DACE: Distribution-Aware Counterfactual Explanation by Mixed-Integer Linear Optimization

Kentaro Kanamori[1] , **Takuya Takagi**[2] , **Ken Kobayashi**[2,3] and **Hiroki Arimura**[1]
[1]Hokkaido University
[2]Fujitsu Laboratories Ltd.
[3]Tokyo Institute of Technology
{kanamori, arim}@ist.hokudai.ac.jp, {takagi.takuya, ken-kobayashi}@fujitsu.com

Abstract

Counterfactual Explanation (CE) is one of the post-hoc explanation methods that provides a perturbation vector so as to alter the prediction result obtained from a classifier. Users can directly interpret the perturbation as an "*action*" for obtaining their desired decision results. However, an action extracted by existing methods often becomes unrealistic for users because they do not adequately care about the characteristics corresponding to the empirical data distribution such as feature-correlations and outlier risk. To suggest an executable action for users, we propose a new framework of CE for extracting an action by evaluating its reality on the empirical data distribution. The key idea of our proposed method is to define a new cost function based on the Mahalanobis' distance and the local outlier factor. Then, we propose a mixed-integer linear optimization approach to extracting an optimal action by minimizing our cost function. By experiments on real datasets, we confirm the effectiveness of our method in comparison with existing methods for CE.

1 Introduction

1.1 Background and Motivation

In recent years, complex machine learning models, such as deep neural networks and tree ensemble models, have achieved high prediction accuracy and been widely used for assisting the decision-making tasks in the real world, such as medical diagnosis and loan approval. As a result, to understand not only why an undesired prediction is obtained, but also how to act to obtain a desirable outcome, post-hoc methods for extracting explanations from the individual prediction of complex models have increasingly been attracting attention [Guidotti *et al.*, 2018; Molnar, 2019]. One of the post-hoc explanation approaches is the *Counterfactual Explanation (CE)* [Wachter *et al.*, 2018]. For a given classifier $H : \mathcal{X} \to \mathcal{Y}$, target class $t \in \mathcal{Y}$, and instance $\bar{x} \in \mathcal{X}$ such that $H(\bar{x}) \neq t$, the aim of CE is to find an optimal solution $a^\star$ of the following optimization problem:

$$a^\star := \arg\min_{a \in \mathcal{A}} \quad C(a \mid \bar{x}) \quad \text{subject to} \quad H(\bar{x} + a) = t,$$

where $\mathcal{A}$ is a set of *actions*, and $C : \mathcal{A} \to \mathbb{R}_{\geq 0}$ is a cost function that measures the required efforts of an action $a \in \mathcal{A}$. This problem is related to the *adversarial examples* [Szegedy *et al.*, 2014] in the sense that it finds a perturbation a that alters the output of a classifier H. In the context of CE, on the other hand, an action a is interpreted as a required action for a user $\bar{x}$ to obtain the desired prediction result (e.g., low risk of default). Therefore, the action suggested by CE should be executable for users. In this paper, we focus on this characteristic property of CE, and discuss how to provide a realistic action so that users can directly refer to and execute.

To extract realistic actions, we need to define a cost function C that considers the empirical distribution. While several useful cost functions, such as the total log-percentile shift (TLPS) [Ustun *et al.*, 2019], have been proposed, we argue that they often extract unrealistic actions. Figure 1 presents two demonstrations on the FICO dataset [FICO *et al.*, 2018], which is a real dataset of Home Equity Line of Credit (HELOC) applications. The task is to predict whether individuals will default on their HELOC. Figure 1 shows actions a (yellow arrows) extracted by using the TLPS from random forest classifiers trained on the dataset, and these modified instances $\bar{x} + a$ (yellow triangles). Table 1(a) shows the actual values of them. From Figure 1, we can observe that these modified instances $\bar{x} + a$ are located in the region predicted as "low risk of default", however, we argue that these actions are not realistic for users from the following two perspectives corresponding to the characteristics of the empirical distribution:

- **Feature-correlation:** Because each feature is often dependent on others, having non-zero correlation, the cost of changing a value with respect to a feature should be evaluated depending both on the amount of its difference and relation to other features. In Figure 1 (left), it seems unnatural to increase only "MSinceOldestTradeOpen" without increasing "AverageMInFile" because these features are correlated.
- **Outlier risk:** By minimizing the cost of a, there is a risk that its modified instance $\bar{x} + a$ becomes an outlier of the empirical distribution. In Figure 1 (right), it seems unrealistic to increase "ExternalRiskEstimate" without decreasing "PercentInstallTrades", because there are no training instances near $\bar{x} + a$.

Based on the above observations, our goals are (i) to model

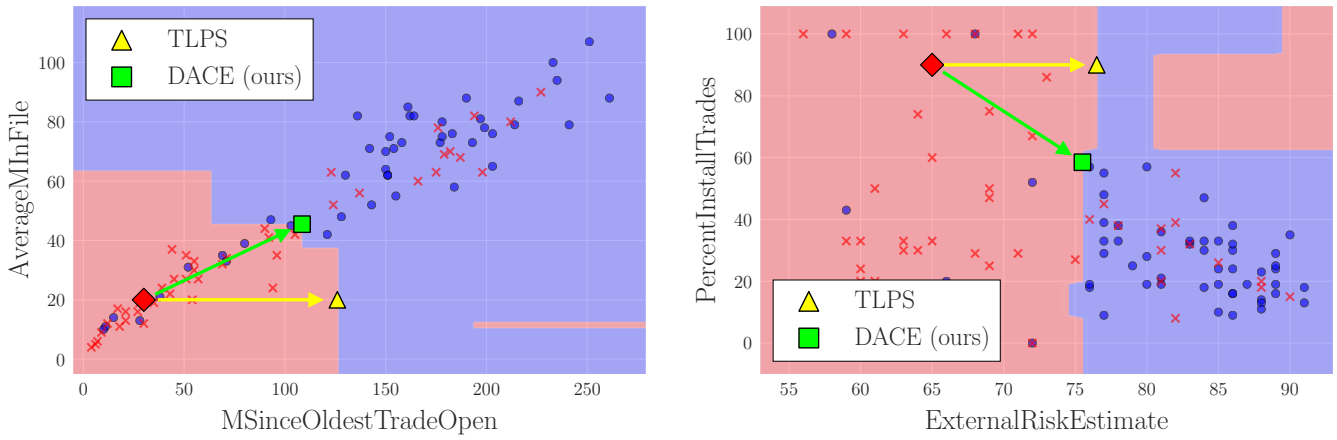

Figure 1: 2-dimensional illustrations of extracted actions on the FICO dataset. The arrows represent actions for the input instances (red diamonds) extracted by TLPS (yellow) and our DACE (green).

