Analyzing the Influence of Training Samples on Explanations

André Artelt[1, 2], **Barbara Hammer**[1]

[1]Bieled University, Germany
[2]University of Cyprus, Cyprus
aartelt@techfak.uni-bielefeld.de, bhammer@techfak.uni-bielefeld.de

Abstract

EXplainable AI (XAI) constitutes a popular method to analyze the reasoning of AI systems by explaining their decision-making, e.g. providing a counterfactual explanation of how to achieve recourse. However, in cases such as unexpected explanations, the user might be interested in learning about the cause of this explanation – e.g. properties of the utilized training data that are responsible for the observed explanation.

Under the umbrella of data valuation, first approaches have been proposed that estimate the influence of data samples on a given model. In this work, we take a slightly different stance, as we are interested in the influence of single training samples on a model explanation rather than the model itself. Hence, we propose the novel problem of identifying training data samples that have a high influence on a given explanation (or related quantity) and investigate the particular case of the cost of computational recourse. For this, we propose an algorithm that identifies such influential training samples.

Code — https://github.com/andreArtelt/AnalyzingInfluenceTrainingSamplesExplanations

Introduction

Nowadays, many Artificial Intelligence (AI-) and Machine Learning (ML-) based systems are deployed in the real world (Zhao et al. 2023; Ho et al. 2022). These systems show an impressive performance but are still not perfect – e.g. failures, issues of fairness, and vulnerability to manipulations such as data poisoning can cause harm when applied in the real world.

Given the threat of failures and other issues, transparency of such deployed AI- and ML-based systems becomes a crucial aspect. Transparency is important not only to prevent failures but also to create trust in such systems and understand where and how it is safe to deploy them. The importance of transparency was also recognized by the policymakers and therefore found its way into legal regulations such as the EU's GDPR (Council of European Union 2016) or the more recent EU AI act (Commission et al. 21-04-2021). Explanations are a popular way of achieving transparency and shaping the field of eXplainable AI (XAI) (Dwivedi et al. 2023). Because of many different use cases, domains, and users, many different explanation methods exist (Dwivedi et al. 2023; Rawal et al. 2021). Popular instances of XAI methods are LIME (Ribeiro, Singh, and Guestrin 2016), SHAP (Sundararajan and Najmi 2020), and counterfactual explanations (Wachter, Mittelstadt, and Russell 2017) that yield computational recourse.

However, in cases of implausible or unexpected explanations such as unexpected attention/focus of the AI to some input regions (e.g. 'Clever Hans' effect (Kauffmann et al. 2020; Anders et al. 2022)), the user might be interested in an "explanation of the explanation". The need for methods to understand the cause of explanations is also emphasized by recent findings that explanations are vulnerable to manipulations and poisonings (Baniecki, Kretowicz, and Biecek 2022; Baniecki and Biecek 2024; Artelt et al. 2024), potentially harming the users' trust in the explanations. One possibility to analyze and understand the cause of an explanation would be to trace it back to the training samples – i.e. identifying training samples that have a strong influence on (i.e. caused) the observed explanation (see Figure 1 for an illustration). Those training samples could provide additional insights into the system and can be then sanity checked, removed, or corrected in case of errors. Furthermore, such an approach for identifying influential training samples would explain the internal reasoning of a model (e.g. classifier) in the training data space and could extend the traditional XAI methods that provide explanations in the feature space or model parameter space only. 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*. We investigate two particular cases of counterfactual explanations for computational recourse: 1) Identifying training samples that have a strong influence on the average cost of recourse; 2) Identifying training samples that have a strong influence on the difference in the cost of recourse between two protected groups, i.e. violation of group fairness of the given counterfactual. We also conduct extensive empirical evaluations where we evaluate our proposed algorithms and compare them to baseline approaches.

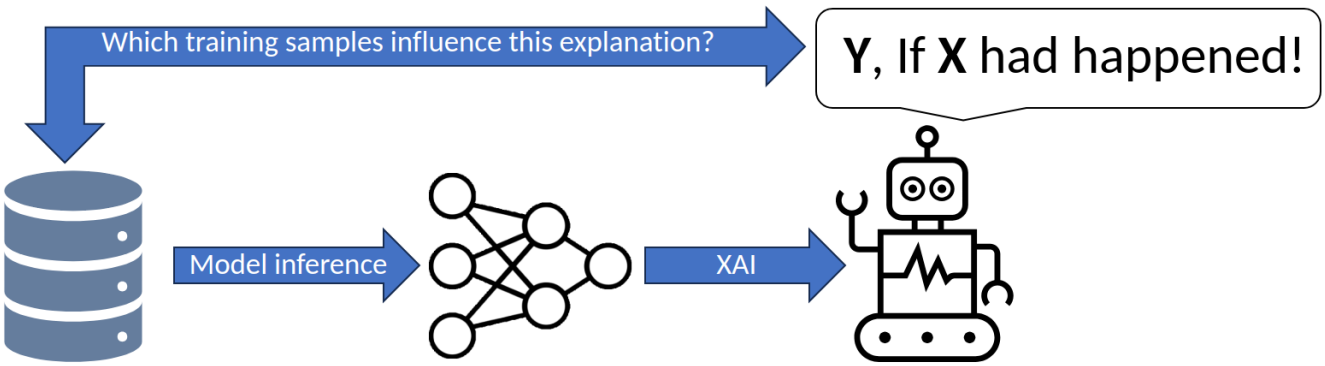

Figure 1: Illustration of our research question – i.e. we aim to identify training samples that have a high influence on an explanation.

