Contrastive Explanations for Explaining Model Adaptations [⋆]

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

CITEC - Cognitive Interaction Technology
Bielefeld University, 33619 Bielefeld, Germany
{aartelt,fhinder,vvaquet,rfeldhans,bhammer}@techfak.uni-bielefeld.de

Abstract. Many decision making systems deployed in the real world are not static - a phenomenon known as model adaptation takes place over time. The need for transparency and interpretability of AI-based decision models is widely accepted and thus have been worked on extensively. 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 adaptations.
In this contribution, we propose and (empirically) evaluate a framework for explaining model adaptations by contrastive explanations. We also propose a method for automatically finding regions in data space that are affected by a given model adaptation and thus should be explained.

Keywords: XAI · Contrastive Explanations · Model Adaptation

1 Introduction

Machine learning (ML) and artificial intelligence (AI) based decision making systems are increasingly affecting our daily life - e.g. predictive policing [35] and loan approval [21,40]. Given the impact of many ML and AI based decision making systems, there is an increasing demand for transparency and interpretability [24] - the importance of these aspects was also emphasized by legal regulations like the EUs GDPR [28]. In the context of transparency and interpretability, fairness and other ethical aspects become relevant [13, 25].

As a consequence, the research community extensively worked on these topics and came up with methods for explaining ML and AI based decision making systems and thus meeting the demands for transparency and interpretability [18, 19,32,36]. Popular explanations methods [26,36] are feature relevance/importance methods [16] and examples based methods [3]. Instances of example based methods are counterfactual explanations [38, 39], influential instances [23] and Prototypes

[⋆] We gratefully acknowledge funding from the German Federal Ministry of Education and Research (BMBF) through the projects *EML4U* (01IS19080 A) and *TiM* (05M20PBA), and the VW-Foundation for the project *IMPACT* funded in the frame of the funding line *AI and its Implications for Future Society.*

& criticisms [22] - these methods use a set or a single example for explaining the behavior of the system. Counterfactual and contrastive explanations in general [26, 38, 39] are popular instances of example based methods for locally explaining decision making systems [26] - the reason why these types of explanation are so popular is that because there exists strong evidence that explanations by humans (which they try to mimic) are often counterfactual in nature [11]. 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 the behavior of the decision making system at a particular instance or in a "small" region in data space [10, 26, 29, 31]. 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 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 [5].

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 (continuously) evolving over time - e.g. the decision making system is adapting or fine tuned on new data [27]. In this context it becomes 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 building block[2] in ethical AI [41]. HCAI allows the human being (the people) to "rule" AI systems instead of being "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 decision making system or the impact of some adaptation or changes applied to a given system. Yet, understanding the impact of changing a system in a given way is crucial for rejecting system changes that violates some (ethical) guidelines or (legal) constraints.

For example if we consider the scenario of a (non-trivial) loan approval system that automatically accepts or rejects loan applications - i.e. we assume that the decision making process of this system is highly complicated and difficult to inspect from the outside (e.g. it might be a Deep Neural Network): *We might encounter a situation in which a loan application was rejected with the argument of a low income and a bad payback rate in the past - which perfectly meets the bank internal guidelines for accepting or rejecting a loan. Next, we adapt the loan approval system on new data - we assume that we got new data for fine tuning the system - but the we assume that the guidelines or policies for accepting or*

[1] E.g. The authors of [37] 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 [33, 34]

rejecting did not change. But after this adaptation, it turns out that the same application that was rejected under the "old" system (before the adaptation), it is now accepted by the new system - that is we assumes that this changed behavior violates the risk-guidelines of the bank because it exposes the bank to an unnecessarily higher risk of loosing money. This is an example in which case we would like to reject the given model adaptation because this adaptation would lead to a system that violates some predefined rules. Since in practice we do not always have a detailed policy available[3], we need a mechanism that makes the impact of model adaptations/changes transparent so that it can be "approved" by a human[4].

Although there exist general overview work that is aware of the challenge of explaining changing ML systems [38], how to exactly do this is still an open research question which we aim to address in this contribution. In this work, we propose a framework that uses contrastive explanations for explaining model adaptation - i.e. we argue that inspecting the changes/differences of contrastive explanations is a reasonable proxy for explaining model adaptations. More precisely, our contributions are:

- We propose to compare contrasting explanations for locally explaining model adaptations.
- We propose a method for finding relevant and interesting regions in data space which are affected by a given model adaptation and thus should be explained to the user.
- We propose persistent local explanations for regularizing the model adaptation towards models with a specific behaviour.

The remainder of this work is structured as follows: After briefly reviewing literature and taking a look at the foundations like contrasting explanations (section 2.2) and model adaptations (section 2.1) we introduce and describe our proposal for using contrastive explanations for locally explaining model adaptations in section 3. In this context, we then study counterfactual explanations as a specific instance of contrastive explanations in section 4 - in particular we study robustness (see section **??**) and propose a method for finding relevant and interesting and relevant regions in data space (see section 4.2). In section 5, 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 in section 6 and close our work with a summary and discussion in section 7.

Due to space constraints and for the purpose of better readability, we include all proofs and derivations in appendix A.

[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 be accessible to ML or AI experts.

Related work While (concept) drift as well as transparency (i.e. methods for explaining decision making systems) have been extensively studied separately, the combination of both have received much less attention so far.

Counterfactual explanations are a popular instance of example based explanation methods but all existing methods so far assume that the underlying model which is explained does not change over time - a strategy for counterfactual explanations of changing/drifting models is still missing [38].

