A Two-Stage Algorithm for Cost-Efficient Multi-instance Counterfactual Explanations

André Artelt[1,2][0000−0002−2426−3126] and Andreas Gregoriades[3][0000−0002−7422−1514]

[1] Bielefeld University, Germany
[2] University of Cyprus, Cyprus
[3] Cyprus University of Technology, Cyprus
aartelt@techfak.uni-bielefeld.de, andreas.gregoriades@cut.ac.cy

Abstract. Counterfactual explanations constitute among the most popular methods for analyzing the predictions of black-box systems since they can recommend cost-efficient and actionable changes to the input to turn an undesired system's output into a desired output. While most of the existing counterfactual methods explain a single instance, several real-world use cases, such as customer satisfaction, require the identification of a single counterfactual that can satisfy multiple instances (e.g. customers) simultaneously. In this work, we propose a flexible two-stage algorithm for finding groups of instances along with cost-efficient multi-instance counterfactual explanations. This is motivated by the fact that in most previous works the aspect of finding such groups is not addressed.

Keywords: XAI · Counterfactual Explanations · Multi-instance Counterfactuals.

1 Introduction

Recently an increasing number of Artificial Intelligence (AI-) and Machine Learning (ML-) based systems have been applied to real-world problems [33]. Although these systems show impressive performance, they are still imperfect when applied to real-world problems and in some cases can cause harm to human users. Therefore, transparency of such AI- and ML-based systems is now of paramount importance and a necessary property before they can be fully deployed. Transparency can assist developers in understanding the logic of such systems and thus assist in validating them before deployment. Moreover, transparency creates trust and assists decision-makers in understanding where and how it is safe to deploy them. The importance of transparency is also recognized by EU policymakers, with recent regulations such as the GDPR [20] and the AI act [7] making explicit reference to the need for explainability of such systems. One of the most popular ways to achieve transparency is through explanations of systems' logic. This gave rise to the field of eXplainable AI (XAI) [8]. However, because explanations are highly user- and situation-dependent, many different explanation methods have been developed over the past few years [8, 1, 23]. One

of the most popular types of explanation methods are counterfactual explanations [29]. This approach mimics the way humans seek explanations [5] which makes them a favorite approach in different scenarios. By definition, a counterfactual explanation states actionable recommendations on how to change a predictive system's output in some desired way – e.g. how to change a rejected loan application into an accepted one.

In many real-world problems, such as customers' repurchase intentions, employee attrition, etc. the decision maker is not only interested in explaining a single instance (e.g finding what needs to be changed so that a specific employee does not quit their job - attrition example) but a group of instances (how to prevent many employees from quitting their jobs in the same organization). Thus, there is a need to find a single explanation that can satisfy (i.e. holds for) a group of instances. To address such use cases, the concept of multi-instance counterfactual explanations has been recently introduced, with an example of recent work being [11, 30]. In the multi-instance problem, the aim is to identify a single explanation that proposes actionable recommendations on how to change the system's output for a group of instances simultaneously. Because of the novelty of this concept, many issues still exist – in particular, how to identify groups of instances for which cost-efficient multi-instance counterfactual explanations can be computed.

Our contributions: In this work, we formalize and investigate the problem of finding groups of instances for which cost-efficient multi-instance counterfactual explanations can be computed. Based on our formal analysis, we propose a model- and data-agnostic two-stage algorithm for computing such multi-instance counterfactual explanations. This work also proposes an evolutionary algorithm for computing multi-instance counterfactual explanations for a given set of instances.

2 Foundations

A counterfactual explanation (often just called counterfactual) proposes actionable changes to the features of a given input instance of a predictive system so that the system's prediction changes to the desired output. A typical example is stating how to turn a rejected loan application into an accepted one. Usually, an explanation is requested in the case of an unexpected or unfavorable outcomes [26] (e.g. rejected loan application) – in the latter case, a counterfactual is also referred to as recourse [12], i.e. recommendations on how to change an unfavorable into a favorable outcome. Because counterfactuals mimic human explanations [5], they constitute among the most popular explanation methods in the XAI literature and a favorable choice in practical problems [18, 28].

The formalization and computation of counterfactual explanations (see Fig. 1a for an illustration) involves the consideration of two important aspects [29]: 1) the contrasting property, which requires that the stated changes indeed alter the output of the system, and 2) the cost of the counterfactual, which define the difficulty and effort it takes to execute the explanation (i.e. recommendations)

in the real world. Both properties have been combined [29] into a single objective optimization problem as stated in Definition 1.

