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

Cynthia Rudin
Departments of Computer Science, Electrical and Computer Engineering, and Statistical Science, Duke University, Durham NC
cynthia@cs.duke.edu

Yaron Shaposhnik
Simon School of Business, University of Rochester, Rochester NY
yaron.shaposhnik@simon.rochester.edu

We develop a method for interpreting 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 real-world data from FICO and demonstrate that our approach effectively produces sparse summary-explanations of these models in short period of time. Our approach is model-agnostic (that is, can be used to interpret any predictive model), and solves a minimum set cover problem to construct its summaries.

Key words: Interpretability, Local models, Association rules, Explanations, Machine learning, Credit-risk.
History: Submitted on May 28, 2019

1. Introduction

As the use of machine learning (ML) 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 ML 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. This has led to a subfield of

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 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 reality, the explanation approximates the global model, but does not actually explain it.

Beyond issues of fidelity, the terminology "explanation" is misleading. It is possible that the explanation's important features are completely different than those of the global model. For instance, race is correlated with age and criminal history in criminal recidivism prediction data, so the explanation "race = black" is actually a reasonable "explanation" of a predictive model, despite the fact that these criminological models do not depend explicitly on race. This is precisely the error made by Propublica (Larson et al. 2016, Angwin et al. 2016) in their accusation about the COMPAS model (Brennan et al. 2009) used in the justice system, as pointed out by criminologists (Flores et al. 2016).

There are modeling choices we can make in order to avoid mistakes like those listed above. For example, we can work with interpretable models, rather than proprietary models or complicated black-box models. Alternatively, we propose to create summary-explanations that are globally consistent with the global model, for all observations that it 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.) Thus, for instance, 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.

Contributions. This paper makes the following contributions. First, we develop a new method for interpreting predictions made by arbitrary ML models, which we refer to as summary-explanations. Our summary-explanations produce sparse rules that locally summarize, and are consistent with

predictions from, a globally-interpretable model, which is how we envision this technique being used in practice. Second, we design algorithms that solve generalizations of the minimum set cover problem that finds optimal sparse rules. These rules are conditioned on a given observation and on its predicted class. They are designed to have large support (coverage) in the dataset. 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 used to interpret predictions made by models in short time, 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].

Related work. Our work is most closely related to the literature on local explanations, which aims to explain specific predictions made by (potentially black-box) global models. A class of algorithms (for instance, the popular LIME algorithm Ribeiro et al. (2016)) randomly perturb the explained observation and use the perturbations to train a local model which 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. These same issues arise when using interpretable machine learning techniques to approximate global models (see for example, Setiono and Liu (1996), Craven and Shavlik (1996)). In Ribeiro et al. (2018), the authors create local rule-based models (unlike the linear models used in Ribeiro et al. (2016)) which guarantee a minimal degree of accuracy around the explained observations. Guidotti et al. (2018) propose to use genetic algorithms to generate samples near the explained observation, which they use to train local models.

Our work also relates to a body of work that studies rule-based prediction models (e.g., Su et al. (2016) and the references therein) and rule extraction methods (see for example the work of Baesens et al. (2003) for an application in the context of credit risk evaluation). However, to the best of our knowledge, our work is first to introduce the concept of global consistency and enforce it as a requirement for explanations. Similar to Ribeiro et al. (2016) and Guidotti et al. (2018), our approach is model-agnostic, but in contrast, does not use sampling and does not require access to the global model beyond the labeling of the dataset and the explained observation.

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

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 for interpreting predictions about credit risk. We conclude and discuss future research directions in Section 5.

2. Problem Formulation

Consider a general binary classification problem defined over a dataset $\mathcal{D} = (\mathbf{X}, \mathbf{y})$, where $\mathbf{X} \in \mathbb{R}^{|N|\times|P|}$ is a data-matrix and $\mathbf{y} \in \mathbb{R}^{|N|}$ is a vector of binary labels. We use N and P to respectively denote the set of observations and features. In addition, we assume access to $\mathbf{y}^m \in \{0,1\}^{|N|}$, the labels predicted by a global, potentially black-box, model h^m; that is, for each observation $i \in N$: $y_i^m = h^m(\mathbf{x}_i)$. We do not make any assumptions about the nature of h^m, nor of having access to h^m 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 $\mathbf{X}_{:,p}$.

