

Prototype-based Counterfactual Explanation for Causal Classification

Tri Dung Duong¹, Qian Li¹, Guandong Xu^{1*}

¹ University of Technology Sydney

TriDung.Duong@student.uts.edu.au, {Qian.Li, Guandong.Xu}@uts.edu.au

Abstract

Counterfactual explanation is one branch of interpretable machine learning that produces a perturbation sample to change the model’s original decision. The generated samples can act as a recommendation for end-users to achieve their desired outputs. Most of the current counterfactual explanation approaches are the gradient-based method, which can only optimize the differentiable loss functions with continuous variables. Accordingly, the gradient-free methods are proposed to handle the categorical variables, which however present several major limitations: 1) causal relationships among features are typically ignored when generating the counterfactuals, possibly resulting in impractical guidelines for decision-makers; 2) the generation of the counterfactual sample is prohibitively slow and requires lots of parameter tuning for combining different loss functions. In this work, we propose a causal structure model to preserve the causal relationship underlying the features of the counterfactual. In addition, we design a novel gradient-free optimization based on the multi-objective genetic algorithm that generates the counterfactual explanations for the mixed-type of continuous and categorical data. Numerical experiments demonstrate that our method compares favorably with state-of-the-art methods and therefore is applicable to any prediction model. All the source code and data are available at <https://github.com/tridungduong16/multiobj-scm-cf>.

1 Introduction

Machine learning is increasingly recognized as an effective approach for large-scale automated decisions in several domains. However, when ML model is deployed in critical decision-making scenarios such as medical and financial domains, many people are skeptical about its accountability and reliability. Hence, interpretable ML is vital to make machine learning models transparent and understandable by humans.

Counterfactual explanation (CE) is the prominent example-based method in interpretable machine learning that generates counterfactual samples for interpreting machine learning model decisions. For example, consider a customer A whose loan application has been rejected by the ML model of a bank. Counterfactual explanations can generate a “what-if” scenario of this person, e.g., “your loan would have been approved if your income was \$5,000 more”. Namely, the goal of counterfactual explanation is to generate perturbations of an input that lead to a different outcome from the ML model. By allowing users to explore such “what-if” scenarios, counterfactual examples are human-interpretable.

Despite recent interests in counterfactual explanations, existing methods suffer three limitations: First, the counterfactual methods neglect the causal relationship among features, leading to the infeasible counterfactual samples for end-users [Ustun *et al.*, 2019; Poyiadzi *et al.*, 2020]. A counterfactual sample is feasible if the changes satisfy constraints entailed by the causal relations. For example, since the education causes the choice of the occupation, changing the occupation without changing the education is infeasible for the loan applicant in the real-world. Namely, the generated counterfactuals need to preserve the causal relations between features in order to be realistic and actionable. Second, on the algorithm level, most counterfactual methods use the gradient-free optimization algorithm to deal with various data and model types [Sharma *et al.*, 2020; Poyiadzi *et al.*, 2020; Dhurandhar *et al.*, 2019; Grath *et al.*, 2018; Lash *et al.*, 2017]. These gradient-free optimizations rely on the heuristic search, which however suffers from inefficiency due to the large heuristic search space. In addition, optimizing the trade-off among different loss terms in the objective function is difficult, which often leads to sub-optimal counterfactual samples [Mahajan *et al.*, 2019; Mothilal *et al.*, 2020; Grath *et al.*, 2018].

To address the above limitations, we propose a prototype-based counterfactual explanation framework (ProCE) in this paper. ProCE is a model-agnostic method and is capable of explaining the classification in the mixed feature space. Overall, our contributions are summarized as follows:

- By integrating the structural causal model and causal loss function, our proposed method can produce the counterfactual samples that satisfy the causal constraints among features.

*Corresponding author

- We utilize the auto-encoder model and class prototype to guide the search progress and speed up the searching speed of counterfactual samples.
- We design a novel multi-objective optimization that can find the optimal trade-off between the objectives meanwhile maintain diversity in the feature space of counterfactual explanations.

2 Related Work

Counterfactual explanation is the example-based approach which is a branch of model-agnostic techniques. Recently, there has been an increasing number of studies in this field.

On the one hand, [Wachter *et al.*, 2017] first proposes using the counterfactual explanation to interpret machine learning models’ decision. Particularly, they generate the counterfactual samples by minimizing the loss between the desired class and the counterfactual instances’ prediction. To extend the previous study [Wachter *et al.*, 2017], another framework called DiCE [Mothilal *et al.*, 2020] proposes using the diversity score to enhance the number of generated samples. They thereafter use the weighted sum to combine different loss functions together and adopt the gradient-descent algorithm to approximately find the optimal solution. This method is however restricted to the differentiable models, and finds it hard to deal with the non-continuous values in tabular data.

