Journal of Machine Learning Research 24 (2023) 1-44 Submitted 5/21; Revised 6/22; Published 1/23

Globally-Consistent Rule-Based Summary-Explanations for Machine Learning Models: Application to Credit-Risk Evaluation

Cynthia Rudin CYNTHIA@CS.DUKE.EDU
Departments of Computer Science and Electrical & Computer Engineering
Duke University, Durham NC, USA

Yaron Shaposhnik YARON.SHAPOSHNIK@SIMON.ROCHESTER.EDU
Simon Business School
University of Rochester, Rochester NY, USA

Editor: David Jensen

Abstract

We develop a method for understanding specific predictions made by (global) predictive models by constructing (local) models tailored to each specific observation (these are also called "explanations" in the literature). Unlike existing work that "explains" specific observations by *approximating* global models in the vicinity of these observations, we fit models that are *globally-consistent* with predictions made by the global model on past data. We focus on rule-based models (also known as association rules or conjunctions of predicates), which are interpretable and widely used in practice. We design multiple algorithms to extract such rules from discrete and continuous datasets, and study their theoretical properties. Finally, we apply these algorithms to multiple credit-risk models trained on the Explainable Machine Learning Challenge data from FICO and demonstrate that our approach effectively produces sparse summary-explanations of these models in seconds. Our approach is model-agnostic (that is, can be used to explain any predictive model), and solves a minimum set cover problem to construct its summaries.

Keywords: Explainable Artificial Intelligence (XAI), Local Explanations, Interpretability, Credit Risk

1. Introduction

As the use of predictive models for high-stakes or other important decisions in society is on the rise, it has become apparent that flaws in these models, or even a flawed understanding of these models, can cause (and has caused) catastrophic harm. In the justice system, mistakes in data entry within predictive models have caused people to be denied parole (Citron 2016, Wexler 2017), or to be released when they are actually dangerous, leading to increased crime (Ho 2017). Proprietary models for air quality in California in 2018 indicated that dangerous levels of ash in the air due to wildfires were actually safe (Mannshardt and Naess 2018). Proprietary credit risk models routinely deny or grant loans, leading to questions of fairness and transparency. This has led to a subfield of machine learning (ML) called "explainable" machine learning, where the goal is to produce accurate predictions

that can be intuitively understood by relevant stakeholders, for example, using a simpler "explanation" of a complex models' prediction.

However, mistakes in applying statistical models to high-stakes decisions will not vanish so easily; placing "explanations" on a complex model has dangers that are almost worse than using the complex models alone. The most serious mistake in applying explanations is arguably that explanations are generally not consistent with the underlying model they are trying to explain. For instance, imagine a person being denied credit by a model, receiving an explanation such as "credit history not greater than 10 years." However, a different person could have a credit history less than 10 years and yet could be granted credit. This is a case where the explanation is not globally-consistent with the underlying model. In some cases, the explanation could actually produce the opposite prediction as the global model, which means it is not trustworthy. In these cases, the explanation approximates the global model, but does not actually explain it. In that sense, even the word "explanation" here is misleading: it does not actually explain the global model, but instead, approximates it, and does so in a way we cannot necessarily trust.

There are modeling choices we can make in order to avoid mistakes like those listed above. First, we can work with interpretable global and local models, rather than proprietary models or complicated black-box models. Second, we can ensure that our explanations are *globally-consistent*, which means that if a local explanation of the global model is provided for a part of the state space, it must agree with the global model for *all* past observations that it claims to apply to. While there has been substantial work on local explanations that do not need to be consistent, there has been (as far as we know) no prior work on globally-consistent explanations.

Our approach. In this work, we present a method to create very sparse *summary-explanations* that are globally-consistent with the global model, for all observations that such a summary applies to. (Here we have changed the misleading term "explanation" to "summary-explanation" so as to avoid the wrong implications that the summary actually explains the global model.) The summary-explanation only applies to a local part of the search space that is designated. For example, a summary-explanation might state that "all 500 individuals who have credit history less than 5 years were predicted to default on a loan." In that case, all individuals who have less than 5 years of credit history would actually be predicted to default by the global model. This summary would be true whether or not the global model uses the length of the credit history as a variable. The summary-explanation would only apply to the local space of observations that have less than 5 years of credit history, and not beyond that. Because the summary-explanation applies only to a part of the search space, it can be optimized to be simultaneously sparse (in its description) and correct. Figure 1 presents examples of summary-explanations generated by one of our algorithms for predictions made by various machine learning models. We used the dataset for the Explainable Machine Learning Challenge (FICO 2018) for these examples.

Graphical illustration. Figure 2 illustrates the overarching process for utilizing globally consistent rules. First, a dataset is collected (Figure 2a) for constructing a global model (Figure 2b). The model is applied for making predictions (Figure 2c), either on the same data or on a new dataset. A user of the model is interested in a sparse characterization of a specific prediction (pointed by the arrow in Figure 2c), perhaps as part of an interaction

For *all* 1342 people where:
- ExternalRiskEstimate$\geq$ 80,
- MSinceOldestTradeOpen$\geq$ 179, and
- NumSatisfactoryTrades$\geq$ 15

the global model predicts a **low** risk of default.

For *all* 462 people where:
- AverageMInFile$<$ 52,
- ExternalRiskEstimate$<$ 66, and
- PercentTradesNeverDelq$<$ 93

the global model predicts a **high** risk of default.

For *all* 272 people where:
- PercentInstallTrades$\geq$ 55 and
- AverageMInFile$<$ 42

the global model predicts a **high** risk of default.

For *all* 936 people where:
- ExternalRiskEstimate$<$ 61 and
- NetFractionRevolvingBurden$\geq$ 54

the global model predicts a **high** risk of default.

For *all* 199 people where:
- ExternalRiskEstimate$\geq$ 84 and
- NetFractionRevolvingBurden$<$ 50

the global model predicts a **low** risk of default.

For *all* 105 people where:
- NumInqLast6M$\geq$ 7 and
- AverageMInFile$<$ 54

the global model predicts a **high** risk of default.

For *all* 1299 people where:
- NumBank2NatlTradesWHighUtilization$\geq$ 1,
- AverageMInFile$<$ 76,
- ExternalRiskEstimate$<$ 73, and
- NumSatisfactoryTrades$<$ 18

the global model predicts a **high** risk of default.

For *all* 177 people where:
- ExternalRiskEstimate$<$ 54

the global model predicts a **high** risk of default.

Figure 1: Examples of globally-consistent summary-explanations for the FICO dataset generated using `ContMaxSupport` for different observations and models (from top to bottom and left to right, the explanations pertain to the global models: Logistic regression, Random forest, Logistic regression, SVM with RBF kernel, SVM with polynomial kernel, Adaboost, and KNN). Description of the features is provided in Appendix B. Some of the more important ones include "AverageMInFile" (average months in file) and "ExternalRiskEstimate" (which is a consolidated set of risk markers).

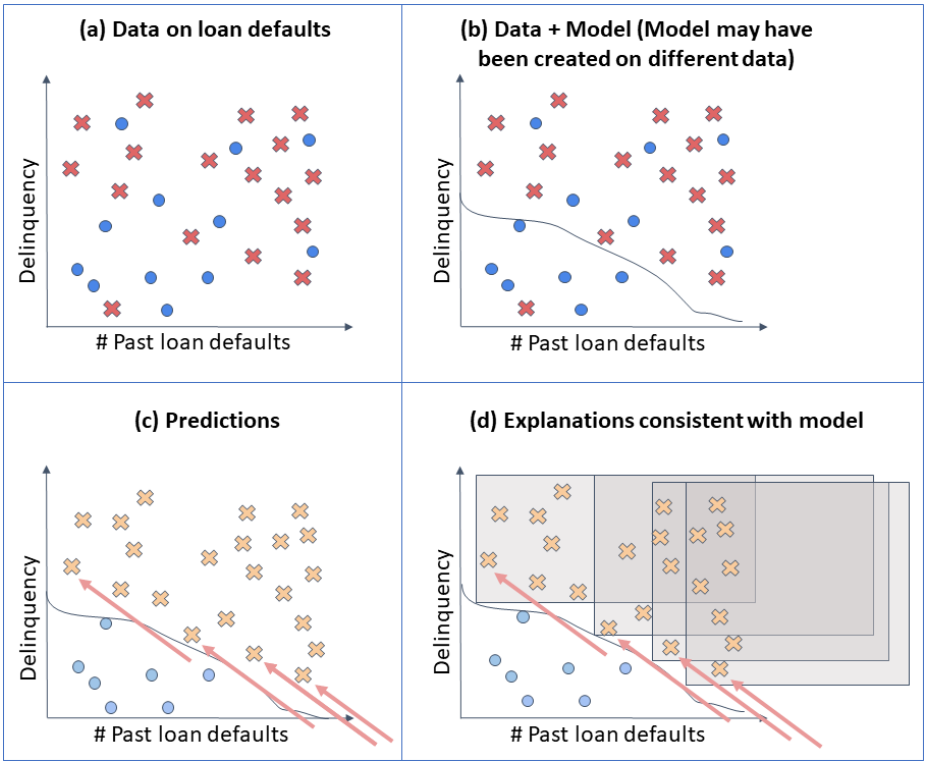

Figure 2: A graphical illustration of globally-consistent rules. The red arrows point to specific predictions that the user of the model (e.g., loan officer) wishes to find a summary-explanation for (e.g., find a rule to explain that the loan application pointed at by a red arrow was denied, along with all other customers like that one). The gray boxes are our rules that provide a sparse summary of conditions under which the loan was denied, including the current observation.

with a customer (e.g., a loan applicant asks "Was everyone like me also denied a loan?"). The user of the model can then generate rules (Figure 2d) that agree with the specific prediction (e.g., the loan officer could reply "Of the 11 people who also have more than 10 past loan defaults and more than 8 month delinquency, all were predicted to default and thus denied a loan."). The user and the customer can both be certain that the rule applies to all other customers in the data and no exceptions were made. To our knowledge, this is the first work that proposes using this form of explanation, formulates the respective problem, and offers algorithmic solutions to the problem.

Contributions. This paper makes the following contributions. First, we develop a new method for understanding predictions made by arbitrary ML models, which we refer to as summary-explanations. Our summary-explanations are sparse rules (i.e., rules defined using a small number of features) that locally summarize, and are consistent with predictions from, a globally-interpretable model, which is how we envision them being used in practice. Our summary-explanations are designed to optimize both sparsity and coverage (i.e.,

support) of the rules, while constraining them to be fully consistent with the global model. Each rule is conditioned on a given observation and on its predicted class; that is, each rule is customize-designed to describe the local space around a given observation. Second, we show that the optimization problem for designing these rules is a generalization of the well-known minimum set cover problem. We present algorithms that solve this problem to find optimal sparse rules. Third, we apply the method in the context of credit risk assessment, which is one of the domains where interpretability of predictions is essential; these models make decisions that critically affect people's lives. A recent explainable machine learning effort by the Fair Isaac Corporation (FICO) challenged researchers to construct explanations for models of credit risk (FICO 2018). They specifically requested rule-based explanations, of the same form as the summary-explanations we provide here (though they did not require that the explanations be consistent with the global model). Our numerical experiments suggest that the resulting summary-explanations can be produced in seconds, which makes them suitable for practical use. In fact, a competition entry based on the algorithms discussed in this work was recognized by the senior executives of the FICO AI team.[1] Finally, we developed a simple programming interface in Python for applying our algorithms, which is publicly available at this link (see Appendix C for more information).

Related work. Our work is most closely related to the literature on local explanations for machine learning models, which aims to explain specific predictions made by (potentially black-box) global models. We therefore focus on this research area and refer the reader to survey papers on interpretable and explainable machine learning to other bodies of work (Rudin et al. 2022, Došilović et al. 2018, Guidotti et al. 2018b, Arya et al. 2020, Tjoa and Guan 2020, Roscher et al. 2020, Burkart and Huber 2021).

A class of algorithms (for instance, the popular LIME algorithm proposed by Ribeiro et al. 2016) randomly perturb the explained observation and use the perturbations to train a model which locally approximates the global model. These explanations, however, are not necessarily accurate, nor globally-consistent, and their important features are not necessarily those of the global model; LIME has been heavily criticized for its lack of fidelity in particular (Slack et al. 2020). Many papers since tried to improve LIME in various ways. For example, Zafar and Khan (2021) proposes DLIME, a deterministic variant of LIME, aimed to address the instability in the resulting explanations due to the random perturbation. Shankaranarayana and Runje (2019) developed ALIME, an alternative approach to address the same issue using an autoenconder which generates synthetic data that is later used to generate local models. Yoon et al. (2019) suggest to replace the sampling-based approach of LIME and use data instead whereby observations are weighted according to a weight function that is learned using reinforcement learning. Another prominent local method is SHAP (Lundberg and Lee 2017) which applies concepts from game theory to derive the linear coefficients of a local model. It is shown to be the only method that satisfies several properties, none of which is global consistency. Like LIME, SHAP requires access to the global model and perturbs the explained observation to compute the local model.

While many of the above papers make use of *linear* local models, there is a line of work that employs *rule-based* models, similar to the ones we use in this work. In Ribeiro

1. The entry won the "FICO recognition award" in the "Explainable Machine Learning Challenge" (FICO 2019).

et al. (2018), the authors propose an approach called Anchors, whereby they create local rule-based models which guarantee a minimal degree of accuracy of the rules around the explained observations. Anchors requires rules to have sufficiently large precision and coverage, which bare similarities with the notions of consistency and support that are relevant to our approach (and which we formally define in later sections). A key conceptional difference from our approach is that in Anchors, the metrics are estimated *locally* whereas we seek for a resolution that applies to the entire feature space. Moreover our approach is not required to use sampling and does not generally require access to the global model beyond the labeling of the dataset and the explained observation. This means that we do not need to make any assumptions about the (local) sampling distribution. This lends itself to different algorithmic ideas than Ribeiro et al. (2018), who use a greedy algorithm based on a multi-armed bandit formulation, while we use linear integer programs and dynamic programming (as there is no need for exploration in our problem since we do not sample new data points). This allows us to directly optimize the sparsity (interpretability) of the resulting rules, which cannot be controlled for under Anchors (and other methods such as LIME), leaving them with less interpretable explanations. Another method for generating local rules is LORE (Guidotti et al. 2018a), which uses a genetic algorithm to generate samples near the explained observation. It then uses the samples to train a classification tree from which a local model is derived. They further use this tree to derive counterfactual explanations in the form of minimal changes to the observation that result in an opposite prediction (for additional examples of counterfactual explanations see also work of Dhurandhar et al. 2019, Van Looveren and Klaise 2021, White and Garcez 2019, Kanamori et al. 2020). However, there are no guarantees of any sort on the fidelity of the local rules or the counterfactual explanations. Note that in contrast to the above, our methods require access to a dataset (the training set or a separate dataset) based on which summary-explanations are generated and to which summary statistics are reported (we discuss the privacy-related implications of this in Section 5).