Definition 1. Let $\mathbf{x}_e, y_e^m$ denote a new observation and the respective label predicted by h^m. We say that the model h^e provides a *globally consistent local summary-explanation* to observation e with respect to the model h^m, class y_e^m, and the dataset $\mathcal{D}$ if the following conditions are met:

1. (Relevance) $h^e(\mathbf{x}_e) = y_e^m$;
2. (Global consistency) for every observation $i \in N$, if $h^e(\mathbf{x}_i) = y_e^m$ then $h^m(\mathbf{x}_i) = y_e^m$ (or equivalently, if $h^m(\mathbf{x}_i) = 1 - y_e^m$ then $h^e(\mathbf{x}_i) = 1 - y_e^m$);
3. (Interpretability) the model h^e belongs to a class of interpretable models $\mathcal{H}$.

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

Condition 1 asserts that the summary-explanation is relevant for explaining the prediction y_e^m by the model h^m. Condition 2 implies that h^e provides sufficient conditions for predicting y_e^m that are consistent with past predictions. That is, if the local model h^e predicts y_e^m for a particular observation in the dataset, it must be the case that the global model h^m also predicts y_e^m for the same observation. This is in sharp contrast to alternative approaches which generate summary-explanations by approximating h^m in the vicinity of $\mathbf{x}_e$, which could result in providing explanations that can be contradicted using observations from the dataset. Over the long run, the existence of such contradictory explanations could negatively impact the trust that customers or users have in the system and its underlying model. In contrast, by design, our summary-explanations cannot be contradicted using past data. Finally, Condition 3 guarantees that the local model can be intuitively understood by the stakeholders. This is subjective and context dependent but nonetheless critical for the purpose of interpreting how a global model works.

We use two measures to assess the quality of a summary-explanation h^e:

- $\Gamma_c(h^e)$: Complexity – a measure for 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.
- $\Gamma_s(h^e)$: Support – the number of observations in the dataset that satisfy the rule, that is,

$$\Gamma_s(h^e) = |\{i \in N : h^e(\mathbf{x}_i) = y_e^m\}|.$$

  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. Note that under the requirement for global consistency, maximizing support is equivalent to maximizing the fidelity of the summary-explanation to the global model (that is, the accuracy of the predictions made by the local model with respect to the global model).

Ideal rules have low complexity and large support. However, the dataset may not be homogeneous and some observations may be more difficult to interpret than others, in which case human intervention and judgment should be exercised.

Using this notation, we can formulate the problem for finding a globally consistent summary-explanation for the point $(\mathbf{x}_e, y_e^m)$:

$$\begin{array}{lll} \max_{h^e \in \mathcal{H}} & w_s \cdot \Gamma_s(h^e) - w_c \cdot \Gamma_c(h^e) & \\ \text{s.t.} & h^e(\mathbf{x}_e) = y_e^m & \\ & h^e(\mathbf{x}_i) = 1 - y_e^m & \forall i \in N : y_i^m = 1 - y_e^m \,. \\ & \Gamma_c(h^e) \leq p_c & \end{array} \tag{1}$$

The coefficients w_s and w_c are user-defined non-negative weights that balance the desired support and interpretability of the resulting rule, while the bound p_c ensures sufficient degree of interpretability. The three types of constraints capture the three conditions of globally consistent local summary-explanations: relevance, global consistency, and interpretability, respectively.

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 if it can be written as a conjunction of one-dimensional step functions with a single step:

$$h_{R,\boldsymbol{\tau},y}(\mathbf{x}) = \begin{cases} y & \bigwedge_{p \in R} \mathbb{1}\left[x_p \geq \tau_p\right], \text{ where } R \subseteq P \\ 1 - y & \text{otherwise} \end{cases} . \tag{2}$$

Note that any rule-based model can provide a globally consistent summary-explanation 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 that 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 $\mathbf{X}$. Note that for a binary dataset $\mathbf{X}$, one could simply add $\mathbf{1}_{|N|\times|P|} - \mathbf{X}$ 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).

