Towards Understanding the Influence of Training Samples on Explanations

André Artelt[1,2][0000−0002−2426−3126] and Barbara Hammer[1][0000−0002−0935−5591]

[1] Bielefeld Universty, Germany
[2] University of Cyprus, Cyprus
{aartelt,bhammer}@techfak.uni-bielefeld.de

Abstract. Explainable AI (XAI) is widely used to analyze AI systems' decision-making, such as providing counterfactual explanations for recourse. When unexpected explanations occur, users may want to understand the training data properties shaping them. Under the umbrella of data valuation, first approaches have been proposed that estimate the influence of data samples on a given model. This process not only helps determine the data's value, but also offers insights into how individual, potentially noisy, or misleading examples affect a model, which is crucial for interpretable AI. In this work, we apply the concept of data valuation to the significant area of model evaluations, focusing on how individual training samples impact a model's internal reasoning rather than the predictive performance only. Hence, we introduce the novel problem of identifying training samples shaping a given explanation or related quantity, and investigate the particular case of the cost of computational recourse. We propose an algorithm to identify such influential samples and conduct extensive empirical evaluations in two case studies.

Keywords: XAI · Data Valuation · Counterfactual Explanations

1 Introduction

Today, numerous AI and ML systems are deployed in real-world applications [30], demonstrating impressive performance yet remaining imperfect. Issues like failures, fairness concerns, and vulnerabilities to manipulations such as data poisoning can pose risks. Therefore, transparency is crucial to prevent failures, build trust, and ensure safe deployment. Policymakers have recognized this, embedding transparency in regulations like the EU's GDPR [9] and the EU AI Act [8]. Explanations are a key way to achieve transparency, shaping the field of Explainable AI (XAI) [10]. Due to diverse use cases and users, many explanation methods exist, including popular ones like LIME [20], SHAP [26], and counterfactual explanations [29], that offer computational recourse.

Although current XAI methods can reveal the internal logic of a model, they do not clarify the reasons behind this logic. However, when faced with surprising or undesirable explanations, like an expensive or impractical recourse recommendation, users might want an "explanation of the explanation." This

need is also heightened by recent findings that explanations can be manipulated or poisoned [3, 5, 6], undermining users' trust. A possible way to address this is by tracing explanations back to influential training samples, which shaped the model's internal logic. The knowledge of such influential training samples could not only reveal insights into the relevance of certain training samples for shaping the model's reasoning, but also allow for sanity checks, removal, or correction of the training data if necessary. *Such an approach would explain a model's reasoning within the training data space, complementing traditional XAI methods that focus on the feature or model parameter space.* To the best of our knowledge, no prior work has looked into this aspect.

Our contributions: We introduce and formalize the novel problem of *analyzing the influence of training samples on explanations* and propose an algorithm for identifying such influential training samples. We focus on two specific cases involving counterfactual explanations for computational recourse derived from the trained model: 1) Identifying training samples that strongly influence the average cost of recourse, and 2) Identifying samples that significantly affect the cost difference of recourse between two protected groups, indicating a group fairness violation. We conduct extensive empirical evaluations of our proposed algorithms and compare them with baseline approaches.

2 Foundations

2.1 Data-Valuation

Data valuation [24] focuses on assessing the importance of individual training samples on predictive performance, quantifying each sample's contribution to the final model. This knowledge can be used to compensate users for their data, verify the accuracy of highly relevant samples, or acquire more data similar to the most significant training samples.

In this context, the Data-SHAP method [13] constitutes a popular model-agnostic method that carries the concept of Shapley-Values [26] over to data valuation. Here, the Shapley value $\phi_i \in \mathbb{R}$ states the contribution of the i-th player (e.g. feature or training data point) to some quantity of interest $V : \mathcal{S} \mapsto \mathbb{R}$ (also called value function) of a predictive function $h : \mathcal{X} \to \mathcal{Y}$ derived from a given training data set $\mathcal{S}$. In data valuation, as stated before, the property of interest is usually the predictive performance of $h(\cdot)$, e.g. the accuracy is used as an implementation of the value function $V(\cdot)$. Requiring some equitable properties, it can be shown [13] that the solution of ϕ_i is given as:

$$\phi_i = C \sum_{\mathcal{S} \subseteq \mathcal{D} - \{i\}} \frac{V(\mathcal{S} \cup \{i\}) - V(\mathcal{S})}{\binom{|\mathcal{D}|-1}{|\mathcal{S}|}} \tag{1}$$

