Erasmus University Rotterdam
Erasmus School of Economics
Bachelor Thesis BS^2 Economics & Econometrics

UDACE: Utility & Distribution Aware Counterfactual Explanation by Mixed-Integer Linear Optimization

Naïm Achahboun (529911)

Supervisor:	Hakan Akyuz
Second assessor:	Paul Bouman
Date final version:	1st July 2023

The views stated in this thesis are those of the author and not necessarily those of the supervisor, second assessor, Erasmus School of Economics or Erasmus University Rotterdam.

Abstract

As machine learning models are becoming larger and more complex, interpretability is becoming increasingly crucial. The counterfactual explanation (CE) method is well-known for giving insight into the inner workings of a large class of models. A CE method generates an action such that the output of an automated decision model is changed. Extensive research has been conducted to ensure the realism of suggested actions, however, there has been a limited focus on assessing the desirability of an action from an individual's perspective. In light of this, we extend the Distribution-Aware Counterfactual Explanation (DACE) method by Kanamori, Takagi, Kobayashi and Arimura (2020) with individual utility. A multi-objective optimization approach is utilized, where we minimize the DACE cost function while maximizing utility. Results show that our new Utility & Distribution-Aware Counterfactual Explanation (UDACE) method is capable of generating multiple Pareto-efficient counterfactual explanations for the same individual. However, for some datasets, UDACE might be impractical due to long execution times.

Contents

1 Introduction

The field of artificial intelligence research is advancing rapidly. Prominent technology companies are trying to capitalize on the new research output by swiftly implementing AI-assisted products. However, these products have inherent biases that must be dealt with, in particular, when these products are autonomously making high-stake decisions. To combat the bias of automated decision-making systems, explainable artificial intelligence (XAI) is becoming crucial. To evaluate if the systems are acting desirably, we must first understand how they come to their conclusions.

Counterfactual explanation (CE) methods are frequently used in the domain of XAI. A CE method aims to change the inputs to an automated decision model, such that the decision of the model changes. CE methods are well suited to explaining which factors contributed most to a decision. Moreover, they are highly suitable for algorithmic recourse. Algorithmic recourse involves modifying the decisions made by automated decision-making models by providing applicants with recommended courses of action.

To find realistic and desirable courses of action, we extend the Distribution-Aware Counterfactual Explanation (DACE) method by Kanamori et al. (2020). Our extension includes the utility of stakeholders during optimization. By taking utility into account we have a framework which captures both effort and utility, and hence should increase the probability of successful recourse. The resulting method is named Utility & Distribution-Aware Counterfactual Explanation (UDACE). Where the DACE cost function is treated as a proxy for effort.

The key stakeholders during a recourse process are the institution that utilizes the machine learning model for automated decision-making, and the applicant who is affected by the decision. In this research, only the preferences of applicants are taken into account, since they will have to perform the actions.

The main research question of our research is: *How can counterfactual explanations take stakeholder preferences into account for personalized recourse?*

To answer our main research question we must tackle the latent nature of preferences. It is common to infer preferences by observing an individual's choices (Samuelson, 1948). If multiple pairwise choices of an individual over a set of items are observed, then a Bradley-Terry model can be estimated to find a ranking of all items. The Bradley-Terry model was also used by Rawal and Lakkaraju (2020) to generate counterfactual explanations (CEs). However, a key difference is that they estimate a joint Bradley-Terry model to find the ranking for all applicants. While we estimate a separate model for each applicant.

Synthetic preference simulation is used to evaluate UDACE since obtaining real pairwise choice data using a field experiment lies outside the scope of our research. We incorporate the preferences into the DACE formulation by extending the formulation to become a multi-objective optimization problem. This problem simultaneously minimizes effort while maximising utility, where effort is the amount of work required by an applicant to act. Since effort is not known a priori, we use the cost function proposed by Kanamori et al. (2020) as a proxy. The multi-objective approach, allows us to generate multiple CEs for the same applicant with a spectrum of associated efforts and utilities.

Our research has both scientific and practical relevance, firstly research on the inclusion of preferences in counterfactual explanation methods is limited in the literature (Verma, Dickerson & Hines, 2020). Secondly, the practical relevance of our research stems from the fact that recourse is more likely to occur if a counterfactual explanation aligns with the preferences of the applicant who has to perform the actions.

Additionally, the economic relevance of our research stems from the fact that we capture the preferences of applicants within a utility framework. Hence, our approach fits within the economic imperialist tradition initiated by Becker (2010), who used the economic toolkit to describe a variety of phenomena, including discrimination and its impact.

The inclusion of preferences is necessary for successful recourse. In essence, a counterfactual explanation method optimizes a cost functions, which is often a distance metric. The goal is to find the smallest possible perturbation that produces a different outcome (Wachter, Mittelstadt & Russell, 2017). However, we can question whether the smallest perturbation results in the highest probability of recourse. For example, suppose a counterfactual explanation in a medical context advises an individual to start consuming chicken. This action could be easy for some people in the population, but if we ask a vegetarian to start eating chicken, the utility loss for the vegetarian is likely to be so large that she will never engage in recourse.

By employing our proposed UDACE method, we demonstrate the applicability of a multi-objective optimization approach in generating multiple Pareto-efficient CEs. This enables applicants to have a range of options to choose from. In cases where an applicant is unable to make a choice, we provide a suggested best action that makes a trade-off between effort and utility.

1.1 Contributions

The main contributions of our work are:

- Design a method to generate synthetic preference orderings for evaluation purposes.
- Enhance the DACE formulation by incorporating applicant preferences into the framework.
- Formulate a multi-objective optimization problem that considers both effort and utility, aiming to identify a subset of the Pareto-frontier.
- Utilize a selection mechanism based on previous studies which makes a trade-off between Pareto-efficient CEs to suggest a single CE.

2 Literature review

In this review we aim to contextualize the position of our UDACE method within the existing literature. We first include a brief review on relevant CE methods and then proceed to discuss how preferences can be captured effectively using utility.

2.1 Counterfactual explanations

The CE method was originally developed by Wachter et al. (2017). Since their publication, there has been a surge of research interest in CEs. Although the name ”counterfactual” has no direct link to counterfactuals in a causal context. Nonetheless, some studies have incorporated causal graphs in the generation of CEs (Mahajan, Tan & Sharma, 2019).

Several studies have incorporated the preferences of applicants in generating CEs. For instance, the study by Mahajan et al. (2019) introduces applicant-specific constraints to ensure that CEs are locally feasible. Downs, Chu, Yacoby, Doshi-Velez and WeiWei (2020) capture preferences by allowing the applicant to specify constraints, which they use with a Conditional Subspace Variational Autoencoder to generate multiple CEs. Rawal and Lakkaraju (2020) consider the preferences of all applicants by estimating a Bradley-Terry model on pairwise comparisons of actions. They also conduct a small-scale field experiment to validate their approach.

There are also studies which implicitly capture institutional preferences, by categorizing an action into one of three distinct categories: immutable, improvement or manipulative (Chen, Wang & Liu, 2020).

Studies which include the preferences of multiple stakeholders are relatively rare. The only study we could find was done by Tsirtsis and Gomez Rodriguez (2020). They take into account the preferences of the institution and the applicant by formulating the process of recourse as a Stackelberg competition.

Most CE methods consider a weighted sum of objectives when looking for an action. The first study which used multi-objective optimization to produce CEs was done by Dandl, Molnar, Binder and Bischl (2020). They use a modified Nondominated Sorting Genetic Algorithm to find multiple CEs. Their Multi-Objective Counterfactuals method is competitive compared to other diverse counterfactual generation methods on the German credit dataset (Dua & Graff, 2017). Mothilal, Sharma and Tan (2020) generate a diverse set of CEs by introducing the Diverse Counterfactual Explanations (DiCE) framework based on detrimental point processes. Overall, the existing literature presents an opportunity for a valuable contribution which integrates studies on preferences with the generation of multiple CEs.

2.2 Utility modelling

A core aspect of the economic discipline has always been to try to understand the behaviour of economic agents. To do so the concept of utility was introduced. The nature of utility has been debated for a long time. Marshall (2009) utilizes a cardinal form of utility, when he introduces the Marshallian demand function. While on the other hand, Vilfredo Pareto introduced an ordinal form of utility, where the focus lies on indifference curves (Wicksteed, 1906).

The key difference between cardinal and ordinal utility is that in an ordinal framework the number of utils does not say anything about how much you prefer one state over another. One can only state whether a certain state is preferred or not preferred when compared to another state, without making any claims about the intensity of the preference. Even though the concept of utility has been widely used and accepted, it has also been criticized. For example by Robinson (1962), who argued that utility theory cannot be tested scientifically because of the assumption of preference stability.

Over time several models have been developed mostly in the field of discrete choice theory. These models can often be derived from an underlying utility specification. One of these models is the Bradley-Terry (BT) model developed by Bradley and Terry (1952), which is a type of logistic regression model with latent variables. The BT model can be used to describe choices between a set of items when pairs of items are compared. The BT model satisfies the independence of irrelevant alternatives assumption, meaning that the outcome of a pair-wise comparison is not influenced by other items present. This assumption is reasonable but does limit the complexity of preferences one can have. For example, if one prefers coffee only if there is a chocolate cake and else one prefers milk. These preferences cannot be captured by a BT model.