Feature to Change	Action	MD	LOF
"MSinceOldestTradeOpen"	30 → 126 (+96)	**4.39**	1.93
"ExternalRiskEstimate"	65→ 77 (+12)	1.35	**7.92**

(a) TLPS [Ustun *et al.*, 2019]

Feature to Change	Action	MD	LOF
"MSinceOldestTradeOpen" "AverageMInFile"	30 → 109 (+79) 20 → 46 (+26)	**1.11**	1.17
"ExternalRiskEstimate" "PercentInstallTrades"	65 → 76 (+11) 90 → 58 (-32)	1.40	**1.04**

(b) DACE (ours)

Table 1: Examples of CE extracted on the FICO dataset. These actions provide how to change feature values so as to be predicted as "low risk of default" from the classifier learned on the dataset.

the reality of an action as a cost function C, and (ii) to propose a method to optimize C for extracting realistic actions.

1.2 Our Contributions

In this paper, we propose a new framework of CE, named *Distribution-Aware Counterfactual Explanation (DACE)*, that extracts a realistic action for users. Our contributions can be summarized as follows:

- We propose a new cost function based on the *Mahalanobis' distance (MD)* [Mahalanobis, 1936] and *Local Outlier Factor (LOF)* [Breunig *et al.*, 2000] to evaluate the reality of actions. MD is known as a metric that captures the relationships between features, and LOF is a popular outlier score that measures how unusual a given instance is by using k-nearest neighbor (k-NN).
- We formulate the problem of finding an optimal action according to our cost function as a *mixed-integer linear optimization (MILO)* problem, which can be solved by modern MILO solvers, such as CPLEX [IBM, 2018]. For computational efficiency, we show that if we use ℓ_1-norm based MD and 1-NN based LOF for the cost function, the number of variables and constraints of the problem can be reduced dramatically.
- We demonstrate the effectiveness of DACE compared to other existing methods including MAD [Wachter *et al.*, 2018; Russell, 2019], TLPS [Ustun *et al.*, 2019], and PCC [Ballet *et al.*, 2019] on real datasets.

Table 1(b) and the green arrows in Figure 1 show the actions extracted by our DACE. These results suggest that the MD and LOF of actions reflect the feature-correlations and outlier risks well, respectively, and that DACE can extract realistic actions in the sense of these properties.

1.3 Related Work

Counterfactual Explanation. A number of post-hoc methods for generating explanations from complex models have been proposed [Ribeiro *et al.*, 2016; Lundberg and Lee, 2017; Koh and Liang, 2017; Ribeiro *et al.*, 2018]. *Counterfactual Explanation (CE)* is one of the post-hoc methods that has been attracting attention in recent years. Most of existing CE methods are either gradient-based [Wachter *et al.*, 2018; Dhurandhar *et al.*, 2018; Moore *et al.*, 2019] or heuristic search [Lash *et al.*, 2017]. These methods can deal with differentiable models over continuous features. On the other hand, there have been increasing demands for learning non-differentiable models over possibly non-continuous features such as tree ensembles over categorical data, to which existing gradient-based methods are not applicable. To overcome these difficulties, several authors proposed *integer linear optimization (ILO)* approaches [Cui *et al.*, 2015; Ustun *et al.*, 2019; Russell, 2019], using linear costs. Our results extend their approach to a cost function containing non-linear terms such as MD and LOF.

Distribution-aware score for CE. Recently, some studies have pointed out the risk that existing post-hoc methods often suffer from a lack of robustness [Ghorbani *et al.*, 2019; Rudin, 2019]. For this problem, the following scores for CE

were proposed. [Laugel *et al.*, 2019b] has introduced the notion of *connectedness* of an action to exclude a meaningless action that transfers a given data to an empty decision region containing no training data. They also pointed out that most of existing CE methods can generate non-connected actions. Note however that their connectedness and our criterion are incomparable. Actually, in Figure 1, all the actions are connected, while some of them are classified as bad according to our criterion. Another distribution-aware criterion, called *proximity* [Laugel *et al.*, 2019a] for CE, is related to the LOF. To the best of our knowledge, our method is a first attempt to optimize it, while the above work proposed criteria only.

2 Preliminaries

2.1 Notations and Settings

For a positive integer $n \in \mathbb{N}$, we denote by $[n] := \{1, \dots, n\}$. For a proposition ψ, $\mathbb{I}[\psi]$ denotes the indicator of ψ, i.e., $\mathbb{I}[\psi] = 1$ if ψ is true, and $\mathbb{I}[\psi] = 0$ if ψ is false.

Throughout this research, we consider a *binary classification problem* as a prediction task, which is sufficient for CE. For a multi-class classification problem, we can reduce it to a binary classification problem between the target class and other classes. We denote input and output domains by $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_D \subseteq \mathbb{R}^D$ and $\mathcal{Y} = \{-1, +1\}$, respectively. Let a vector $x = (x_1, \dots, x_D) \in \mathcal{X}$ be an *instance*, and a function $H : \mathcal{X} \to \mathcal{Y}$ be a *classifier*.

We assume that categorical features included by a dataset are one-hot encoded by pre-processing. Let $G \subseteq [D]$ be a set of features that represents a one-hot encoded categorical feature with $|G|$ categories. Then, $\mathcal{X}_g = \{0, 1\}$ for $g \in G$ and $\sum_{g \in G} x_g = 1$ for any $x \in \mathcal{X}$. We denote a set of one-hot encoded categorical features G by $\mathcal{G} \subseteq 2^{[D]}$.