On the other hand, CERTIFAI [Sharma *et al.*, 2020] is a recent gradient-free based approach that customizes the genetic algorithm for the counterfactuals search progress. When dealing with the categorical features, CERTIFAI adopts the indicator functions (1 for different values, else 0). [Poyiadzi *et al.*, 2020] introduces a method called FACE that adopts Dijkstra’s algorithm to generate counterfactual samples by finding the shortest path of the original input and the existing data points. The generated samples of this method are limited to the input space without generating new data. Meanwhile, [Mahajan *et al.*, 2019] builds the generative model based on the variational auto-encoder (VAE) to generate multiple counterfactual samples for all input data points. The research [Van Looveren and Klaise, 2019] utilizes the class prototype to guide the search progress to fall into the distribution of the expected class. This method however does not consider the causal relationship among features.

Finally, there are also some other recent methods [Russell, 2019; Kanamori *et al.*, 2020] that use linear programming, mixed-integer programming or solvers to deal with the objective optimization effectively. These approaches can be applied to linear models only. Our method extends the line of studies [Van Looveren and Klaise, 2019; Mahajan *et al.*, 2019] by integrating both structural causal model and class prototype. We also formulate the problem as the multi-objective optimization problem and propose an algorithm to find the counterfactual samples effectively.

3 Methodology

In this section, we firstly present the objective functions that can generate the counterfactual samples. The structural causal model and causal distance are also investigated to exploit the underlying causal relationship among features.

Then, we formulate the counterfactual sample generation according by defining the loss functions as a multi-objective optimization problem, and propose an algorithm based on the non-dominated sorting genetic algorithm (NSGA-II) to effectively find the optimal solution.

To begin with, we consider a classifier $h : \mathcal{X} \rightarrow \mathcal{Y}$, with the input of D -dimensional feature space $\mathcal{X} = \mathcal{X}^1 \dots \mathcal{X}^D \subseteq \mathbb{R}^D$ and the output as $\mathcal{Y} = \{0, 1\}$. Let a vector $x = (x^1, \dots, x^D) \in \mathcal{X}$ be an instance and x^k be a feature k -th of x .

Definition 1 (Counterfactual Explanation) *With the original instance $x_0 = (x_0^1, \dots, x_0^D) \in \mathcal{X}$, and original prediction $y_0 \in \mathcal{Y}$, the counterfactual explanation aims to find the nearest counterfactual sample x_{cf} such that the outcome of classifier for x_{cf} changes to desired output class y_{cf} . In general, the counterfactual explanation x_{cf} for the individual x_0 is the solution of the following optimization problem:*

$$x_{cf}^* = \arg \min_{x_{cf} \in \mathcal{X}} f(x_{cf}) \quad \text{subject to} \quad h(x_{cf}^*) = y_{cf} \quad (1)$$

where f is the distance metric between x_0 and x_{cf} . For such explanations to be plausible, they should only suggest small changes in a few features.

3.1 Prototype-based Causal Model

Counterfactuals provide these explanations in the form of “if these features had different values, your credit application would have been accepted”. This indicates that counterfactual samples should be constrained. We first provide detailed definitions of each constraint and further tie them together as a multi-objective optimization problem.

Prediction Loss

In order to achieve the desired outcome, the basic loss term is to calculate the distance between the counterfactual prediction and the expected outcome. For the classification scenario, we particularly use the cross-entropy loss to minimize the counterfactual label and desired label. Specifically, the prediction loss is:

$$f_{\text{pred}}(x_{cf}) = -y_{cf} \log(h(x_{cf})) + (1 - y_{cf}) \log(1 - h(x_{cf})) \quad (2)$$

Prototype-based Loss

The counterfactuals search is incredibly slow, due to the enormous solutions in search space. Inspired by the study [Van Looveren and Klaise, 2019], we utilize the class prototype to guide the search progress toward the counterfactual solution. From the concept view, prototype is defined as the representative of the whole or subset of the data. For each class i in the dataset, we first compute the k nearest neighbors of x_0 . To compute the distance, we resort to an encoder function parametrized by ϕ is denoted by Q_ϕ with $Q_\phi : \mathcal{X} \rightarrow \mathcal{Z}$. This encoder projects the input feature \mathcal{X} to the E -dimensional latent space $\mathcal{Z} \subseteq \mathbb{R}^E$. Then, k nearest neighbors of x_0 can be computed based on the latent distance