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 $w_c = 1$ and $w_s = 10$, 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:

$$\begin{array}{ll} \max_{R,\tau} & w_s \cdot \Gamma_s(h_{R,\tau,y_e^m}) - w_c \cdot |R| \\ \text{s.t.} & \forall p \in R : x_{e,p} \geq \tau_p \\ & \exists p \in R : x_{i,p} < \tau_p \qquad\qquad \forall i \in N : y_i^m = 1 - y_e^m. \\ & |R| \leq p_c \end{array} \tag{3}$$

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. Computational Complexity

We show that `OptConsistentRule` generalizes the *Minimum Set Cover* Problem (`MinSetCover`), which is known to be *NP*-hard (Bernhard and Vygen 2008).

LEMMA 1. *`OptConsistentRule` is NP-hard.*

Proof. Our proof is based on a reduction to the unweighted `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 $\cup_{p\in R} S_p = E$. The unweighted `MinSetCover` can be written as a subset selection problem

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

which is equivalent to

$$\begin{array}{ll} \min_{R\subseteq\{1,\ldots,\rho\}} & |R| \\ \text{s.t.} & \exists p \in R : \mathbb{1}[i \in E \setminus S_p] = 0 \qquad \forall i \in \{1, \ldots, n\}\,. \end{array} \tag{4}$$

For each instance of `MinSetCover`, we construct an instance of `OptConsistentRule` where the data matrix $\mathbf{X}$ consists of n observations (corresponding to elements) and ρ binary features (corresponding to sets). 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, $e_i \notin S_p$. All labels are set to $+1$. The observation we wish to explain is intialized as $\mathbf{x}_e = \mathbf{1}$ and its label is set to 0. The other coefficients 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 (a threshold value of $\tau_p > 1$ or $\tau_p \leq 0$ is not feasible for any feature $p \in R$, 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:

$$\begin{array}{lll} \min_{R\subseteq\{1,\ldots,\rho\}} & |R| & \\ \text{s.t.} & \forall p \in R : x_{e,p} \geq 1 & \\ & \exists p \in R : x_{i,p} = 0 & \forall i \in \{1,\ldots,n\} \\ & |R| \leq \rho. & \end{array} \tag{5}$$

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 second constraint $R \leq \rho$ trivially holds). Moreover, by construction $x_{i,p} = 0$ if and only if $e_i \in S_p$. Therefore, the remaining second constraint in Equation (5) is equivalent to the constraint in Equation (4), and therefore the formulations are equivalent. This means that a polynomial time solution to `OptConsistentRule` can be used to solve in polynomial time `MinSetCover` and therefore `OptConsistentRule` is *NP*-hard. Q.E.D.

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 Lemma 1, 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 formulation to

$$\begin{array}{lll} \min_R & -w_s \cdot \Gamma_s(h_{R,\mathbf{1},y_e^m}) + w_c \cdot |R| & \\ \text{s.t.} & \forall p \in R : x_{e,p} \geq 1 & \\ & \exists p \in R : x_{i,p} = 0 & \forall i \in N : y_i^m = 1 - y_e^m \\ & |R| \leq p_c. & \end{array} \tag{6}$$

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, which provides a feasible lower-bound for the simplest summary-explanation. This is the smallest value of p_c in Equation 6 for which a feasible solution exists.

The problem of minimizing sparsity is a special case of Equation (6) where $w_s = 0, w_c = 1$, and $p_c = |P|$. The respective optimization problem can be written as

$$\begin{array}{lll} \min_{R\subseteq\{1,\ldots,m\}} & |R| & \\ \text{s.t.} & \sum_{p\in\{1,\ldots,m\}} \mathbb{1}[i\in R]\cdot\mathbb{1}[x_{e,p}=1] \geq |R| & \\ & \sum_{p\in\{1,\ldots,m\}} \mathbb{1}[i\in R]\cdot\mathbb{1}[x_{i,p}=0] \geq 1 & \forall i : y_i^m = 1 - y_e^m. \end{array} \tag{7}$$

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 (6).

Let $P^e \triangleq \{p : x_{e,p} = 1\}$, denote the set of features that are equal to 1 in observation e. We can write the following equivalent IP:

$$\begin{array}{lll} \min_{\{b_p:p\in P^e\}} & \sum_{p\in P^e} b_p & \\ \text{s.t.} & \sum_{p\in P^e} b_p \cdot \mathbb{1}[x_{i,p}=0] \geq 1 & \forall i : y_i^m = 1 - y_e^m\,, \\ & b_p \in \{0,1\} & \forall p \in P^e \end{array} \tag{8}$$

where the binary decision variable b_p indicates whether feature p is part of the rule $h_{R,\mathbf{1},y}$ (that is, $b_p = \mathbb{1}[p \in R]$). This 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 (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.