2.2 Mahalanobis' Distance (MD)

The *Mahalanobis' distance (MD)* is a popular metric in the literature on statistics and metric learning [Mahalanobis, 1936; Kulis, 2013]. For two vectors $x, x' \in \mathbb{R}^D$ and a positive semi-definite matrix $M \in \mathbb{R}^{D \times D}$, Mahalanobis' distance $d_{\mathrm{M}} : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}_{\geq 0}$ between x and x' is defined by

$$d_{\mathrm{M}}(x, x' \mid M) := \sqrt{(x' - x)^\top M (x' - x)}.$$

Since M is positive semi-definite, M can be decomposed as $M = U^\top U$, where $U \in \mathrm{R}^{D \times D}$. Hence, $d_{\mathrm{M}}(x, x' \mid M)$ can be also expressed as follows:

$$d_{\mathrm{M}}(x, x' \mid M) = \|U(x' - x)\|_2,$$

where $\| \cdot \|_p$ denotes the ℓ_p-norm. In statistics, the inverse matrix of the covariance matrix Σ of the distribution where x and x' follow is often used as M. It is known that the MD $d(x, x' \mid \Sigma^{-1})$ is scale-invariant and takes the feature correlations into account [Maesschalck *et al.*, 2000].

2.3 Local Outlier Factor (LOF)

The *local outlier factor (LOF)* is a prominent outlier score based on the local densities of instances [Breunig *et al.*, 2000]. We assume a metric space $(\mathcal{X}, \Delta)$ and a set of N instances $X \subseteq \mathcal{X}$. We omit them if it is clear from context. For $x \in \mathcal{X}$, let $N_k(x)$ be its *k-nearest neighbors* (k-NN) on X. The *k-reachability distance* rd_k of x with respect to x' is defined by $rd_k(x, x') := \max\{\Delta(x, x'), d_k(x')\}$, where $d_k(x')$ is the distance Δ between x' and its the k-th nearest instance on X. The *k-local reachability density* of x is defined by $lrd_k(x) := |N_k(x)| \cdot (\sum_{x' \in N_k(x)} rd_k(x, x'))^{-1}$. Then, the k-LOF of x on X is defined as follows:

$$q_k(x \mid X) := \frac{1}{|N_k(x)|} \sum_{x' \in N_k(x)} \frac{lrd_k(x')}{lrd_k(x)}.$$

As a metric $\Delta : \mathcal{X} \times \mathcal{X} \to \mathrm{R}_{\geq 0}$, we assume $\Delta(x, x') = \sum_{d=1}^{D} \Delta_d(x_d, x'_d)$, where $\Delta_d : \mathcal{X}_d \times \mathcal{X}_d \to \mathrm{R}_{\geq 0}$ is some dissimilarity measure of the feature $d \in [D]$.

2.4 Additive Classifiers

In this paper, we focus on *additive classifiers* $H : \mathcal{X} \to \mathcal{Y}$ expressed as the following additive form:

$$H(x) = \mathrm{sgn}\left(\sum_{t=1}^{T} w_t \cdot h_t(x) - b \right),$$

where $h_1, \dots, h_T : \mathcal{X} \to \mathbb{R}$ are base learners, $w_t \in \mathbb{R}$ is a weight value of h_t for $t \in [T]$, and $b \in \mathbb{R}$ is an intercept.

Linear Model. *Linear models (LM)*, such as Logistic Regression and Linear Support Vector Machines, are one of the most standard classifiers [Hastie *et al.*, 2009]. If H is a LM, $T = D$ and each base learner $h_d(x) = x_d$. A LM makes a prediction depending on the sign of the inter product $\langle w, x \rangle$, where $w = (w_1, \dots, w_D) \in \mathbb{R}^D$.

Tree Ensemble Model. *Tree ensemble models (TEM)*, such as Random Forest [Breiman, 2001] and Gradient Boosted Trees [Friedman, 2000; Chen and Guestrin, 2016; Ke *et al.*, 2017], are renowned for their high prediction performances in machine learning competitions. If H is a TEM, each base learner h_t is a *decision tree*, which is a classifier that consists of a set of if-then-else rules expressed by a binary tree structure. It makes the prediction according to a leaf that the input instance $x \in \mathcal{X}$ reaches, and the corresponding leaf is determined by traversing the tree from the root depending on whether the statement $x_d \leq t$ is true or not, where $d \in [D]$ and $t \in \mathbb{R}$ are a pair of parameters corresponding to the internal node. A TEM makes a prediction by combining the prediction results from T decision trees.

3 Problem Statement

3.1 Action and Action Set

For a classifier $H : \mathcal{X} \to \mathcal{Y}$, and an instance $\bar{x} \in \mathcal{X}$ such that $H(\bar{x}) = -1$, we define an *action* as a perturbation vector $a \in \mathbb{R}^D$ such that $H(\bar{x} + a) = +1$. An *action set* $\mathcal{A} = A_1 \times \cdots \times A_D$ is a finite set of feasible actions such that $0 \in A_d$ and $A_d \subseteq \{a \in \mathbb{R} \mid \bar{x}_d + a_d \in \mathcal{X}_d\}$ for $d \in [D]$. We denote by $I_d = |A_d|$ for $d \in [D]$.

We can automatically determine each A_d depending on the type of the classifier H [Ustun *et al.*, 2019; Cui *et al.*, 2015] and the feature $d \in [D]$. For example, if x_d is a feature representing "age", then $a_d \in \mathbb{N} \cup \{0\}$ holds for any $a_d \in A_d$. If x_d is an immutable feature (e.g., gender) then $A_d = \{0\}$.

3.2 Cost Function

As a score to evaluate whether an action is realistic for users, we introduce a new cost function $C_{\mathrm{DACE}} : \mathcal{A} \to \mathbb{R}_{\geq 0}$. Given a positive semi definite matrix $M \in \mathbb{R}^{D \times D}$, set of N instances $X \subseteq \mathcal{X}$, positive integer $k \in [N]$, and $\lambda \geq 0$, we define C_{DACE} with respect to an input instance $\bar{x} \in \mathcal{X}$ as

$$C_{\mathrm{DACE}}(a \mid \bar{x}) := d^2_{\mathrm{M}}(\bar{x}, \bar{x} + a \mid M) + \lambda \cdot q_k(\bar{x} + a \mid X),$$

where

- $d^2_{\mathrm{M}}(\bar{x}, \bar{x} + a \mid M)$ is the squared MD between the input instance $\bar{x}$ and its modified instance $\bar{x} + a$,
- $q_k(\bar{x} + a \mid X)$ is the k-LOF of $\bar{x} + a$ on X, and
- $\lambda \geq 0$ is a trade-off parameter between d^2_{M} and q_k.

