Contrasting Explanations for Understanding and Regularizing Model Adaptations

André Artelt[1*], Fabian Hinder[1], Valerie Vaquet[1], Robert Feldhans[1] and Barbara Hammer[1]

[1]Faculty of Technology, Bielefeld University, Universitätsstrasse 25, Bielefeld, 33615, NRW, Germany.

*Corresponding author(s). E-mail(s): aartelt@techfak.uni-bielefeld.de;
Contributing authors: fhinder@techfak.uni-bielefeld.de; vvaquet@techfak.uni-bielefeld.de; rfeldhans@techfak.uni-bielefeld.de; bhammer@techfak.uni-bielefeld.de;

Abstract

Many of today's decision making systems deployed in the real world are not static – they are changing and adapting over time, a phenomenon known as model adaptation takes place. Because of their wide reaching influence and potentially serious consequences, the need for transparency and interpretability of AI-based decision making systems is widely accepted and thus have been worked on extensively – e.g. a very prominent class of explanations are contrasting explanations which try to mimic human explanations. However, usually, explanation methods assume a static system that has to be explained. Explaining non-static systems is still an open research question, which poses the challenge how to explain model differences, adaptations and changes. In this contribution, we propose and (empirically) evaluate a general framework for explaining model adaptations and differences by contrasting explanations. We also propose a method for automatically finding regions in data space that are affected by a given model adaptation – i.e. regions where the internal reasoning of the other (e.g. adapted) model changed – and thus should be explained. Finally, we also propose a regularization for model adaptations to ensure that the internal reasoning of the adapted model does not change in an unwanted way.

Keywords: XAI, Contrasting Explanations, Model Adaptation, Human-Centered AI

1 Introduction

Machine learning (ML) and artificial intelligence (AI) based decision making systems are increasingly affecting our daily life – e.g. predictive policing [1] and loan approval [2, 3]. Given the impact of many ML and AI based decision making systems, there is an increasing demand for transparency and interpretability [4] – the importance of these aspects have been also emphasized by legal regulations like the EU's "General Data Protection Right" (GDPR) [5] that contain a "right to an explanation". In the context of transparency and interpretability, fairness and other ethical aspects become relevant as well [6, 7].

As a consequence, the research community extensively worked on these aspects and came up with methods for explaining ML and AI based decision making systems and thus meeting the demands for transparency and interpretability [8–10]. A popular class of explanations methods [9, 11] highlight relevant features of a given model or model decision, such as feature interaction methods [12], feature importance methods [13], partial dependency plots [14], and local methods that approximates the model locally by an explainable model (e.g. a decision tree) [15]. These methods explain the models by using features as vocabulary. Another class of explanation methods are examples based methods [16]. Instances of example based methods are counterfactual explanations [17, 18] and prototypes & criticisms [19] – these methods use a set or a single example for explaining the behavior of the system. Counterfactual and contrastive explanations in general (also called contrasting explanations) [11, 17, 18] are popular instances of example based methods for locally explaining decision making systems [11] – popularity comes from the fact there exists strong evidence that explanations by humans (which these methods try to mimic) are often counterfactual in nature [20]. While some of these methods are global methods – i.e. explaining the system globally – most of the example based methods are local methods that try to explain the behavior of the decision making system at a particular instance or in a "small" region in data space [11, 15, 21, 22]. Further, existing explanation methods can be divided into model agnostic and model specific methods. While model agnostic methods view the decision making system as a black-box and thus do not need access to any model internals, model specific methods rely and usually exploit model internal structures and knowledge for computing the explanation. However, distinguishing between model agnostic and model specific methods is not that strict because there exist model specific methods that aim for efficiently computing (initially model agnostic) explanations of specific models [23].

The majority of the proposed explanation methods in literature assume fixed models – i.e. explaining the decisions of a fixed decision making system. However, in practice decision making systems are usually not fixed but evolving

over time – e.g. the system is adapted or fine tuned on new data [24], or replaced by a completely new and different model. In this context, it is relevant to explain the changes of the decision making system [1] – in particular in the context of Human-Centered AI (HCAI) which, besides explainability, is another important component [2] in ethical AI [28]. HCAI allows the human to "rule" AI systems instead of being ruled (e.g. "discriminated" or "cheated") by AI. Given the complexity of many modern ML or AI systems (e.g. Deep Neural Networks), it is usually difficult for a human to understand the reasoning of a decision making system or the impact of adaptations applied to a given system. Yet, understanding the impact of such changes is crucial for mediating changes that violate the user expectations, some (ethical) guidelines or (legal) constraints.

For example, consider the scenario of a complex loan approval system that automatically accepts or rejects loan applications, and assume that the decision making process of this system is difficult to inspect from the outside: *Consider the rejection of a loan application with the argument of a low income and a bad payback rate in the past – this latter explanation perfectly meets the bank internal guidelines for accepting or rejecting a loan. However, after adapting the loan approval system on new data, it could happen that the same application that was rejected under the "old" system (before the adaptation), is now accepted by the new system.* In such a case we would like to reject the adaptation/update because the resulting system violates the bank policy by exposing it to a higher risk of loosing money – in the remainder of this work, we will come back many times to this example for illustrating our proposed methods. Since in practice we do not always have a detailed policy available[3], we need a mechanism that makes the impact of model adaptations/updates transparent so that it can be "approved" by a human.[4].

Although there exists work that is aware of the challenge of explaining changing ML systems [18], how to explain model adaptations is still an open research question which we aim to address in this contribution. In this work, we propose a framework that uses contrasting explanations for explaining model adaptations – and more generally differences between two given models –, in particular to match settings where the observed behavior on the given data only slightly changed, but the underlying mechanisms/reasoning might have changed significantly. We argue that inspecting the differences of contrasting explanations which are caused by the adaptation is a reasonable proxy for explaining model adaptations. More precisely, our contributions are as follows:

1. We propose to compare (contrasting) explanations for locally explaining model adaptations and differences.

[1] E.g. The authors of [25] discuss the problem that explanations of a changing classifier can become invalid (i.e. expire) after some time and thus pose a major problem in algorithmic recourse.

[2] Some people even argue that explainability and transparency are an essential part of HCAI [26, 27]

[3] Otherwise there would be no or very limited need for using some ML or AI system that learns such a policy from data.

[4] Ideally the explanation of the adaptations/changes are simple enough to be understood by lay persons instead of only being accessible to ML or AI experts.

2. We propose a method for finding relevant regions in data space which are affected by a given model adaptation/difference and should be explained to a human for approval.
3. We propose persistent local explanations for regularizing the model adaptation towards models with a specific behavior – i.e. avoiding unwanted changes in the internal reasoning of the adapted model.

The remainder of this work is structured as follows: After briefly reviewing related work (Section 2) and the foundations, in particular contrasting explanations (Section 3.2) and model adaptations (Section 3.1), we introduce our proposal for comparing (contrasting) explanations for locally explaining model adaptations in Section 4. We then study counterfactual explanations as a specific instance of contrasting explanations in Section 5 – their changes (Section 5.1), and propose a method for finding relevant regions in data space (Section 5.2). In Section 6, we introduce our idea of persistent contrasting explanations – we consider different types of persistent explanations constraints and study how to add them to the model adaptation optimization problem. Finally, we empirically evaluate our proposed methods on several data sets and scenarios in Section 7, and close our work with a summary and discussion in Section 8.

For the purpose of better readability, we put all proofs and derivations in Appendix A.

2 Related work

While incremental model adaptation as well as transparency (i.e. methods for explaining decision making systems) have been extensively studied separately, the combination of both has received much less attention so far. Counterfactual explanations are a popular instance of example based explanation methods as well as an instance/realization of contrasting explanations. Yet existing methods rely on the assumption of a static underlying model – a strategy for counterfactual explanations of model adaptations is still missing [18].

The authors of [29] propose a method called "counterfactual metrics" which they use for explaining drifting feature relevances of metric based models like learning vector quantization models. In contrast to a counterfactual explanation, their method focuses on feature relevance rather than changes of counterfactual examples. However, this method is limited to metric based models and feature relevance drifts – i.e. it is not applicable to other models and more importantly not to other types of drift or changes.

In [30], another instance of contrasting explanations called contrastive explanations, and in particular counterfactual explanations, are used for explaining concept drift in data space. Unlike our approach, its focus lies on an explanation of the different temporal characteristics of concept drift in the data. For this purpose, a classifier is constructed which tries to separate two batches of data that are assumed to be affected by concept drift – the concept drift is explained by using contrastive explanations that contrast a sample

from one class (i.e. batch) to the other class/batch under the trained classifier. The authors also propose a method for finding interesting and relevant samples which they call "characteristic samples" that are affected by the concept drift and thus promising candidates for explaining the present drift.

Our contributions

In contrast to the related work discussed above, the two key/novel contributions this paper makes are as follows:

1. A general framework for explaining model adaptations or differences by comparing explanations of the old model with explanations of the new model. Although this is a very general approach, we realize it with contrasting explanations – we chose contrasting explanations because these are widely accepted as intuitive and human-friendly explanations.
2. A method for identifying regions in data space where the two models behave the same (e.g. their predictions are the same) but do so for different reasons – i.e. their internal reasoning differs significantly.

3 Foundations

3.1 Model Adaptions

We assume a model adaptation scenario in which we are given a prediction function, also called model, $h(\cdot)$ and a set of (new) labeled data points $\mathcal{D}$. Adapting the model $h(\cdot)$ to the data $\mathcal{D}$ means that we want to find a model $h'(\cdot)$ which is as similar as possible to the original model $h(\cdot)$ while performing well on labeling the (new) samples in $\mathcal{D}$ [24]. Model adaptation can be formalized as an optimization problem [24] as stated in Definition 1.

