose of enforcing plausibility of counterfactuals, we propose to add a target specific density constraint to Eq. (4):

$$\underset{\boldsymbol{x}' \in \mathbb{R}^d}{\arg\min}\ \theta(\boldsymbol{x}', \boldsymbol{x}) \tag{5a}$$

$$\text{s.t. } h(\boldsymbol{x}') = y^c \tag{5b}$$

$$\hat{p}_y(\boldsymbol{x}') \geq \delta \tag{5c}$$

where $\hat{p}_y(\cdot)$ denotes a class dependent density estimator.

There exists a variety of different density estimators that estimate the density based on training samples.

A kernel density estimator (KDE) is a popular choice when it comes to estimate densities from training data. A kernel density estimator is a non-parametric model and is defined as:

$$\hat{p}_{\text{KDE}}(\boldsymbol{x}) = \sum_i \alpha_i k(\boldsymbol{x}, \boldsymbol{x}_i) \tag{6}$$

where $k(\cdot, \cdot)$ denotes a suitable kernel function, $\boldsymbol{x}_i$ denotes the i-th sample in the training data set and $\alpha_i > 0$ denotes the weighting of the i-th sample. However, in case of non-linear kernels (e.g. Gaussian kernel) the resulting density estimator is highly non-convex and does not induce an efficient optimization problem.

In a Gaussian mixture model (GMM) the density is modeled as a mixture of multivariate normal distributions. The density under a GMM with m components is defined as:

$$\hat{p}_{\text{GMM}}(\boldsymbol{x}) = \sum_{j=1}^{m} \pi_j \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \tag{7}$$

where π_j denotes the prior probability of the j-th component, $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ denote the mean and covariance of the j-th component. Although the GMM Eq.(7) is much simpler (has fewer components/parameters) than a kernel density estimator Eq. (6), it still does not induce convex constraints for Eq. (5c).

Here we propose to approximate the density of a GMM Eq. (7) by a component wise maximum:

$$\hat{p}(\boldsymbol{x}) = \max_j \Big(\hat{p}_j(\boldsymbol{x}) \Big) \tag{8}$$

where

$$\hat{p}_j(\boldsymbol{x}) = \pi_j \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \tag{9}$$

By construction, the approximation Eq. (8) is always a lower bound of the true GMM density Eq. (7). More precisely, the following bound holds:

$$\hat{p}(\boldsymbol{x}) \leq \hat{p}_{\text{GMM}}(\boldsymbol{x}) \leq m \cdot \hat{p}(\boldsymbol{x}) \tag{10}$$

The inequality constraint of a single component Eq. (9)

$$\hat{p}_j(\boldsymbol{x}) = \pi_j \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \geq \delta \tag{11}$$

can be rewritten as a convex quadratic constraint:

$$(\boldsymbol{x} - \boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j) + c_j \leq \delta' \tag{12}$$

where

$$c_j = -2\log(\pi_j) + d\log(2\pi) - \log\Big(\det(\boldsymbol{\Sigma}_j^{-1})\Big) \qquad \delta' = -2\log(\delta) \tag{13}$$

By making use of the approximation Eq. (8), the original constraint Eq. (5c) becomes:

$$\min_j \Big((\boldsymbol{x}' - \boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x}' - \boldsymbol{\mu}_j) + c_j \Big) \leq \delta' \tag{14}$$

Although Eq. (14) is still non-convex, we can turn it into a set of convex constraints by observing the following:

Let $\boldsymbol{x}'_*$ be a solution of Eq. (5) where we substituted Eq. (5c) by Eq. (14). Then it holds that:

$$\exists\, j \in \{1, \ldots, m\} : \; (\boldsymbol{x}'_* - \boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x}'_* - \boldsymbol{\mu}_j) + c_j \leq \delta' \tag{15}$$

Note that there might exists more than one j for which Eq. (11) holds. Because we do not know for which j Eq. (11) holds, we simply try all possible $j \in \{1, \ldots, m\}$ and select the counterfactual $\boldsymbol{x}'$ that yields the smallest value of the objective Eq. (5a) - that is the closest to the original input $\boldsymbol{x}$. Note that depending on the prediction function $h(\cdot)$ it can happen that Eq. (5) is not feasible for all j. Because each constraint Eq. (11) can be rewritten as a convex quadratic constraint, the final optimization problem Eq. (5) becomes convex iff the objective Eq.(5a) and the prediction constraint Eq. (5b) are convex. The Manhattan distance as well as the Mahalanobis distance as regularizers $\theta(\cdot, \cdot)$ together with common ML models - like generalized linear models, linear SVM, LDA, matrix LVQ, decision tree, etc. - yield convex programs [5] that can be solved efficiently [7].