3.3 Problem Definition

Our aim is to find an action $a \in \mathcal{A}$ that minimizes the cost $C_{\mathrm{DACE}}(a \mid \bar{x})$, and this problem can be defined as follows:

Problem 1. *Given an additive classifier $H : \mathcal{X} \to \mathcal{Y}$, input instance $\bar{x} \in \mathcal{X}$ such that $H(\bar{x}) = -1$, positive semi definite matrix $M \in \mathbb{R}^{D \times D}$, set of N instances $X \subseteq \mathcal{X}$, positive integer $k \in [N]$, and $\lambda \geq 0$, find an action $a^* \in \mathcal{A}$ such that is an optimal solution for the following optimization problem:*

$$\underset{a \in \mathcal{A}}{\text{minimize}} \quad C_{\mathrm{DACE}}(a \mid \bar{x}) \quad \text{subject to} \quad H(\bar{x} + a) = t,$$

4 MILO Formulation

In this section, we propose an MILO formulation for solving Problem 1.

4.1 Basic Ideas

While C_{DACE} is non-linear and has a discrete structure caused by $N_k(\bar{x} + a)$, we can exactly formulate Problem 1 as an MILO problem using modeling techniques on integer optimization. However, to linearize the non-linearity and discreteness, this naive formulation requires $\mathcal{O}(|\mathcal{A}|^2 + N^2)$ auxiliary variables and constraints, where $|\mathcal{A}| := \sum_{d=1}^{D} |A_d|$. Preliminary experiments made clear that this formulation was computationally infeasible on standard datasets.

In order to avoid introducing $\mathcal{O}(|\mathcal{A}|^2 + N^2)$ auxiliary variables and constraints, we introduce a surrogate objective function and optimize it instead of C_{DACE}. Our main ideas are (i) fixing $k = 1$ for the LOF $q_k(\bar{x} + a)$ of C_{DACE}, and (ii) replacing the MD d^2_{M} of C_{DACE} with *ℓ_1-norm based Mahalanobis' distance (ℓ_1-MD)* $\hat{d}_{\mathrm{M}}$ defined as

$$\hat{d}_{\mathrm{M}}(x, x' \mid M) := \|U(x' - x)\|_1,$$

which is based on the fact that d^2_{M} can be expressed as $d_{\mathrm{M}}(x, x' \mid M) := \|U(x' - x)\|_2$. Overall, we formulate the following problem as an MILO problem instead of Problem 1:

$$\begin{aligned} \underset{a \in \mathcal{A}}{\text{minimize}} \quad & \hat{d}_{\mathrm{M}}(\bar{x}, \bar{x} + a \mid M) + \lambda \cdot q_1(\bar{x} + a \mid X) \\ \text{subject to} \quad & H(\bar{x} + a) = +1. \end{aligned}$$

First, we introduce binary variables $\pi_{d,i} \in \{0, 1\}$ for $d \in [D]$ and $i \in [I_d]$, which indicate that the action $a_{d,i} \in A_d$ is selected ($\pi_{d,i} = 1$) or not ($\pi_{d,i} = 0$). Then, $\pi_{d,i}$ must satisfy the following constraints:

$$\sum_{i=1}^{I_d} \pi_{d,i} = 1, \forall d \in [D], \tag{1}$$

$$\sum_{d \in G} (\bar{x}_d + \sum_{i=1}^{I_d} a_{d,i} \pi_{d,i}) = 1, \forall G \in \mathcal{G}. \tag{2}$$

Each element of the action $a = (a_1, \ldots, a_D) \in \mathcal{A}$ can be expressed as $a_d = \sum_{i=1}^{I_d} a_{d,i} \pi_{d,i}$. Constraint (2) is corresponding to one-hot encoded categorical features $\mathcal{G}$. In the rest of this section, we formulate the constraint and objective function by using linear constraints of $\pi_{d,i}$.

4.2 Base Learner Constraints

Because the output value of each base learner h_t for $\bar{x} + a$ varies depending on the value of a, i.e., the program variables $\pi_{d,i}$, we must express the value of $h_t(\bar{x} + a)$ by linear constraints of $\pi_{d,i}$. We introduce variables $\xi_t \in \mathbb{R}$ such that $\xi_t = h_t(\bar{x} + a)$ for $t \in [T]$. From the definition of additive classifiers, the constraint $H(\bar{x} + a) = +1$ is equivalent to the following linear constraint of ξ_t:

$$\sum_{t=1}^{T} w_t \xi_t \geq b. \tag{3}$$

In the following, we show how to express ξ_t when H is a linear model (LM) or tree ensemble model (TEM).

Linear Models. From the definition of LM, $T = D$ and $h_d(\bar{x} + a) = \bar{x}_d + a_d$ holds for $d \in [D]$. Hence, we can simply express the base learner of the LM as follows:

$$\xi_d = \bar{x}_d + \sum_{i=1}^{I_d} a_{d,i} \pi_{d,i}, \forall d \in [D]. \tag{4}$$

