Proceedings of Machine Learning Research 266:1–20, 2025 Conformal and Probabilistic Prediction with Applications

Counterfactual Explanations for Conformal Prediction Sets

Aicha Maalej AICHA.MAALEJ@JU.SE
Dept. of Computing, Jönköping University, Sweden
School of Informatics, University of Skövde, Sweden

Cecilia Sönströd CECILIA.SONSTROD@JU.SE
Dept. of Computing, Jönköping University, Sweden

Ulf Johansson ULF.JOHANSSON@JU.SE
Dept. of Computing, Jönköping University, Sweden

Editor: Khuong An Nguyen, Zhiyuan Luo, Harris Papadopoulos, Tuwe Löfström, Lars Carlsson and Henrik Boström

Abstract

Conformal classification outputs prediction sets with formal guarantees, making it suitable for uncertainty-aware decision support. However, explaining such prediction sets remains an open challenge, as most existing explanation methods, including counterfactual ones, are tailored to point predictions. In this paper, we introduce a novel form of counterfactual explanations for conformal classifiers. These counterfactuals identify minimal changes that modify the conformal prediction set at a fixed significance level, thereby explaining how and why certain classes are included or excluded. To guide the generation of informative counterfactuals, we consider *proximity*, *sparsity*, and *plausibility*. While proximity and sparsity are commonly used in the literature, we introduce *credibility* as a new measure of how well a counterfactual conforms to the underlying data distribution, and hence its plausibility. We empirically evaluate our method across multiple tabular datasets and optimization criteria. The findings demonstrate the potential of using counterfactual explanations for conformal classification as informative and trustworthy explanations for conformal prediction sets.

Keywords: Conformal prediction, Explainable AI (XAI), Counterfactual explanations.

1. Introduction

In data-driven decision making, predictive modeling is often used to learn patterns from observed data and generalize these patterns to unseen cases. The primary goal of such models is to make accurate predictions that support informed decisions. However, accuracy alone is not always sufficient. In many applications, decision makers are not only concerned with the model's predictions and confidence, but also expect the predictions to be informative.

To address the need for informativeness, conformal prediction provides a formal framework for generating prediction regions with statistical guarantees (Vovk et al., 2005). Rather then returning a point prediction output, a conformal classifier produces a set of plausible labels that contain the true target with a user-specified probability. These prediction sets allow for an explicit representation of model uncertainty. The informativeness of a prediction is, consequently, reflected in the size of its conformal set: small sets indicate high certainty, whereas larger sets indicate the opposite. This makes conformal prediction particularly suitable for decision-making contexts where understanding uncertainty is as important as the prediction itself.

While conformal prediction has been widely used for its theoretical guarantees and the informativeness of the conformal set predictions produced, the outputs it generate remain difficult to assess. In particular, there is limited support for explaining why a given prediction set includes or excludes certain labels.

In recent years, the field of explainable machine learning has produced a range of techniques to increase the interpretability of black-box models. One particularly intuitive form of explanation is the *counterfactual* explanation: given a prediction, a counterfactual describes a minimal change to the feature values that would result in a different prediction (Wachter et al., 2017). Counterfactuals contribute to the highest level of interpretability according to Pearl's causal hierarchy (Pearl, 2009) and are especially valuable for individual-level interpretability and actionable insights.

Existing counterfactual methods are typically designed for point predictors, where the goal is to flip the predicted class label. However, in the context of conformal prediction, where outputs are sets rather than single labels, existing methods fall short, and despite growing interest in conformal prediction, existing counterfactual methods cannot explain changes in prediction set outputs.

To address this gap, we argue that counterfactual explanations for conformal prediction that target the inclusion or exclusion of specific labels in the prediction set enable more fine-grained insights into model uncertainty and individual label decisions. Moreover, set-based counterfactuals often entail smaller, more plausible changes in feature space than those that enforce a complete class switch, resulting in explanations that are both more informative and realistic.

In this paper, we propose a novel integration of counterfactual explanations into the conformal prediction framework. Specifically, we introduce a method that generates counterfactual instances that alter the conformal prediction set under a fixed confidence level. This offers explanations that reveal how small changes in the input can affect the uncertainty in predictions, not just the outcome. As part of this contribution, we also propose using the metric *credibility* (Vovk et al., 2005) to quantify the degree to which a counterfactual aligns with the training data distribution.

In the next section 2, we present background information on explanations, counterfactual explanations, and conformal prediction, and review related work. Section 3 describes the proposed method, including the optimization criteria and the search strategies used to identify the counterfactuals. Section 4 presents the results of our empirical evaluation across multiple datasets. Finally, Section 5 concludes the paper and outlines directions for future work.

2. Background

2.1. Explanations

Interpretable machine learning aims to provide human-understandable insights into how a model arrives at its predictions. Approaches to interpretability can generally be divided into two categories: inherently interpretable models and post-hoc explanation methods for complex or opaque models (Molnar et al., 2020).

Inherently interpretable models, such as decision trees or linear regression, are transparent by design. Their structure can be directly inspected and understood, which allows

users to trace how input feature values contribute to a specific prediction. For example, a decision tree provides a sequence of if-then-else rules from root to leaf, while linear models explicitly assign weights to each feature. However, such models often underperform in terms of predictive accuracy when compared to more complex models like neural networks, support vector machines, or ensemble methods.

Most high-performance models are typically treated as black boxes because their internal mechanisms are too complex for human interpretation. To make such models more interpretable, post-hoc explainability techniques are used. These methods are applied after the model has been trained and are designed to explain either individual predictions (local explanations) or the general behavior of the model (global explanations).

Some global explainability techniques attempt to characterize the patterns learned by the model (Koh et al., 2020), find simpler models learned from the representation of complex models (Dhurandhar et al., 2018), or find prototypical points that summarize a dataset (Kim et al., 2016).

Local explainability aims to explain why a model made a specific prediction for a given instance. Among the most commonly used local explainability methods in the literature are the following:

- Feature importance (Lundberg and Lee, 2017; Ribeiro et al., 2016): These methods look at which features are important to the model when performing predictions. Common techniques under this approach are Local Interpretable Model Agnostic Explanations (LIME) (Ribeiro et al., 2016), and SHapley Additive exPlanations (SHAP) (Lundberg and Lee, 2017).

- Training point importance (Koh and Liang, 2017; Yeh et al., 2018): These methods ask the question: which data points in the training dataset were the most important or influential to a particular prediction?

- Counterfactual explanation (Wachter et al., 2017): These techniques answer the question: what minimal change to the input is needed to change a model's prediction?

In summary, feature and training point importance typically aim to convey the internal logic or reasoning of a model that leads to a prediction. In contrast, counterfactuals illustrate how changes in feature values would alter that prediction (Wachter et al., 2017).

2.2. Counterfactuals

Counterfactual explanations are applicable to supervised machine learning, and most research in this area has applied counterfactuals to classification tasks. Here, models are trained to predict discrete labels based on input features, and counterfactuals offer insights into how minimal changes to feature values could lead to a different predicted label. A formal representation of a counterfactual explanation as proposed by Wachter et al. (2017) is as follows: given an input x, a model f, and a distance metric d, a counterfactual explanation c is found by solving the optimization problem in Equation 1:

$$\min_{c \in X} d(x, c) \quad \text{subject to} \quad f(x) \neq f(c) \tag{1}$$

2.2.1. Criteria for counterfactuals

To systematically study the quality of counterfactuals, research has focused on identifying the criteria that counterfactual explanations should satisfy to ensure that the resulting explanations are truly actionable and trustworthy.