Foundations

Data-Valuation

The young field of data-valuation (Sim, Xu, and Low 2022) is concerned with assessing the value/importance of individual training samples on the predictive performance – i.e. quantifying the importance of each sample to the predictive performance of the final trained model. Such knowledge could for instance be used to pay users for utilizing their data, sanity check (the labeling of) highly relevant training samples, or to acquire more data samples that are similar to the most relevant ones.

In this context, the Data-SHAP method (Ghorbani and Zou 2019) carries the concept of Shapley-Values (Sundararajan and Najmi 2020) over to data valuation by computing the influence $\phi_i \in \mathbb{R}$ of each training sample $(\vec{x}_i, y_i) \in \mathcal{S}$ on a value function $V : \mathcal{S} \mapsto \mathbb{R}$. This value function $V(\cdot)$ states the property of interest of a predictive function $h : \mathcal{X} \to \mathcal{Y}$ derived from the given training data $\mathcal{S}$. In data valuation, as stated before, the property of interest is usually the predictive performance of $h(\cdot)$, e.g. the accuracy. Requiring some equitable properties, it can be shown (Ghorbani and Zou 2019) 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 (Ghorbani and Zou 2019) a Monte-Carlo approximation of Eq. (1) is proposed:

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

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 (Ghorbani and Zou 2019) 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 $h(\cdot)$.

Counterfactuals for Computational Recourse

A counterfactual explanation (often just called counterfactual) states (actionable) changes to the features of a given input such that the system's output changes. Usually, an explanation is requested in the case of an unfavorable outcome (Riveiro and Thill 2022) – in the latter case, a counterfactual is also referred to as (computational) *recourse* (Karimi et al. 2021), i.e. recommendations on how to change the unfavorable into a favorable outcome. Because counterfactuals can mimic ways in which humans explain (Byrne 2019), they constitute among one of the most popular explanation methods in literature and in practice (Verma, Dickerson, and Hines 2020).

When computing a counterfactual $\vec{\delta}_{\text{cf}}$ (Wachter, Mittelstadt, and Russell 2017), it must be ensured that 1) it indeed changes the output of the system; and 2) the cost of $\vec{\delta}_{\text{cf}}$ – i.e. the cost and effort it takes to execute the counterfactual in the real world should be kept to a minimum. The computation of a counterfactual $\vec{\delta}_{\text{cf}} \in \mathbb{R}^d$ for a given instance $\vec{x}_{\text{orig}} \in \mathbb{R}^d$ is phrased as the following optimization problem:

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

where $\ell(\cdot)$ penalizes deviation of the prediction $h(\vec{x}_{\text{cf}} := \vec{x}_{\text{orig}} + \vec{\delta}_{\text{cf}})$ 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 the two properties. The short-hand notation $\text{CF}(\vec{x}, h)$ denotes the counterfactual $\vec{\delta}_{\text{cf}}$ of an instance $\vec{x}$ under a classifier $h(\cdot)$ iff the target outcome y_{cf} is uniquely determined. Note that the cost $\theta(\cdot)$ is highly domain and use-case specific. However, in many implementations and toolboxes (Guidotti 2022), the p-norm is used as a default.

Remark 1. *In the case of recourse – i.e. a counterfactual $\vec{\delta}_{cf}$ for turning an unfavorable into a favorable outcome –, we refer to the cost $\theta(\vec{\delta}_{cf})$, 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 properties in Eq. (3), there exist additional

relevant aspects such as plausibility (Looveren and Klaise 2021; Poyiadzi et al. 2020), diversity (Mothilal, Sharma, and Tan 2020), robustness (Artelt et al. 2021; Jiang et al. 2024), etc. which have been addressed in literature (Guidotti 2022). However, the basic formalization Eq. (3) is still very popular and widely used in practice (Verma, Dickerson, and Hines 2020; Guidotti 2022).

A critical and still unsolved fairness issue in computational recourse is the difference in the cost of recourse $\theta(\vec{\delta}_{\text{cf}})$ between protected groups (Artelt and Hammer 2023; Von Kügelgen et al. 2022; Sharma et al. 2021) – 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 (Artelt et al. 2024; Slack et al. 2021).

There exist numerous methods and implementations/toolboxes for computing counterfactual explanations in practice (Guidotti 2022) – most include some additional aspects such as plausibility, diversity, etc. : *Counterfactuals Guided by Prototypes* (Looveren and Klaise 2021) focuses on plausibility. Here a set of plausible instances (so-called prototypes) are used to pull the final counterfactual instance (i.e. $\vec{x}_{\text{orig}} + \vec{\delta}_{\text{cf}}$) 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 $\vec{x}_{\text{cf}}$.