in the projected space \mathcal{Z} , i.e., $\|Q_\phi(x_k^i) - Q_\phi(x_0)\|_2^2$. Finally, the prototype is computed by the mean of these neighbors:

$$\text{proto}_i = \frac{1}{K} \sum_{k=1}^K Q_\phi(x_k^i) \quad (3)$$

In the latent \mathcal{Z} space, we define the prototype loss function as

$$f_{\text{proto}}(x_{cf}) = \|Q_\phi(x_{cf}) - \text{proto}_j\|_2^2 \quad (4)$$

Note that proto_j is the prototype of class j that has smallest distance to the encoding of x_0 . Given y_0 be the label for the input sample x_0 , we have j as

$$j = \arg \min_{i \neq y_0} \|Q_\phi(x_0) - \text{proto}_i\|_2^2 \quad (5)$$

Proximity Loss

In general, the counterfactual samples should be as close to the original instance as possible to make it more useful and understandable by users. When it comes to the mixed-type tabular data which contains both the categorical and continuous features, it is challenging to define the loss function. The previous studies [Sharma *et al.*, 2020; Mothilal *et al.*, 2020; Dandl *et al.*, 2020] normally apply the indicator function that returns 1 when two categorical values match and return 0 otherwise, and adopt L_2 -norm distance for continuous features. However, the indicator function fails to produce the distance degree for categorical values. In this study, we use the encoder model Q_ϕ to map the categorical features into the latent space before estimating the distance. The main advantage of this approach is that the encoder model has the capability to capture the underlying relationship and pattern between each categorical value. This means that manual feature engineering such as assigning weight for each category is not necessary, thus reducing a great deal of time and effort. The distance between two instances is

$$f_{\text{dist}}(x_{cf}, x_0) = \begin{cases} \|x_{cf}^k - x_0^k\|_2^2, & \text{if } x^k \text{ is continuous} \\ \|Q_\phi(x_{cf}^k) - Q_\phi(x_0^k)\|_2^2, & \text{if } x^k \text{ is categorical} \end{cases} \quad (6)$$

Causality-preserving Loss

Although the distance function in Eq. (6) demonstrates the closeness distance between two instances, it fails to capture the causal relationship between each feature. To deal with this problem, we integrate the structural causal model, and thus construct the causal loss function to ensure the features' causal relationship in generated samples.

In general, structural causal model [Pearl, 2009] consists of two main components: the causal graph and structural equations. A causal graph is the probabilistic graphical model representing the assumption about data generating mechanism. A causal graph is defined as $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ where \mathcal{V} is the set of nodes and \mathcal{E} is the set of edges. Structural equation is a set of equations representing the causal effect illustrated by the edge in the causal graph. We classify the set of variables into two node groups including:

- U as a set of exogenous variables are independent from other models' variable.
- V as a set of endogenous variables are determined by its relationship with other variables within the model.

We consider a setting that the structural causal model is provided along with the observational data. For each endogenous node $v \in V$, and its parent nodes $(v_{p1}, v_{p2}, \dots, v_{pk})$, we construct the structural causal equation $v = g(v_{p1}, v_{p2}, \dots, v_{pk})$ to represent their causal relationship. During the counterfactuals generation progress, we firstly produce the predicted value of endogenous node x^v based on their parents before estimating the distance, which is measured as:

$$\begin{aligned} f_{\text{causal}}(x_{cf}^v, x_0^v) &= \|x_{cf}^v - x_0^v\|_2^2 \\ &= \|g(x_{cf}^{vp1}, x_{cf}^{vp2}, \dots, x_{cf}^{vpk}) - x_0^v\|_2^2 \end{aligned} \quad (7)$$

Based on the Eq. (6) and Eq. (7), we come up with the distance between the original and counterfactual instance is the sum of distance of endogenous variables and exogenous variables, measured as:

$$f_{\text{final_dist}}(x_{cf}) = \sum_u f_{\text{dist}}(x_{cf}^u, x_0^u) + \sum_v f_{\text{causal}}(x_{cf}^v, x_0^v) \quad (8)$$

3.2 Multi-objective Optimization

With the loss functions from the sections 3.1 including f_{pred} , f_{proto} , $f_{\text{final_dist}}$, the majority of existing studies [Mahajan *et al.*, 2019; Mothilal *et al.*, 2020; Grath *et al.*, 2018] uses the trade-off parameter sum assigning each loss function a weight, and combines them together. However, it is very challenging to balance the weights for each loss, resulting in a great deal of efforts and time into hyperparameter tuning. To address this issue, we propose to formulate the counterfactual explanation search as the multi-objective problem (MOP) as

$$x_{cf}^* = \arg \min_{x_{cf} \in \mathcal{X}} \{f_{\text{pred}}(x_{cf}), f_{\text{proto}}(x_{cf}), f_{\text{final_dist}}(x_{cf})\} \quad (9)$$

In this study, we modify the elitist non-dominated sorting genetic algorithm (NSGA-II) [Deb *et al.*, 2000] to deal with this optimization problem. Its main superiority is to optimize each loss function simultaneously as well as provide the solutions presenting the trade-offs among objective functions. We first present some definitions.