where C is a constant, and $\{i\}$ refers to the i-th sample in a given set $\mathcal{D}$. Like Shapley-Values, the computation of Eq. (1) is computationally infeasible. Therefore, in [13] a Monte-Carlo approximation of Eq. (1) is proposed:

$$\phi_i = \mathbb{E}_{\pi \sim \Pi}[V(\mathcal{S}^i_\pi \cup \{i\}) - V(\mathcal{S}^i_\pi)] \tag{2}$$

where S_π^i denotes the first $i-1$ samples in the training data set $\mathcal{S}$ under the permutation π. Furthermore, $\mathcal{S}_\pi^i \cup \{i\}$ denotes the addition of the i-th training data sample to the S_π^i. In order to completely avoid the computationally expensive refitting of $h(\cdot)$ in Eq. (2), the same authors [13] propose to only perform a single gradient descent step instead of completely refitting $h(\cdot)$ in Eq. (2) – i.e. the influence scores ϕ_i are estimated "on the fly" while training the model $h(\cdot)$.

2.2 Counterfactuals for Computational Recourse

A counterfactual explanation (often just called counterfactual) suggests changes to an input's features to alter the system's output, often requested for unfavorable outcomes [21] as a form of (computational) *recourse* [16]. They are popular [27] because they mimic human reasoning in explanations [7].

The computation of a classic counterfactual [29] $\boldsymbol{\delta}_{\text{cf}} \in \mathbb{R}^d$ for a given instance $\boldsymbol{x}_{\text{orig}} \in \mathbb{R}^d$ is phrased as the following optimization problem:

$$\underset{\boldsymbol{\delta}_{\text{cf}} \in \mathbb{R}^d}{\arg\min}\ \ell\left(h(\boldsymbol{x}_{\text{orig}} + \boldsymbol{\delta}_{\text{cf}}), y_{\text{cf}}\right) + C \cdot \theta(\boldsymbol{\delta}_{\text{cf}}) \tag{3}$$

where $\ell(\cdot)$ penalizes deviation of the prediction $h(\boldsymbol{x}_{\text{orig}} + \boldsymbol{\delta}_{\text{cf}})$ from the requested outcome y_{cf}, $\theta(\cdot)$ states the cost of the explanation (e.g. cost of recourse) which should be minimized, and $C > 0$ denotes the regularization strength balancing those two objectives. The short-hand notation $\text{CF}(\boldsymbol{x}, h)$ denotes the solution to Eq. (3) iff the target outcome y_{cf} is uniquely determined. Note that the cost $\theta(\cdot)$ is domain-specific, but many implementations default to using the p-norm [14].

Remark 1. In the case of recourse – i.e. turning an unfavorable into a favorable outcome –, we refer to the cost $\theta(\boldsymbol{\delta}_{\text{cf}})$, as the *cost of recourse*.

In this work, w.l.o.g., we refer to $y = 0$ as the unfavorable, and $y = 1$ as the favorable outcome. Besides those two essential objectives in Eq. (3), there exist additional relevant aspects such as plausibility [17, 19], diversity [18], robustness [4, 15], etc. which have been addressed in literature [14]. However, the basic formalization Eq. (3) is still very popular and widely used in practice [14, 27].

A critical and still unsolved fairness issue in computational recourse is the difference in the cost of recourse $\theta(\boldsymbol{\delta}_{\text{cf}})$ between protected groups [2, 22, 28] – i.e. individuals from one protected group (e.g. gender) get more costly recommendations on how to achieve recourse. It was shown that such cases could be created intentionally by targeted attacks [3, 25].

Implementation There exist numerous methods and implementations/toolboxes for computing counterfactual explanations in practice [14] – most include some additional aspects such as plausibility, diversity, etc. *Counterfactuals Guided by Prototypes* [17] focuses on plausibility. Here a set of plausible instances (so-called prototypes) are used to pull the final counterfactual instance $\boldsymbol{x}_{\text{cf}}$ (i.e. $\boldsymbol{x}_{\text{cf}} := \boldsymbol{x}_{\text{orig}} + \boldsymbol{\delta}_{\text{cf}}$) closer to these plausible instances and by this make them

more plausible. The *Nearest Unlike Neighbor method* is a straightforward baseline method for computing plausible counterfactuals by picking the closest sample, with the requested output y_{cf}, from a given set (e.g. training set) as the counterfactual instance $\boldsymbol{x}_{\text{cf}}$.

