**Computing Rule-Based Explanations by Leveraging Counterfactuals**

Zixuan Geng, Maximilian Schleich, Dan Suciu – Fatima Vahora, Haeun Suh, Prashant Ravic summary

**ABSTRACT**

We introduce a new method to efficiently compute rule-based explanations for automated high-stakes decisions, by leveraging counterfactual explanations, for which many systems are already in place. To validate our approach, we present a Duality Theorem that establishes a relationship between rule-based and counterfactual explanations. Through comprehensive experiments, we demonstrate that our system outperforms or matches the performance of previous systems like MinSetCover and Anchor.

1. **INTROCUTION**

Due to the increasing adoption of machine learning in high-stakes decisions, there is an urgent need for more explainable and debuggable models. As a result, explainable machine learning has become a crucial research topic.

The extensive literature on explanation techniques is well summarized in a book on interpretable machine learning [14]. While there are both local explanations (focusing on individual instances) and global explanations (addressing the model as a whole), this paper emphasizes local explanations.

The Counterfactual Explanation (also known as Actionable Recourse) is a form of local explanation. It suggests modifications to an "undesired" instance to achieve a "desired" outcome. Essentially, it informs users what features must change for a machine learning model to predict a positive outcome from a previously negative one.

Counterfactual explanations may be insufficient for high-stakes machine learning applications due to their potential to mislead by not reflecting all influential features. Rudin et al. [3, 22] advocate for rule-based explanations, which are conjunctions of predicates on features consistently leading to certain outcomes. Unlike prescriptive counterfactual explanations, rule-based explanations descriptively provide core reasons for decisions, making them preferred by financial institutions.

Black-box explanation systems derive explanations by probing the classifier using inputs from specific instances and large datasets, either from training data or historical decisions. Counterfactual explanations answer questions with an existential approach, identifying features that, when altered, lead to a positive outcome. In contrast, rule-based explanations use a universal approach, pointing out features whose current values always result in a negative outcome regardless of other features. Finding counterfactual explanations is easier, with systems like Mace[8], Geco[23], and Dice[15] providing efficient solutions. However, obtaining rule-based explanations is more challenging, often requiring complex solutions such as converting the issue into a minimum set-cover problem.

In the paper, we introduce a novel method for rule-based explanations by leveraging existing counterfactual systems. We demonstrate that counterfactual and rule-based explanations are duals, implying that every rule-based explanation must incorporate at least one feature from its counterfactual counterpart. This duality principle is foundational to our approach.

Using the duality theorem, we've developed a method to compute rule-based explanations by employing counterfactual explanations as a black box. Our base algorithm, GeneticRule, uses a genetic algorithm to find candidate rules for instances with bad outcomes. We propose two enhancements: GeneticRule with Counterfactual (GeneticRuleCF) and Greedy Algorithm with Counterfactual (GreedyRuleCF). GeneticRuleCF incorporates a counterfactual system to refine candidate rules. If a rule isn't globally consistent, it asks for a counterfactual explanation while ensuring features already in the rule remain unchanged. On the other hand, GreedyRuleCF applies the counterfactual approach solely to the top-performing candidate rule.

To validate a rule-based explanation, its global consistency must be checked, a task that's resource-intensive. The set-cover method in [22] conducts this test only on database instances. In contrast, our approach examines every possible combination of attribute values. To manage the vastness of this task, we employ a counterfactual explanation system. Specifically, a rule is considered globally consistent only if no counterfactual exists when keeping specific rule features unchanged.

In our experimental evaluation comparing our three algorithms with MinSetCover [22] and Anchor [21], we found the latter two too often return rules lacking global consistency. Specifically, MinSetCover had a 97.4% inconsistency rate for the Adult dataset, and Anchor produced rules with redundant predicates 87.0% of the time. Our GeneticRuleCF algorithm, on the other hand, always produced globally consistent rules with only 12.4% redundancy, while our GreedyRuleCF algorithm always generated globally consistent rules without any redundant predicates.

An orthogonal approach to explanations involves the creation of interpretable machine learning models. Rule-based models, as described in [10], shouldn't be confused with rule-based explanations. While the former serves as a decision mechanism, the latter provides explanations for decisions typically made by uninterpretable models.