A method called "counterfactual metrics" [8] can be used for explaining drifting feature relevances of metric based models. In contrast to a counterfactual explanation, it focuses on feature relevance rather than change of counterfactual examples. The authors of [8] consider a scenario in which a metric based model is adapted to drifting feature relevances and the resulting model adaptation is explained by the the changes made to the distance metric which they call "counterfactual metrics".

In [20], contrastive explanations (in particular counterfactual explanations) are used for explaining concept drift. 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 illustrating and explaining the present drift.

2 Foundations

2.1 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 $\mathcal{D}$ [27]. Model adaptation can be formalized as an optimization problem [27] 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:*

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

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 $h'(\cdot)$.*

[5] In case of a parameterized model, one possible regularization measures the difference in the parameters.

Note that for large $\mathcal{D}$, e.g. caused by abrupt concept drift [17], one could completely retrain $h(\cdot)$ and abandon the requirement of closeness. In such situations it is still interesting to explain the difference of $h'(\cdot)$ and $h(\cdot)$.

2.2 Contrastive Explanations

Counterfactual explanations are a popular instance of contrastive explanations. A counterfactual explanations - often just called counterfactual or pertinent positive by some authors [14] - states a change to some features of a given input such that the resulting data point (called counterfactual) has a different (specified) 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 [26, 39]. Formally, a (closest) counterfactual can be defined as follows:

Definition 2 ((Closest) Counterfactual Explanation [39]). *Assume a prediction function $h : \mathbb{R}^d \to \mathcal{Y}$ is given. Computing a counterfactual $\vec{x}_{cf} \in \mathbb{R}^d$ for a given input $\vec{x} \in \mathbb{R}^d$ is phrased as an optimization problem:*

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

where $\ell(\cdot)$ denotes the loss function, y' the requested prediction, and $\theta(\cdot)$ a penalty term for deviations of $\vec{x}_{cf}$ from the original input $\vec{x}$. $C > 0$ 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 $\vec{x}$ under a prediction function $h(\cdot)$ as $\vec{x}_{\text{cf}} = \text{CF}(\vec{x}, h)$ as the desired target is uniquely determined.

The authors of [14] define a contrastive explanations consisting of two parts: a pertinent negative (counterfactual) and a pertinent positive. A pertinent positive [7, 14, 15] describes a minimal set of features that are already sufficient for the given prediction. As already mentioned, pertinent positives are usually considered as a part or addition of contrastive explanations and are assumed to provide additional insights why the model took a particular decision [14].

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 as the following multi-objective optimization problem [7]:

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

$$\min_{\mathcal{I}} \; |\mathcal{I}| \tag{3b}$$

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

where $[\cdot]_{\mathcal{I}}$ denotes the selection operator on the set $\mathcal{I}$ and $\mathcal{I}$ denotes the set of all "turned on" features.[6] $\mathcal{I}$ is defined as follows:

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

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[7]) pertinent positives have been proposed [7, 14, 15].

3 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. Hence, instead of finding these samples, where the two models compute different predictions[8], we aim for an explanation of their learned generalization rules. Because of the constraint in Eq. (1), it is likely the case that both models compute the same prediction on all samples in a given test. However, the reason for these predictions can be arbitrarily different (depending on the regularization $\theta(\cdot)$) - i.e. the internal prediction rules of both models are different. We think that explaining and thus understanding how and where the reasons and rules for predictions 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.

In the following, we propose that a contrastive explanation can serve as a proxy for the model generalization at a given point; hence a comparison of the possibly different contrastive explanations of two models at a given point can be considered as an explanation of how the different 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

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

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

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

not easily understood by a human [11, 26, 38]. 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 constrastive explanation under a single model itself is tractable - i.e. no additional computational complexity is introduced when using this approach for explaining model adaptations.

3.1 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 contrastive 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 classifiers $h'(\cdot)$ and $h(\cdot)$ w.r.t. given input locations, albeit $h'(\cdot)$ does not constitute an adaptation of $h(\cdot)$. For simplicity, we assume uniqueness of contrastive explanations in the definition, either by design such as given for linear models or by suitable algorithmic tie breaks.

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 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 by the cosine of their respective angle:

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

Remark 3. Note that considering the angle, instead of the actual distance, has the advantage that it is scale invariant - i.e. it is more interesting if different features have to be changed, rather than the same features have to be 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.

4 Counterfactual Explanations for Explaining Model Adaptations

Counterfactual explanations Definition 2 are a popular instance of contrastive explanations (section 2.2). 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. Finally, we propose a method for finding relevant regions and samples for comparing counterfactual explanations.

4.1 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 $h : \mathbb{R}^d \to \{-1, 1\}$:

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

and assume w.l.o.g. that the weight vector $\vec{w} \in \mathbb{R}^d$ has unit length, $\|\vec{w}\|_2 = 1$. We assume for an adaptation $h'(\vec{x}) = \operatorname{sign}(\vec{w}_*^\top \vec{x})$ with unit weight vector $\vec{w}_*$

In Theorem 1 we state how to use counterfactuals for approximately computing the local cosine similarity between two models - we interpret this as evidence for the usefulness of counterfactual explanations for measuring the difference between two given models.