Definition 1 (Model Adaptation) Let $h : \mathcal{X} \to \mathcal{Y}$, $h \in \mathcal{H}$ be a prediction function (also called model) and $\mathcal{D} = \{(\vec{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}$ a set of (new) labeled data points. Adapting the model $h(\cdot)$ to the data $\mathcal{D}$ is formalized as the following optimization problem:

$$\underset{h' \in \mathcal{H}}{\arg\min}\ \theta(h, h') \quad \text{s.t. } h'(\vec{x}_i) \approx y_i \quad \forall\, (\vec{x}_i, y_i) \in \mathcal{D} \tag{1}$$

where $\theta : \mathcal{H} \times \mathcal{H} \to \mathbb{R}_+$ denotes a regularization that measures the similarity between two given models [5] and $\approx$ refers to a suitable prediction error (e.g. zero-one loss or squared error) which is minimized by $h'(\cdot)$.

Note that for large $\mathcal{D}$, e.g. caused by abrupt concept drift [31], one could completely retrain $h(\cdot)$ and abandon the requirement of closeness. In such situations it is still of interest to explain the difference of $h'(\cdot)$ and $h(\cdot)$.

[5] In case of a parameterized model, one possible regularization measures the difference in the parameters (e.g. by a p-norm).

3.2 Contrasting Explanations

Counterfactual explanations are a popular instance of contrasting explanations. A counterfactual explanation – often just called counterfactual – states a change to some features of a given input such that the resulting data point, called counterfactual, causes a different (specified) behavior (e.g. prediction) than the original input. The rational is considered to be intuitive, human-friendly and useful because it tells the user which minimum changes can lead to a desired outcome [11, 17]. Formally, a (closest) counterfactual can be defined as follows:

Definition 2 ((Closest) Counterfactual Explanation [17]) Assume a prediction function $h : \mathbb{R}^d \to \mathcal{Y}$ is given. Computing a counterfactual $\vec{x}_{\text{cf}} \in \mathbb{R}^d$ for a given input $\vec{x}_{\text{orig}} \in \mathbb{R}^d$ is phrased as an optimization problem:

$$\underset{\vec{x}_{\text{cf}} \in \mathbb{R}^d}{\arg\min} \; \ell\left(h(\vec{x}_{\text{cf}}), y'\right) + C \cdot \theta(\vec{x}_{\text{cf}}, \vec{x}_{\text{orig}}) \tag{2}$$

where $\ell(\cdot)$ denotes a loss function that penalizes deviation of the prediction $h(\vec{x}_{\text{cf}})$ from the requested target prediction y' – for a classification problem, a reasonable choice of loss could be the cross-entropy loss, while in case of regression the squared-error $\ell(h(\vec{x}_{\text{cf}}), y') = (h(\vec{x}_{\text{cf}}) - y')^2$ might be a reasonable –, $\theta(\cdot)$ denotes a penalty for dissimilarity of $\vec{x}_{\text{cf}}$ and $\vec{x}_{\text{orig}}$[6], and $C > 0$ denotes the regularization strength.

Remark 1 In the following, we assume a binary classification problem: In this case, we denote a (closest) counterfactual $\vec{x}_{\text{cf}}$ according to Definition 2 of a given sample $\vec{x}_{\text{orig}}$ under a prediction function $h(\cdot)$ simply as $\vec{x}_{\text{cf}} = \text{CF}(\vec{x}_{\text{orig}}, h)$ and drop the target label y' because it is uniquely determined.

Another instance of contrasting explanations are contrastive explanations. The authors of [32] define a contrastive explanations consisting of two parts: a pertinent negative (equivalent to a counterfactual) and a pertinent positive. A pertinent positive [32–34] describes a minimal set of features that are already sufficient for the given prediction. Pertinent positives are usually considered as an addition to counterfactual explanations and are assumed to provide additional insights why the model took a particular decision [32].

A pertinent positive of a sample $\vec{x}_{\text{orig}}$ describes a minimal set of features $\mathcal{I}$ (of this particular sample $\vec{x}_{\text{orig}}$) that are sufficient for getting the same prediction – these features are also called "turned on" features and all other features are called "turned off", meaning that they are set to zero or any other specified default value. One could either require that all "turned on" features are equal to their original values in $\vec{x}_{\text{orig}}$ or one could relax this and only require that they are close to their original values in $\vec{x}_{\text{orig}}$. The computation of a pertinent positive (also called sparsest pertinent positive) can be phrased

[6] This acts as a regularization of the complexity of the final explanation – i.e. prefer "simple/low complexity" explanations.

as the following multi-objective optimization problem [34]:

$$\min_{\vec{\delta} \in \mathbb{R}^d} \left\| [\vec{x}_{\text{orig}} - \vec{x}_{\text{cf}}]_{\mathcal{I}} \right\| \quad \text{where} \qquad \min_{\mathcal{I}} \ \|\mathcal{I}\| \tag{3a}$$

$$\text{s.t.} \quad h(\vec{x}_{\text{cf}}) = y_{\text{orig}} \tag{3b}$$

where $[\cdot]_{\mathcal{I}}$ denotes the selection operator on the set $\mathcal{I}$ and $\mathcal{I}$ denotes the set of all "turned on" features.[7] $\mathcal{I}$ itself is defined as follows:

$$\mathcal{I} = \left\{ i : \left\| (\vec{x}_{\text{cf}})_i \right\| > \epsilon \right\} \tag{4}$$

where $\epsilon \in \mathbb{R}_+$ denotes a tolerance threshold at which we consider a feature "to be turned on" – e.g. a strict choice would be $\epsilon = 0$.

Since (3) is difficult to solve – a number of different methods for efficiently computing (approximate[8]) pertinent positives have been proposed [32–34].

4 Explaining Model Adaptations

An obvious approach for explaining a model adaptation would be to explain and communicate the regions in data space where the prediction of the new and the old model are different – i.e. $\{\vec{x} \in \mathcal{X} \mid h(\vec{x}) \neq h'(\vec{x})\}$. However, in particular for incremental model adaptations, this set might be small and its characterization not meaningful – because of the constraint in Eq. (1), it is likely that both models compute the same prediction on all samples in a given test set. However, the reason for these predictions can be arbitrarily different – i.e. the internal prediction rules (reasoning) of both models are different. *For example, while both models reject or accept a loan application, the reasons for doing so could be completely different.*. We think that explaining, and thus understanding, how and where the reasons and prediction rules differ are much more useful than just inspecting points where the two models disagree – in particular when it comes to understand and judging decision making systems in the context of human centered AI. Hence, instead of finding the samples where the two models compute different predictions, [9] we aim for an explanation of possible differences in the learned generalization rules underlying the models. Note that this objective is still meaningful, even if the models completely coincide on the given data set. *In the example of loan application, the bank might want to make sure that generalization rules have not changed arbitrarily after updating the model.*

In the following, we propose that a contrasting explanation can serve as a proxy for the model generalization at a given point; hence, a comparison of the possibly different (contrasting) explanations of two models at a given point can

[7] The selection operator returns a vector, whereby it only selects a subset of indices from the original vector as specified in the set $\mathcal{I}$.

[8] The approximation comes from giving up closeness – many methods successfully compute pertinent positives but can not guarantee that they are globally optimal.

[9] Which of course is an obvious and reasonable starting point for explaining the difference between two models.

be considered as an explanation of the differences in the underlying principles based on which the models propose a decision. As it is common practice, we thereby look at local differences, since a global comparison might be too complex and not easily understood by a human, and because it is widely accepted that local and example-based (in particular contrasting) explanations are "easily" accessible to (lay-)persons [11, 18, 20]. *In the context of loan applications, this means to study the entire decision making system by looking at individual cases/applications.* Furthermore, it might also be easier to add constraints on specific samples, instead of constraints on the global decision boundary, to the model adaptation problem Eq. (1) in Definition 1. The computation of such differences of explanations is tractable as long as the computation of the (contrasting) explanation under a single model itself is tractable – i.e. no additional computational complexity is introduced when using this approach for explaining model adaptations and differences.

4.1 Modeling

We define this type of explanation as follows:

Definition 3 (Explaining Model Differences) We assume that we are given a set of labeled data points $\mathcal{D}^* = \{(\vec{x}_i^*, y_i^*) \in \mathcal{X} \times \mathcal{Y}\}$ whose labels are correctly predicted by both models $h : \mathcal{X} \to \mathcal{Y}$ and $h' : \mathcal{X} \to \mathcal{Y}$.

For every $(\vec{x}_i^*, y_i^*) \in \mathcal{D}^*$, let $\vec{\delta}_h^i \in \mathcal{X}$ be a contrasting explanation under $h(\cdot)$ and $\vec{\delta}_{h'}^i \in \mathcal{X}$ under the new model $h'(\cdot)$. The explanation of the model differences at point $(\vec{x}_i^*, y_i^*)$ and its magnitude is then given by the comparison of both explanations:

$$\psi(\vec{\delta}_h^i, \vec{\delta}_{h'}^i) \text{ and } \Psi(\vec{\delta}_h^i, \vec{\delta}_{h'}^i) \tag{5}$$

where $\psi(\cdot)$ denotes a suitable operator which can compare the information contained in two given explanations, and $\Psi(\cdot)$ denotes a real-valued distance measure judging the difference of explanations.

Remark 2 Note that the explanation, as defined in Definition 3, can be more generally applied to compare two arbitrary classifiers $h'(\cdot)$ and $h(\cdot)$ w.r.t. given input locations, albeit $h'(\cdot)$ does not constitute an adaptation of $h(\cdot)$ – e.g. one could compare the internal reasoning of a Support Vector Machine with the one of a Deep Neural Network. For simplicity, we assume uniqueness of contrasting explanations in the definition, either by design such as given for linear models or by suitable algorithmic tie breaks.

In the context of loan application, this means that we explain or define the difference in the internal reasoning of two given models by comparing the contrasting explanations of a specific case/loan application – i.e. comparing the recommended actions for changing the models behavior.

Remark 3 Although we use contrasting explanations in Definition 3, the idea of explaining the difference by comparing explanation, can be realized with any kind of explanation as long as comparing explanations of this particular type is meaningful.