- Proximity: a counterfactual should differ as little as possible from the original instance. By minimizing the overall change in feature values, the counterfactual remains close in feature space, making it more realistic and easier to interpret and act upon.
- Plausibility (Data Manifold): a meaningful counterfactual should lie within or near the distribution of the training data. If a counterfactual falls into regions where the model has never seen examples, it risks being unrealistic and untrustworthy. For instance, proposing a set of feature values that never occurred in the training set may yield misleading or even invalid suggestions.
- Sparsity: to be interpretable and easy to implement, a counterfactual should involve changing as few features as possible. A sparse explanation, where only one or two attributes are modified, is generally preferred over complex changes affecting multiple features simultaneously, as it is more likely to be acted upon by a decision maker.
- Actionability: a counterfactual is considered actionable if it prescribes modifications to features that can realistically be changed. For instance, suggesting an increase in income or a reduction in debt is actionable, whereas recommending a change in birthplace or ethnicity is not. Actionable counterfactuals are crucial for ensuring that the provided explanations are practically feasible for the individual.
- Causality: the features in a dataset are often dependent of each other, and changing one feature value will probably affect other feature values. For example, obtaining a driver's license typically requires reaching a minimum legal age. A realistic counterfactual should respect such constraints, rather than suggesting a license acquisition without meeting the age requirement.
- Speed: in many decision support applications, counterfactuals must be generated quickly for new, incoming data points to be useful in real-time settings.
- Model Access: some counterfactual explanation approaches require detailed knowledge of model internals (model specific techniques). Others can work in a black-box fashion and are model-agnostic.

2.3. Conformal classification

Conformal prediction produces prediction regions with guarantees. In the classification setting, a conformal classifier produces prediction sets that are statistically valid by design, i.e., given a significance level $\epsilon \in (0, 1)$, the error rate of a conformal predictor will, in the long run, be exactly ϵ. Here, an error is committed when the prediction set does not contain the true label.

In conformal classification, prediction sets are constructed by testing each possible label for a given test instance. Each candidate label $\tilde{y}$ is temporarily assigned to the test input

x_{k+1}, forming the pair $(x_{k+1}, \tilde{y})$. A p-value is then computed to assess whether this candidate label can be statistically rejected as the true label, given a predefined significance level ϵ. This process is repeated for all labels in $\mathcal{Y}$, resulting in a prediction set $\tilde{Y} \subseteq \mathcal{Y}$ that includes all labels that were not rejected. Under the assumption of exchangeability, the resulting prediction set is guaranteed to contain the true label y_{k+1} with probability of at least $1 - \epsilon$.

To produce prediction sets, conformal classification relies on a *nonconformity function* $\mathrm{A} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to measure how 'strange' the instance $(x,\tilde{y})$ is compared to a set of instances with known target values. The label is rejected if the p-value for this combination is lower than ϵ.

In this paper, inductive conformal prediction (ICP) is used. In ICP, a single model is generated from a training set and subsequently used to produce predictions for all test instances in batch mode. To obtain valid prediction sets, ICP requires an additional labeled dataset, called the *calibration set*, which is not used during model training.

In our implementation, we employ the `WrapClassifier` from the *Crepes* Python package (Boström, 2024) to construct the conformal predictor. This wrapper implements the *hinge loss* as the default nonconformity function:

$$\Delta(h(x_i), \tilde{y}) = 1 - \hat{P}_h(\tilde{y} \mid x_i), \tag{2}$$

where $\hat{P}_h(\tilde{y} \mid x_i)$ is the probability assigned by the underlying classifier h to the instance x_i and the candidate label $\tilde{y}$.

To train an ICP for classification (Papadopoulos et al., 2002; Vovk et al., 2005; Papadopoulos, 2008), we use the following procedure:

1. Divide the available labeled training data Z into two disjoint subsets: a proper training set Z^t and a calibration set Z^c, where $|Z^c| = q$.

2. Train the underlying machine learning model h using the proper training set Z^t.

3. Apply the nonconformity function (see Eq. 2) to all examples in the calibration set Z^c, producing calibration scores $\alpha_1, \ldots, \alpha_q$.

To predict the label set for a test instance x_{k+1}:

1. Obtain the probabilistic prediction $h(x_{k+1})$ from the underlying model.

2. For each possible label $\tilde{y} \in \mathcal{Y}$, compute the nonconformity score of $(x_{k+1}, \tilde{y})$.

3. Compute the corresponding p-value:

$$p^{\tilde{y}}_{k+1} = \frac{\left|\left\{z_i \in Z^c : \alpha_i \geq \alpha^{\tilde{y}}_{k+1}\right\}\right| + 1}{q+1} + \theta_{k+1} \frac{\left|\left\{z_i \in Z^c : \alpha_i = \alpha^{\tilde{y}}_{k+1}\right\}\right|}{q+1}, \tag{3}$$

   where $\theta_{k+1} \sim \mathcal{U}[0,1]$.

4. Compare the p-values to the significance level ϵ; reject all labels $\tilde{y}$ such that $p^{\tilde{y}}_{k+1} < \epsilon$.

5. Return the set of non-rejected labels as the final prediction set Γ^{ϵ}_{k+1}, defined as:

$$\Gamma^{\epsilon}_{k+1} = \left\{ \tilde{y} \in \mathcal{Y} : p^{\tilde{y}}_{k+1} \geq \epsilon \right\}$$

Using this procedure, and under the assumption of exchangeability between the calibration and test data, the conformal prediction sets will include the true label with probability exactly $1 - \epsilon$.

A complementary approach to assessing conformal classifications is to use the metrics *confidence* and *credibility*. These are derived from the vector of p-values associated with a test instance. Given a test instance x, the conformal predictor produces a vector of p-values $\mathbf{p}(x) = [p_1(x), p_2(x), \ldots, p_K(x)]$, where $p_k(x)$ quantifies how well the instance conforms to class k. The *credibility* is defined as the largest p-value, i.e., $\max_k p_k(x)$, and reflects the overall conformity of the test instance across all labels. The *confidence* is defined as one minus the second largest p-value, and corresponds to the smallest significance level at which a singleton prediction is obtained.

In the conformal setting, credibility indicates how well a test instance prediction, comprising the input features and predicted label set, aligns with the training data distribution. Accordingly, the credibility of a counterfactual provides a direct measure of its plausibility. In our approach, we introduce credibility as a novel metric for both evaluating and optimizing the quality of counterfactual explanations.

2.4. Related Work

All existing counterfactual explanation methods assume point predictive models and aim to identify minimal changes in feature values that flip the predicted class label. These methods optimize properties such as proximity and sparsity (Wachter et al., 2017), actionability (Ustun et al., 2019; Mothilal et al., 2020), plausibility (Artelt and Hammer, 2020), and causal validity (Karimi et al., 2021), often using optimization or heuristic search strategies (Guidotti, 2024).

