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

André Artelt[1,2], Andreas Gregoriades[3]

[1]*Faculty of Technology, Bielefeld University, Inspiration 1, 33615 Bielefeld, Germany*

[2]*University of Cyprus, Panepistimiou 1, 2109 Aglantzia, Nicosia, Cyprus*

[3]*Department of Management, Entrepreneurship and Digital Business, Cyprus University of Technology, 30 Arch. Kyprianos Str., 3036 Limassol, Cyprus*

Abstract

Counterfactual explanations constitute among the most popular methods for analyzing black-box systems since they can recommend cost-efficient and actionable changes to the input of a system to obtain the desired system output. While most of the existing counterfactual methods explain a single instance, several real-world problems, such as customer satisfaction, require the identification of a single counterfactual that can satisfy multiple instances (e.g. customers) simultaneously. To address this limitation, in this work, we propose a flexible two-stage algorithm for finding groups of instances and computing cost-efficient multi-instance counterfactual explanations. The paper presents the algorithm and its performance against popular alternatives through a comparative evaluation.

Keywords

XAI, Counterfactual Explanations, Multi-instance Counterfactuals

1. Introduction

Recently an increasing number of Artificial Intelligence (AI) systems have been applied to important problems, such as medical image classification [1] and many more. Although these systems show impressive performance when used on experimental data, they are still imperfect when applied to real-world problems, and in some cases can cause harm to humans due to biases embedded in their logic [2]. Therefore, transparency of such systems is of paramount importance, since it assists in understanding their logic and thus allows decision-makers to decide where and how it is safe to deploy them [3]. The importance of transparency is also stressed at EU level, with recent regulations such as the AI act [4] making explicit reference to the need for explainability. The field of explainability is not new and focuses on answering "why" a system behaves in a certain way. Recently the term eXplainable AI (XAI) [5] has been coined, which boosted the popularity of the field and led to the introduction of many different XAI methods in different domains [5, 6]. One of the most popular types of explanation methods are counterfactual explanations [7], which mimic the way humans seek explanations [8]. By definition, a counterfactual explanation provides 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 business problems where AI systems are deployed such as customer repurchase prediction (how to make customers buy again from a firm), and employee attrition (how to prevent employees from leaving the organisation), the decision maker is not only interested in explaining a single instance of the predictive system but a group of instances – e.g. how to prevent many employees from quitting, instead of only one. To address such use cases, the concept of multi-instance counterfactual explanations has been recently introduced [9, 10]. Here, the aim is to identify a single explanation of 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 cost-efficient counterfactual explanations for groups of instances (multi-instance). Based on our formal analysis, we propose a model (data-agnostic) two-stage algorithm for computing such multi-instance counterfactual explanations.

2. Foundations

A counterfactual explanation (often just called counterfactual) proposes cost-effective and actionable changes to the features of a given input instance of a model such that its prediction changes to the desired output. Because counterfactuals mimic human explanations [8], they constitute among the most popular explanation methods and a favorable choice in many practical problems [11]. The computation of counterfactual explanations involves the consideration of two important aspects [7, 12]: 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 defines the difficulty and effort it takes to realise the explanation (i.e. recommendations) in the real world.

Definition 1 (Counterfactual Explanation). *Assume a prediction function $h : \mathcal{X} \to \mathcal{Y}$ is given. Computing a counterfactual explanation $\vec{\delta}_{cf} \in \mathcal{X}$ for a given instance $\vec{x}_{orig} \in \mathcal{X}$ is phrased as the following optimization problem:* $\underset{\vec{\delta}_{cf} \in \mathcal{X}}{\arg\min} \ \ell\left(h(\vec{x}_{orig} \oplus \vec{\delta}_{cf}), y_{cf}\right) + C \cdot \theta(\vec{\delta}_{cf})$ *where $\ell(\cdot)$ denotes a loss function that penalizes deviation of the output $h(\vec{x}_{orig} \oplus \vec{\delta}_{cf})$ from the requested output y_{cf}, $\theta(\cdot)$ states the cost of $\vec{\delta}_{cf}$, and $C > 0$ denotes the regularization strength.*

The symbol $\oplus$ denotes the application/execution of the counterfactual $\vec{\delta}_{cf}$ to the original instance $\vec{x}_{orig}$ – i.e. for $\mathcal{X} = \mathbb{R}^d$ this reduces to $(\vec{x}_{cf})_i = (\vec{x}_{orig})_i + (\vec{\delta}_{cf})_i$, and to $(\vec{x}_{cf})_i = (\vec{\delta}_{cf})_i$ in the case of categorical features. Also note that the cost of the counterfactual, here modeled by $\theta(\cdot)$, is highly domain and often use-case specific [13], with p-norm being the default. Furthermore, there exist additional relevant aspects such as plausibility [14], robustness [15, 16], fairness [17], etc. The basic formalization Definition 1, however, is still very popular and widely used in practice [11, 13].