**Contributions.** In summary, in this paper we make the following contributions.

(1) We prove the Duality Theorem between counterfactual and rule-based explanations. Section 3.1.

(2) We show how to use the Duality Theorem to compute rule-based explanations by using a counterfactual-based explanation system. Section 3.2.

(3) We describe three algorithms: GeneticRule, GeneticRuleCF, and GreedyRuleCF for generating rule-based explanations. Section 4.

(4) We conduct an extensive experimental evaluation of GeneticRule, GeneticRuleCF, and GreedyRuleCF algorithms, and compare them with Anchor and MinSetCover. Section 5.

1. **DEFINITIONS**

|  |
| --- |
| * 𝐹1, . . ., 𝐹𝑛 be n features, with domains 𝑑𝑜𝑚(𝐹1), . . ., 𝑑𝑜𝑚(𝐹𝑛) [Ordered] * Define Inst = 𝑑𝑜𝑚(𝐹1) × · · · × 𝑑𝑜𝑚(𝐹𝑛) * Let an element 𝑥 ∈ Inst an instance. * Let C ais a black box classifier. * any instance 𝑥 ∈ Inst, returns a prediction 𝐶(𝑥) within range [0, 1] * If C(x) <= 0.5, it's classified as “undesired” or “bad”, [Binary classifier: 0] * If C(x) > 0.5, it's classified as “desired” or “good”, [Binary classifier: 1] * Let a database, D, consisting of m instances: D = {x1, …, xm}. * For every instance xi in D, its feature values are given by xi = (fi1, …, fin). |

* 1. Rule-based Explanation

For a given instance xi = (fi1, …, fin) ∈ 𝐷, a rule component, denoted as RC(x), is formulated by either 𝑓𝑗 ≤ 𝑓𝑖𝑗 or 𝑓𝑗 ≥ 𝑓𝑖𝑗. A rule relevant to 𝑥𝑖 is a set of rule components, represented as 𝑅 = {𝑅𝐶1, . . ., - 𝑅𝐶𝑐} where R(x) = RC1 X … X -RCC. To indicate that a feature value 𝐹𝑗 = 𝑓𝑖𝑗, it is essential to incorporate both ≤ and ≥ rule components, meaning the cardinality constraint is 0 ≤ 𝑐 ≤ 2𝑛.

Instances with the "undesired" label have C(xi) <= 0.5. As suggested by Rudin and Shaposhnik [22], for a rule R to elucidate the undesired outcome of xi, it should satisfy:

1. Relevance: Ensure xi ∈ INSTR

\* The paper inherently satisfies this requirement as it only considers rules relevant to instance xi

1. Global Consistency: ∀𝑥 ∈ Inst𝑅, 𝐶(𝑥) ≤ 0.5.
2. Interpretability: The rule must be simple, favoring minimal cardinality.

The paper identifies a trivial rule, Rtriv, that guarantees global consistency. However, its cardinality, being 2n, compromises its interpretability. For computational feasibility, the study emphasizes the need to derive a minimal set of rule components that maintain global consistency. Recognizing the computational demands of exhaustive global consistency checks, the paper advocates "Data Consistency". This approach evaluates consistency relative to a predefined database D = {x1, …, xm}, ensuring ∀𝑥𝑘 ∈ 𝐷 ∩ Inst𝑅, 𝐶 (𝑥𝑘) ≤ 0.5.

It is noted that certain methods, such as MinSetCover and Anchor, might not always uphold global consistency.

* 1. Counterfactual Explanation
  2. Discussion

1. DUALITY
   1. The Duality Theorem
   2. Using the Duality Theorem
2. ALGORITHMS
   1. GeneticRule
   2. GeneticRuleCF
   3. GreedyRuleCF
   4. Fitness Score Function
3. EXPERIMENTS
   1. Experiment Setup
   2. Quality in terms of Consistency and Interpretability
   3. Runtime Comparison
   4. Microbenchmarks
4. LIMITATIONS AND FUTURE WORK
5. CONCLUSION