The concrete form of the explanation heavily depends on the comparison function $\psi(\cdot)$ and $\Psi(\cdot)$ – this allows us to take specific use-cases and target groups into account. In this work we assume $\mathcal{X} = \mathbb{R}^d$ and $\psi(\cdot)$ is given by the component-wise absolute value $\|(\vec{\delta}_h^i)_l - (\vec{\delta}_{h'}^i)_l\|_l$, and we consider two possible realizations of $\Psi(\cdot)$:

Euclidean similarity

An obvious measurement of the difference of two explanations in $\mathbb{R}^d$ is to compute the Euclidean distance between them:

$$\Psi(\vec{\delta}_h, \vec{\delta}_{h'}) = \|\vec{\delta}_h - \vec{\delta}_{h'}\|_2 \tag{6}$$

where on could also use a different p-norm (e.g. $p = 1$).

Cosine similarity

We can also measure the difference between two explanations in $\mathbb{R}^d$ by the cosine of their respective angle:

$$\Psi(\vec{\delta}_h, \vec{\delta}_{h'}) = \cos\left(\angle\vec{\delta}_h, \vec{\delta}_{h'}\right) \tag{7}$$

Note that the angle is scale invariant – i.e. it is more interesting if different features have to be changed, rather than the same features have to been changed slightly more. Since the image of the cosine is limited to $[-1, 1]$, we can directly compare the values of different samples – whereas the general norm is only capable of comparing nearby points with each another as it contains information regarding the local topological structure.

5 Counterfactual Explanations for Explaining Model Adaptations

Counterfactual explanations are a popular instance of contrasting explanations (Section 3.2). In the following, we study counterfactual explanations for explaining model adaptations as proposed in Definition 3. We first relate the difference between two linear classifiers to their counterfactuals and, vice versa, the change of counterfactuals to model change (Section 5.1). Then, we propose a method for finding relevant regions and samples for comparing counterfactual explanations (Section 5.2).

5.1 Counterfactuals for Linear Models

First, we highlight the possibility to relate the similarity of two linear models at a given point to their counterfactuals. We consider a linear binary classifier $h : \mathbb{R}^d \to \{-1, 1\}$:

$$h(\vec{x}) = \text{sign}(\vec{w}^\top \vec{x}) \tag{8}$$

and assume w.l.o.g. that the weight vector $\vec{w} \in \mathbb{R}^d$ has unit length, i.e. $\|\vec{w}\|_2 = 1$. We consider an adaptation $h'(\vec{x}) = \text{sign}(\vec{w}_*^\top \vec{x})$ with unit weight vector $\vec{w}_*$. We can then relate the similarity of the two classifier as stated in Theorem 1.

Theorem 1 (Cosine Similarity of Linear Models) *Let $h(\cdot)$ and $h'(\cdot)$ be two linear models Eq. (8), and $\vec{x}_{orig}$ a data point. Let $\vec{x}_{cf} = CF(\vec{x}_{orig}, h)$ and $\vec{x}_{cf*} = CF(\vec{x}_{orig}, h')$ be the closest counterfactual of a data point $\vec{x}_{orig} \in \mathbb{R}^d$ under the original model resp. the adapted model $h'(\cdot)$. Then*

$$\cos(\angle \vec{w}, \vec{w}_*) = \frac{\vec{x}_{cf}^\top \vec{x}_{cf*} + \vec{x}_{orig}^\top \vec{x}_{orig} - \vec{x}_{cf}^\top \vec{x}_{orig} - \vec{x}_{cf*}{}^\top \vec{x}_{orig}}{\sqrt{\left(\vec{x}_{cf}^\top \vec{x}_{cf} + \vec{x}_{orig}^\top \vec{x}_{orig} - 2\vec{x}_{cf}^\top \vec{x}_{orig}\right)\left(\vec{x}_{cf*}{}^\top \vec{x}_{cf*} + \vec{x}_{orig}^\top \vec{x}_{orig} - 2\vec{x}_{cf*}{}^\top \vec{x}_{orig}\right)}} \tag{9}$$

Note that the angle between two separating hyperplanes can be considered as a measurement for their differences. Since every (possibly nonlinear) model can be locally be approximated by a linear function, Theorem 1 also indicates the relevance of counterfactuals to characterize local differences of two models.

Conversely, it is possible to limit the difference of counterfactual explanations by the difference of classifiers as stated in Theorem 2.

Theorem 2 (Change of a Closest Counterfactual Explanation) *Let $h(\cdot)$ be a binary linear classifier Eq. (8) and $h'(\cdot)$ be its adaptation. Then, the difference between the two closest counterfactuals of an arbitrary sample $(\vec{x}_{orig}, y_{orig}) \in \mathbb{R}^d \times \{-1, 1\}$ can be bounded as:*

$$\|CF(\vec{x}_{orig}, h) - CF(\vec{x}_{orig}, h')\|_2 \leq \sqrt{8}\|\vec{x}_{orig}\|_2 \left(1 - \cos(\angle \vec{w}, \vec{w}_*)\right)^{1/2} \tag{10}$$

5.2 Finding Relevant Regions in Data Space

Since we are interested in local explanations, we have to choose locations/-points $\vec{x}$ that are most informative for highlighting differences in the reasoning of the two models. *In our running example of loan application, this translates to finding cases (i.e. loan applications) where the two given models agree in their predictions (i.e. either they both reject or accept the application) but the reasons for doing so are significantly different. These cases might be of interest to the bank because by inspecting these cases the bank could check whether the generalization rules are still in their interest or not.*

In the following, we formalize the notion of characteristic samples in the context of model change, to enable an algorithmic (i.e. automatic) selection of such interesting points.

The idea is to provide an interest function, i.e. a function that marks the regions of the data space that are of interest for our consideration – we could use such a function for automatically finding interesting samples by applying it to a set of points to get a ranking, or optimizing over the function for coming up with interesting samples. This function $i(\cdot)$ should have the following properties:

1. For every pair of fixed models $h, h' \in \mathcal{H}$ it maps every point $x \in \mathcal{X}$ in the data space to a non-negative number indicating the interest – i.e. $i : \mathcal{X} \times (\mathcal{H} \times \mathcal{H}) \to \mathbb{R}_+$.
2. It should be continuous with respect to the classifiers and in particular $i(x, h, h') = 0$ for all x if and only if $h = \pm h'$.
3. Points that are "more interesting" as measured by a difference of local explanations should take on higher values.
4. Regions where the classifiers "coincide structurally", i.e. their local explanations coincide, are not of interest.

The last two properties are basically a localized version of the second in the sense that it forces $i(\cdot)$ to turn global properties of the decision boundary into local, i.e. point wise, properties.

An obvious choice for $i(\cdot)$ is to directly use the explanation Definition 3 itself together with a difference measurement $\Psi(\cdot)$ as stated in Eq. (6) and Eq. (7):

$$i(\vec{x}, h, h') = \Psi\big(\mathrm{CF}(\vec{x}, h), \mathrm{CF}(\vec{x}, h')\big) \tag{11}$$

It is easy to see that the four properties are fulfilled, if we assume that $\Psi(\cdot)$ is chosen suitably.

The first property follows from the definition of dissimilarity functions and so does the second. The fourth property follows from the fact that if the classifiers intrinsically perform the same computations then the counterfactuals are the same and hence $i(\cdot) = 0$; on the other hand, if the classifiers (intrinsically) perform different computations (thought the overall output may be the same) then the counterfactuals are different and hence the $i(\cdot) \neq 0$. In a comparable way the third property is reflected by the idea that obtaining counterfactuals is faithful to the computations in the sense that slight resp. very different computation will lead to slight resp. very different counterfactuals.

Besides the Euclidean distance Eq. (6), the cosine similarity Eq. (7) is a potential choice for comparing two counterfactuals in $\Psi(\cdot)$. Since the cosine always takes values between -1 and 1, we scale it to a positive codomain:

$$\Psi\big(\mathrm{CF}(\vec{x}, h), \mathrm{CF}(\vec{x}, h')\big) = 2 - \cos\big(\angle \mathrm{CF}(\vec{x}, h), \mathrm{CF}(\vec{x}, h')\big) \tag{12}$$

However, the measure $\Psi(\cdot)$ as suggested in Eq. (12) is discontinuous if we approach the decision boundary of one of the classifiers. This problem can be resolved by using an relaxed version of Eq. (12):

$$\Psi(\cdot) = 2 - \frac{\langle \mathrm{CF}(\vec{x}, h) - \vec{x}, \mathrm{CF}(\vec{x}, h') - \vec{x} \rangle}{\|\mathrm{CF}(\vec{x}, h) - \vec{x}\|_2 \|\mathrm{CF}(\vec{x}, h') - \vec{x}\|_2 + \varepsilon} \tag{13}$$

for some small $\varepsilon > 0$ – in this case the samples on the decision boundary are marked as not interesting, which fits the finding that the counterfacutals for those samples basically coincide with the samples them self and therefore do not provide any (additional) information.

Approximation for gradient based models

While the definition of the interest function Eq. (11) captures our goal of identifying interesting samples, the computation of (closest) counterfactuals can be computationally challenging for many models [23], hence finding local maxima of $i(\cdot)$ is infeasible. It is therefore of importance to find a surrogate for the counterfactual $\mathrm{CF}(\cdot,\cdot)$ that allows for fast and easy computation.

In these cases, an efficient approximation is possible, provided the classifier $h(\cdot)$ is induced by a differentiable function $f(\cdot)$ in the form $h(\vec{x}) = \mathrm{sign}\,(f(\vec{x}))$. Then, the gradient of $f(\cdot)$ enables the following approximation of the counterfactual $\mathrm{CF}(\vec{x}, h)$:

$$\mathrm{CF}(\vec{x}, h) = \vec{x} - \eta h(\vec{x}) \nabla_{\vec{x}} f(\vec{x}) \tag{14}$$

for a sufficient $\eta > 0$[10]. In this case Eq. (14), the cosine similarity approach Eq. (12) works particularly well because it is invariant with respect to the choice of η – i.e. η can be ignored. Under some smoothness assumptions, this modeling admits simple geometric interpretations since the gradient points towards the closes point on the decision boundary in this case. This way it (locally) reduces the interpretation to linear classifiers for which counterfactuals are well understood [23].