Most existing methods do not address the case where we have to assign the same actions to multiple instances simultaneously [9] – i.e. explain more than a single instance within a single counterfactual [18, 9]. In contrast to Definition 1, a multi-instance counterfactual states a

single change $\vec{\delta}_{\text{cf}}$ that alters the output of a classifier $h : \mathcal{X} \to \mathcal{Y}$ for many instances $\vec{x}_i \in \mathcal{X}$ simultaneously. While multi-instance counterfactuals are formalized slightly differently in the literature [10, 18, 9, 19], related work agrees that the same two properties as in Definition 1 must be considered: 1) The cost of the explanation $\vec{\delta}_{\text{cf}}$ 2) an extension of the contrasting property from Definition 1: the explanation δ should be valid for all/many instances in a given set of instances $\mathcal{D}$. In this work, we formalize a multi-instance counterfactuals explanation as follows:

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. $h(\vec{x}_i) = y \quad \forall \vec{x}_i \in \mathcal{D}$. We call all pareto-optimal solutions $\vec{\delta}_{cf}$ to the following multi-objective optimization problem "multi-instance counterfactuals":*

$$\min_{\vec{\delta}_{cf} \in \mathcal{X}} \left(\theta(\vec{\delta}_{cf}) \, , \, \ell(h(\vec{x}_1 \oplus \vec{\delta}_{cf}), y_{cf}), \dots, \ell(h(\vec{x}_{|\mathcal{D}|} \oplus \vec{\delta}_{cf}), y_{cf}) \right)$$

Note that the main difference to 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 $\vec{\delta}_{\text{cf}}$ must be valid for all (or as many as possible) instances in $\mathcal{D}$.

Related Work   One of the earliest works that addresses this problem proposes a counterfactual explanation tree [9], which assigns counterfactuals to the instances at the leaves of a decision tree, derived from $\mathcal{D}$ – i.e. each leaf is interpreted as a group. This method groups instances and computes 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 groupings and also lacks any formal guarantees due to the use of a heuristic (local search) in the implementation. In general, a large part of existing work for multi-instance counterfactuals can be interpreted as summarizing or aggregating individual counterfactual explanations [10, 18, 20, 21]. For instance, in [10], multi-instance counterfactuals are generated by first computing individual counterfactuals and then applying a sampling strategy to select the one that satisfies most instances from a given set, for which a multi-instance counterfactual is requested. In previous work [19], multi-instance counterfactuals are implemented utilizing convex programming for linear classifiers only. However, those methods assume that a grouping is already given and also often suffer from poor performance (e.g. low coverage and correctness). A related branch of research is counterfactual robustness with respect to input changes [22, 23, 15]. Robust counterfactuals [22] should not change for similar instances – i.e. those robust counterfactuals would constitute multi-instance counterfactuals for their local neighborhood in data space. However, if instances are too different from each other, robust counterfactuals do not provide a solution to the multi-instance counterfactual explanation problem.

3. A Two-Stage Algorithm for Multi-instance Counterfactuals

As stated in Definition 2, a multi-instance counterfactual states changes $\vec{\delta}_{\text{cf}}$ that are valid for a set of instances $\mathcal{D}$. While in some scenarios, the $\mathcal{D}$ might be given a priori, 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 intentions. 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 $h : \mathcal{X} \to \mathcal{Y}$ and a set of instances $\mathcal{D} \subset \mathcal{X}^n$ with $h(\vec{x}_i) = y \; \forall \vec{x}_i \in \mathcal{D}, y \in \mathcal{Y}$, we are looking for N partitions $\mathcal{G}_i$ of the instances such that cost-efficient multi-instance counterfactuals (Definition 2) exists:*

$$\min N \quad \max\left(|\mathcal{G}_1|, \ldots, |\mathcal{G}_N|\right) \quad s.t. \bigcup_i \mathcal{G}_i = \mathcal{D}, \quad \mathcal{G}_i \cap \mathcal{G}_j = \emptyset \; \forall i \neq j \tag{1a}$$

$$\min\left(\theta(\vec{\delta}_{cf_1}), \ldots, \theta(\vec{\delta}_{cf_N})\right) \quad \min\left(\ell(h(\vec{x}_j \oplus \vec{\delta}_{cf}), y_{cf}) \; \vec{x}_j \in \mathcal{G}_i, \ldots, \mathcal{G}_N\right) \tag{1b}$$

In this work we study Problem 1 and propose the following process for computing multi-instance counterfactuals: Stage 1) Finding a grouping of instances and then Stage 2) Computing multi-instance counterfactual explanations for each of those groups – by this, we aim to reduce the effect of outliers on the cost of the final multi-instance counterfactuals.