3.2 A Formal Approach

We aim for a formal description of plausible counterfactuals as modelled in Eq. (5).

We assume a classification setting with an underlying generating process $\Psi = (\mathcal{X}, \mathcal{Y}, p)$ where the measurable set $\mathcal{X}$ denotes the data domain, the discrete and finite set $\mathcal{Y}$ denotes the set of possible labels and $p : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}_+$ denotes the joint density - we assume that $\{(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y} \mid p(\boldsymbol{x}, y) \geq \delta\}$ is closed for all $\delta > 0$. Furthermore, let $\theta : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}_+$ be a distance metric on $\mathcal{X}$. Following Eq. (5), we propose to define a *plausible counterfactual* according to Definition 2.

Definition 2 (δ-plausible counterfactual). *Let $h : \mathcal{X} \mapsto \mathcal{Y}$ be a classifier. We call a counterfactual explanation $(\boldsymbol{x}', y^c)$ of a particular sample $\boldsymbol{x} \in \mathcal{X}$ δ-plausible iff the following holds:*

$$\boldsymbol{x}' = \underset{\boldsymbol{x}' \in \mathcal{X}}{\arg\min}\ \theta(\boldsymbol{x}', \boldsymbol{x}) \quad s.t.\ h(\boldsymbol{x}') = y^c\ \wedge\ p(\boldsymbol{x}', y^c) \geq \delta \tag{16}$$

where $\delta > 0$ denotes a minimum density at which we consider a sample plausible. Note that we state the definition of an δ-plausible counterfactual as an optimization problem Eq.(16) which makes the definition particular appealing from a practical perspective.

Next, in Theorem 1 we state under what conditions δ-plausible counterfactuals do not depend on the classifier.

Theorem 1 (Model free δ-plausible counterfactuals under zero risk classifiers). *Let $\mathcal{H}$ be the set of all classifiers $h : \mathcal{X} \mapsto \mathcal{Y}$ that have zero risk on the generating process Ψ - that is: $h \in \mathcal{H} \Leftrightarrow \underset{\boldsymbol{x}, y \sim p}{\mathbb{E}} [\mathbb{1}\left(h(\boldsymbol{x}) \neq y\right)] = 0$. Then the following holds $\forall\, h \in \mathcal{H}, (\boldsymbol{x}, y^c) \in \mathcal{X} \times \mathcal{Y} \setminus \{y\}$:*

$$\begin{aligned} &\underset{\boldsymbol{x}' \in \mathcal{X}}{\arg\min}\ \theta(\boldsymbol{x}', \boldsymbol{x})\ s.t.\ h(\boldsymbol{x}') = y^c\ \wedge\ p(\boldsymbol{x}', y^c) \geq \delta \\ &\Leftrightarrow \underset{\boldsymbol{x}' \in \mathcal{X}}{\arg\min}\ \theta(\boldsymbol{x}', \boldsymbol{x})\ s.t.\ p(\boldsymbol{x}', y^c) \geq \delta \end{aligned} \tag{17}$$

Note that Theorem 1 states that in the case of perfect classifiers, δ-plausible counterfactuals become independent from the specific classifiers - thus we can compute the δ-plausible counterfactuals solely in the data domain without taking the classifiers into account.

However, in practice we usually do not have a perfect classifier because either the class wise densities are overlapping or the classifier itself can not model a zero risk decision boundary. Therefore, we state a weaker version of Theorem 1 in Theorem 2, in which we assume that a classifiers h is locally δ-sufficient perfect at a sample $(\boldsymbol{x}, y)$ (Definition 3) - that is: the classifier h classifies the sample $\boldsymbol{x}$ as y, which is consistent with the ground truth induced by the generating process Ψ, and the decision boundary does not "cut to deep" into the closest parts of high density regions of the other classes.