Based on the previous works on utility, we extract three axioms that must hold for UDACE to be a valid approach to obtaining CEs. The first axiom is that preferences are stable over time. The second axiom is that people reveal their preferences when they make choices. The third axiom is that people maximize their utility when deciding their course of action.

3 Data

We will utilize the same data sources as Kanamori et al. (2020), which are the FICO dataset (FICO et al., 2018) and German credit dataset (Dua & Graff, 2017). Both datasets are publicly available in the UCI machine learning repository (Dua & Graff, 2017).

The FICO dataset is a dataset on the home equity line of credit (HELOC). A HELOC uses the underlying property of a lender as collateral for a loan with a fixed term. A classifier is used to determine whether an applicant should receive a line of credit or not. The FICO dataset has 23 integer-valued features.

The German credit dataset is a dataset with loan applications submitted to German banks. The prediction task for a classifier is to predict whether an individual will default on their loan.[1] The German credit dataset has 61 features, 54 of which are binary and 7 are integer-valued.

Besides these data sources, synthetic preference ranking data will be generated. For reproducibility, the procedure used to generate the data will be described and the synthetic datasets will be made publicly available.[2]

[1] The authors of this paper do not endorse interest-based lending, the FICO and German dataset are used solely for comparison with prior studies.

[2] The datasets can be retrieved from the zip file submitted with this paper or from ht-tps://github.com/achasol/udace

4 Methodology

The methodology is split into several sections. Firstly, the general notation will be introduced. Afterwards, the construction of a synthetic preference dataset will be discussed. Then the problem will be defined and a formulation for UDACE will be derived. Subsequently, we will discuss the algorithm used to find Pareto-efficient CEs. In the end, we will discuss the algorithm we use to find a single suggested CE.

4.1 General notation

Since UDACE is an extension of the DACE method, the notation introduced by Kanamori et al. (2020) is used. For an assertion ψ, let $\mathbb{1}[\psi]$ be it's characteristic function. Such that $\mathbb{1}[\psi] = 0$, if ψ is false and $\mathbb{1}[\psi] = 1$ if ψ is true.

The type of automated decision-making model we consider are binary additive classifiers. Suppose a classifier has D features. Let the set of all possible values for a feature i be denoted by $\mathcal{X}_i \subseteq \mathbb{R}$ with $i = 1, .., D$. Then the entire feature space is defined as $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_D$, and the output range is denoted by $\mathcal{Y} = \{-1, 1\}$. Let a classifier be defined as $\mathcal{H} : \mathcal{X} \to \mathcal{Y}$, and let an instance be defined as $x = (x_1, ..., x_D) \in \mathcal{X}$.

If a dataset includes categorical variables, it is necessary to apply one-hot encoding to these variables. Let $\mathcal{G} \subseteq 2^{1,...,D}$ represent the set of all sets of features that constitute a categorical variable. Each set $G \in \mathcal{G}$ represents a one-hot encoded variable with $|G|$ possible values. The feature space of $g \in G$ can be defined as $\mathcal{X}g = \{0, 1\}$, and it must hold that $\sum_{g \in G} x_g = 1$ holds for all $x \in \mathcal{X}$.

For a given instance $\bar{x} \in \mathcal{X}$ and a classifier $\mathcal{H} : \mathcal{X} \to \mathcal{Y}$, an action $\alpha \in \mathbb{R}^D$ is defined as a perturbation vector which results in $\mathcal{H}(\bar{x} + \alpha) = 1$, while $\mathcal{H}(\bar{x}) = -1$. The finite set of all feasible actions is denoted by $\mathcal{A} = A_1 \times \cdots \times A_D$. Where the set of feasible values a single feature can take is $A_d \subseteq \{\alpha_d \in \mathbb{R} \mid \bar{x}_d + \alpha_d \in \mathcal{X}_d\}$, for $d = 1, ..., D$. For all features it follows that $0 \in A_d$ and specifically for immutable features, it must hold that $A_d = \{0\}$. Kanamori et al. (2020) mention that the feasible set A_d of a feature can be determined in advance, but the feasible set does depend on the properties of the classifier.

4.2 Classifiers

We consider two types of binary additive classifiers. The first type of classifier is a logistic regression model, which is a member of the class of linear models.

The second type is a Random Forest model, as part of the class of tree ensemble models. Any classifier $\mathcal{H} : \mathcal{X} \to \mathcal{Y}$ part of the aforementioned classes can be represented as:

$$\mathcal{H}(x) = \text{sgn}(\sum_{t=1}^{T} w_t \cdot h_t(x) - b)$$

The weight of a base-learner $h_t : \mathcal{X} \to \mathbb{R}$ is defined by w_t for $t = 1, .., , T$. And the intercept is denoted by b.

Notably for the logistic regression model, the base-learners are equal to the features of the dataset. While for the random forest model, each base-learner is a decision tree. The Random forest model classifies an applicant by averaging the predictions of all T decision trees.

4.3 Components in the objective

In this subsection, each component used in our objective functions is introduced.

4.3.1 Local Outlier Factor

To make sure that the generated CEs are not considered outliers, Kanamori et al. (2020) use the Local Outlier Factor (LOF) as a cost component.

The local outlier factor was initially developed by Breunig, Kriegel, Ng and Sander (2000). To define the LOF, we define the distance metric $\triangle : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ on $\mathcal{X}$. We also select a set of N instances $X \subseteq \mathcal{X}$. For any instance $x \in \mathcal{X}$, let the set of k-nearest neighbours of x within X be denoted by $N_k(x)$. Let the distance from x to it's k-th nearest neighbour be denoted by $d_k(x)$. Then the k-local reachability-density of x is defined as:

$$lrd_k(x) = \frac{|N_k(x)|}{\sum_{z \in N_k(x)} \max(\triangle(x, z), d_k(x))}$$

Using the local reachability-density, the k-LOF of x can be defined as:

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

4.3.2 Mahalanobis distance

Kanamori et al. (2020) utilize the Mahalanobis distance (MD) introduced by Chandra et al. (1936), to evaluate how realistic a counterfactual explanation is relative to other applicants in the population. For any two instances $x, z \in \mathbb{R}^D$ and a positive semi-definite matrix $\Sigma \in \mathbb{R}^{D \times D}$. The Mahalanobis distance is defined as:

$$d_M(x, z, \Sigma) = \sqrt{(x - z)^T \Sigma (x - z)} = \|U(x - z)\|_2$$

Where the second equality follows from the spectral decomposition because any positive semi-definite matrix Σ can be decomposed such that $\Sigma = U^T U$.

Kanamori et al. (2020) argue that optimizing the Mahalanobis distance directly is computationally infeasible. Hence they use the l1-norm MD as a surrogate distance measure. Let U_d be the d'th row vector of U. Then the l1-norm Mahalanobis distance can be defined as:

$$\hat{d}_M(x, x + \alpha, \Sigma) = \|U\alpha\|_1 = \sum_{d=1}^{D} |\langle U_d, \alpha \rangle|$$

4.3.3 Applicant utility

To account for the preferences of an applicant they have to be measured. However, directly measuring preferences over the feature space $\mathcal{X}$ is infeasible. Depending on the size of D, even ranking the features itself could be considered an arduous process.

Hence we propose that the features are partitioned into categories. This way an applicant only needs to rank the categories.

Define Q to be a partition of $\{1, ..., D\}$. into categories. For an instance x, let the corresponding applicant have a preference relation on Q. Such that for all $c_1, c_2 \in Q$ it either holds that $c_1 \succ c_2$ or $c_2 \succ c_1$. Meaning an applicant prefers changes in one category over another category.

Let c_i denote the i'th category with $i = 1, ..., |Q|$. The latent utility of a category c_i for an applicant is denoted by $u_i \in \mathbb{R}_{\geq 0}$. Where it must hold that $u_i > u_j$ if and only if $c_i \succ c_j$, for all $c_i, c_j \in Q$ with $i, j = 1, ..., |Q|$, $i \neq j$. The latent utility of an action $\alpha \in \mathcal{A}$ is defined as:

$$u(\alpha) = \sum_{i=1}^{|Q|} \sum_{d \in c_i} u_i \cdot \mathbb{1}[\alpha_d \neq 0]$$

So an applicant values actions which change features in preferable categories more. While no utility is assigned to features which remain unchanged.

4.4 Problem definition

We look for a set of Pareto-efficient actions for the optimization problem stated below, given binary additive classifier $\mathcal{H} : \mathcal{X} \to \mathcal{Y}$, an instance $\bar{x} \in \mathcal{X}$ such that $\mathcal{H}(\bar{x} = -1)$, a partition of the features Q, a set of instances $X \in \mathcal{X}$, a weighting factor $\lambda \geq 0$ and a positive semi-definite matrix $\Sigma \in \mathbb{R}^{D \times D}$.

$$\begin{aligned} \min_{\alpha \in \mathcal{A}} \quad & \hat{d}_M(\bar{x}, \bar{x} + \alpha, \Sigma) + \lambda \cdot q_1(\bar{x} \mid X) \\ \max_{\alpha \in \mathcal{A}} \quad & u(\alpha) \\ \text{subject to} \quad & \mathcal{H}(\bar{x} + \alpha) = 1 \end{aligned}$$

4.5 Measuring utility

Both the FICO and German datasets do not contain any utility information on the applicants. Hence latent utilities for all applicants are simulated. To take into account that latent utilities are not observed, a Bradley-Terry model is used to infer the latent utilities from simulated pair-wise comparisons.