Influence of Training Samples on Explanations

We consider settings, where we have a predictive model $h : \mathcal{X} \to \mathcal{Y}$, derived from a training data set $\mathcal{D}_{\text{train}}$, together with an explanation $z \in \mathcal{E}$ of $h(\cdot)$ – we use the shorthand notation $Z(h) \mapsto z \in \mathcal{E}$ to denote the explanation generation method. The explanation z could be any type of explanation, e.g. a counterfactual explanation, or some kind or property of an explanation such as the cost of recourse in the case of a counterfactual. *We are interested in identifying training samples that have a high influence on the explanation z – e.g. outlier or malicious training samples that "caused" the observed explanation z.*

To be able to compute the influence (a scalar) of a single training sample on an explanation, we introduce a function $\Psi : \mathcal{E} \times \mathcal{E} \to \mathbb{R}$ that computes the similarity of two given explanations $z_1, z_2 \in \mathcal{E}$ of the same type, whereby larger absolute numbers correspond to a larger dissimilarity – the sign of the similarity may or may not provide additional insights. 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 a p-norm: $\Psi(z_1, z_2) = \|z_1 - z_2\|_p$ or the cosine similarity $\Psi(z_1, z_2) = \frac{z_1^\top z_2}{\|z1\|\|z1\|}$. In the context of counterfactual explanations, the first instantiation would compare the costs of the counterfactuals, while the latter one compares the feature overlap.

Next, we can formally define and characterize training samples that have a strong influence on an explanation.

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 z iff the absence of $\mathcal{D}_{infl}$ changes the explanation z significantly:*

$$|\Psi(z_{\mathcal{D}_{train}}, z_{\mathcal{D}_{train} \setminus \mathcal{D}_{infl}})| >> 0 \tag{4}$$

where $z_{\mathcal{D}_?}$ refers to the explanation of $h(\cdot)$ trained on $\mathcal{D}_?$.

In our running example of counterfactuals, Definition 1 together with $\Psi(z_1, z_2) = \|z_1 - z_2\|_p$ (for some fixed instance $\vec{x}$ of interest) would mean that $\mathcal{D}_{\text{infl}}$ significantly affects the cost of the counterfactual of $\vec{x}$ – i.e. either increasing or decreasing it significantly. Alternatively, we could also consider multiple instances $\vec{x}_i$ and merge their corresponding counterfactuals into a single explanation z – this would allow us to specify the influence of training samples on multiple explanations.

In this work, we propose to find such influential training samples (Definition 1) by conceptualizing the Data-SHAP method (Ghorbani and Zou 2019) from the data valuation literature. We aim to identify training instances that have a strong influence on the explanation z of a given $h(\cdot)$. For this, we substitute the difference of the two value functions in Eq. (2) with Eq. (4) from Definition 1:

$$V(\mathcal{D} \cup \{i\}) - V(\mathcal{D}) := \Psi(z_{\mathcal{D} \cup \{i\}}, z_{\mathcal{D}}) \tag{5}$$

This allows us to use Eq. (2) to compute the influence score ϕ_i of each training sample in $\mathcal{D}$.

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 as done in the original Data-SHAP paper (Ghorbani and Zou 2019) – 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 $\vec{w}$ and trained for K iterations. In each iteration, the entire training data set $\mathcal{D}$ is shuffled (line 5) and then each training sample $(\vec{x}_i, y_i)$ is processed as follows: 1) update the model parameters $\vec{w}$ using gradient descent (line 8); 2) Recompute the explanation of interest (line 9); 3) Compare current explanation with the explanation from the previous iteration (line 10) – this allows us to estimate the influence of the current training sample $(\vec{x}_i, y_i)$ on the explanation of interest. The training procedure is repeated several times until the influence scores ϕ_i converged. 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):

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

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

Algorithm 1: Finding Influential Training Samples

Input: Labeled training samples $\mathcal{D}$; K number of training iterations; Differentiable model $h(\cdot)$

Output: Influence score ϕ_i of each training sample.

for $j = 1, ..., N$ or until convergence **do**
  Random initialization of $h(\cdot)$ weights $\vec{w}$
  for $t = 1, ..., K$ **do** ▷ Training loop
    $z_{\mathcal{D}} = Z(h)$ ▷ Initial explanation
    π = random_permutation($|\mathcal{D}|$)
    for idx $\in \pi$ **do**
      $(\vec{x}_i, y_i) = \mathcal{D}$[idx] ▷ Get current sample
      $\vec{w} -= \lambda \nabla_{\vec{w}} \ell(\vec{x}_i, y_i \mid \vec{w})$ ▷ Gradient descent
      $z_{\mathcal{D} \cup \{i\}} = Z(h)$ ▷ Compute new explanation
      $\phi^{j,t}_{[\text{idx}]} = \Psi(z_{\mathcal{D} \cup \{i\}}, z_{\mathcal{D}})$ ▷ Compute influence of $(\vec{x}_i, y_i)$
      $z_{\mathcal{D}} = z_{\mathcal{D} \cup \{i\}}$ ▷ Update current explanation
    end for
  end for
end for
$\phi_i = \frac{1}{N \cdot K} \sum_{j,t} \phi_i^{j,t}$ ▷ Final influence scores