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

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

Since every (possibly nonlinear) model can locally be approximated linearly, this result 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 follows:

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 $(\vec{x}, y) \in \mathbb{R}^d \times \{1, 1\}$ can be bounded as:*

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

4.2 Finding Relevant Regions in Data Space

The task of sample based model comparison obviously requires the selection of feasible samples, as the amount of samples is usually to large to be dealt with by hand. Thus, we need to formalize a notion of characteristic samples in the context of model change to perform this sub-task automatically. In this section, we aim to formalize this notion and give a number of possible choices and respective approximation regarding this problem.

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 certain 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 - 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" should take on higher values.
4. Regions where the classifiers 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 properties of the decision boundary (which are global) into local, i.e. point wise, properties. It is possible to make the properties 3 and 4 rigorous, but this would require an inadequate amount of theory.

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

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

Then it is easy to see that the four properties are fulfilled, if we assume that $\Psi(\cdot)$ and is chosen correctly:

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, $\Psi(\cdot)$ as defined 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):

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

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.

An approximation for gradient based models While the definition of the interest function Eq. (11) perfectly captures our goal of identifying interesting samples, it can be computational difficult to compute. In particular the computation of (closest) counterfactual explanations can be computational expensive and "challenging" for many models [5] - this becomes a major issue when optimizing, that is searching for local maxima of $i(\cdot)$. It is hence 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 an approximation of the counterfactual in the following form:

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

for a sufficient $\eta > 0$. 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 and we only need the gradient. Another benefit of this choice is that, under some smoothness assumptions regarding the classifier, admits simple geometric interpretations of the obtained values as the gradient always point towards the closes point on the decision boundary. This way it (locally) reduces the interpretation to linear classifiers for which counterfactual explanations are well understood [5].

In the remainder of this work, we use the gradient approximation together with the cosine similarity for computing the "usefulnes" of given samples for comparing their counterfactual explanations.

5 Constrained Model Adaptation for Persistent Explanations

In the previous sections we proposed and studied the idea of comparing contrastive (in particular counterfactual) explanations for explaining model adaptations. In the experiments (see section 6) we observe this method is indeed able to detected and explain less obvious and potential problematic changes of the internal decision rules of adapted models.

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 - see the introduction for an example. In such a case, we want to constrained the model adaptation Definition 1 such that (some) local explanations remain the same or valid under the new model - i.e. making some local explanations persistent - and to push the new model $h'(\cdot)$ towards globally accepted behavior by making use of such local constraints.

5.1 Persistent Local Explanations

For the purpose of “freezing” a local explanation in the form of a contrastive 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 contrastive explanations, we have to augment the original optimization problem Eq. (1) from Definition 1 for adapting a given model $h(\cdot)$ as follows:

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

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

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

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.

5.2 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 the different requirements/constraints as listed in section 5.1 - 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 5.3 for details):

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

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"/"cose" 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 [12] - 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:

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

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 (specified) prediction 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 defined $\vec{x}_{\text{cf}} = \vec{x}_{\text{orig}} + \vec{z}$.

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 [6, 26] and for sufficiently complex models, computing the closest counterfactual becomes computational difficult [5]. Furthermore, closeness becomes even less important when dealing with plausible counterfactuals which are usual not the closest ones [6] - 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.

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 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 $h'(\cdot)$ - 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 [7], computing a pertinent positive (even of "classic" ML models) is not that easy.

5.3 Model specific Implementation

We consider a scenario where we have a sample wise loss function[9] $\ell_{h'} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ that penalizes prediction errors and a set of (new) labeled data 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:

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

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

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

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 5.2. Considering these constraints, we rewrite the constrained model adaptation optimization problem Eq. (15) as follows:

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

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 Eq. (15c) in the original informal modelling 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 exploit model specific structures for efficiently solving Eq. (27) - e.g. write Eq. (27) in constrained form and turn it into a convex program.

6 Experiments

We empirically evaluate each of our proposed methods separately. We demonstrate the usefulness of comparing contrastive explanations for explaining model adaptations in section 6.2, and in section 6.3 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 4.2). Finally, we demonstrate the effectiveness of persistent local explanation for pushing the model adaptation towards a desired behaviour.

The Python implementation of all experiments is available on GitHub[10]. We use the Python toolbox CEML [4] for computing counterfactual explanations and use MOSEK[11] as a solver for all mathematical programs.

6.1 Data Sets

We use the following data sets in our experiments:

Gaussian Blobs Data Set This artificial toy data set consists of a binary classification problem and is generated by sampling from two different two dimensional Gaussian distributions - each class has is its own Gaussian distribution. The drift is introduced by changing the Gaussian distributions between the two batches. In the first batch the two classes can be separated with a threshold on the first feature, whereas in the second batch the second feature must be also considered.

[10] https://github.com/andreArtelt/ContrastiveExplanationsForModelAdaptations

[11] We gratefully acknowledge a academic license provided by MOSEK ApS.