Stage 1- Grouping of Instances. For this task, a naive approach would have been 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 large changes in the input [15], this approach is likely to fail. Furthermore, such an approach does not take into account 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.

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

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. 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. The complete procedure is described in Algorithm 1.

Algorithm 1 Grouping of Instances For Cost-Efficient Multi-instance Counterfactuals

---

Input: Instances $\vec{x}_i$ with the same prediction $h(\vec{x}_i) = \vec{y}$, counterfactual generation $\mathrm{CF}_h(\cdot)$
Output: Grouping of instances

1: $\{\vec{\delta}_{\mathrm{cf}i} = \mathrm{CF}_h(\vec{x}_i)\}$ ▷ Compute a counterfactual $\vec{\delta}_{\mathrm{cf}i}$ for each instance $\vec{x}_i$

2: **for** Different number of clusters **do**   ▷ Optimize number of clusters if requested/needed

3:      Cluster with $\mathrm{d}(\vec{\delta}_{\mathrm{cf}i}, \vec{\delta}_{\mathrm{cf}j}) = \frac{\vec{\delta}_{\mathrm{cf}i}^{\top} \vec{\delta}_{\mathrm{cf}j}}{\|\vec{\delta}_{\mathrm{cf}i}\|_2 \|\vec{\delta}_{\mathrm{cf}j}\|_2}$ ▷ Cluster based on the directions of $\vec{\delta}_{\mathrm{cf}i}$

4:      Sub-cluster with $\mathrm{d}(\vec{\delta}_{\mathrm{cf}i}, \vec{\delta}_{\mathrm{cf}j}) = \|\theta(\vec{\delta}_{\mathrm{cf}i}) - \theta(\vec{\delta}_{\mathrm{cf}j})\|_2$ ▷ Cluster based on the cost $\theta(\vec{\delta}_{\mathrm{cf}i})$

5: **end for**

Stage 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/domain-agnostic methods are limited and often show sub-optimal performance with respect to correctness, we propose an evolutionary method for solving Definition 2. This not only constitutes a model/domain-agnostic method but also a very flexible solution since additional constraints can be easily introduced. Our evolutionary method is an instance of the classic $(\mu + \lambda)$ genetic algorithm [24] and can handle all types of variables. In order to guarantee the feasibility of the final multi-instance counterfactual $\vec{\delta}_{\mathrm{cf}}$ for the given problem domain, we construct the set of feasible changes for each feature of numerical variables as follows – assuming non-negativity which can be achieved by adding a constant: $l_i = \alpha_i - \min_j\{(\vec{x}_j)_i\}$ and $u_i = \beta_i - \max_j\{(\vec{x}_j)_i\}$, where α_i and β_i denote the maximum and minimum feasible value of the i-th feature, and the final set of feasible changes is then given as $[l_i, u_i]$. These sets are used when computing mutations in our evolutionary algorithm of existing individuals during the optimization. Furthermore, we merge the objectives in Eq. (1b) into a single objective as follows: $\arg\min_{\vec{\delta}_{\mathrm{cf}} \in \mathcal{X}} \theta(\vec{\delta}_{\mathrm{cf}}) + C \cdot \sum_{\vec{x}_i \in \mathcal{D}} \ell(h(\vec{x}_j \oplus \vec{\delta}_{\mathrm{cf}}), y_{\mathrm{cf}})$ where the cost $\theta(\vec{\delta}_{\mathrm{cf}})$ is defined as: $\theta(\vec{\delta}_{\mathrm{cf}}) = \sum_i \psi((\vec{\delta}_{\mathrm{cf}})_i)$ where $\psi((\vec{\delta}_{\mathrm{cf}})_i) = |(\vec{\delta}_{\mathrm{cf}})_i|$ or 1 if i-feature is categorical.

4. Experiments

The following experiments are conducted to showcase the application and merits of the proposed method. All experiments and the proof of Lemma 1 are publicly available on GitHub[1].

We consider two datasets that have been used in other work on multi-instance counterfactuals [19, 9]: The *IBM human resource attrition dataset* [25] (*Attr.*) containing 35 features for 1467 unique employees. The *Law school data set* [26] (*Law*) containing 20798 law school admission records, each described by 12 attributes.