Definition 3 (Locally δ-sufficient perfect classifier). *Let $h : \mathcal{X} \mapsto \mathcal{Y}$ be a classifier and denote the set of all $\boldsymbol{x} \in \mathcal{X}$ that have a class dependent density of at least δ by $\mathcal{X}_\delta(y^c)$ - that is: $\mathcal{X}_\delta(y) = \{\boldsymbol{x} \in \mathcal{X} \mid p(\boldsymbol{x}, y) \geq \delta\}$. We call h locally δ-sufficient perfect at a sample $(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}$ iff the following holds:*

$$h(\boldsymbol{x}) = \underset{y_i \in \mathcal{Y}}{\arg\max}\, p(\boldsymbol{x}, y_i) = y\ \wedge\ h(\boldsymbol{x}_*) = y^c\ \ \forall\, y^c \in \mathcal{Y} \setminus \{y\},\ \boldsymbol{x}_* = \underset{\boldsymbol{z} \in \mathcal{X}_\delta(y^c)}{\arg\min}\ \theta(\boldsymbol{z}, \boldsymbol{x}) \tag{18}$$

Theorem 2 (Model free δ-plausible counterfactual under locally δ-sufficient perfect classifiers). *Let $\mathcal{H}(\boldsymbol{x}, y)$ be the set of locally δ-sufficient*

perfect classifiers (Definition 3) at a sample $(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}$*. Then the following holds* $\forall\, h \in \mathcal{H}(\boldsymbol{x}, y), (\boldsymbol{x}, y^c) \in \mathcal{X} \times \mathcal{Y} \setminus \{y\}$*:*

$$\begin{aligned} &\arg\min_{\boldsymbol{x}' \in \mathcal{X}} \; \theta(\boldsymbol{x}', \boldsymbol{x}) \; s.t. \; h(\boldsymbol{x}') = y^c \wedge p(\boldsymbol{x}', y^c) \geq \delta \\ &\Leftrightarrow \arg\min_{\boldsymbol{x}' \in \mathcal{X}} \; \theta(\boldsymbol{x}', \boldsymbol{x}) \; s.t. \; p(\boldsymbol{x}', y^c) \geq \delta \end{aligned} \tag{19}$$

Note that Theorem 2 states that for a set of classifiers that are locally δ-sufficient perfect at a sample $(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}$ (Definition 3), the δ-plausible counterfactuals of this particular sample $\boldsymbol{x}$ are exactly the same for all classifiers in this set. Because we only assume locally δ-sufficient perfectness of the classifier, Theorem 2 is very appealing for practice when we actually have to compute a counterfactual explanation of a particular sample under a particular model - the theorem tells us when we can drop the classification constraint and thus simplify the optimization problem Eq. (16).

In practice, when the true density (or a density estimation) is not available, one could try to check for locally δ-sufficient perfectness at a given sample $\boldsymbol{x}$ by checking if the "closest" training samples (incl. samples from different classes) around $\boldsymbol{x}$ are classified correctly.

4 Experiments

We perform experiments on several data sets[2] for empirically evaluating our proposed density constraints Eq. (14). We use the "Breast Cancer Wisconsin (Diagnostic) Data Set" [33], the "Iris Plants Data Set" [10], the "Wine Data Set" [27], the "Boston Housing Data Set" [1][3] and the "Optical Recognition of Handwritten Digits Data Set" [3]. We repeat the following procedure in a 4-fold cross validation: First, we fit class dependent kernel density estimators (we use the Gaussian kernel) and a GMM to the training data set - where we use a 5-fold cross validation grid search for hyperparameter tuning. Next, we fit a classifier (either a softmax regression or decision tree)[4] to the training data set. After this, for each sample in the test set, we compute two counterfactuals (both with the same but random target class) - one counterfactual without any additional density/plausibility constraints and another counterfactual with our proposed density constraint Eq. (12). We set the density threshold δ from Eq. (11) to the median density Eq. (8) of the training samples under the approximated GMM of the target class y^c. To enforce sparsity, both counterfactuals are computed under the Manhattan distance as a regularizer $\theta(\cdot, \cdot)$. Finally, we compute the Manhattan distance to the original sample and the log-density of both counterfactuals under

[2] Our source code is available on GitHub - https://github.com/andreArtelt/ConvexDensityConstraintsForPlausibleCounterfactuals

[3] We turn it into a binary classification problem by setting the target to 1 if the price is greater or equal to 20k$.