Case-Study I: Cost of Recourse

Here, we consider the challenge of identifying influential training samples (see Algorithm 1) on the average cost of recourse – i.e. the average cost it takes to change the outcome for negatively 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 $\mathcal{D}$, the average cost of recourse is defined as follows:

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

Note that the training data set $\mathcal{S}$ and the 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 $\text{CF}(\vec{x}_i, h_{\mathcal{S}})$ 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 (Sharma et al. 2021):

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

where g_1, g_2 denote the logits of the neural network $h_{\mathcal{S}}(\cdot)$ The authors of (Sharma et al. 2021) show that Eq. (8) constitutes a good approximation of the distance to the decision boundary.

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

$$c_{\mathcal{S}} = \frac{1}{N_{\mathcal{S}}} \sum_{\vec{x}_i \in \mathcal{D} | h_{\mathcal{S}}(\vec{x}_i) = 0} \Big| g_0(\vec{x}_i) - g_1(\vec{x}_i) \Big| \tag{9}$$

where we only consider negatively classified samples in a given set $\mathcal{D}$ – the number of those negative classified samples is denoted by $N_{\mathcal{S}} := |\{\vec{x}_i \in \mathcal{D} \mid h_{\mathcal{S}}(\vec{x}_i) = 0\}|$.

The function $\Psi(\cdot)$ for evaluating the difference between two explanations (here the average cost of recourse) as follows:

$$\begin{aligned} &\Psi(z_{\mathcal{S} \cup \{i\}}, z_{\mathcal{S}}) := c_{\mathcal{S} \cup \{i\}} - c_{\mathcal{S}} \text{ where} \\ &c_{\mathcal{S} \cup \{i\}} = \frac{1}{N_{\mathcal{S} \cup \{i\}}} \sum_{\vec{x}_i \in \mathcal{D} \cup \{i\} | h_{\mathcal{S} \cup \{i\}}(\vec{x}_i) = 0} \Big| g_0(\vec{x}_i) - g_1(\vec{x}_i) \Big| \\ &c_{\mathcal{S}} = \frac{1}{N_{\mathcal{S}}} \sum_{\vec{x}_i \in \mathcal{D} | h_{\mathcal{S}}(\vec{x}_i) = 0} \Big| g_0(\vec{x}_i) - g_1(\vec{x}_i) \Big| \end{aligned} \tag{10}$$

Note that we consider logits of $h_{\mathcal{S} \cup \{i\}}(\cdot)$ in $c_{\mathcal{S} \cup \{i\}}$, and of $h_{\mathcal{S}}(\cdot)$ in $c_{\mathcal{S}}$ respectively – we omitted a specific notation for improved readability.

We can then substitute Eq. (10) in Algorithm 1 for computing the influence of each training sample. The training samples of interest to us are those with a large positive influence score, i.e. $\phi_i >> 0$, since those are increasing the average cost of recourse Eq. (7) when added to the training set. In the empirical evaluation, we investigate the effect of removing those training samples from the training data.

Case-Study II: Difference in the Cost of Recourse

As another particular instantiation of our proposed method for identifying influential training samples (see Algorithm 1), we consider the emerging problem of analyzing the reasons for the difference in the cost of recourse between two protected groups which constitutes a major fairness issue in computational recourse (Artelt and Hammer 2023; Sharma, Henderson, and Ghosh 2020). That is, we aim to identify those training samples that have an increasing effect on the difference in the cost of recourse between two protected groups – those training samples might be suitable candidates for manual inspection and (potential) deletion in order to decrease the difference in the cost of recourse between two protected groups and thereby making the computational recourse more fair.

Here, we consider the difference in the worst-case cost of recourse between two protected groups, which we formalize as follows:

$$\left| \max_{\vec{x}_i \in \mathcal{D}} (\theta \circ \text{CF}(\vec{x}_i \mid s = 0, h_{\mathcal{S}})) - \max_{\vec{x}_i \in \mathcal{D}} (\theta \circ \text{CF}(\vec{x}_i \mid s = 1, h_{\mathcal{S}})) \right| \tag{11}$$

where $s =?$ denotes the protected attribute, and we also only consider negatively classified samples in the given set $\mathcal{D}$ – we drop the explicit constraint $h_{\mathcal{S}}(\vec{x}_i) = 0$ for better readability. Consequently, we define the value function $V(\cdot)$ to be equal to Eq. (11).