Definition 1 (Counterfactual Explanation). *Assume a prediction function $h : \mathcal{X} \to \mathcal{Y}$ is given. Computing a counterfactual explanation $\boldsymbol{\delta}_{cf} \in \mathcal{X}$ for a given instance $\boldsymbol{x}_{orig} \in \mathcal{X}$ is phrased as the following optimization problem:*

$$\underset{\boldsymbol{\delta}_{cf} \in \mathcal{X}}{\arg\min}\ \ell\left(h(\boldsymbol{x}_{orig} \oplus \boldsymbol{\delta}_{cf}), y_{cf}\right) + C \cdot \theta(\boldsymbol{\delta}_{cf}) \tag{1}$$

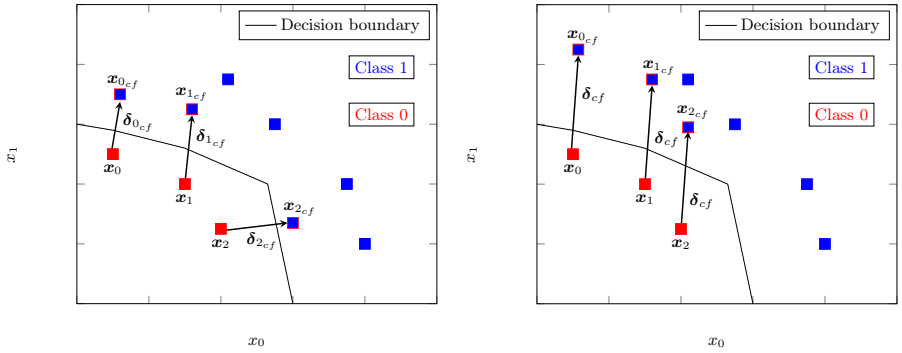
where $\ell(\cdot)$ denotes a loss function that penalizes deviation of the output $h(\boldsymbol{x}_{orig} \oplus \boldsymbol{\delta}_{cf})$ from the requested output y_{cf}, $\theta(\cdot)$ implements the cost of $\boldsymbol{\delta}_{cf}$ – i.e. prefer "simple, cheap & easy to execute" explanations –, and $C > 0$ denotes the regularization strength.

In order to not make any assumptions on the data domain, we use the symbol $\oplus$ to denote the application/execution of the counterfactual $\boldsymbol{\delta}_{\text{cf}}$ to the original instance $\boldsymbol{x}_{\text{orig}}$. While in the case of real and integer numbers (e.g. $\mathcal{X} = \mathbb{R}^d$) this reduces to the translation (i.e. $(\boldsymbol{x}_{\text{cf}})_i = (\boldsymbol{x}_{\text{orig}})_i + (\boldsymbol{\delta}_{\text{cf}})_i$, in the case of categorical features it denotes a substitution – i.e. $(\boldsymbol{x}_{\text{cf}})_i = (\boldsymbol{\delta}_{\text{cf}})_i$.

Also note that the cost of the counterfactual, here modeled by $\theta(\cdot)$, is highly domain and use-case specific. In many implementations & toolboxes [9], the p-norm is used as a default. Besides those two essential properties (contrasting and cost), there exist additional relevant aspects such as plausibility [17, 22], diversity [19], robustness [3, 27], fairness [2, 15], etc. which have been addressed in literature [9]. However, the basic formalization Definition 1 is still very popular and widely used in practice [28, 9].

Most existing counterfactual explanation methods focus on providing a single recommendation or multiple diverse recommendations [19] for a given single instance only (see Definition 1) – but they do not address the case where we have to assign the same actions to multiple instances simultaneously [11]. However, there are first attempts of multi-instance counterfactual explanations (also called group counterfactuals) that aim to cover (i.e. explain) more than a single instance within a single counterfactual [16, 11] – e.g. what to do in order to get a set of rejected loan applications accepted.

In contrast to Definition 1, a multi-instance counterfactual states a single change $\boldsymbol{\delta}_{\text{cf}} \in \mathbb{R}^d$ that changes the output of a classifier $h : \mathbb{R}^d \to \mathcal{Y}$ for many instances $\boldsymbol{x}_i \in \mathbb{R}^d$ simultaneously – see Fig. 1b for an illustration. While multi-instance counterfactuals are formalized slightly differently in different works [30, 16, 11, 4], existing work in literature agrees that the same two properties, as in the case of a standard counterfactual explanation (see Definition 1), must be considered: 1) The cost of the explanation $\boldsymbol{\delta}_{\text{cf}}$ (i.e. it should be simple and easy to execute) 2) contrasting property: the explanation δ should be valid for all (or at least as many as possible) instances in a given set of instances $\mathcal{D}$ – this extends the contrasting property from Definition 1 to multiple instances. In this work, we formalize a multi-instance counterfactuals explanation as a multi-objective optimization problem as stated in Definition 2.