Tree Ensemble Models. Each base learner h_t of the TEM is a decision tree. It is known that a decision tree $h_t : \mathcal{X} \to \mathcal{Y}$ with L_t leaves represents a partition $\{r_{t,1}, \ldots, r_{t,L_t}\}$ of the input domain $\mathcal{X}$ [Hastie *et al.*, 2009], and can be expressed as $h_t(x) = \sum_{l=1}^{L} \hat{y}_{t,l} \cdot \mathbb{I}[x \in r_{t,l}]$, where $\hat{y}_{m,l} \in \mathcal{Y}$ is a predictive label corresponding to the leaf $l \in [L_t]$ of h_t. In order to express the statement $\bar{x} + a \in r_{t,l}$, we can utilize the *decision logic constraint* proposed by [Cui *et al.*, 2015] expressed as

$$\phi_{t,l} \in \{0, 1\}, \forall t \in [T], l \in [L_t], \tag{5}$$

$$\sum_{l=1}^{L_t} \phi_{t,l} = 1, \forall t \in [T], \tag{6}$$

$$D \cdot \phi_{t,l} \leq \sum_{d=1}^{D} \sum_{i \in I^{(d)}_{t,l}} \pi_{d,i}, \forall t \in [T], l \in [L_t], \tag{7}$$

where $I^{(d)}_{t,l} = \{i \in [I_d] \mid \bar{x}_d + a_{d,i} \in r^{(d)}_{t,l}\}$ and $r^{(d)}_{t,l}$ is the subspace of $\mathcal{X}_d$ such that $r_{t,l} = r^{(1)}_{t,l} \times \cdots \times r^{(D)}_{t,l}$. $\phi_{t,l}$ is an

indicator variable such that $\phi_{t,l} = \mathbb{I}[\bar{x} + a \in r_{t,l}]$. Then, we can express the base learner of the TEM as follows:

$$\xi_t = \sum_{l=1}^{L_t} \hat{y}_{t,l} \cdot \phi_{t,l}, \forall t \in [T]. \tag{8}$$

Because $I_{t,l}^{(d)}$, L_t, and $\hat{y}_{t,l}$ can be computed when H and $\mathcal{A}$ are given, they are constant values.

4.3 Surrogate Objective Function

ℓ_1-norm based Mahalanobis' Distance (ℓ_1-MD). The ℓ_1-MD between $\bar{x}$ and $\bar{x} + a$ can be expressed as follows:

$$\hat{d}_{\mathrm{M}}(\bar{x}, \bar{x} + a \mid M) = \|Ua\|_1 = \sum_{d=1}^{D} |\langle U_d, a \rangle|,$$

where $U_d = (U_{d,1}, \ldots, U_{d,D})$ is the d-th row vector of U. We introduce variables $\delta_d \geq 0$ for $d \in [D]$ such that $\delta_d = |\langle U_d, a \rangle|$. Then, $\hat{d}_{\mathrm{M}}(\bar{x}, \bar{x} + a \mid M)$ can be expressed as

$$\hat{d}_{\mathrm{M}}(\bar{x}, \bar{x} + a \mid M) = \sum_{d=1}^{D} \delta_d,$$

with the following constraints:

$$-\delta_d \leq \sum_{d'=1}^{D} U_{d,d'} \sum_{i=1}^{I_d} a_{d',i} \pi_{d',i} \leq \delta_d, \forall d \in [D]. \tag{9}$$

$\hat{d}_{\mathrm{M}}$ is not exactly the same as d_{M}^2 however, we show the approximation ratio of $\hat{d}_{\mathrm{M}}$ with respect to d_{M}^2 as follows:

Proposition 1. *Let $a^*, \hat{a} \in \mathbb{R}^D$ be two vectors such that $a^* = \arg\min_{a \in \mathbb{R}^D} d_{\mathrm{M}}^2(\bar{x}, \bar{x} + a \mid M)$ and $\hat{a} = \arg\min_{a \in \mathbb{R}^D} \hat{d}_{\mathrm{M}}(\bar{x}, \bar{x} + a \mid M)$, respectively. Then $d_{\mathrm{M}}(\bar{x}, \bar{x} + \hat{a} \mid M) \leq \sqrt{D} \cdot d_{\mathrm{M}}(\bar{x}, \bar{x} + a^* \mid M)$ holds.*

Proof. Let $U \in \mathbb{R}^{D \times D}$ be a matrix such that $M = U^\top U$. By the definitions, $d_{\mathrm{M}}(\bar{x}, \bar{x} + a \mid M)$ and $\hat{d}_{\mathrm{M}}(\bar{x}, \bar{x} + a \mid M)$ are expressed as $d_{\mathrm{M}}(\bar{x}, \bar{x} + a \mid M) = \|Ua\|_2$ and $\hat{d}_{\mathrm{M}}(\bar{x}, \bar{x} + a \mid M) = \|Ua\|_1$, respectively. From the properties of L_p-norm, it holds that $\|U\hat{a}\|_2 \leq \|U\hat{a}\|_1$ and $\|Ua^*\|_1 \leq \sqrt{D} \cdot \|Ua^*\|_2$. Recall the definitions of a^* and $\hat{a}$, $\|U\hat{a}\|_1 \leq \|Ua^*\|_1$ holds. By combining these inequalities, we have $\|U\hat{a}\|_2 \leq \sqrt{D} \cdot \|Ua^*\|_2$, which is equivalent to $d_{\mathrm{M}}(\bar{x}, \bar{x} + \hat{a} \mid M) \leq \sqrt{D} \cdot d_{\mathrm{M}}(\bar{x}, \bar{x} + a^* \mid M)$. □

1-Local Outlier Factor (1-LOF). From the definitions of q_k and lrd_k, $q_k(\bar{x} + a \mid X)$ for $k = 1$ can be expressed as

$$q_1(\bar{x} + a \mid X) = lrd_1(x^{(m)}) \cdot rd_1(\bar{x} + a, x^{(m)}),$$

where $m = \arg\min_{n \in [N]} \Delta(\bar{x} + a, x^{(n)})$, i.e., $N_1(\bar{x} + a) = \{x^{(m)}\}$. Because m and $rd_1(\bar{x} + a, x^{(m)})$ varies depending on $\bar{x} + a$, i.e., the variables $\pi_{d,i}$, we need to formulate it by linear constraints of $\pi_{d,i}$. We introduce variables $\nu_n \in \{0, 1\}$ and $\rho_n \geq 0$ for $n \in [N]$ such that $\nu_n = \mathbb{I}\left[x^{(n)} \in N_1(\bar{x} + a)\right]$ and $\rho_n = rd_1(\bar{x} + a, x^{(n)}) \cdot \nu_n$, respectively. Then, $q_1(\bar{x} + a)$ can be expressed as a linear form of ρ_n as follows:

$$q_1(\bar{x} + a \mid X) = \sum_{n=1}^{N} l^{(n)} \cdot \rho_n,$$

with the following constraints:

$$\sum_{n=1}^{N} \nu_n = 1, \tag{10}$$

$$\sum_{d=1}^{D} \sum_{i=1}^{I_d} (c_{d,i}^{(n)} - c_{d,i}^{(n')}) \pi_{d,i} \leq C_n(1 - \nu_n), \forall n, n' \in [N], \tag{11}$$

$$\rho_n \geq d^{(n)} \cdot \nu_n, \forall n \in [N], \tag{12}$$

$$\rho_n \geq \sum_{d=1}^{D} \sum_{i=1}^{I_d} c_{d,i}^{(n)} \pi_{d,i} - C_n(1 - \nu_n), \forall n \in [N], \tag{13}$$

where $c_{d,i}^{(n)}$, C_n, d_n, and $l^{(n)}$ are constant values such that $c_{d,i}^{(n)} = \Delta_d(\bar{x}_d + a_{d,i}, x_d^{(n)})$, $C_n \geq \max_{a \in \mathcal{A}} \Delta(\bar{x} + a, x^{(n)})$, $d^{(n)} = d_1(x^{(n)})$, and $l^{(n)} = lrd_1(x^{(n)})$. Constraints (10) and (11) are based on the statement $\nu_m = 1 \Rightarrow \forall n \in [N] : \Delta(\bar{x} + a, x^{(m)}) \leq \Delta(\bar{x} + a, x^{(n)})$, which is for expressing the nearest instance of $\bar{x} + a$. Note that $\Delta(\bar{x} + a, x^{(n)}) = \sum_{d=1}^{D} \sum_{i=1}^{I_d} c_{d,i}^{(n)} \pi_{d,i}$, and Constraints (12) and (13) are based on the definition of k-reachability distance rd_k. All constant values can be computed when X and $\mathcal{A}$ are given.

	#Vars	#Consts
d_{M}^2	$\mathcal{O}(\lvert\mathcal{A}\rvert^2)$	$\mathcal{O}(\lvert\mathcal{A}\rvert^2)$
N_k	$\mathcal{O}(N^2)$	$\mathcal{O}(N^2)$
q_k	$\mathcal{O}(N^2)$	$\mathcal{O}(N^2)$

	#Vars	#Const
$\hat{d}_{\mathrm{M}}$	$\mathcal{O}(D)$	$\mathcal{O}(D)$
N_1	$\mathcal{O}(N)$	$\mathcal{O}(N^2)$
q_1	$\mathcal{O}(N)$	$\mathcal{O}(N)$

Table 2: The numbers of variables (#Vars) and constraints (#Consts) required for the exact formulation of Problem 1 (left) and our formulation (14) (right). Note that the problem (14) optimizes ℓ_1-MD $\hat{d}_{\mathrm{M}}$ and 1-LOF q_1 instead of d_{M}^2 and q_k for $k > 1$.

4.4 Overall Formulation

Finally, we show our overall formulation as follows:

$$\begin{aligned}
\text{minimize} \quad & \sum_{d=1}^{D} \delta_d + \lambda \cdot \sum_{n=1}^{N} l^{(n)} \cdot \rho_n \\
\text{subject to} \quad & \text{Constraint } (1-3), \\
& \begin{cases} \text{Constraint } (4), & \text{if } H \text{ is a LM,} \\ \text{Constraint } (5-8), & \text{if } H \text{ is a TEM,} \end{cases} \\
& \text{Constraint } (9-13), \\
& \pi_{d,i} \in \{0, 1\}, \forall d \in [D], \forall i \in [I_d], \\
& \delta_d \geq 0, \forall d \in [D], \\
& \nu_n \in \{0, 1\}, \rho_n \geq 0, \forall n \in [N].
\end{aligned} \tag{14}$$

	Logistic Regression			Random Forest		
	$d_{\mathrm{M}}(\bar{x}, \bar{x}+a \mid \Sigma^{-1})$	$q_{10}(\bar{x}+a \mid X_+)$	Time[s]	$d_{\mathrm{M}}(\bar{x}, \bar{x}+a \mid \Sigma^{-1})$	$q_{10}(\bar{x}+a \mid X_+)$	Time[s]
MAD	5.42 ± 4.04	1.65 ± 1.29	0.0261 ± 0.00439	2.29 ± 1.58	1.56 ± 1.14	34.4 ± 57.8
TLPS	9.09 ± 2.97	3.86 ± 1.49	0.0208 ± 0.00761	2.22 ± 1.31	1.49 ± 1.07	22.5 ± 36.6
PCC	9.46 ± 6.66	1.61 ± 1.31	0.0238 ± 0.0036	3.76 ± 2.36	1.6 ± 1.27	29.8 ± 78.7
DACE	$\mathbf{1.97 \pm 1.46}$	$\mathbf{1.54 \pm 1.12}$	67.9 ± 75.8	$\mathbf{1.54 \pm 1.18}$	$\mathbf{1.33 \pm 0.496}$	519 ± 171

(a) FICO dataset ($D = 23$)

	Logistic Regression			Random Forest		
	$d_{\mathrm{M}}(\bar{x}, \bar{x}+a \mid \Sigma^{-1})$	$q_{10}(\bar{x}+a \mid X_+)$	Time[s]	$d_{\mathrm{M}}(\bar{x}, \bar{x}+a \mid \Sigma^{-1})$	$q_{10}(\bar{x}+a \mid X_+)$	Time[s]
MAD	8.89 ± 2.73	1.94 ± 3.41	0.0488 ± 0.00215	7.5 ± 6.12	1.91 ± 2.19	33.6 ± 47.1
TLPS	3.73 ± 2.26	1.78 ± 0.718	0.0113 ± 0.00163	2.44 ± 3.91	1.34 ± 0.75	133 ± 193
PCC	8.14 ± 3.15	1.94 ± 3.41	0.042 ± 0.0022	6.73 ± 4.12	1.95 ± 2.18	11.9 ± 10.9
DACE	$\mathbf{2.27 \pm 1.51}$	$\mathbf{1.27 \pm 0.35}$	1.03 ± 0.276	$\mathbf{1.23 \pm 1.34}$	$\mathbf{1.13 \pm 0.27}$	240 ± 239

(b) german dataset ($D = 61$)

Table 3: Experimental results on the FICO and german datasets.