As in the first case study, we approximate the cost of recourse by the difference in the logits:

$$\theta \circ \text{CF}(\vec{x}_i \mid s =?, h_{\mathcal{S}})) \approx |g_0(\vec{x}_i) - g_1(\vec{x}_i)| \tag{12}$$

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

$$c_{\mathcal{S}} = \Big| \max_{\vec{x}_i \in \mathcal{D}|s=0} (|g_0(\vec{x}_i) - g_1(\vec{x}_i)|) - \max_{\vec{x}_i \in \mathcal{D}|s=1} (|g_0(\vec{x}_i) - g_1(\vec{x}_i)|) \Big| \tag{13}$$

We define the difference $\Psi(\cdot)$ between two explanations (here the difference in the cost of recourse) as follows:

$$\begin{aligned} &\Psi(z_{\mathcal{S}\cup\{i\}}, z_{\mathcal{S}}) := c_{\mathcal{S}\cup\{i\}} - c_{\mathcal{S}} \\ &\text{where} \\ &c_{\mathcal{S}\cup\{i\}} = \Big| \max_{\vec{x}_i \in \mathcal{D}|s=0} (|g_0(\vec{x}_i) - g_1(\vec{x}_i)|) - \\ &\qquad \max_{\vec{x}_i \in \mathcal{D}|s=1} (|g_0(\vec{x}_i) - g_1(\vec{x}_i)|) \Big| \\ &c_{\mathcal{S}} = \Big| \max_{\vec{x}_i \in \mathcal{D}|s=0} (|g_0(\vec{x}_i) - g_1(\vec{x}_i)|) - \\ &\qquad \max_{\vec{x}_i \in \mathcal{D}|s=1} (|g_0(\vec{x}_i) - g_1(\vec{x}_i)|) \Big| \end{aligned} \tag{14}$$

Note that we consider logits of $h_{\mathcal{S}\cup\{i\}}(\cdot)$ in $c_{\mathcal{S}\cup\{i\}}$, and of $h_{\mathcal{S}}(\cdot)$ in $c_{\mathcal{S}}$ respectively – we omitted a specific notation for improved readability.

We can then substitute Eq. (14) 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. $\phi_i >> 0$, since those are increasing the difference in the cost of recourse Eq. (11) when added to the training data set. In the empirical evaluation, we investigate the effect of removing those training samples from the training data set.

Experiments

We empirically evaluate our proposed Algorithm 1, for identifying influential training samples, separately for each of the two case studies and evaluate how the deletion of those training samples influences the counterfactuals, as well as the predictive performance of the classifier.

Data

For both case studies, we consider the following two benchmark data sets from the fairness literature (Friedler et al. 2019):

- The "Diabetes" data set (Efron et al. 2004) (denoted as *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.
- The "German Credit Data set" (Ger 1994) (denoted as *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.

Setup

The general setup is the same for both case studies:

We use the ℓ_1 norm as a popular implementation (Guidotti 2022) of the cost of recourse – i.e. $\theta(\cdot) = \|\cdot\|_1$. For all data sets, we utilize a Deep Neural Network as the classifier $h(\cdot)$.

We conduct the experiments in a 5-fold cross-validation: After fitting the classifier, we compute the most influential training samples using our proposed method, whereby all negative classified samples (i.e. $h(\cdot) = 0$) from the test set are considered when computing the computational recourse.

We compute the computational recourse (i.e. counterfactual explanations) using different popular recourse methods: Nearest Unlike Neighbor Sample (denoted as *NUN*), as simple baseline for plausible counterfactuals; Counterfactuals guided by Prototypes (Looveren and Klaise 2021) (denoted as *Proto*) as an advanced method for computing plausible counterfactuals; Classic counterfactuals (Wachter, Mittelstadt, and Russell 2017) (denoted as *Wachter*) by solving Eq. (3).

To evaluate the results, we increasingly remove (1% - 30%) of the most influential training samples from the training set, retrain the classifier, and evaluate the quantity of interest – i.e. either the average cost of recourse (case study I) or the difference in the cost of recourse (case study II). We report the mean as well as the variance of all measurements.

Baselines 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 (Ghorbani and Zou 2019).

Case Study I

The results for all combinations of data set and recourse method are shown in Figure 2. Note that for the average cost of recourse, smaller numbers are better, while for the F1-score (predictive performance), larger numbers are better.

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. The other two baselines (random removal and Data-SHAP (Ghorbani and Zou 2019)) return training samples that, if removed, often increase, instead of decrease, the average cost of recourse. This demonstrates that training samples that have a decreasing effect on the average cost of recourse differ from those that have a positive influence on the predictive performance of the model (i.e. those identified by Data-SHAP).

Regarding the effect on the predictive performance of the model, we observe that random removal usually does not have any effect at all, while our proposed method, as well as Data-SHAP (Ghorbani and Zou 2019)), have a negative impact on the predictive performance. In the case of Data-SHAP, this is to be expected since this is exactly what the method was built for and is consistent with the results reported in the original paper (Ghorbani and Zou 2019)). While our proposed method also decreases the predictive performance, the drop in predictive performance is less severe compared to Data-SHAP.