4.5.1 Synthetic preference simulation

The preferences of applicants are simulated by repeatedly asking them to compare two categories and state their preferences. Initially, it is assumed that any applicant has some latent rational ordering of preference over the partition Q denoted by $(c_1, ..., c_{|Q|})$. Where rational is defined as a transitive and complete ordering over Q.

During the simulation, two distinct components are captured that differ randomly across applicants. The first is a motivation component which determines how many pairwise comparisons an applicant is willing to perform.

The second component is the fatigue effect, which captures the fact that as an applicant performs a large amount of pair-wise comparisons, we expect the probability of a mistake to increase over time. Meaning that the applicant was not able to properly convey their latent preference. The algorithm for the preference simulation is given below:

Algorithm 1 Algorithm to generate synthetic preference data.

```
Require: $n \geq 0$
Ensure: $y = x^n$
  $\psi \sim Beta(2, 32)$
  $f_0 \leftarrow 1$
  $f_e \sim Beta(10, 2)$
  $K \sim Geom(\psi)$
  $N = min(|Q| - 1 + K, 0.5|Q|(|Q| - 1)$
  $i = 0$
  $pairs = \{\}$
  while $i < N$ do
      $(c_1, c_2) = drawUniformUniquePairWithPreference(pairs)$
      $J \sim Bernoulli(f_0 + (f_e - f_0)\sqrt{\frac{i}{N}})$
      if $J == 1$ then
          $pairs = pairs \bigcup \{(c_1, c_2)\}$
      else if $J == 0$ then
          $pairs = pairs \bigcup \{(c_2, c_1)\}$
      end if
      $i \leftarrow i + 1$
  end while
  return $pairs$
```

The definition of the drawUniformUniquePairWithPreference() subroutine is excluded for brevity. This subroutine uses a Priority Queue to uniformly draw pairs which have not yet been drawn from the preference relation. For more information on this subroutine, we refer to the source code included with this paper.

We repeat the steps above for each applicant in a dataset and store the pair-wise comparisons for all applicants.

4.5.2 Bradley-Terry model

Given the simulated pair-wise comparisons we estimate a Bradley-Terry model. The Bradley-Terry (BT) model is a logistic regression model with latent parameters for each category. Let the total number of categories be denoted by $w = |Q|$, then for any pair of categories $i, j = 1, ..., w$ the probability that category c_i is preferred over c_j is

$$P(c_i \succ c_j) = \frac{e^{u_1}}{e^{u_1} + e^{u_2}}$$

The Bradley-Terry model is estimated using a Bayesian approach similar to Caron and Doucet (2012). The parameters $u_1, ..., u_w$ are assigned LogNormal(1,1) prior distributions, to ensure that they are positive. Let the maximum a posteriori probability (MAP) estimates of the parameters be denoted by $\hat{u}_1, ..., \hat{u}_w$. The MAP estimates will be used in the utility objective as weights.

4.6 MOMIP formulation

We extend the MILO formulation proposed by Kanamori et al. (2020). To describe the formulation we first discretize the feasible action space. Let $\pi_{d,i} \in \{0,1\}$ be a binary variable which is 1, if $\alpha_{d,i} \in A_d$ is chosen and 0 otherwise for $d = 1, ..., D$, $i = 1, ..., |A_d|$. Our goal in this section is to express all objectives and constraints as linear combinations of $\pi_{d,i}$

Starting with the constraints, since only one value for a feature can be included in an action, we introduce the constraints:

$$\sum_{i=1}^{|A_d|} \pi_{d,i} = 1, \forall d \in \{1, ..., D\} \tag{1}$$

For one-hot encoded categorical features we introduce constraints which preserve their one-hot encoding. For these constraints we make use of the decomposition $a_d = \sum_{i=1}^{|A_d|} \alpha_{d,i}\pi_{d,i}$

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

To effectively formulate the constraint based on the classifier, we observe that based on the definition of an additive classifier, if $\mathcal{H}(\bar{x} + \alpha) = 1$ then

$$\sum_{t=1}^{T} w_t h_t(\bar{x} + \alpha) \geq b \tag{3}$$

For a linear model we know that $T = D$ and $h_d(\bar{x} + \alpha) = \bar{x}_d + \alpha_d$, for $d = 1, ..., D$ by definition. Hence using the decomposition of α_d the classifier constraint for linear models is given by

$$\sum_{d=1}^{D} w_d(\bar{x}_d + \sum_{i=1}^{|A_d|} \alpha_{d,i}\pi_{d,i}) \geq b \tag{4}$$

For tree-ensemble models, the classifier constraint can be derived based on two observations made by Kanamori et al. (2020). The first observation using the work by Hastie, Tibshirani, Friedman and Friedman (2009) is that a decision tree $h_t : \mathcal{X} \to \mathcal{Y}$ with L_t leaves, can be represented as a partition of the feature space $\mathcal{X}$ into a set $\{r_{t,1}, ..., r_{t,L_t}\}$. Let $\hat{y}_{t,l}$ denote the prediction of leaf $l \in \{1, ..., |L_t|\}$ of tree h_t. Then the prediction of tree h_t is the prediction of the leaf which contains the instance, expressed as

$$h_t(x) = \sum_{l=1}^{L_t} \hat{y}_{t,l} \mathbb{1}[x \in r_{t,l}]$$

The second observation is that $x + \alpha \in r_{t,l}$ can be expressed using the decision logic constraint introduced by Cui, Chen, He and Chen (2015).

To utilize the decision logic constraints, introduce the binary variable $\phi_{t,l}$ (constraint 5), and constrain it such that $\phi_{t,l} = \mathbb{1}[x + \alpha \in r_{t,l}]$. Additionally introduce the set $I_{t,l}^{(d)} = \{i \in$

$\{1, ..., |A_d|\} \mid \bar{x}_d + \alpha_{d,i} \in r_{t,l}^{(d)}\}$, where $r_{t,l}^d \subseteq \mathcal{X}_d$, such that $r_{t,l} = r_{t,l}^{(1)} \times ... \times r_{t,l}^{(D)}$. Where the set $I_{t,l}^d$ can be regarded as an indicator because for a feature d, it only contains the discrete action candidates, that fall within segment $r_{t,l}^{(d)}$ of the partition.

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

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

$$\phi_{t,l} \leq \frac{1}{D} \sum_{d=1}^{D} \sum_{i \ \in I_{t,l}^{(d)}} \pi_{d,i}, \forall t \in \{1, ..., T\}, l \in \{1, ..., |L_t|\} \tag{7}$$

Constraint (6) makes sure that a perturbed instance can only end up in a single leaf of a tree. Constraint (7) is derived by Cui et al. (2015) based on the finding that if $\phi_{t,l} = 0$ then a perturbed instance is not contained within leaf l, and hence there must exist at least one feature k such that $\sum_{i \in I_{t,l}^{(k)}} \pi_{k,i} = 0$. On the other hand if $\phi_{t,l} = 1$ then an perturbed instance is contained within leaf l and hence $\sum_{i \in I_{t,l}^{(k)}} \pi_{k,i} = 1, \forall d \in \{1, ..., D\}$.

In combination with constraints (5), (6) and (7), the classifier constraint for tree ensemble models is given by

$$\sum_{t=1}^{T} w_t (\sum_{l=1}^{L_t} \hat{y}_{t,l} \cdot \phi_{t,l}) \geq b, \forall t \in \{1, ..., T\} \tag{8}$$

4.6.1 Mahalanobis distance

To include $\hat{d}_M(x, x + \alpha, \Sigma)$ as a linear objective function. We introduce $\delta_d \geq 0$, and define constraints such that $\delta_d = |\langle \ U_d, \alpha \ \rangle|$. The MD distance can then be expressed as:

$$\hat{d}_M(x, x + \alpha, \Sigma) = \sum_{d=1}^{D} \delta_d$$

subject to the constraints:

$$-\delta_d \leq \sum_{d'=1}^{D} U_{d,d'} (\sum_{i=1}^{|A_d|} \alpha_{d',i} \pi_{d',i}) \leq \delta_d, \forall d \in \{1, ..., D\} \tag{9}$$

4.6.2 Local outlier factor

To make the computation of the local outlier factor feasible, Kanamori et al. (2020) choose to fix $k = 1$. The 1-LOF component can then be written as

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

with the closest neighbour of $\bar{x} + \alpha$ denoted by $m = argmin_{n \in 1,..,N} \triangle(\bar{x} + \alpha, x^{(n)})$. To express the variables dependent on $\bar{x} + \alpha$ in terms of $\pi_{d,i}$, define $\nu_n \in \{0, 1\}$ and $\rho_n \geq 0$ for $n = \{1, ..., N\}$.

We will add constraints such that $\nu_n = \mathbb{1}[x^{(n)} \in N_1(\bar{x} + \alpha)]$ and $\rho_n = rd_1(\bar{x} + \alpha, x^{(n)}) \cdot \nu_n$. If this is the case then $q_1(\bar{x} + \alpha)$ can be written linearly in ρ_n as follows:

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

Where $l^{(n)} = lrd_1(x^{(n)})$. The constraints to ensure the definitions of ν_n and ρ_n hold are:

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

Which ensures that $|N_1(\bar{x} + \alpha)| = 1$.