[4] An implementation of the experiments including other models like LDA, linear SVM, matrix LVQ, etc. is available online.

Table 1: Median log-density (under the KDE) and median Manhattan distance to the original sample of the computed counterfactuals - with vs. without density constraints. Best values are **highlighted** - larger densities and smaller distances are better.

		Without density constraints		With density constraints	
	Data set	Density	Distance	Density	Distance
Softmax regression	Iris	-34.55	**1.80**	**-0.75**	4.06
	Digits	-164.03	**36.74**	**-112.40**	110.10
	Wine	-82.31	**5.19**	**-37.58**	49.59
	Breast cancer	-46.52	**33.26**	**-27.0**	81.47
	House prices	39.51	**5.0**	**-38.12**	9.54
Decision tree	Iris	-40.55	**1.19**	**-0.73**	4.06
	Digits	-170.25	**36.69**	**-110.48**	114.78
	Wine	-102.44	**3.92**	**-34.38**	66.92
	Breast cancer	-43.44	**0.01**	**-25.55**	22.27
	House prices	-40.49	**0.01**	**-37.84**	14.92

the kernel density estimator. We use the kernel density estimator instead of the GMM because our proposed density constraint is an approximation of the GMM which itself can be interpreted as an approximation of the kernel density estimator. In order to increase the accuracy of the classifiers and density estimators, we apply a PCA to the breast cancer data set (5 components), the house prices data set (10 components), the wine data set (8 components) and the digits data set (40 components). Since the PCA transformation is affine, it can be easily integrated into our convex programs - so that we can still compute counterfactuals in the original space.

The results of the experiments are listed in Table 1. We observe that our proposed density constraint consistently yields counterfactuals that have a higher density than the counterfactuals without any additional density/plausibility constraints - whereby we only observe a minor increase in computation time (e.g. from 30ms to 70ms per sample). However, the distance to the original sample is much higher for the "more plausible" counterfactuals than for arbitrary (e.g. closest) counterfactuals. This seems reasonable because one would expect that samples from a different class look quite differently. In addition, we observe that the distances of the counterfactuals to the original samples on the Iris data set and Digits data set are more or less the same for both models, whereas the opposite is true for the wine, breast cancer and house prices data sets. This observation can be explained by the hypothesis that in the case of Iris and digits data set, both models learned a locally δ-sufficient perfect classifier (Definition 3) at most samples - then Theorem 2 states that the counterfactuals are model independent which explains the observed numbers. Conversely, this suggests that the two classifiers learned on the other three data sets are quite different in the sense that they are not all locally δ-sufficient perfect classifiers (Definition 3) at

most samples - hence, the distances of the counterfactuals to the original samples are quite different.