All of the above works suffer from the problem we discussed, namely that the local explanations are not globally consistent with the model. In fact, Ribeiro et al. (2016) acknowledge the possibility of conflicting rules arising from their methodology, which would never happen in our globally consistent setting. We find conflicting explanations to be confusing, and ultimately uninterpretable, undermining the point of providing explanations in the first place. This lack of *fidelity* of a rule can be problematic; a loan applicant confronting a rule that is even occasionally (say 5%) broken can ask "If 5% of customers that are like me are granted a loan, why can't I also be granted a loan?". In that case, despite the sparsity of the rule, its lack of consistency with respect to the underlying model leads it to be confusing and uninterpretable. If there exists an inconsistent rule, it means there exist two data points on which a rule that is valid for one of them would be incorrect for the other. (E.g., "those with less than 3 loan defaults in the last year are generally granted a loan" would be a valid explanation for someone granted a loan, but the same rule does not apply to someone with 2 defaults who is denied such a loan, even though the rule applies to them.)

The only example in the literature that does not seem to suffer the same problems as the above works is that of Ignatiev et al. (2019), done in parallel to our work (see Rudin and Shaposhnik 2019, for an earlier version of this manuscript). They approach the

problem of finding globally-consistent rules from a logic-based perspective, assuming that their global machine learning model can be described by a set of constraints and logical conditions, and that the model is known (which we do not). They apply an iterative search procedure that aims to identify the smallest subset of features, which added as constraints to the global ML model, would guarantee consistency (that is, that the prediction of the global model coincides with the explained observation). To this end, they solve constraint satisfaction problems and use a mixed linear integer program to check the satisfiability of the rule with respect to the global model (but not to construct the rule in the first place, like we do). In contrast, we approach the problem from a data-centered (rather than model-centered) perspective, using labeled data to generate rules, and assume no knowledge about the underlying model, which makes our approach truly model agnostic. This also leads us to adopting different algorithmic ideas using dynamic programming and taking advantage of modern software tools for solving mixed integer linear programs to directly extract consistent rules. We briefly note the follow up work to ours by Izza et al. (2020), Marques-Silva et al. (2020) who develop similar solutions to Ignatiev et al. (2019) in the context of other ML models (decision trees and linear classifiers). In short, our approach is completely different, but these works (like us) recognize the value of global consistency with respect to the underlying global model.

Finally, we note the recent work of Geng et al. (2022) and Kanamori et al. (2022) which build on our work and apply the ideas of global-consistency in the context of counterfactual explanations using sampling of new data points and querying of the global model.

Organization. In Section 2, we formally define globally-consistent rules and formulate the optimization problem of identifying these rules in datasets. In Section 3, we develop algorithms for solving the optimization problem, and in Section 4, we apply these algorithms to predictions about credit risk. We discuss relevant practical aspects of globally-consistent rules in Section 5 and conclude in Section 6.

2. Problem Formulation

Consider a general binary classification problem defined over a data-matrix $\mathbf{X} \in \mathbb{R}^{|N|\times|P|}$. We use N and P to respectively denote the set of observations and features. In addition, we assume access to $\{h^{\text{global}}(\mathbf{x}_i)\}_{i=1}^{N}$, the labels predicted by a global (potentially black-box or potentially somewhat complex globally-interpretable) model h^{global}; that is, for each observation $i \in N$ we have $h^{\text{global}}(\mathbf{x}_i)$. We do not make any assumptions about the nature of h^{global}, nor of having access to h^{global} for making predictions on arbitrary observations. With a slight abuse of notation, we use both $i \in N$ and $\mathbf{x}_i$ to denote an observation. We denote the values of a feature p, that is a column of $\mathbf{X}$, as $\mathbf{X}_{:,p}$. Note that we do not make assumptions about nor require access to the labels of the original data, that were presumably used to train the global model h^{global}. Our technique can thus be used even when we have only an observational dataset containing predictions from a black box model. Our goal is to simply provide insights that improve our understanding of how the given model h^{global} makes predictions.

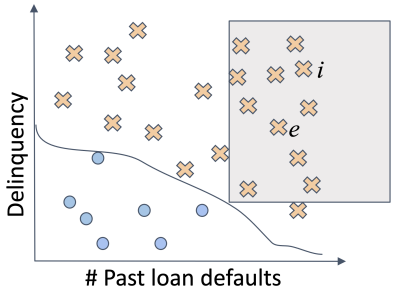

Figure 3: Illustration of a globally-consistent local summary-explanation $h^{\text{expln}(e)}$ for observation e. The curve represents the decision boundary of a global model h^{global} that respectively predicts "×" above the curve and "○" below the curve. In the shaded area, the summary-explanation $h^{\text{expln}(e)}$ predicts "×" and in other parts of the feature space it abstains. The summary-explanation satisfies Properties 1 and 2 of Definition 1: relevance (it predicts "×" for e, which is what the global model predicts) and global consistency (for every observation where $h^{\text{expln}(e)}$ predicts "×", the global model h^{global} also predicts "×"). Observe that the labels "×" and "○" are determined by the global model h^{global}, which are presented in the figure (rather than the labels of the actual data which are irrelevant for our purposes). The point i labeled in the figure contributes to the relevancy of e's summary-explanation.

Definition 1 *Let $\boldsymbol{x}_e$ denote a new observation and its label predicted by the global model is $h^{\text{global}}(\boldsymbol{x}_e)$. We say that the model $h^{\text{expln}(e)}$ provides a globally-consistent local summary-explanation to observation e with respect to the model h^{global} and the data-matrix $\boldsymbol{X}$ if the following conditions are met:*

1. *(Relevance) $h^{\text{expln}(e)}(\boldsymbol{x}_e) = h^{\text{global}}(\boldsymbol{x}_e)$.* That is, e's summary-explanation must agree with e's prediction from the global model at the point e itself*;*

2. *(Global consistency) For every observation $i \in N$, if $h^{\text{expln}(e)}(\boldsymbol{x}_i) = h^{\text{global}}(\boldsymbol{x}_e)$ then $h^{\text{global}}(\boldsymbol{x}_i) = h^{\text{expln}(e)}(\boldsymbol{x}_i) = h^{\text{global}}(\boldsymbol{x}_e) = h^{\text{expln}(e)}(\boldsymbol{x}_e)$.* That is, if point e's summary-explanation covers point i, then point i must have the same prediction as e according to both the global model and e's summary-explanation.
   Equivalently, if i has the opposite global prediction as e, then e's explanation must disagree with the global model's prediction of point i. That is, if $h^{\text{global}}(\mathbf{x}_i) \neq h^{\text{global}}(\mathbf{x}_e)$ then $h^{\text{expln}(e)}(\mathbf{x}_i) \neq h^{\text{global}}(\mathbf{x}_e)$.

3. *(Interpretability) the model $h^{\text{expln}(e)}$ belongs to a class of interpretable models $\mathcal{H}$.*

Throughout the work we use the terms *local-models*, *local-rules*, and simply *rules* synonymously to refer to summary-explanations.

Figure 3 illustrates properties 1 and 2 of Definition 1. Condition 2 implies that $h^{\text{expln}(e)}$ is consistent with past predictions. If i participate in e's summary explanation, it must

also have the same global model label as point e. There are two important aspects to this condition:

- The summary-explanation $h^{\text{expln}(e)}$ pertains *only* to a subset of data local to point e. For instance, if the global model's prediction for e is $h^{\text{global}}(\mathbf{x}_e) = 1$, then the summary-explanation pertains only to points where its value $h^{\text{expln}(e)}$ is 1. The summary-explanation says nothing about (abstains on) points where $h^{\text{expln}(e)}$ is 0. $h^{\text{expln}(e)}$ must be consistent with points i where $h^{\text{expln}(e)}$=1, but says nothing about other points and does not need to be consistent with them.

- $h^{\text{expln}(e)}$ must be consistent with *all* points i where $h^{\text{expln}(e)}$=1. This is in sharp contrast to alternative approaches which generate explanations by approximating h^{global} in the vicinity of $\mathbf{x}_e$, which could result in providing explanations that can be contradicted using observations from the dataset and their predictions by the global model. As discussed above, the existence of such contradictory explanations could negatively impact the trust that customers or users have in the system and its underlying model, as well as in the interpretability of the rules. In contrast, by design, our summary-explanations cannot be contradicted using past data.

Condition 3 (interpretability) favors sparse and reliable summary-explanations. (In practice users could choose other metrics for interpretability as well.) We use two measures to assess the quality of a summary-explanation $h^{\text{expln}(e)}$:

- $\Gamma_c(h^{\text{expln}(e)})$: Complexity—a measure for (the inverse of) interpretability based on the class $\mathcal{H}$. For example, the complexity of a model could be the number of non-zero coefficients in linear models, or the number of leaves in decision trees. Sparse models (within the same hypotheses class) would typically be considered less complex.

- $\Gamma_s(h^{\text{expln}(e)})$: Support—the number of observations in the dataset that satisfy the rule, that is,

$$\Gamma_s(h^{\text{expln}(e)}) = |\{i \in N : h^{\text{expln}(e)}(\mathbf{x}_i) = h^{\text{global}}(\mathbf{x}_e)\}|.$$

  Intuitively, the support measures the coverage of a summary-explanation. Larger support values indicate that the rule applies to a greater number of past observations, which increases trust and confidence in the explanation, and helps to ensure statistical generalization and reduce overfitting of the summary-explanations.

Ideal rules have low complexity and large support.

Using these conditions, we can formulate the problem for finding a globally-consistent summary-explanation for the point $(\mathbf{x}_e, h^{\text{global}}(\mathbf{x}_e))$. Here we recall that $h^{\text{global}}(\mathbf{x}_i)$ is given for each i, and our goal is to construct $h^{\text{expln}(e)}(\mathbf{x})$, which is the summary-explanation for

point e:

$$\begin{array}{lll} \max_{h^{\text{expln}(e)} \in \mathcal{H}} & w_s \cdot \Gamma_s(h^{\text{expln}(e)}) - w_c \cdot \Gamma_c(h^{\text{expln}(e)}) & \text{(maximize support, minimize complexity)} \\ \text{s.t.} & h^{\text{expln}(e)}(\mathbf{x}_e) = h^{\text{global}}(\mathbf{x}_e) & \text{(summary-explanation is relevant)} \\ \forall i \in N: & \text{If } h^{\text{global}}(\mathbf{x}_i) = 1 - h^{\text{global}}(\mathbf{x}_e) & \text{(summary-explanation is consistent)} \\ & \text{Then } h^{\text{expln}(e)}(\mathbf{x}_i) = 1 - h^{\text{global}}(\mathbf{x}_e) & \\ & \Gamma_c(h^{\text{expln}(e)}) \leq p_c. & \text{(interpretability)} \end{array} \quad . \tag{1}$$

The coefficients w_s and w_c are user-defined non-negative weights that balance the desired support and sparsity of the resulting rule, while the bound p_c ensures sufficient sparsity. The three types of constraints capture the three conditions of globally-consistent local summary-explanations: relevance, global consistency, and interpretability, respectively. Note that the antecedent in the second constraint (consistency) is independent of the decision variables and therefore it can be written as a linear constraint applied only to a subset of the observations. This constraint says that any point i that is classified differently than our point e must not be part of the explanation (which is equivalent to stating that if the explanation applies to a point i, then the global model's prediction at point i must be identical to the prediction at point e).

To provide more detail on the consistency condition, if we provide a summary-explanation for a point $\mathbf{x}_e$ where $h^{\text{global}}(\mathbf{x}_e) = 1$, then every point $\mathbf{x}_i$ where $h^{\text{global}}(\mathbf{x}_i) = 0$ cannot be part of the explanation, and must have $h^{\text{expln}(e)}(\mathbf{x}_i) = 0$. Symmetrically, if we provide a summary-explanation for a point $\mathbf{x}_e$ where $h^{\text{global}}(\mathbf{x}_e) = 0$, then every point $\mathbf{x}_i$ where $h^{\text{global}}(\mathbf{x}_i) = 1$ cannot be part of the explanation, and must have $h^{\text{expln}(e)}(\mathbf{x}_i) = 1$.

In the remainder of this paper, we focus on a specific class of interpretable models $\mathcal{H}$ that are *rule-based*. We say that a model h is a rule-based classifier predicting $a \in \{0, 1\}$ if it can be written as a conjunction of one-dimensional step functions with a single step:

$$h_{R,\boldsymbol{\tau},a}(\mathbf{x}) = \begin{cases} a & \text{if } x_p \geq \tau_p, \forall p \in R \text{ where } R \subseteq P \\ 1 - a & \text{otherwise} \end{cases} \quad . \tag{2}$$

This defines a hyperbox in the feature space. Note that any rule-based model can provide a globally-consistent summary-explanation for an observation $\mathbf{x}_e$ if it satisfies the conditions of Definition 1. Moreover, the canonical form of the rule-based model above can capture more general rules:

1. Strict inequalities can be expressed using inequalities since the dataset values are finite (e.g., by increasing the value of τ_p by a small value ϵ).

2. Opposite inequalities of the form $x_p \leq \tau_p$ can be expressed using the above representation of rules by expanding the feature space to include features with opposite signs. That is, we can double the size of the data matrix to include -$\mathbf{X}$ in addition to $\mathbf{X}$, in which case a predicate of the form $x_p \leq \tau_p$ would be represented using the new feature $-\mathbf{X}_{:,p}$ as $-x_p \geq -\tau_p$. Note that for a binary dataset $\mathbf{X}$, one could simply append the features $\mathbf{1}_{|N|\times|P|} - \mathbf{X}$ to achieve the same result while keeping the domain of the feature space binary (we use $\mathbf{1}_{|N|\times|P|}$ to denote a matrix of dimensions $|N| \times |P|$ whose entries are all equal to 1).

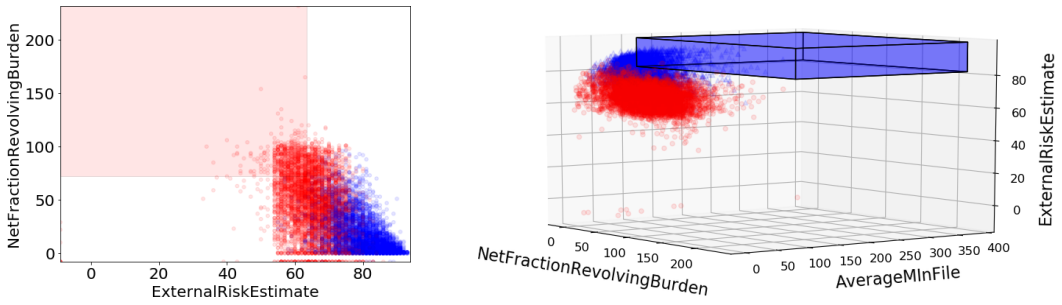

Figure 4: Graphical illustration of two rules (e.g., the left rule is $h_e(\mathbf{x}) = \mathbb{1}[\text{ExternalRiskEstimate} \leq 63 \ \& \ 73 \leq \text{NetFractionRevolvingBurden}]$). These rules are summary-explanations of all the points they contain, and are based on features that are interpretable within their context.

For rule-based models, we use cardinality as a natural measure for complexity, that is, $\Gamma_c(h_{R,\tau,y}) = |R|$. In this case, w_s and w_c have a more concrete interpretation: if $w_c = 10$ and $w_s = 1$, then we would trade 10 points of the support for one fewer condition in the rule.