Definition 2 (Dominance in the Objective Space) *In the multi-objective optimization problem, the goodness of a solution is evaluated by the dominance [Deb et al., 2002]. Given two solutions x and \hat{x} along with a set of m objective functions f_i , we have:*

- x weakly dominates \hat{x} ($x \succeq \hat{x}$) iff $f_i(x) \geq f_i(\hat{x}) \forall i \in 1, \dots, m$
- x dominates \hat{x} ($x \succ \hat{x}$) iff $x \succeq \hat{x}$ and $x \neq \hat{x}$

Definition 3 (Pareto Front) *Pareto front [Ngatchou et al., 2005] is the set of solutions that are non-dominated by each other but are superior to other solutions in the objective space. Pareto front is denoted as \mathcal{F} .*

Definition 4 (Crowding Distance) *To maintain the diversity of the candidate solutions in the population, one of the simplest methods is to choose the individuals having the low density. Therefore, the crowding distance [Raquel and Naval Jr, 2005] is used to rank each candidate solution. The crowding distance between two individuals x and y is measured:*

$$d_{xy} = \sqrt{\sum_{i=1}^M \left(\frac{f_i(x) - f_i(y)}{f_i^{\min} - f_i^{\max}} \right)^2} \quad (10)$$

with f_i is the i -th objective function, f_i^{\min} or f_i^{\max} is its minimum or maximum value.

The optimization process for objective function (9) is given by Algorithm 1. The main idea behinds this approach is that for each generation, the algorithm chooses the non-dominated solutions for each objective function and evolves to the better ones. We firstly find the nearest class prototype of the original instance x_0 , which is used to measure the prototype loss function later. For the optimal counterfactual x_{cf}^* finding progress, each candidate solution is represented by the D -dimensional feature as the genes. A random candidate population is initialized with the Gaussian distribution. Thereafter, the objective functions including f_{pred} , f_{proto} , $f_{\text{final_dist}}$ are calculated for each candidate. In the non-dominated sorting step, all the non-dominated solutions are selected from the population and are assigned to the Pareto front \mathcal{F}_1 . After that, the non-dominated solutions are chosen from the remaining population. The process is repeated until all the solutions are assigned to a front.

The crowding distance function in Eq. (10) then is adopted to select the individuals for the current population with the purpose of maintaining the population diversity. The algorithm then only keeps the candidate solutions having the greatest ranking score. The cross-over and mutation procedure [Whitley, 1994] are finally performed to generate the next population. Particularly, the cross-over of two parents generates the new candidate solutions by randomly swapping parts of genes. Meanwhile, the mutation procedure randomly alters some genes in the candidate solutions to encourage diversity and avoid local minimum. We repeat this process through many generations to find the optimal counterfactual solution.

4 Experiments

We conduct experiments on three datasets to prove the effectiveness of our proposed method by comparing with other existing methods. All implementations are conducted in Python 3.7.7 with 64-bit Red Hat, Intel(R) Xeon(R) Gold 6150 CPU @ 2.70GHz. For our proposed method, we construct the multi-objective optimization algorithm with the support of library Pymoo¹ [Blank and Deb, 2020].

¹<https://pymoo.org/algorithms/nsga2.html>

Algorithm 1 Multi-objective Optimization for Prototype-based Counterfactual Explanation (ProCE)

Input: An instance x_0 and label y_0 , desired class y_{cf} , and a provided machine learning classifier h , encoder model Q_ϕ .