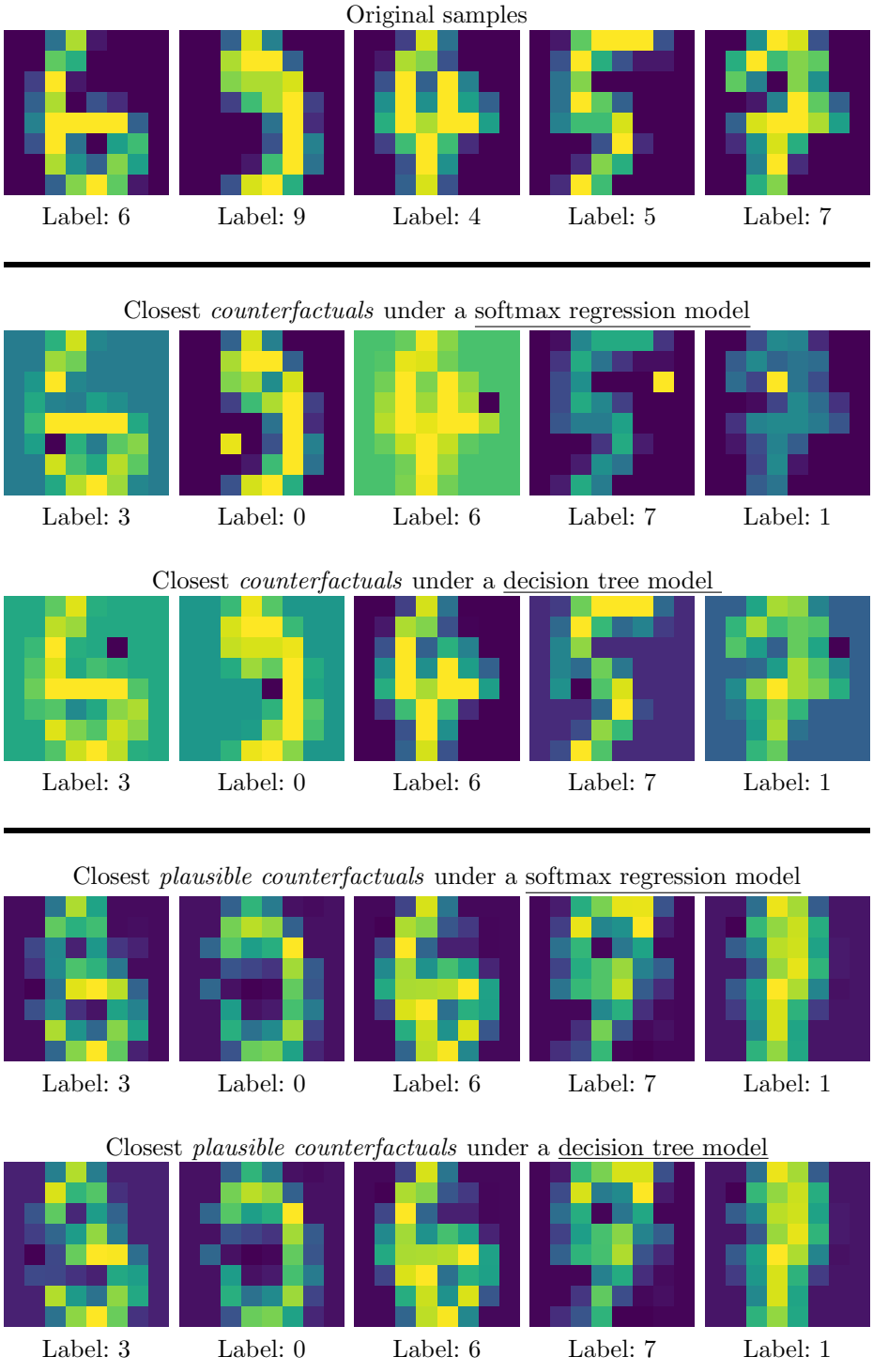
Furthermore, Fig. 1 shows some samples from the digit data set and compares the counterfactuals generated with and without density constraints of both models. Most of the samples in the second block - counterfactuals without any density/plausibility constraints - look like adversarials in the sense that the original label can be still recognized but the requested label can not be inferred. However, most of the samples in the third block - counterfactuals that have been computed with our proposed density constraint - look like samples from the requested target class. This suggests that our method in fact yields plausible counterfactuals. We also observe that the two models yield different counterfactuals in the second block but more or less exactly the same counterfactuals in the third block. As already discussed in the case of the very similar distances in Table 1, this can be explained by assuming that both models are (close to) locally δ-sufficient perfect (Definition 3) at most samples, which confirms the observations as it is predicted by Theorem 2. However, please note that a visual inspection of some samples does not replace a proper evaluation by doing an expert user study and subsequent hypotheses testings.

5 Discussion and Conclusion

In this work, we proposed and studied a formal definition of plausible counterfactual explanations. In this definition we proposed to add density constraints to the optimization problem for computing counterfactual explanations to ensure that the resulting counterfactual is plausible in the given data domain. For practical purposes, we proposed convex approximations of a Gaussian mixture model to get tractable density constraints. These constraints give rise to convex optimization problems for computing plausible counterfactual explanations many common models like linear models and decision trees. In addition, these constraints allow to specify a lower bound on the density of the resulting counterfactual that is guaranteed to be full filled. Finally, we empirically evaluate our proposed methods on several data sets and observe that our method consistently yields counterfactual explanations that are located in high density regions. A visual inspection of samples from the digits data set suggests that in fact our method seems to yield plausible counterfactuals.