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

Where C_n, $c_{d,i}^{(n)}$ are constants such that $C_n \geq \max_{\alpha \in \mathcal{A}} \triangle(\bar{x} + \alpha, x^{(n)})$ and $c_{d,i}^{(n)} = \triangle_d(\bar{x}_d + \alpha_{d,i}, x_d^{(n)})$. Constraint 11 ensures that if a neighbour $x^{(n)}$ is selected i.e. $\nu_n = 1$. Then, this neighbour must be the nearest neighbour of $\bar{x} + \alpha$

$$\rho_n \geq d^{(n)} \cdot \nu_n, \forall n, \in \{1, ..., N\} \tag{12}$$

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

With $d^{(n)}$ a constant defined as $d^{(n)} = d_1(x^n)$. Constraint (12) and (13) together enforce the definition of the k-reachability distance. Since if a neighbour is selected i.e. $\nu_n = 1$, then they jointly require that $\rho_n \geq \max(d^{(n)}, \triangle(\bar{x} + \alpha, x^{(n)})$ for an action to be feasible. If a neighbour is not selected ($\nu_n = 0$) then both constraints become trivial since $\rho_n \geq 0$ by definition.

4.6.3 Utility of the applicant

To linearize the utility of the applicant, $\mathbb{1}[\alpha_d \neq 0]$ needs to be expressed in terms of $\pi_{d,i}$. Define $\pi_{d,1}$ to be the binary variable such that $a_{d,1} = 0$. This can always be achieved by shuffling the elements of the set A_d. The utility of the applicant can then be expressed as:

$$u(a) = \sum_{j=1}^{|Q|} \sum_{d \in c_j} \hat{u}_j \cdot \mathbb{1}[\alpha_d \neq 0] = \sum_{j=1}^{|Q|} \sum_{d \in c_j} \hat{u}_j \sum_{i=2}^{|A_d|} \pi_{d,i} \tag{14}$$

The expression follows from the constraint that $\sum_{i=1}^{|A_d|} \pi_{d,i} = 1$ for $d = 1, ..., D$. Since if $\pi_{d,1} = 0$ indicating that $\alpha_d \neq 0$, then it must hold that: $\sum_{i=2}^{|A_d|} \pi_{d,i} = 1$, otherwise if $\pi_{d,1} = 1$ then $\sum_{i=2}^{|A_d|} \pi_{d,i} = 0$. Hence it follows that $\mathbb{1}[\alpha_d \neq 0]$ can be represented as $\sum_{i=2}^{|A_d|} \pi_{d,i}$

4.6.4 Complete formulation

Using the aforementioned constraints and objectives the complete formulation for UDACE is:

$$
\begin{aligned}
\text{min} \quad & \sum_{d=1}^{D} \delta_d + \lambda \sum_{n=1}^{N} l^{(n)} \cdot \rho_n \\
\text{max} \quad & \sum_{j=1}^{|Q|} \sum_{d \in c_j} \hat{u}_j \sum_{i=2}^{|A_d|} \pi_{d,i} \\
\text{subject to} \quad & \text{Constraint } (1 \text{ - } 3) \\
& \begin{cases} \text{Constraint } (4), & \text{if } \mathcal{H} \text{ is a LM} \\ \text{Constraint } (5 \text{ - } 8), & \text{if } \mathcal{H} \text{ is a TEM} \end{cases} \\
& \text{Constraint } (9 \text{ - } 13) \\
& \pi_{d,i} \in \{0,1\}, \forall d \in \{1,...,D\}, \forall i \in \{1,...,|A_d|\} \\
& \delta_d \geq 0, \forall d \in \{1,...,D\} \\
& \nu_n \in \{0,1\}, \rho_n \geq 0, \forall n \in \{1,...,N\}
\end{aligned}
$$

4.7 Searching pareto-efficient solutions

To solve the multi-objective optimization problem, the algorithm proposed by Kirlik and Sayın (2014) is used. This algorithm is based on the ϵ-constraint method but has a novel search mechanism which effectively partitions the search space of ϵ. The method of Kirlik and Sayın (2014) is capable of generating all non-inferior solutions for a discrete optimization problem. However, the UDACE formulation is not fully discrete, and hence there are no guarantees that the method will produce all non-inferior solutions. Practical experiments show that the method does produce a set of Pareto-efficient solutions.

For bi-objective optimization problems, the ϵ-constraint method would also suffice. The reason we choose the extension of Kirlik and Sayın (2014) is twofold. The first reason is that we are not interested in enumerating all Pareto-efficient solutions, hence a method which prioritizes the search process should be more time efficient. The second reason is that the method by Kirlik and Sayın (2014) can be used for p-dimensional optimization problems, so it would be trivial to incorporate multiple utility objectives of different stakeholders in our UDACE method.

The core of the algorithm is based on solving a two-stage single-objective optimization problem. For any bi-objective minimization problem, let the objectives be denoted by $(f_1(z), f_2(z))$, with $z \in \mathcal{W}$. Where $\mathcal{W}$ is the feasible region of the problem. The first sub-problem in the two-stage process is given by:

$$
\begin{aligned}
\underset{z}{\text{minimize}} \quad & f_1(z) \\
\text{subject to} \quad & f_2(z) \leq \epsilon_2, \\
& z \in \mathcal{W}
\end{aligned} \tag{15}
$$

In this problem, we look for a solution z^*, which is optimal for the first-objective given a threshold

ϵ_2 on the second objective. To find an optimal solution, We then solve the following problem:

$$\begin{aligned} \underset{z}{\text{minimize}} \quad & f_1(z) + f_2(z) \\ \text{subject to} \quad & f_2(z) \leq \epsilon_2, \\ & f_1(z) = z^*, \\ & z \in \mathcal{W} \end{aligned} \tag{16}$$

For the second stage problem we look for the optimal value of $f_2(z)$, while restricting $f_1(z)$ to the optimal value found previously. Kirlik and Sayın (2014) prove that an optimal solution of the two-stage problem is an efficient solution for the bi-objective optimization problem. To find the threshold parameters, the search space is partitioned into rectangles, and a volume measure is used to determine the order in which they are searched.

4.7.1 Suggesting a single pareto-efficient solution

Once we have obtained a set of Pareto-efficient actions, we can choose to present this set to the applicant, and let her pick the action which suits her best.

We also explore suggesting a "best" action using the selection algorithm proposed by Wang, Zhao, Wu and Wu (2017). The algorithm uses the price-performance ratio to select a single solution from a set of non-inferior solutions.

In our context, the algorithm makes a trade-off between the effort required to act, and the utility yielded by an action. Where we argue that the weighted sum of the MD and LOF is a valid proxy for the effort of an action. Intuitively Wang et al. (2017) explain that their method looks for an efficient solution which is acceptable for both objectives.

The algorithm of Wang et al. (2017) can be formulated as follows. Suppose M Pareto-efficient actions for an applicant have been found. Let the actions be sorted by the values of their first objective in ascending order. We denote the objective values of the m'th action as $(f_1^{(m)}, f_2^{(m)})$, for $m = 1, ..., M$. The algorithm is based on the average variability, which is the average of the slopes between a point and its two direct neighbours, except for the two points at the edges of the sorted set. The average variability of the j'th objective of the m'th action $k_j^{(m)}$ is defined by:

$$k_j^{(m)} = \left\{ \begin{array}{ll} (\frac{f_2^{(2)} - f_2^{(1)}}{f_1^{(2)} - f_1^{(1)}})^{(-1)^{j-1}}, & \text{for } m = 1 \\ \frac{1}{2}(\frac{f_2^{(m)} - f_2^{(m-1)}}{f_1^{(m)} - f_1^{(m-1)}})^{(-1)^{j-1}} + \frac{1}{2}(\frac{f_2^{(m+1)} - f_2^{(m)}}{f_1^{(m+1)} - f_1^{(m)}})^{(-1)^{j-1}}, & \text{for } m = 2, 3, ..., M-1 \\ (\frac{f_2^{(M)} - f_2^{(M-1)}}{f_1^{(M)} - f_1^{(M-1)}})^{(-1)^{j-1}}, & \text{for } m = M \end{array} \right\}$$

With $j = 1, 2$ since we have two objectives. The sensitivity ratio of an action scales the average variability by the value of the j'th objective, and is given by $\delta_j^{(m)} = \frac{k_j^{(m)}}{f_j^{(m)}}$, for $m = 1, ..., M$. Since we want to compare sensitivity-ratio's, we define the non-dimensionalized sensitivity ratio as $\epsilon_j^{(m)} = \frac{\delta_j^{(m)}}{\sum_{i=1}^{M} \delta_j^{(i)}}$. The position m^* of the action suggested by the algorithm is given by:

$$m^* = arg\, min_{m=1,...M} \mid \epsilon_1^{(m)} - \epsilon_2^{(m)} \mid \tag{17}$$

Every time a set of Pareto-efficient CEs is found, we use the algorithm described above to find the suggested CE.