- 1: Evaluate prototypes for each class by Eq. (3).
- 2: Compute proto_j by Eq. (5).
- 3: Initialize a batch of population $P = \{\Delta_1, \dots, \Delta_m\}$ with $\Delta_i \sim \mathcal{N}(\mu, \nu)$
- 4: $Q = \emptyset$
- 5: **for** G generation **do**
- 6: $P = P \cup Q$
- 7: **for** $k = 1, \dots, m$ **do**
- 8: Compute $f_{\text{pred}}(\Delta_k)$ based on Eq. (2).
- 9: Use proto_j to compute $f_{\text{proto}}(\Delta_k)$ based on Eq. (4).
- 10: Compute $f_{\text{final_dist}}(\Delta_k)$ based on Eq. (8).
- 11: **end for**
- 12: Compute $\mathcal{F} = \text{non-dominated-sorting}(P)$
- 13: $P = \emptyset$
- 14: **while** $|P| + |\mathcal{F}_i| < m$ **do**
- 15: $P = P \cup \mathcal{F}_i$
- 16: $i = i + 1$
- 17: **end while**
- 18: Compute ranking score for P based on Eq. (10).
- 19: Keep n individual in P based on ranking score.
- 20: Randomly pair $\lceil m/2 \rceil \{\Delta_1, \Delta_2\} \in P$
- 21: **for** each pair $\{\Delta_1, \Delta_2\}$ **do**
- 22: Perform crossover $(\Delta_1, \Delta_2) \rightarrow \Delta'_1, \Delta'_2$
- 23: Perform mutation $\Delta'_1 \rightarrow \tilde{\Delta}_1, \Delta'_2 \rightarrow \tilde{\Delta}_2$
- 24: $Q = Q \cup \{\tilde{\Delta}_1, \tilde{\Delta}_2\}$
- 25: **end for**
- 26: **end for**

Output: $x_{cf} = \Delta^*$

4.1 Datasets

Qian ▶ *experimental settings are missing, like cross-validation, training/test ratio* ◀ This section provides information about the datasets, on which we perform the experiments. To prove our method effectiveness in generating counterfactual samples that maintaining the causal relationship, for each dataset, we consider some feature conditions that a generated sample has to satisfy. To simplify, we denote $a \propto b$ meaning the condition that (a increase $\Rightarrow b$ increase) AND (a decrease $\Rightarrow b$ decrease). The datasets used include:

Simple-BN [Mahajan et al., 2019] is a synthetic dataset containing 10,000 records with three features (x_1, x_2, x_3) and a binary output (y). We consider the causal relationship $(x_1, x_2) \propto x_3$.

Sangiovese [Magrini et al., 2017] dataset ² evaluates the impact of several agronomic settings on the quality of the Tuscan grapes. It has 14 continuous features along with the binary output representing the grapes' quality. The conditional linear Bayesian network is also provided within the dataset. We consider a causal relationship $\text{BunchN} \propto \text{SproutN}$.

Adult [Dua and Graff, 2017] dataset ³ is the real-world dataset providing information about people applying for loan in the financial organization. This dataset consists of both

²<https://www.bnlearn.com/bnrepository/clgaussian-small.html>

³<https://archive.ics.uci.edu/ml/datasets/adult>

continuous features and categorical features. The main task is to determine whether a person has an income exceeding \$50k dollars annually. Similar to the study [Mahajan *et al.*, 2019], we consider two conditions:

$$x_{cf}^{\text{age}} \geq x_0^{\text{age}} \text{ and } \text{age} \propto \text{education}$$

4.2 Evaluation Metrics

In this section, we briefly describe six quantitative metrics that are used to evaluate the performance of our proposed method and baselines.

Target-class validity measures the percentage of the counterfactual samples belonging to the desired class, evaluating how well the algorithm can produce valid samples.

Causal-constraint validity measures the percentage of counterfactual samples satisfying the pre-defined causal conditions. With this metric, the main aim is to evaluate how well our algorithm can generate feasible counterfactual samples that do not violate the causal relationship among features [Mahajan *et al.*, 2019].

Categorical proximity measures the proximity for categorical features representing the total number of matches on the categorical value between x_{cf} and x_0 . Higher categorical proximity is better, implying that the counterfactual sample preserves the minimal changes from the original [Mothilal *et al.*, 2020].

Continuous proximity illustrates the proximity of the continuous features, which is calculated as the l_2 -distance between the continuous features in x_{cf} and x_0 . Lower continuous proximity is preferable, implying that the distance between the continuous features of x_0 and x_{cf} should be as close as possible [Mothilal *et al.*, 2020].

IM1 and IM2 are two interpretability metric (IM) proposed in [Van Looveren and Klaise, 2019]. Let the original class be y_0 and the counterfactual class be y_{cf} . AE_0 , AE_{cf} and AE are the auto-encoder model trained specifically on instances of class y_0 , instances of class y_{cf} and the full dataset, respectively. IM1 measures the ratio of reconstruction errors of x_{cf} using AE_{cf} and AE_0 , while IM2 evaluates the similarity between the reconstructed instance using AE_{cf} and AE . Lower values for both IM1 and IM2 are preferable, implying that the generated counterfactual is more interpretable.

