Local and Regional Counterfactual Rules: Summarized and Robust Recourses

Salim I. Amoukou 1 2  Nicolas J.B. Brunel 1 3

Abstract

Counterfactual Explanations (CE) face several unresolved challenges, such as ensuring stability, synthesizing multiple CEs, and providing plausibility and sparsity guarantees. From a more practical point of view, recent studies (Pawelczyk et al., 2022) show that the prescribed counterfactual recourses are often not implemented exactly by individuals and demonstrate that most state-of-the-art CE algorithms are very likely to fail in this noisy environment. To address these issues, we propose a probabilistic framework that gives a sparse local counterfactual rule for each observation, providing rules that give a range of values capable of changing decisions with high probability. These rules serve as a summary of diverse counterfactual explanations and yield robust recourses. We further aggregate these local rules into a regional counterfactual rule, identifying shared recourses for subgroups of the data. Our local and regional rules are derived from the Random Forest algorithm, which offers statistical guarantees and fidelity to data distribution by selecting recourses in high-density regions. Moreover, our rules are sparse as we first select the smallest set of variables having a high probability of changing the decision. We have conducted experiments to validate the effectiveness of our counterfactual rules in comparison to standard CE and recent similar attempts. Our methods are available as a Python package.

1. Introduction

In recent years, many explanation methods have been developed for explaining machine learning models, with a strong focus on local analysis, i.e., generating explanations for individual prediction (Molnar, 2022). Among these, Counterfactual Explanations (Wachter et al., 2017) have emerged as a popular technique. In contrast to local attribution methods (Lundberg et al., 2020; Ribeiro et al., 2016), which assign importance scores to each feature, Counterfactuals Explanations (CE) describe the smallest modification to the feature values that changes the prediction to a desired target. These modifications are often called recourses. While CE can be intuitive and user-friendly, they have practical limitations. Most CE methods depend on gradient-based algorithms or heuristic approaches (Karimi et al., 2020a), which may fail to identify the most natural modification and lack guarantees. Most algorithms either do not ensure sparsity (changes to the smallest number of features) or fail to generate plausible samples (Verma et al., 2020; Chou et al., 2022). Several studies (Parmentier & Vidal, 2021; Poyiadzi et al., 2019; Looveren & Klaise, 2019) attempt to address the plausibility and the sparsity issues by incorporating ad-hoc constraints.

In another direction, numerous papers (Mothilal et al., 2020; Karimi et al., 2020b; Russell, 2019) encourage the generation of diverse counterfactuals to find actionable recourse (Ustun et al., 2019). Actionability is a vital desideratum, as some features may be non-actionable, and generating many counterfactuals increases the chance of getting actionable recourse. However, the diversity of CE compromises the intelligibility of the explanation, and the synthesis of various CE, in general, remains an unsolved challenge (Lakkaraju et al., 2022). Recently, (Pawelczyk et al., 2022) highlights a new problem of CE called: noisy responses to prescribed recourses. In real-world scenarios, some individuals may not be able to implement exactly the prescribed recourses, and they show that most CE methods fail in this setting.

Consequently, we propose to reverse the usual way of explaining with counterfactual by computing Counterfactual rules. We introduce a new line of counterfactuals, constructing interpretable policies (or rules) for changing a decision with a high probability while ensuring the stability of the derived recourse. These policies are sparse, faithful to the data distribution and their computation comes with statistical guarantees. Our proposal is to find a general rule that permits changing the decision while fixing some features instead of generating many counterfactual samples. These rules can be seen as a summary of the possible diverse counterfactual samples. Additionally, we show that this method can be extended to create a common counterfactual policy.
for subgroups of the data, which aids model debugging and bias detection. We introduce an algorithm to sample CEs or recourses from the generated rules. Notably, our approach is model-agnostic, meaning it does not need the model to make predictions or calculate other quantities, such as gradients. Instead, it is an inferential approach and relies solely on historical data. As a result, our approach can be applied not only to generate counterfactuals for a specific model but also in scenarios where we don’t have access to the model or when making predictions from it is costly. Our method handles categorical and continuous features and classification and regression problems. In addition, we can use it to generate recourses directly for the data-generating process. An illustration of the counterfactual rules we introduce is illustrated in Figure 1.

Notes on technical novelties. While the Local Counterfactual Rule is a novel concept, the Regional Counterfactual Rule shares similarities with some recent works. Indeed, (Rawal & Lakkaraju, 2020) proposed Actionable Recourse Summaries (AReS), a framework that constructs global counterfactual recourses to have a global insight into the model and detect unfair behavior. Despite similarities with the Regional Counterfactual Rule, there are notable differences. Our methods can handle regression problems and work directly with continuous features and classification and regression problems. AReS requires discretizing continuous features, leading to a trade-off between speed and performance (Ley et al., 2022). Too few bins yield unrealistic recourse, while too many bins result in excessive computation time. AReS employs a greedy heuristic search approach to find global recourse, which may result in unstable and inaccurate recourse. Our approaches overcome these limitations by leveraging on the informative partitions obtained from a Random Forest (RF), removing the need for an extensive search space, and focusing on high-density regions for plausibility. Additionally, we prioritize changes to the smallest number of features by identifying the smallest subset of variables \( S \) and associated value ranges for each observation or subgroup that have the highest probability of changing the prediction. We compute this probability with a consistent estimator of the conditional distribution \( Y|X_S \) obtained from a RF.

2. Minimal Counterfactual Rules

Consider a dataset \( D_n = \{(X_i, Y_i)\}_{i=1}^n \) consisting of i.i.d. observations of \( (X, Y) \sim P_X P_{Y|X} \), where \( X \in \mathcal{X} \) (typically \( \mathcal{X} \subseteq \mathbb{R}^p \)) and \( Y \in \mathcal{Y} \). The output set \( \mathcal{Y} \) can be either discrete or continuous. We denote \( |p| = \{1, \ldots, p\} \), and for a given subset \( S \subseteq |p| \), \( X_S = (X_i)_{i \in S} \) represents a subgroup of features, and we write \( x = (x_S, x_{\bar{S}}) \).

For a given observation \( (x, y) \), we consider a target set \( \mathcal{Y}^* \subseteq \mathcal{Y} \), such that \( y \notin \mathcal{Y}^* \). In the case of a classification problem, \( \mathcal{Y}^* = \{y^*\} \) is a singleton where \( y^* \in \mathcal{Y} \) and \( y^* \neq y \). Unlike conventional approaches, our definition of CE also accommodates regression problems by considering \( \mathcal{Y}^* = [a, b] \subseteq \mathbb{R} \), and the definitions and computations remain the same for both classification and regression. The classic CE problem, defined here only for classification, considers a predictor \( f : \mathcal{X} \rightarrow \mathcal{Y} \), trained on dataset \( D_n \) and search a function \( a : \mathcal{X} \rightarrow \mathcal{X} \), such that for all observations \( x \in \mathcal{X} \), \( f(x) \neq y^* \), we have \( f(a(x)) = y^* \). The function is defined point-wise by solving an optimisation program. Most often \( a(\cdot) \) is not a single-output function, as \( a(x) \) may evade the predicted class or output.