4.8 Implementation

All methods described above are implemented using Julia 1.8.5. The Turing.jl package by Ge, Xu and Ghahramani (2018) is used for Bayesian inference of the Bradley-Terry model. The JuMP library (Lubin et al., 2023) in combination with the HiGHS solver (Huangfu & Hall, 2018) are used to solve the MOMIP formulation. For more details on our implementation we refer to Appendix B and the included source code with this paper.

5 Experiments

5.1 Reproducing DACE

To ensure the correctness of our Julia implementation, we reproduce the results by Kanamori et al. (2020). This means that we solve the complete formulation while removing the second utility objective. An experiment is performed with the FICO and German datasets. Both datasets are split at random into a 70% estimation sample and 30% validation sample. For the trade-off parameter λ, the values found using a sensitivity analysis by Kanamori et al. (2020), are used which are $\lambda = 0.01$ for the German dataset and $\lambda = 1.0$ for the FICO dataset.

Using the estimation sample, a l_2 regularized logistic regression model is estimated. And a random forest model with 100 trees and a maximum depth of four is estimated as well. Using the validation sample and the estimated models a prediction is made whether an applicant will default on her loan. Counterfactual explanations are generated for 50 of the applicants who are predicted to default on their loans. We impose that $|A_d| \leq 100$, for $d \in \{1, \ldots, D\}$, restricting the number of states a feature can take, and enforcing a time-limit of 600 seconds per CE.

For comparison, several baseline methods are implemented. These methods only differ in the distance function used. The first baseline is the total log percentile shift (TLPS) developed by Ustun, Spangher and Liu (2019). The second baseline is the weighted l_1-norm using the inverse of the mean absolute deviation (MAD) as introduced by Wachter et al. (2017). The final baseline is the l_2-norm of the Pearson correlation coefficients (PCC), introduced by Ballet et al. (2019).

The same evaluation metrics as Kanamori et al. (2020) are reported, which are the Mahalanobis distance, the 10-LOF and solver time in seconds. To compute the Mahanobis distance, the estimated covariance matrix of the estimation sample Σ^{-1} is used. To calculate the 10-LOF the applicants who were approved within the estimation sample are used, denoted by X_+. To run the experiments, we use a machine with 16 GB of RAM and an AMD Ryzen 3600 6-core processor with a clock speed of 3.6 GHz.

The results can be seen in tables 1, 2, 3 and 4. Our results are similar to those obtained by Kanamori et al. (2020). However, with our choice of validation sample the DACE method does not always, outperform the other methods. The slight differences between the results are likely to be caused by the difference in random number generation between Julia and python. Which affects the estimation of the models, and the split of the dataset.

Overall the methods produce relatively similar values for the 10-LOF. The Mahalanobis distance appears to differ between the methods, but can also be quite close, see for example Table 3. As noted by Kanamori et al. (2020), the DACE method is very slow compared to the other methods. Looking at the results it is not clear whether this additional time pays off, compared to the TLPS method which seems to be quite similar to DACE, but uses a fraction of the time.

In particular for the random forest model and the FICO dataset (Table 3), the optimal solution is not always found within the time limit. The probable reason behind this observation is that the FICO dataset solely consists of integer features, which in turn provides the solver with a broader feasible region to explore.

Counterfactual explanations for the German dataset can be solved within a relatively quick time (Table 2, 4). This was to be expected given that 52 out of the 61 features are binary and encode a much smaller set of categorical variables.

Table 1

Results for the Logistic regression model on the FICO dataset ($N = 50$, $D = 23$)

		Logistic Regression	
	$d_M(\bar{x}, \bar{x} + \alpha)\|\Sigma^{-1})$	$q_{10}(\bar{x}, \bar{x} + \alpha\|X_+)$	Time[s]
MAD	6.96 (5.67)	1.23 (0.23)	0.04 (0.01)
TLPS	5.85 (5.36)	1.22 (0.19)	0.12 (0.53)
PCC	8.12 (5.23)	1.23 (0.23)	0.04 (0.01)
DACE	2.05 (1.35)	1.28 (0.33)	45.66 (30.31)

Table 2

Results for the logistic regression model on the German dataset ($N = 50$, $D = 61$)

		Logistic Regression	
	$d_M(\bar{x}, \bar{x} + \alpha)\|\Sigma^{-1})$	$q_{10}(\bar{x}, \bar{x} + \alpha\|X_+)$	Time[s]
MAD	7.11 (4.53)	1.58 (1.71)	0.01 (0.01)
TLPS	2.94 (1.70)	1.33 (0.21)	0.09 (0.52)
PCC	8.82 (3.64)	1.59 (1.70)	0.01 (0.01)
DACE	2.49 (1.37)	1.06 (0.06)	2.89 (1.21)

Table 3

Results for the Random forest model on the FICO dataset ($N = 50$, $D = 23$, $T = 100$)

		Random forest	
	$d_M(\bar{x}, \bar{x} + \alpha)\|\Sigma^{-1})$	$q_{10}(\bar{x}, \bar{x} + \alpha\|X_+)$	Time[s]
MAD	2.40 (1.72)	1.27 (0.30)	104.26 (123.04)
TLPS	2.14 (1.49)	1.26 (0.32)	120.89 (137.74)
PCC	2.64 (1.81)	1.27 (0.31)	140.99 (172.68)
DACE	2.62 (1.92)	1.26 (0.27)	577.75 (73.80)

Table 4

Results for the Random forest model on the German dataset ($N = 6$, $D = 61$, $T = 100$)

		Random forest	
	$d_M(\bar{x}, \bar{x} + \alpha)\|\Sigma^{-1})$	$q_{10}(\bar{x}, \bar{x} + \alpha\|X_+)$	Time[s]
MAD	3.55 (2.75)	2.62 (3.27)	0.96 (0.56)
TLPS	1.64 (1.65)	2.53 (3.32)	1.98 (1.22)
PCC	8.34 (1.85)	2.6 (3.29)	1.0 (0.59)
DACE	2.14 (2.02)	1.01 (0.05)	134.0 (122.99)

Note. The results in the table are based on $N = 6$ observations. This is an unintended side-effect of our random seed, and the class imbalance in the German credit dataset.

5.2 Evaluating UDACE

To evaluate the UDACE model we use the same procedure described above when reproducing the results for the DACE model. Only this time we do include the second utility objective and search for Pareto-efficient solutions. We also include the other cost functions TLPS, MAD, and PCC with the second utility objective and refer to these cost functions as UTLPS, UMAD, UPPC. We enforce a strict time limit of 600 seconds, to ensure the practical feasibility of our approach.

Using the descriptions provided by the German and FICO dataset, a partition of the features Q is manually constructed. For details see Appendix A.

The results can be seen in tables 5, 6, 7 and 8. Overall UDACE seems to produce the actions with the lowest average Mahalanobis distance, however, the utility of these actions also seems lower compared to the other methods.

The execution time has increased compared to tables (1 - 4), which was to be expected given that we attempt to enumerate all non-inferior solutions, before using the selection algorithm which picks a single solution. In particular, for UDACE on the FICO dataset, no actions could be obtained within the 600 second time limit. Hence we perform an additional experiment where we estimate a Random forest with 25 decision trees ($T = 25$), and we also restrict the number of discrete action candidates to 25 per feature instead of 100. This way the total computation time can be reduced drastically (Table 9).

Comparing UDACE to the other methods in Table 9, we observe that it appears to give slightly better CEs. Since the average MD distance and 10-LOF are lower, while the utility is higher.

We also observe that in general, compared to tables (1 - 4), the MD distance and 10-LOF have increased. This finding can be explained by the fact that the lowest MD distance and 10-LOF can only be obtained at the outer edge of the Pareto-front. But this same edge is likely to correspond with very low utility values. If does were not the case, then it would be likely that an ideal point exists, that simultaneously optimizes both objectives.

Overall there does not appear to be a cost function which always outperforms the other cost functions. It could be argued that each cost function has its strengths and that a careful choice must be made based on the properties of the dataset at hand.

Table 5

Results for the logistic regression model on the German dataset using the formulation with utility ($N = 50,\ D = 61$).

	Logistic Regression			
	$d_M(\bar{x}, \bar{x}+\alpha)\vert\Sigma^{-1})$	$q_{10}(\bar{x}, \bar{x}+\alpha\vert X_+)$	$u(\alpha)$	Time[s]
UMAD	6.81 (5.85)	1.23 (0.23)	7.89 (2.7)	1.68 (1.31)
UTLPS	6.19 (5.72)	1.23 (0.21)	8.65 (2.56)	2.2 (1.7)
UPCC	8.85 (6.47)	1.24 (0.24)	8.05 (2.93)	0.78 (0.35)
UDACE	2.32 (1.64)	1.26 (0.27)	8.17 (2.44)	462.42 (182.51)

5.2.1 Counterfactual exploration

To get a better picture of the generated CEs, we plot all efficient solutions by UTLPS, UMAD, UPCC and UDACE for a single applicant for both the Logistic regression model and Random forest model in a plot (Figure 1). The suggested action using the selection algorithm has a cross marker.

Table 6

Results for the logistic regression model on the FICO dataset using the formulation with utility ($N = 50,\ D = 23$).