More recent approaches have attempted to incorporate aspects of uncertainty into counterfactual generation using Bayesian modeling or epistemic uncertainty quantification (Raman et al., 2023). However, these models provide probabilistic scores without distribution-free guarantees, and none of them are situated within the conformal framework. Another study by Löfström et al. (2024) proposes a method that provides uncertainty-aware factual and counterfactual explanations of classification predictions. Their approach highlights the impact of feature perturbations on calibrated probability intervals rather than labels or label sets.

Beyond explanation methods, some techniques have been proposed to reduce the size of prediction sets while preserving the coverage guarantee. For instance Romano et al. (2019) proposed a method that constructs sharper prediction intervals by combining quantile regression with conformal prediction.

The notion of *credibility*, defined as the highest p-value across candidate labels in conformal prediction, has been previously used to assess prediction sets (Vovk et al., 2005), but has not been explored as a measure to guide counterfactual generation. In this context, our work is the first to: (i) formalize counterfactual explanations for conformal classifiers

by introducing a method that generates counterfactual instances that alter the conformal prediction set under a fixed confidence level, and (ii) propose *credibility* as a measure of plausibility by directly quantifying how well a counterfactual conforms with the training data.

3. Method

The purpose of the paper is to investigate how counterfactual explanations can be used to explain the prediction sets produced by conformal classifiers. Specifically, we propose a method for generating counterfactual explanations tailored to conformal prediction sets. These explanations identify minimal input changes that alter the conformal prediction set at a fixed confidence level, thereby interpreting how and why specific classes are included or excluded. Our approach also optimizes *proximity* (sometimes referred to as *distance*), *sparsity*, and *plausibility* as criteria for counterfactual explanations.

As described above, given an input instance (test instance) $x \in \mathcal{X}$, a conformal predictor Γ^ε returns a prediction set $\Gamma^\varepsilon(x) \subseteq \mathcal{Y}$, which includes all labels not rejected at significance level ε. We define a *conformal counterfactual* as an instance $x' \in \mathcal{X}$ such that $\Gamma^\varepsilon(x') \neq \Gamma^\varepsilon(x)$. The instance x' can then be optimized according to a specified criterion, depending on the goal of the explanation.

Our method consists of the following three steps:

1. Identify the closest real counterfactual instance x^{real} from the dataset.
2. Construct a discrete search space using observed feature values from x and x^{real}.
3. Search within the constructed space to generate a surrogate counterfactual x' that changes the prediction set and optimizes one of three criteria: *distance*, *sparsity*, or *credibility*.

We describe each of these steps in detail in the following subsections; for an overview, see the pseudo code in Algorithm 1.

3.1. Conformal classification model

We adopt ICP for classification, using the WrapClassifier from the crepes Python library. The data is split into training, calibration, and test sets. We train a Random Forest classifier on the training set, then calibrate it using Leave-One-Out (LOO) cross-validation on the calibration set. For the examples below, we use the significance level $\epsilon = 0.05$

3.2. Finding a real counterfactual

To ensure plausible counterfactuals, we first search for a *real counterfactual* x^{real} within the calibration set. Specifically, for a given test instance x, we identify the closest calibration instance x^{real} (measured using standardized Euclidean distance) such that it has a different conformal prediction set:

$$x^{\text{real}} = \arg\min_{x_i \in Z^c} \left\{ \|x - x_i\|_2 \;\middle|\; \Gamma^\varepsilon(x_i) \neq \Gamma^\varepsilon(x) \right\}, \tag{4}$$

Algorithm 1 Counterfactual Generation for Conformal Prediction Sets

Input: Test instance x, Conformal predictor Γ^ϵ, Calibration set Z_c, Training set Z_t, Distance metric D, Optimization criterion C, Maximum search size T

Output: Counterfactual x' such that $\Gamma^\epsilon(x') \neq \Gamma^\epsilon(x)$ and $C(x')$ is optimized

Compute prediction set $\Gamma^\epsilon(x)$
Find real counterfactual $x_{\text{real}} = \arg\min_{z \in Z_c,\ \Gamma^\epsilon(z) \neq \Gamma^\epsilon(x)} D(x, z)$
Let d be the number of input features
Construct search space $S = \prod_{i=1}^{d} V_i$, where each $V_i = \{z_i \mid z \in Z_t \cup Z_c,\ \min(x_i, x_{\text{real},i}) \leq z_i \leq \max(x_i, x_{\text{real},i})\}$
Define objective: Find $x' \in S$ such that $\Gamma^\epsilon(x') \neq \Gamma^\epsilon(x)$ and $C(x')$ is maximized
if $|S| < T$ **then**
  Perform exhaustive search over S to find x'
end
else
  Use a Genetic Algorithm to search over S and find x'
end
return x'

where Z^c denotes the calibration set. It should be noted that in this step, the prediction sets of the calibration instances are derived from the internal LOO cross-validation described above.

3.3. Surrogate counterfactual search

Using the original instance x and the real counterfactual x^{real}, we construct a discrete search space. For each feature j, we extract the set of observed values in the dataset that lie between x_j and x_j^{real}, inclusive. The search is then restricted to combinations of these values to ensure that candidate counterfactuals have realistic feature values and remain within a plausible region of the input space.

If the total number of combinations is below a predefined threshold (here set to 10000), an exhaustive search is performed; we generate all possible combinations of candidate feature values and select the best counterfactual according to the optimization criterion used. If the number of combinations becomes computationally infeasible, we instead employ a genetic algorithm (GA) for the search implemented via the *geneticalgorithm* Python package[1]. We use the following parameters: population size = 100, mutation probability = 0.1, elitism ratio = 0.01, crossover probability = 0.5, and up to 1000 iterations. The fitness function corresponds to the selected optimization criterion combined with a large penalty if the candidate is not valid.

In both cases, a candidate x' is considered valid if it changes the conformal prediction set:

$$\Gamma^\varepsilon(x') \neq \Gamma^\varepsilon(x). \tag{5}$$

1. https://github.com/rmsolgi/geneticalgorithm While more efficient genetic algorithm implementations are available, we chose this one based on its ease of use and accessibility, which we consider appropriate for the purposes of our proof-of-concept study.

Surrogate counterfactuals are thus constructed by combining feature values observed in the data between the original test instance and a real counterfactual. This results in a discrete search space composed of plausible values for each feature, from which new (synthetic) instances are generated through either exhaustive or heuristic search.

3.4. Optimization criteria

We generate a surrogate counterfactual from the constructed search space that modifies the original prediction set, and optimizes one of the following criteria:

- *Proximity* (distance): minimize the distance between the standardized feature vectors of the original and surrogate instances thereby ensuring minimal changes. For datasets with only numerical features, we use the ℓ_2 norm:

$$\text{Proximity}(x, x') = \|x - x'\|_2 \tag{6}$$

  For mixed-type data (e.g., numerical and categorical), we use the Gower distance:

$$\text{Proximity}_{\text{Gower}}(x, x') = \frac{1}{p} \sum_{j=1}^{p} d_j(x_j, x'_j), \tag{7}$$

  where p is the total number of features and d_j is a normalized distance function: for categorical features, $d_j(x_j, x'_j) = 0$ if $x_j = x'_j$ and 1 otherwise; for numerical features, $d_j(x_j, x'_j)$ is the absolute difference between x_j and x'_j, scaled by the observed range of feature j.