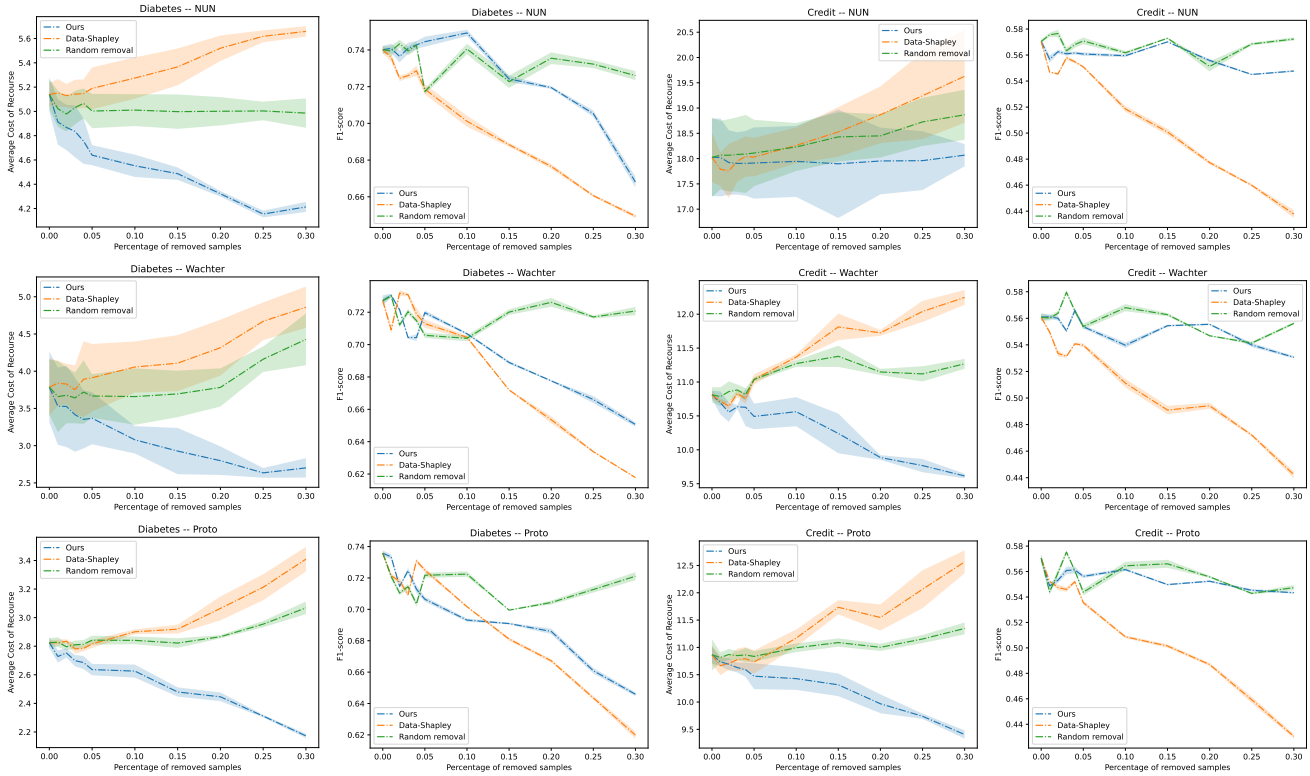

Figure 2: Effect of removing training samples that have a high influence on the average cost of recourse – we show (mean & variance over all folds) the effect on the average cost of recourse, as well as on the predictive performance (i.e. F1-score).

Both effects are getting stronger the more training samples are removed.

Case Study II

The results for all combinations of data set and recourse method are shown in Figure 3. Note that smaller numbers are better for the difference in the cost of recourse (i.e. unfairness), whereas larger numbers are better for the F1-score.

First of all, we observe a large variance in the initial difference in the cost of recourse Eq. (11) – i.e. it differs significantly between different train-test splits. Nevertheless, there are still clear trends visible in the mean of the computed measurements.

For decreasing the difference in the cost of recourse Eq. (11) – i.e. improving group fairness in computational recourse –, we observe that our method almost always achieves a bit better or at least competitive performance with the Data-SHAP method (Ghorbani and Zou 2019). However, 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 recourse cost.

Similar to our finding 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 too much as compared to the Data-SHAP method.

Summary & Conclusion

In this work, we introduced and formalized the novel problem of identifying training samples that have a strong influence on an explanation. We proposed an algorithm to identify such influential training samples. Furthermore, we studied the particular cases of cost of recourse (case study I) and the difference in the cost of recourse between two protected groups (case study II). We empirically evaluated that removing those identified training samples, indeed reduces the average cost of recourse or, respectively, the difference between two protected groups significantly without hurting the predictive performance too much. In particular, we observed that in the cases where Data-SHAP (Ghorbani and Zou 2019) is competitive with our proposed Algorithm 1, it comes with a much larger drop in predictive performance compared to our method. This suggests that there is a difference between training samples affecting the computational recourse (i.e. counterfactuals) and training samples affecting the predictive performance.