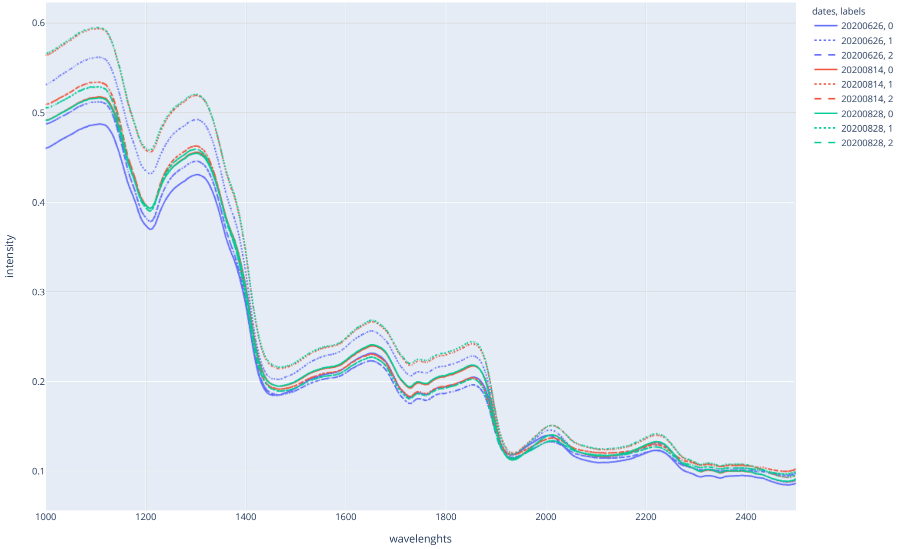
Boston Housing Data Set The “Boston Housing Data Set” [1] is a data set for predicting house-prices (regression) and contains 506 samples each annotated with 13 real and positive dimensional features. We introduce drift by putting all samples with a *NOX* value lower or equal than 0.5 into the first batch and all other samples into the second batch.

Human Activity Recognition Data Set The human activity recognition (HAR) data set by [30] 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 computed 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 batch - i.e. the classifier has to distinguish walking vs. walking upstairs - and all other samples, the other half of *walking* together with samples labeled as *waking downstairs* into the other batch - i.e. for the second batch the classifier has to distinguish normal walking vs. walking downstairs.

German Credit Data Set The “German Credit Data set” [2] 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 use only 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 putting all samples where *age in years* is less or equal than 35 into the first batch and all other samples into the second batch.

Coffee

Fig. 1: Labelwise mean spectra per measurement time.



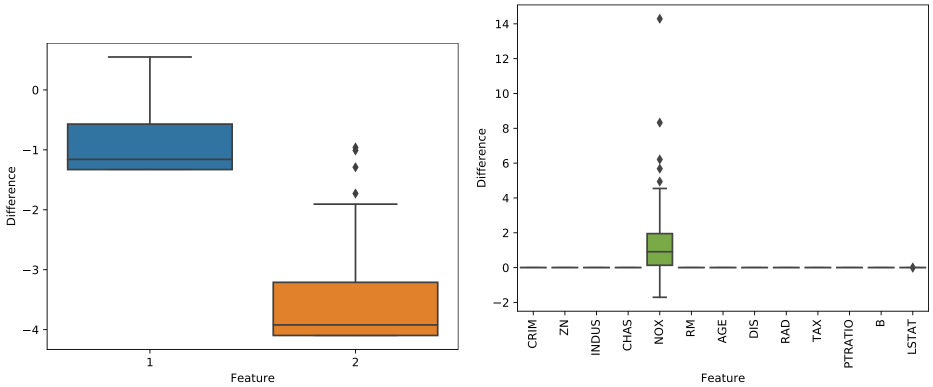
The data set consists of hyperspectral measurements of three types of coffee beans measured at three distinct times within three month of 2020. Samples of Arabica, Robusta and immature Arabica beans were measured by a SWIR_384 hyperspectral camera produced by Norsk Elektro Optikk. The sensor measures the reflectance of the samples for 288 wavelengths lying in the area between 900 and 2500nm. For our experiments, we standardize and subsample the data by a factor of 5. Prior analysis of the data set indicates, that there the data distribution is drifting between the measurement times. Labelwise means of the data per measurement time are shown in Fig. 1.

6.2 Comparing Counterfactual Explanations for Explaining Model Adaptation

Gaussian Blobs We fit a Gaussian Naive Bayes classifier to the first batch and then adapt the model to the second batch of the Gaussian blobs data set. Besides the both batches, we also generate 200 samples (located between the two Gaussians) for explaining the model changes using counterfactual explanations. We compute counterfactual explanations for all test samples under the old and the adapted model. The differences of the counterfactuals are shown in the right 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 batch, the second feature is necessary for discriminating the data in the second batch.

Predicting House Prices We fit a linear regression model to the first batch and then completely refit the model on the first and second batch 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 left plot of Fig. 2. We observe that basically only the feature *NOX* changes - which makes sense because we split the data into two batches 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.