3 Influence of Training Samples on Explanations

We consider scenarios, where a predictive model $h_{\mathcal{D}_{\text{train}}} : \mathcal{X} \to \mathcal{Y}$ is derived from a given training data set $\mathcal{D}_{\text{train}}$. Additionally, we assume an explanation generation mechanism $\text{expl}(h_{\mathcal{D}_{\text{train}}}, X)$ that produces either a local or global explanation z within the set $\in \mathcal{E}$ for the given $h_{\mathcal{D}_{\text{train}}}(\cdot)$. In the case of a local explanation, X represents the sample or region for which an explanation is computed; otherwise, X is ignored. Note that we do not make any assumption on $\text{expl}(\cdot, \cdot)$ – it could be any explanation or a related metric, such as the cost of recourse in the context of counterfactual explanations.

In this work, we are interested in identifying training samples in $\mathcal{D}_{train}$ that had a high influence on an observed explanation $expl(h_{\mathcal{D}_{train}}, X)$ – e.g. outlier or malicious training samples that shaped the observed explanation.

3.1 Quantifying the Influence of Training Samples on Explanations

To quantify the impact of training samples on a given explanation, we need a metric for evaluating the similarity between explanations – i.e. a mechanism that evaluates how similar or different two given explanations are. For this purpose, we introduce a function $\Psi : \mathcal{E} \times \mathcal{E} \to \mathbb{R}$ that computes the similarity of two given explanations $z_1, z_2 \in \mathcal{E}$ – where we assume the same explanation generation mechanism $\text{expl}(h, X)$ for z_1 and z_2 but not necessarily the same training data set from which $h(\cdot)$ was derived. We require that $\Psi(z_1, z_2) = 0 \leftrightarrow z_1 = z_2$; other than that, the computed real number is supposed to indicate their difference – i.e. a larger output corresponds to a larger difference of the two given explanations z_1, z_2. Note, that the sign of the output of $\Psi(\cdot, \cdot)$ may offer additional insights, such as indicating direction or comparing magnitudes – e.g. comparing the cost of recourse of two counterfactual explanations. *Example: In the case of explanations that are stated as real-valued vectors (i.e. $\mathcal{E} = \mathbb{R}^m$), $\Psi(\cdot)$ could be implemented by comparing their lengths: $\Psi(z_1, z_2) = \|z_1\|_p - \|z_2\|_p$, i.e. comparing the recourse costs of the counterfactuals.*

Based on $\Psi(\cdot, \cdot)$ we characterize influential training samples as follows:

Definition 1 (Influential Training Samples). *For a given training set $\mathcal{D}_{train}$, we say that $\mathcal{D}_{infl} \subset \mathcal{D}_{train}$ has a strong influence on the explanation $expl(h_{\mathcal{D}_{train}}, X)$ iff the absence of $\mathcal{D}_{infl}$ changes the explanation $expl(h_{\mathcal{D}_{train}}, X)$ significantly, i.e.:*

$$\left| \Psi\big(\mathbb{E}[expl(h_{\mathcal{D}_{train}}, X)], \mathbb{E}[expl(h_{\mathcal{D}_{train} \setminus \mathcal{D}_{infl}}, X)]\big) \right| \gg 0 \tag{4}$$

Note that the expected value $\mathbb{E}[\cdot]$ in Eq. (4) is accounting for randomness in the training process of $h_{\mathcal{D}}(\cdot)$ which for instance naturally occurs in the training of neural networks.

Algorithm 1 Finding Influential Training Samples

Input: Labeled training set $\mathcal{D}$; K number of training steps; Differentiable model $h(\cdot)$; Single or group of samples X which are going to be explained
Output: Influence score ϕ_i of each training sample.

1: **for** $j = 1, ..., N$ or until convergence **do**
2: Random initialization of $h(\cdot)$ weights $\boldsymbol{w}$
3: **for** $t = 1, ..., K$ **do** ▷ Training loop
4: $z_{\mathcal{D}} = \text{expl}(h, X)$ ▷ Initial explanation
5: $\pi = \text{random\_permutation}(|\mathcal{D}|)$
6: **for** idx $\in \pi$ **do**
7: $(\boldsymbol{x}_i, y_i) = \mathcal{D}[\text{idx}]$ ▷ Get current sample
8: $\boldsymbol{w} - = \lambda \nabla_{\boldsymbol{w}} \ell(\boldsymbol{x}_i, y_i \mid \boldsymbol{w})$ ▷ Gradient descent
9: $z_{\mathcal{D} \cup \{i\}} = \text{expl}(h, X)$ ▷ Compute new explanation
10: $\phi^{j,t}_{[\text{idx}]} = \Psi(z_{\mathcal{D} \cup \{i\}}, z_{\mathcal{D}})$ ▷ Compute influence of $(\boldsymbol{x}_i, y_i)$
11: $z_{\mathcal{D}} = z_{\mathcal{D} \cup \{i\}}$ ▷ Update current explanation
12: **end for**
13: **end for**
14: **end for**
15: $\phi_i = \frac{1}{N \cdot K} \sum_{j,t} \phi_i^{j,t}$ ▷ Final influence scores