Note that whenever the dataset $\mathcal{D}$ contains 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 feasible whenever there is no observation in $\mathcal{D}$ with identical values to $\mathbf{x}_e$ that is oppositely labeled (setting all $b_p = 1$ provides such feasible solution). This would be impossible anyway when summarizing predictions from models, as the prediction function should have a single value for each observation (otherwise it would not actually be a function).

3.2.2. Algorithm `BinMaxSupport`. We use the Big-M method (Bertsimas and Tsitsiklis 1997) to formulate an IP for the general optimization problem (6). Let $N_e \triangleq \{i : y_i^m = y_e^m\}$ denote the set

of observations that are labeled as y_e^m. We define r_i as a decision variable that indicates whether the rule $h_{R,\mathbf{1},y_e^m}$ predicts y_e^m for observation i. The optimization problem can be written as follows:

$$\begin{array}{lll} \max_{\mathbf{b},\mathbf{r}} & w_s \cdot \sum_{i \in N^e} r_i - w_c \cdot \sum_{p \in P^e} b_p & \\ \text{s.t.} & \sum_{p \in P^e} b_p \cdot \mathbb{1}[x_{i,p} = 0] \geq 1 & \forall i :\in N \setminus N^e \\ & \sum_{p \in P^e} b_p \cdot \mathbb{1}[x_{i,p} = 0] \leq M \cdot (1 - r_i) & \forall i \in N^e \\ & b_p \in \{0,1\} & \forall p \in P^e \\ & r_i \in \{0,1\} & \forall i \in N^e. \\ & \sum_p b_p \leq p_c. & \end{array} \tag{9}$$

To ensure global consistency, the first constraint guarantees that the resulting rule does not apply to any observation in $N \setminus N^e$. The first term in the objective counts the number of observations for which the rule applies (i.e., the support). The second constraint ensures that the resulting rule indeed applies to the counted observations. Note that for all practical purposes the constant M could be set to $|P^e|$.

3.3. Algorithms for Continuous Datasets

We now consider datasets with continuous values. For ease of notation, we make the following assumption:

ASSUMPTION 1. *For each feature $p \in P$, there exists a feature $p^c \in P$ such that $\boldsymbol{X}_{\cdot,p^c} = -\boldsymbol{X}_{\cdot,p}$*

Similarly to Section 3.2, we first focus on sparsity and later proceed to a general algorithm.

3.3.1. Algorithm `ContMinSetCover`. We begin with an observation about rules that attain optimal sparsity.

THEOREM 1. *For any dataset $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$, model h^m, observation $\boldsymbol{x}_e, y_e^m$, and globally consistent rule $h_{R,\boldsymbol{\tau},y_e^m}$, there exists a globally consistent rule $h'_{R',\boldsymbol{\tau}',y_e^m}$ where $R' = R$, $\tau'_p = x_{e,p}$, and whose complexity is equal to or smaller than that of $h_{R,\boldsymbol{\tau},y_e^m}$ (that is, $\Gamma_c(h_{R,\boldsymbol{\tau},y_e^m}) = \Gamma_c(h'_{R',\boldsymbol{\tau}',y_e^m})$).*

Proof. Geometrically, any rule h defines a box polytope that contains $\mathbf{x}_e$ but excludes observations whose labels are $1 - y_e^m$. 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 modifying the threshold values τ does not increase the cardinality Γ_c of the rule, we obtain a new rule with identical cardinality. Q.E.D.

The main insight from Theorem 1 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 to 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 $1 - y_e^m$. This is exactly the problem solved by `BinMinSetCover`. We formulate Algorithm 1 which first constructs a data matrix $\mathbf{X}^e \in \{0,1\}^{|N| \times |P|}$, where $x^e_{i,p} = \mathbb{1}[x_{i,p} \geq x_{e,p}]$ (that is, each entry denotes if including the feature p would exclude observation i). 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 predictions $\mathbf{y}^m, y_e^m$.

Output: globally-consistent rule h_{R,τ,y_e^m}.