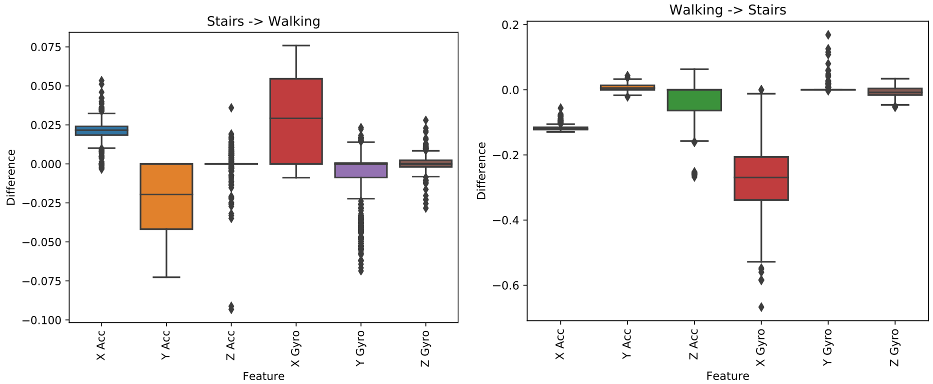


Gaussian blobs

House prices

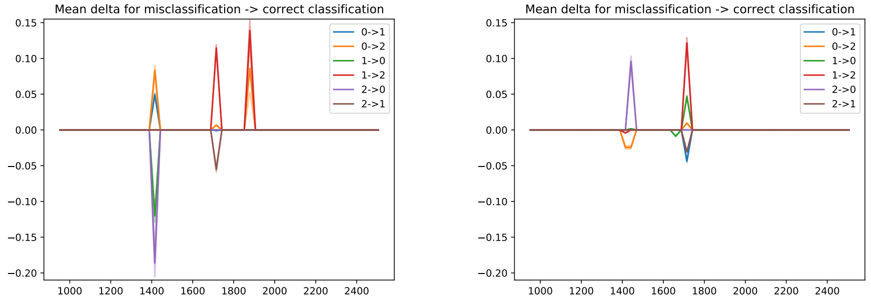
Human Activity Recognition We fit a Gaussian Naive Bayes classifier to the first batch and then adapt the model to the second batch of the human activity recognition data set. We use the the test data from both batches for explaining the model changes using counterfactual explanations. The differences of the counterfactuals (separated by the target label) are show in Fig. 3. 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 looks plausible because switching between walking up-/downstairs should affect the Y axis of the acceleration sensor while walking straight might be measurable by the X axis of the gyroscope, but since this is a real world data set, we do not really know the ground truth.

Fig. 3: Changes in counterfactual explanations - each target label is shown separately.



Coffee We are considering the model drift between a model trained with the data collected on the 26th June and another model based on the data from 14th August (from the 28th August in a second experiment). As the we know that the drift in our 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 or visualized in Fig. 4. We observe that there are (interestingly) only a few frequencies which a are consistently differently treated by both model.

Fig. 4: Counterfactual explanations for two updated models.



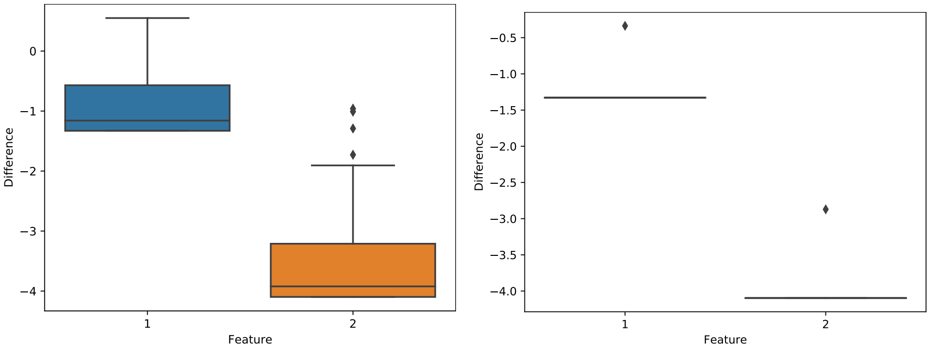
t_2 = 14th August.

t_2 = on 28th August.

6.3 Finding Relevant Regions in Data Space

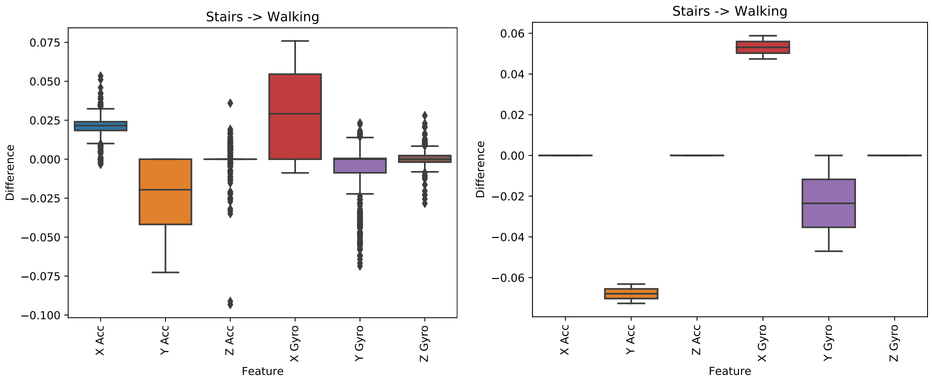
Gaussian Blobs We follow the same procedure like in section 6.2 but this time we do not use all test samples but only the 10 (approximately 5% of the test samples) most relevant as determined by our method proposed in section 4.2. In Fig. 5 we plot the changes of the counterfactual explanations for both cases. 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 4.2 successfully identifies relevant samples for highlighting and explaining the specific model changes.

Fig. 5: Left: Changes in counterfactual explanations considering all test samples. Right: Changes in counterfactual explanations considering the most relevant test samples.



Human Activity Recognition We follow the same procedure like in section 6.2 but this time we do not use all test samples but only the 500 (approximately $\frac{1}{3}$ of the test samples) most relevant as determined by our method proposed in section 4.2. In Fig. 6 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 (we also clearly observe a little change in the Y axis of gyroscope which is not that strong in case of using all samples) - this suggests that our method from section 4.2 successfully identifies relevant samples for highlighting and explaining the specific 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).