3.2 Reduction to a Game-theoretic Approach

In this work, we propose to find such influential training samples (Definition 1) by conceptualizing a game-theoretic approach, namely the Data-SHAP method [13].

For this, we assume that $\Psi(z_1, z_2)$ can be decomposed as $\Psi(z_1, z_2) = V(z_1) - V(z_2)$ for some $V : \mathcal{E} \to \mathbb{R}$. This assumption allows us to use the Data-SHAP method [13] (Eq. (1)) for quantifying the influence ϕ_i of a single training sample:

$$V(\mathcal{D} \cup \{i\}) - V(\mathcal{D}) := \Psi\big(\text{expl}(h_{\mathcal{D} \cup \{i\}}, X), \text{expl}(h_{\mathcal{D}}, X)\big) \tag{5}$$

Given the computational complexity of Eq. (1) and the potentially complex computation of explanations in Eq. (5), we propose to use a gradient-based Monte-Carlo approach similar to the one in the original Data-SHAP paper [13] – assuming that the predictive model $h(\cdot)$ is differentiable and can be trained using gradient-descent. Here, we estimate the ϕ_i scores while training the model $h(\cdot)$ – i.e. instead of training the model $h(\cdot)$ to convergence, we already evaluate the explanations at training time. The complete gradient-based Monte-Carlo algorithm for computing the influence score ϕ_i of each training sample on an explanation is described in Algorithm 1.

The core part of Algorithm 1 are lines 3-11. The neural network (or any other differentiable model $h(\cdot)$) is initialized with random parameters $\boldsymbol{w}$ and trained for K iterations. In each iteration, the entire training data set $\mathcal{D}$ is shuffled (line 5) and then each training sample $(\boldsymbol{x}_i, y_i)$ is processed as follows: 1) update the model parameters $\boldsymbol{w}$ using gradient descent (line 8); 2) Recompute the explanation $\text{expl}(h, X)$ of interest (line 9); 3) Compare current explanation with the explanation from the previous iteration (line 10) – this allows us to estimate

(Eq. (5)) the influence of the current training sample $(\boldsymbol{x}_i, y_i)$ on the explanation of interest. This procedure is repeated several times until the influence scores ϕ_i converge. Then, we select the training samples with the largest absolute influence score as the set of most influential training samples $\mathcal{D}_{\text{infl}}$ (Definition 1):

$$\mathcal{D}_{\text{infl}} = \{(\boldsymbol{x}_{i_0}, y_{i_0}), (\boldsymbol{x}_{i_1}, y_{i_1}), \dots\} \text{ s.t. } |\phi_{i_0}| \geq |\phi_{i_1}| \geq \dots \tag{6}$$

where the number of influential training samples could either be given by the user or determined automatically by applying a minimum threshold to $|\phi_i|$.

Computational Considerations Since Algorithm 1 likely requires many iterations to converge, the computation of the explanation $\text{expl}(h, X)$ in each step might become a (computational) bottleneck. Therefore, approximations of the explanation generation mechanism $\text{expl}(h, X)$ might be necessary as proposed in the two case studies.

4 Case-Study I: Cost of Recourse

Here, we consider the challenge of identifying training samples (see Algorithm 1) that significantly impact the average cost of recourse – i.e., the average cost it takes to change the outcome for negatively (i.e. unfavorable) classified instances. We aim to identify those training samples that have an increasing effect on the average cost of recourse – those training samples might be suitable candidates for manual inspection and (potential) deletion in order to decrease the average cost of recourse.