Our contributions can be summarised as follows:

- We redefine the problem of CE generation by introducing Counterfactual Rules that provide summarized and robust CE. We also introduce an algorithm to sample recourses efficiently from these rules.
- Our approach leverages the learned partitions of a Random Forest (RF) as an efficient tool for navigating high-dimensional spaces. It focuses on high-density regions, ensuring the generation of plausible CE. Additionally, it facilitates the identification of the minimal set of features requiring modification to change prediction towards desired outcomes with high probability.
- Our methods offer both local and regional explanations and handle regression and classification tasks.
- We quantitatively demonstrate the effectiveness of our approach across multiple desiderata, such as accuracy, plausibility, sparsity and robustness.

Figure 1: Illustration of local and regional Counterfactual Rules for a fictitious dataset with four variables: Age, Salary, Sex, and HoursPerWeek. Local rules change a single instance’s decision, while regional rules apply to a sub-population. Blue indicates the suggested rules for changing decisions.
be in fact a collection of (random) values \( \{x_1^{CF}, \ldots, x_k^{CF}\} \), which represent the counterfactual samples. A more recent perspective, proposed by (Kanamori et al., 2022), defines \( \mathbf{a} \) as a decision tree, where for each leaf \( L \), a common action is predicted for all instances \( x \in L \) to change their predictions.

Our approach diverges slightly from the traditional model-based definition of CE as we can directly consider observation \((\mathbf{X}, Y)\) rather than model prediction \((\mathbf{X}, f(\mathbf{X}))\). To illustrate the concept, let’s consider a binary classification problem, where the input space can be divided into two regions \( R_0, R_1 \). These correspond to the support of the distributions \( \mathbf{X}|Y = 0 \) and \( \mathbf{X}|Y = 1 \). These regions may not be disjoint or convex spaces and can be represented as a union of several sets. Given an observation \( x = (x_S, x_S) \) with label \( y = 0 \), our method consists of finding the minimal subset of variables \( S \subseteq \{1, \ldots, p\} \) to move \( x \) by modifying \( x_S \) into a set within the region \( R_1 \). The objective is to move \( x \) to a high-density set with low variance with respect to the target variable \( Y \), while altering as few variables as possible. (2021) shows that it coincides with causal counterfactual under appropriate assumptions. In the following discussion, we consider the data \((\mathbf{X}, Y)\) for the presentation of the methods, although they can also be applied to generate recourses for a model prediction \((\mathbf{X}, f(\mathbf{X}))\) as well.

Our approach is hybrid, as we do not suggest a single action for each observation or subspace of \( \mathcal{X} \) but provide sets of possible perturbations. A Local Counterfactual Rule (L-CR) for target \( \mathcal{Y}^* \) and observation \((x, y)\) (with \( y \not\in \mathcal{Y}^* \)) is a rectangle \( C_S(x, \mathcal{Y}^*) = \prod_{i \in S} [a_i, b_i], a_i, b_i \in \mathbb{R} \) such that for all counterfactual samples of \( x = (x_S, x_S) \) obtained as \( x^{CF} = (z_S, x_S) \) with \( z_S \in C_S(x, \mathcal{Y}^*) \) and \( x^{CF} \) an in-distribution sample, then \( y^{CF} \) is in \( \mathcal{Y}^* \) with a high probability, where \( y^{CF} \) is the output of \( x^{CF} \) given by the model \( f \) or the data-generating process. Similarly, a Regional Counterfactual Rule (R-CR) \( C_S(R, \mathcal{Y}^*) \) is defined for target \( \mathcal{Y}^* \) and a rectangle \( R = [1]_{i=1} b_i, a_i, b_i \in \mathbb{R} \), which represent a subspace of \( \mathcal{X} \) of similar observations, if for all observations \( x = (x_S, x_S) \in R \), the counterfactual samples obtained as \( x^{CF} = (z_S, x_S) \) with \( z_S \in C_S(R, \mathcal{Y}^*) \) and \( x^{CF} \) an in-distribution sample are such that \( y^{CF} \) is in \( \mathcal{Y}^* \) with high probability. Our approach constructs such rectangles in a sequential manner. Firstly, we identify the minimal directions \( S \subseteq \{p\} \) that offer the highest probability of changing the decision. Next, we determine the optimal intervals \([a_i, b_i]\) for \( i \in S \) that change the decision to the desired target. Additionally, we propose a method to derive traditional Counterfactual Explanations (CE) (i.e., actions that alter the decision) or recourses using our Counterfactual Rules. A central tool in this approach is the Counterfactual Decision Probability presented below.

**Definition 2.1. Counterfactual Decision Probability (CDP).** The Counterfactual Decision Probability of the subset \( S \subseteq \{1, \ldots, p\} \), w.r.t. \( x = (x_S, x_S) \), output \( y \) and the desired target \( \mathcal{Y}^* \) (s.t. \( y \not\in \mathcal{Y}^* \)) is

\[
CDP_S(x, \mathcal{Y}^*) = \mathbb{P}(Y \in \mathcal{Y}^* | X_S = x_S).
\]

The CDP of the subset \( S \) is the probability that the decision changes to the desired target \( \mathcal{Y}^* \) by sampling the features \( X_S \) given \( X_S = x_S \). It is related to the Same Decision Probability \( SDP_S(\mathcal{Y}; x) = \mathbb{P}(Y \in \mathcal{Y} | X_S = x_S) \) used in (Amoukou & Brunel, 2021) for solving the dual problem of selecting the most local important variables for obtaining and maintaining the decision \( Y \in \mathcal{Y} \), where \( \mathcal{Y} \subset \mathcal{Y}^* \). The set \( S \) is called the Minimal Sufficient Explanation. Indeed, we have \( CDP_S(x, \mathcal{Y}^*) = SDP_S(x, \mathcal{Y}^*) \). The computation of these probabilities is challenging and discussed in Section 3. Next, we define the minimal subset of features \( S \) that allows changing the decision to the target set with a given high probability \( \pi \).

**Definition 2.2. (Minimal Divergent Explanations).** Given an instance \((x, y)\) and a desired target \( \mathcal{Y}^* \not\in y \), \( S \) is a Diver-
gent Explanation for probability $\pi > 0$ if $\text{CDP}_S(x, \mathcal{Y}^*) \geq \pi$, and no subset $Z$ of $S$ satisfies $\text{CDP}_Z(x, \mathcal{Y}^*) \geq \pi$. Hence, a Minimal Divergent Explanation is a Divergent Explanation with the smallest size.

The set satisfying these properties is not unique, and we can have several Minimal Divergent Explanations. Note that the probability $\pi$ represents the minimum level required for a set to be chosen for generating counterfactuals, and its value should be as high as possible and depends on the use case. With these concepts established, we can now define our main criterion for constructing a Local Counterfactual Rule (L-CR).

**Definition 2.3. (Local Counterfactual Rule).** Given an instance $(x, y)$, a desired target $\mathcal{Y}^* \neq y$, a Minimal Divergent Explanation $S$, the rectangle $C_S(x, \mathcal{Y}^*) = \prod_{i \in S} [a_i, b_i]$ is a Local Counterfactual Rule with probability $\pi_C$ if $C_S(x, \mathcal{Y}^*) = \arg\max_C \mathbb{P}_x(X_S \in C \mid X_S = x_S)$ such that $\text{CRP}_S(x, \mathcal{Y}^*) = \mathbb{P}(Y \in \mathcal{Y}^* \mid X_S \in C_S(x, \mathcal{Y}^*), X_S = x_S)$ satisfies

$$\text{CRP}_S(x, \mathcal{Y}^*) \geq \pi_C. \quad (2)$$

$\mathbb{P}_x(X_S \in C_S(x, \mathcal{Y}^*) \mid X_S = x_S)$ represent the plausibility of the rule and by maximizing it, we ensure that the rule lies in a high-density region. $\text{CRP}_S$ is the Counterfactual Rule Probability. The higher the probability $\pi_c$ is, the better the relevance of the rule $C_S(x, \mathcal{Y}^*)$ is for changing the decision to the desired target.