The problem of finding a consistent rule-based summary-explanation for a point $\mathbf{x}_e$, which we will denote as `OptConsistentRule`, can then be written as follows, combining the formulation (1) with the specific model choice in (2):

$$\begin{array}{lll} \max_{R,\tau} & w_s \cdot \Gamma_s(h_{R,\tau,h^{\text{global}}(\mathbf{x}_e)}) - w_c \cdot |R| & \text{(maximize support, minimize complexity)} \\ \text{s.t.} & \forall p \in R : x_{e,p} \geq \tau_p & \text{(summary-explanation is relevant)} \\ \forall i \in N: & \text{If } h^{\text{global}}(\mathbf{x}_i) = 1 - h^{\text{global}}(\mathbf{x}_e) & \text{(summary-explanation is consistent)} \\ & \text{Then } \exists p \in R : x_{i,p} < \tau_p & \\ & |R| \leq p_c & \text{(interpretability)} \end{array} \quad (3)$$

Throughout the paper, we also use the terms cardinality and sparsity as synonym and antonym to complexity.

Geometrically, a globally-consistent rule is a hyperbox in the feature space that contains the observation we wish to explain and only other observations that are similarly labeled. It need not cover all similarly-labeled observations, but it must not cover observations from the opposite class. `OptConsistentRule` tries to find a box that simultaneously maximizes support and minimizes the number of facets. Figure 4 illustrates two rules. The first (on the left) represents a rule with two features (ExtrernalRiskEstimate and NetFraction-RevolovingBurden) and the second rule (on the right) representes a rule with 3 features (NetFractionRevolovingBurden, AverageMInFile, and ExtrernalRiskEstimate).

Observe that global consistency by Definition 1 is only one-way, not two-way. That is, *inside* the box, the box must agree with the underlying global model. This is not heavily restrictive as long as the global model does not have a singularity around the data point. *Outside* the box, there is no restriction.

3. Algorithms

We now study the properties of `OptConsistentRule` and develop algorithms for solving it. We begin in Section 3.1 by showing that `OptConsistentRule` is a theoretically challenging combinatorial optimization problem. We then use insights about its structure to design algorithms that are computationally tractable. Specifically, in Section 3.2, we develop algorithms for the case of binary datasets, and in Section 3.3 we address the case of datasets with continuous features.

3.1 Equivalence to Minimum Set Cover and Computational Complexity

We show that `OptConsistentRule` generalizes the *Minimum Set Cover* Problem (`MinSetCover`), which can be stated as follows (Williamson and Shmoys 2011): there is a ground set of elements $E = \{1, \ldots, n\}$, a collection of subsets $S_1, \ldots, S_\rho \subseteq E$, and the goal is to find the smallest collection of subsets that covers E; that is, find $R \subseteq \{1, \ldots, \rho\}$ where $\cup_{p \in R} S_p = E$ where R has small size. Formally,

$$\begin{array}{ll} \min_{R \subseteq \{1,\ldots,\rho\}} & |R| \\ \text{s.t.} & \exists p \in R : \mathbb{1}[i \in S_p] = 1 \qquad \forall i \in \{1, \ldots, n\} \end{array} . \tag{4}$$

While `MinSetCover` is known to be *NP*-hard (Bernhard and Vygen 2008), this does not mean we cannot solve it in practice, as we will show later. Let us show an equivalence between these two problems, and show that `OptConsistentRule` is more general, as follows.

Theorem 2 `OptConsistentRule` *generalizes the* `MinSetCover` *problem.*

Proof Equation (4) can be equivalently written as follows (since $i \in S_p \iff i \notin E \setminus S_p$, which is a double negative statement):

$$\begin{array}{ll} \min_{R \subseteq \{1,\ldots,\rho\}} & |R| \\ \text{s.t.} & \exists p \in R : \mathbb{1}[i \in E \setminus S_p] = 0 \qquad \forall i \in \{1, \ldots, n\} \end{array} . \tag{5}$$

We reduce `OptConsistentRule` to this version of the `MinSetCover` problem. Specifically, for each instance of `MinSetCover`, we construct an instance of `OptConsistentRule` where the data matrix $\mathbf{X}$ consists of n observations (corresponding to elements of the `MinSetCover` problem) and ρ binary features (corresponding to sets from `MinSetCover`). Figure 5 illustrates the idea. The globally consistent rule for `OptConsistentRule` is formed from a minimum set of hyperplanes that cover all observations whose labels differ from the global model's prediction on $\mathbf{x}_e$.

We assign feature p of observation i to 1 if and only if element e_i does not belong to set S_p, that is, $e_i \notin S_p$, that is,

$$x_{i,p} = 1 \iff i \in E \setminus S_p.$$

All predictions from the global model are set to +1. The observation we wish to explain is initialized as $\mathbf{x}_e = \mathbf{1}$ and its label is set to 0. The coefficient values are: $w_s = 0$, $w_c = 1$, and $p_c = \rho$.

The dataset is binary, that is, $\mathbf{X} \in \{0, 1\}^{|N| \times |P|}$. Without loss of generality, we set all threshold values to 1 (for any feature $p \in R$, a threshold value of $\tau_p > 1$ is not feasible; a

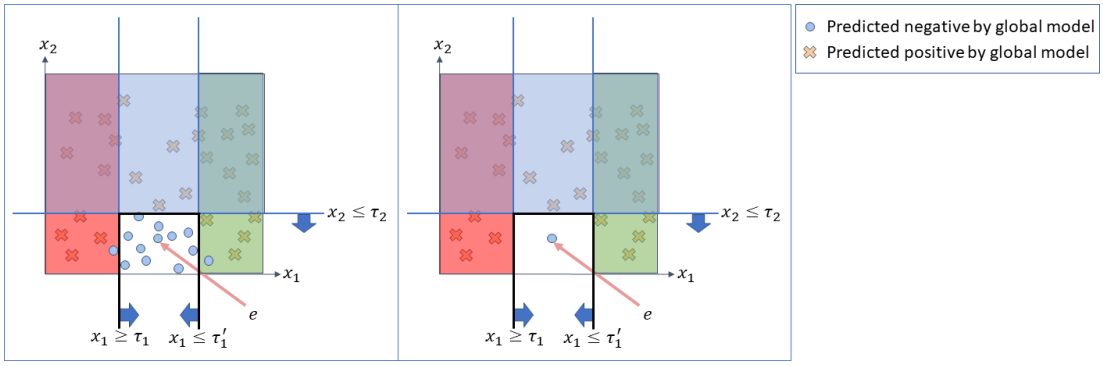

Figure 5: Idea of proof. The optimization problem finds a set of predicates that form a globally-consistent rule. It does this by forming a minimal cover for all observations whose labels are opposite to the global model's prediction on $\mathbf{x}_e$, while refraining from covering $\mathbf{x}_e$ (left panel). In the special case where $w_s = 0$ and $w_c = 1$, all observations whose prediction is equal to that of the global model's prediction on $\mathbf{x}_e$ (denoted in the figure as circles that are not $\mathbf{x}_e$) can be ignored when constructing the summary explanation (right panel).

threshold value of $\tau_p \leq 0$ is redundant; and any threshold value $0 < \tau_p < 1$ is equivalent to $\tau_p = 1$). Therefore, the only decision variables in Equation (3) are the set of features R, and the optimization problem can be written as:

$$\begin{array}{lll} \max_{R \subseteq \{1,\ldots,\rho\}} & -|R| & \text{(minimize complexity)} \\ \text{s.t.} & \forall p \in R : x_{e,p} \geq 1 & \text{(summary-explanation is relevant)} \\ \forall i \in N : & \text{If } h^{\text{global}}(\mathbf{x}_i) = 1 & \text{(summary-explanation is consistent)} \\ & \text{Then } \exists p \in R : x_{i,p} < 1 & \\ & |R| \leq \rho. & \text{(sparsity)} \end{array} \tag{6}$$

Now, by construction, the first and third constraints in the above formulation become trivial and can be removed (the first constraint holds because, by definition, $x_{e,p} = 1$ for all p, and the third constraint $|R| \leq \rho$ trivially holds). Moreover, the second constraint applies to all observations (whose predictions were all defined to be 1), and we obtain the following equivalent formulation:

$$\begin{array}{ll} \min_{R \subseteq \{1,\ldots,\rho\}} & |R| \\ \text{s.t.} & \forall i \in N : \exists p \in R : x_{i,p} = 0 \end{array} . \tag{7}$$

By construction $x_{i,p} = 0$ if and only if $i \in E \setminus S_p$. Therefore, the constraint in Equation (7) is equivalent to the constraint in Equation (5), and therefore the formulations are equivalent. This means that `OptConsistentRule` generalizes `MinSetCover`, since we had chosen a special case of `OptConsistentRule` that is equivalent to an arbitrary given instance of `MinSetCover`. ■

Since `MinSetCover` is NP-hard, we have the following:

Corollary 3 `OptConsistentRule` *is NP-hard.*

Let us provide some intuition about the equivalence between these two problems. The optimization problem for `OptConsistentRule` tries to find a set of predicates that form a globally-consistent rule by covering observations whose labels are opposite to $h^{\text{global}}(\mathbf{x}_e)$. In the proof of Theorem 2, we showed that, given a `MinSetCover` problem, we can construct an instance of `OptConsistentRule` whose formulation coincides with the former. This construction provides us with the insight that adding a predicate $x_p \geq \tau_p$ to a rule-based model has the effect of excluding parts of the space where observations whose sign differs from $h^{\text{global}}(\mathbf{x}_e)$ reside.

While `MinSetCover` is a theoretically difficult problem, from a practical standpoint, it can be easily implemented and solved using current computing technologies and Integer Programming (IP) solution techniques. Later in the numerical experiments section, we show that the running time for solving various problem instances is sufficiently low and can be used in practice.

3.2 Algorithms for Binary Datasets

Assume that the dataset is binary, that is, $\mathbf{X} \in \{0,1\}^{|N|\times|P|}$. As shown in the proof of Theorem 2, we may assume without loss of generality that all threshold values τ_p are equal to 1 and the only decision variables in Equation (1) are the set of features R. We can then simplify the optimization problem of `OptConsistentRule` to

$$\begin{array}{lll} \min_R & -w_s \cdot \Gamma_s(h_{R,\mathbf{1},h^{\text{global}}(\mathbf{x}_e)}) + w_c \cdot |R| & \text{(maximize support, minimize complexity)} \\ \text{s.t.} & \forall p \in R : x_{e,p} \geq 1 & \text{(summary-explanation is relevant)} \\ \forall i \in N: & \text{If } h^{\text{global}}(\mathbf{x}_i) = 1 - h^{\text{global}}(\mathbf{x}_e) & \text{(summary-explanation is consistent)} \\ & \text{Then } \exists p \in R : x_{i,p} = 0 & \\ & |R| \leq p_c. & \text{(sparsity)} \end{array} \tag{8}$$

Figure 6 illustrates a rule-based summary-explanation for a simple binary dataset: crosses and filled circles denote observations of two classes, the empty red circle denotes the observation being explained (which in this case, is also part of the data), while the shaded area depicts the part of the feature-space where the summary explanation applies. In this case, Rule (3) is dominated by Rule (1) and Rule (2), both of which are considered optimal. Later in Section 4, we find the best γ rules by applying a cutting-plane method that iteratively solves (8), each time with an additional constraint that turns the previously obtained solution into an infeasible solution. This method might first find Rule (1), then Rule (2) and then Rule (3).

Next, in Section 3.2.1, we develop the algorithm `BinMinSetCover` which solves the problem for the special case of optimizing for sparsity (that is, $w_s = 0$). In Section 3.2.2, we formulate an IP for solving the general optimization problem.

3.2.1 Algorithm BinMinSetCover.

We address the problem of finding summary-explanations with optimal sparsity. The problem of minimizing sparsity is a special case of Equation (8) where $w_s = 0, w_c = 1$, and

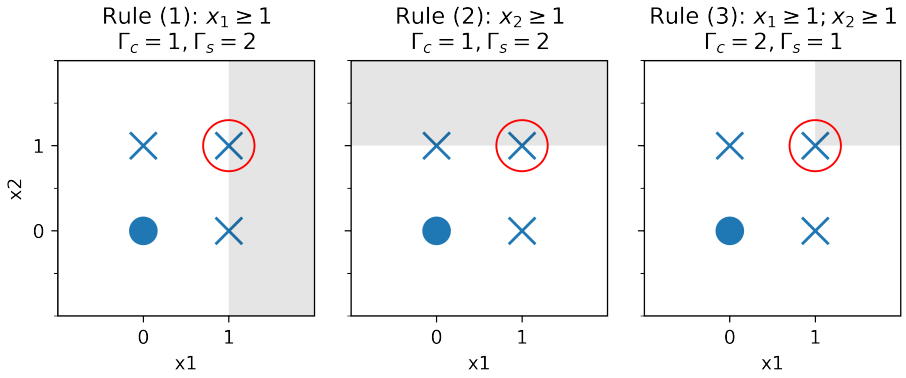

Figure 6: A simple example of rule-based summary-explanations for a binary dataset. The red circle denotes the observation being explained (which in this case, is also part of the data), while the shaded area depicts the part of the feature space where the summary-explanation applies. Rule (1) and Rule (2) both dominate Rule (3) because they cover more than just the one point of Rule (3) while using fewer thresholds.

$p_c = |P|$. The respective optimization problem can be written as

$$\begin{array}{lll} \min_{R \subseteq \{1,\ldots,m\}} & |R| & \text{(minimize complexity)} \\ \text{s.t.} & \sum_{p \in \{1,\ldots,m\}} \mathbb{1}[p \in R] \cdot \mathbb{1}[x_{e,p} = 1] \geq |R| & \text{(explanation is relevant)} \\ \forall i \in N: & \text{If } h^{\text{global}}(\mathbf{x}_i) = 1 - h^{\text{global}}(\mathbf{x}_e) & \text{(explanation is consistent)} \\ & \text{Then } \sum_{p \in \{1,\ldots,m\}} \mathbb{1}[p \in R] \cdot \mathbb{1}[x_{i,p} = 0] \geq 1. & \end{array} \tag{9}$$

Note that for the first constraint to hold, the set R must only contain features for which $x_{e,p} = 1$. Therefore, we may discard all features p where $x_{e,p} = 0$ and Constraint 1 will trivially hold. The second constraint is an equivalent way of representing Constraint 2 in Equation (8).

Recall that to allow our approach to generate rules that account for inequalities in both directions (e.g., $x_1 \geq \tau_1$ and/or $x_1 \leq \tau_1'$), we add to the data-matrix $\mathbf{X}$ all complementary values (that is, for each feature p where $\mathbf{X}_{:,p} \in \mathbf{X}$ it holds that $[\mathbf{1}_{|N|} - \mathbf{X}_{:,p}] \in \mathbf{X}$). `BinMinSetCover` is then feasible whenever there is no observation in $\mathbf{X}$ with identical values to $\mathbf{x}_e$ that is oppositely labeled (selecting every feature p such that $x_{e,p} = 1$ provides one such feasible solution). In other words, the solution to the problem is *always* feasible except in the pathological case when a positive observation and a negative observation sit on top of each other and we want a consistent summary explanation for one of these points. This would be impossible when summarizing predictions from models, as the prediction function must have a single value for each observation (otherwise it would not actually be a function). Thus, there is always a feasible solution when summarizing predictions from a global model at any point $\mathbf{x}_e$.