In this work, we mainly focused on the cost of recourse as the quantity of interest, however, the proposed methodology and Algorithm 1 are more general and can be applied to many other explanations as well.

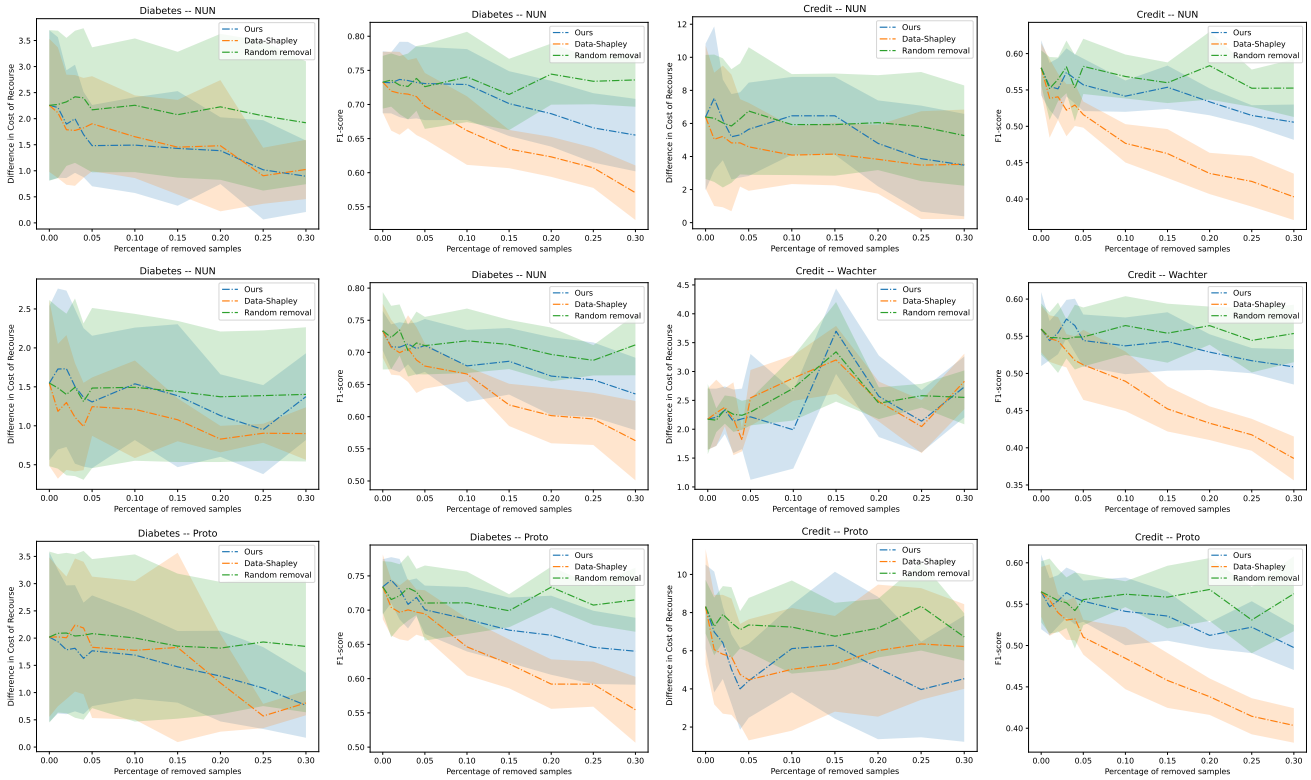

Figure 3: Effect of removing training samples that have a high influence on the difference in the cost of recourse Eq. (11) – we show (mean & variance over all folds) the effect on the difference in the cost of recourse, as well as on the predictive performance (i.e. F1-score).

Based on this initial work, there exist a couple of directions for future work: Because groups of samples might have a strong influence only together, it would be interesting to extend Algorithm 1 to consider groups of training samples instead of individual samples. Here the challenge of finding such groups of interacting/dependent training samples arises. Furthermore, although our proposed algorithm shows good performance in the empirical evaluation, it misses formal guarantees when it comes to the proposed approximation using the logits Eq. (8) as well as the results when removing more than a single training sample. We leave these aspects as future work.

References

1994. Statlog (German Credit Data) Data Set.

Anders, C. J.; Weber, L.; Neumann, D.; Samek, W.; Müller, K.-R.; and Lapuschkin, S. 2022. Finding and removing clever hans: Using explanation methods to debug and improve deep models. *Information Fusion*, 77: 261–295.

Artelt, A.; and Hammer, B. 2023. "Explain it in the Same Way!" – Model-Agnostic Group Fairness of Counterfactual Explanations. In Ofra, A.; Miller, T.; and Baier, H., eds., *Workshop on XAI*.

Artelt, A.; Sharma, S.; Lecué, F.; and Hammer, B. 2024. The Effect of Data Poisoning on Counterfactual Explanations. *arXiv preprint arXiv:2402.08290*.