In practice, we often observe that the Local CR $C_S(\cdot, \mathcal{Y}^*)$ for neighboring observations $x$ and $x'$ are quite similar, as the Minimal Divergent Explanations tend to be alike, and the corresponding hyperrectangles frequently overlap. This observation motivates a generalization of these Local CRs to hyperrectangles $R = \prod_{i=1}^d [a_i, b_i]$ which group together similar observations. We denote $\text{supp}(R) = \{ i : [a_i, b_i] \neq \mathbb{R} \}$ as the support of the rectangle and extend the Local CRs to Regional Counterfactual Rules (R-CR). To achieve this, we denote $R_S = \prod_{i \in S} [a_i, b_i]$ as the rectangle with intervals of $R$ in $\text{supp}(R) \cap S$, and define the corresponding Counterfactual Decision Probability (CDP) for rule $R$ and subset $S$ as $\text{CDP}_S(R, \mathcal{Y}^*) = \mathbb{P}(Y \in \mathcal{Y}^* \mid X_S \in R_S)$. Consequently, we can compute the Minimal Divergent Explanation for rule $R$ using the corresponding CDP for rules, following Definition (2.2). The Regional Counterfactual Rules (R-CR) correspond to Definition (2.3) with the associated CDP for rules.

### 3. Estimation of the CDP and CRP

To compute the probabilities $\text{CDP}_S$ and $\text{CRP}_S$ for any $S$, we use a dedicated Random Forest (RF) that learns to predict the output of the model or the data-generating process. Indeed, the conditional probabilities $\text{CDP}_S$ and $\text{CRP}_S$ can be easily computed from a RF by combining the Projected Forest algorithm (Bénard et al., 2021a) and the Quantile Regression Forest (Meinshausen & Ridgeway, 2006). As a result, we can estimate the probabilities $\text{CDP}_S(x, \mathcal{Y}^*)$ consistently. This method has been previously utilized by (Amoukou & Brunel, 2021) for calculating the Same Decision Probability $\text{SDP}_S$.

#### 3.1. Projected Forest and CDPs

The estimator of the $\text{SDP}_S$ is based on the Random Forest (Breiman et al., 1984) algorithm. Assuming that we have trained a RF $m(\cdot)$ using the dataset $D_n$, the model consists of a collection of $k$ randomized trees (for a detailed description of decision trees, see (Loh, 2011)). For each instance $x$, the predicted value of the $l$-th tree is denoted as $m_l(x; \Theta_l)$, where $\Theta_l$ represents the resampling data mechanism in the $j$-th tree and the subsequent random splitting directions. The predictions of the individual trees are then averaged to produce the prediction of the forest as $m(x; \Theta_1, \ldots, \Theta_k) = \frac{1}{k} \sum_{l=1}^k m_l(x; \Theta_l)$. The RF can also be interpreted as an adaptive nearest neighbor predictor (Lin & Jeon, 2006; Biau & Devroye, 2010) or kernel methods (Breiman, 2000; Geurts et al., 2006; Scornet, 2016).

For every instance $x$, the observations in $D_n$ are weighted by $w_{n,i}(x)$, with $i = 1, \ldots, n$. As a result, the prediction of the RF can be reformulated as $m(x; \Theta_1, \ldots, \Theta_k) = \sum_{i=1}^n w_{n,i}(x)Y_i$. This emphasizes the central role played by the weights in the RF’s algorithm. See (Meinshausen & Ridgeway, 2006; Amoukou & Brunel, 2021) for a detailed description of the weights. Consequently, it naturally gives estimators for other quantities, e.g., cumulative hazard function (Ishwaran et al., 2008), treatment effect (Wager & Athey, 2017; Jocteur et al., 2023), conditional density (Du et al., 2021). For instance, the authors also introduced an algorithmic trick that computes the output of the Projected Forest efficiently without modifying the initial tree structures. It consists of dropping the observations down in the initial trees, ignoring the splits that use a variable not in $S$: when it encounters a split involving a variable $i \notin S$, the observations are sent to the left and right
children nodes. Therefore, each instance falls in multiple terminal leaves of the tree. To compute the prediction of $x_S$, we follow the same procedure and gather the set of terminal leaves where $x_S$ falls. Next, we collect the training observations which belong to every terminal leaf of this collection, in other words, we keep only the observations that fall in the intersection of the leaves where $x_S$ falls. Finally, we average their outputs $Y_i$ to estimate $E[Y | X_S = x_S]$. The authors show this algorithm converges asymptotically to the true projected conditional expectation $E[Y | X_S = x_S]$ under suitable assumptions. As the RF, the Projected Forest (PRF) assigns a weight to each training observation. The associated PRF is denoted $m_i^{(S)}(x_S) = \sum_{i=1}^n w_n_i(x_S) Y_i$. Therefore, as the weights of the original forest were used to estimate the CDF, (Amoukou & Brunel, 2021) used the weights of the Projected Forest Algorithm to estimate $SDP$ as $\overline{SDP}_S(x, Y^*) = \sum_{i=1}^n w_{n,i}(x_S) I_{Y_i \in Y^*}$. The idea is essentially to replace $Y_i$ by $I_{Y_i \in Y^*}$ in the Projected Forest equation defined above. (Amoukou & Brunel, 2021) also show that this estimator converges to the true $SDP_S$ under suitable assumptions and works very well in practice. Especially for tabular data, where tree-based models are known to perform well (Grinsztajn et al., 2022). Similarly, we can estimate the CDP with statistical guarantees (Amoukou & Brunel, 2021) using the following estimator $\overline{CDP}_S(x, Y^*) = \sum_{i=1}^n w_{n,i}(x_S) I_{Y_i \in Y^*}$.

**Remark:** We only give the estimator of $CDP_S$ of an instance $x$. The estimator for $CDP_S$ of a rule $R$ will be discussed in the next section, as it is closely related to the estimator of the $CRP_S$.

### 3.2. Regional RF and $CRP_S$

Here, we focus on estimating the $CRP_S(x, Y^*) = \mathbb{P}(Y \in Y^* | X_S \in C_S(x, Y^*), X_S = x_S)$ and $CRP_S(R, Y^*) = \mathbb{P}(Y \in Y^* | X_S \in C_S(R, Y^*), X_S \in R_S)$. For ease of reading, we remove the dependency of the rectangles $C_S$ in $Y^*$. Based on the previous section, we already know that the estimators using the RF will take the form of

$$\overline{CRP}_S(x, Y^*) = \sum_{i=1}^n w_{n,i}(x_S) I_{Y_i \in Y^*},$$

so we only need to determine the appropriate weighting. The main challenge lies in the fact that we have a condition based on a region, e.g., $X_S \in C_S(x)$ or $X_S \in R_S$ (regional-based) instead of a condition of type $X_S = x_S$ (fixed value-based) as usual. However, we introduced a natural extension of the RF algorithm to handle predictions when the conditions are both regional-based and fixed value-based. As a result, cases with only regional-based conditions can be naturally derived.