Let $P^e \triangleq \{p : x_{e,p} = 1\}$ denote the set of features that are equal to 1 in observation e. (We omit the features for which $x_{e,p} = 0$ from the optimization problem, since they can

never be part of the summary-explanation for e.) We can write the following equivalent integer program (IP):

$$\begin{array}{lll} \min_{\{b_p : p \in P^e\}} & \sum_{p \in P^e} b_p & \\ \text{s.t.} & \sum_{p \in P^e} b_p \cdot \mathbb{1}[x_{i,p} = 0] \geq 1 & \forall i : h^{\text{global}}(\mathbf{x}_i) = 1 - h^{\text{global}}(\mathbf{x}_e) \ , \\ & b_p \in \{0,1\} & \forall p \in P^e \end{array} \tag{10}$$

where the binary decision variable b_p indicates whether feature p is part of the rule $h_{R,\mathbf{1},y}$ (that is, $b_p = \mathbb{1}[p \in R]$). The sum only over $p \in P^e$ ensures that e will satisfy its summary-explanation, the min ensures that the solution is sparse, the first constraint again ensures that no i that disagrees with e on the global model's prediction is part of the summary-explanation, and the last constraint $b_p \in \{0,1\}$ encodes that b_p is an indicator as to whether p is part of the definition of the summary-explanation. Problem (10) is an instance of the minimum set cover problem, which can be solved by commercial solvers exactly using the formulation above, or approximately using approximation algorithms (e.g., those of Williamson and Shmoys 2011, Vazirani 2013). For example, a simple greedy algorithm for selecting features provides a $\ln(|P|)$-approximation; this can be used to efficiently generate summary-explanations for big datasets.

Now that we have a formulation to generate summary-explanations with optimal sparsity, let us move onto the more general problem we care about, where we trade off between sparsity and support in our summary-explanations.

3.2.2 Algorithm BinMaxSupport.

We use the Big-M method to formulate an IP for the general optimization problem (8). Let $N_e \triangleq \{i : h^{\text{global}}(\mathbf{x}_i) = h^{\text{global}}(\mathbf{x}_e)\}$ denote the set of observations that are labeled by the global model in the same way as e, whose label is $h^{\text{global}}(\mathbf{x}_e)$. We define r_i as a decision variable that indicates whether the rule we are trying to construct, namely $h_{R,\mathbf{1},h^{\text{global}}(\mathbf{x}_e)}$, predicts $h^{\text{global}}(\mathbf{x}_e)$ for observation i. We would like a large number of r_i's to be 1 in order to have high support. The optimization problem can be written as follows:

$$\begin{array}{lll} \max_{\mathbf{b},\mathbf{r}} & w_s \cdot \sum_{i \in N^e} r_i - w_c \cdot \sum_{p \in P^e} b_p & \\ \text{s.t.} & \sum_{p \in P^e} b_p \cdot \mathbb{1}[x_{i,p} = 0] \geq 1 & \forall i :\in N \setminus N^e \\ & \sum_{p \in P^e} b_p \cdot \mathbb{1}[x_{i,p} = 0] \leq M \cdot (1 - r_i) & \forall i \in N^e \\ & b_p \in \{0,1\} & \forall p \in P^e \\ & r_i \in \{0,1\} & \forall i \in N^e \\ & \sum_p b_p \leq p_c. & \end{array} \tag{11}$$

To ensure global consistency, the first constraint guarantees that the rule we are constructing does not apply to any observation in $N \setminus N^e$, which are points that disagree with the global model's prediction on point e. The first term in the objective counts the number of observations for which the rule applies (i.e., the support), while the second term characterizes the complexity. The second constraint ensures correctness of r_i, so that the rule indeed applies to the observations counted in the support. Note that for all practical purposes, the constant M could be set to $|P^e|$.

3.3 Algorithms for Continuous Datasets

We now leverage algorithms developed in the previous section for binary datasets to generate *optimal* summary-explanations for continuous datasets. When we do this, there will be no discretization choices by the user, and thus no performance loss due to discretization. The method is more sophisticated than previous approaches and uses dynamic programming, as we will discuss.

Specifically, our approach is to first apply `BinMinSetCover` to generate "basic solutions," which are sparse continuous summary-explanations, using a reduction of the continuous dataset to a binary dataset (Section 3.3.1). Then, we apply a dynamic programming (DP) based algorithm to expand these summary-explanations in order to optimize their support while maintaining sparsity (Section 3.3.2). While using discrete optimization methods for continuous data may seem unnatural, given the uncountable number of possible boxes for continuous data, there are effectively a finite number of equivalence classes of boxes. Any two boxes are equivalent if they use the same variables and contain the same data. This results in a discrete and finite set of possible boxes.

Example. Consider Figures 7 and 8, which illustrate a 2-dimensional continuous dataset (the values of features x_1 and x_2 vary between 10 and 12) where we wish to explain observation $\mathbf{x}_e = (12, 11)$ (circled). Figure 7 illustrates the first step in our approach of generating "basic" summary-explanations where the value of each threshold is equal to the value of the respective feature in the observation being explained (i.e., feature x_1 is either at most 12, at least 12, both (i.e., equal to 12) or none (i.e., x_1 is not part of the summary-explanation); the same holds for feature x_2 and the threshold value 11). In total, we generate 6 summary-explanations to start with, each shown in a different subfigure in Figure 7. The shaded area in each plot corresponds to the part of the feature space where that plot's explanation model (and the underlying model due to global-consistency) predicts $h^{\text{global}}(\mathbf{x}_e)$. According to formulation (3), the quality of a summary-explanation is measured by its support Γ_s (the number of observations contained within the shaded region) and its complexity Γ_c (the number of facets, or thresholds used to define the shaded area), both of which are illustrated in the figure (Section 3.3.1 below discusses how such basic summary-explanations can be obtained). Figure 8 shows the result of applying the second step of our approach. The illustrated rule is a potentially optimal solution (summary-explanation) that was generated by expanding "Rule (6)" of Figure 7. The solution in Figure 8 was obtained by systematically relaxing the thresholds of Rule (6), which was obtained using `BinMinSetCover`, in every possible direction until violating global consistency (i.e., relaxing the rule $h_e(x_1, x_2) = \mathbb{1}\left[12 \leq x_1 \leq 12 \;\&\; 11 \leq x_2 \leq 11\right]$ to $h_e(x_1, x_2) = \mathbb{1}\left[11 \leq x_1 \leq 12 \;\&\; x_2 \leq 11\right]$). This systematic relaxation is done by applying a DP-based algorithm (discussed below in Section 3.3.2). Note that the same solution might have been obtained by expanding several boxes. E.g., the rule in Figure 8 can be obtained from either Rules (4) or (6) in Figure 7, both of which were obtained from the first step using `BinMinSetCover`.

Similarly to the algorithms for binary datasets, here too we expand the data-matrix $\mathbf{X}$ to include the opposite features. That is, for each feature $p \in P$, there exists a feature $p^c \in P$ such that $\mathbf{X}_{\cdot,p^c} = -\mathbf{X}_{\cdot,p}$.

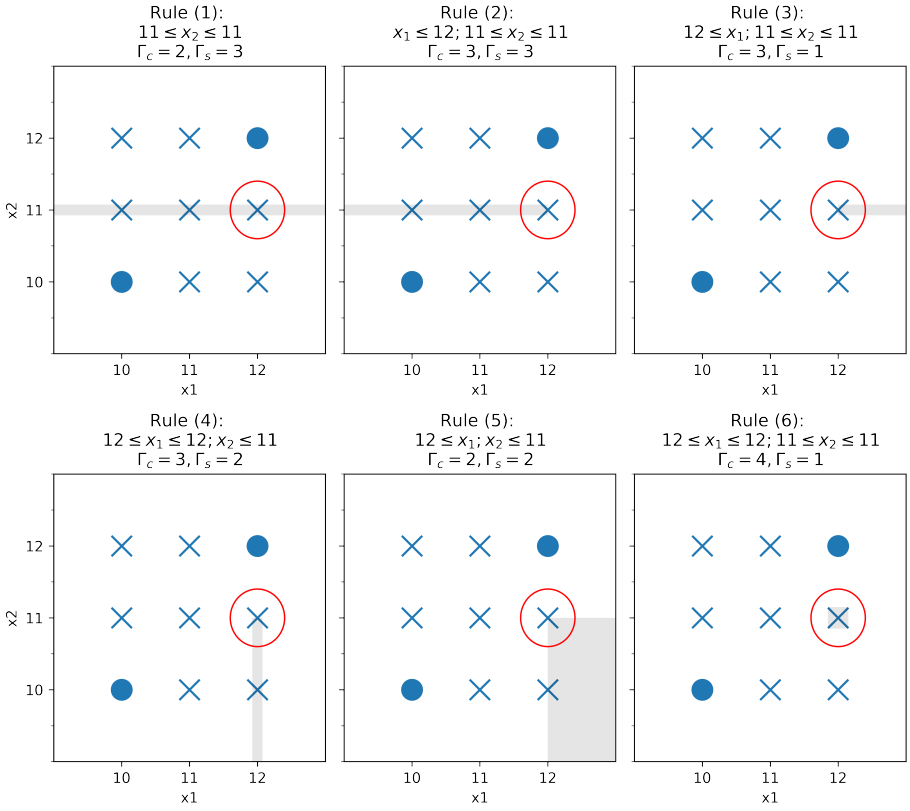

Figure 7: A simple example of "basic" rule-based summary-explanations for a continuous dataset.

3.3.1 Algorithm ContMinSetCover.

We begin with an observation about rules that attain optimal sparsity. Namely, the following lemma states that for each globally-consistent rule $h_{R,\boldsymbol{\tau},h^{\text{global}}(\mathbf{x}_e)}$, where a threshold τ_p is not equal to $x_{e,p}$, there is another globally-consistent rule $h_{R,\boldsymbol{\tau}',h^{\text{global}}(\mathbf{x}_e)}$ whose threshold τ_p is equal to $x_{e,p}$. This follows from the requirement for global-consistency which guarantees that $\tau_p \leq x_{e,p}$ (and therefore we can always raise the threshold to $x_{e,p}$).

Lemma 4 *For any data-matrix $\boldsymbol{X}$, model h^{global}, observation $\boldsymbol{x}_e, h^{\text{global}}(\boldsymbol{x}_e)$, and globally-consistent rule $h_{R,\boldsymbol{\tau},h^{\text{global}}(\boldsymbol{x}_e)}$, there exists a globally-consistent rule $h'_{R',\boldsymbol{\tau}',h^{\text{global}}(\boldsymbol{x}_e)}$ where $R' = R$, $\tau'_p = x_{e,p}$, and whose complexity is equal to that of $h_{R,\boldsymbol{\tau},h^{\text{global}}(\boldsymbol{x}_e)}$ (that is, $\Gamma_c(h_{R,\boldsymbol{\tau},h^{\text{global}}(\boldsymbol{x}_e)}) = \Gamma_c(h'_{R',\boldsymbol{\tau}',h^{\text{global}}(\boldsymbol{x}_e)})$).*

Proof Geometrically, any rule h defines a box polytope that contains $\mathbf{x}_e$ but excludes observations whose labels are $1 - h^{\text{global}}(\mathbf{x}_e)$. Therefore, any other box that is contained within h and contains $\mathbf{x}_e$ is a globally-consistent rule-based summary-explanation. This holds when we alter any facet of the box $x_p \geq \tau$ to $x_p \geq x_{e,p}$. Since raising the threshold values τ up to $x_{e,p}$ does not change the cardinality Γ_c of the rule, we obtain a new rule with

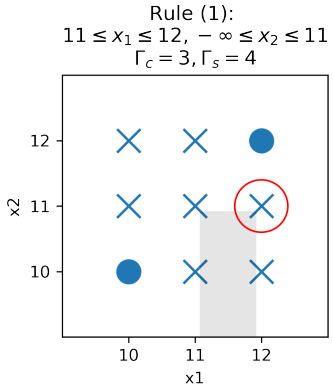

Figure 8: A simple example of a potentially optimal rule-based summary-explanation for a continuous dataset, found in the second step of our approach. We will later determine that this rule is not optimal, because a sparser rule with higher support exists.

identical cardinality. ■

The main insight from Lemma 4 is that in order to compute the most sparse solution, we need only to decide on the subset of features R, while τ can be fixed to $\mathbf{x}_e$. Including a feature within R excludes certain observations from the support of the resulting rule, and the goal is to find the minimal set of features that would exclude all observations that are labeled as $1 - h^{\text{global}}(\mathbf{x}_e)$. This is exactly the problem solved by `BinMinSetCover`. We formulate Algorithm 1, which first constructs a data matrix $\mathbf{X}^e \in \{0,1\}^{|N \setminus N_e| \times |P|}$, where $x^e_{i,p} = \mathbb{1}[x_{i,p} < x_{e,p}]$ (that is, each entry denotes if including the feature p would exclude observation $i \in N \setminus N_e$). Then, the algorithm applies `BinMinSetCover` to find the subset of features to be included in the most sparse rule that is globally-consistent.

Algorithm 1 Algorithm `ContMinSetCover`

Input: data matrix $\mathbf{X}$, observation to explain $\mathbf{x}_e$, and global model predictions $\{h^{\text{global}}(\mathbf{x}_i)\}_{i=1}^N$, and $h^{\text{global}}(\mathbf{x}_e)$.
Output: globally-consistent rule $h_{R,\tau,h^{\text{global}}(\mathbf{x}_e)}$.

1. Compute $\mathbf{X}^e \in \{0,1\}^{|N \setminus N_e| \times |P|}$, where $x^e_{i,p} = \mathbb{1}[x_{i,p} < x_{e,p}]$ for all $i \in N$ and $p \in P$.
2. Solve `BinMinSetCover` using $\mathbf{X}^e, \{h^{\text{global}}(\mathbf{x}_i)\}_{i=1}^N, \mathbf{x}_e, h^{\text{global}}(\mathbf{x}_e)$ to compute the subset of features R.
3. Return the rule $h_{R,\mathbf{x}_e,h^{\text{global}}(\mathbf{x}_e)}$.

Next, we will include support, and we will use dynamic programming to construct the summary-explanations.

3.3.2 Algorithm ContMaxSupport.

We now use `ContMinSetCover` to develop the algorithm `ContMaxSupport` for identifying globally-consistent rules with optimized support. The basic ideas behind the algorithm are:

1. Use `ContMinSetCover` to extract a "basic" rule with optimal sparsity. For example, a sparse summary explanation for $\mathbf{x}_e = (10, 20, 30, 40), h^{\text{global}}(\mathbf{x}_e) = +1$ could be the rule
$$h_{R,\mathbf{x}_e,h^{\text{global}}(\mathbf{x}_e)} = [(x_2 \geq 20 \ \ \& \ \ x_3 \geq 30) \rightarrow +1]$$
where $R = \{2, 3\}$ and whose support could be 100 observations.

2. Expand the rule by decreasing the thresholds to increase support while maintaining optimal sparsity. For example, expanding the above rule could result in the rule
$$h_{R=\{2,3\},\tau=\{15,27\},h^{\text{global}}(\mathbf{x}_e)} = [(x_2 \geq 15 \ \ \& \ \ x_3 \geq 27) \rightarrow +1],$$
whose support could be 150 observations. The expansion is carried out by solving a DP for each rule we want to construct.

3. Repeat the previous steps γ times, each time excluding previously encountered basic solutions. Such rules could be obtained by rerunning `BinMinSetCover` (as a subroutine of `ContMinSetCover`) with an additional constraint that prohibits previous solutions. E.g., to prohibit the basic rule with $R = 2, 3$, the following constraint could be added to Equation (10): $(1 - b_1) + b_2 + b_3 + (1 - b_4) \leq 3$.