- *Sparsity*: minimize the number of features changed relative to the original instance:

$$\text{Sparsity}(x, x') = \sum_{j=1}^{p} \mathbf{1}\{x_j \neq x'_j\} \tag{8}$$

- *Credibility*: maximize the *credibility* of the counterfactual using:

$$\text{Credibility}(x') = \max_k p_k(x'), \tag{9}$$

  where $p_k(x')$ is the p-value computed by the conformal predictor for class k at x'. Optimizing for *credibility* encourages the generation of counterfactuals that conform to training and calibration instances, thus improving the counterfactual's plausibility. Here it must be noted that since the search space is limited to the discrete grid described above, even counterfactuals generated from optimizing credibility are likely to lie between the real counterfactual and the original instance.

In the experimentation, the search for a surrogate counterfactual is guided by one of these criteria. In cases where a real counterfactual was found in the calibration set but the optimization procedure fails to find a better surrogate counterfactual, the real counterfactual is returned. In the very rare case when a real counterfactual cannot be found, no counterfactual is returned.

The 11 benchmarking data sets used (see Table 1) are all publicly available from either the UCI repository (Bache and Lichman, 2013) or the PROMISE Software Engineering Repository (Sayyad Shirabad and Menzies, 2005).

Table 1: Summary of datasets

Dataset	#Instances	#Features	#Classes	Source
adult income	1000	11	2	UCI
diabetes	768	9	2	UCI
german	1000	21	2	UCI
iris	150	4	3	UCI
kc1	1192	21	2	PROMISE
kc2	522	22	2	PROMISE
kc3	325	39	2	PROMISE
liver	345	7	2	UCI
sonar	208	61	2	UCI
vote	435	17	2	UCI
wbc	699	10	2	UCI

4. Results

4.1. Examples

To illustrate the differences among counterfactuals generated by optimizing for *credibility*, *distance*, and *sparsity*, we present some examples from four datasets: Diabetes, Wisconsin Breast Cancer (WBC), Iris, and Congressional Votes. For each case, we contrast the original instance and its counterfactual, and we show how the prediction set, p-values, confidence, credibility, and feature values change. These examples also demonstrate how the resulting explanations can be interpreted in their domain context.

Distance-optimized counterfactuals (e.g., Fig. 1 and 2) are constructed to remain close to the original input in Euclidean space. For instance, in both the Diabetes and Breast Cancer datasets, the counterfactuals produced contain relatively small changes, but often for several feature values. In the Diabetes example in Fig. 1, the prediction set shifts from {*Healthy, Diabetic*} to {*Healthy*} by slightly reducing 'Glucose', 'Blood pressure', 'BMI', and 'DPF' values, which are features clinically linked to diabetes risk. So, for a patient with these lower values, it would have been possible to reject the label 'Diabetic', at the chosen significance level. Similarly, in the Breast Cancer (WBC) example in Fig. 2, decreasing the values of 'Bland Chromatin' from 4 to 2 and 'Normal Nucleoli' from 8 to 7 moves the instance closer to the benign region of the feature space. The conformal predictor consequently expands the prediction set to include 'Benign', which reflects the increased uncertainty but is also consistent with medical understanding that lower values of these features are associated with Benign tumors.

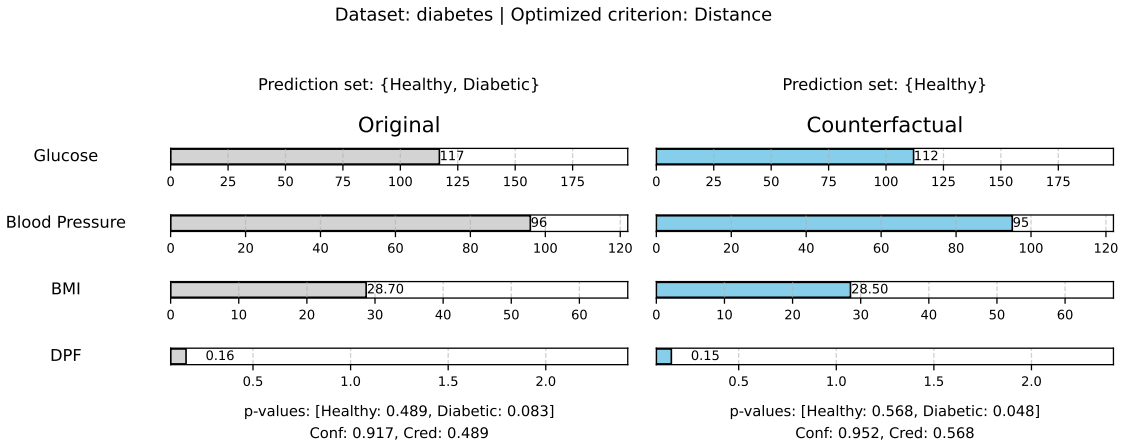

Figure 1: Counterfactual explanation for a conformal prediction set generated by optimizing for *distance* on the Diabetes dataset. Only changed features are shown.