**Regional RF to estimate** $CRP_S(x, Y^*) = \mathbb{P}(Y \in Y^* | X_S \in C_S(x), X_S = x_S)$. The algorithm is based on a slight modification of RF and works as follows: we drop the observations in the trees, if a split used variable $i \in \bar{S}$, i.e., fixed value-based condition, we use the classic rules of RF, if $x_i \leq t$, the observations go to the left children, otherwise the right children. However, if a split used variable $i \in S$, i.e., regional-based condition, we use the rectangles $C_S(x) = \prod_{i=1}^n [a_i, b_i]$. The observations are sent to the left children if $b_i \leq t$, right children if $a_i > t$ and if $t \in [a_i, b_i]$, the observations are sent both to the left and right children. Consequently, we use the weights of the Regional RF algorithm $w_{n,i}(x)$ to estimate $CRP_S$, the estimator is $\overline{CRP}_S(x, Y^*) = \sum_{i=1}^n w_{n,i}(x) I_{Y_i \in Y^*}$. In addition, the number of observations at the leaves is used as an estimate of $\mathbb{P}(X_S \in C_S(x) | X_S = x_S)$. A more comprehensive description and discussion of the algorithm are provided in the Appendix (A).

To estimate the CDP of a rule $CDP_S(R, Y^*) = \mathbb{P}(Y \in Y^* | X_S \in R_S)$, we just have to apply the Projected Forest algorithm to the Regional RF, i.e., when a split involving a variable outside of $\bar{S}$ is met, the observations are sent both to the left and right children nodes, otherwise we use the Regional RF split rule, i.e., if an interval of $R_S$ is below $t$, the observations go to the left children, otherwise the right children and if $t$ is in the interval, the observations go to the left and right children. The estimator of the $CRP_S(R, Y^*) = \mathbb{P}(Y \in Y^* | X_S \in$
We determine the Minimal Divergent Explanation, akin to Lakkaraju, 2020), on the other hand, relies on binned variables to generate candidate rules, testing all possible rules to focus only on the non-empty leaves containing training observations and their leaves. We then combine the possible ones connected space, which may result in multiple disconnected counterfactual rules.

We apply the same approach to find the regional CR. Given a rule $\mathbf{R}$ and its Minimal Divergent Explanation $S$, we use the Projection given $\mathbf{X}_S \in R_S$ to identify compatible observations and their leaves. We then combine the possible ones that satisfy $\text{CRF}_{S}(\mathbf{R}, \mathcal{Y}^*) \geq \pi_C$ to obtain the regional CR. For instance, if we consider Leaf 5 of the original forest as a rule (i.e., if $\mathbf{X} \in \text{Leaf 5}$, then predict blue), its Minimal Divergent Explanation is also $S = \text{Leaf 1}$. The Regional CR would be the green region in Figure 3(c). Indeed, satisfying the $X_0$ condition of Leaf 5 and the $X_1$ condition of Leaves 3 and 4 would cause the decision to change to green.

5. Sampling CE using the CR

Our approaches cannot be directly compared with traditional CE methods, as they return counterfactual samples, whereas we provide rules (ranges of vector values) that permit changing the decision with high probability. In some applications, users might prefer recourse to CR. Hence, we adapt the CR to generate counterfactual samples using a generative model. For example, given an instance $x = (x_S, x_S)$, target set $\mathcal{Y}^*$ and its counterfactual rule $C_S(x, \mathcal{Y}^*)$, we want to find a sample $x^{\text{CF}} = (x_S, x_S)$ with $x_S \in C_S(x, \mathcal{Y}^*)$ such that $x^{\text{CF}}$ is a realistic sample and $y^{\text{CF}} \in \mathcal{Y}^*$. Instead of using a complex conditional generative model as (Xu et al., 2018; Bénard et al., 2021a), we use the estimators introduced in the previous section. First, we compute the Local and Regional CR, focusing only on the non-empty leaves containing training observations belonging to the projected RF leaf 2 (Figure 3(b)) where $x$ falls, we observe in Figure 3(c) that the optimal rectangle for changing the decision, given $X_0 = x_0$ or being in the projected RF leaf 2, is the union of the intervals on $X_1$ of the leaf 3 and 4 of the RF (see the green region in Figure 3(c)).

Given an instance $x$ and its Minimal Divergent Explanation $S$, the first step is to collect observations that belong to the leaf of the Projected Forest given $S$, where $x$ falls. These observations correspond to those with positive weights in the computation of $CDP_S(x, \mathcal{Y}^*) = \sum_{i=1}^{n} w^R_{n,i}(x_S) \mathbb{I}_{Y_i \in \mathcal{Y}^*}$, i.e., $\{x_i : w^R_{n,i}(x_S) > 0\}$. Then we use the partition of the original forest to find the possible leaves in the direction $S$. The possible leaves are among the RF’s leaves of the collected observations $\{x_i : w^R_{n,i}(x_S) > 0\}$. Let denote $L(X_i)$ the leaf of the observation $x_i$ with $w_{n,i}(x_S) > 0$. A possible leaf is a leaf $L(X_i)$ s.t. $\mathbb{P}(y \in \mathcal{Y}^* | \mathbf{X}_S \in L(X_i) S, \mathbf{X}_S = x_S) \geq \pi_C$. Finally, we merge all the possible neighboring leaves to get the largest rectangle, and this maximal rectangle is the counterfactual rule. It is important to note that the union of possible leaves is not necessarily a connected space, which may result in multiple disconnected counterfactual rules.
We use the default settings of Random Forest and Neural work experiments to enhance plausibility. In all experiments, where the Random Forest is the query model, we add two

Appendix B. The parameters used for AReS are
detailed parameter descriptions, see Appendix (C).

To demonstrate the performance of our framework, we conduct two experiments on real-world datasets. In the first experiment, we showcase the utility of the Local Counterfactual Rules for explaining a regression model. In the second experiment, we compare our approaches with baseline methods in the context of classification problems. We compare the methods only in classification problems, as all prior works do not deal with regression problems. We use the library CARLA (Pawelczyk et al., 2021) designed to benchmark counterfactual explanation methods across different data sets and machine learning models.

We use the default settings of Random Forest and Neural Networks of CARLA as query models. We compute the recourses given by our main competitor, AReS (Rawal & Lakkaraju, 2020), which performs an exhaustive search for finding global counterfactual rules. We use the implementation of (Kanamori et al., 2022) that adapts AReS for returning counterfactual samples instead of rules. In cases where the Random Forest is the query model, we add two popular CF methods for tree-based models, FOCUS (Lucic et al., 2022) and FeatureTweaking (Tolomei et al., 2017) computed by CARLA. We add CHVAE (Pawelczyk et al., 2020), a CF method based on autoencoder for Neural Network experiments to enhance plausibility. In all experiments, we split our dataset into train (75%) - test (25%). We learn a model f, which could be either a Random Forest (RF) or Neural Network (NN), on the train set, which is the model we want to explain. We learn f’s predictions on the train set with an RF (estimators=20, max depth=10) that will be used to generate the Counterfactual Rules (CR) with π = 0.9. The parameters used for AReS are max rules=8, bins=10, and the default parameters set by CARLA for the other. For detailed parameter descriptions, see Appendix (C).