We next describe the DP formulation for expanding rules and present `ContMaxSupport` in Algorithm 2. Given a rule $h_{R,\mathbf{x}_e,h^{\text{global}}(\mathbf{x}_e)}$ that needs to be expanded, we define the following DP:

- State-space: the state-space $\{\tau\}$ is $|R|$-dimensional, representing the possible values of the thresholds τ_p for $p \in R$: $\tau_p \in \{-\infty, x_{e,p}\} \cup \{x_{i,p} : i \in N_e, x_{i,p} \leq x_{e,p}\}$ (observe that threshold values larger than $x_{e,p}$ do not satisfy relevance, see Definition 1). We define the initial state as $\tau_0 = (x_{e,p} : p \in R)$.

- Reward function: the objective function, that is, $w_s \cdot \Gamma_s(h_{R,\tau,h^{\text{global}}(\mathbf{x}_e)}) - w_c \cdot |R|$ (see Formulation (3)).

- Action space: we define R actions corresponding to the p thresholds. An action p decreases a certain threshold τ_p to the next highest value in $\{-\infty, x_{e,p}\} \cup \{x_{i,p} : i \in N_e, x_{i,p} \leq x_{e,p}\}$. We denote the outcome of action p by $\tau \ominus p$. Note that the action corresponding to feature p cannot be selected when $\tau_p = -\infty$.

- Bellman's equation: can be written as
$$J(\tau) = \max \begin{cases} -\infty & // \ h_{R,\tau,h^{\text{global}}(\mathbf{x}_e)} \text{ is not globally-consistent} \\ w_s \cdot \Gamma_s(h_{R,\tau,h^{\text{global}}(\mathbf{x}_e)}) - w_c \cdot |R| & // \text{ select current state} \\ J(\tau \ominus p) & // \text{ explore the next state.} \end{cases}, \tag{12}$$
and we find the optimal expanded rule $h_{R,\tau,h^{\text{global}}(\mathbf{x}_e)}$ by computing $J(\tau_0)$.

Note that a different DP is constructed for expanding each basic rule.

Figure 9 illustrates the construction of the state-space for a 2-dimensional dataset. The figure on the left shows a basic summary-explanation for the positively labeled observation e using the rule: $x_1 \leq \tau_1$, $\tau_1 \leq x_1$, and $x_2 \leq \tau_2$. The state-space in the corresponding DP is 3-dimensional (one dimension per each of the 3 thresholds), and the initial state is equal to (τ_1, τ_1, τ_2). The state-space is the Cartesian product of the projected positively labeled observations onto each of the axes, truncated by the feature values of $\mathbf{x}_e$, and appended by the bounding values of $x_{e,p}$ and $-\infty$. The figure on the right shows additional rules (boxes), each corresponding to a state reached while computing Bellman's equation. Each tick on each axis corresponds to a value in one of the dimensions of the state-space. The DP algorithm systematically expands the basic solution in every direction, exploring all feasible solutions.

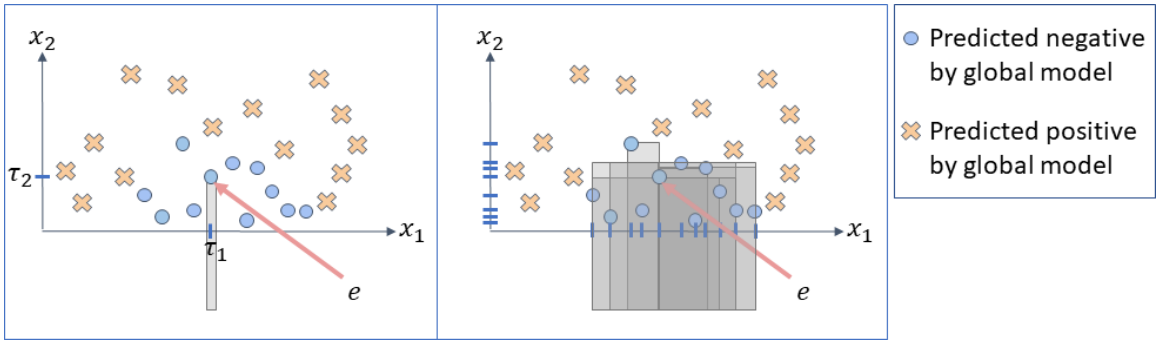

Figure 9: An illustration of basic rule and states of the corresponding DP. Left: the initial state of the DP, which is a summary-explanation for observation e (in gray). Right: states of the DP, each corresponding to a summary-explanation.

`ContMaxSupport` obtains rules with good sparsity by running `ContMinSetCover` and performing expansion operations that, while improving support, do not increase the cardinality of the resulting rules (i.e., no features are added by the DP). In fact, when a selected threshold is expanded to $-\infty$, sparsity improves as well; sending a boundary to $-\infty$ removes a box boundary.

The value γ in Algorithm 2 is the number of times we repeat the process of finding a rule (that is not equivalent to one we have already seen). Observe that Algorithm 2 returns the optimal solution when γ is sufficiently large. This is because each of the γ solutions is unique: all previous solutions are excluded when constructing the next solution. In this way, large γ would allow us to enumerate all feasible (equivalence classes of) boxes, after which point, the optimization problem becomes infeasible.

Thus, we now have a method that produces globally-consistent summary-explanations for continuous variables that are sparse and have high support.

Algorithm 2 Algorithm `ContMaxSupport`

Input: γ (number of initial rules to extract), w_s and w_c (objective coefficients), $\mathbf{X}$ (data matrix), observation to explain $\mathbf{x}_e$, and predictions $\{h^{\text{global}}(\mathbf{x}_i)\}_{i=1}^N, h^{\text{global}}(\mathbf{x}_e)$.
Output: globally-consistent rule $\left(R, \tau, h^{\text{global}}(\mathbf{x}_e)\right)$.

1. Apply `ContMinSetCover` to extract γ rules with optimal sparsity (e.g., by running `ContMinSetCover` iteratively with additional cutting-planes that prohibit previous solutions).

2. Apply the DP formulation (12) to each of the extracted rules $h_{R,\tau,h^{\text{global}}(\mathbf{x}_e)}$ to increase their support.

3. Return the expanded rule whose objective value is maximal.

4. Numerical Experiments

We conducted a computational study to assess the quality of the summary-explanations generated by our algorithms on a real-world dataset provided by FICO. We show that the algorithms can be used to effectively generate sparse summary-explanations with large support in seconds (in the case of `BinMinSetCover`, `ContMinSetCover`, and `ContMaxSupport`) or at most minutes (in the case of `BinMaxSupport`) that could be used in practice. In Section 4.1 we describe the dataset and the computational setting, and in Sections 4.2 and 4.3 we discuss the results obtained by applying the algorithms to binary and continuous datasets, respectively. *There are no other approaches that aim to generate globally-consistent rule-based explanations*, so our goal in these experimental sections is to show that our method is useful, scalable, and practical for this new task. In Section 4.4, we conduct additional tests to evaluate our algorithms on a different dataset. In Section 4.5, we compare our results to an alternative method for generating rule-based explanations.

4.1 The Computational Setting

Data: The dataset[2] contains "an anonymized dataset of Home Equity Line of Credit (HELOC) applications made by real homeowners. A HELOC is a line of credit typically offered by a bank as a percentage of home equity (the difference between the current market value of a home and its purchase price)." The features are based on credit bureau data with interpretable features, and the target variable indicates whether a consumer was 90 days past due or worse at least once over a period of 24 months from when the credit account was opened. The dataset contains a total of 9,871 observations and 23 categorical and numerical features. The dataset is balanced, with 52% of customers late on payments. The sample appears to arise from a subpopulation that is harder to classify than the general population of individuals seeking a loan.

Preprocessing: We used the following methods to preprocess the data prior to training models:

2. The dataset can be downloaded, see ref. FICO (2018).

- "Original"—use the dataset as is (missing values were encoded as -7,-8, and -9);
- "Missing as binary"—add binary features for each variable with missing values to indicate a missing value of a particular feature (e.g., $x_7\_missing$ is a new feature indicating that the value of feature x_7 is missing for a particular observation), which could be useful in the case that missingness is correlated with outcome information;
- "x quantiles"—in order to attain binary features (to apply algorithms for binary datasets), we discretized continuous features based on the distribution of each feature using 2, 4, 8, and 16 equal quantiles (e.g., assuming the values of the variable x_7 are $[1, 2, 3, \ldots, 100]$, using 4-quantiles would result in 4 binary variables $x_7[Q1]$, $x_7[Q2]$, $x_7[Q3]$, and $x_7[Q4]$ where $x_7[Q2] = 1$ if and only if $26 \leq x_7 \leq 50$);
- "Manual"—we discretized each feature manually by examining the conditional expectation of the output variable as a function of the input variable, defining break points at feature values where the conditional distribution is significantly different or non-smooth. This type of analysis of one dimension at a time is typical in practice (and easy, since it involves only one dimension at a time).

The last two preprocessing methods yield binary datasets (to which the algorithms `BinMinSetCover` and `BinMaxSupport` could be applied) while the first two yield continuous datasets (to which the algorithms `ContMinSetCover` and `ContMaxSupport` could be applied).

We compared the accuracy of different global models (described below) using the different preprocessing methods and found that the models are comparable in terms of their accuracy (see Figures 21, 22, and 23 in Appendix A). Therefore, we simply apply `BinMinSetCover` and `BinMaxSupport` on the "Manual" preprocessing method, and we apply `ContMinSetCover` and `ContMaxSupport` on the "Missing as binary" method.

We note in passing that a considerable effort in which we tried various transformations of the features and other discretization methods did not lead to improved predictions (Chen et al. 2022). The accuracy level attained for the prediction task is consistent with other studies on other (structured) datasets for credit risk analysis (e.g., Baesens et al. 2003).

Global predictive models: Our global models include: K-nearest neighbors (KNN), logistic regression (Log. Reg.) SVM with linear, polynomial, and radial basis function (RBF) kernels, classification trees (CART), random forests (RF), and boosted decision trees (AdaBoost).

We trained the aforementioned global models on a random training set consisting of 75% of all observations. We performed hyper-parameter tuning for the global models using cross-validation within the training set on a wide range of parameter values to optimize the choice of parameters. Models were then evaluated on the test data (the remaining 25%). Summary-explanations were constructed using our method for all of these global models. Whereas the global models were created using training sets, summary-explanations were created using the entire dataset. (In Section 4.4 we evaluate the algorithms using out-of-sample test data).

Figure 24 in Appendix A presents a comparison between the predictions made by the different models. We see that while accuracy is generally similar, the models differ in

how they make predictions. Constructing summary-explanations for this variety of models serves as a robustness test for our algorithms: ideally, our algorithms should perform well regardless of which machine learning approach generated the underlying model. We evaluate our summary explanations in the following subsections.

Practical adaptation to our algorithms: To find summary-explanations on binary datasets, we first solve `BinMinSetCover` to find summary-explanations with optimal sparsity (denoted as "Min Features"). Denoting by p_c^{OPT} the minimal number of features obtained by optimizing for the sparsity of the resulting rule, we then solve `BinMaxSupport` 3 times, setting $w_s = 1$, $w_c = 0$, and p_c to $p_c^{\text{OPT}}, p_c^{\text{OPT}} + 1$, and $p_c^{\text{OPT}} + 2$. That is, we maximize support and gradually relax the restriction on the maximal number of features (i.e., sparsity) of the resulting rules. We denote the latter algorithm as "Max support + a", where $a \in \{0, 1, 2\}$. In addition, while generating summary-explanations for each observation e, we use as initial feasible solutions the solution of the previous optimization problem (that is, the solution to `BinMinSetCover` serves as an initial solution to Max support + 0; the solution to Max support + 0 then serves as an initial solution to Max support + 1, and so forth). To find summary-explanations on continuous datasets, we applied `ContMinSetCover` and `ContMaxSupport` as is.

Implementation: The code for the numerical experiment was implemented in Python using scikit-learn (Pedregosa et al. 2011) and Gurobi (Gurobi 2014). The total running time of the experiment was approximately two weeks on 3 personal computers, setting a maximal timeout of 1 minute for solving IPs. A total of 394,940 summary-explanations were generated.

4.2 Binary datasets

The results are summarized in Figure 10 (number of terms/sparsity and support per algorithm). Figure 11 provides more detail, giving the average number of terms, support and runtime per model and algorithm. The key findings are listed below:

- **Our rules tend to be very sparse, even on real datasets.** In Figure 10, we see that the resulting rules (summary-explanations) are surprisingly sparse, requiring on average less than 3 features, and in 90% of cases requiring 4 or less features. At the same time, the support of these rules is large with an average value of 778 observations per rule, and in 90% of cases the support is larger than 17. Moreover, the average support increases by roughly 300 observations by adding only one feature to the summary-explanations.

- **`BinMinSetCover`'s running time is fast.** Figure 11 shows that `BinMinSetCover` is typically solved to optimality in less than 15 seconds. All other rules were obtained by setting a time limit of 60 seconds. These run times are sufficient for most practical applications, and additional algorithmic improvements can be made to speed up the generation of summary-explanations. For example, the support of some rules includes more than 1000 observations. This means that generating a rule for one of these observations could be used for the remaining 999 observations; this saves the computation of generating rules 999 more times. Pre-computing some rules and using those as

initial solutions in future optimization problems would also be a way to significantly improve the running time in practice.

- **Robustness to the underlying global model.** We see in Figure 11 that our approach produces high quality rules (in terms of sparsity and support) for the entire collection of considered global models. Our rules are (by design) always consistent with the underlying global model in the local region where they apply.

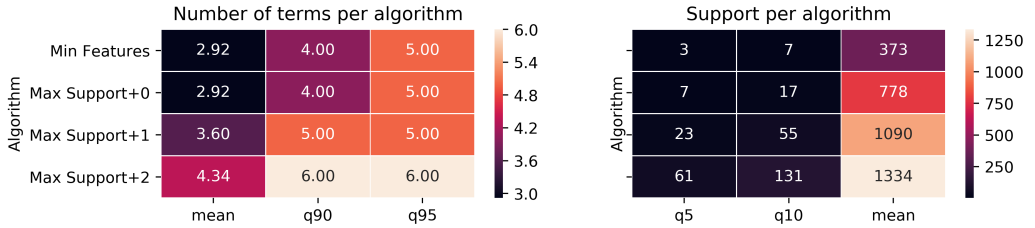

Figure 10: Number of terms and support per algorithm, averaged over the rules generated using the complete binary FICO dataset using Algorithms `BinMinSetCover` (denoted as "Min Features") and `BinMaxSupport` for binary datasets (denoted as "Max Support + d" with d indicating the relaxation of the sparsity constraint), and averaged over global models. Each cell in the tables is an average over 78,968 rules. Here, "q90" denotes .9-quantile (similar notation is used for other quantiles). Figure 11 provides a breakdown of these results per global model and gives timing information.

4.3 Continuous datasets

We generated summary-explanations for the predictive models described in Section 4.1 and measured sparsity, support and running time. The main results are presented in Figure 12 and more detailed results are in Figure 13. Similarly to the binary datasets, we find that:

- **The summary-explanations generated using `ContMinSetCover` and `ContMaxSupport` are sparse,** consisting of 2.14 terms on average, and in 95% of all cases consist of 3 or less terms.

- **The support of the generated rules is quite large**, with an average value of 1020, and in 90% of all cases the support is larger than 29 (Figure 12).