1. Compute $\mathbf{X}^e \in \{0,1\}^{|N|\times|P|}$, where $x^e_{i,p} = \mathbb{1}[x_{i,p} \geq x_{e,p}]$.
2. Solve `BinMinSetCover` using $\mathbf{X}^e, \mathbf{y}^m, \mathbf{x}_e, y_e^m$ to compute the subset of features R.
3. Return the rule $h_{R,\mathbf{x}_e,y_e^m}$.

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 base rule with optimal sparsity. For example, a sparse summary explanation for $\mathbf{x}_e = (10, 20, 30, 40), y_e^m = +1$ could be the rule

$$h_{R,\mathbf{x}_e,y_e^m} = x_2 \geq 20 \wedge x_3 \geq 20 \rightarrow +1,$$

   where $R = \{2, 3\}$ and whose support could be 100.
2. Expand these rules by decreasing the thresholds to increase support while maintaining optimal sparsity. For example, expanding the above rule could result in the rule

$$h_{R=\{2,3\},\tau,y_e^m} = x_2 \geq 15 \wedge x_3 \geq 22 \rightarrow +1$$

   whose support could be 150.
3. Repeat the previous steps with other rules whose sparsity is optimal. In the above example these would be other rules where $R = 2$.

We first describe the dynamic programming (DP) formulation for expanding rules and present `ContMaxSupport` in Algorithm 2.

Given a rule $h_{R,\mathbf{x}_e,y_e^m}$, we define the following DP:

- State space: the set of thresholds $\tau \in \mathbb{R}^{|R|}$ that constitute globally-consistent rules to $\mathbf{x}_e$:
  1. $\forall p \in R : \tau_p = x_{e,p}$ or $\exists i \in N : \tau_p = x_{i,p}$,
  2. $\forall p \in R : \tau_p \leq x_{e,p}$, and
  3. $\forall i \in N \setminus N^e, \exists p \in R : x_{i,p} < \tau_p$.

  The first condition ensures that the state space is finite as it is defined using the values in the data matrix $\mathbf{X}$ and $\mathbf{x}_e$.
- Reward function: the support of the corresponding rule, that is, $\Gamma_s(h_{R,\tau,y_e^m})$.
- Bellman's equation:

$$J(\tau) = \max \begin{cases} -\infty & // \; h_{R,\tau,y_e^m} \text{ is not globally consistent} \\ \Gamma_s(h_{R,\tau,y_e^m}) & // \text{ select current state} \\ J(\tau \ominus p) & // \text{ explore the next state .} \end{cases} \tag{10}$$

The operator $\ominus$ in the term $\tau \ominus p$ reduces the value of τ_p to the largest value in $\mathbf{X}_{:,p}$ that is smaller than τ_p (this action cannot be selected when $\tau_p = \min(\mathbf{X}_{:,p})$). Maximizing the support of a rule h_{R,τ,y_e^m} is done by computing $J(\mathbf{x}_e)$.

Algorithm 2 Algorithm `ContMaxSupport`

Input: data matrix $\mathbf{X}$, observation to explain $\mathbf{x}_e$, and predictions $\mathbf{y}^m, y_e^m$.

Output: globally-consistent rule (R, τ, y_e^m).

1. Apply `ContMinSetCover` to extract γ rules with optimal sparsity (e.g., by running `ContMinSetCover` iteratively with additional cuts that prohibit previous solutions).
2. Apply the DP formulation (10) to each of the extracted rules h_{R,τ,y_e^m} to increase their support.
3. Return the expanded rule whose support is maximal.

4. Numerical Experiments

We conduct an elaborate 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 relatively short time that could be used in practice by companies. 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.

4.1. The computational setting

Data: The dataset[2] contains "...an anonymized data on 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 that are interpretable, 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.

Preprocessing: We used the following methods to preprocess the data:

- "Original" – using the dataset as is;
- "Missing as binary" – adding binary features to indicate missing values;
- "x quantiles" – discretizing continuous values to 2,4,8, and 16 equal quantiles of each feature;
- "Manual" – using visualization to manually discretize each continuous feature x_p based on marginal changes in the empirical mean of the labels $\widehat{\mathbb{E}[y]}$ as a function of x_p.

The specific method used in each particular experiment will be clear from the context.