We evaluate the methods on the test set using four metrics. The first, Accuracy, quantifies the average rate at which the recommended actions from each method successfully change the prediction to the desired outcome. The second, Plausibility, measures the average number of counterfactual samples that are classified as inliers by an Isolation Forest trained on the training set. The third, Sparsity, measures the average number of features that have been changed. The fourth metric is the Cost, as defined in (Rawal & Lakkaraju, 2020). For categorical variables, any change is counted as a change of magnitude 1. The continuous features are converted into ordinal features by binning the feature values.

Local counterfactual rules for regression. We apply our approach to the California House Price dataset (n=20640, p=8) (Kaggle, 1997), which contains information about each district such as income, population, and location, and the goal is to predict the median house value of each district. To demonstrate the effectiveness of our Local CR method, we focus on a subset of the test set consisting of 1566 houses with prices lower than 100k. We aim to find recourse that would increase house prices, bringing them within the target range $\mathcal{Y}^* = [200k, 250k]$. For each instance x, we compute the Minimal Divergent Explanation S, the Local CR $C_S(x, [200k, 250k])$, and generate a counterfactual sample using the Simulated Annealing technique described earlier. We succeed in changing the decision for all observations, achieving Accuracy = 100%. Furthermore, the majority of counterfactual samples passed the outlier test, with a Plausibility score of 0.92. Additionally, our Local CR method achieves high sparsity, with Sparsity = 4.45.

For instance, the Local CR for the observation $x = [Longitude=-118.2, latitude=33.8, housing\ median\ age=26, total\ rooms=703, total\ bedrooms=202, population=757, households=212, median\ income=2.52] \ in\ \ percentage\ of\ rooms\ in\ the\ district$ will certainly increase the price. This means that if the number of rooms and total bedrooms satisfy the conditions in $C_S(x, [200k, 250k])$, and the remaining features of x are fixed, then the probability that the price falls within the target set $\mathcal{Y}^* = [200k, 250k]$ is 0.97. This makes sense as increasing the number of rooms and bedrooms in the district will certainly increase the price.

Comparisons of Local and Regional CR with baseline methods. We evaluate our framework on three real-world datasets: Compas (n=6172, p=12) (Washington, 2018) is used to predict criminal recidivism, Diabetes (n=768, p=8) (Kaggle, 2016) aims to predict whether a patient has diabetes or not, and Nhanesi (CDC, 1999-2022) which also aims to predict a disease. Our evaluation reveals that AReS is highly sensitive to the number of bins and the maximal number of
rules or actions, as previously noted by (Ley et al., 2022). Poor parameterization can result in completely ineffective recourses. Furthermore, these methods require separate models for each target class, while our framework only requires a single RF with good precision.

Table 1 shows the results of each method across all datasets and models (RF, NN). The results for the RF are in the first five rows, and those below are for the NN. In each cell, we have two values, the blue corresponding to the metric value for the positive class (label=1) and the red for the negative class (label=0). Table 1 demonstrates that the Local and Regional CR methods achieve high accuracy in changing decisions on all datasets, surpassing baseline methods in almost all experiments. Furthermore, the baselines struggle to change both the positive and negative classes simultaneously, e.g., ARes has Acc=0.72 in the positive class, and 0.15 for the negative class on (RF) - Comps or when they have a good Acc, the CE are not plausible. For instance, FOCUS has Acc=1 for the negative class and Psb=0 in this experiment, meaning that all the counterfactual samples are outliers. These trends are consistently observed across most experiments. We also noted that Focus and FTW fail completely to change the positive class, and CHVAE often does not even present a CF for the positive class. This could be attributed to these methods being more sensitive to imbalances in the data. Our method is the only one that works consistently with the positive and negative classes. Interestingly, baseline methods exhibit improved performance on the NN model compared to the RF model.

Regarding plausibility, our method generally outperforms others, albeit at a higher cost. This is not surprising, as all baseline methods optimise this criterion, unlike ours. In the future, we intend to add this criterion to the CF generation process (Section 5).

We compute the Stability, which is the fraction of unseen instances where the action and perturbed action lead to the same output, for the Comps and Diabetes datasets. We used the simulated annealing approach of Section 5 with the Local CR to generate the actions. The Stability metrics for the different noise levels were 0.98, 0.98, 0.98 for Comps and 0.96, 0.97, 0.96 for Diabetes.

In summary, our CR approach is easier to train, and provides more accurate and plausible rules than the baseline methods. Furthermore, our resulting CE is robust against noisy responses.

7. Conclusion

We propose a novel approach that formulates CE as Counterfactual Rules. These rules are simple policies that can change the decision of an individual or sub-population with a high probability. Our method is designed to learn robust, plausible, and sparse adversarial regions that indicate where observations should be moved to satisfy a desired outcome. Random Forests are central to our approach, as they provide consistent estimates of the probabilities of interest and naturally give rise to the counterfactual rules we seek. This also allows us to handle regression problems and continuous features, making our method applicable to a wide range of data sets where tree-based models perform well, such as tabular data (Grinsztajn et al., 2022). An interesting avenue to explore would be to incorporate the $l_1$ cost into our approach. Currently, our method aims to minimize the $l_0$ distance between the query $x^{obs}$ and the counterfactual $x^{CF}$ by altering as few features as possible. However, deriving a counterfactual observation within a counterfactual rule that minimizes the $l_1$ cost is straightforward with an explicit solution. Given the counterfactual rules (hyperrectangles), represented as a box $(l, r)$, with $l, r \in \mathbb{R}^p$, the following optimization problem $x^{CF} = \arg\min_{x} d(x, x^{obs})$ such that $l \leq x \leq r$ has a closed form solution when the distance is the $l_1$ or $l_2$ norm. The solution is $x^{CF} = \max(l, \min(r, x^{obs}))$ elementwise (Carreira-Peñalh & Hada, 2021). In future work, we will incorporate the $l_1$ constraint and assess the effectiveness of our approach in terms of cost relative to other methods.

Table 1: Results of the Accuracy (Acc), Plausibility (Psb), Sparsity (Sprs), and Cost (Cost) of the different methods. We compute each metric according to the positive (Pos) and negative (Neg) class. The blue value corresponds to the metric for the positive class (recourse for label=1 to label=0), and the red for the negative class (recourse for label=0 to label=1).

| Dataset | Comps | Diabetes | Nameski |
|---------|-------|----------|---------|
| (RF) L-CR | 0.98 / 0.93 | 0.90 / 0.92 | 0.99 / 0.94 | 0.90 / 0.93 | 0.92 / 0.94 | 0.90 / 0.93 |
| (RF) R-CR | 0.97 / 0.98 | 0.93 / 0.88 | 0.92 / 0.99 | 0.90 / 0.86 | 0.90 / 0.87 | 0.90 / 0.86 |
| (RF) ARes | 0.72 / 0.10 | 0.83 / 0.88 | 2.09 / 1.00 | 0.92 / 0.86 | 0.92 / 0.87 | 0.92 / 0.87 |
| (RF) Focus | 0.00 / 0.00 | 0.00 / 0.00 | 8.61 / 7.47 | 0.00 / 0.00 | 0.00 / 0.00 | 0.00 / 0.00 |
| (RF) FTW | N/A / 1.00 | N/A / 0.76 | 1.08 / 1.00 | N/A / 1.00 | N/A / 1.00 | N/A / 1.00 |
| (NN) L-CR | 0.95 / 0.88 | 0.87 / 0.79 | 2.64 / 3.25 | 0.83 / 0.86 | 0.83 / 0.86 | 0.83 / 0.86 |
| (NN) R-CR | 0.91 / 0.89 | 0.82 / 0.91 | 3.36 / 3.29 | 1.21 / 1.00 | 1.21 / 1.00 | 1.21 / 1.00 |
| (NN) ARes | 0.92 / 0.30 | 0.83 / 0.62 | 1.18 / 1.02 | 0.52 / 0.47 | 0.52 / 0.47 | 0.52 / 0.47 |
| (NN) CHVAE | N/A / 1.00 | N/A / 1.00 | N/A / 1.00 | N/A / 1.00 | N/A / 1.00 | N/A / 1.00 |
References