Assuming that we have a set of negatively classified instances $\mathcal{D}$, our quantity of interest, the average cost of recourse is defined as follows:

$$\text{expl}(h_{\mathcal{S}}, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{\boldsymbol{x}_i \in \mathcal{D}} \theta \circ \text{CF}(\boldsymbol{x}_i, h_{\mathcal{S}}) \tag{7}$$

Note that the training data set $\mathcal{S}$ and the (test) set $\mathcal{D}$ are not necessarily the same – i.e. one might be interested in evaluating on a hold-out data set $\mathcal{D}$ that is disjunct from the training data set $\mathcal{S}$.

However, the evaluation of Eq. (7) is computationally expensive because it requires computing a counterfactual $\text{CF}(\boldsymbol{x}_i, h_{\mathcal{S}})$ for every sample in $\mathcal{D}$. In order to make the computation of the influence scores ϕ_i feasible, we assume that the cost of recourse $\theta(\cdot)$ is related to the distance to the decision boundary and propose an approximation based on the difference in the logits [22]:

$$\theta \circ \text{CF}(\boldsymbol{x}_i, h_{\mathcal{S}})) \approx |g_0(\boldsymbol{x}_i) - g_1(\boldsymbol{x}_i)| \tag{8}$$

where g_1, g_2 denote the logits of the neural network $h_{\mathcal{S}}(\cdot)$. The authors of [22] show that Eq. (8) constitutes a good approximation of the distance to the decision boundary. We make use of it as a proxy for the cost of recourse.

The final value function $V(\mathcal{S})$ is then given as:

$$V(\mathcal{S}) = \frac{1}{N_{\mathcal{S}}} \sum_{\boldsymbol{x}_i \in \mathcal{D} | h_{\mathcal{S}}(\boldsymbol{x}_i)=0} \Big| g_0(\boldsymbol{x}_i) - g_1(\boldsymbol{x}_i) \Big| \quad N_{\mathcal{S}} := |\{\boldsymbol{x}_i \in \mathcal{D} \mid h_{\mathcal{S}}(\boldsymbol{x}_i) = 0\}| \tag{9}$$

We can then substitute Eq. (9) for Eq. (5) in Algorithm 1 for computing the influence of each training sample. The training samples of interest are those with a large positive influence score, i.e. $\phi_i \gg 0$, since those are increasing the average cost of recourse Eq. (7) (for negatively classified instances) when added to the training set. In the empirical evaluation (Section 6), we investigate the effect of removing those training samples from the training data.

5 Case-Study II: Difference in the Cost of Recourse

We consider the problem of analyzing the reasons for differences in the cost of recourse between two protected groups (here denoted by a binary attribute $q \in \{0, 1\}$), a significant fairness issue in computational recourse [2, 23]. Our goal is to identify training samples that increase this cost difference. These samples could be candidates for manual inspection and potential removal to reduce the disparity and enhance fairness in computational recourse. Here, we consider the difference in the worst-case cost of recourse between two protected groups as our quantity of interest, which is formalized as follows:

$$\text{expl}(h_{\mathcal{S}}, \mathcal{D}) = \left| \max_{\boldsymbol{x}_i \in \mathcal{D}} (\theta \circ \text{CF}(\boldsymbol{x}_i \mid q = 0, h_{\mathcal{S}})) - \max_{\boldsymbol{x}_i \in \mathcal{D}} (\theta \circ \text{CF}(\boldsymbol{x}_i \mid q = 1, h_{\mathcal{S}})) \right| \tag{10}$$

where we only consider negatively classified samples in the given set $\mathcal{D}$ – we drop the explicit constraint $h_{\mathcal{S}}(\boldsymbol{x}_i) = 0$ for better readability. As in the first case study, we achieve computational feasibility by approximating the cost of recourse by the difference in the logits:

$$\theta \circ \text{CF}(\boldsymbol{x}_i \mid q = ?, h_{\mathcal{S}})) \approx |g_0(\boldsymbol{x}_i) - g_1(\boldsymbol{x}_i)| \tag{11}$$

The final value function $V(\mathcal{S})$ is then given as:

$$V(\mathcal{S}) = \left| \max_{\boldsymbol{x}_i \in \mathcal{D} | q=0} (|g_0(\boldsymbol{x}_i) - g_1(\boldsymbol{x}_i)|) - \max_{\boldsymbol{x}_i \in \mathcal{D} | q=1} (|g_0(\boldsymbol{x}_i) - g_1(\boldsymbol{x}_i)|) \right| \tag{12}$$

We can then substitute Eq. (12) for Eq. (5) in Algorithm 1 for computing the influence of each training sample. Again, the training samples of interest to us are those with a large positive influence score, i.e. $\phi_i \gg 0$. In the empirical evaluation (Section 6), we investigate the effect of removing those training samples from the training data set.