In the remainder of this work, we use the gradient approximation together with the cosine similarity for computing the interest of given samples.

6 Constrained Model Adaptation for Persistent Explanations

In the previous sections, we proposed and studied the idea of comparing contrasting (in particular counterfactual) explanations for explaining model adaptations and differences. In the experiments (see Section 7), we observe that this method is indeed able to detected and explain less obvious and potential problematic changes of the internal reasoning (decision rules) of adapted models. *We will see in Section 7 that our method can find crucial differences in the internal reasoning of a given (adapted) loan application system, although both models agree on either rejecting or accepting the given application.*

As already discussed in the introduction, in this context of explaining model adaptations, Human-Centered AI comes into play when the user rejects the computed model adaptation based on the explanations. For instance, it might happen that the local explanation under the old model $h(\cdot)$ was accepted, but the new local explanation under the new model $h'(\cdot)$ violates some rules or laws – *e.g. while the reasons for rejecting a given loan application under the old model coincide with the banks guidelines, the reasons for rejecting the same application under the new model might not be acceptable for the bank because it might not be in the interest of the bank or it might be discriminating with*

[10] Although a sufficient $\eta > 0$ guarantees a valid counterfactual, this counterfactual is not necessarily a closest counterfactual.

respect to some sensitive attributes which in turns would cause ethical and legal problems. In such a case, we want to constrained the model adaptation (see Definition 1) such that (some) local explanations remain the same or valid under the new model – i.e. making some local explanations persistent – and by this guiding the new model $h'(\cdot)$ towards globally accepted behavior by making use of such local constraints. *In the context of our loan application example, this means that we want to ensure that the reason for rejecting or accepting a given loan application does not change in an unwanted way when updating the model.*

6.1 Persistent Local Explanations

For the purpose of "freezing" a local explanation in the form of a contrasting explanation – i.e. making it persistent –, we propose the following (informal) requirements:

- Distance to the decision boundary must be within a given interval.
- Counterfactual explanation must be still (in-)valid.
- Pertinent positive must be still (in-)valid.

Aiming for persistent contrasting explanations, we have to augment the original optimization problem Eq. (1) from Definition 1 for adapting a given model $h(\cdot)$ as follows:

$$\arg\min_{h' \in \mathcal{H}} \theta(h, h') \tag{15a}$$

$$\text{s.t. } h'(\vec{x}_i) \approx y_i \quad \forall\, (\vec{x}_i, y_i) \in \mathcal{D} \tag{15b}$$

$$\text{Some local contrastive explanations under } h \text{ are still true under } h' \tag{15c}$$

where the additional (informal) constraint Eq. (15c) is the only difference to the original optimization problem Eq. (1).

Next, we study how we can formalize Eq. (15c) and thus how to solve Eq. (15) for different models and different explanation constraints.

6.2 Modeling

We always assume that we are given a labeled sample $(\vec{x}_{\text{orig}}, y_{\text{orig}}) \in \mathcal{X} \times \mathcal{Y}$ that is correctly labeled by the old model $h(\cdot)$ as well as the new model $h'(\cdot)$ – i.e. $h(\vec{x}_{\text{orig}}) = h'(\vec{x}_{\text{orig}}) = y_{\text{orig}}$. In the subsequent section, we study how to write constraint Eq. (15c) in Eq. (15) for different requirements/constraints as listed in Section 6.1 – it turns out that we can often write the constraints (at least after a reasonable relaxation) as additional labeled samples which enable a straight forward incorporation into many model adaption procedures (see Section 6.3 for details):

$$h'(\vec{x}') = y' \tag{16}$$

6.2.1 Persistent Distance to Decision Boundary

In case of a classifier, one might require that the distance to the decision boundary $\mathrm{d}_h(\vec{x}_{\text{orig}})$ is not larger than some fixed $\lambda \in \mathbb{R}_+$. Applying this to our model adaption setting, we get the following constraint:

$$\mathrm{d}_{h'}(\vec{x}_{\text{orig}}) \leq \lambda \tag{17}$$

However, because reasoning over distances to decision boundaries might be a too complicated and often difficult to formalize as a computational tractable constraint, one might instead require that all samples that are "close" or "similar" to $\vec{x}_{\text{orig}}$ must be have the same prediction y_{orig}:

$$h(\vec{x}) = y_{\text{orig}} \quad \forall\, \vec{x} \in \mathcal{E}(\vec{x}_{\text{orig}}, \mathcal{X}) \tag{18}$$

where we defined the set of all "similar"/"close" points as follows:

$$\mathcal{E}(\vec{x}_{\text{orig}}, \mathcal{X}) = \left\{\vec{x} \in \mathcal{X} \mid \mathrm{d}(\vec{x}, \vec{x}_{\text{orig}}) \leq \lambda\right\} \tag{19}$$

where $\mathrm{d} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ denotes an arbitrary similarity/closeness measure – e.g. in case of real valued features the p-norm might be a popular choice. If $\mathcal{E}(\vec{x}_{\text{orig}}, \mathcal{X})$ contains a "small" number of elements only, then Eq. (18) is computational tractable and can be added as a set of constraints to the optimization problem Eq. (15). However, in case of real valued features where we use the p-norm (e.g. $p = 1$ or $p = 2$) as a closeness/similarity measure, we get the following constraint:

$$h(\vec{x}) = y_{\text{orig}} \quad \forall\, \vec{x} \in \mathcal{X} :\ \|\vec{x} - \vec{x}_{\text{orig}}\|_p \leq \lambda \tag{20}$$

Note that constraints of the form of Eq. (20) are well known and studied in adversarial robustness literature [35] – these constraints reduce the problem to a training an locally adversarial robust model $h'(\cdot)$.

Further relaxing the idea of a persistent distance to the decision boundary might lead to requirements where a set of features is increased or decreased such that the original prediction remains the same. For instance one might have a set of $\vec{z}_j \in \mathbb{R}^d$ which must not change the prediction if added to the original sample $\vec{x}_{\text{orig}}$, yielding the following constraint:

$$h'(\vec{x}_{\text{orig}} + \vec{z}_j) = y_{\text{orig}} \quad \forall\, j \tag{21}$$

6.2.2 Persistent Counterfactual Explanation

Recall that in a counterfactual explanation, we add a perturbation $\vec{z} \in \mathbb{R}^d$ to the data point $\vec{x}_{\text{orig}}$ which results in a (requested) prediction y' different from y_{orig}:

$$h(\vec{x}_{\text{orig}} + \vec{z}) = h(\vec{x}_{\text{cf}}) = y' \neq y_{\text{orig}} \tag{22}$$

where we define $\vec{x}_{\text{cf}} = \vec{x}_{\text{orig}} + \vec{z}$.

Requiring that the same counterfactual explanations holds for the adapted model $h'(\cdot)$, yields the following constraint:

$$h'(\vec{x}_{\text{orig}} + \vec{z}) = h'(\vec{x}_{\text{cf}}) = y' \tag{23}$$

Note that with constraint Eq. (23) alone, we can not guarantee that $\vec{x}_{\text{cf}}$ will be the closest counterfactual of $\vec{x}_{\text{orig}}$ under $h'(\cdot)$ – although it is guaranteed to be a valid counterfactual explanation. However, we think that computing the closest counterfactual is not that important because the closest counterfactual is very often an adversarial which might not be that useful for explanations [11, 36] and for sufficiently complex models, computing the closest counterfactual becomes computational difficult [23]. Furthermore, closeness becomes even less important when dealing with plausible counterfactuals which are usual not the closest ones [36] – if $\vec{x}_{\text{cf}}$ is a plausible counterfactual under $h(\cdot)$ one would expect that it is also plausible under $h'(\cdot)$ because the data manifold itself is not expected to change that much.

6.2.3 Persistent Pertinent Positive

Recall that a pertinent positive $\vec{x}_{\text{pp}} \in \mathbb{R}^d$ describes a sparse sample where all non-zero feature values are as close as possible to the feature values of the original sample $\vec{x}_{\text{orig}}$ and the prediction is still the same:

$$h(\vec{x}_{\text{orig}}) = h(\vec{x}_{\text{pp}}) = y_{\text{orig}} \tag{24}$$

Requiring that $\vec{x}_{\text{pp}}$ is still a pertinent positive of $\vec{x}_{\text{orig}}$ under the adapted model $h'(\cdot)$, yields the following (rather trivial) constraint:

$$h'(\vec{x}_{\text{pp}}) = y_{\text{orig}} \tag{25}$$

Similar to the case of persistent counterfactual explanations, Eq. (25) does not guarantee that $\vec{x}_{\text{pp}}$ is the sparsest or closest pertinent positive of $\vec{x}_{\text{orig}}$ under $h'(\cdot)$ – i.e. it could happen that there exists an even sparser or closer pertinent positive of $\vec{x}_{\text{orig}}$ under $h'(\cdot)$ which was invalid under the old model $h(\cdot)$. However, it is guaranteed that $\vec{x}_{\text{pp}}$ is a sparse pertinent positive of $\vec{x}_{\text{orig}}$ under $h'(\cdot)$ which we consider to be sufficient for practical purposes, in particular if taking into account the computational difficulties of computing a pertinent positive – as stated in [34], computing a pertinent positive (even of "classic" ML models) is not that easy.

6.3 Model specific Implementation

We consider a scenario where we have a sample wise loss function[11] $\ell_{h'} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ that penalizes prediction errors and a set of (new) labeled data

[11] E.g. smth. like the squared error or negative log-likelihood.

points $\mathcal{D} = \{(\vec{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}$ to which we wish to adapt our original model $h(\cdot)$ – we rewrite the model adaptation optimization problem Eq. (1) as follows:

$$\arg\min_{h' \in \mathcal{H}} \theta(h, h') + C \sum_i \ell_{h'}(\vec{x}_i, y_i) \tag{26}$$

where the hyperparameter $C \in \mathbb{R}_+$ allows us to balance between closeness and correct predictions[12].