Amoukou, S. I. and Brunel, N. J. Consistent sufficient explanations and minimal local rules for explaining regression and classification models. arXiv preprint arXiv:2111.04658, 2021.

Basu, S., Kumbier, K., Brown, J. B., and Yu, B. Iterative random forests to discover predictive and stable high-order interactions. Proceedings of the National Academy of Sciences, 115(8):1943–1948, 2018.

Bénard, C., Biau, G., Da Veiga, S., and Scornet, E. Shaff: Fast and consistent shapley effect estimates via random forests. arXiv preprint arXiv:2102.13347, 2021c.

Bénard, C., Da Veiga, S., and Scornet, E. Interpretable random forests via rule extraction. In International Conference on Artificial Intelligence and Statistics, pp. 937–945. PMLR, 2021b.

Bérand, C., Biau, G., Veiga, S., and Scornet, E. Mda for random forests: inconsistency, and a practical solution via the sobol-md. arXiv preprint arXiv:2102.13347, 2021c.

Biau, G. and Devroye, L. On the layered nearest neighbour estimate, the bagged nearest neighbour estimate and the random forest method in regression and classification. Journal of Multivariate Analysis, 101(10):2499–2518, 2010. ISSN 0047-259X. doi: https://doi.org/10.1016/j.jmva.2010.06.019. URL https://www.sciencedirect.com/science/article/pii/S0047259X10001387.

Black, E., Yeom, S., and Fredrikson, M. Fliftest: fairness testing via optimal transport. In Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pp. 111–121, 2020.

Breiman, L. Some infinity theory for predictor ensembles. Technical report, Citeseer, 2000.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. Classification and regression trees. wadsworth int. Group, 37 (15):237–251, 1984.

Carreira-Perpiñan, M. Á. and Hada, S. S. Counterfactual explanations for oblique decision trees: Exact, efficient algorithms. In Proceedings of the AAAI conference on artificial intelligence, volume 35, pp. 6903–6911, 2021.

CDC. National health and nutrition examination survey, 1999-2022. URL https://wwwn.cdc.gov/Nchs/NHANES/Default.aspx.

Chou, Y.-L., Moreira, C., Bruza, P., Ouyang, C., and Jorge, J. Counterfactuals and causability in explainable artificial intelligence: Theory, algorithms, and applications. Information Fusion, 81:59–83, 2022. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2021.11.003. URL https://www.sciencedirect.com/science/article/pii/S1566253521002281.

De Lara, L., González-Sanz, A., Asher, N., and Loubes, J.-M. Transport-based counterfactual models. arXiv preprint arXiv:2108.13025, 2021.

Du, Q., Biau, G., Petit, F., and Porcher, R. Wasserstein random forests and applications in heterogeneous treatment effects. In International Conference on Artificial Intelligence and Statistics, pp. 1729–1737. PMLR, 2021.

FICO. Fico. explainable machine learning challenge, 2018. URL https://community.fico.com/s/explainable-machine-learning-challenge.

Geurts, P., Ernst, D., and Wehenkel, L. Extremely randomized trees. Machine learning, 63:3–42, 2006.

Grathwohl, W., Wang, K.-C., Jacobsen, J.-H., Duvenaud, D., Norouzi, M., and Swersky, K. Your classifier is secretly an energy based model and you should treat it like one. In International Conference on Learning Representations, 2020.

Grinsztajn, L., Oyallon, E., and Varoquaux, G. Why do tree-based models still outperform deep learning on typical tabular data? In Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2022.

Guilmeau, T., Chouzenoux, E., and Elvira, V. Simulated annealing: a review and a new scheme. pp. 101–105, 07 2021. doi: 10.1109/SSP49050.2021.9513782.

Haghighi, K., Kogalur, U. B., Blackstone, E. H., and Lauer, M. S. Random survival forests. The annals of applied statistics, 2(3):841–860, 2008.

Jocteur, B.-A., Maume-Deschamps, V., and Ribereau, P. Heterogeneous treatment effect based random forest: Hterf. 2023.

Kaggle. Pima indians diabetes database, 2016. URL https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database.

Kanamori, K., Takagi, T., Kobayashi, K., and Ike, Y. Counterfactual explanation trees: Transparent and consistent actionable recourse with decision trees. In Proceedings of The 25th International Conference on Artificial Intelligence and Statistics, PMLR 151:1846-1870, 2022.

Karimi, A., Barthe, G., Schölkopf, B., and Valera, I. A survey of algorithmic recourse: definitions, formulations, solutions, and prospects. CoRR, abs/2010.04050, 2020a. URL https://arxiv.org/abs/2010.04050.
Karimi, A.-H., Barthe, G., Balle, B., and Valera, I. Model-agnostic counterfactual explanations for consequential decisions. ArXiv, abs/1905.11190, 2020b.

Kelley Pace, R. and Barry, R. Sparse spatial autoregressions. Statistics, Probability Letters, 33(3):291–297, 1997. ISSN 0167-7152. doi: https://doi.org/10.1016/S0167-7152(96)00140-X. URL https://www.sciencedirect.com/science/article/pii/S016771529600140X.

Lakkaraju, H., Slack, D., Chen, Y., Tan, C., and Singh, S. Rethinking explainability as a dialogue: A practitioner’s perspective. CoRR, abs/2202.01875, 2022. URL https://arxiv.org/abs/2202.01875.

Lecun, Y., Chopra, S., and Hadsell, R. A tutorial on energy-based learning. 01 2006.

Ley, D., Mishra, S., and Magazzeni, D. Global counterfactual explanations: Investigations, implementing, and improvements, 2022. URL https://arxiv.org/abs/2204.06917.

Lin, Y. and Jeon, Y. Random forests and adaptive nearest neighbors. Journal of the American Statistical Association, 101(474):578–590, 2006.

Liu, F. T., Ting, K. M., and Zhou, Z.-H. Isolation forest. In 2008 eighth IEEE international conference on data mining, pp. 413–422. IEEE, 2008.

Loh, W.-Y. Classification and regression trees. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 1, 2011.

Looveren, A. V. and Klaise, J. Interpretable counterfactual explanations guided by prototypes. CoRR, abs/1907.02584, 2019. URL http://arxiv.org/abs/1907.02584.

Lucic, A., Oosterhuis, H., Haned, H., and de Rijke, M. Focus: Flexible optimizable counterfactual explanations for tree ensembles. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, pp. 5313–5322, 2022.