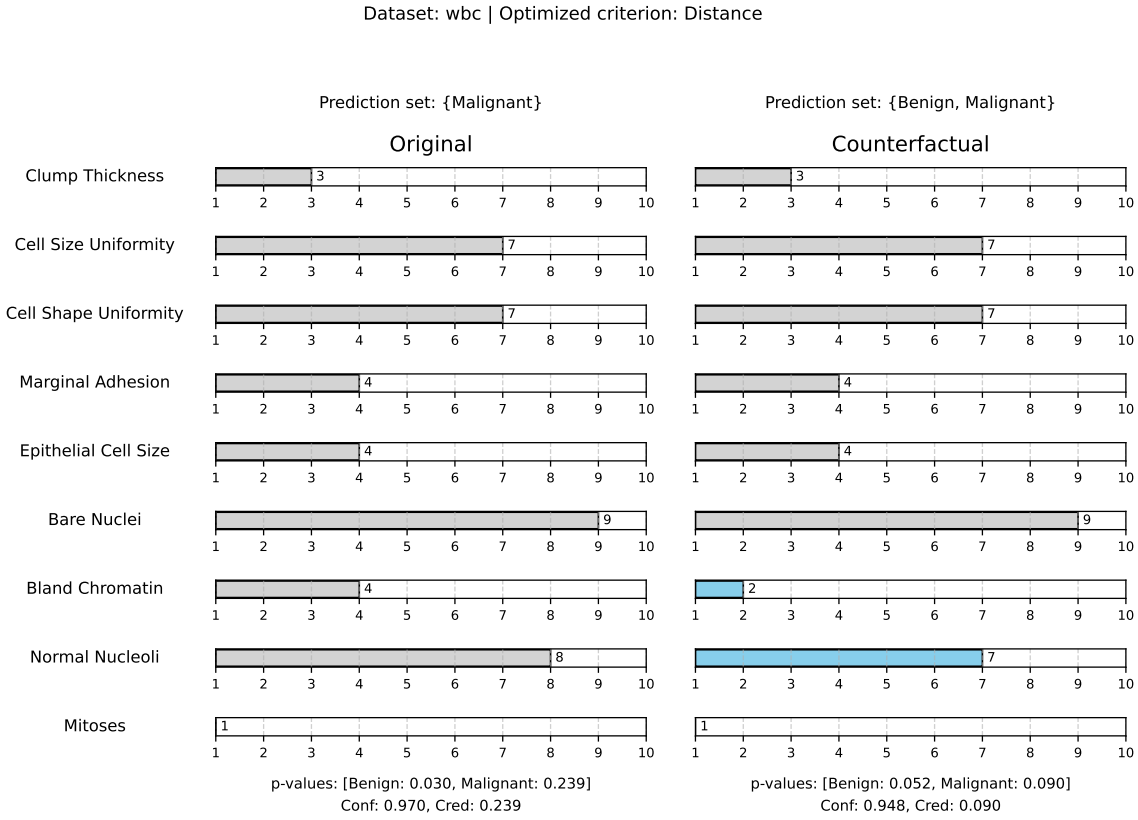

Figure 2: Counterfactual explanation for a conformal prediction set generated by optimizing for *distance* on the WBC dataset. Changed features are shown in blue.

Sparsity-optimized counterfactuals (Fig. 3, 4, and 5) aim to change as few features as possible. In the Diabetes example in Fig. 3, only two feature values are changed: 'DPF' (Diabetes Pedigree Function: quantitative measure of the genetic predisposition to diabetes) reduced from 0.69 to 0.50 and 'Age' is decreased from 30 to 26, though we note that age, while typically treated as a not actionable feature in the literature, was not constrained as such in this proof of concept study. As a result, the class Diabetic is ruled out, resulting in a singleton prediction *{Healthy}*. This counterfactual suggests that a four-year younger patient with a lower genetic predisposition to diabetes (as measured by DPF) would be confidently classified as healthy at the 0.05 significance level. Both features are known to influence diabetes risk making this explanation easy to interpret in clinical settings.

In the Income dataset example (Fig. 4), the sparsity-optimized counterfactual modifies only two features: increasing 'age' from 33 to 41 and changing 'occupation' from Sales to Craft-repair, which resulted in the inclusion of the '>50K' class. This explanation is plausible in context: a higher age may suggest greater work experience, while shifting to a craft-related occupation may suggest a transition to a more skilled or better paying role, both potentially contributing to a prediction of a higher income.

In Fig. 5, showing an example from the Iris dataset, the only feature value changed is 'Petal Length', which increases from 4.50 to 5.10. Since Virginica is a species characterized by longer petals, this change explains its inclusion in the prediction set.

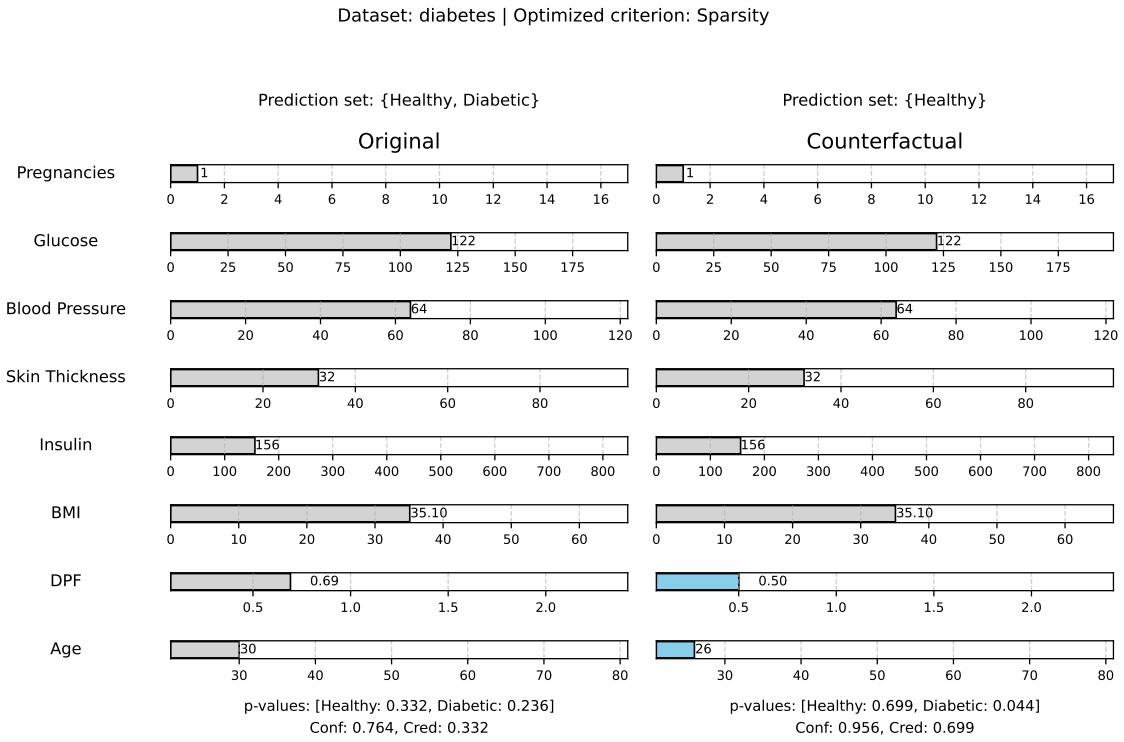

Figure 3: Counterfactual explanation for a conformal prediction set generated by optimizing for *sparsity* on the Diabetes dataset. Changed features are shown in blue.