Setup.   We empirically compare our proposed evolutionary algorithm (denoted by *EA*) from Section 3, against two methods [10, 9], which similarly to our method are also model and data-agnostic. In this context, we evaluate two properties of the computed multi-instance counterfactual explanations, and the results are shown in Table 1 and Table 2. The properties

---

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

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.

Data	Method	No Clustering ↑	Clustering $\vec{x}_i$ ↑	Clustering CFs ↑
	EA [Ours]	0.98 ± 0.0	0.95 ± 0.02	1.0 ± 0.0
Attr.	Warren et al. [10]	0.37 ± 0.02	0.34 ± 0.04	0.52 ± 0.06
	Kanamori et al. [9]	0.98 ± 0.0	0.95 ± 0.01	0.86 ± 0.05
	EA [Ours]	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Law	Warren et al. [10]	0.06 ± 0.0	0.09 ± 0.02	0.03 ± 0.01
	Kanamori et al. [9]	0.97 ± 0.0	0.99 ± 0.0	0.81 ± 0.05

are: 1) The correctness, i.e. evaluating for how many samples the explanation is correct: $\frac{1}{|\mathcal{D}|} \sum_i \mathbb{1}\left(h(\vec{x}_i \oplus \vec{\delta}_{\mathrm{cf}}) = y_{\mathrm{cf}}\right)$ – Table 1. 2) The cost $\theta(\cdot)$ expressed as the number of changed features, i.e. $\theta(\vec{\delta}_{\mathrm{cf}}) = \sum_i \mathbb{1}\left((\vec{\delta}_{\mathrm{cf}})_i \neq 0\right)$ - Table 2. All experiments were done using a five-fold cross-validation and we report the mean and variance of the results over all folds. An XGBoost classifier is fitted to the training set and all negatively classified instances (i.e. $h(\vec{x}_i) = 0$) from the test-set define the set $\mathcal{D}$ used by the multi-instance counterfactual method. We compute a multi-instance counterfactual for the entire set of selected instances $\mathcal{D}$, and also cluster (using DBSCAN) the set $\mathcal{D}$ in two different ways: 1) Clustering based on the individual counterfactuals as proposed in Algorithm 1 using the cosine-similarity, and for comparison 2) clustering based on the individual instances $\vec{x}_i$ using the Euclidean distance.

Results & Discussion. From the results, we observe that our proposed method achieves excellent performance (with respect to correctness and cost) across all settings. The method by Warren et al [10] 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 and therefore have a higher cost if implemented in practice. The method by Kanamori et al. [9] often achieves a competitive performance 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 in Algorithm 1 often improves significantly the correctness and the cost of the computed multi-instance counterfactuals. The results demonstrate the merits of our proposed two-stage algorithm.

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 groups instances so that the single multi-instance counterfactual for each group is as simple as possible (i.e. cost efficient). From the empirical evaluation of the method, we conclude that our proposed algorithm (the grouping and the proposed evolutionary method)

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.

Data	Method	No Clustering ↓	Clustering $\vec{x}_i$ ↓	Clustering CFs ↓
	EA [Ours]	0.73 ± 0.02	0.69 ± 0.02	0.55 ± 0.04
Attr.	Warren et al. [10]	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
	Kanamori et al. [9]	0.02 ± 0.0	0.06 ± 0.02	0.05 ± 0.01
	EA [Ours]	0.67 ± 0.01	0.67 ± 0.01	0.64 ± 0.01
Law	Warren et al. [10]	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
	Kanamori et al. [9]	0.11 ± 0.01	0.05 ± 0.01	0.14 ± 0.01

has either superior or competitive performance compared to existing methods for computing multi-instance counterfactual explanations. The main limitation of our method is that it suffers from the necessity of computing single counterfactuals for each instance and this impacts its computational performance. Therefore, as part of our future work, we will investigate how to improve computational performance through the use of approximations of counterfactuals, such as gradients.

Acknowledgments

References

[1] M. Goyal, T. Knackstedt, S. Yan, S. Hassanpour, Artificial intelligence-based image classification methods for diagnosis of skin cancer: Challenges and opportunities, Computers in biology and medicine 127 (2020) 104065.

[2] X. Ferrer, T. Van Nuenen, J. M. Such, M. Coté, N. Criado, Bias and discrimination in ai: a cross-disciplinary perspective, IEEE Technology and Society Magazine 40 (2021) 72–80.