Next, we assume that we have a bunch of persistence constraints $\mathcal{D}^* = \{(\vec{x}'_j, y'_j) \in \mathcal{X} \times \mathcal{Y}\}$ of the form $h'(\vec{x}'_j) = y'_j$ as discussed in the previous Section 6.2. Considering these constraints, we rewrite the constrained model adaptation optimization problem Eq. (15) as follows:

$$\arg\min_{h' \in \mathcal{H}} \theta(h, h') + C \sum_i \ell_{h'}(\vec{x}_i, y_i) + C' \sum_j \ell_{h'}(\vec{x}'_j, y'_j) \tag{27}$$

where we introduce a hyperparameter $C' \in \mathbb{R}_+$ that denotes a regularization strength which, similar to the hyperparameter C, helps us enforcing satisfaction of the additional persistence constraints – encoded as constraint Eq. (15c) in the original informal modeling Eq. (15).

Assuming a parameterized model, we can use any black-box optimization method (like Downhill-Simplex) or a gradient-based method if Eq. (27) happens to be fully differentiable with respect to the model parameters. However, such methods usually come without any guarantees and are highly sensitive to the solver and the chosen hyperparameters C and C'. Therefore, one would be advised to use and exploit model specific structures for efficiently solving Eq. (27) – e.g. writing Eq. (27) in constrained form and turn it into a convex program.

7 Experiments

We empirically evaluate each of our proposed methods separately. We demonstrate why it is appropriate to compare contrasting explanations for explaining model adaptations and differences in Section 7.2. In Section 7.3, we evaluate our method for finding relevant regions in data space that are affected by the model adaptations, and thus are interesting candidates for illustrating the corresponding difference in counterfactual explanations (see Section 5.2). Finally, we demonstrate the effectiveness of persistent local explanation for avoiding arbitrary, unwanted, changes to the internal reasoning rules.

All experiments are implemented[13] in Python. We use CEML [37] for computing counterfactuals and use MOSEK[14] as a solver for all mathematical programs.

[12] This approach of getting rid of the constraints is also called *penalty method*.
[13] https://github.com/andreArtelt/ContrastiveExplanationsForModelAdaptations
[14] We gratefully acknowledge an academic license provided by MOSEK ApS.

Empirically evaluating our proposed methods is quite challenging because it is not clear how to evaluate "differences in the internal reasoning". Therefore, there do not exist any standard benchmarks or metrics (e.g. numerical scores that could be computed and compared). We construct (artificial) concept drift in data sets – i.e. we know how the concept drift looks like – which then results in adapted models that should differ in their internal reasoning from the original model. By this, we can make sure that there exist differences in the internal reasoning and we also have some clues how these differences should look like – we use this approach as some kind of a ground truth and (manually) check whether our proposed methods are able to detect these differences or not. Especially, we empirically investigate the following questions:

(**R0**): Is our proposed method for explaining model adaptions/differences by comparing counterfactual explanations (see Section 4) able to come up with reasonable explanations that matches the ground truth (if available)?

(**R1**): Is our proposed method for detecting relevant regions in data space (see Section 5.2) able to find a small sets of samples from the training/test set that are affected by the model adaption – i.e. samples where the prediction itself did not change but the reasons for the predictions changed?

(**R2**): Is our proposed regularization for model adaptations (see Section 6) able to prevent unwanted changes in the internal reasoning?

We study each of these questions in a separate subsection, but first, we introduce the data sets used in the experiments.

7.1 Data Sets

We use a variety of artificial (toy) data sets as well as real world data sets for evaluating our proposed methods. In case of the artificial data sets, we introduce drift to the data and subsequently to the adapted model by changing the class conditional distribution of the features. In case of real world data sets, we split the data into two sets such that there are differences in the feature distribution which then should lead to differences in the adapted model.

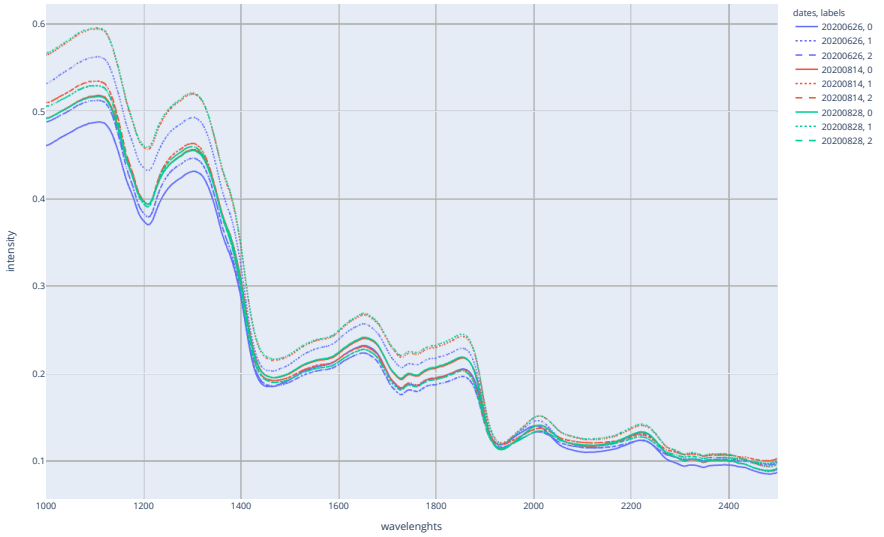
Gaussian Blobs Data Set

This artificial toy data set consists of a binary classification problem. It is generated by sampling from two different two dimensional Gaussian distributions – each class (200 samples) has is its own Gaussian distribution. The drift is introduced by changing the Gaussian distributions between the two sets. In the first set, the two classes can be separated with a threshold on the first feature, whereas in the second set the second feature must be also considered.

Coffee Data Set

The data set consists of hyperspectral measurements of three types of coffee beans measured at three distinct times within three month of 2020. Samples of Arabica, Robusta and immature Arabica beans were measured by a SWIR_384 hyperspectral camera. The sensor measures the reflectance of the samples for 288 wavelengths in the range between 900 and 2500nm. For our experiments,

Fig. 1 Coffee data set: Classwise mean spectra per measurement time.



we standardize and subsample the data by a factor of 5 leading to approx. 120000 samples with 58 dimensions. It is known that the data distribution is drifting between the measurement times – i.e. by splitting the data set into two sets (before and after a specific point in time), we get a data set with abrupt concept drift between the two sets. Classwise means of the data per measurement time are shown in Fig. 1.

Human Activity Recognition Data Set

The human activity recognition (HAR) data set by [38] contains data from 30 volunteers performing activities like walking, walking downstairs and walking upstairs. Volunteers wear a smartphone recording the three-dimensional linear and angular acceleration sensors. We use a time window of length 64 to aggregate the data stream and compute the median per sensor axis and time window. We only consider the activities *walking*, *walking upstairs* and *walking downstairs*. We create drift by putting half of all samples with label *walking* or *walking upstairs* into the first set (approx. 2500 samples) – i.e. walking vs. walking upstairs – the other half of *walking* together with samples labeled as *walking downstairs* into the other set (approx. 2500 samples) – i.e. walking vs. walking downstairs.

German Credit Data Set

The "German Credit Data set" [39] is a data set for loan approval and contains 1000 samples each annotated with 20 attributes (7 numerical and 13 categorical) with a binary target value ("accept" or "reject"). We only use the

first seven features: *duration in month*, *credit amount*, *installment rate in percentage of disposable income*, *present residence since*, *age in years*, *number of existing credits at this bank* and *number of people being liable to provide maintenance for*. We introduce drift by splitting the data set into two sets. We put all samples where *age in years* is less or equal than 35 into the first set and all other samples into the second set.

Boston Housing Data Set

The "Boston Housing Data Set" [40] is a data set for predicting house-prices (regression) and contains 506 samples each annotated with 13 real and positive dimensional features. We introduce drift by putting all samples with a *NOX* value lower or equal than 0.5 into the first set and all other samples into the second set.

7.2 Comparing Counterfactuals for Explaining Model Adaptation

In the following, we investigate **(R0):** Is our proposed explanation (see Section 4) able to come up with reasonable explanations that match the ground truth (if available)? For this purpose, we consider different data sets and scenarios.

7.2.1 Gaussian Blobs

We fit a Gaussian Naive Bayes classifier to the first set and then adapt the model to the second set of the Gaussian blobs data set. Besides the both set, we also generate 200 samples (located between the two Gaussians) for explaining the model changes. We compute counterfactuals for all test samples under the old and the adapted model.

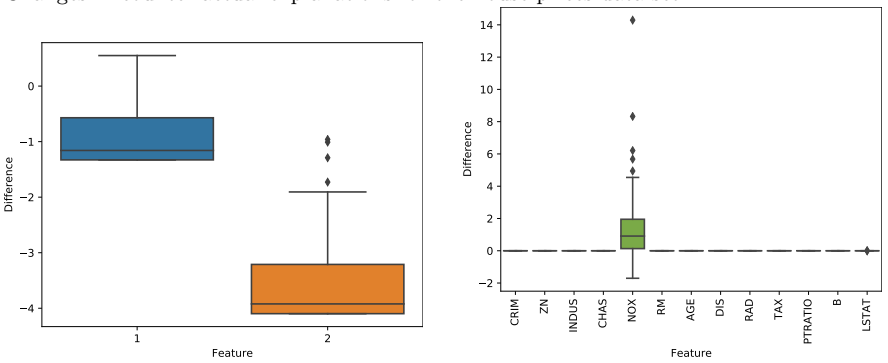
The differences of the counterfactuals are shown in the left plot of Fig. 2. We observe a significant change in the second feature of the adapted model – which makes sense since we know that, in contrast to the first set, the second feature is necessary for discriminating the data in the second set.