6 Experiments

We seek to empirically answer the following two research questions:

RQ1 Correctness of our proposed Algorithm 1: Evaluate how the deletion of those identified training samples influences the counterfactuals, as well as the predictive performance (F1-score) of the classifier.

RQ2 Improvement of our proposed Algorithm 1 over existing data valuation method: Are highly influential training samples on explanations (Definition 1) different from those for (only) influencing the predictive performance?

6.1 Data

We consider the following two benchmark data sets from the fairness literature [12]: 1) The "Diabetes" data set [11] (denoted *Diabetes*) contains data from 442 diabetes patients, each described by 9 numeric attributes together with the sensitive binary attribute "sex". The target is a binarized quantitative measure of disease progression one year after baseline. ; 2) The "German Credit Data set" [1] (denoted *Credit*) is a data set for loan approval and contains 1000 samples each annotated with 7 numerical and 13 categorical attributes, including the sensitive binary attribute "sex", with a binary target value. We only use the seven numerical features.

6.2 Setup

We use the ℓ_1 norm as a popular implementation [14] of the cost of recourse – i.e. $\theta(\cdot) = \|\cdot\|_1$ – and utilize a Multi-Layer Perceptron (with two hidden layers) as the classifier $h(\cdot)$.

We conduct the experiments in a 5-fold cross-validation: 1) fitting the classifier; 2) computing the most influential training samples, whereby all negative classified samples (i.e. $h(\cdot) = 0$) from the test set are considered. We compute the computational recourse (i.e. counterfactual) on the test set using three different popular recourse methods: Nearest Unlike Neighbor (denoted *NUN*), as a simple baseline for plausible counterfactuals; Counterfactuals guided by Prototypes [17] (denoted *Proto*) as an advanced method for computing plausible counterfactuals; Classic counterfactuals [29] (denoted *Wachter*) by solving Eq. (3).

For the purpose of investigating RQ1, we increasingly remove (1% - 30%) of the most influential training samples from the training set, retrain the classifier, and re-evaluate the quantity of interest on the (unchanged) test set – i.e. either the average cost of recourse (case study I) or the difference in the cost of recourse (case study II). To account for the randomness in training the neural network classifier, we take the average (expectation) of the quantity of interest over five training runs.

Baselines For the purpose of investigating RQ2, we compare our method (i.e. Algorithm 1) to two baselines: 1) Removal of random training samples; 2) Removal of the most influential training samples on the predictive performance as returned by the original Data-SHAP method [13].

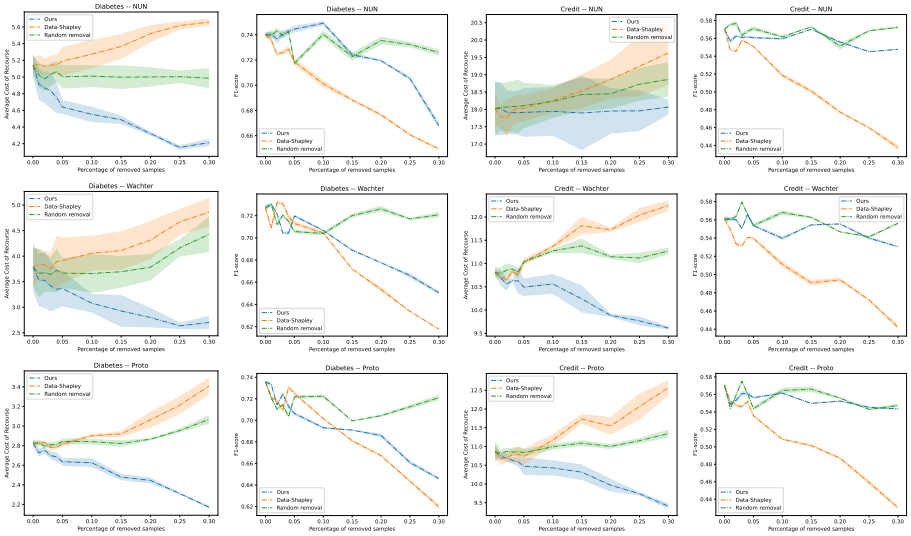

Fig. 1. Case-Study I: Effect of removing training samples that have a high influence on the average cost of recourse Eq. (7) – we show (mean & variance over all folds) the effect on the average cost of recourse, as well as on the predictive performance (i.e. F1-score).