Table 2 presents the numbers of variables and constraints required for the naive formulation of Problem 1 and the problem (14). It shows that the latter reduces variables and constraints dramatically compared to the former. Therefore, our DACE solves the problem (14) to extract the desired action for computational efficiency.

As with the existing ILO-based methods [Ustun *et al.*, 2019; Russell, 2019], our formulation can be (i) efficiently solved by powerful off-the-shelf MILO solvers, such as CPLEX [IBM, 2018], (ii) customized by adding constraints that users desire, such as a limitation of features changed by actions, and (iii) applied to an algorithm for enumerating distinct actions as its subroutine. To summarize the above advantages, we can obtain actions that satisfy user-defined constraints without implementing designated algorithms.

5 Experiments

We conduct experiments on real datasets to investigate the effectiveness of our DACE by comparing the performance with existing methods for CE. All codes were implemented in Python 3.6 with scikit-learn and IBM ILOG CPLEX v12.8. All experiments were conducted on 64-bit Ubuntu 18.04.1 LTS with Intel Xeon E5-1620 v4 3.50GHz CPU and 62.8GiB memory, and we imposed a 600 second time limit for solving.

5.1 Experimental Setting

We used the FICO dataset ($D = 23$) [FICO *et al.*, 2018] and german dataset ($D = 61$) [Dua and Graff, 2017], where D is the number of features. For german dataset, each categorical feature was transformed into as many one-hot encoded features as its distinct values. The task of these datasets is to predict whether individuals will default on their loan. We randomly split each dataset into train (70%) and test (30%) instances, and trained ℓ_2-regularized logistic regression (LR) classifiers and random forest (RF) classifiers with $T = 100$ decision trees on each training dataset, respectively. Then, we extracted actions for the instances $\bar{x}$ of each test dataset who have been received bad prediction results, i.e., predicted as "high risk of default" from each classifier.

Baseline Methods. We compared our proposed method (**DACE**) to three existing methods. A main difference between DACE and the others is a cost function to be optimized. One cost function is the weighted ℓ_1-norm based on the inverse of median absolute deviation (**MAD**) [Wachter *et al.*, 2018; Russell, 2019]. Another cost function is the total log-percentile shift (**TLPS**) [Ustun *et al.*, 2019] that evaluates actions based on the cumulative distribution functions estimated from training instances. In addition to these cost functions, we also compared to the weighted ℓ_2-norm based on the Pearson's correlation coefficients (**PCC**) proposed by [Ballet *et al.*, 2019] to generate imperceptible adversarial examples.

Evaluation Scores. In order to compare the qualities of obtained actions a, we measured (i) the MD $d_{\mathrm{M}}(\bar{x}, \bar{x}+a \mid \Sigma^{-1})$, where Σ is the covariance matrix estimated from the training instances X, (ii) the 10-LOF $q_{10}(\bar{x}+a \mid X_+)$ on the training instances labeled as "low risk of default" $X_+ \subseteq X$, and (iii) running times for solving each MILO problem. The MD $d_{\mathrm{M}}(\bar{x}, \bar{x}+a \mid \Sigma^{-1})$ can measure the effort for $\bar{x}$ to execute an action a by taking the feature-correlations into account [Maesschalck *et al.*, 2000]. k-LOF $q_k(\bar{x}+a \mid X_+)$ represents the risk of that the action a leads $\bar{x}$ to be an outlier on the instances with the target label. We evaluate whether actions extracted by baselines and DACE have realities for users in terms of the above criteria.

5.2 Comparison with Existing Methods

We compared the actions extracted by DACE with ones by the baselines. We set $\lambda = 1.0$ for the FICO dataset and $\lambda = 0.01$ for the german dataset, respectively. These parameters are selected based on the sensitivity analyses described below. Table 3 presents the average MD, 10-LOF, and running time for each classifier and dataset, and shows that DACE achieved lower MD and 10-LOF than those of the baselines methods regardless of classifiers and datasets. These results suggest that DACE obtain more realistic actions than the other baselines do by considering the feature correlation and the risk of leading to an outlier. Regarding the average running time,

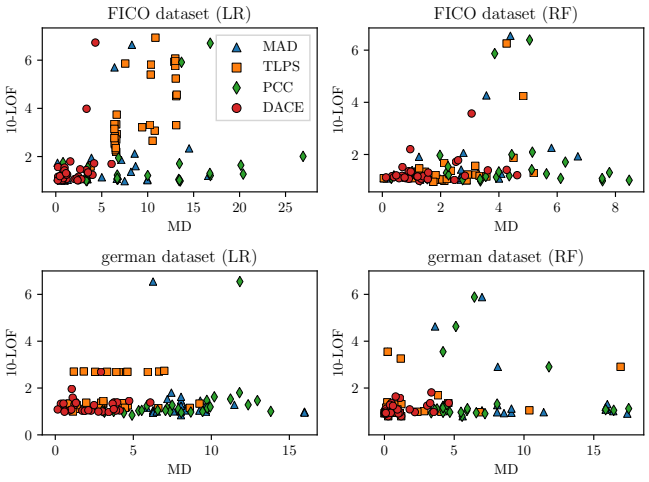

Figure 2: Scatter plots between MD and 10-LOF of the actions extracted by baseline methods and our DACE.

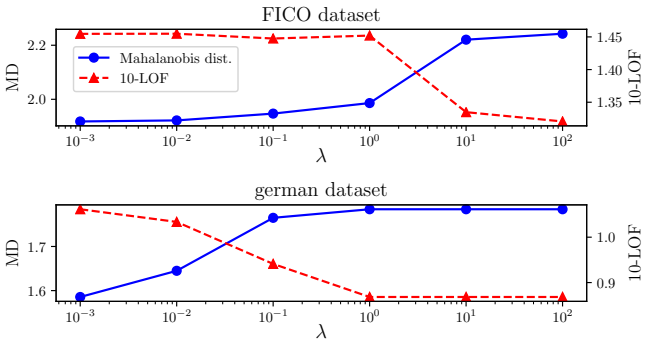

Figure 3: Sensitivity analyses of the trade-off parameter λ of DACE between the average MD and LOF.