As future work, we plan to conduct a proper user study where humans judge the plausibility of generated counterfactual explanations - counterfactuals generated with and without density constrains. Furthermore, we want to explore density estimators for high dimensional data so that our method can be used for high dimensional data, too. We also plan to investigate how to add density constraints for computing counterfactual explanations of more complex models - in particular non-linear models (e.g. Deep neural networks). Lastly, our source code will be released as part of our open-source toolbox CEML [4], a Python toolbox for computing counterfactual explanations of ML models, so that our proposed method can be easily used by practitioners.

Fig. 1: Samples from the digit data set. *First block:* Original samples. *Second block:* Counterfactuals generated without any density/plausibility constraints. *Third block:* Counterfactuals generated with our proposed density constraint. The corresponding labels are shown below each image - note that the shown labels of the counterfactuals are the requested labels.



A Proofs and Derivations

1. *Proof (Theorem 1).* For a given generating process Ψ, zero risk classifiers exist iff the class-dependent densities are non-overlapping:

$$\begin{aligned} &\exists\, h : \mathop{\mathbb{E}}_{\boldsymbol{x},y\sim p}[\mathbb{1}\left(h(\boldsymbol{x}) \neq y\right)] = 0 \\ &\Leftrightarrow \forall\, \boldsymbol{x} \in \mathcal{X} :\ p(\boldsymbol{x}, y) \neq 0 \Leftrightarrow p(\boldsymbol{x}, y_i) = 0\ \forall\, y_i \in \mathcal{Y} \setminus \{y\} \end{aligned} \tag{20}$$

Therefore, for a zero risk classifier h it holds that:

$$p(\boldsymbol{x}, y) > 0 \implies h(\boldsymbol{x}) = y \tag{21}$$

If follows that $\forall\, (\boldsymbol{x}, y^c) \in \mathcal{X} \times \mathcal{Y}$:

$$\begin{aligned} &\arg\min_{\boldsymbol{x}' \in \mathcal{X}}\ \theta(\boldsymbol{x}', \boldsymbol{x}) \quad \text{s.t. } h(\boldsymbol{x}') = y^c\ \wedge\ p(\boldsymbol{x}', y^c) \geq \delta \\ &\Leftrightarrow \arg\min_{\boldsymbol{x}' \in \mathcal{X}}\ \theta(\boldsymbol{x}', \boldsymbol{x}) \quad \text{s.t. } p(\boldsymbol{x}', y^c) \geq \delta \end{aligned} \tag{22}$$

Thus, the constraint $h(\boldsymbol{x}') = y^c$ in Eq. (16) becomes redundant - the counterfactuals of zero risk classifiers do not depend on these classifiers. □

2. *Proof (Theorem 2).* For a locally δ-sufficient perfect classifier h (Definition 3) at $\boldsymbol{x} \in \mathcal{X}$, it holds that:

$$h(\boldsymbol{x}_*) = y^c\ \ \forall\, y^c \in \mathcal{Y} \setminus \{y\},\ \boldsymbol{x}_* = \arg\min_{\boldsymbol{z}\,\in\, \mathcal{X}_\delta(y^c)}\ \theta(\boldsymbol{z}, \boldsymbol{x}) \tag{23}$$

where $\boldsymbol{x}_*$ is a δ-plausible counterfactual (Definition 2) of $\boldsymbol{x}$. It follows that:

$$\begin{aligned} &\arg\min_{\boldsymbol{x}' \in \mathcal{X}}\ \theta(\boldsymbol{x}', \boldsymbol{x}) \quad \text{s.t. } h(\boldsymbol{x}') = y^c\ \wedge\ p(\boldsymbol{x}', y^c) \geq \delta \\ &\Leftrightarrow \arg\min_{\boldsymbol{x}' \in \mathcal{X}}\ \theta(\boldsymbol{x}', \boldsymbol{x}) \quad \text{s.t. } p(\boldsymbol{x}', y^c) \geq \delta \end{aligned} \tag{24}$$