4.3 Baseline Methods

We compare our proposed method (ProCE) with two baselines, namely CERTIFAI and DiCE. To the best of our knowledge, there is not much similar work in this area and CERTIFAI and DiCE are two state-of-the-art and prominent approaches in the counterfactual explanation.

- **DiCE** [Mothilal *et al.*, 2020]. DiCE is the popular counterfactual explanation framework. This calculates the weighted sum of different loss functions including proximity, diversity and sparsity together, and approximately finds the optimal solution via gradient-descent algorithm. For implementation, we utilize the provided source code ⁴ from the authors with their default settings.

- **CERTIFAI** [Sharma *et al.*, 2020]. CERTIFAI is the latest study, which constructs the counterfactuals search approach based on the genetic algorithm. Since there is no available source code for this method, we implement the algorithm in Python with the support from the library PyGAD ⁵.

For all the experiments, we build a machine learning classifier h by a neural network with three hidden layers and the sigmoid function on the last layer. For the feature engineering, we normalize the continuous feature to range (0,1) and transform the categorical features by using the label encoder.

4.4 Results and Discussions

In this section, we firstly report the experimental results of different methods across three datasets to prove our proposed method’s effectiveness. Then, we present the variation of proposed method performance with different auto-encoder models embedding sizes and different numbers of k -nearest instances in the class prototype finding.

Table 1 illustrates the target-class validity and causal-constraint validity of all methods across three datasets. In terms of target-class validity, all three methods perform well, except the CERTIFAI performance in Adult dataset with only 60%. Regarding the percentage of samples satisfying the causal constraints, by far the greatest performance is achieved by ProCE with 86.67%, 80% and 93.33% for Simple-BN, Sangiovese and Adult dataset, respectively. CERTIFAI is ranked the second across three datasets in terms of this metric, while the majority of generated samples from DiCE violate the causal constraints. These results suggest that by integrating the structural causal model, our proposed method can effectively produce the counterfactual samples preserving the features’ causal relationships. Meanwhile, interpretability scores (IM1 and IM2) are shown in the Table 2. In general, our proposed method achieved the best IM1 and IM2 in three datasets. DiCE also produces a very competitive result in Adult dataset, whereas there is a good performance in Simple-BN and Sangiovese for CERTIFAI.

Table 1: Baseline results in terms of **Target-class validity** (Tcv%) and **Causal-constraint validity** (Ccv%)

Method	Dataset	%Tcv	%Ccv
CERTIFAI	Simple-BN	100.00	43.33
DiCE	Simple-BN	100.00	36.67
ProCE	Simple-BN	100.0	86.67
CERTIFAI	Sangiovese	100.00	50.00
DiCE	Sangiovese	100.00	36.67
ProCE	Sangiovese	100.00	80.00
CERTIFAI	Adult	60.00	85.70
DiCE	Adult	100.00	75.00
ProCE	Adult	100.00	93.33

Figure 1 provides information about the categorical proximity in the Adult dataset and continuous proximity in three datasets. For the categorical feature proximity, ProCE achieves an average of 5 out of the total 6 categorical features

⁴<https://github.com/divyat09/cf-feasibility>

⁵<https://github.com/ahmedfgad/GeneticAlgorithmPython>

Table 2: Baseline results in terms of **IM1** and **IM2** with 95% confidence bound

Method	Dataset	IM1	IM2 (x10)
CERTIFAI	Simple-BN	0.045 ± 0.003	0.040 ± 0.014
DiCE	Simple-BN	0.066 ± 0.004	0.070 ± 0.030
ProCE	Simple-BN	0.024 ± 0.002	0.020 ± 0.031
CERTIFAI	Sangiovese	0.217 ± 0.002	0.090 ± 0.012
DiCE	Sangiovese	0.200 ± 0.003	0.090 ± 0.032
ProCE	Sangiovese	0.189 ± 0.000	0.040 ± 0.022
CERTIFAI	Adult	0.600 ± 0.030	0.510 ± 0.021
DiCE	Adult	0.365 ± 0.050	0.160 ± 0.025
ProCE	Adult	0.099 ± 0.040	0.070 ± 0.015

in the dataset, whereas the lowest result is recorded in the CERTIFAI algorithm. These results illustrate that with the gradient-free based approach, we can achieve an outstanding performance when handling the non-continuous features in tabular data. When it comes to the continuous feature proximity, ProCE produces the counterfactual sample with the smallest distance from continuous features. The most significant variation is also seen in the CERTIFAI, whereas our proposed method produces the least variation in continuous proximity.