[2] The dataset can be downloaded from FICO (2018).

Predictive models: Summary-explanations were generated for a variety of commonly used predictive models for classification. These 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 Boosing (AdaBoost).

Training and evaluation: We trained the aforementioned models on a random training set consisting of 75% of all observations. We applied hyper-parameter tuning using cross validation on a wide set of parameters to optimize the choice of parameters. Models were then evaluated on the test data (the remaining 25%).

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 machines using 12 processes and a maximal timeout of 1 minute for solving IPs. A total of 394,940 summary-explanations were generated.

4.2. Algorithms for binary datasets

Preprocessing methods. In order to apply the algorithms for binary datasets, we first discretize the features using the different approaches discussed in the previous section. Figures 1, 2, and 3 compare the accuracy of different models using different preprocessing methods. We see that overall, the models are comparable in terms of their accuracy and we therefore simply use the manual encoding of continuous variables as binary features. Note that the first two preprocessing methods in Figure 1 contain continuous features and are provided as benchmarks. 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. 2018). The accuracy level is consistent with other studies on similar datasets (e.g., Baesens et al. (2003)).

Predictive models. Figure 4 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 would act as a robustness test for our algorithms.

Algorithms. We first solve `BinMinSetCover` to find summary-explanations with optimal sparsity (denoted as "Min Features"). Denoting by d the optimal sparsity of the resulting rule, we then solve `BinMaxSupport` 3 times, setting $w_s = 1$, $w_c = 0$, and p_c to $d, d+1$, and $d+2$. That is, we maximize support and gradually relax the restriction about the maximal sparsity of the resulting rules. We denote the latter algorithm as "Max support + d", where $d \in \{0, 1, 2\}$.

We note that 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

Figure 1 **Test data accuracy of models using various discretization methods**

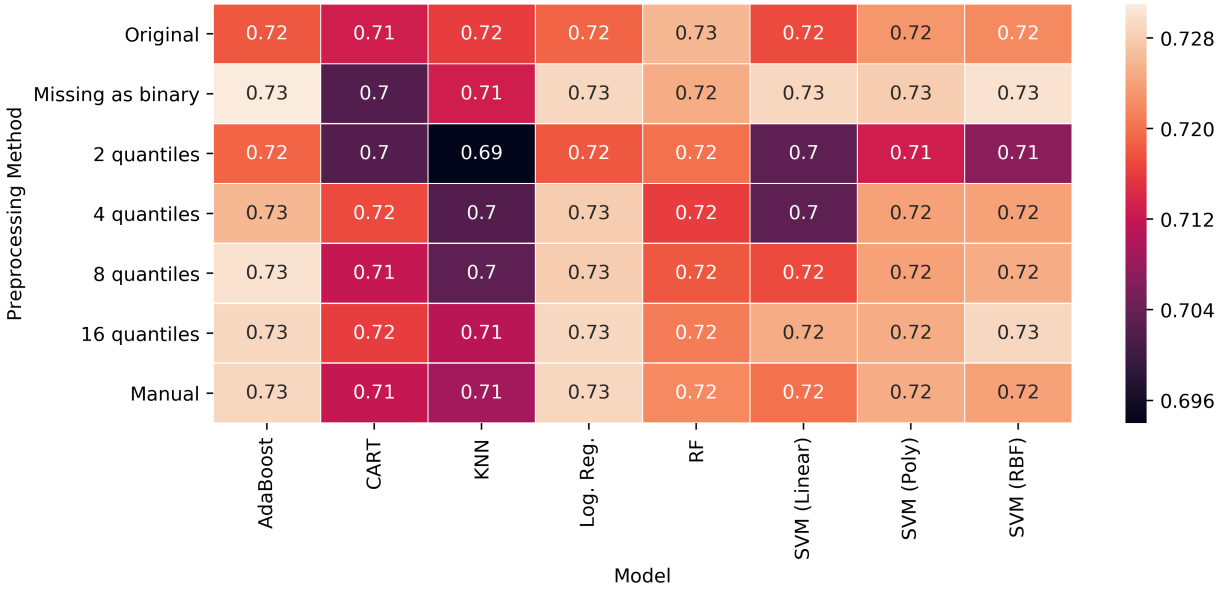


Figure 2 **Average accuracy of different preprocessing methods (averaged over ML models)**