7.2.2 Predicting House Prices

We fit a linear regression model to the first set and then completely refit the model on the first and second set of the house prices data set. We use the the test data from both batches for explaining the model changes using counterfactual explanations. We compute counterfactual explanations under the old and the adapted model whereas we always use a target prediction of 20 and allow a deviation of 5.

The differences of the counterfactuals are show in the right plot of Fig. 2. We observe that basically only the feature *NOX* changes – which looks reasonable because we split the data into two sets based on this feature and we would also consider this feature to be relevant for predicting house prices.

Fig. 2 Left: Changes in counterfactual explanations for the Gaussian blob data set. Right: Changes in counterfactual explanations for the house prices data set.



7.2.3 Coffee

We consider the model drift between a model trained with the data collected on the 26th of June and another model based on the data from 14th of August. As we know that the drift in the data set is abrupt, we train a logistic regression classifier on the training data collected at the first measurement time (model_1), and another on the second measurement time (model_2). We compute counterfactual explanations for all the samples in test set of the first measurement time that are classified correctly by model_1 but misclassified by model_2. The target label of the explanation is the original label. This way, we analyze how the model changes for the different measurement times.

The mean difference between the counterfactual explanation and the original sample are visualized in Fig. 3. We observe that surprisingly there are only a few frequencies which are consistently treated different by both models.

Fig. 3 Mean difference in counterfactual explanations for two updated models. Left plot: t_2 = 14th August. Right plot: t_2 = on 28th August.

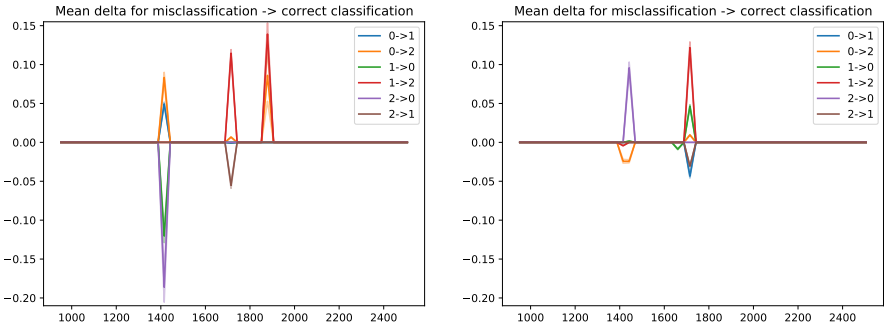
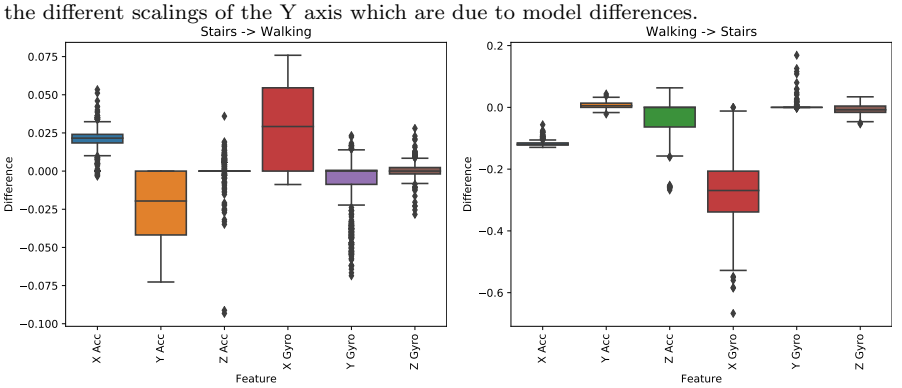


Fig. 4 Changes in counterfactual explanations – each target label is shown separately. Note the different scalings of the Y axis which are due to model differences.



7.2.4 Human Activity Recognition

We fit a Gaussian Naive Bayes classifier to the first set and then adapt the model to the second set of the HAR data set. We use the the test data from both sets for explaining the model changes using counterfactual explanations.

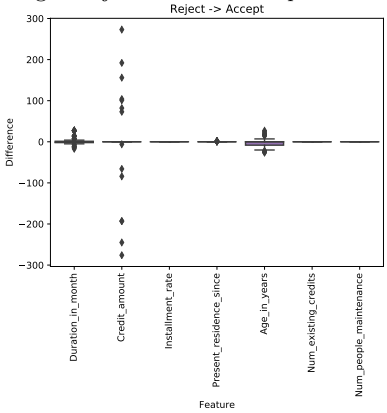
The differences in the counterfactuals (separated by the target label) are show in Fig. 4. In both cases we observe some noise but also a significant change in the Y axis of the acceleration sensor and the X axis of the gyroscope – both changes look plausible[15] but since this is a real world data set, we do not really know the ground truth.

7.2.5 Loan approval

We fit a decision tree classifier to the first set (*age in years* ≤ 35) and completely refit the model to the first and second set (*age in years* > 35) of the credit data set. The test data from both sets is used for computing counterfactual explanations for explaining the model changes.

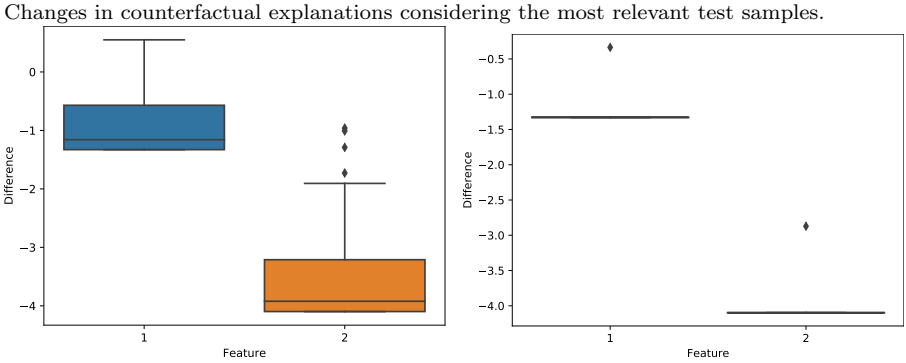
The changes in the counterfactual explanations for switching from "reject" to "accept" are shown in the Fig. 5. We observe that after adapting the model to the second set, there are a couple of cases where increasing the credit amount would turn a rejection into an acceptance

Fig. 5 Changes in counterfactuals when turning a "reject" into an "accept".



[15] Because walking can be considered as horizontal movement (i.e. along the X axis), while going upstairs or downstairs is a vertical movement (i.e. along the Y axis), it seems reasonable that these two axis are flagged as relevant by the counterfactuals.

Fig. 6 Left: Changes in counterfactual explanations considering all test samples. Right: Changes in counterfactual explanations considering the most relevant test samples.



– we consider this as inappropriate
and unwanted behavior which our method is able to detect.

7.3 Finding Relevant Regions in Data Space

In the following, we investigate **(R1):** Is our proposed method for finding relevant regions/samples in data space (see Section 5.2) able to identify samples that are affected by a change in the internal reasoning (caused by a model adaptation)?

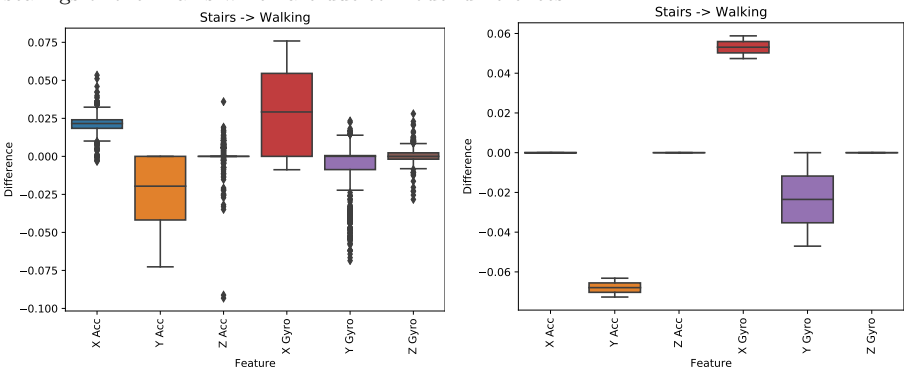
Because our proposed method needs a differentiable model (ideally this model is a classifier), we limit our empirical evaluation to two data sets – the artificial Gaussian Blobs data set where we know the ground truth and the real world Human Activity Recognition data set for illustrative purposes (recall that we do not know the ground truth in this case). For both data sets, we investigate whether our method is able to highlight the relevant (in case the ground truth is known) or reasonable features in the explanation by only using a small set of relevant samples, according to our proposed method from Section 5.2.

7.3.1 Gaussian Blobs

We follow the same procedure like in Section 7.2, but this time we do not use all test samples but only the 10 most relevant (approximately 5% of the test samples) as determined by our method proposed in Section 5.2.

In Fig. 6 we plot the changes in the counterfactual explanations for both cases – i.e. all samples from the test set vs. only the 10 most relevant samples from the test set. We observe the same effects in both cases but with less noise in case of using only a few relevant samples – this suggests that our method from Section 5.2 successfully identifies relevant samples for highlighting and explaining the specific model changes.

Fig. 7 Left: Changes in counterfactuals considering all test samples. Right: Changes in counterfactual explanations considering the most relevant test samples. Note the different scalings of the Y axis which are due to model differences.



7.3.2 Human Activity Recognition

We fit a Gaussian Naive Bayes classifier to the first set and then adapt the model to the second set of the HAR data set. In the first setting, we use the the test data from both sets for explaining the model changes using counterfactual explanations, and in the second setting we only use approx. $\frac{1}{3}$ of all samples sorted by relevance as determined by our method proposed in Section 5.2.

In Fig. 7 we plot the changes of the counterfactual explanations for both cases when switching from walking up-/downstairs to walking straight. We observe the same effects in both cases but with much less noise in case of using only a few relevant samples – this suggests that our method successfully identifies relevant samples for explaining the model changes. Considering only the most relevant samples yields the same (but much stronger) results while saving a lot of computation time – this becomes even more handy when every sample has to be inspected manually (e.g. in some kind of manual quality assurance).