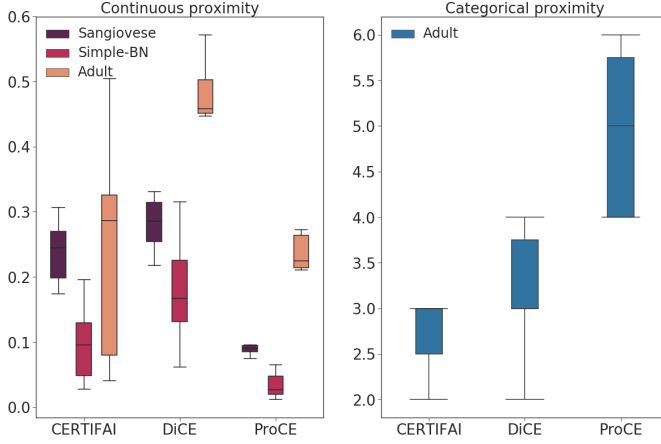


Figure 1: Baseline results in terms of **continuous proximity** and **categorical proximity**. Lower continuous proximity is better and higher categorical proximity is better.

Figure 2 and 3 show the variation of our method’s performance with the different numbers of nearest neighbors for class prototype and the embedding size of auto-encoder model, respectively. It is clear from Figure 3 that although there are some fluctuations in all four metrics, the performance nearly reaches stable when the embedding size is larger than 32. On the other hand, as can be seen from Figure 2, IM1 and IM2 otherwise the worst performance when the number of instances is 15, followed by a stagnant performance with from 25 to 45 instances. Meanwhile, there is no significant fluctuation in the performance of continuous and categorical proximity across three datasets. These results suggest that the performance of our proposed method in all evaluation metrics is nearly stable with different embedding

sizes and numbers of nearest neighbors, possibly implying the robustness of our method.

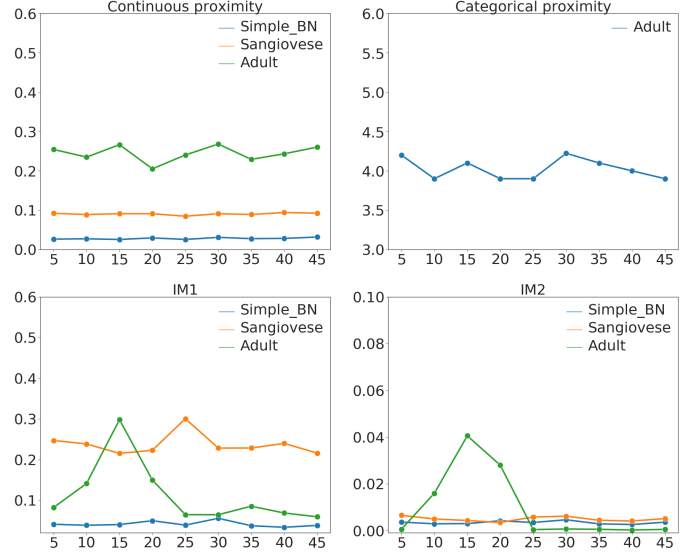


Figure 2: Our performance under different numbers of k -nearest neighbors for class prototype (proto_j with $j = 1$).

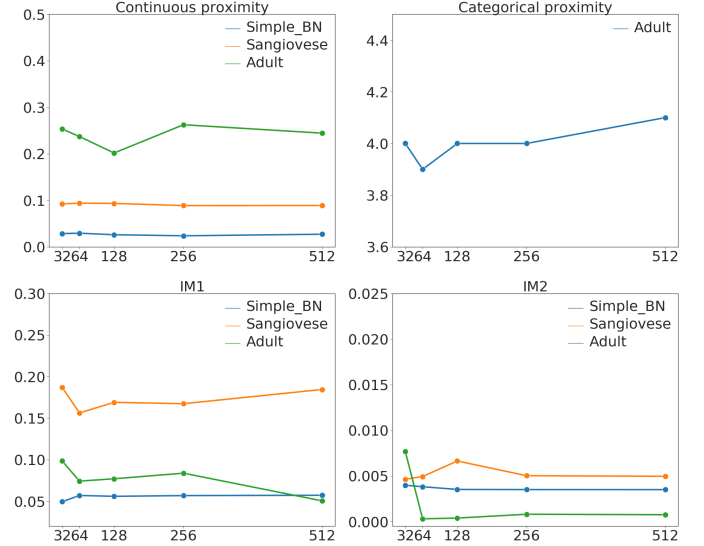


Figure 3: Our performance under different sizes of E -dimensional embedding for encoder function Q_ϕ .