Thus, the constraint $h(\boldsymbol{x}') = y^c$ in Eq. (16) becomes redundant - the counterfactuals of a sample $\boldsymbol{x} \in \mathcal{X}$ of classifiers that are locally δ-sufficient perfect at $\boldsymbol{x}$ do not depend on these classifiers. □

3. *Proof (Bound in Eq. (10)).* It holds that:

$$\pi_j \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \geq 0 \quad \forall\, j \in \{1, \ldots, m\} \tag{25}$$

Therefore, it follows that:

$$\hat{p}(\boldsymbol{x}) = \max_j \Big(\pi_j \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)\Big) \leq \sum_{j=1}^{m} \pi_j \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) = \hat{p}_{\text{GMM}}(\boldsymbol{x}) \tag{26}$$

which proves the lower bound in Eq. (10).
It holds that:

$$\hat{p}(\boldsymbol{x}) = \max_j \Big(\pi_j \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)\Big) \geq \pi_i \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \quad \forall\, i \in \{1, \ldots, m\} \tag{27}$$

Because of Eq. (25) and Eq. (27), it follows that:

$$\hat{p}_{\text{GMM}}(\boldsymbol{x}) = \sum_{j=1}^{m} \pi_j \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \leq m \cdot \max_j \Big(\pi_j \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \Big) = m \cdot \hat{p}(\boldsymbol{x}) \tag{28}$$

which proves the upper bound in Eq. (10). □

4. Eq. (11) can be rewritten as the convex quadratic constraint Eq. (12):

$$\begin{aligned}
&\pi_j \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \geq \delta \\
&\Leftrightarrow \log \Big(\pi_j \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \Big) \geq \log(\delta) \\
&\Leftrightarrow -\log \Big(\pi_j \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \Big) \leq -\log(\delta) \\
&\Leftrightarrow (\boldsymbol{x} - \boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j) - 2\log(\pi_j) + d\log(2\pi) - \log\Big(\det(\boldsymbol{\Sigma}_j^{-1}) \Big) \leq -2\log(\delta)
\end{aligned} \tag{29}$$

References

1. Boston housing data set (1978), https://archive.ics.uci.edu/ml/datasets/Housing
2. Aamodt, A., Plaza., E.: Case-based reasoning: Foundational issues, methodological variations, and systemapproaches. AI communications (1994)
3. Alpaydin, E., Kaynak, C.: Optical recognition of handwritten digits data set. https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits (1998)
4. Artelt, A.: Ceml: Counterfactuals for explaining machine learning models - a python toolbox. https://www.github.com/andreArtelt/ceml (2019)
5. Artelt, A., Hammer, B.: On the computation of counterfactual explanations - A survey. CoRR **abs/1911.07749** (2019), http://arxiv.org/abs/1911.07749
6. Artelt, A., Hammer, B.: Efficient computation of counterfactual explanations of LVQ models. In: Verleysen, M. (ed.) Proceedings of the 28th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2020) (2020), accepted
7. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, New York, NY, USA (2004)
8. Dhurandhar, A., Chen, P., Luss, R., Tu, C., Ting, P., Shanmugam, K., Das, P.: Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In: Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada. pp. 590–601 (2018)
9. Fisher, A., Rudin, C., Dominici, F.: All Models are Wrong but many are Useful: Variable Importance for Black-Box, Proprietary, or Misspecified Prediction Models, using Model Class Reliance. arXiv e-prints arXiv:1801.01489 (Jan 2018)
10. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annual Eugenics **7 Part II**, 179–188 (1936)
11. Gilpin, L.H., Bau, D., Yuan, B.Z., Bajwa, A., Specter, M., Kagal, L.: Explaining explanations: An overview of interpretability of machine learning. In: 5th IEEE International Conference on Data Science and Advanced Analytics, DSAA 2018, Turin, Italy, October 1-3, 2018. pp. 80–89 (2018). https://doi.org/10.1109/DSAA.2018.00018, https://doi.org/10.1109/DSAA.2018.00018