Artelt, A.; Vaquet, V.; Velioglu, R.; Hinder, F.; Brinkrolf, J.; Schilling, M.; and Hammer, B. 2021. Evaluating robustness of counterfactual explanations. In *2021 IEEE Symposium Series on Computational Intelligence*, 01–09. IEEE.

Baniecki, H.; and Biecek, P. 2024. Adversarial attacks and defenses in explainable artificial intelligence: A survey. *Information Fusion*, 102303.

Baniecki, H.; Kretowicz, W.; and Biecek, P. 2022. Fooling partial dependence via data poisoning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 121–136. Springer.

Byrne, R. M. J. 2019. Counterfactuals in Explainable Artificial Intelligence (XAI): Evidence from Human Reasoning. In *IJCAI-19*.

Commission, E.; for Communications Networks, D.-G.; Content; and Technology. 21-04-2021. Proposal for a Regulation laying down harmonised rules on Artificial Intelligence (Artificial Intelligence Act) and amending certain Union legislative acts. *Policy and Legistlation*.

Council of European Union. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC. *Official Journal of the European Union*, L 119: 4.5.

Dwivedi, R.; Dave, D.; Naik, H.; Singhal, S.; Omer, R.; Patel, P.; Qian, B.; Wen, Z.; Shah, T.; Morgan, G.; et al. 2023. Explainable AI (XAI): Core ideas, techniques, and solutions. *ACM Computing Surveys*, 55(9): 1–33.

Efron, B.; Hastie, T.; Johnstone, I.; and Tibshirani, R. 2004. Least angle regression.

Friedler, S. A.; Scheidegger, C.; Venkatasubramanian, S.; Choudhary, S.; Hamilton, E. P.; and Roth, D. 2019. A comparative study of fairness-enhancing interventions in machine learning. In *Proceedings of the conference on fairness, accountability, and transparency*, 329–338.

Ghorbani, A.; and Zou, J. 2019. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*, 2242–2251. PMLR.

Guidotti, R. 2022. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery*, 1–55.

Ho, J.; Saharia, C.; Chan, W.; Fleet, D. J.; Norouzi, M.; and Salimans, T. 2022. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research*, 23(1): 2249–2281.

Jiang, J.; Leofante, F.; Rago, A.; and Toni, F. 2024. Robust counterfactual explanations in machine learning: A survey. *arXiv preprint arXiv:2402.01928*.

Karimi, A.-H.; Barthe, G.; Schölkopf, B.; and Valera, I. 2021. A survey of algorithmic recourse: contrastive explanations and consequential recommendations. *ACM Computing Surveys*.

Kauffmann, J.; Ruff, L.; Montavon, G.; and Müller, K.-R. 2020. The clever Hans effect in anomaly detection. *arXiv preprint arXiv:2006.10609*.

Looveren, A. V.; and Klaise, J. 2021. Interpretable counterfactual explanations guided by prototypes. 650–665.

Mothilal, R. K.; Sharma, A.; and Tan, C. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, 607–617.

Poyiadzi, R.; Sokol, K.; Santos-Rodriguez, R.; De Bie, T.; and Flach, P. 2020. FACE: feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 344–350.

Rawal, A.; Mccoy, J.; Rawat, D. B.; Sadler, B.; and Amant, R. 2021. Recent advances in trustworthy explainable artificial intelligence: Status, challenges and perspectives. *IEEE Transactions on Artificial Intelligence*, 1(01): 1–1.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *22Nd ACM SIGKDD*, 1135–1144. New York, NY, USA: ACM. ISBN 978-1-4503-4232-2.

Riveiro, M.; and Thill, S. 2022. 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*, 110–120.

Sharma, S.; Gee, A. H.; Paydarfar, D.; and Ghosh, J. 2021. FaiR-N: Fair and Robust Neural Networks for Structured Data. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, 946–955.

Sharma, S.; Henderson, J.; and Ghosh, J. 2020. 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*, 166–172.

Sim, R. H. L.; Xu, X.; and Low, B. K. H. 2022. Data Valuation in Machine Learning:" Ingredients", Strategies, and Open Challenges. In *IJCAI*, 5607–5614.

Slack, D.; Hilgard, A.; Lakkaraju, H.; and Singh, S. 2021. Counterfactual explanations can be manipulated. *Advances in Neural Information Processing Systems*, 34: 62–75.

Sundararajan, M.; and Najmi, A. 2020. The many Shapley values for model explanation. In *International conference on machine learning*, 9269–9278. PMLR.

Verma, S.; Dickerson, J.; and Hines, K. 2020. Counterfactual Explanations for Machine Learning: A Review. arXiv:2010.10596.

Von Kügelgen, J.; Karimi, A.-H.; Bhatt, U.; Valera, I.; Weller, A.; and Schölkopf, B. 2022. On the fairness of causal algorithmic recourse. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 9584–9594.

Wachter, S.; Mittelstadt, B.; and Russell, C. 2017. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. JL & Tech.*, 31: 841.

Zhao, W. X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.