- **Our algorithms are robust to the underlying global model.** As noted above, there is some variability between different global models. The resulting summary-explanations are good for all of these models (Figure 13).

- **The running time is quite fast.** Without any time limit, the rules were obtained consistently in under 30 seconds (in contrast to the algorithms for binary datasets where a time limit of 1 minute was set for solving the IPs).

Figure 11: Average number of terms, support and runtime (in seconds) for generating 1 rule per model, using the complete binary FICO dataset, and Algorithms `BinMinSetCover` and `BinMaxSupport` for binary datasets. Each cell in each table is an average over 9,871 rules.

In conclusion, the numerical experiments suggest that the proposed algorithms for generating summary-explanations from binary and continuous datasets work well and can be used in practice.

4.4 Robustness checks

Here, we consider an alternative dataset with a larger number of features, and study how the size of set used to generate explanations affects the resulting summary-explanations in terms of run time, sparsity of the generated rules, support, and global-consistency using both in-sample and out-of-sample data.

Dataset. We use the US 1990 Census dataset (Meek et al. 2022) whereby we predict whether an individual's annual income exceeds \$15,000 ($dRpincome \geq 3$) based on demographic and financial information. We binarize the categorical features to obtain a binary

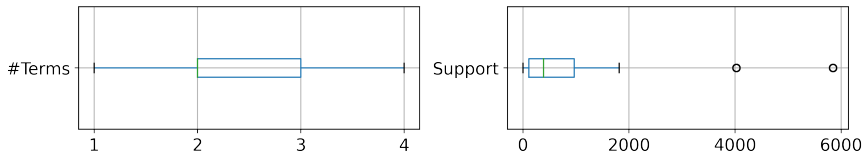

Figure 12: Distribution of the number of terms and support per rule for algorithm `ContMaxSupport` for continuous datasets (using the complete continuous FICO dataset across all global models, resulting in a total of 78,968 rules). The average, 90th quantile, and 95th quantile of the number of terms are 2.14, 3.00, and 3.00, respectively. The 5th quantile, 10th quantile, and average of the support are 13, 29, and 1020, respectively. Figure 13 has a breakdown of results with respect to the different global models.

Figure 13: Average number of terms, support and runtime (in seconds) per rule, for each global model on the FICO dataset (Algorithm `ContMaxSupport` for continuous datasets).

dataset with a total of 368 features. 33% of the labels are 1 (income over 15K) and the remaining observations are labeled 0. We used the first 100K observations of the dataset.

Global models and hyper-parameters. We split the dataset into a training set and a test set, each consisting of 50K observations. Using the training set, we train and evaluate

3 global models: LR ($max\_iter = 1000$), RF ($max\_depth = 7$, $min\_samples\_split = 5$), and ANN ($max\_iter = 1000$, $hidden\_layer\_sizes = (4, 2)$). Figure 14 compares the global models and presents their accuracy, which is close to 90%.

Generating and evaluating summary-explanations. Using the trained models and the test set, we generate and evaluate the global consistency of the summary-explanations. Specifically, we generate summary-explanations for 1,000 observations in the training set. These summary-explanations are generated using an *explanation train-set* consisting of 1,000–9,000 training observations, and, for each of these explanations, we measure the number of times global consistency is violated on the 50,000 test observations (which were not used in the training of the model or the generation of explanations).

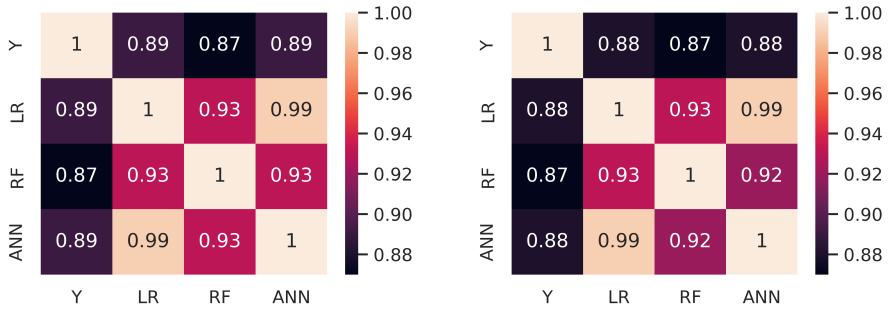

Figure 14: Census dataset global model comparison. The values indicate the percentage of observations in the training set (left) and test set (right) for which pairs of models predict identically; Y indicates the ground truth and the corresponding values indicate models' accuracy.

Results. We summarize below the key results. In what follows, we denote the set for which explanations are generated as the *explanation train-set*.

- **Runtime.** Figure 15 shows the average running time for generating summary-explanations as a function of the size of the explanation train-set. We observe that the running time increases roughly at a linear rate. As suggested earlier, for very large datasets, one could use a sample of the explanation train-set, or devise more efficient algorithms that leverage the structure of the problem, to improve running time. For instance, it is possible that almost all points far away from the decision boundary may be summarized with only one or two summary-explanations. Also, we might choose to omit points that are very far away from the current point for which we are designing a summary-explanation in order to save computation.

- **Sparsity.** Figure 16 shows the average number of features used in the summary-explanations as a function of the explanation train-set size. While we observe a decline in sparsity, overall it is quite mild and it appears to be converging.

- **Support.** Figure 17 shows the average support of the summary-explanations as a function of the explanation train-set size. This appears to be increasing at a roughly linear rate. Observe that while adding observations adds constraints that make the generation of rules more difficult, it also increases the overall number of observations, which results in higher support values.

- **Global consistency.** Figure 18 shows the average number of inconsistencies in the out-of-sample data as a function of the explanation train-set size.

  We see that when the explanation train-set size is relatively small (1000 observations for a dataset with 368 features), there are some inconsistencies (under 70 on average out of 50,000 test observations). As the size of the explanation train-set increases, the number of inconsistencies becomes almost negligible (under 10 on average out of 50,000 test observations).

  Of course, we cannot completely ensure global-consistency with respect to out-of-sample data, but the number of violations is relatively small and the summary-explanations are truthful (they make an accurate statement about previous cases, which are in the training set).

  Comparing `BinMinSetCover` with `BinMaxSupport`, we observe that `BinMaxSupport` returns rules with a higher support, but has a higher risk of overfitting in comparison with `BinMinSetCover`. This is potentially because it expands its rules to the nearest decision boundaries.

Overall, these experiments illustrate the effectiveness of our algorithms, and the importance of taking global-consistency into consideration in the design of algorithms that generate explanations.

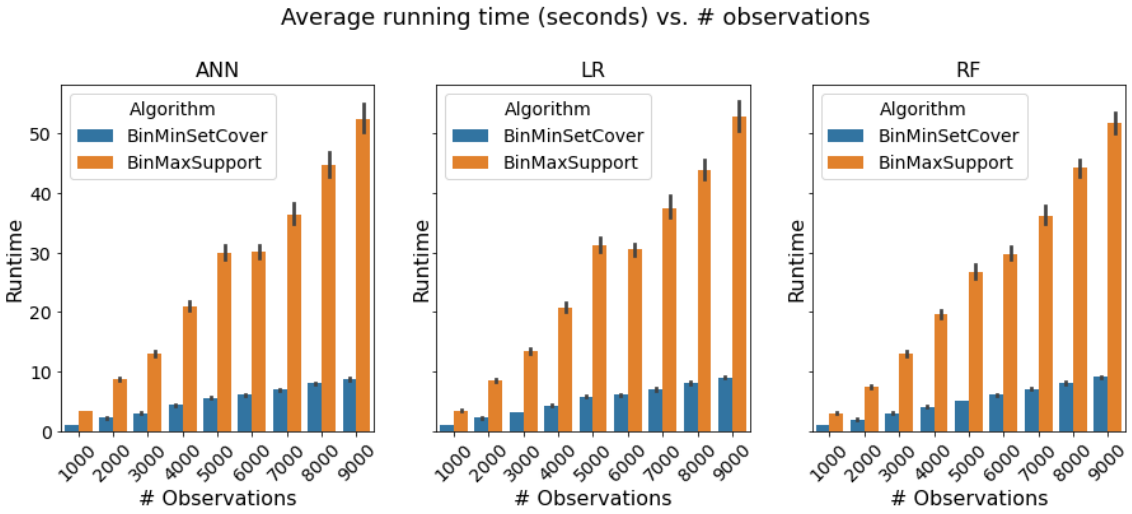

Figure 15: Average time for generating summary-explanations as a function of the explanation train-set size (Census dataset).

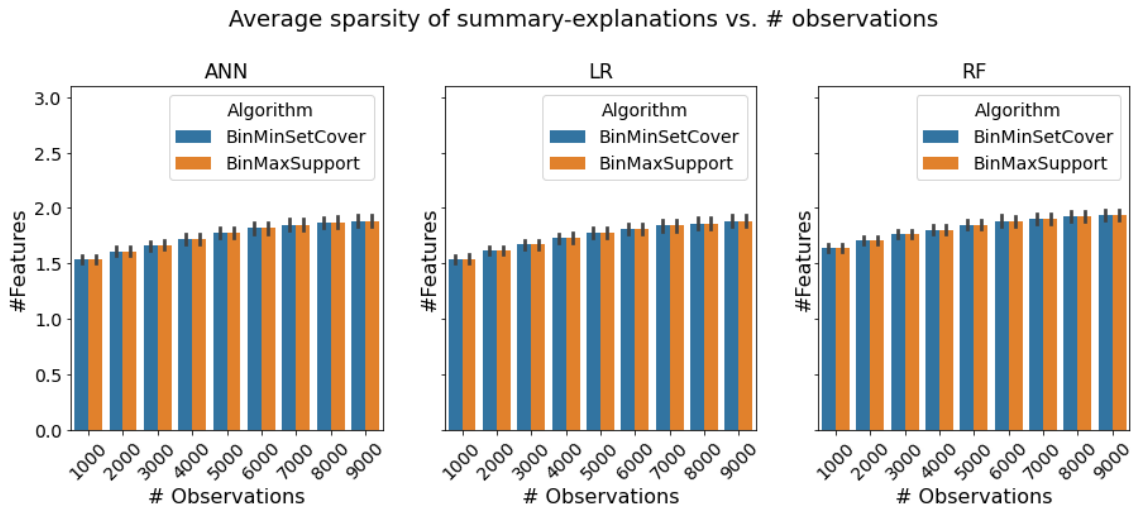

Figure 16: Average sparsity as a function of the explanation train-set size (Census dataset).

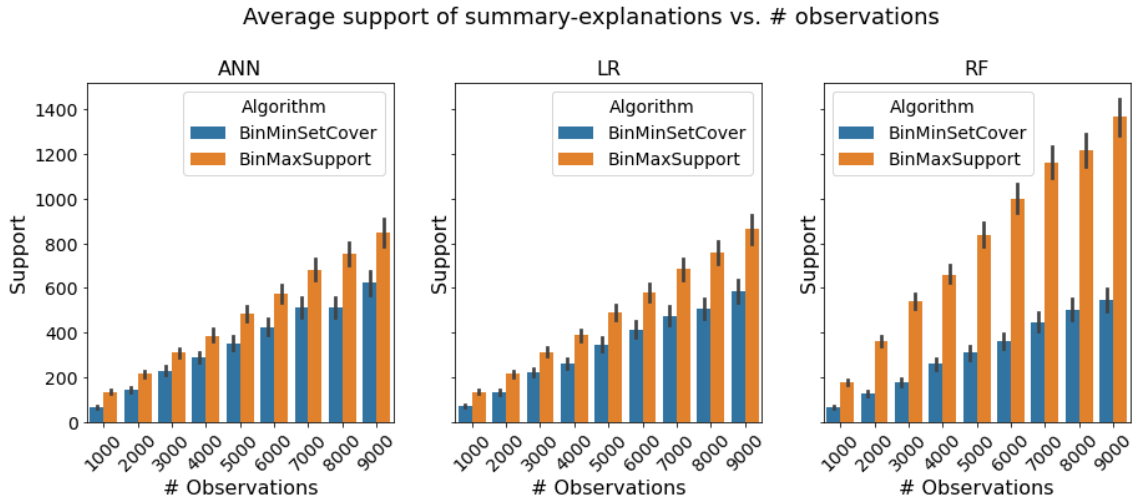

Figure 17: Average support as a function of the explanation train-set size (Census dataset).

4.5 Comparison with Anchors

We also compare our methods to Anchors (Ribeiro et al. 2018), which can also be used to create rule-based explanations. Figure 19 uses the same setup and dataset as the previous subsection. We set the maximal running time of our algorithms to 10 minutes and used the default parameters of Anchors (the resulting running times were comparable). Similarly to the previous experiment, we generate Anchor-explanations and summary-explanations for observations in the training set and count the number of inconsistencies in the training set (top right) and test set (top left). Specifically, we generated explanations for 1,000

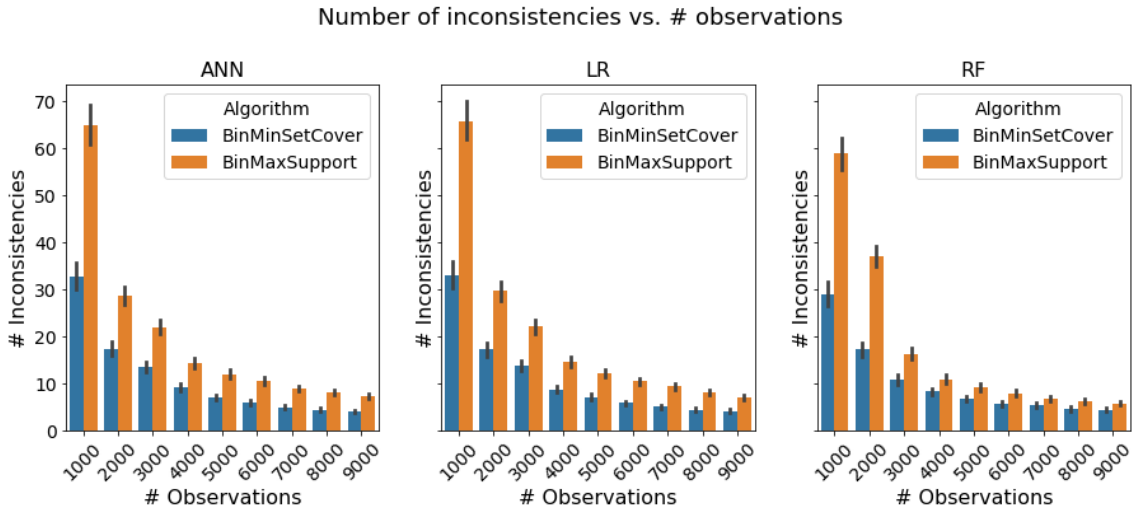

Figure 18: The average number of inconsistencies in the test set as a function of the explanation train-set size (Census dataset).

training set observations using an explanation train-set consisting of 9,000 training set observations, which resulted in a total of 6,000 instances (3 global models $\times$ 2 algorithms, namely `BinMinSetCover` with `BinMaxSupport`, $\times$ 1,000 observations). We generated the Anchor-explanations for the same 1,000 train-set observations using the default parameters of Anchors. We note that Anchors did not finish its execution in 20 out of 3000 instances, and our algorithms timed out in 31 out of 6,000 instances (3 global models $\times$ 2 algorithms, namely `BinMinSetCover` with `BinMaxSupport`, $\times$ 1,000 observations).