(a) Illustration of standard counterfactual explanations: Each instance $\boldsymbol{x}_i$ gets a **different** counterfactual explanation $\boldsymbol{\delta}_{i_{cf}}$.

(b) Illustration of a multi-instance counterfactual: Each instance $\boldsymbol{x}_i$ gets the **same** counterfactual explanation $\boldsymbol{\delta}_{\text{cf}}$.

Definition 2 (Multi-instance Counterfactual Explanation). *Let $h : \mathcal{X} \to \mathcal{Y}$ denote a prediction function, and let $\mathcal{D}$ be a set of labeled instances with the same prediction $y \in \mathcal{Y}$ under $h(\cdot)$ – i.e. $h(\boldsymbol{x}_i) = y \quad \forall \boldsymbol{x}_i \in \mathcal{D}$. We are looking for a single change $\boldsymbol{\delta}_{cf} \in \mathbb{R}^d$ that, if applied to the instances in $\mathcal{D}$, changes as many of their predictions to some requested output $y_{cf} \in \mathcal{Y}$.*

We call all pareto-optimal solutions $\boldsymbol{\delta}_{cf}$ to the following multi-objective optimization problem multi-instance counterfactuals:

$$\min_{\boldsymbol{\delta}_{cf} \in \mathcal{X}} \left(\theta(\boldsymbol{\delta}_{cf}) \;,\; \ell(h(\boldsymbol{x}_i \oplus \boldsymbol{\delta}_{cf}), y_{cf}) \; \forall \boldsymbol{x}_i \in \mathcal{D} \right) \tag{2}$$

where $\theta(\cdot)$ denotes the cost of the counterfactual, and $\ell(\cdot)$ denotes a suitable loss function penalizing deviations from the requested outcome y_{cf} – suitable loss functions might be the mean-squared error or cross-entropy loss, while the cost $\theta(\cdot)$ might be implemented by a p-norm.

Note that the main difference to a standard counterfactual explanation Eq. (1) from Definition 1 is that the contrasting property leads to multiple objects (i.e. one objective for each instance in $\mathcal{D}$) – i.e. the change $\boldsymbol{\delta}_{\text{cf}}$ must be valid for all (or as many as possible) instances in $\mathcal{D}$.

2.1 Related Work

Since multi-instance counterfactual explanations are a novel concept, existing work on this is rather limited. One of the earliest work [11] proposes a counterfactual explanation tree, which assigns counterfactuals to the leaves in a decision tree derived from $\mathcal{D}$ – each leaf in this tree is interpreted as a group. This method solves both tasks (grouping of instances and computing multi-instance counterfactuals) in a single step. While this might be beneficial in some scenarios, it also constitutes a limitation since the user cannot customize the grouping and also lacks any formal guarantees due to the implementation as a heuristic (local

search). Also, restrictions on the supported data domains apply. The authors of [21] propose to generate a set of basis explanations that are used to construct explanations between groups whereby the groups are assumed to be given. In [4], multi-instance counterfactuals are implemented utilizing convex programming for linear classifiers only.

In general, a large part of existing work for multi-instance counterfactuals can be interpreted as summarizing or aggregating individual counterfactual explanations [30, 16, 24, 21]. For instance, in [30], multi-instance counterfactuals are generated by first computing individual counterfactuals and then applying a sampling strategy to select the one that maximizes the cover of a given set of instances for which a multi-instance counterfactual is requested. Similarly to [30], the authors of [16] compute global counterfactuals by introducing instance-specific scaling of a set of counterfactual explanations such that as many instances as possible are covered. However, those methods assume that a grouping is already given and also often suffer from poor performance (e.g. low coverage and correctness).

3 A Two-Stage Algorithm for Computing Multi-instance Counterfactual Explanations

As stated in Definition 2, a multi-instance counterfactual state changes $\boldsymbol{\delta}_{\text{cf}}$ that are valid for a set of instances $\mathcal{D}$. While in some scenarios, the $\mathcal{D}$ might be given apriori and thus be fixed, in other scenarios it might be more flexible and require finding groups along with cost-efficient multi-instance counterfactuals. For instance, business owners might be interested in identifying groups of customers along with recommendations on how to improve their repurchase intention.

In these cases, it is important to identify large groups of instances for which cost-efficient multi-instance counterfactuals (Definition 2) can be computed. We formalize this as a multi-objective optimization problem as stated in Problem 1.