	Logistic Regression			
	$d_M(\bar{x}, \bar{x}+\alpha)\lvert\Sigma^{-1})$	$q_{10}(\bar{x}, \bar{x}+\alpha\lvert X_+)$	$u(\alpha)$	Time[s]
UMAD	9.37 (4.03)	1.56 (1.67)	13.59 (5.52)	0.55 (0.28)
UTLPS	7.71 (4.14)	1.26 (0.37)	15.18 (6.92)	0.93 (1.21)
UPCC	10.19 (3.24)	1.59 (1.7)	14.75 (5.81)	0.17 (0.07)
UDACE	3.36 (0.99)	1.05 (0.06)	9.11 (3.41)	31.51 (1.48)

Table 7

Results for the Random forest model on the German dataset using the formulation with utility ($N = 6,\ D = 61,\ T = 100$) .

	Random forest			
	$d_M(\bar{x}, \bar{x}+\alpha)\lvert\Sigma^{-1})$	$q_{10}(\bar{x}, \bar{x}+\alpha\lvert X_+)$	$u(\alpha)$	Time[s]
UMAD	7.34 (3.6)	2.56 (3.31)	10.01 (3.72)	29.3 (17.82)
UTLPS	4.53 (3.11)	2.66 (3.26)	9.74 (5.01)	30.1 (8.51)
UPCC	10.88 (2.43)	2.6 (3.29)	15.0 (8.67)	11.83 (8.46)
UDACE	3.25 (1.43)	0.99 (0.06)	8.25 (3.42)	283.83 (204.51)

Table 8

Results for the Random forest model with 100 decision trees, on the FICO dataset using the formulation with utility ($N = 50,\ D = 23,\ T = 100$).

	Random forest			
	$d_M(\bar{x}, \bar{x}+\alpha)\lvert\Sigma^{-1})$	$q_{10}(\bar{x}, \bar{x}+\alpha\lvert X_+)$	$u(\alpha)$	Time[s]
UMAD	1.61 (1.15)	1.23 (0.18)	6.18 (2.77)	369.82 (217.13)
UTLPS	1.25 (0.69)	1.24 (0.28)	6.19 (2.32)	323.11 (216.04)
UPCC	2.47 (1.34)	1.23 (0.23)	7.38 (3.0)	412.03 (174.03)
UDACE	-	-	-	-

Note. None of the results for the UDACE method could not be produced within the 600 second time limit.

Table 9

Results for the Random forest model with 25 decision trees, on the FICO dataset using the formulation with utility ($N = 50,\ D = 23,\ T = 25$).

	Random forest			
	$d_M(\bar{x}, \bar{x}+\alpha)\lvert\Sigma^{-1})$	$q_{10}(\bar{x}, \bar{x}+\alpha\lvert X_+)$	$u(\alpha)$	Time[s]
UMAD	2.12 (1.5)	1.29 (0.31)	7.85 (2.48)	41.87 (38.38)
UTLPS	1.85 (1.26)	1.27 (0.29)	7.41 (2.81)	35.43 (32.15)
UPCC	2.79 (1.87)	1.24 (0.23)	7.66 (2.27)	31.68 (31.28)
UDACE	1.74 (1.11)	1.24 (0.25)	8.13 (2.50)	405.42 (179.6)

Note. To allow for relatively fast computation time we imposed the restriction that $|A_d| \leq 25$, for $d = 1, \dots D$

Actions which require low amounts of effort, and yield high utility are preferable. As is clear from Figure 1, there is a trade-off present between these two factors.

Notably From Figure 1b, the actions produced by the UTLPS and UDACE methods seem to align. While the actions proposed by UMAD seem to have the highest variation.

Looking at Figure 1a, the suggested action of UTLPS is superior to the suggested action by

UDACE. As can be seen from the fact that the action of UTLPS has a lower value of effort, but a higher value of utility. Also the UPCC method in Figure 1a, appears to have a very steep Pareto-front compared to the other methods. Since the additional effort required to achieve higher levels of utility is very low.

5.2.2 Number of CEs

We plot the distribution of the number of Pareto-efficient CEs found for each applicant by each of the methods. The distribution for the Logistic regression model on the German dataset can be seen in Figure 2.

The distribution of UDACE appears to be quite different compared to UMAD, UTLPS and UPCC. For UDACE the majority of applicants have exactly three Pareto-efficient CEs.

In Figure 3 the distribution of the Pareto-efficient actions by the Random Forest model on the FICO dataset can be seen. The distributions of the methods seem quite similar, which is a desirable property since it implies that there is less dependence on the choice of cost function.

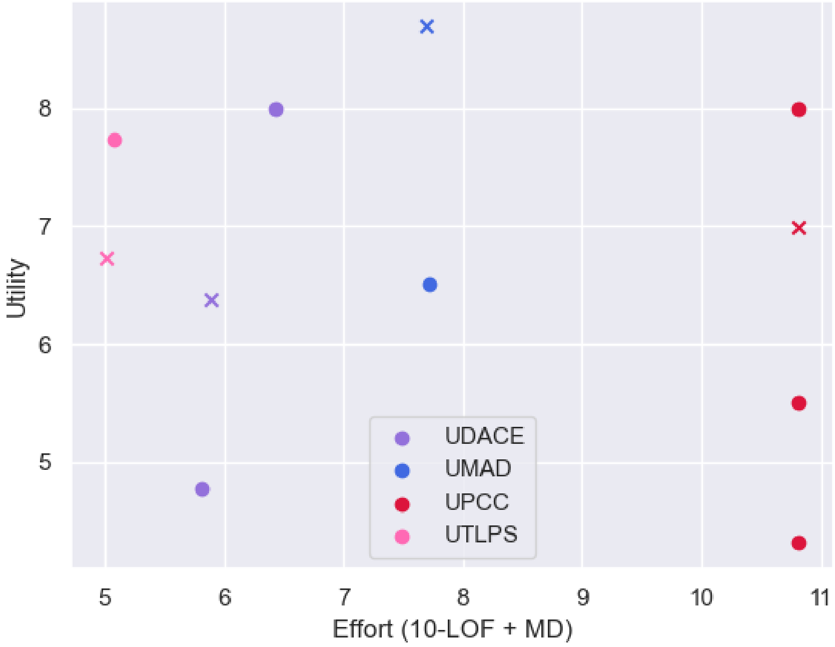
A distribution with less variance would be desirable since it would imply that all applicants receive a similar number of potential CEs. A large set of Pareto-efficient CEs is not necessarily beneficial. Since it could complicate the choice for an applicant. However, having a set which is too small can limit the choice too severely.

5.2.3 Illustration of a counterfactual explanation

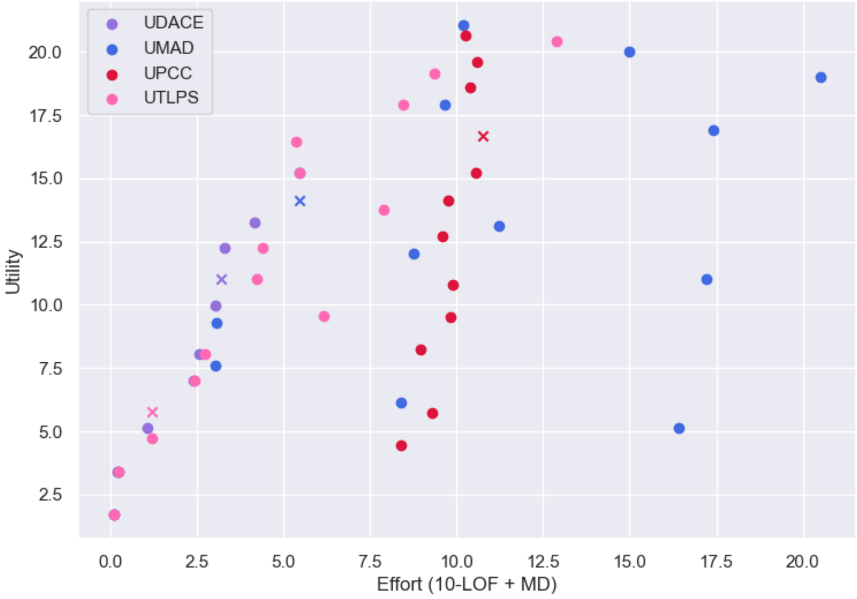
To illustrate how the utility affects the generated CEs. We compare a CE generated by DACE against the suggested CE of UDACE. The comparison is performed for a random applicant in the German Dataset. The results of the comparison can be seen in Figure 4. The biggest differences between the CE generated by DACE and UDACE appear to be in the amount of credit and the age of the applicant. In particular, the CE of DACE suggests drastically lowering the credit and waiting for four years before re-applying for the loan.

On the contrary, the CE of UDACE suggests that the applicant does not have to age, and can reduce the amount of credit in a less drastic fashion. The difference between DACE and UDACE can be explained by the preferences of this applicant. Since this applicant prefers changes to her income and loan application over changes to her personal traits like age. The UDACE method seems to be capable of accommodating these preferences. However to achieve this the UDACE method does suggest an income increase and a decrease of the credit usage at the bank which will provide the loan.

Figure 1. A Figure containing the pareto-efficient solutions for all different cost functions for a single applicant. Subfigure (a) corresponds to the logistic regression model evaluated on the FICO dataset.Subfigure (b) corresponds to the random forest model evaluated on the German credit dataset. The cross marker indicates the efficient action which was obtained using the selection algorithm.