We observe that there is a significantly higher number of inconsistencies (top row) in Anchors in comparison with our algorithms, which is zero (by design) in the training set and is very low in the test set. In terms of the support size (middle row), Anchors performs better than BinMinSetCover and worse than BinMaxSupport. Our algorithms outperform Anchors in terms of both sparsity (bottom row, right) and runtime (bottom row, left).

Finally, we note that while the parameters of Anchors could potentially be configured to expand its search strategy and to return rules with a higher precision (and consequently better consistency), its running time is already longer than ours, which suggests that perhaps more significant design changes would need to be made to Anchors in order to compete along this dimension.

5. Discussion and Future Directions

We next discuss a few aspects related to the practical application and limitations of our approach, as well as future directions for research.

- **Running time**. While the algorithms ran sufficiently fast in our experiments, in general, the combinatorial nature of the problem could indicate that for larger datasets, longer running times would be needed to generate rules. This could potentially be

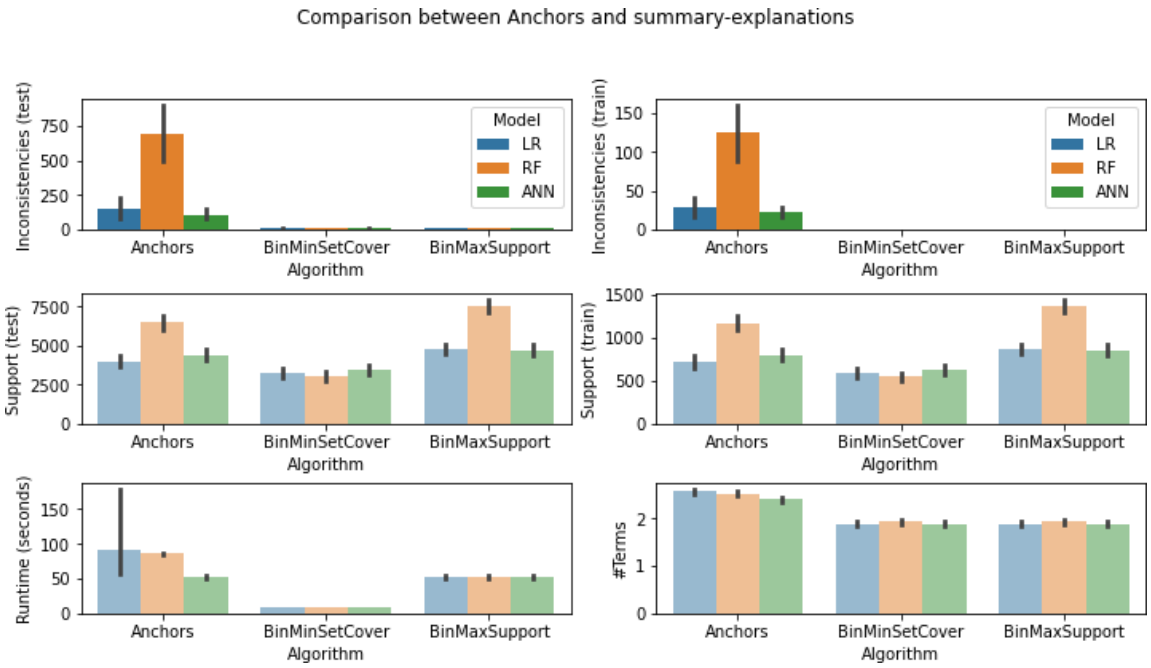

Figure 19: A comparison between Anchor-explanations and summary-explanations (Census dataset). Most importantly, fewer inconsistencies (see top row) is better. Larger support is better (middle row) and smaller runtime and number of terms is better (bottom row).

exacerbated by our reliance on a commercial-grade solver (Gurobi) that is free for academic use but might hinder a widespread adoption of the algorithms.

On the other hand, our algorithms are closely related to the minimal set cover problem, which is a well understood problem. Clever data structures and preprocessing are two approaches that could potentially improve the scalability of the algorithms.

- **Privacy concerns**. The explanations generated by the algorithms provide aggregated information about the users and the model (but not on the true labels, which are not used by the algorithms). This aggregated information could potentially be used for inference about the distribution of the data and the model, which a company may not be interested in revealing. To mitigate this issue, one could consider discretizing the reported values of the support to impede such inferences (e.g., report that the support of a rule is low (under 10), medium (over 10), or high (over 100)). However, many scholars advocate for transparency, arguing that releasing information about how models work is not necessarily a problem but even a desirable model property for increasing trust.

- **Misinterpretation of results**. While our rules are globally consistent, it is important to recognize that they do not imply a causal relation. The rules do not imply that the global model is using particular features that are included in the summary-

explanation, only that the summary-explanation reliably describes a pattern that exists in the data, and that other patterns exist simultaneously without there being a contradiction.

- **Interpretable features**. An implicit assumption we make is that the features are interpretable. This is true in most tabular prediction problems where data are structured. However, in other cases, the data may contain features that are not interpretable or unstructured. Generating summary-explanations directly on such data would not lead to meaningful explanations.

  One possible direction to address this issue is to run the summary-explanation algorithms with a subset of the original feature set that are interpretable. This could potentially work well if the global model's decision boundary is smooth in the subspace of features selected.

- **Stability of explanations.** For a given observation, there can exist many globally consistent rules that are functions of different features, particularly when features are correlated. This phenomenon, which could occur with any explanation method, could result in situations where, over time, users receive explanations that vary even though little if nothing has changed on their part.

  If one wanted to have summary-explanations that remain relatively stable over time and across observations, one could potentially rank the summary-explanations in a way that encourages consistency in feature usage. For example, we could do this by generating summary-explanations for the entire training dataset and identifying which features appear more frequently and prioritizing such features (e.g., through the objective function).

  In general, however, we may want to find a way to present all equally good rules to the user, which would be a fruitful direction for future work. Enumerating all such rules (which could be done with our approach by adding cutting planes to remove previously discovered rules) may be overwhelming to the user without a sophisticated user interface.

  Interestingly, it may be useful to create rules that depend on different features than those used in the model. For instance, if race features are not used in the model, but if a rule arises such as "all individuals with race X and no credit history were denied a loan," it might suggest that a change in the global model is warranted.

- **ϵ-inconsistency.** One could try to relax the strict requirement of global consistency with the hope that the resulting rules would have higher support and thus generalize better to new observations. While this could be done technically by modifying the optimization problem (Formulation (3)), such an attempt could hinder interpretability by undermining the fundamental premise of the proposed approach which seeks to provide clarity and trust through consistency. Note that in contrast to typical applications of ML algorithms which apply directly to the labels y of the data, in generating local summary-explanations, one uses a *model*, h^{global}, for labeling the data. This results in training and test data that are much smoother and are not subject to noise, which reduces the need to protect oneself against noisy data, and is more likely

to result in better generalization. Hence ϵ-inconsistency is less likely to be needed in practice, compared to global consistency.

Also, it is easily possible that the observations that are the most important and difficult to explain (those near the decision boundary) are exactly the points where an ϵ-inconsistent explanation is likely to make a mistake.

One can see in the summary-explanations for the FICO data that the rules tend to be sparse even with the requirement of global consistency, so it is not clear that any benefit would be gained from allowing inconsistencies.

If one did find that no sparse rule could be constructed that has sufficient support, this could be an indicator of an underlying problem with the smoothness of the global model that warrants investigation, which would help us troubleshoot it, as we discuss next.

- **Troubleshooting the global model with summary-explanations.** If there are issues with the summary-explanations, for instance if all summary-explanations for an observation cover only that one observation, there is probably an issue with the global model's lack of smoothness, which may be indicative of overfitting. Also, if the rules do not make intuitive sense (e.g., if a rule indicates that high-risk prediction is made when a certain feature is too small, but we would expect it to be the opposite), it would be worthwhile to go back and examine the global model. How the rules might be used in other ways to troubleshoot the global model could be a useful direction for future research.

- **Generalization to new observations (the risk of overfitting).** Typically, if a new observation is presented that is not in the training set, we could simply generate a new rule for it that is globally consistent, using the algorithms above. However, if we add more data without regenerating the rules, the rules we have previously generated may no longer be globally consistent with respect to the new data. In that case, we are interested in generalization of the summary-explanations to new data points. While global consistency is achieved with respect to the training data, it may not be satisfied with respect to new observations. However, the likelihood of such an event decreases as the number of observations in the training data increases. We observe this behavior in Section 4.4, and it also follows from classic results in statistical learning theory: for a finite hypothesis class (in the case of binary datasets there are $2^{|P|}$ hypotheses corresponding to subsets of features), the deviation from the empirical loss (likelihood of inconsistency) is bounded by a sublinear function of the number of features and is inversely proportional to the number of observations. The same bound also holds for the size of the support. Moreover, while perfect generalization cannot be guaranteed, the summary-explanation remains truthful when stated with respect to previously observed data. We believe that similar generalizations could potentially be obtained for continuous datasets as well (possibly by utilizing the VC-dimension of box-classifiers).

- **Enhancing the quality of rules by adding extrapolated data, or checking explicitly for violations of global consistency.** Let us assume we have access to

query the global model to provide a value for $h^{\text{global}}(\mathbf{x})$ for any given feature vector $\mathbf{x}$; note that this assumption was not made previously in the paper. With query access to h^{global}, we can use this access to gain better guarantees on global consistency.

The feature data used for the summary-explanation algorithms do not all need to arise from the training distribution. The data used for constructing rules can be sampled from a distribution, or chosen any other way. (The labels come from the global model h^{global}, which we could query for any given $\mathbf{x}$.) Sampling many points near $\mathbf{x}_e$ (See Figure 20) would help ensure that the generated summary-explanation is globally consistent across all data that falls into it. (When doing this, one should calculate the support from the actual training data, or importance sampling weights, rather than the simulated points though. This is because it does not make sense to us to suggest that "All 400 observations, 350 of which were simulated, had a rejected loan application.")

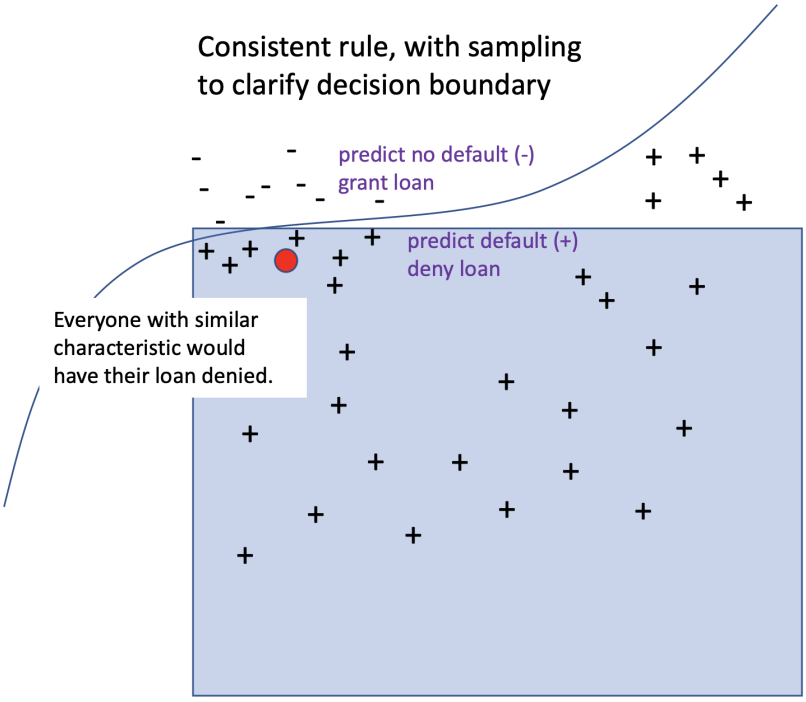

Figure 20: Rule created with extra sampled data. Additional samples were drawn near the decision boundary of the global model and the edge of the rule to help with generalization. Here, $\mathbf{x}_e$ is in red.

If the global function is known or easy to work with, one could check explicitly whether the rule is globally consistent by minimizing (respectively, maximizing) the global function inside of the rule to see whether any point inside the rule violates global consistency. Let us consider the standard form for machine learning models, where h^{global} is a threshold model for a real-valued model f^m, so that $h^{\text{global}}(x) = 1$ if $f^m(x) \geq \theta$ for some threshold θ. Then, if $h^{\text{global}}(\mathbf{x}_e) = 1$, we check for violations in

global consistency of $h^{\text{expln}(e)}$ by solving:

$$\theta^e := \min_x f^m(x) \text{ s.t. } h^{\text{expln}(e)}(x) = h^{\text{global}}(\mathbf{x}_e).$$

If the solution of this obeys $\theta^e \geq \theta$, then global consistency holds for $h^{\text{expln}(e)}$. (Symmetrically, if $h^{\text{global}}(\mathbf{x}_e)$=0, the max becomes a min, and we check whether $\theta^e < \theta$.) In many cases this optimization problem is not difficult:

- If the global model is piecewise constant, such as a boosted decision tree or random forest, one could examine whether each piece (here an intersection of leaves of the trees) overlaps with the proposed rule. For each such overlapping region, we can check whether its predicted label agrees with the rule's label. If all such regions agree, then the rule is globally consistent.
- If the global model f^m is a linear model, maximizing (or minimizing) the model within the boundaries of a proposed rule is a linear program. The solution of this linear program will tell us directly whether global consistency holds.
- If the global model is a neural network, one could try to use neural network optimization approaches to find extreme values of the global model within the boundaries of the rule.

These explicit checks can be useful to ensure that new data points arising within a rule would maintain consistency. Indeed some recent work which builds on our work makes use of sampling to improve consistency in the context of counterfactual explanations (Geng et al. 2022, Kanamori et al. 2022).

- **Summarizing global models.** Multiple summary-explanations that cover most or all of the predictions can be generated to precisely represent (or approximate) global models in a consistent and relatively compact way. This collection could serve as a useful summary for the predictions of a global model.

- **Model obfuscation.** Our approach can also be used for an interpretable model obfuscation, that is, scenarios in which a model designer does not wish to share a model, but is still interested in explaining its predictions. To this end, one could generate a large collection of summary-explanations for a dataset, and then when a new observation arrives, make a prediction using one of the existing summary-explanations (which are globally-consistent), or generate an explanation on-the-fly if the observation is not covered by any of the summary-explanations.

- **Counterfactual explanations.** One could generate recommendations for what the user could do to increase the odds of being evaluated differently by the model. For example, in the spirit of Section 3, a similar optimization problem could be formulated to generate the following summary-explanation: "all 500 individuals who have credit history that is greater than 5 years and whose average standing balance is smaller than \$5000 were predicted to pay back their loan on time." This could focus the user on particular aspects of his or her application that can be changed to potentially reverse the prediction made by the model (though one would need to check that this reversal would be possible with the global model itself).

- **Case-based explanations.** Given a summary-explanation, one could display past observations that satisfy the rule to illustrate that similar predictions are being made for other cases in the data.

6. Conclusions

This work studies summary-explanations for predictions made by ML models that are globally-consistent, thus avoiding scenarios where an explanation offered to one customer does not apply to other customers. We developed algorithms for generating such summary-explanations and showed that while these are theoretically challenging optimization problems, numerical experiments on real-world datasets suggest that these problems can be solved in seconds. Our approach can be used for summarizing predictions from black boxes but also for summarizing patterns from machine learning models that are inherently interpretable, but that are not as concise as a single rule from our summary-explanation algorithms. The foundation of all of these summary-explanation problems relies upon being able to compute them efficiently. The approaches proposed here pave the way not just for more research in this area, but for direct usage in practice, enabling more informed decision-making.