DACE was certainly slower than the other baselines. However, Table 3 also indicates that DACE found actions of better quality in terms of both MD and 10-LOF within 600 seconds, which is a reasonable calculation time. Figure 2 presents scatter plots between MD and 10-LOF of each obtained action, and we can see that DACE stably achieved lower MD and 10-LOF than the other baselines did. From these results, the effectiveness of DACE has been confirmed in real datasets, and we also argue that our method is favorable when the quality of an action is required by decision-makers and their customers.

5.3 Sensitivity Analysis of Trade-off Parameter

Finally, we show the sensitivity of λ in C_{DACE} on LR classifiers. Figure 3 presents the average MD and 10-LOF of obtained actions a for each λ. We can see that there is a trade-off between MD and LOF of actions obtained by DACE with respect to the value of λ. Consequently, we need to choose λ depending on whether a user emphasizes the preference or reliability of an action. In other words, by varying the value of λ, we can obtain several distinct actions that have diverse characteristics in terms of MD and LOF. As mentioned in [Wachter *et al.*, 2018], suggesting multiple actions may help users for referring to as their future guidelines. Figure 3 indicates that by varying the value of λ, we can obtain several distinct actions that have diverse characteristics in terms of MD and LOF.

6 Conclusion

In this paper, we proposed a new framework of CE for extracting a realistic action by considering the empirical distribution on labeled examples. We introduced a new cost function based on the Mahalanobis' distance (MD) and the local outlier factor (LOF), and then proposed a MILO formulation for optimizing it. By experiments, we confirmed the effectiveness of our method by comparing with existing methods. For future work, there are some directions. First, we plan to devise a more efficient MILO formulation and extend our framework to deal with other classifiers, such as kernel SVMs and deep neural networks. Also, it is interesting to learn the matrix M for MD based on an empirical distribution and a given instance. To clarify when the use of DACE makes more sense, we also plan to conduct further detailed experiments where our assumptions do not hold strongly — e.g., correlations between features are low, etc.

Acknowledgements

We wish to thank Satoshi Hara for making a number of valuable suggestions. We also thank anonymous reviewers for their insightful comments. This work was supported in part by JSPS KAKENHI Grant-in-Aid for Scientific Research (A) 20H00595 and JST CREST JPMJCR18K3.

References

[Ballet *et al.*, 2019] Vincent Ballet, Xavier Renard, Jonathan Aigrain, Thibault Laugel, Pascal Frossard, and Marcin Detyniecki. Imperceptible adversarial attacks on tabular data. In *NeurIPS 2019 Workshop on Robust AI in Financial Services: Data, Fairness, Explainability, Trustworthiness and Privacy*, 2019.

[Breiman, 2001] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[Breunig *et al.*, 2000] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. LOF: Identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104, 2000.

[Chen and Guestrin, 2016] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 785–794, 2016.

[Cui *et al.*, 2015] Zhicheng Cui, Wenlin Chen, Yujie He, and Yixin Chen. Optimal action extraction for random forests and boosted trees. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 179–188, 2015.

[Dhurandhar *et al.*, 2018] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, and Payel Das. Explanations based on the missing: Towards contrastive explanations with pertinent

negatives. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 590–601, 2018.

[Dua and Graff, 2017] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[FICO *et al.*, 2018] FICO, Google, Imperial College London, MIT, University of Oxford, UC Irvine, and UC Berkeley. Explainable Machine Learning Challenge, 2018.

[Friedman, 2000] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

[Ghorbani *et al.*, 2019] Amirata Ghorbani, Abubakar Abid, and James Y. Zou. Interpretation of neural networks is fragile. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pages 3681–3688, 2019.

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

[Hastie *et al.*, 2009] Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition*. Springer Series in Statistics. Springer, 2009.

[IBM, 2018] IBM. CPLEX Optimizer — IBM. https://www.ibm.com/analytics/cplex-optimizer, 2018.

[Ke *et al.*, 2017] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 3149–3157, 2017.

[Koh and Liang, 2017] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, page 1885–1894, 2017.

[Kulis, 2013] Brian Kulis. Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4):287–364, 2013.

[Lash *et al.*, 2017] Michael T. Lash, Qihang Lin, W. Nick Street, Jennifer G. Robinson, and Jeffrey W. Ohlmann. Generalized inverse classification. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 162–170, 2017.

[Laugel *et al.*, 2019a] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. Issues with post-hoc counterfactual explanations: a discussion. In *2019 ICML Workshop on Human in the Loop Learning*, 2019.

[Laugel *et al.*, 2019b] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. The dangers of post-hoc interpretability: Unjustified counterfactual explanations. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 2801–2807, 2019.

[Lundberg and Lee, 2017] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4765–4774, 2017.

[Maesschalck *et al.*, 2000] R. De Maesschalck, D. Jouan-Rimbaud, and D.L. Massart. The mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, 50(1):1 – 18, 2000.

[Mahalanobis, 1936] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences*, 2:49–55, 1936.

[Molnar, 2019] Christoph Molnar. *Interpretable Machine Learning - A Guide for Making Black Box Models Explainable*. https://christophm.github.io/interpretable-ml-book/, 2019.

[Moore *et al.*, 2019] Jonathan Moore, Nils Hammerla, and Chris Watkins. Explaining deep learning models with constrained adversarial examples. In *Proceedings of the 16th Pacific Rim International Conference on Artificial Intelligence*, pages 43–56, 2019.

[Ribeiro *et al.*, 2016] Marco Túlio 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*, page 1135–1144, 2016.

[Ribeiro *et al.*, 2018] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 1527–1535, 2018.

[Rudin, 2019] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206–215, 2019.

[Russell, 2019] Chris Russell. Efficient search for diverse coherent explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, page 20–28, 2019.

[Szegedy *et al.*, 2014] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *Proceedings of the 2nd International Conference on Learning Representations*, 2014.

[Ustun *et al.*, 2019] Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, page 10–19, 2019.

[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*, 31:841–887, 2018.