Problem 1. For a classifier $h : \mathcal{X} \rightarrow \mathcal{Y}$ and a set of instances $\mathcal{D} \subset \mathcal{X}^n$ with $h(\boldsymbol{x}_i) = y \; \forall \boldsymbol{x}_i \in \mathcal{D}, y \in \mathcal{Y}$, we are looking for a partition of all instances such that cost-efficient multi-instance counterfactuals (Definition 2) exists:

$$\min N \tag{3a}$$

$$\max \left(|\mathcal{G}_i| \; \forall i, \dots N \right) \tag{3b}$$

$$\min \left(\theta(\boldsymbol{\delta}_{\text{cf}i}) \; \forall i, \dots N \right) \tag{3c}$$

$$\min \left(\ell(h(\boldsymbol{x}_j \oplus \boldsymbol{\delta}_{\text{cf}}), y_{\text{cf}}) \; \forall \boldsymbol{x}_j \in \mathcal{G}_i, \forall i, \dots N \right) \tag{3d}$$

$$\text{s.t.} \bigcup_i \mathcal{G}_i = \mathcal{D}, \quad \mathcal{G}_i \cap \mathcal{G}_j = \emptyset \; \forall i \neq j \tag{3e}$$

In the following sub-sections, we study Problem 1 and propose to split the process of computing multi-instance counterfactuals into two stages for performance and flexibility: 1) Finding a grouping of instances and then 2) Computing multi-instance counterfactual explanations for each group of those group – by

this, we aim to reduce the effect of outliers on the cost of the final multi-instance counterfactuals.

3.1 Grouping of Instances

A naive approach would be to group the instances based on their spatial similarity/distances – e.g. by using a clustering method such as k-means. However, because counterfactuals are known not to be robust with respect to changes in the input [3] (i.e. similar instances do not have similar counterfactuals), this approach is likely to fail. Also, such an approach does not take in any knowledge about the cost $\theta(\cdot)$, which is necessary to compute cost-efficient counterfactuals – we empirically confirm this in the experiments in Section 4.

Under some assumptions, interesting statements concerning Problem 1 can be made (see Lemma 1) when considering individual counterfactual explanations (Definition 1) instead of the instances in data space.

Lemma 1. *Assume a monotonic binary classifier* $h : \mathbb{R}^d \to \{0, 1\}$ *and* $\theta(\cdot) = \|\cdot\|_p$*. Furthermore, for a given set of instances* $\boldsymbol{x}_i \in \mathbb{R}^d$ *we denote their counterfactual explanation (Definition 1) as* $\boldsymbol{\delta}_{cf_i}$*. If* $\boldsymbol{\delta}_{cf_i}^\top \boldsymbol{\delta}_{cf_j} = \|\boldsymbol{\delta}_{cf_i}\|_2 \cdot \|\boldsymbol{\delta}_{cf_j}\|_2 \; \forall i \neq j$*, then the cost* $\theta(\cdot)$ *of the multi-instance counterfactual* $\boldsymbol{\delta}_{cf}$ *(Definition 2) is given as follows:*

$$\theta(\boldsymbol{\delta}_{cf}) = \max_i \theta(\boldsymbol{\delta}_{cf_i}) \tag{4}$$

Proof. Sketch: $\boldsymbol{\delta}_{\text{cf}_i}^\top \boldsymbol{\delta}_{\text{cf}j} = \|\boldsymbol{\delta}_{\text{cf}i}\|_2 \cdot \|\boldsymbol{\delta}_{\text{cf}j}\|_2 \; \forall i \neq j$ implies that $\exists \alpha_j \in \mathbb{R} : \boldsymbol{\delta}_{\text{cf}j} = \alpha_j \cdot \boldsymbol{\delta}_{\text{cf}i} \; \forall j$. Monotonicty of $h(\cdot)$ implies $\exists \alpha \in \mathbb{R} : \boldsymbol{\delta}_{\text{cf}j} = \alpha \cdot \boldsymbol{\delta}_{\text{cf}i} \; \forall j$. The statement follows from selecting α and $\boldsymbol{\delta}_{\text{cf}i}$□

Lemma 1 states that if the individual counterfactuals all have the same direction, then a multi-instance counterfactual not only exists but we can also state a tight upper bound on its cost. Although Lemma 1 is stated for a linear classifier, it can also be applied to arbitrary classifiers that can be approximated locally by a linear classifier. This suggests that groups of instances where the individual counterfactuals (Definition 1) point in similar directions are good candidates for which cost-efficient multi-instance counterfactuals (Definition 2) might exist. We, therefore, propose to 1) compute single counterfactuals (Definition 1) for each instance, and then 2) cluster those into groups based on their direction (i.e. based on their cosine similarity) – optionally, in addition, one could also cluster in a second step according to their amount of change if this is to be minimized as well. In the remainder of this work, we limit ourselves to minimizing the number of changes – i.e. we cluster only based on the direction of the individual counterfactuals. The number of groups (i.e. clusters) might be given by the user or might be determined automatically, e.g. using the Elbow method [14]. The complete procedure for finding groups of instances is described in Algorithm 1.