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., and Lee, S.-I. From local explanations to global understanding with explainable ai for trees. Nature Machine Intelligence, 2(1):2522–5839, 2020.

Meinshausen, N. and Ridgeway, G. Quantile regression forests. Journal of Machine Learning Research, 7(6), 2006.

Molnar, C. Interpretable Machine Learning. 2 edition, 2022. URL https://christophm.github.io/interpretable-ml-book.

Moithilal, R. K., Sharma, A., and Tan, C. Explaining machine learning classifiers through diverse counterfactual explanations. In Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, FAT* ’20, pp. 607–617, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450369367. doi: 10.1145/3351095.3372850. URL https://doi.org/10.1145/3351095.3372850.

Parmentier, A. and Vidal, T. Optimal counterfactual explanations in tree ensembles. CoRR, abs/2106.06631, 2021. URL https://arxiv.org/abs/2106.06631.

Patki, N., Wedge, R., and Veeramachaneni, K. The synthetic data vault. In 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 399–410, Oct 2016. doi: 10.1109/DSAA.2016.49.

Pawlczyk, M., Broelemann, K., and Kasneci, G. Learning model-agnostic counterfactual explanations for tabular data. In Proceedings of the web conference 2020, pp. 3126–3132, 2020.

Pawlczyk, M., Bieławski, S., van den Heuvel, J., Richter, T., and Kasneci, G. Carla: A python library to benchmark algorithmic recourse and counterfactual explanation algorithms, 2021.

Pawlczyk, M., Datta, T., van-den Heuvel, J., Kasneci, G., and Lakkaraju, H. Algorithmic recourse in the face of noisy human responses, 2022. URL https://arxiv.org/abs/2203.06768.

Poyiadzi, R., Sokol, K., Santos-Rodriguez, R., Bie, T. D., and Flach, P. A. FACE: feasible and actionable counterfactual explanations. CoRR, abs/1909.09369, 2019. URL http://arxiv.org/abs/1909.09369.

Rawal, K. and Lakkaraju, H. Beyond individualized recourse: Interpretable and interactive summaries of actionable recourses. Advances in Neural Information Processing Systems, 33:12187–12198, 2020.

Ribeiro, M. T., Singh, S., and 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, 2016.

Russell, C. Efficient search for diverse coherent explanations. In Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT * ’19, pp. 20–28, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361255. doi: 10.
Scornet, E. Random forests and kernel methods. *IEEE Transactions on Information Theory*, 62(3):1485–1500, 2016.

Tolomei, G., Silvestri, F., Haines, A., and Lalmas, M. Interpretable predictions of tree-based ensembles via actionable feature tweaking. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 465–474, 2017.

Ustun, B., Spangher, A., and Liu, Y. Actionable recourse in linear classification. *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2019.

Verma, S., Dickerson, J. P., and Hines, K. Counterfactual explanations for machine learning: A review. *CoRR*, abs/2010.10596, 2020. URL https://arxiv.org/abs/2010.10596.

Wachter, S., Mittelstadt, B. D., and Russell, C. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Cybersecurity*, 2017.

Wager, S. and Athey, S. Estimation and inference of heterogeneous treatment effects using random forests, 2017.

Washington, A. L. How to argue with an algorithm: Lessons from the compas-propublica debate. *Colo. Tech. LJ*, 17: 131, 2018.

Xu, L., Skoularidou, M., Cuesta-Infante, A., and Veeramachaneni, K. Modeling tabular data using conditional gan. In *NeurIPS*, 2019.
A. Regional RF detailed

In this section, we give a simple application of the Regional RF algorithm to better understand how it works. Recall that the Regional RF is a generalization of the RF's algorithm to give prediction even when we condition given a region, e.g., to estimate \( E(f(\mathbf{X}) | \mathbf{X}_S \in C_S(\mathbf{x}), \mathbf{X}_S = \mathbf{x}_S) \) with \( C_S(\mathbf{x}) = \prod_{i=1}^{S}[a_i, b_i], a_i, b_i \in \mathbb{R} \) a hyperrectangle. The algorithm works as follows: we drop the observations in the initial trees, if a split used variable \( i \in S \), a fixed value-based condition, we used the classic rules, i.e., if \( x_i \leq t \), the observations go to the left children, otherwise the right children. However, if a split used variable \( i \in S \), regional-based condition, we used the hyperrectangle \( C_S(\mathbf{x}) = \prod_{i=1}^{S}[a_i, b_i] \). The observations are sent to the left children if \( b_i \leq t \), right children if \( a_i > t \) and if \( t \in [a_i, b_i] \) the observations are sent both to the left and right children.

To illustrate how it works, we use a two dimensional variables \( \mathbf{X} = (X_0, X_1) \in \mathbb{R}^2 \), a simple decision tree \( f \) represented in Figure 4, and want to compute for \( \mathbf{x} = [1.5, 1.9], E(f(\mathbf{X}) | X_1 \in [2, 3.5], X_0 = 1.5) \). We assume that \( P(X_1 \in [2, 3.5] | X_0 = 1.5) > 0 \) and denoted \( T_1 \) as the set of the values of the splits based on variables \( X_1 \) of the decision tree.

One way of estimating this conditional mean is by using Monte Carlo sampling. Therefore, there are two cases:

\[
\begin{align*}
E(f(\mathbf{X}) | X_1 \in [2, 3.5], X_0 = 1.5) &= P(X_1 \leq 2.9 | X_0 = 1.5) \times E[f(\mathbf{X}) | \mathbf{X} \in L_4] \\
&+ P(X_1 > 2.9 | X_0 = 1.5) \times E[f(\mathbf{X}) | \mathbf{X} \in L_5] 
\end{align*}
\]

Figure 4: Representation of a simple decision tree (right Figure) and its associated partition (left Figure). The gray part in the partition corresponds to the region \([2, 3.5] \times [1, 2]\)

- If \( \forall t \in T_1, t \leq 2 \) or \( t > 3.5 \), then all the observations sampled s.t. \( \tilde{X}_1 \sim P_{\mathbf{X} | X_1 \in [2, 3.5], X_0 = 1.5} \) follow the same path and fall in the same leaf. The Monte Carlo estimator of the decision tree \( E(f(\mathbf{X}) | X_1 \in [2, 3.5], X_0 = 1.5) \) is equal to the output of the Regional RF algorithm.

  - For instance, a special case of the case above is: if \( \forall t \in T_1, t \leq 2 \), and we sample using \( P_{\mathbf{X} | X_1 \in [2, 3.5], X_0 = 1.5} \), then all the observations go to the right children when they encounters a node using \( X_1 \) and fall in the same leaf.