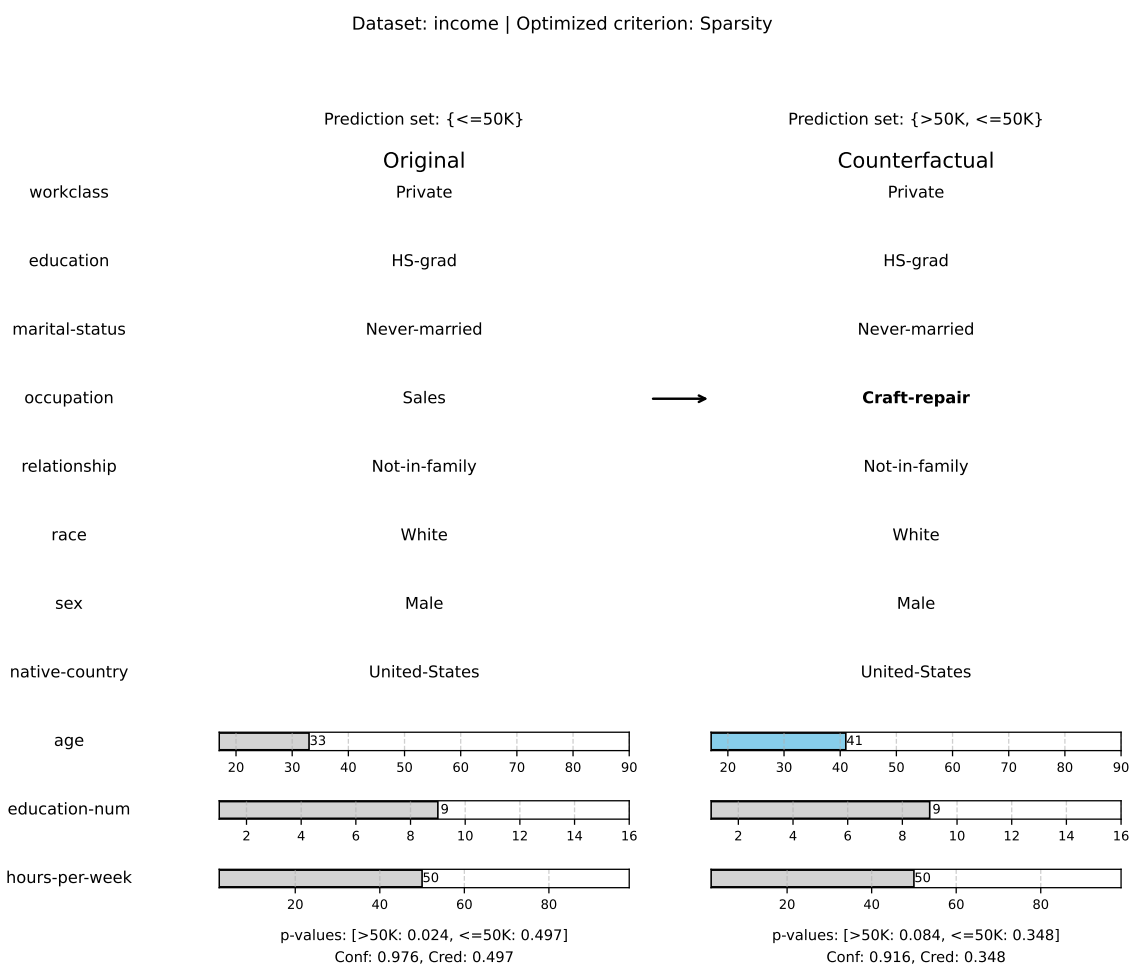

Figure 4: Counterfactual explanation for a conformal prediction set generated by optimizing for *sparsity* on the Income dataset. Changed features are shown in blue or in bold.

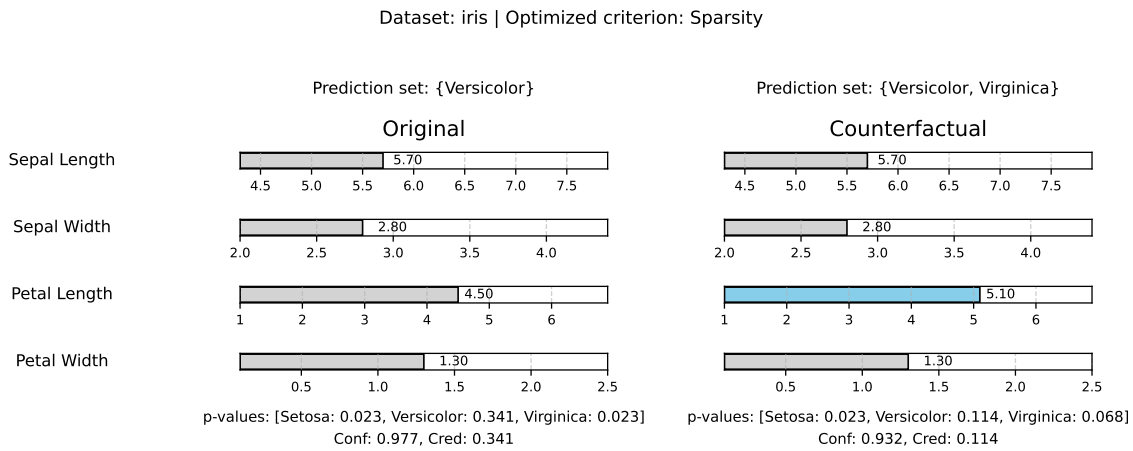

Figure 5: Counterfactual explanation for a conformal prediction set generated by optimizing for *sparsity* on the Iris dataset. Changed features are shown in blue.

Credibility-optimized counterfactuals (6, 7, 8, and 9) aim to produce counterfactuals conforming to the training distribution by maximizing the credibility score. In Fig. 6 (Income dataset), the counterfactual increases weekly working hours and changes the country from Taiwan to the United States, both contributing to a higher likelihood of high income and leading to a prediction set flip from $\{>50K, \leq 50K\}$ to $\{>50K\}$, with credibility rising from 0.321 to 0.615.

In Fig. 7, increases in feature values of known risk factors for diabetes like 'Glucose', 'Insulin', 'DPF', and 'Age' resulted in a plausible explanation for why the counterfactual is confidently classified as Diabetic with 0.987 confidence and 0.834 credibility.

In the example of the Iris dataset in Fig. 8, the original instance yields an empty prediction set due to low conformity across all classes. By suggesting a wider petal and longer sepal and petal values, which are characteristics of Iris Virginica, the counterfactual becomes highly conforming to that class, resulting in a singleton prediction set {Virginica} with credibility equals to 1.

Finally, by switching the vote on the 'physician-fee-free' issue from 'in favor' to 'against' in Fig. 9, the counterfactual shifts the prediction set from including both Republican and Democrat to only Republican with a credibility of 0.662. This change aligns with historical voting patterns, where opposition to fee freeze was typical of republicans.

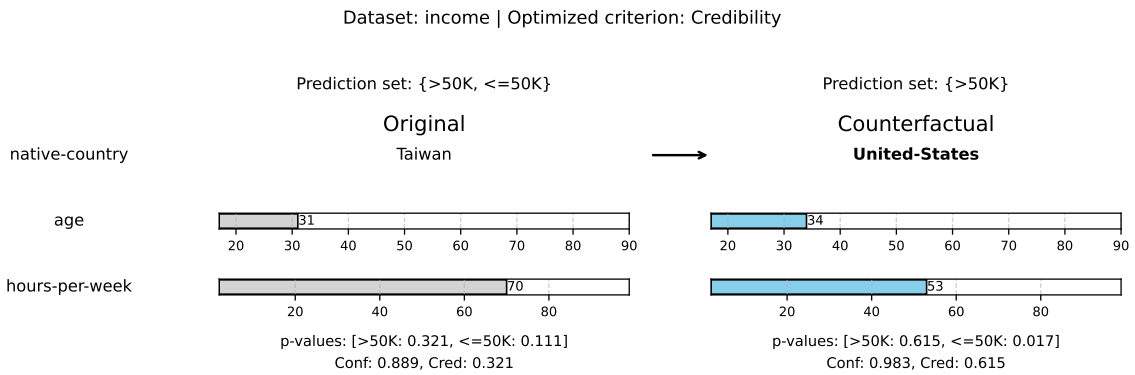

Figure 6: Counterfactual explanation for a conformal prediction set generated by optimizing for *credibility* on the income dataset. Only changed features are shown.