Algorithm 1 Grouping of Instances For Cost-Efficient Multi-instance Counterfactual Explanations

Input: Instances $\boldsymbol{x}_i$ with the same prediction $h(\boldsymbol{x}_i) = \boldsymbol{y}$, counterfactual generation method $\mathrm{CF}_h(\cdot)$

Output: Grouping of instances

1: $\{\boldsymbol{\delta}_{\mathrm{cf}i} = \mathrm{CF}_h(\boldsymbol{x}_i)\}$ ▷ Compute a counterfactual $\boldsymbol{\delta}_{\mathrm{cf}i}$ for each instance $\boldsymbol{x}_i$
2: **for** Different number of clusters **do** ▷ Optimize number of clusters if requested/needed
3: Cluster with $\mathrm{d}(\boldsymbol{\delta}_{\mathrm{cf}i}, \boldsymbol{\delta}_{\mathrm{cf}j}) = \frac{\boldsymbol{\delta}_{\mathrm{cf}i}^\top \boldsymbol{\delta}_{\mathrm{cf}j}}{\|\boldsymbol{\delta}_{\mathrm{cf}i}\|_2 \|\boldsymbol{\delta}_{\mathrm{cf}j}\|_2}$ ▷ Cluster based on the directions of $\boldsymbol{\delta}_{\mathrm{cf}i}$
4: Sub-cluster with $\mathrm{d}(\boldsymbol{\delta}_{\mathrm{cf}i}, \boldsymbol{\delta}_{\mathrm{cf}j}) = \|\theta(\boldsymbol{\delta}_{\mathrm{cf}i}) - \theta(\boldsymbol{\delta}_{\mathrm{cf}j})\|_2$ ▷ Sub-cluster based on the cost $\theta(\boldsymbol{\delta}_{\mathrm{cf}i})$
5: **end for**

3.2 Computing Multi-instance Counterfactuals

Given a group of instances, we can use any existing method from the literature for computing multi-instance counterfactuals (Definition 2). However, because existing model- and domain-agnostic methods are limited and often show sub-optimal performance with respect to correctness, we propose an evolutionary method for solving Eq. (2). This not only constitutes a model- and domain-agnostic method but also a very flexible solution since additional constraints can be easily added.

Our evolutionary method constitutes an instance of the $(\mu + \lambda)$ genetic algorithm [25] and can handle all types of variables (i.e. real, integer, and categorical).

In order to guarantee the feasibility of the final multi-instance counterfactual $\boldsymbol{\delta}_{\mathrm{cf}}$ in the data domain, we construct the set of feasible changes for each dimension of real or integer variables as follows – assuming non-negativity which can also be achieved by adding a constant:

$$l_i = \alpha_i - \min_j\{(\boldsymbol{x}_j)_i\} \quad u_i = \beta_i - \max_j\{(\boldsymbol{x}_j)_i\} \tag{5}$$

where α_i and β_i denote the maximum and minimum feasible value of the i-th feature/dimension, and the final set of feasible changes for the i-th feature/dimension is given as $[l_i, u_i]$. These sets/intervals are used in the initialization phase and when computing mutations of existing individuals during the optimization.

We merge the objectives Eq. (3d) and Eq. (3c) from Problem 1 into a single objective as follows:

$$\underset{\boldsymbol{\delta}_{\mathrm{cf}} \in \mathcal{X}}{\arg\min}\ \theta(\boldsymbol{\delta}_{\mathrm{cf}}) + C \cdot \sum_{\boldsymbol{x}_i \in \mathcal{D}} \ell(h(\boldsymbol{x}_j \oplus \boldsymbol{\delta}_{\mathrm{cf}}), y_{\mathrm{cf}}) \tag{6}$$

where $C > 0$ denotes a hyperparameter and the cost $\theta(\boldsymbol{\delta}_{\text{cf}})$ is defined as follows:

$$\theta(\boldsymbol{\delta}_{\text{cf}}) = \sum_i \psi((\boldsymbol{\delta}_{\text{cf}})_i) \text{ where } \psi((\boldsymbol{\delta}_{\text{cf}})_i) = \begin{cases} 1 & \text{if } i\text{-feature is categorical} \\ |(\boldsymbol{\delta}_{\text{cf}})_i| & \text{otherwise} \end{cases} \tag{7}$$

4 Experiments

All experiments are implemented in Python and are publicly available on GitHub[4].