- If \( \exists t \in T_1 \) and \( t \in [2, 3.5] \), then the observations sampled s.t. \( \tilde{X}_1 \sim P_{\mathbf{X} | X_1 \in [2, 3.5], X_0 = 1.5} \) can fall in multiple terminal leaf depending on if their coordinates \( x_1 \) is lower than \( t \). Following our example, if we generate samples using \( P_{\mathbf{X} | X_1 \in [2, 3.5], X_0 = 1.5} \), the observations will fall in the gray region of Figure 4, and thus can fall in node 4 or 5. Therefore, the true estimate is:

\[
E(f(\mathbf{X}) | X_1 \in [2, 3.5], X_0 = 1.5) = P(X_1 \leq 2.9 | X_0 = 1.5) \times E[f(\mathbf{X}) | \mathbf{X} \in L_4] \\
+ P(X_1 > 2.9 | X_0 = 1.5) \times E[f(\mathbf{X}) | \mathbf{X} \in L_5] 
\]

Concerning the last case \((t \in [2, 3.5])\), we need to estimate the different probabilities \( P(X_1 \leq 2.9 | X_0 = 1.5), P(X_1 > 2.9 | X_0 = 1.5) \) to compute \( E(f(\mathbf{X}) | X_1 \in [2, 3.5], X_0 = 1.5) \), but these probabilities are difficult to estimate in practice. However, we argue that we can ignore these splits, and thus do no need to fragment the query region using the leaves of the tree. Indeed, as we are no longer interest in a point estimate but regional (population mean) we do not need to go to the level of the leaves. We propose to ignore the splits of the leaves that divide the query region. For instance, the leaves 4 and 5...
split the region [2, 3.5] in two cells, by ignoring these splits we estimate the mean of the gray region by taking the average output of the leaves 4 and 5 instead of computing the mean weighted by the probabilities as in Equation (3). Roughly, it consists to follow the classic rules of a decision tree (if the region is above or below a split) and ignore the splits that are in the query region, i.e., we average the output of all the leaves that are compatible with the condition \( X_1 \in [2, 3.5], X_0 = 1.5 \).

We think it leads to a better estimation for two reasons. First, we observe that the case where \( t \) is in the region and thus divides the query region does not occur often. Moreover, the leaves of the trees are very small in practice, and taking the mean of observations that fall into the union of leaves that belong to the query region is more reasonable than computing the weighted mean and thus trying to estimate the different probabilities \( \mathbb{P}(X_1 \leq 2.9 \mid X_0 = 1.5), \mathbb{P}(X_1 > 2.9 \mid X_0 = 1.5) \).

B. Sampling Recourses from counterfactual rules

Algorithm 1: Simulated Annealing to generate counterfactual samples using the Counterfactual Rules

\textbf{Input:} Observation \( x \), Divergent Explanation \( S \), counterfactual rule \( C_S(x, \mathcal{Y}^*) \), \( \mathcal{D}_n \) training data set, number of iterations \( \text{maxIter} \), temperature \( T \), cooling rate \( r \)

\textbf{Output:} Inlier sample \( x_{\text{best}} \)

\begin{enumerate}
  \item Set \( x_{\text{current}} \leftarrow x \), and \( x_{\text{best}} \leftarrow x \)
  \item \textbf{for} \( j \in S \) \textbf{do}
    \item \( x_{\text{current}} \leftarrow \text{sample uniformly from the set } \{X_{i,j} : X_i \in \mathcal{D}_n \text{ and } X_{i,S} \in C_S(x, \mathcal{Y}^*) \} \); /* Generate \( x_{\text{current}} \) = \( z_S, x_S \) with \( z_S \) drawn using \( z_S \sim \prod_{S \in \mathcal{S}} \mathbb{P}(X_j \mid X_S \in C_S(x, \mathcal{Y}^*)) \). */
    \item \( x_{\text{best}} \leftarrow x_{\text{current}} \); /* Initialize \( x_{\text{best}} \) */
  \item \textbf{end for}
  \item \textbf{for} \( i \) from 1 to \( \text{maxIter} \) \textbf{do}
    \item \( x_{\text{new}} \leftarrow x_{\text{current}} \)
    \item \( S' \leftarrow \text{sample uniformly from the set } S \)
    \item \textbf{for} \( j \) in \( S' \) \textbf{do}
      \item \( x_{\text{new}} \leftarrow \text{sample uniformly from the set } \{X_{i,j} : X_i \in \mathcal{D}_n \text{ and } X_{i,S} \in C_S(x, \mathcal{Y}^*) \} \)
    \item \textbf{end for}
    \item Compute the Outlier score difference \( \Delta O \) between \( x_{\text{new}} \) and \( x_{\text{current}} \)
    \item \textbf{if} \( \Delta O < 0 \text{ or } \exp(-\Delta O/T) > \text{random}(0,1) \) \textbf{then}
      \item Set \( x_{\text{current}} \leftarrow x_{\text{new}} \)
    \item \textbf{end if}
    \item \textbf{if} Outlier score of \( x_{\text{best}} \) < Outlier score of \( x_{\text{current}} \) \textbf{then}
      \item Set \( x_{\text{best}} \leftarrow x_{\text{current}} \)
    \item \textbf{end if}
    \item Decrease \( T \) by \( T = T \ast r \)
  \item \textbf{end for}
\end{enumerate}

\textbf{return} \( x_{\text{best}} \)

C. Additional experiments

In table 2, we compare the \textit{Accuracy} (Acc), \textit{Plausibility} (Psb), and \textit{Sparsity} (Sprs) of the different methods on additional real-world datasets: FICO (FICO, 2018), NHANESI (CDC, 1999-2022).

We observe that the L-CR, and R-CR outperform the baseline methods by a large margin on \textit{Accuracy} and \textit{Plausibility}. The baseline methods still struggle to change at the same time the positive and negative class. ArEs and CET give better sparsity, but their counterfactual samples are less plausible than the ones generated by the CR.
Table 2: Results of the *Accuracy* (Acc), *Plausibility*, and *Sparsity* (Sprs) of the different methods. We compute each metric according to the positive (Pos) and negative (Neg) class.

|        | FICO |             | NHANESI |             |
|--------|------|-------------|---------|-------------|
|        | Acc  | Psb | Sps | Acc  | Psb | Sps | Acc  | Psb | Sps |
| Pos    | Neg  |     |     | Pos    | Neg  |     | Pos    | Neg  |     |
| L-CR   | 0.98 | 0.94| 0.98| 9      | 8.43 | 9   | 0.99  | 0.98 | 0.97 |
| R-CR   | 0.90 | 0.94| 0.98| 9      | 8.43 | 9   | 0.86  | 0.95 | 0.96 |
| AReS   | 0.34 | 0.01| 0.86| 2      | 1    | 0.06| 0.06  | 1    | 0.87 |
| CET    | 0.76 | 0    | 0.60| 2      | 2    | 0   | 0.40  | 0.82 | 0.56 |

D. Parameters detailed

In this section, we give the different parameters of each method. For all methods and datasets, we first used a greedy search given a set of parameters. For AReS, we use the following set of parameters:

- max rule = \{4, 6, 8\}, max rule length = \{4, 8\}, max change num = \{2, 4, 6\}.
- minimal support = 0.05, discretization bins = \{10, 20\},
- \(\lambda_{acc} = \lambda_{cov} = \lambda_{cst} = 1\).

Lastly, for the Counterfactual Rules, we used the following parameters:

- nb estimators = \{20, 50\}, max depth = \{8, 10, 12\},
- \(\pi = 0.9, \pi_C = 0.9\).

We obtained the same optimal parameters for all datasets:

- AReS: max rule = 4, max rule length = 4, max change num = 4, minimal support = 0.05, discretization bins = 10, \(\lambda_{acc} = \lambda_{cov} = \lambda_{cst} = 1\)
- CET: max iterations = 1000, max leaf size = -1, \(\lambda = 0.01, \gamma = 1\)
- CR: nb estimators = 20, max depth = 10, \(\pi = 0.9, \pi_C = 0.9\)