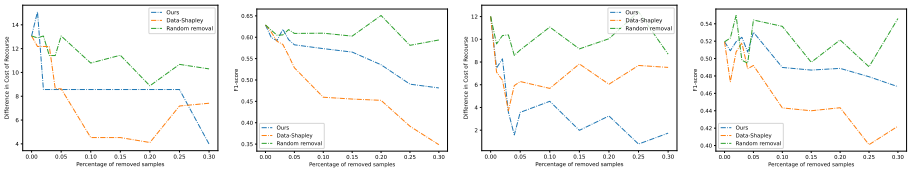

Fig. 2. Case Study II: Effect of removing training samples that have a high influence on the difference in the cost of recourse Eq. (10) – we show the effect on the difference in the cost of recourse, as well as on the predictive performance (i.e. F1-score). Note that we only consider the train-test split with the worst original difference.

6.3 Case Study I

We consider all possible combinations of the two aforementioned data sets and the three aforementioned counterfactual explanation methods. In Figure 1, we report the mean as well as the variance of all measurements. Note that for the average cost of recourse, smaller numbers are better, while for the F1-score, larger numbers are better.

RQ1) We observe that in almost all cases, our method is able to identify influential training samples (Definition 1) that, if removed, indeed decrease the average cost of recourse significantly. RQ2) The two baselines (random removal and Data-SHAP [13]) return training samples that, if removed, often increase (instead of decrease) the average cost of recourse. This shows that training sam-

Table 1. Case Study II: Differences in the cost of recourse between the protected groups Eq. (10) for the Credit data set. We report maximum and average & variance.

NUN		Proto		Wachter	
Max.	Avg. ± Var.	Max.	Avg. ± Var.	Max.	Avg. ± Var.
13.06	5.77 ± 4.43	12.01	8.31 ± 2.20	2.97	2.07 ± 0.53

ples reducing the average cost of recourse differ from those positively affecting the model's predictive performance (as identified by Data-SHAP). Thus, existing data valuation methods may be insufficient, stressing the necessity of specialized methods like our proposed Algorithm 1 for identifying training samples that shaped a given explanation.

We observe that random removal typically has little effect, whereas our method and Data-SHAP negatively impact predictive performance – which is to be expected given the fact that (potentially useful) information is removed from the training data. For Data-SHAP, this result aligns with its design and the original paper's findings [13]. Although our method also reduces predictive performance, the drop is less severe than with Data-SHAP. Finally, all effects are getting stronger the more training samples are removed.

6.4 Case Study II

We focus on the German Credit dataset [1] due to its known fairness issues in certain train-test splits [12]. We assess the difference in recourse costs between protected groups for each split and counterfactual explanation method, as shown in Table 1. Significant differences are found for the Nearest Unlike Neighbor (NUN) and prototype-guided counterfactuals, while Wachter's method [29] shows no significant difference. High variance is observed because not all splits exhibit unfairness. We evaluate the three methods on the split with the highest unfairness in recourse cost differences. The results are shown in Figure 2, where for the cost difference smaller numbers indicate better fairness, and larger numbers indicate better F1-scores.

RQ1) For decreasing the difference in the cost of recourse Eq. (10) , we observe that our method almost always achieves better or at least competitive performance with the Data-SHAP method [13]. RQ2) However (similar to case study I), our method consistently maintains a much better predictive performance of the classifier than the Data-SHAP method which often leads to a drastic drop in predictive performance. Random removal of training samples does not affect the F1-score as much as the other two methods do, however, it also does not significantly decrease the difference in the cost of recourse.

Similar to our findings in the first case study, these results demonstrate that our proposed method is able to identify influential training samples (Definition 1) and that those samples do not affect the predictive performance as much as the Data-SHAP method.

7 Summary & Conclusion

We introduced the novel problem of identifying training samples that have a strong impact on given explanations and proposed a Data-SHAP [13]-based algorithm for this purpose. We explored two case studies: the cost of recourse (case study I) and the difference in recourse costs between two protected groups (case study II). Although we focused on the cost of recourse, our methodology and Algorithm 1 are applicable to other explanations as well. Our empirical results show that removing identified samples significantly reduces recourse costs or group disparities without greatly harming predictive performance. Notably, when Data-SHAP [13] competes with our method, it results in a much larger drop in predictive performance, indicating a fundamental difference between training samples affecting computational recourse and those affecting predictive performance.