4.1 Benchmark data sets

We consider 3 different data sets:

The *IBM human resource attrition dataset* [10] (*Attr.*) contains 35 features for 1467 unique employees. The dataset contains human resources properties such as age, education, promotion, education, rate, etc.

The *Credit card clients data set* [32] (*Credit*) contains 30000 data records of customers used for default payment prediction. Each record is described by 23 attributes (8 categorical, 14 numerical and 1 binary). The sensitive attribute is "sex" but we also remove "age" and "marriage".

The *Law school data set* [31] (*Law*) contains 20798 law school admission records. Each record (student) is described by 12 attributes (3 categorical, 3 binary and 6 numerical). The binary target attribute describes whether the student is admitted or not. The sensitive attributes "sex" and "race" are removed.

4.2 Setup

All experiments are run in a five-fold cross-validation. An XGBoost classifier [6] is fitted to the training set and all negative classified instances (i.e. $h(\boldsymbol{x}_i) = 0$) are selected to create the set of instances $\mathcal{D}$ to be explained by a multi-instance counterfactual.

We compute a multi-instance counterfactual for the entire set of selected instances $\mathcal{D}$, and also cluster the set $\mathcal{D}$ in two different ways: 1) Clustering based on the individual counterfactual explanations (Definition 1) as proposed in Algorithm 1, and for a comparison 2) clustering based on the individual instances $\boldsymbol{x}_i$.

For the clustering of instances and counterfactuals, we use the DBSCAN method [13] which determines the number of clusters automatically. For the clustering of the instances $\boldsymbol{x}_i$ we use the standard Euclidean distance, whereas we use the cosine-similarity (i.e. for comparing directions) for the clustering of the individual counterfactuals $\boldsymbol{\delta}_{\text{cf}i}$ as described in Algorithm 1.

For the computation of multi-instance counterfactuals (Definition 2) we consider three different methods: 1) Our proposed evolutionary algorithm (denoted

[4] https://github.com/andreArtelt/TwoStageMultiinstCFs

Data	Method	No Clustering ↑	Clustering $\boldsymbol{x}_i$ ↑	Clustering CFs ↑
Attr.	EA [Ours]	0.98 ± 0.0	0.95 ± 0.02	1.0 ± 0.0
	Warren et al. [30]	0.37 ± 0.02	0.34 ± 0.04	0.52 ± 0.06
	Kanamori et al. [11]	0.98 ± 0.0	0.95 ± 0.01	0.86 ± 0.05
Credit	EA [Ours]	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
	Warren et al. [30]	0.8 ± 0.0	0.78 ± 0.01	0.73 ± 0.02
	Kanamori et al. [11]	–	–	–
Law	EA [Ours]	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
	Warren et al. [30]	0.06 ± 0.0	0.09 ± 0.02	0.03 ± 0.01
	Kanamori et al. [11]	0.97 ± 0.0	0.99 ± 0.0	0.81 ± 0.05

Table 1: Correctness (in percentage) of the generated multi-instance counterfactuals. We report the mean and variance (over all folds) rounded to two decimal points – *larger* numbers are better.

by *EA*) from Section 3.2, 2) the method proposed by Warren et al. [30], and 3) the method proposed by Kanamori et al. [11].

We evaluate two properties of the computed multi-instance counterfactual explanations: 1) The correctness – i.e. evaluating for how many samples the explanation is correct. The results are shown in Table 1. 2) The cost $\theta(\cdot)$ by means of the number of changed features – the results are shown in Table 2.

4.3 Results & Discussion

The results show that the proposed evolutionary algorithm for computing multi-instance counterfactuals achieves an excellent performance (with respect to correctness and cost) across all settings. The method by Warren et al. often struggles to find correct multi-instance counterfactuals (i.e. counterfactuals that cover as many as possible instances), also their method almost always produces multi-instance counterfactuals that use all available features (which is due to the design of their method) and therefore have a higher cost when executing those in practice. The method by Kanamori et al. often achieves a competitive performance (except for the *Credit* data sets where it fails to compute any multi-instance counterfactuals[5]) and yields the most cost-efficient solutions while sacrificing correctness – however, this method automatically creates additional sub-groups and is therefore difficult to compare to the other methods. Furthermore, we observe that our proposed clustering of individual counterfactuals (see Algorithm 1) often improves the correctness as well as the cost (i.e. complexity) of the computed multi-instance counterfactuals significantly – in particular in the case of our proposed evolutionary method for computing multi-instance counterfactuals.