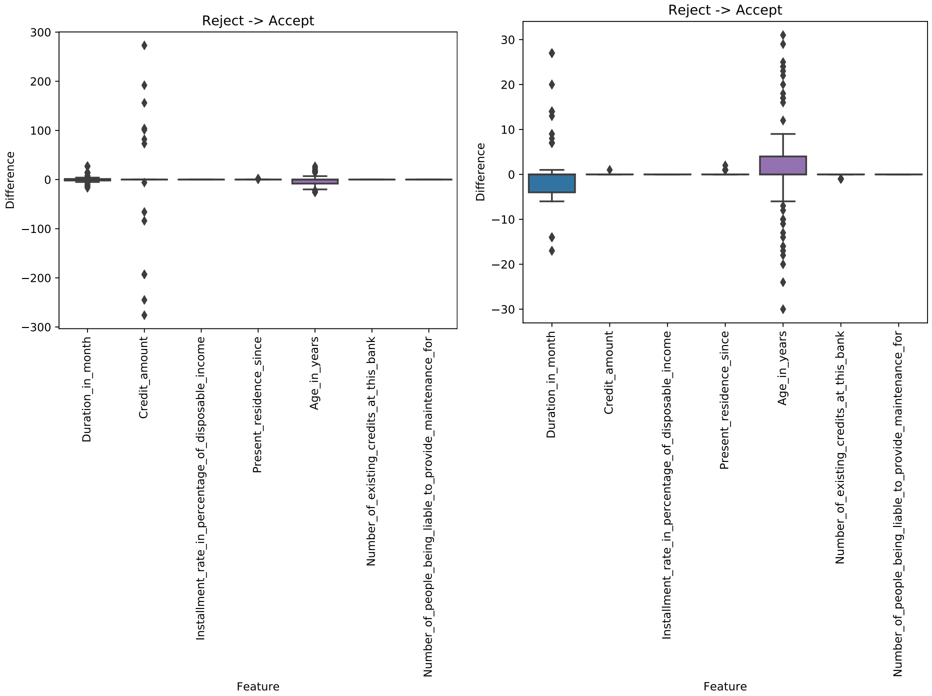
Fig. 6: Left: Changes in counterfactual explanations considering all test samples. Right: Changes in counterfactual explanations considering the most relevant test samples.



6.4 Persistent Local Explanations

Loan approval We fit a decision tree classifier to the first batch and completely refit the model to the first and second batch of the credit data set. The test data from both batches is used for computing counterfactual explanations for explaining the model changes. The changes in the counterfactual explanations for switching from “reject” to “accept” is shown in the left plot of Fig. 7. We observe that after adapting the model to the second batch (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 as inappropriate and unwanted behaviour. We therefore use our proposed method for persistent local explanations from section 5.1 and section 5 to avoid this observed behaviour. The results of the constrained model adaptation is shown in the right plot of Fig. 7. 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 pushed to the model towards our requested behaviour.

Fig. 7: Left: Changes in counterfactual explanations. Right: Changes in counterfactual explanations under persistent counterfactual explanations.



7 Discussion and Conclusion

In this work we proposed to compare contrastive explanation as a proxy for explaining and understanding model adaptations - i.e. highlighting differences in the underlying 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. Finally, we proposed persistent constrative explanations for pushing the model adaptation towards a specific behaviour - i.e. ensuring that the model (after adaptation) satisfies some specified criteria. We empirically demonstrated the functionality of all our proposed methods.

In future research we would like to study the benefits of comparing contrastive explanations for explaining model adaptations from a psychological perspective - i.e. conducting a user study to learn more on how people perceive model adaptations and how useful they find these explanations for understanding and assessing model adaptations.

References

1. Boston housing data set (1978), https://archive.ics.uci.edu/ml/datasets/Housing
2. Statlog (german credit data) data set (1994), https://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29
3. Aamodt, A., Plaza., E.: Case-based reasoning: Foundational issues, methodological variations, and systemapproaches. AI communications (1994)
4. Artelt, A.: Ceml: Counterfactuals for explaining machine learning models - a python toolbox. https://www.github.com/andreArtelt/ceml (2019-2021)
5. Artelt, A., Hammer, B.: On the computation of counterfactual explanations - A survey. CoRR **abs/1911.07749** (2019), http://arxiv.org/abs/1911.07749
6. Artelt, A., Hammer, B.: Convex density constraints for computing plausible counterfactual explanations. 29th International Conference on Artificial Neural Networks (ICANN) (2020)
7. Artelt, A., Hammer, B.: Efficient computation of contrastive explanations (2021)
8. Artelt, A., Hammer, B.: Efficient computation of counterfactual explanations and counterfactual metrics of prototype-based classifiers (2021)
9. Artelt, A., Hammer, B.: Fairness and robustness of contrasting explanations (2021)
10. Botari, T., Hvilshøj, F., Izbicki, R., de Carvalho, A.C.P.L.F.: Melime: Meaningful local explanation for machine learning models (2020)
11. Byrne, R.M.J.: Counterfactuals in explainable artificial intelligence (xai): Evidence from human reasoning. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19. pp. 6276–6282. International Joint Conferences on Artificial Intelligence Organization (7 2019). https://doi.org/10.24963/ijcai.2019/876, https://doi.org/10.24963/ijcai.2019/876
12. 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)
13. Caton, S., Haas, C.: Fairness in machine learning: A survey. CoRR **abs/2010.04053** (2020), https://arxiv.org/abs/2010.04053