5 Conclusion

This paper introduced a novel counterfactual explanation algorithm by integrating the structural causal model and class prototype. We also proposed formulating the counterfactual generation as a multi-objective problem and construct an optimization algorithm to find the optimal solution effectively. Our experiments proved that our method surpasses other existing methods in many evaluation metrics. For future work,

we plan to consider the imperfect structural causal model that is very commonplace in real-world scenarios. Other multi-objective optimization algorithms such as reinforcement learning and multi-task learning are also worthy of investigating.

References

- [Blank and Deb, 2020] J. Blank and K. Deb. Pymoo: Multi-objective optimization in python. *IEEE Access*, 8:89497–89509, 2020.
- [Dandl *et al.*, 2020] Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. Multi-objective counterfactual explanations. *arXiv preprint arXiv:2004.11165*, 2020.
- [Deb *et al.*, 2000] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and Tanaka Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *International conference on parallel problem solving from nature*, pages 849–858. Springer, 2000.
- [Deb *et al.*, 2002] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.
- [Dhurandhar *et al.*, 2019] Amit Dhurandhar, Tejaswini Pedapati, Avinash Balakrishnan, Pin-Yu Chen, Karthikeyan Shanmugam, and Ruchir Puri. Model agnostic contrastive explanations for structured data. *arXiv preprint arXiv:1906.00117*, 2019.
- [Dua and Graff, 2017] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.
- [Grath *et al.*, 2018] Rory Mc Grath, Luca Costabello, Chan Le Van, Paul Sweeney, Farbod Kamiab, Zhao Shen, and Freddy Lecue. Interpretable credit application predictions with counterfactual explanations. *arXiv preprint arXiv:1811.05245*, 2018.
- [Kanamori *et al.*, 2020] Kentaro Kanamori, Takuya Takagi, Ken Kobayashi, and Hiroki Arimura. Dace: Distribution-aware counterfactual explanation by mixed-integer linear optimization. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, Christian Bessiere (Ed.). International Joint Conferences on Artificial Intelligence Organization*, pages 2855–2862, 2020.
- [Lash *et al.*, 2017] Michael T Lash, Qihang Lin, Nick Street, Jennifer G Robinson, and Jeffrey Ohlmann. Generalized inverse classification. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 162–170. SIAM, 2017.
- [Magrini *et al.*, 2017] Alessandro Magrini, Stefano Di Blasi, and Federico Mattia Stefanini. A conditional linear gaussian network to assess the impact of several agronomic settings on the quality of tuscan sangiovese grapes. *Biometrical Letters*, 54(1):25–42, 2017.
- [Mahajan *et al.*, 2019] Divyat Mahajan, Chenhao Tan, and Amit Sharma. Preserving causal constraints in counterfactual explanations for machine learning classifiers. *arXiv preprint arXiv:1912.03277*, 2019.
- [Mothilal *et al.*, 2020] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 607–617, 2020.
- [Ngatchou *et al.*, 2005] Patrick Ngatchou, Anahita Zarei, and A El-Sharkawi. Pareto multi objective optimization. In *Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems*, pages 84–91. IEEE, 2005.
- [Pearl, 2009] Judea Pearl. Causal inference in statistics: An overview. *Statistics Surveys*, 3:96–146, 01 2009.
- [Poyiadzi *et al.*, 2020] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. Face: feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 344–350, 2020.
- [Raquel and Naval Jr, 2005] Carlo R Raquel and Prospero C Naval Jr. An effective use of crowding distance in multiobjective particle swarm optimization. In *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, pages 257–264, 2005.
- [Russell, 2019] Chris Russell. Efficient search for diverse coherent explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 20–28, 2019.
- [Sharma *et al.*, 2020] Shubham Sharma, Jette Henderson, and Joydeep Ghosh. Certifai: A common framework to provide explanations and analyse the fairness and robustness of black-box models. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 166–172, 2020.
- [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*, pages 10–19, 2019.
- [Van Looveren and Klaise, 2019] Arnaud Van Looveren and Janis Klaise. Interpretable counterfactual explanations guided by prototypes. *arXiv preprint arXiv:1907.02584*, 2019.
- [Wachter *et al.*, 2017] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.
- [Whitley, 1994] Darrell Whitley. A genetic algorithm tutorial. *Statistics and computing*, 4(2):65–85, 1994.