Together, these results demonstrate the high efficiency of our proposed two-stage algorithm.

[5] This might be fixed by a data set specific hyper-parameter tuning.

Data	Method	No Clustering ↓	Clustering $\boldsymbol{x}_i$ ↓	Clustering CFs ↓
Attr.	EA [Ours]	0.73 ± 0.02	0.69 ± 0.02	0.55 ± 0.04
	Warren et al. [30]	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
	Kanamori et al. [11]	0.02 ± 0.0	0.06 ± 0.02	0.05 ± 0.01
Credit	EA [Ours]	0.75 ± 0.0	0.78 ± 0.0	0.75 ± 0.0
	Warren et al. [30]	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
	Kanamori et al. [11]	–	–	–
Law	EA [Ours]	0.67 ± 0.01	0.67 ± 0.01	0.64 ± 0.01
	Warren et al. [30]	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
	Kanamori et al. [11]	0.11 ± 0.01	0.05 ± 0.01	0.14 ± 0.01

Table 2: Cost (percentage of changed features) of the generated multi-instance counterfactuals. We report the mean and variance (over all folds) rounded to two decimal points – *smaller* numbers are better.

5 Conclusion & Summary

In this work, we proposed a flexible two-stage algorithm for finding groups of instances for which we can compute cost-efficient multi-instance counterfactual explanations. Our proposed algorithm 1) groups instances such that 2) the single multi-instance counterfactual for each group is as simple as possible (i.e. cost efficient). The empirical evaluation shows that our proposed algorithm (the proposed grouping as well as the proposed evolutionary method) has either a superior or competitive performance compared to existing methods for computing multi-instance counterfactual explanations.

Based on this initial work, there exist a couple of interesting directions for future research: Although the proposed method for computing groupings for cost-efficient multi-instance counterfactuals showed good results, it suffers from the necessity of computing single counterfactuals for each instance. In this context, it would be interesting to study the performance of approximations of counterfactuals, such as gradients if available. Given the importance of the grouping to the final multi-instance counterfactuals, it would be interesting to highlight the contribution/influence of each instance in the group on the final multi-instance counterfactual. In this context, hierarchical clustering in combination with diverse counterfactuals might constitute an interesting first approach.

Acknowledgments. This research was supported by the Ministry of Culture and Science NRW (Germany) as part of the Lamarr Fellow Network. This publication reflects the views of the authors only.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al.: Explainable artifi-

cial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. Information fusion **58**, 82–115 (2020)
2. Artelt, A., Hammer, B.: "explain it in the same way!" – model-agnostic group fairness of counterfactual explanations. In: Ofra, A., Miller, T., Baier, H. (eds.) Workshop on XAI (2023), https://sites.google.com/view/xai2023
3. Artelt, A., Vaquet, V., Velioglu, R., Hinder, F., Brinkrolf, J., Schilling, M., Hammer, B.: Evaluating robustness of counterfactual explanations. In: IEEE Symposium Series on Computational Intelligence, SSCI 2021, Orlando, FL, USA, December 5-7, 2021. pp. 1–9. IEEE (2021). https://doi.org/10.1109/SSCI50451.2021.9660058
4. Artelt, A., Gregoriades, A.: "how to make them stay?": Diverse counterfactual explanations of employee attrition. In: Proceedings of the 25th International Conference on Enterprise Information Systems, ICEIS 2023, Volume 1, Prague, Czech Republic. pp. 532–538. SCITEPRESS (2023). https://doi.org/10.5220/0011961300003467
5. Byrne, R.M.J.: Counterfactuals in explainable artificial intelligence (xai): Evidence from human reasoning. In: IJCAI-19. pp. 6276–6282 (7 2019). https://doi.org/10.24963/IJCAI.2019/876
6. Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., et al.: Xgboost: extreme gradient boosting. R package version 0.4-2 **1**(4), 1–4 (2015), https://cran.ms.unimelb.edu.au/web/packages/xgboost/vignettes/xgboost.pdf
7. Commission, E., for Communications Networks, D.G., Content, Technology: Proposal for a regulation laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts (2021)
8. Dwivedi, R., Dave, D., Naik, H., Singhal, S., Omer, R., Patel, P., Qian, B., Wen, Z., Shah, T., Morgan, G., et al.: Explainable ai (xai): Core ideas, techniques, and solutions. ACM Computing Surveys **55**(9), 1–33 (2023)
9. Guidotti, R.: Counterfactual explanations and how to find them: literature review and benchmarking. Data Mining and Knowledge Discovery pp. 1–55 (2022). https://doi.org/10.1007/s10618-022-00831-6
10. IBM: Ibm hr analytics employee. https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset (2020)
11. Kanamori, K., Takagi, T., Kobayashi, K., Ike, Y.: Counterfactual explanation trees: Transparent and consistent actionable recourse with decision trees. In: Camps-Valls, G., Ruiz, F.J.R., Valera, I. (eds.) International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event. vol. 151, pp. 1846–1870. PMLR (2022), https://proceedings.mlr.press/v151/kanamori22a.html
12. Karimi, A.H., Barthe, G., Schölkopf, B., Valera, I.: A survey of algorithmic recourse: contrastive explanations and consequential recommendations. ACM Computing Surveys (2021)
13. Khan, K., Rehman, S.U., Aziz, K., Fong, S., Sarasvady, S.: Dbscan: Past, present and future. In: The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014). pp. 232–238. IEEE (2014)
14. Kodinariya, T.M., Makwana, P.R., et al.: Review on determining number of cluster in k-means clustering. International Journal **1**(6), 90–95 (2013)
15. von Kügelgen, J., Karimi, A., Bhatt, U., Valera, I., Weller, A., Schölkopf, B.: On the fairness of causal algorithmic recourse. In: Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Virtual Event, February 22 - March 1, 2022. pp. 9584–9594. AAAI Press (2022). https://doi.org/10.1609/AAAI.V36I9.21192