7.4 Persistent Local Explanations

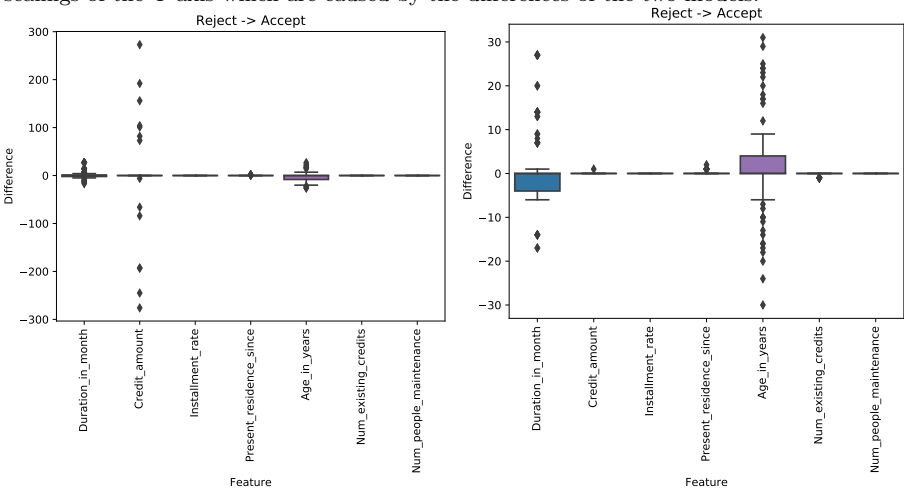
In the following, we investigate **(R2):** Is our proposed regularization *persistent local explanation* (see Section 6) able to prevent unwanted changes in the internal reasoning.

Since the credit data set is the most illustrative data set in our experiments, we demonstrate the usefulness of our proposed persistent local explanations on this data set only.

We fit a decision tree classifier to the first set and completely refit the model to the first and second set of the credit data set. The test data from both sets is used for computing counterfactual explanations for explaining the model changes.

The changes in the counterfactual explanations for switching from “reject” to “accept” are shown in the left plot of Fig. 8. Again, like in Section 7.2.5,

Fig. 8 Left: Changes in counterfactual explanations. Right: Changes in counterfactual explanations under persistent counterfactual explanations. Note the large differences in the scalings of the Y axis which are caused by the differences of the two models.



we observe that after adapting the model to the second set (recall that we split data based on age) there are a couple of cases where increasing the credit amount would turn a rejection into an acceptance which we consider to be inappropriate and unwanted behavior.

We therefore use our proposed method for persistent local explanations from Section 6 to avoid this observed behavior. The results of the constrained model adaptation is shown in the right plot of Fig. 8. We observe that now there is nearly no case in which increasing the credit amount turns a rejection into an acceptance – this suggests that our proposed method for persistent local explanations successfully prevented these unwanted changes.

8 Discussion and Conclusion

In this work, we proposed to compare (contrasting) explanation as a proxy for explaining and understanding model adaptations and differences – i.e. highlighting differences in the underlying reasoning and decision making rules of the models. In this context, we also proposed a method for finding samples where the explanation changed significantly and thus might be illustrative for understanding the model adaptation and differences. Finally, we proposed persistent contrasting explanations for preventing unwanted changes in the internal reasoning of the adapted model. Because we are in a setting where a straight forward evaluation is not established or known – i.e. it is not clear how to "properly" evaluate explanations of model differences – we designed data sets with known ground truth changes/differences and evaluate whether our computed explanations find these intended ground truth, which they did.

8.1 Limitations and Future Work

Although our proposed methods show promising results in the empirically evaluation, we want to highlight some limitations which could be addressed in future research:

- The idea of comparing explanations for understanding and explaining model adaptations or differences is very abstract and the final method heavily depends on the chosen type of explanations that is used for implementing this idea. We used contrasting explanations in this work but as already mentioned, other explanations can be used as well – in this context, it would be interesting to compare other explanations with contrasting explanations and study what kind of changes can be spotted and explained by which type of explanation. The same also applies to the proposed persistent explanations for regularizing model adaptations.
- Our proposed methods for finding regions in data space where the internal reasoning of the models is different, is realized using counterfactual explanations. However, it is unclear if there are changes or differences that cannot be spotted by counterfactual explanations. Furthermore, other types of explanations could be used for realizing a method for finding these regions of interest – in this context, it would be interesting to compare this to our counterfactual approach and study if the same, different or more regions where their reasoning is different can be spotted.
- In this work, we are concerned with explaining and highlighting differences in the internal reasoning of two models – however, it is not clear what exactly is meant by "different reasoning". We "somewhat avoided" this problem by not proposing a precise definition of "different reasoning" but instead demonstrating that our proposed method is able to highlight (manually constructed) differences in the feature processing which then lead to different explanations.
- Our empirically evaluation focuses on detecting differences in the models internal reasoning only. While this covers one important aspect, it completely ignores the human and their needs. How useful are these kinds of explanations to humans, are these explanations "understandable"? Another aspect is the target audience: understanding model differences is clearly of interest to model developers (e.g. ML engineers) but it might also be of interest to normal users or other people with only very little to no technical background knowledge. Contrasting explanation, which we use in this work, are widely accepted for being human friendly and intuitive to lay persons – while this type of explanation might be suited for non technical users, other explanation might be necessary for ML experts who want to debug the model and "deeply understand" what is going on and how to fix it at a technical level. Therefore, it would be of interest and of high relevance to study the benefits of comparing (contrasting) explanations for explaining model adaptations from a psychological perspective – i.e. studying how different people (like lay users vs. ML experts)

perceive model adaptations and how useful they find these explanations for understanding and assessing model adaptations.

Acknowledgments. We gratefully acknowledge funding from the German Federal Ministry of Education and Research (BMBF) through the projects *EML4U* (01IS19080 A) and *TiM* (05M20PBA), funding from the federal state government of North Rhine-Westphalia (NRW) for the project *Bias von KI-Modelle bei der Informationsbildung und deren Implikationen in der Wirtschaft*, and the VW-Foundation for the project *IMPACT* funded in the frame of the funding line *AI and its Implications for Future Society.*

We also thank the anonymous reviewers for their valuable feedback which helped a lot improving the quality and understandability of this work.

Appendix A Proofs and Derivations

- *Theorem 1* Given a sample $\vec{x} \in \mathbb{R}^d$ and a correspond closest counterfactual $\vec{x}_{\text{cf}} \in \mathbb{R}^d$ (Definition 2) under a classifier $h : \mathbb{R}^d \to \mathcal{Y}$, we can compute the weight vector $\vec{w} \in \mathbb{R}^d$ of a locally linear approximation of the classifier $h(\cdot)$ between $\vec{x}$ and $\vec{x}_{\text{cf}}$ as follows:

$$\vec{w} = \vec{x}_{\text{cf}} - \vec{x} \tag{A1}$$

Given a sample $\vec{x} \in \mathbb{R}^d$ and a closest counterfactual $\vec{x}_{\text{cf}} \in \mathbb{R}^d$ before the model drift and another one $\vec{x}_{\text{cf}*} \in \mathbb{R}^d$ after the model adaptation, we can compute the corresponding locally linear approximations of the decision boundaries Eq. (A1) and compute the cosine angle between the two weight vectors Eq. (A1) as follows:

$$\begin{aligned}\frac{\vec{w}_1^\top \vec{w}_2}{\|\vec{w}_1\|_2 \|\vec{w}_2\|_2} &= \frac{(\vec{x}_{\text{cf}} - \vec{x})^\top (\vec{x}_{\text{cf}*} - \vec{x})}{\|\vec{x}_{\text{cf}} - \vec{x}\|_2 \|\vec{x}_{\text{cf}*} - \vec{x}\|_2} \\ &= \frac{\vec{x}_{\text{cf}}^\top \vec{x}_{\text{cf}*} + \vec{x}^\top \vec{x} - \vec{x}_{\text{cf}}^\top \vec{x} - {\vec{x}_{\text{cf}*}}^\top \vec{x}}{\sqrt{\left(\vec{x}_{\text{cf}}^\top \vec{x}_{\text{cf}} + \vec{x}^\top \vec{x} - 2\vec{x}_{\text{cf}}^\top \vec{x}\right)\left({\vec{x}_{\text{cf}*}}^\top \vec{x}_{\text{cf}*} + \vec{x}^\top \vec{x} - 2{\vec{x}_{\text{cf}*}}^\top \vec{x}\right)}}\end{aligned} \tag{A2}$$

which concludes the proof. □

- *Theorem 2* The closest counterfactual $\vec{x}_{\text{cf}} = \text{CF}(\vec{x}, h)$ of a sample $\vec{x}$ under a linear binary classifier Eq. (8) can be stated analytically [41]:

$$\vec{x}_{\text{cf}} = \vec{x} - (\vec{w}^\top \vec{x})\vec{w} \tag{A3}$$

Working out $\|\vec{x}_{\text{cf}} - \vec{x}_{\text{cf}*}\|_2^2$, where $\vec{x}_{\text{cf}} = \text{CF}(\vec{x}, h')$, by making use of Eq. (A3) and $\|\vec{w}\|_2 = \|\vec{w}_*\|_2 = 1$ yields:

$$\begin{aligned}\|\vec{x}_{\text{cf}} - \vec{x}_{\text{cf}*}\|_2 &= \|\vec{x} - (\vec{w}^\top \vec{x})\vec{w} - \vec{x} + (\vec{w}_*^\top \vec{x})\vec{w}_*\|_2 \\ &= \|(\vec{w}_*^\top \vec{x})\vec{w}_* - (\vec{w}^\top \vec{x})\vec{w}_* + (\vec{w}^\top \vec{x})\vec{w}_* - (\vec{w}^\top \vec{x})\vec{w}\|_2 \\ &= \|\left((\vec{w}_* - \vec{w})^\top \vec{x}\right)\vec{w}_* + (\vec{w}^\top \vec{x})(\vec{w}_* - \vec{w})\|_2\end{aligned} \tag{A4}$$