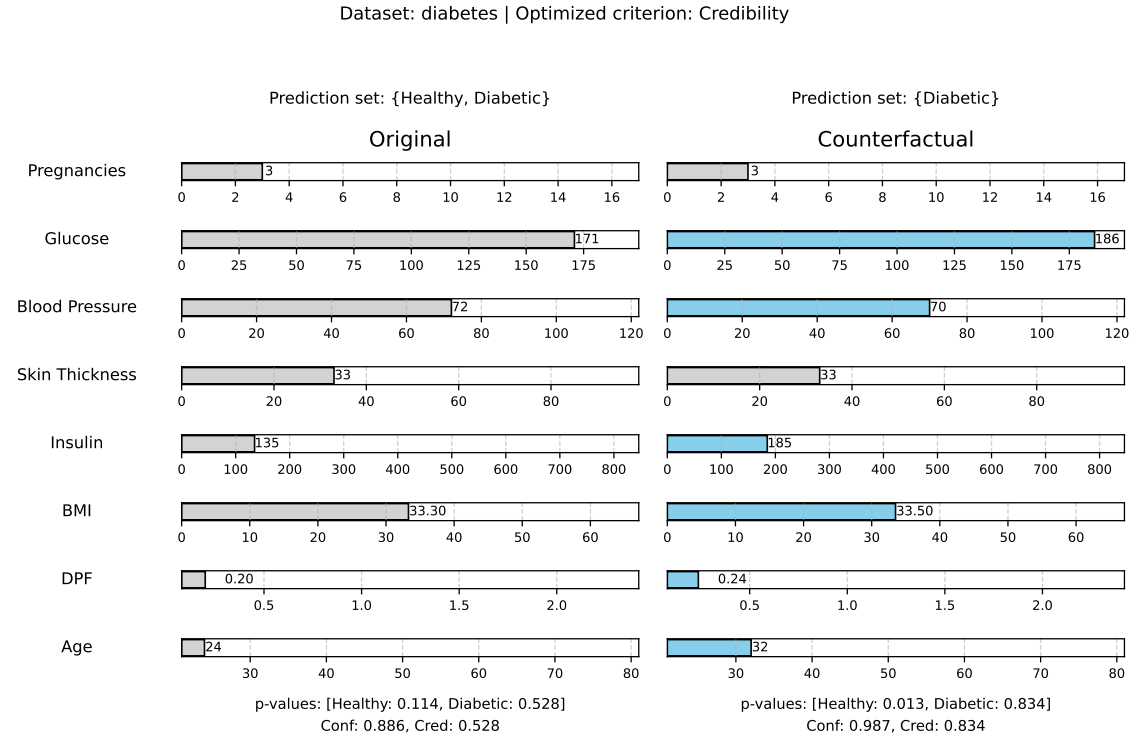

Figure 7: Counterfactual explanation for a conformal prediction set generated by optimizing for *credibility* on the Diabetes dataset. Changed features are shown in blue.

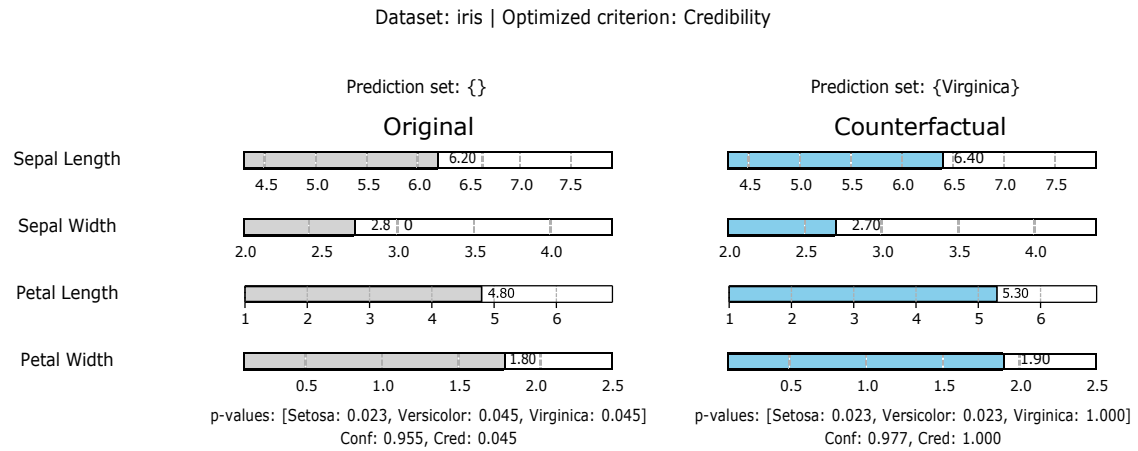

Figure 8: Counterfactual explanation for a prediction set generated by optimizing for *credibility* on the Iris dataset. Changed features are shown in blue.

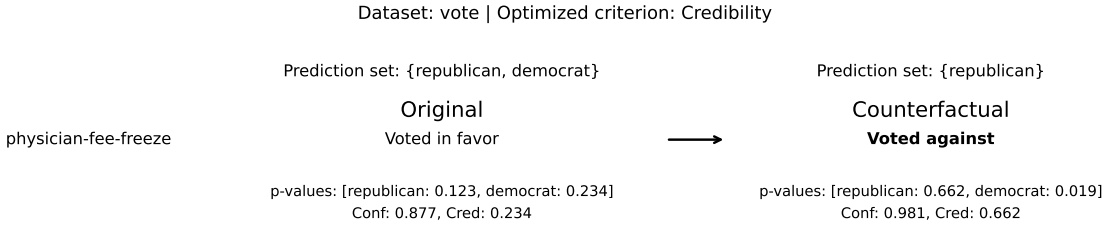

Figure 9: Counterfactual explanation for a conformal prediction set generated by optimizing for *credibility* on the vote dataset. Only changed features are shown.

4.2. Multi-Dataset Evaluation

We conducted a systematic evaluation to compare the effects of different optimization criteria on counterfactual quality across 10 benchmark datasets, i.e., all data sets in Table 1 except Adult income.

For each dataset, we selected 50 random test instances and generated counterfactuals using the previously described criteria: *distance*, *sparsity*, and *credibility*. To assess the quality of surrogate counterfactuals, we measured the following three metrics:

- proximity, computed as the Euclidean distance between the original instance and the counterfactual.
- sparsity, defined as the number of feature values changed relative to the original instance.
- credibility.

Unlike the examples in Section 4.1, which report absolute values for these metrics, the results here are normalized relative to the real counterfactual. Specifically, for each surrogate

counterfactual, we report the ratio of its value to that of the real counterfactual, i.e., it is an indication of the improvement accomplished by using a surrogate counterfactual instead of the real one.

The results are summarized in Table 2, which reports the relative values of each metric (*credibility*, *distance*, and *sparsity*) achieved by counterfactuals optimized under each of the three criteria over all datasets. For *proximity* and *sparsity*, values below 1 indicate improvement, corresponding to a smaller distance or fewer feature changes, respectively. For credibility, which is a quantity to be maximized, values above 1.0 reflect increased conformity with the training data, relative to the real counterfactual.