[3] S. Larsson, F. Heintz, Transparency in artificial intelligence, Internet Policy Review (2020).

[4] E. Commission, Proposal for a regulation laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts, 2021.

[5] R. Dwivedi, D. Dave, H. Naik, S. Singhal, R. Omer, P. Patel, B. Qian, Z. Wen, T. Shah, G. Morgan, et al., Explainable ai (xai): Core ideas, techniques, and solutions, ACM Computing Surveys 55 (2023) 1–33.

[6] A. Rawal, J. Mccoy, D. B. Rawat, B. Sadler, R. Amant, Recent advances in trustworthy explainable artificial intelligence: Status, challenges and perspectives, IEEE Transactions on Artificial Intelligence 1 (2021) 1–1.

[7] S. Wachter, B. Mittelstadt, C. Russell, Counterfactual explanations without opening the black box: Automated decisions and the gdpr, Harv. JL & Tech. 31 (2017) 841.

[8] R. M. J. Byrne, Counterfactuals in explainable artificial intelligence (xai): Evidence from human reasoning, in: IJCAI-19, 2019, pp. 6276–6282. doi:10.24963/IJCAI.2019/876.

[9] K. Kanamori, T. Takagi, K. Kobayashi, Y. Ike, Counterfactual explanation trees: Transparent and consistent actionable recourse with decision trees, in: AISTATS 2022, 2022. URL: https://proceedings.mlr.press/v151/kanamori22a.html.

[10] G. Warren, M. T. Keane, C. Gueret, E. Delaney, Explaining groups of instances counterfactually for xai: A use case, algorithm and user study for group-counterfactuals, arXiv:2303.09297 (2023).

[11] S. Verma, J. Dickerson, K. Hines, Counterfactual explanations for machine learning: A review, 2020. `arXiv:2010.10596`.

[12] M. T. Keane, B. Smyth, Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable ai (xai), in: ICCBR, 2020.

[13] R. Guidotti, Counterfactual explanations and how to find them: literature review and benchmarking, Data Mining and Knowledge Discovery (2022) 1–55. doi:`10.1007/s10618-022-00831-6`.

[14] R. Poyiadzi, K. Sokol, R. Santos-Rodriguez, T. De Bie, P. Flach, Face: Feasible and actionable counterfactual explanations, Association for Computing Machinery, New York, NY, USA, 2020. doi:`10.1145/3375627.3375850`.

[15] A. Artelt, V. Vaquet, R. Velioglu, F. Hinder, J. Brinkrolf, M. Schilling, B. Hammer, Evaluating robustness of counterfactual explanations, in: IEEE SSCI, 2021. doi:`10.1109/SSCI50451.2021.9660058`.

[16] D. Slack, A. Hilgard, H. Lakkaraju, S. Singh, Counterfactual explanations can be manipulated, Advances in Neural Information Processing Systems 34 (2021) 62–75.

[17] A. Artelt, B. Hammer, "explain it in the same way!" – model-agnostic group fairness of counterfactual explanations, in: IJCAI Workshop on XAI, 2023. URL: https://sites.google.com/view/xai2023.

[18] D. Ley, S. Mishra, D. Magazzeni, GLOBE-CE: A translation based approach for global counterfactual explanations 202 (2023) 19315–19342. URL: https://proceedings.mlr.press/v202/ley23a.html.

[19] A. Artelt, A. Gregoriades, "how to make them stay?": Diverse counterfactual explanations of employee attrition, in: ICEIS, 2023. doi:`10.5220/0011961300003467`.

[20] K. Rawal, H. Lakkaraju, Beyond individualized recourse: Interpretable and interactive summaries of actionable recourses, in: NeurIPS, 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/8ee7730e97c67473a424ccfeff49ab20-Abstract.html.

[21] G. Plumb, J. Terhorst, S. Sankararaman, A. Talwalkar, Explaining groups of points in low-dimensional representations, in: ICML 2020, volume 119, PMLR, 2020, pp. 7762–7771. URL: http://proceedings.mlr.press/v119/plumb20a.html.

[22] F. Leofante, N. Potyka, Promoting counterfactual robustness through diversity, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, 2024, pp. 21322–21330.

[23] R. Dominguez-Olmedo, A. H. Karimi, B. Schölkopf, On the adversarial robustness of causal algorithmic recourse, in: ICML, 2022.

[24] C. R. Reeves, Genetic algorithms, Handbook of metaheuristics (2010) 109–139.

[25] IBM, Ibm hr analytics employee, https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset, 2020.

[26] L. F. Wightman, Lsac national longitudinal bar passage study. lsac research report series. (1998).