Applying the triangle and Cauchy-Schwarz inequality to Eq. (A4) yields:

$$\begin{aligned}\|\vec{x}_{\text{cf}} - \vec{x}_{\text{cf}*}\|_2 &= \|\left((\vec{w}_* - \vec{w})^\top \vec{x}\right)\vec{w}_* + (\vec{w}^\top \vec{x})(\vec{w}_* - \vec{w})\|_2 \\ &\leq \|\left((\vec{w}_* - \vec{w})^\top \vec{x}\right)\vec{w}_*\|_2 + \|(\vec{w}^\top \vec{x})(\vec{w}_* - \vec{w})\|_2 \\ &= |(\vec{w}_* - \vec{w})^\top \vec{x}|_2 \|\vec{w}_*\|_2 + |\vec{w}^\top \vec{x}|_2 \|\vec{w}_* - \vec{w}\|_2\end{aligned} \tag{A5}$$

Applying the Cauchy-Schwarz inequality to Eq. (A5) yields:

$$\begin{aligned}\|\vec{x}_{\text{cf}} - \vec{x}_{\text{cf}*}\|_2 &\leq |(\vec{w}_* - \vec{w})^\top \vec{x}|_2 \|\vec{w}_*\|_2 + |\vec{w}^\top \vec{x}|_2 \|\vec{w}_* - \vec{w}\|_2 \\ &\leq \|\vec{w}_* - \vec{w}\|_2 \|\vec{x}\|_2 \|\vec{w}_*\|_2 + \|\vec{w}_*\|_2 \|\vec{x}\|_2 \|\vec{w}_* - \vec{w}\|_2 \\ &= 2\|\vec{x}\|_2 \|\vec{w}_* - \vec{w}\|_2\end{aligned} \tag{A6}$$

Substituting $\|\vec{w}_* - \vec{w}\|_2 = \sqrt{2 - 2\cos(\angle\vec{w}, \vec{w}_*)}$ in Eq. (A6) yields the stated bound:

$$\begin{aligned}\|\vec{x}_{\text{cf}} - \vec{x}_{\text{cf}*}\|_2 &\leq 2\|\vec{x}\|_2 \|\vec{w}_* - \vec{w}\|_2 \\ &= 2\|\vec{x}\|_2 \sqrt{2 - 2\cos(\angle\vec{w}, \vec{w}_*)} \\ &= 2\sqrt{2}\|\vec{x}\|_2 \big(1 - \cos(\angle\vec{w}, \vec{w}_*)\big)^{1/2} \\ &= \sqrt{8}\|\vec{x}\|_2 \big(1 - \cos(\angle\vec{w}, \vec{w}_*)\big)^{1/2}\end{aligned} \tag{A7}$$

which concludes the proof. □

References

[1] Stalidis, P., Semertzidis, T., Daras, P.: Examining deep learning architectures for crime classification and prediction **abs/1812.00602** (2018) https://arxiv.org/abs/1812.00602

[2] Khandani, A.E., Kim, A.J., Lo, A.: Consumer credit-risk models via machine-learning algorithms. Journal of Banking & Finance **34**(11) (2010)

[3] Waddell, K.: How algorithms can bring down minorities' credit scores. The Atlantic (2016)

[4] Leslie, D.: Understanding artificial intelligence ethics and safety. CoRR **abs/1906.05684** (2019) https://arxiv.org/abs/1906.05684

[5] parliament, E., council: General Data Protection Regulation: Regulation (EU) 2016/679 of the European Parliament. Official Journal of the European Union (2016)

[6] Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: A survey on bias and fairness in machine learning. CoRR **abs/1908.09635** (2019) https://arxiv.org/abs/1908.09635

[7] Caton, S., Haas, C.: Fairness in machine learning: A survey. CoRR **abs/2010.04053** (2020) https://arxiv.org/abs/2010.04053

[8] Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM Comput. Surv. **51**(5) (2018)

[9] Tjoa, E., Guan, C.: A survey on explainable artificial intelligence (XAI): towards medical XAI. CoRR **abs/1907.07374** (2019) https://arxiv.org/abs/1907.07374

[10] Samek, W., Wiegand, T., Müller, K.: Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. CoRR **abs/1708.08296** (2017) https://arxiv.org/abs/1708.08296

[11] Molnar, C.: Interpretable Machine Learning, (2019)

[12] Greenwell, B.M., Boehmke, B.C., McCarthy, A.J.: A simple and effective model-based variable importance measure. CoRR **abs/1805.04755** (2018) https://arxiv.org/abs/1805.04755

[13] Fisher, A., Rudin, C., Dominici, F.: All Models are Wrong but many are Useful: Variable Importance for Black-Box, Proprietary, or Misspecified Prediction Models, using Model Class Reliance. arXiv e-prints, 1801–01489 (2018) https://arxiv.org/abs/1801.01489 [stat.ME]

[14] Zhao, Q., Hastie, T.: Causal interpretations of black-box models. Journal of Business & Economic Statistics (2019)

[15] Ribeiro, M.T., Singh, S., Guestrin, C.: "why should i trust you?": Explaining the predictions of any classifier. KDD '16. ACM, New York, NY, USA (2016)

[16] Aamodt, A., Plaza., E.: Case-based reasoning: Foundational issues, methodological variations, and systemapproaches. AI communications (1994)

[17] Wachter, S., Mittelstadt, B.D., Russell, C.: Counterfactual explanations without opening the black box: Automated decisions and the GDPR. CoRR **abs/1711.00399** (2017) https://arxiv.org/abs/1711.00399

[18] Verma, S., Dickerson, J., Hines, K.: Counterfactual Explanations for Machine Learning: A Review (2020)

[19] Kim, B., Koyejo, O., Khanna, R.: Examples are not enough, learn to criticize! criticism for interpretability. In: Advances in Neural Information Processing Systems 29 (2016)

[20] Byrne, R.M.J.: Counterfactuals in explainable artificial intelligence (xai): Evidence from human reasoning. In: IJCAI-19 (2019)

[21] Pedapati, T., Balakrishnan, A., Shanmugam, K., Dhurandhar, A.: Learning global transparent models consistent with local contrastive explanations. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin,

H. (eds.) Advances in Neural Information Processing Systems, vol. 33. Curran Associates, Inc., ??? (2020)

[22] Botari, T., Hvilshøj, F., Izbicki, R., de Carvalho, A.C.P.L.F.: MeLIME: Meaningful Local Explanation for Machine Learning Models (2020)

[23] Artelt, A., Hammer, B.: On the computation of counterfactual explanations - A survey. CoRR **abs/1911.07749** (2019) https://arxiv.org/abs/1911.07749

[24] Parisi, G.I., Kemker, R., Part, J.L., Kanan, C., Wermter, S.: Continual lifelong learning with neural networks: A review. Neural Networks **113** (2019)

[25] Venkatasubramanian, S., Alfano, M.: The philosophical basis of algorithmic recourse. FAT* '20 (2020)

[26] Sample, I.: Computer says no: why making AIs fair, accountable and transparent is crucial. The Guardian (2017)

[27] Shneiderman, B.: Bridging the gap between ethics and practice: Guidelines for reliable, safe, and trustworthy human-centered ai systems. ACM Trans. Interact. Intell. Syst. **10**(4) (2020)

[28] Wortman Vaughan, J., Wallach, H.: A Human-Centered Agenda for Intelligible Machine Learning (2021)

[29] Artelt, A., Hammer, B.: Efficient computation of counterfactual explanations and counterfactual metrics of prototype-based classifiers. Neurocomputing **470**, 304–317 (2022). https://doi.org/10.1016/j.neucom.2021.04.129

[30] Hinder, F., Hammer, B.: Counterfactual Explanations of Concept Drift (2020)

[31] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. ACM computing surveys (CSUR) **46**(4) (2014)

[32] Dhurandhar, A., Chen, P., Luss, R., Tu, C., Ting, P., Shanmugam, K., Das, P.: Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In: Bengio, S., Wallach, H.M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada (2018)

[33] Dhurandhar, A., Pedapati, T., Balakrishnan, A., Chen, P., Shanmugam, K., Puri, R.: Model agnostic contrastive explanations for structured data. CoRR **abs/1906.00117** (2019) https://arxiv.org/abs/1906.00117

[34] Artelt, A., Hammer, B.: Efficient computation of contrastive explanations. In: International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021, pp. 1–9. IEEE, ??? (2021). https://doi.org/10.1109/IJCNN52387.2021.9534454. https://doi.org/10.1109/IJCNN52387.2021.9534454

[35] Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I., Madry, A., Kurakin, A.: On evaluating adversarial robustness. arXiv preprint arXiv:1902.06705 (2019)

[36] Artelt, A., Hammer, B.: Convex density constraints for computing plausible counterfactual explanations. In: Farkas, I., Masulli, P., Wermter, S. (eds.) Artificial Neural Networks and Machine Learning - ICANN 2020 - 29th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 15-18, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12396, pp. 353–365. Springer, ??? (2020). https://doi.org/10.1007/978-3-030-61609-0_28. https://doi.org/10.1007/978-3-030-61609-0_28

[37] Artelt, A.: CEML: Counterfactuals for Explaining Machine Learning models - A Python toolbox. GitHub (2019-2021)

[38] Reyes-Ortiz, J., Oneto, L., Samà, A., Parra, X., Anguita, D.: Transition-aware human activity recognition using smartphones. Neurocomputing **171** (2016)

[39] Statlog (German Credit Data) Data Set (1994). https://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29

[40] Boston Housing Data Set (1978). https://archive.ics.uci.edu/ml/datasets/Housing

[41] Artelt, A., Vaquet, V., Velioglu, R., Hinder, F., Brinkrolf, J., Schilling, M., Hammer, B.: Evaluating robustness of counterfactual explanations. arXiv preprint arXiv:2103.02354 (2021)