(a)



(b)

Figure 2. The distribution of the number of pareto-efficient solutions found for all applicants, using the Logistic Regression model on the German dataset.

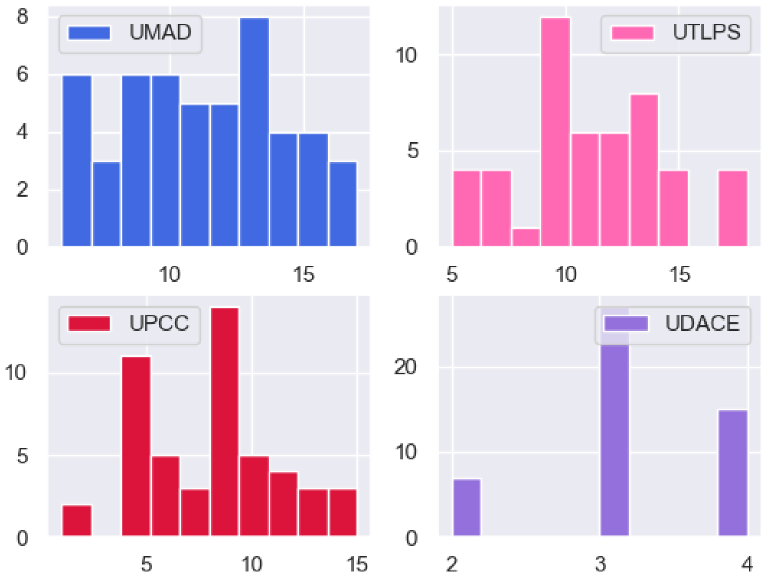


Figure 3. The distribution of the number of pareto-efficient solutions found for all applicants, using a Random forest model with 25 decision trees on the FICO dataset.

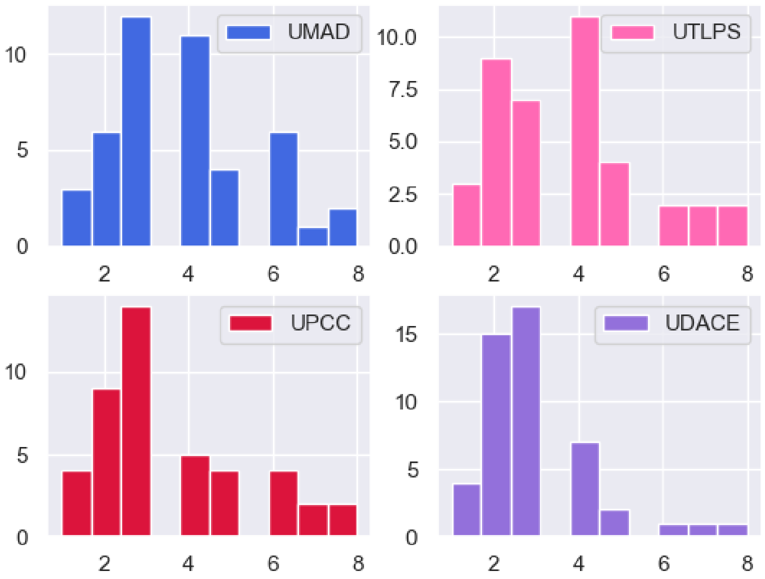
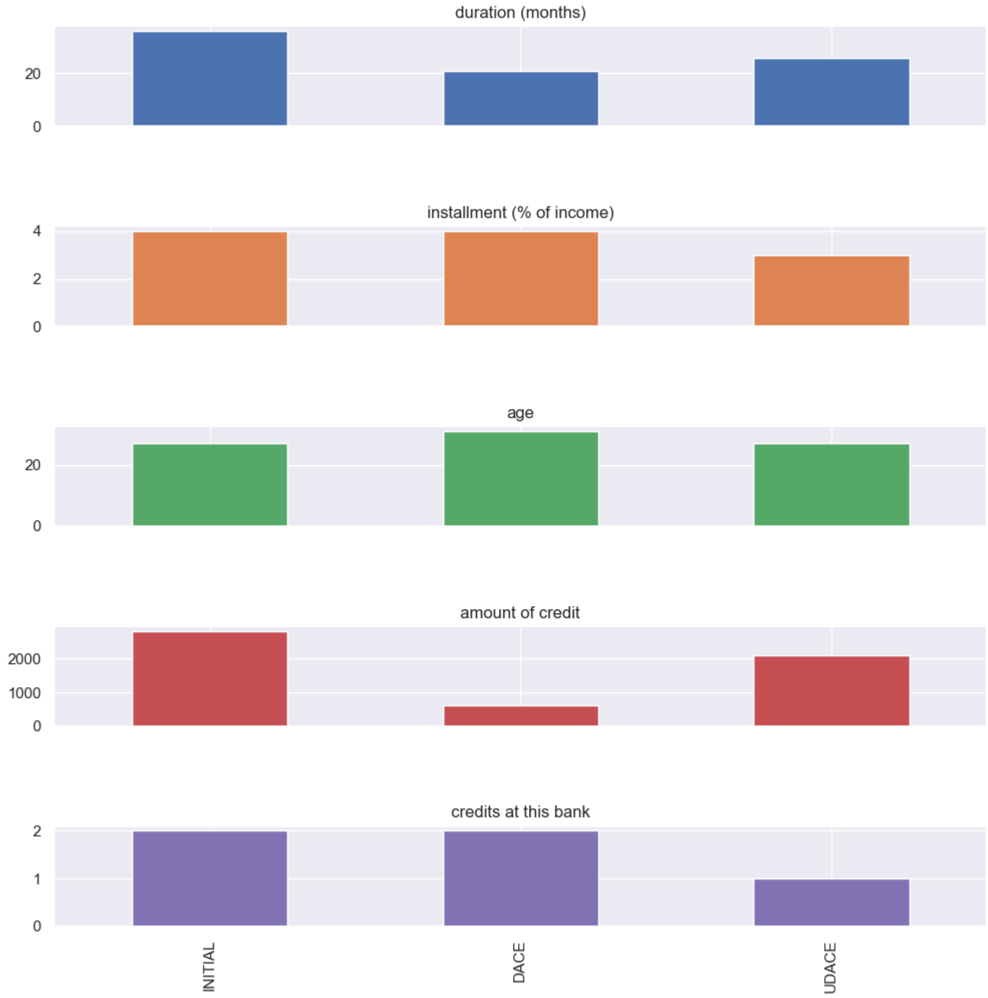


Figure 4. A figure showing the Initial state, CE of DACE and CE of UDACE, for an applicant part of the German dataset. The CE's were generated using the logistic regression model.



6 Conclusion

In this study, we extended the DACE method proposed by Kanamori et al. (2020), with the preferences of applicants. To this end, an algorithm which can simulate the preferences of applicants was developed. The output was subsequently used as an input for a Bayesian Bradley-Terry model to estimate utilities. We have shown that multiple Pareto-efficient CEs can be extracted, by extending the DACE formulation into a multi-objective optimization problem. Additionally, we have shown that using the selection algorithm of Wang et al. (2017), a single CE can be extracted which strikes a balance between effort and utility. This answers our research question: *How can counterfactual explanations take stakeholder preferences into account for personalized recourse?*

Our study possesses certain limitations that should be acknowledged. Firstly, it is worth noting that the process of acquiring CEs for certain datasets, particularly when utilizing the UDACE method, is time-consuming. This temporal constraint has the potential to impede the practical implementation and adoption of UDACE. Consequently, an essential avenue for future investigations would involve the development of more efficient formulations to expedite this process. Specifically, exploring alternative measures beyond the Local Outlier Factor (LOF) for the purpose of encompassing outlier risk could yield promising outcomes.

An additional constraint inherent in our study is the absence of an exhaustive comparison encompassing various cost functions, such as employing k-fold cross-validation as a means to evaluate their performance. The adoption of such an approach would prove advantageous, as we have noted the results are sensitive, to the choice of the validation sample. Consequently, incorporating this evaluation methodology would offer valuable insights into the robustness and generalizability of our findings.

Furthermore, it is essential to acknowledge another limitation in our study, namely the absence of a comparative analysis between the UDACE method and alternative counterfactual explanation (CE) techniques that yield a diverse set of counterfactual instances. By neglecting this comparative aspect, our study does not provide a comprehensive assessment of the UDACE method's peformance in relation to other CE methods.

Several promising directions for future research have emerged from our study. Firstly, one fruitful avenue entails conducting field experiments to empirically examine the tangible benefits experienced by applicants when given the chance to share their preferences. Additionally, the use of field experiments would enable a systematic assessment of various cost functions, showing which measure effectively captures the inherent effort exerted by applicants.

Secondly, future research endeavors can focus on the development of heuristic approaches that efficiently retrieve the optimal solution suggested by the selection algorithm, without exhaustively enumerating the entire non-dominated set. Lastly, an important area for future research involves incorporating the preferences of multiple stakeholders, including both the institution and the applicants, into the decision-making framework. By integrating diverse perspectives, this inclusive approach has the potential to enhance the recourse process.

7 Acknowledgements

I would like to thank Bezanson, Edelman, Karpinski and Shah (2017) who contributed to the creation of the open-source Julia ecosystem, which was critical for my research.