It would be interesting to extend Algorithm 1 to consider groups of training samples, as they might have a strong influence collectively. This presents the challenge of identifying interacting or dependent groups of samples. Additionally, while our algorithm performs well empirically, it lacks formal guarantees regarding the logit approximation and the effects of removing multiple samples. We leave these aspects for future work.

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.

References

1. Statlog (German Credit Data) Data Set (1994)
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., Sharma, S., Lecué, F., Hammer, B.: The Effect of Data Poisoning on Counterfactual Explanations. arXiv preprint arXiv:2402.08290 (2024)
4. Artelt, A., Vaquet, V., Velioglu, R., Hinder, F., Brinkrolf, J., Schilling, M., Hammer, B.: Evaluating robustness of counterfactual explanations. In: 2021 IEEE Symposium Series on Computational Intelligence. pp. 01–09. IEEE (2021)
5. Baniecki, H., Biecek, P.: Adversarial attacks and defenses in explainable artificial intelligence: A survey. Information Fusion p. 102303 (2024)
6. Baniecki, H., Kretowicz, W., Biecek, P.: Fooling partial dependence via data poisoning. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 121–136. Springer (2022)
7. Byrne, R.M.J.: Counterfactuals in Explainable Artificial Intelligence (XAI): Evidence from Human Reasoning. In: IJCAI-19 (2019)
8. Commission, E., for Communications Networks, D.G., Content, Technology: Proposal for a Regulation laying down harmonised rules on Artificial Intelligence (Artificial Intelligence Act). Policy and Legistlation (21-04-2021)
9. Council of European Union: GDPR – Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 and repealing Directive 95/46/EC. Official Journal of the European Union **L 119**, 4.5 (2016)

10. 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)
11. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression (2004)
12. Friedler, S.A., Scheidegger, C., Venkatasubramanian, S., Choudhary, S., Hamilton, E.P., Roth, D.: A comparative study of fairness-enhancing interventions in machine learning. In: Proceedings of the conference on fairness, accountability, and transparency. pp. 329–338 (2019)
13. Ghorbani, A., Zou, J.: Data shapley: Equitable valuation of data for machine learning. In: International conference on machine learning. pp. 2242–2251. PMLR (2019)
14. Guidotti, R.: Counterfactual explanations and how to find them: literature review and benchmarking. Data Mining and Knowledge Discovery pp. 1–55 (2022)
15. Jiang, J., Leofante, F., Rago, A., Toni, F.: Robust Counterfactual Explanations in Machine Learning: A Survey pp. 8086–8094 (2024), https://www.ijcai.org/proceedings/2024/894
16. Karimi, A.H., Barthe, G., Schölkopf, B., Valera, I.: A survey of algorithmic recourse: contrastive explanations and consequential recommendations. ACM Computing Surveys (2021)
17. Looveren, A.V., Klaise, J.: Interpretable counterfactual explanations guided by prototypes pp. 650–665 (2021)
18. Mothilal, R.K., Sharma, A., Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. In: Proceedings of the 2020 conference on fairness, accountability, and transparency (2020)
19. Poyiadzi, R., Sokol, K., Santos-Rodriguez, R., De Bie, T., Flach, P.: FACE: feasible and actionable counterfactual explanations. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society. pp. 344–350 (2020)
20. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In: 22Nd ACM SIGKDD. pp. 1135–1144. ACM, New York, NY, USA (2016)
21. 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)
22. Sharma, S., Gee, A.H., Paydarfar, D., Ghosh, J.: FaiR-N: Fair and Robust Neural Networks for Structured Data. In: Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society (2021)
23. Sharma, S., Henderson, J., Ghosh, J.: 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 (2020)
24. Sim, R.H.L., Xu, X., Low, B.K.H.: Data Valuation in Machine Learning: "Ingredients", Strategies, and Open Challenges. In: IJCAI. pp. 5607–5614 (2022)
25. Slack, D., Hilgard, A., Lakkaraju, H., Singh, S.: Counterfactual explanations can be manipulated. Advances in Neural Information Processing Systems **34** (2021)
26. Sundararajan, M., Najmi, A.: The many Shapley values for model explanation. In: International conference on machine learning. pp. 9269–9278. PMLR (2020)
27. Verma, S., Dickerson, J., Hines, K.: Counterfactual Explanations for Machine Learning: A Review (2020)
28. Von Kügelgen, J., Karimi, A.H., Bhatt, U., Valera, I., Weller, A., Schölkopf, B.: On the fairness of causal algorithmic recourse. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 9584–9594 (2022)

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)
30. 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)