14. 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. pp. 590–601 (2018), http://papers.nips.cc/paper/7340-explanations-based-on-the-missing-towards-contrastive-explanations-with-pertinent-negatives
15. Dhurandhar, A., Pedapati, T., Balakrishnan, A., Chen, P., Shanmugam, K., Puri, R.: Model agnostic contrastive explanations for structured data. CoRR **abs/1906.00117** (2019), http://arxiv.org/abs/1906.00117
16. 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 arXiv:1801.01489 (Jan 2018)
17. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. ACM computing surveys (CSUR) **46**(4), 1–37 (2014)
18. Gilpin, L.H., Bau, D., Yuan, B.Z., Bajwa, A., Specter, M., Kagal, L.: Explaining explanations: An overview of interpretability of machine learning. In: 5th IEEE International Conference on Data Science and Advanced Analytics, DSAA 2018, Turin, Italy, October 1-3, 2018. pp. 80–89 (2018). https://doi.org/10.1109/DSAA.2018.00018, https://doi.org/10.1109/DSAA.2018.00018
19. 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), 93:1–93:42 (Aug 2018). https://doi.org/10.1145/3236009, http://doi.acm.org/10.1145/3236009
20. Hinder, F., Hammer, B.: Counterfactual explanations of concept drift (2020)
21. Khandani, A.E., Kim, A.J., Lo, A.: Consumer credit-risk models via machine-learning algorithms. Journal of Banking & Finance **34**(11), 2767–2787 (2010), https://EconPapers.repec.org/RePEc:eee:jbfina:v:34:y:2010:i:11:p:2767-2787
22. Kim, B., Koyejo, O., Khanna, R.: Examples are not enough, learn to criticize! criticism for interpretability. In: Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain. pp. 2280–2288 (2016)
23. Koh, P.W., Liang, P.: Understanding black-box predictions via influence functions. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. pp. 1885–1894 (2017), http://proceedings.mlr.press/v70/koh17a.html
24. Leslie, D.: Understanding artificial intelligence ethics and safety. CoRR **abs/1906.05684** (2019), http://arxiv.org/abs/1906.05684
25. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: A survey on bias and fairness in machine learning. CoRR **abs/1908.09635** (2019), http://arxiv.org/abs/1908.09635
26. Molnar, C.: Interpretable Machine Learning (2019), https://christophm.github.io/interpretable-ml-book/
27. Parisi, G.I., Kemker, R., Part, J.L., Kanan, C., Wermter, S.: Continual lifelong learning with neural networks: A review. Neural Networks **113**, 54 – 71 (2019). https://doi.org/https://doi.org/10.1016/j.neunet.2019.01.012, http://www.sciencedirect.com/science/article/pii/S0893608019300231
28. parliament, E., council: 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 (general data protection regulation). https://eur-lex.europa.eu/eli/reg/2016/679/oj (2016)

29. 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, pp. 3592–3602. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/24aef8cb3281a2422a59b51659f1ad2e-Paper.pdf
30. Reyes-Ortiz, J., Oneto, L., Samà, A., Parra, X., Anguita, D.: Transition-aware human activity recognition using smartphones. Neurocomputing **171** (2016)
31. Ribeiro, M.T., Singh, S., Guestrin, C.: "why should i trust you?": Explaining the predictions of any classifier. In: Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1135–1144. KDD '16, ACM, New York, NY, USA (2016). https://doi.org/10.1145/2939672.2939778, http://doi.acm.org/10.1145/2939672.2939778
32. Samek, W., Wiegand, T., Müller, K.: Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. CoRR **abs/1708.08296** (2017), http://arxiv.org/abs/1708.08296
33. Sample, I.: Computer says no: why making ais fair, accountable and transparent is crucial. https://www.theguardian.com/science/2017/nov/05/computer-says-no-why-making-ais-fair-accountable-and-transparent-is-crucial (2017)
34. 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) (Oct 2020). https://doi.org/10.1145/3419764, https://doi.org/10.1145/3419764
35. Stalidis, P., Semertzidis, T., Daras, P.: Examining deep learning architectures for crime classification and prediction **abs/1812.00602** (2018), http://arxiv.org/abs/1812.00602
36. Tjoa, E., Guan, C.: A survey on explainable artificial intelligence (XAI): towards medical XAI. CoRR **abs/1907.07374** (2019), http://arxiv.org/abs/1907.07374
37. Venkatasubramanian, S., Alfano, M.: The philosophical basis of algorithmic recourse. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency. p. 284–293. FAT* '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3351095.3372876, https://doi.org/10.1145/3351095.3372876
38. Verma, S., Dickerson, J., Hines, K.: Counterfactual explanations for machine learning: A review (2020)
39. Wachter, S., Mittelstadt, B.D., Russell, C.: Counterfactual explanations without opening the black box: Automated decisions and the GDPR. CoRR **abs/1711.00399** (2017), http://arxiv.org/abs/1711.00399
40. Waddell, K.: How algorithms can bring down minorities' credit scores. The Atlantic (2016)
41. Wortman Vaughan, J., Wallach, H.: A human-centered agenda for intelligible machine learning (February 2021), https://www.microsoft.com/en-us/research/publication/a-human-centered-agenda-for-intelligible-machine-learning/, this is a draft version of a chapter in a book to be published in the 2020 - 21 timeframe.