Table 2: Comparison of optimization criteria across datasets

Datasets	**Credibility**			**Distance**			**Sparsity**		
	Cred_opt	**Dist_opt**	**Spars_opt**	**Cred_opt**	**Dist_opt**	**Spars_opt**	**Cred_opt**	**Dist_opt**	**Spars_opt**
diabetes	1.283	1.133	1.060	0.708	0.299	0.552	0.842	0.460	0.315
german	1.077	1.033	1.021	0.892	0.718	0.794	0.888	0.670	0.667
iris	1.145	0.343	0.478	0.968	0.456	0.628	0.980	0.342	0.317
kc1	1.121	1.005	1.024	0.791	0.513	0.680	0.840	0.621	0.622
kc2	1.149	1.027	1.021	0.681	0.257	0.450	0.840	0.436	0.384
kc3	1.152	1.100	1.052	0.682	0.169	0.429	0.797	0.488	0.352
liver	1.108	1.036	1.010	0.826	0.326	0.560	0.891	0.447	0.330
sonar	1.253	1.036	0.980	0.638	0.257	0.556	0.941	0.918	0.633
vote	1.146	0.988	1.003	0.911	0.772	0.772	0.844	0.632	0.632
wbc	1.141	0.909	0.908	0.951	0.750	0.851	0.928	0.747	0.680
Average	1.157	0.961	0.956	0.805	0.452	0.627	0.879	0.576	0.493

As expected, each optimization criterion achieves the best performance on its corresponding target metric. Counterfactuals optimized for *credibility* (Cred_opt) yield the highest relative *credibility* across datasets, with an average ratio of 1.157, corresponding to a 15.7% improvement compared to the real counterfactuals. This confirms that the method successfully identifies counterfactuals in regions where the conformal predictor assigns higher p-values. While perhaps unsurprising in hindsight, it is still noteworthy that optimizing for credibility also improves *proximity* (0.805) and *sparsity* (0.879) relative to the real counterfactuals. This results from the search space being constrained to values between the original instance and the real counterfactual.

Counterfactuals optimized for *distance* (Dist_opt) achieve the lowest distance, with an average distance ratio of 0.452, indicating a substantial reduction in distance to the original instance compared to the real counterfactuals. Although the method directly optimizes distance, the average *sparsity* ratio remains well below 1.0 (0.576), indicating that these counterfactuals also tend to modify fewer features. Notably, *credibility* remains close to 1.0, suggesting that distance-optimized counterfactuals still largely conform to the calibration distribution, only slightly less so than the real counterfactuals.

Likewise, counterfactuals optimized for *sparsity* (Spars_opt) yield the smallest number of feature changes, with an average sparsity ratio of 0.493, representing a 51% reduction in altered features relative to the real counterfactuals. Proximity also improves, with an average ratio of 0.627, though in some datasets, credibility slightly decreases. This results in an average credibility ratio of 0.956, still relatively close to that of the real instances.

These results are particularly encouraging: even when optimizing for distance or sparsity, the generated counterfactuals exhibit credibility scores that are, on average, only marginally

lower than those of real counterfactuals, despite the latter being actual instances drawn from the dataset.

Overall, the ability to generate counterfactuals that achieve substantial gains in proximity and sparsity while remaining highly conforming demonstrates the strong potential of the proposed method. The fact that these synthetic instances, which are designed to be at least as close to the original input as any real calibration point, still align well with the underlying distribution underscores their plausibility and practical value. This controllability, combined with high conformity, makes the approach especially promising for applications where actionable and realistic explanations are critical.

5. Concluding remarks

This paper introduces a novel approach for explaining conformal prediction sets using counterfactuals. The central idea is to generate surrogate instances that modify the conformal prediction set at a fixed significance level, thereby offering insight into how and why certain labels are included or excluded.

We explored three optimization criteria for generating counterfactual explanations in conformal classification: *distance*, *sparsity*, and *credibility*. *Distance* and *sparsity* are well-established criteria in the literature, aiming to minimize the overall change and the number of altered features, respectively. We proposed using *credibility* as a new way to measure the plausibility of counterfactual explanations. Defined as the highest p-value across candidate labels, credibility guides the search toward regions of the feature space with greater conformity to the data set and the underlying model. The approach is model-agnostic, relying solely on conformal p-values.

We present this study as a proof of concept, leaving several directions open for future work. Notably, we have not explicitly addressed criteria such as actionability and causality, which are important for ensuring that counterfactuals are both feasible and meaningful in real-world contexts. Similarly, we did not explore steering the search toward a specific label set, which could be the actual use case in many scenarios. Moreover, the genetic algorithm used in this work, while straightforward and effective for the present study, limits the applicability of the method to more complex or high-dimensional datasets, highlighting the need for more efficient or scalable alternatives. A promising avenue for future research is to integrate the three optimization criteria, enabling the generation of counterfactuals that balance multiple objectives. Such an approach could improve both the quality and practical relevance of counterfactual explanations across a broader range of applications.

Finally, empirical validation through user studies is essential to assess whether conformal counterfactuals are perceived as informative, trustworthy, and useful by human decision makers. Such studies would help quantify how different optimization criteria align with human preferences and interpretation needs in real-world settings.

Acknowledgements

The authors acknowledge the Swedish Knowledge Foundation, Jönköping University, and the industrial partners for financially supporting the research through the following projects:

AFAIR (grant 20200223), PREMACOP (grant 20220187) and ETIAI (grant 20230036), as part of the research and education environment SPARK at Jönköping University, Sweden.

References

André Artelt and Barbara Hammer. Convex density constraints for computing plausible counterfactual explanations. In *International conference on artificial neural networks*, pages 353–365. Springer, 2020.

Kevin Bache and Moshe Lichman. UCI machine learning repository, 2013.

Henrik Boström. Conformal prediction in python with crepes. In *Proc. of the 13th Symposium on Conformal and Probabilistic Prediction with Applications*, pages 236–249. PMLR, 2024.

Amit Dhurandhar, Karthikeyan Shanmugam, Ronny Luss, and Peder A Olsen. Improving simple models with confidence profiles. *Advances in Neural Information Processing Systems*, 31, 2018.

Riccardo Guidotti. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery*, 38(5):2770–2824, 2024.

Amir-Hossein Karimi, Bernhard Schölkopf, and Isabel Valera. Algorithmic recourse: from counterfactual explanations to interventions. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 353–362, 2021.

Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. *Advances in neural information processing systems*, 29, 2016.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020.

Helena Löfström, Tuwe Löfström, Ulf Johansson, and Cecilia Sönströd. Calibrated explanations: With uncertainty information and counterfactuals. *Expert Systems with Applications*, 246:123154, 2024.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. Interpretable machine learning–a brief history, state-of-the-art and challenges. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 417–431. Springer, 2020.

Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 607–617, 2020.

Harris Papadopoulos. Inductive conformal prediction: Theory and application to neural networks. *Tools in Artificial Intelligence*, 18:315–330, 2008.

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In *Machine Learning: ECML 2002*, pages 345–356. Springer, 2002.

Judea Pearl. Causal inference in statistics: An overview. *Statistics Surveys*, 3:96 – 146, 2009.

Natraj Raman, Daniele Magazzeni, and Sameena Shah. Bayesian hierarchical models for counterfactual estimation. In *International Conference on Artificial Intelligence and Statistics*, pages 1115–1128. PMLR, 2023.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "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*, pages 1135–1144, 2016.

Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. *Advances in neural information processing systems*, 32, 2019.

J. Sayyad Shirabad and T.J. Menzies. The PROMISE Repository of Software Engineering Databases. School of IT and Engineering, Univ. of Ottawa, Canada, 2005.

Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 10–19, 2019.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer, 2005.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. Representer point selection for explaining deep neural networks. *Advances in neural information processing systems*, 31, 2018.