Acknowledgments

We thank the action editor David Jensen and the anonymous members of the review team for their constructive comments that helped improving this study.

Appendix A. Additional Plots and Figures

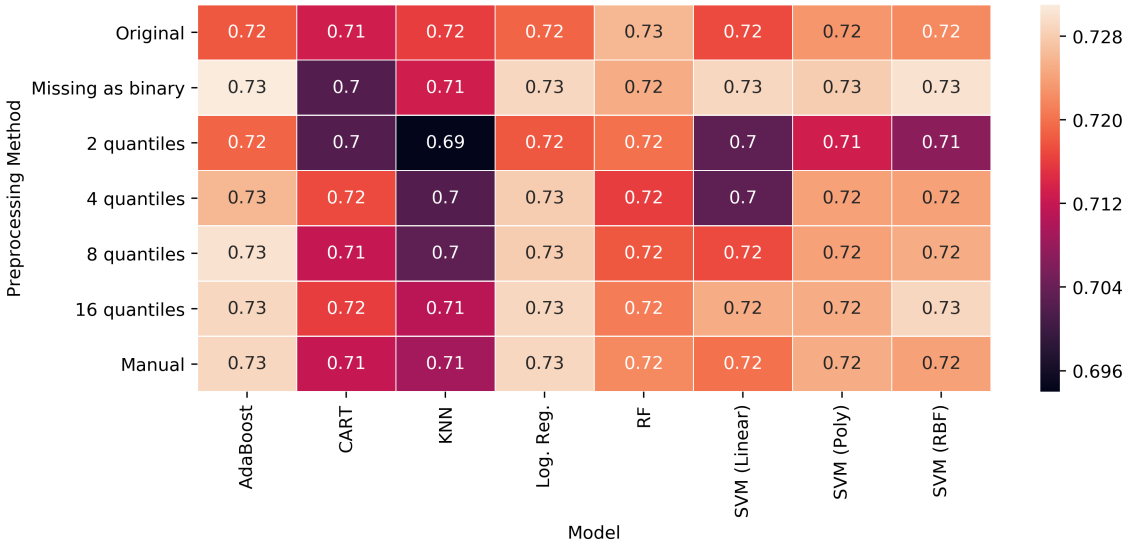

Figure 21: Test data accuracy of global models using various discretization methods. All global models tend to perform similarly. No summary-explanations are involved in this analysis yet.

Original	2 quantiles	4 quantiles	8 quantiles	16 quantiles	Missing as binary	Manual
0.719	0.709	0.718	0.72	0.723	0.723	0.723

Figure 22: Average accuracy of different preprocessing methods (averaged over ML models)

KNN	CART	SVM (Linear)	RF	SVM (RBF)	SVM (Poly)	AdaBoost	Log. Reg.
0.707	0.711	0.716	0.722	0.722	0.724	0.726	0.726

Figure 23: Average accuracy of different ML models (averaged over preprocessing methods)

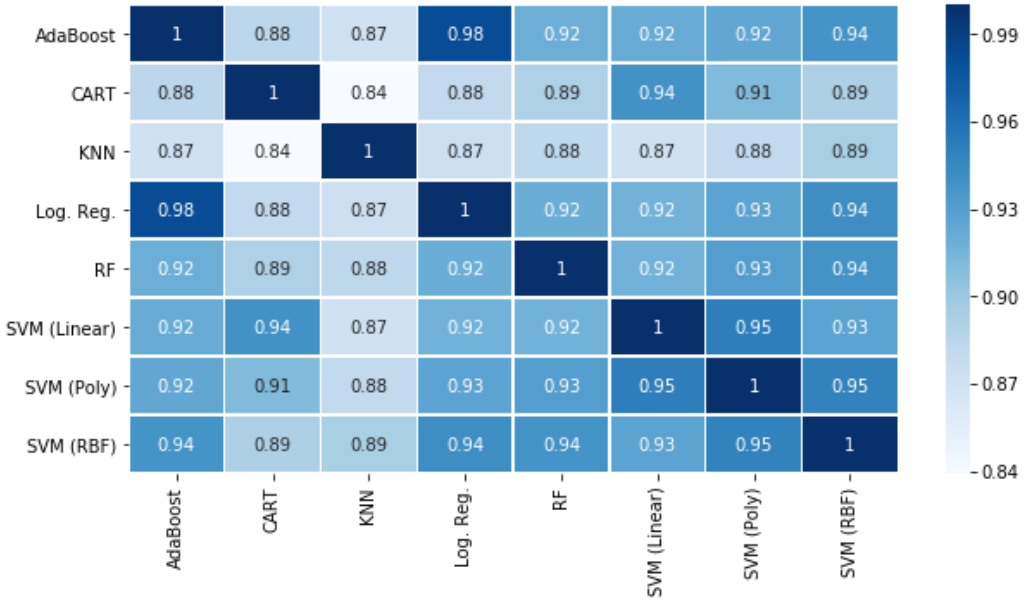

Figure 24: Similarity in predictions on test data of various models (Manual discretization)

Appendix B. The FICO Dataset

Table B describes the features in the FICO dataset (FICO 2018).

Feature	Description
ExternalRiskEstimate	Consolidated version of risk markers
MSinceOldestTradeOpen	Months Since Oldest Trade Open
MSinceMostRecentTradeOpen	Months Since Most Recent Trade Open
AverageMInFile	Average Months in File
NumSatisfactoryTrades	Number Satisfactory Trades
NumTrades60Ever2DerogPubRec	Number Trades 60+ Ever
NumTrades90Ever2DerogPubRec	Number Trades 90+ Ever
PercentTradesNeverDelq	Percent Trades Never Delinquent
MSinceMostRecentDelq	Months Since Most Recent Delinquency
MaxDelq2PublicRecLast12M	Max Delq/Public Records Last 12 Months. See tab "MaxDelq" for each category
MaxDelqEver	Max Delinquency Ever. See tab "MaxDelq" for each category
NumTotalTrades	Number of Total Trades (total number of credit accounts)
NumTradesOpeninLast12M	Number of Trades Open in Last 12 Months
PercentInstallTrades	Percent Installment Trades
MSinceMostRecentInqexcl7days	Months Since Most Recent Inq excl 7days
NumInqLast6M	Number of Inq Last 6 Months
NumInqLast6Mexcl7days	Number of Inq Last 6 Months excl 7days. Excluding the last 7 days removes inquiries that are likely due to price comparision shopping.
NetFractionRevolvingBurden	Net Fraction Revolving Burden. This is revolving balance divided by credit limit
NetFractionInstallBurden	Net Fraction Installment Burden. This is installment balance divided by original loan amount
NumRevolvingTradesWBalance	Number Revolving Trades with Balance
NumInstallTradesWBalance	Number Installment Trades with Balance
NumBank2NatlTradesWHighUtilization	Number Bank/Natl Trades w high utilization ratio
PercentTradesWBalance	Percent Trades with Balance

Table 1: Description of the FICO dataset (source: FICO 2018).

Appendix C. Programming interface

A Python interface for our algorithms is publicly available online (link). Figure 25 illustrates a basic working example.

```
from ConsistentLocalRules import *
explainer = ConsistentRulesExplainer(X, Y_global)
df = explainer.explain(X_e, Y_e, objective='SPARSITY',
                       n_explanations=1, max_features=999999, max_runtime=1)
```

Figure 25: A basic working example for the programming interface.

The first line of code imports the code, which is contained in a single Python file (ConsistentLocalRules.py). In the second line we create an *explainer* which is initalized using a

data matrix X and the predictions of some global model Y_global. In the third line, we generate summary-explanations for a collection of observations X_e whose predictions by the global model are Y_e. This returns a dataframe with multiple summary-explanations which are consistent with X,Y_global for each observation (x_e,y_e) in (X_e,Y_e). The parameters of the function *explain( )* are:

- objective : specifies whether the algorithm should aim to find rules that have the smallest possible number of features (SPARSITY) or rules that are satisfied by the largest number of observations (SUPPORT).
- n_explanations : the maximal number of returned explanations.
- max_features : the maximal number of features that constitute each explanation.
- max_runtime : the maximal running time per explanation (in seconds; a mixed integer programming solver is used to generate summary-explanations and this parameter sets a limit on its runtime).

Sample output. Figure 26 illustrates the output of the code for a single observation when the objective is to optimize sparsity (objective=SPARSITY) while limiting the number of resulting summary-explanations to 3 (n_explanations=3). The resulting dataframe consists of 3 rows, one per summary-explanation, and the columns provide information about each summary-explanation, such as the rule, its support, sparsity, running time, and the algorithm used to generate the explanation. Here, the algorithm is chosen automatically, based on whether all input features are binary.

#Observation	#Explanation	Rule	Prediction	Support	#Features	Runtime	Algorithm
0	0	MSinceMostRecentDelq>=2.00, ExternalRiskEstima...	1	204	2	5.3	ContMinSetCover
0	1	NumTrades60Ever2DerogPubRec>=3.00, ExternalRis...	1	85	2	5.3	ContMinSetCover
0	2	ExternalRiskEstimate<=55.00, NumTrades90Ever2D...	1	730	2	5.3	ContMinSetCover

Figure 26: Example of summary-explanations returned by the programming interface.

Dependencies. The code was developed in Python v3.7 and makes use of the following libraries: pandas, matplotlib, numpy, scikit learn, pulp (Mitchell et al. 2011), and Gurobi (Gurobi 2014).

References

Vijay Arya, Rachel KE Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C Hoffman, Stephanie Houde, Q Vera Liao, Ronny Luss, Aleksandra Mojsilovic, et al. Ai explainability 360: An extensible toolkit for understanding data and machine learning models. *J. Mach. Learn. Res.*, 21(130):1–6, 2020.

Bart Baesens, Rudy Setiono, Christophe Mues, and Jan Vanthienen. Using neural network rule extraction and decision tables for credit-risk evaluation. *Management Science*, 49(3):312–329, 2003.

Korte Bernhard and J Vygen. Combinatorial optimization: Theory and algorithms. *Springer, Third Edition, 2005.*, 2008.

Nadia Burkart and Marco F Huber. A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, 70:245–317, 2021.

Chaofan Chen, Kangcheng Lin, Cynthia Rudin, Yaron Shaposhnik, Sijia Wang, and Tong Wang. A holistic approach to interpretability in financial lending: Models, visualizations, and summary-explanations. *Decision Support Systems*, 152:113647, 2022.

Danielle Citron. (Un)fairness of risk scores in criminal sentencing. *Forbes, Tech section*, July 2016.

Amit Dhurandhar, Tejaswini Pedapati, Avinash Balakrishnan, Pin-Yu Chen, Karthikeyan Shanmugam, and Ruchir Puri. Model agnostic contrastive explanations for structured data. *arXiv preprint arXiv:1906.00117*, 2019.

Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. Explainable artificial intelligence: A survey. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 0210–0215. IEEE, 2018.

FICO. Explainable machine learning challenge, 2018. URL `https://community.fico.com/s/explainable-machine-learning-challenge`. Accessed: 2018-11-02.

FICO. FICO announces winners of inaugural xML challenge — FICO, 2019. URL `https://www.fico.com/en/newsroom/fico-announces-winners-of-inaugural-xml-challenge`. Accessed: 2019-5-21.

Zixuan Geng, Maximilian Schleich, and Dan Suciu. Computing rule-based explanations by leveraging counterfactuals. *arXiv preprint arXiv:2210.17071*, 2022.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. Local rule-based explanations of black box decision systems. *arXiv preprint arXiv:1805.10820*, 2018a.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018b.

Gurobi. Gurobi optimizer reference manual, 2015. *URL: http://www. gurobi. com*, 2014.

Vivian Ho. Miscalculated score said to be behind release of alleged twin peaks killer. *SFGate (San Francisco Chronicle)*, August 2017.

Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva. Abduction-based explanations for machine learning models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1511–1519, 2019.

Yacine Izza, Alexey Ignatiev, and Joao Marques-Silva. On explaining decision trees. *arXiv preprint arXiv:2010.11034*, 2020.

Kentaro Kanamori, Takuya Takagi, Ken Kobayashi, and Hiroki Arimura. DACE: Distribution-aware counterfactual explanation by mixed-integer linear optimization. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 2855–2862, 2020.

Kentaro Kanamori, Takuya Takagi, Ken Kobayashi, and Yuichi Ike. Counterfactual explanation trees: Transparent and consistent actionable recourse with decision trees. In *International Conference on Artificial Intelligence and Statistics*, pages 1846–1870. PMLR, 2022.

Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proc. 31st Conference on Neural Information Processing Systems (NIPS)*, 2017.

Elizabeth Mannshardt and Liz Naess. Air quality in the USA. *Significance*, 2018.

Joao Marques-Silva, Thomas Gerspacher, Martin C Cooper, Alexey Ignatiev, and Nina Narodytska. Explaining Naive Bayes and other linear classifiers with polynomial time and delay. In *34th Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Chris Meek, Bo Thiesson, and David Heckerman. US Census data (1990) data set, 2022. URL `https://archive.ics.uci.edu/ml/datasets/US+Census+Data+(1990)`. Accessed: 2022-05-26.

Stuart Mitchell, Michael OSullivan, and Iain Dunning. PuLP: a linear programming toolkit for python. *The University of Auckland, Auckland, New Zealand*, page 65, 2011.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

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.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Ribana Roscher, Bastian Bohn, Marco F Duarte, and Jochen Garcke. Explainable machine learning for scientific insights and discoveries. *IEEE Access*, 8:42200–42216, 2020.

Cynthia Rudin and Yaron Shaposhnik. Globally-consistent rule-based summary-explanations for machine learning models: Application to credit-risk evaluation. *Available at SSRN 3395422*, 2019.

Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, 16:1 – 85, 2022.

Sharath M Shankaranarayana and Davor Runje. ALIME: Autoencoder based approach for local interpretability. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 454–463. Springer, 2019.

Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186, 2020.

Erico Tjoa and Cuntai Guan. A survey on explainable artificial intelligence (XAI): Toward medical XAI. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

Arnaud Van Looveren and Janis Klaise. Interpretable counterfactual explanations guided by prototypes. *Machine Learning and Knowledge Discovery in Databases, Lecture Notes in Computer Science (ECML-PKDD)*, 12976:650–665, 2021.

Vijay V Vazirani. *Approximation Algorithms*. Springer Science & Business Media, 2013.

Rebecca Wexler. Code of silence: How private companies hide flaws in the software that governments use to decide who goes to prison and who gets out. *Washington Monthly*, June/July/August 2017.

Adam White and Artur d'Avila Garcez. Measurable counterfactual local explanations for any classifier. *arXiv preprint arXiv:1908.03020*, 2019.

David P Williamson and David B Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.

Jinsung Yoon, Sercan O Arik, and Tomas Pfister. RL-LIM: Reinforcement learning-based locally interpretable modeling. *arXiv preprint arXiv:1909.12367 (ICLR 2019)*, 2019.

Muhammad Rehman Zafar and Naimul Khan. Deterministic local interpretable model-agnostic explanations for stable explainability. *Machine Learning and Knowledge Extraction*, 3(3):525–541, 2021.