A Proofs and Derivations

- *Proof (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:

  $$\vec{w} = \vec{x}_{\text{cf}} - \vec{x} \tag{28}$$

  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 drift, we can compute the corresponding locally linear approximations of the decision boundaries Eq. (28) and compute the cosine angle between the two weight vectors Eq. (28) as follows:

  $$\begin{aligned}\frac{\vec{w}_1^\top \vec{w}_2}{\|\vec{w}_1\|_2 \|\vec{w}_2\|_2} &= \frac{\left(\vec{x}_{\text{cf}} - \vec{x}\right)^\top \left(\vec{x}_{\text{cf}*} - \vec{x}\right)}{\|\vec{x}_{\text{cf}} - \vec{x}\|_2 \|\vec{x}_{\text{cf}*} - \vec{x}\|_2} \\ &= \frac{\vec{x}_{\text{cf}}^\top \vec{x}_{\text{cf}*} + \vec{x}^\top \vec{x} - \vec{x}_{\text{cf}}^\top \vec{x} - \vec{x}_{\text{cf}*}{}^\top \vec{x}}{\sqrt{\left(\vec{x}_{\text{cf}}^\top \vec{x}_{\text{cf}} + \vec{x}^\top \vec{x} - 2\vec{x}_{\text{cf}}^\top \vec{x}\right)\left(\vec{x}_{\text{cf}*}{}^\top \vec{x}_{\text{cf}*} + \vec{x}^\top \vec{x} - 2\vec{x}_{\text{cf}*}{}^\top \vec{x}\right)}}\end{aligned} \tag{29}$$

  which concludes the proof. □
- *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 [9]:

  $$\vec{x}_{\text{cf}} = \vec{x} - (\vec{w}^\top \vec{x})\vec{w} \tag{30}$$

  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. (30) and $\|w\|_2 = \|\vec{w}_*\|_2 = 1$ yields:

  $$\begin{aligned}\|\vec{x}_{\text{cf}} - \vec{x}_{\text{cf}*}\|_2 &= \|\vec{x} - (\vec{w}^\top \vec{x})\vec{w} - \vec{x} + (\vec{w}_*^\top \vec{x})\vec{w}_*\|_2 \\ &= \|(\vec{w}_*^\top \vec{x})\vec{w}_* - (\vec{w}^\top \vec{x})\vec{w}_* + (\vec{w}^\top \vec{x})\vec{w}_* - (\vec{w}^\top \vec{x})\vec{w}\|_2 \\ &= \|\left((\vec{w}_* - \vec{w})^\top \vec{x}\right)\vec{w}_* + (\vec{w}^\top \vec{x})(\vec{w}_* - \vec{w})\|_2\end{aligned} \tag{31}$$

  Applying the triangle inequality to Eq. (31) yields:

  $$\begin{aligned}\|\vec{x}_{\text{cf}} - \vec{x}_{\text{cf}*}\|_2 &= \|\left((\vec{w}_* - \vec{w})^\top \vec{x}\right)\vec{w}_* + (\vec{w}^\top \vec{x})(\vec{w}_* - \vec{w})\|_2 \\ &\leq \|\left((\vec{w}_* - \vec{w})^\top \vec{x}\right)\vec{w}_*\|_2 + \|(\vec{w}^\top \vec{x})(\vec{w}_* - \vec{w})\|_2 \\ &= |(\vec{w}_* - \vec{w})^\top \vec{x}| \|\vec{w}_*\|_2 + |\vec{w}^\top \vec{x}| \|\vec{w}_* - \vec{w}\|\end{aligned} \tag{32}$$

  Applying the Cauchy-Schwarz inequality and making use of the assumption $\vec{w}_*^\top \vec{w} \leq \epsilon$ to Eq. (32) yields:

  $$\begin{aligned}\|\vec{x}_{\text{cf}} - \vec{x}_{\text{cf}*}\|_2 &\leq |(\vec{w}_* - \vec{w})^\top \vec{x}| \|\vec{w}_*\|_2 + |\vec{w}^\top \vec{x}| \|\vec{w}_* - \vec{w}\|_2 \\ &\leq \|\vec{w}_* - \vec{w}\|_2 \|\vec{x}\|_2 \|\vec{w}_*\|_2 + \|\vec{w}_*\|_2 \|\vec{x}\|_2 \|\vec{w}_* - \vec{w}\|_2 \\ &= 2\|x\|_2 \|\vec{w}_* - \vec{w}\|_2\end{aligned} \tag{33}$$

Substituting $\|\vec{w}_* - \vec{w}\|_2 = 2 - 2\cos(\angle\vec{w}, \vec{w}_*)$ in Eq. (33) yields the stated bound:

$$\begin{aligned}\|\vec{x}_{\text{cf}} - \vec{x}_{\text{cf}*}\|_2 &\leq 2\|x\|_2\|\vec{w}_* - \vec{w}\|_2 \\ &= 2\|\vec{x}\|_2\big(2 - 2\cos(\angle\vec{w}, \vec{w}_*)\big) \\ &= \|\vec{x}\|_2\big(4 - 4\cos(\angle\vec{w}, \vec{w}_*)\big)\end{aligned} \tag{34}$$

which concludes the proof. We could also rewrite Eq. (34) as follows:

$$\|\vec{x}_{\text{cf}} - \vec{x}_{\text{cf}*}\|_2 \leq \|\vec{x}\|_2\big(4 - 4\cos(\angle\vec{w}, \vec{w}_*)\big) \leq \sqrt{8}\|\vec{x}\|_2\big(1 - \cos(\angle\vec{w}, \vec{w}_*)\big)^{1/2} \tag{35}$$

□