12. Goel, S., Rao, J.M., Shroff, R.: Precinct or prejudice? understanding racial disparities in new york city's stop-and-frisk policy (2016)
13. Greenwell, B.M., Boehmke, B.C., McCarthy, A.J.: A simple and effective model-based variable importance measure. CoRR **abs/1805.04755** (2018), http://arxiv.org/abs/1805.04755
14. Guidotti, R., Monreale, A., Ruggieri, S., Pedreschi, D., Turini, F., Giannotti, F.: Local rule-based explanations of black box decision systems. CoRR **abs/1805.10820** (2018), http://arxiv.org/abs/1805.10820
15. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM Comput. Surv. **51**(5), 93:1–93:42 (Aug 2018). https://doi.org/10.1145/3236009, http://doi.acm.org/10.1145/3236009
16. Khandani, A.E., Kim, A.J., Lo, A.: Consumer credit-risk models via machine-learning algorithms. Journal of Banking & Finance **34**(11), 2767–2787 (2010), https://EconPapers.repec.org/RePEc:eee:jbfina:v:34:y:2010:i:11:p:2767-2787
17. Kim, B., Koyejo, O., Khanna, R.: Examples are not enough, learn to criticize! criticism for interpretability. In: Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain. pp. 2280–2288 (2016), http://papers.nips.cc/paper/6300-examples-are-not-enough-learn-to-criticize-criticism-for-interpretability
18. Koh, P.W., Liang, P.: Understanding black-box predictions via influence functions. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. pp. 1885–1894 (2017), http://proceedings.mlr.press/v70/koh17a.html
19. Looveren, A.V., Klaise, J.: Interpretable counterfactual explanations guided by prototypes. CoRR **abs/1907.02584** (2019), http://arxiv.org/abs/1907.02584
20. Luss, R., Chen, P., Dhurandhar, A., Sattigeri, P., Shanmugam, K., Tu, C.: Generating contrastive explanations with monotonic attribute functions. CoRR **abs/1905.12698** (2019), http://arxiv.org/abs/1905.12698
21. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: A survey on bias and fairness in machine learning. CoRR **abs/1908.09635** (2019), http://arxiv.org/abs/1908.09635
22. Molnar, C.: Interpretable Machine Learning (2019), https://christophm.github.io/interpretable-ml-book/
23. parliament, E., council: Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation). https://eur-lex.europa.eu/eli/reg/2016/679/oj (2016)
24. Poyiadzi, R., Sokol, K., Santos-Rodriguez, R., Bie, T.D., Flach, P.A.: FACE: feasible and actionable counterfactual explanations. CoRR **abs/1909.09369** (2019), http://arxiv.org/abs/1909.09369
25. Ribeiro, M.T., Singh, S., Guestrin, C.: Model-agnostic interpretability of machine learning. In: ICML Workshop on Human Interpretability in Machine Learning (WHI) (2016)
26. Ribeiro, M.T., Singh, S., Guestrin, C.: "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. pp. 1135–1144. KDD '16, ACM, New York, NY, USA (2016). https://doi.org/10.1145/2939672.2939778, http://doi.acm.org/10.1145/2939672.2939778

27. S. Aeberhard, D.C., de Vel, O.: Comparison of classifiers in high dimensional settings. Tech. Rep. no. 92-02 (1992)
28. Samek, W., Wiegand, T., Müller, K.: Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. CoRR **abs/1708.08296** (2017), http://arxiv.org/abs/1708.08296
29. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R.: Intriguing properties of neural networks. In: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings (2014)
30. Tjoa, E., Guan, C.: A survey on explainable artificial intelligence (XAI): towards medical XAI. CoRR **abs/1907.07374** (2019), http://arxiv.org/abs/1907.07374
31. Wachter, S., Mittelstadt, B.D., Russell, C.: Counterfactual explanations without opening the black box: Automated decisions and the GDPR. CoRR **abs/1711.00399** (2017), http://arxiv.org/abs/1711.00399
32. Waddell, K.: How algorithms can bring down minorities' credit scores. The Atlantic (2016)
33. William H. Wolberg, W. Nick Street, O.L.M.: Breast cancer wisconsin (diagnostic) data set. https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic) (1995)
34. Zhao, Q., Hastie, T.: Causal interpretations of black-box models. Journal of Business & Economic Statistics **0**(ja), 1–19 (2019). https://doi.org/10.1080/07350015.2019.1624293, https://doi.org/10.1080/07350015.2019.1624293