Appendix A

In order to test our method we developed a partition of the features for both the FICO and German credit datasets. In practice, this could be done through latent topic modelling if descriptions of the features are given, or manually by a domain expert.

For the FICO dataset, we partition the features into five distinct categories. These categories are labelled: Bad payment behaviour, lending frequency, usage of other credit lines, credit usage and good payment behaviour.

For the German dataset, all features are partitioned into six categories. These are Employment, Relationships, Wealth, Debts, Loan application, and Personal traits.

To see which features belong to which categories we refer to the included category specification with the source code of the paper.

Appendix B

The code written for our research can be decomposed into three separate parts which build on top of each other. The first part is the module which deals with preference simulation, estimation of the Bradley-Terry model and storing the preferences in an easy-to-use CSV file. This part can be found in the preferences subfolder of the source code.

The second part is comprised of several modules which together have the objective of formulating the MOMIP formulation and solving the problem. Examples of modules in this part are the ActionCandidate module, which deals with the generation of the feasible action space $\mathcal{A}$ and the MahalanobisDistance module which contains several functions used to compute the MD distance and construct the interaction matrix Σ. Several other modules exist which each deal with a small part of the formulation. This part consists of several subfolders.

The third part is the ParetoSelection module which contains an implementation of the algorithm by Wang et al. (2017). This module is called by the CE methods once the Pareto-efficient actions have been identified. This part can be found within the actions subfolder.

The entry point of our code is contained within the UtilDace module. This module contains a detailed description and example code which can be used to reproduce the results of our research. [3]

References

Ballet, V., Renard, X., Aigrain, J., Laugel, T., Frossard, P. & Detyniecki, M. (2019). Imperceptible adversarial attacks on tabular data. *arXiv preprint arXiv:1911.03274*. Retrieved from https://arxiv.org/abs/1911.03274 doi: 10.48550/arXiv.1911.03274

Becker, G. S. (2010). *The economics of discrimination*. University of Chicago press.

Bezanson, J., Edelman, A., Karpinski, S. & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, *59*(1), 65–98. Retrieved from https://epubs.siam.org/doi/10.1137/141000671 doi: 10.1137/141000671

[3] For more details on our implementation we refer to the included zip file with our source code.

Bradley, R. A. & Terry, M. E. (1952). Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, *39*(3/4), 324–345. Retrieved from https://www.jstor.org/stable/2334029 doi: 10.2307/2334029

Breunig, M. M., Kriegel, H.-P., Ng, R. T. & Sander, J. (2000). Lof: identifying density-based local outliers. In *Proceedings of the 2000 acm sigmod international conference on management of data* (pp. 93–104). Retrieved from https://dl.acm.org/doi/10.1145/335191.335388 doi: 10.1145/335191.335388

Caron, F. & Doucet, A. (2012). Efficient bayesian inference for generalized bradley–terry models. *Journal of Computational and Graphical Statistics*, *21*(1), 174–196. Retrieved from https://arxiv.org/abs/1011.1761 doi: 10.48550/arXiv.1011.1761

Chandra, M. P. et al. (1936). On the generalised distance in statistics. In *Proceedings of the national institute of sciences of india* (Vol. 2, pp. 49–55).

Chen, Y., Wang, J. & Liu, Y. (2020). Strategic recourse in linear classification. *arXiv preprint arXiv:2011.00355*, *236*.

Cui, Z., Chen, W., He, Y. & Chen, Y. (2015). Optimal action extraction for random forests and boosted trees. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining* (pp. 179–188). Retrieved from https://dl.acm.org/doi/10.1145/2783258.2783281 doi: 10.1145/2783258.2783281

Dandl, S., Molnar, C., Binder, M. & Bischl, B. (2020). Multi-objective counterfactual explanations. In *Parallel problem solving from nature–ppsn xvi: 16th international conference, ppsn 2020, leiden, the netherlands, september 5-9, 2020, proceedings, part i* (pp. 448–469). Retrieved from https://link.springer.com/chapter/10.1007/978-3-030-58112-1_31 doi: 10.1007/978-3-030-58112-1_31

Downs, M., Chu, J., Yacoby, Y., Doshi-Velez, F. & WeiWei, P. (2020). Cruds: Counterfactual recourse using disentangled subspaces. *ICML Workshop on Human Interpretability in Machine Learning*, 1-23.

Dua, D. & Graff, C. (2017). *UCI machine learning repository.* Retrieved from http://archive.ics.uci.edu/ml

FICO, Google, Imperial College London, MIT, University of Oxford, UC Irvine & UC Berkeley. (2018). *Explainable machine learning challenge.*

Ge, H., Xu, K. & Ghahramani, Z. (2018). Turing: a language for flexible probabilistic inference. In *International conference on artificial intelligence and statistics, AISTATS 2018, 9-11 april 2018, playa blanca, lanzarote, canary islands, spain* (pp. 1682–1690). Retrieved from http://proceedings.mlr.press/v84/ge18b.html

Hastie, T., Tibshirani, R., Friedman, J. H. & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2). Springer. Retrieved from https://link.springer.com/book/10.1007/978-0-387-84858-7

Huangfu, Q. & Hall, J. J. (2018). Parallelizing the dual revised simplex method. *Mathematical Programming Computation*, *10*(1), 119–142. Retrieved from https://arxiv.org/abs/1503.01889 doi: 10.48550/arXiv.1503.01889

Kanamori, K., Takagi, T., Kobayashi, K. & Arimura, H. (2020, 7). Dace: Distribution-aware counterfactual explanation by mixed-integer linear optimization. In C. Bessiere (Ed.), *Proceedings of the twenty-ninth international joint conference on artificial intelligence, IJCAI-20* (pp. 2855–2862). International Joint Conferences on Artificial Intelligence Organization. Retrieved from https://doi.org/10.24963/ijcai.2020/395 (Main track) doi: 10.24963/ijcai.2020/395

Kirlik, G. & Sayın, S. (2014). A new algorithm for generating all nondominated solutions of multiobjective discrete optimization problems. *European Journal of Operational Research*, *232*(3), 479–488. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0377221713006474 doi: 10.1016/j.ejor.2013.08.001

Lubin, M., Dowson, O., Garcia, J. D., Huchette, J., Legat, B. & Vielma, J. P. (2023). Jump 1.0: Recent improvements to a modeling language for mathematical optimization. *Mathematical Programming Computation*. Retrieved from https://arxiv.org/abs/2206.03866 (In press.) doi: 10.48550/arXiv.2206.03866

Mahajan, D., Tan, C. & Sharma, A. (2019). Preserving causal constraints in counterfactual explanations for machine learning classifiers. *arXiv preprint arXiv:1912.03277*. Retrieved from https://arxiv.org/abs/1912.03277 doi: 10.48550/arXiv.1912.03277

Marshall, A. (2009). *Principles of economics: unabridged eighth edition*. Cosimo, Inc.

Mothilal, R. K., Sharma, A. & Tan, C. (2020). Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency* (pp. 607–617). Retrieved from https://arxiv.org/abs/1905.07697 doi: 10.1145/3351095.3372850

Rawal, K. & Lakkaraju, H. (2020). Beyond individualized recourse: Interpretable and interactive summaries of actionable recourses. *Advances in Neural Information Processing Systems*, *33*, 12187–12198. Retrieved from https://arxiv.org/abs/2009.07165 doi: 10.48550/arXiv.2009.07165

Robinson, J. (1962). *Economic philosophy* (Vol. 1). CA Watts.

Samuelson, P. A. (1948). Consumption theory in terms of revealed preference. *Economica*, *15*(60), 243–253. Retrieved from https://www.jstor.org/stable/2549561 doi: 10.2307/2549561

Tsirtsis, S. & Gomez Rodriguez, M. (2020). Decisions, counterfactual explanations and strategic behavior. *Advances in Neural Information Processing Systems*, *33*, 16749–16760. Retrieved from https://arxiv.org/abs/2002.04333 doi: 10.48550/arXiv.2002.04333

Ustun, B., Spangher, A. & Liu, Y. (2019). Actionable recourse in linear classification. In *Proceedings of the conference on fairness, accountability, and transparency* (pp. 10–19). Retrieved from https://arxiv.org/abs/1809.06514 doi: 10.1145/3287560.3287566

Verma, S., Dickerson, J. & Hines, K. (2020). Counterfactual explanations for machine learning: A review. *arXiv preprint arXiv:2010.10596*. Retrieved from https://arxiv.org/abs/2010.10596 doi: 10.48550/arXiv.2010.10596

Wachter, S., Mittelstadt, B. & Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harvard Journal of Law & Technology*, *31*, 841. Retrieved from https://arxiv.org/abs/1711.00399 doi: 10.48550/arXiv.1711.00399

Wang, N., Zhao, W.-j., Wu, N. & Wu, D. (2017). Multi-objective optimization: a method for selecting the optimal solution from pareto non-inferior solutions. *Expert Systems with Applications*, *74*, 96–104. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0957417417300040 doi: https://doi.org/10.1016/j.eswa.2017.01.004

Wicksteed, P. H. (1906). Pareto. manuale di economia politica, con una introduzione alla scienza sociale. *The Economic Journal*, *16*(64), 553–557.