16. Ley, D., Mishra, S., Magazzeni, D.: GLOBE-CE: A translation based approach for global counterfactual explanations **202**, 19315–19342 (2023), https://proceedings.mlr.press/v202/ley23a.html
17. Looveren, A.V., Klaise, J.: Interpretable counterfactual explanations guided by prototypes **12976**, 650–665 (2021). https://doi.org/10.1007/978-3-030-86520-7_40
18. Molnar, C.: Interpretable Machine Learning (2019)
19. Mothilal, R.K., Sharma, A., Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3351095.3372850
20. parliament, E., council: General data protection regulation: Regulation (eu) 2016/679 of the european parliament (2016)
21. Plumb, G., Terhorst, J., Sankararaman, S., Talwalkar, A.: Explaining groups of points in low-dimensional representations. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event. Proceedings of Machine Learning Research, vol. 119, pp. 7762–7771. PMLR (2020), http://proceedings.mlr.press/v119/plumb20a.html
22. Poyiadzi, R., Sokol, K., Santos-Rodriguez, R., De Bie, T., Flach, P.: Face: Feasible and actionable counterfactual explanations. Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3375627.3375850
23. Rawal, A., Mccoy, J., Rawat, D.B., Sadler, B., Amant, R.: Recent advances in trustworthy explainable artificial intelligence: Status, challenges and perspectives. IEEE Transactions on Artificial Intelligence **1**(01), 1–1 (2021)
24. Rawal, K., Lakkaraju, H.: Beyond individualized recourse: Interpretable and interactive summaries of actionable recourses. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems 33: NeurIPS 2020, December 6-12, 2020, virtual (2020), https://proceedings.neurips.cc/paper/2020/hash/8ee7730e97c67473a424ccfeff49ab20-Abstract.html
25. Reeves, C.R.: Genetic algorithms. Handbook of metaheuristics pp. 109–139 (2010)
26. Riveiro, M., Thill, S.: The challenges of providing explanations of ai systems when they do not behave like users expect. In: Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization. pp. 110–120 (2022)
27. Slack, D., Hilgard, A., Lakkaraju, H., Singh, S.: Counterfactual explanations can be manipulated. Advances in Neural Information Processing Systems **34**, 62–75 (2021)
28. Verma, S., Dickerson, J., Hines, K.: Counterfactual explanations for machine learning: A review (2020)
29. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: Automated decisions and the gdpr. Harv. JL & Tech. **31**, 841 (2017), https://arxiv.org/ftp/arxiv/papers/1711/1711.00399.pdf
30. Warren, G., Keane, M.T., Gueret, C., Delaney, E.: Explaining groups of instances counterfactually for xai: A use case, algorithm and user study for group-counterfactuals. arXiv:2303.09297 (2023)
31. Wightman, L.F.: Lsac national longitudinal bar passage study. lsac research report series. (1998)
32. Yeh, I.C., Lien, C.h.: The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert systems with applications **36**(2), 2473–2480 (2009)
33. Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al.: A survey of large language models. arXiv preprint arXiv:2303.18223 (2023)