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 3 **Average accuracy of different ML models (averaged over preprocessing methods)**

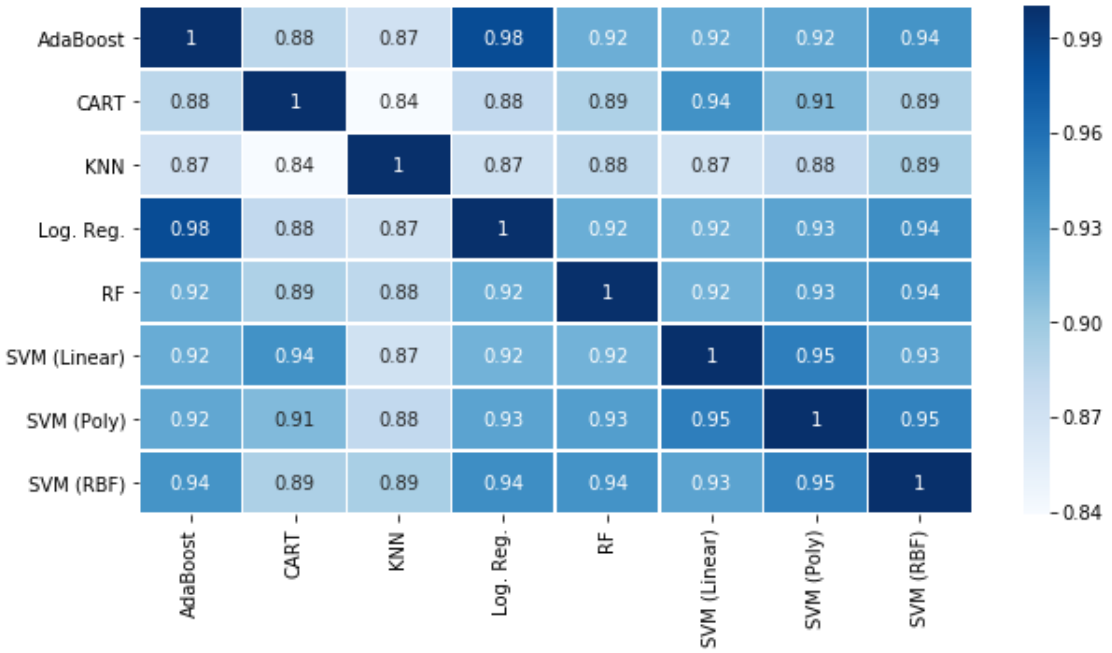
KNN	CART	SVM (Linear)	RF	SVM (RBF)	SVM (Poly)	AdaBoost	Log. Reg.
0.707	0.711	0.716	0.722	0.722	0.724	0.726	0.726

to `BinMinSetCover` serves as an initial solution to `BinMaxSupport` with $d = 0$; the solution to `BinMaxSupport` then serves as an initial solution to `BinMaxSupport` with $d + 1$). The latter algorithms are denoted as

Results. The results are summarized in Figure 8 (number of terms and support per algorithm) and Figure 9 (the average number of terms, support and runtime per model and algorithm). The key findings are listed below:

- **Sparse rules.** In Figure 8, we see that the resulting rules 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 of 778 observations per rule, and in 90% of cases the support is larger than 17. Moreover, the average support could be increased by roughly 300 observations by adding one feature on average to the summary-explanations.

Figure 4 Similarity in predictions on test data of various models (Manual discretization)



- **Short running time.** Figure 9 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. In the considered application, all the computations can be done offline as data is streamlined automatically rather than being manually typed while interacting with customers, and therefore the presented approach is practical. Note that additional algorithmic improvements can be made to speedup 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. Pre-computing rules and using those as initial solutions in future optimization problems would be a way to significantly improve the running time in practice.
- **Robustness to the underlying global model.** We see in Figure 9 that our approach produces high quality rules (in terms of sparsity and support) for the entire range of considered models.
- **Geometrical interpretation.** Figure 7 illustrates two rules. The first (on the left) represents a rule with two features (ExtrernalRiskEstimate and NetFractionRevolovingBurden) and the second rule (on the right) representes a rule with 3 features (NetFractionRevolovingBurden, AverageMInFile, and ExtrernalRiskEstimate). Geometrically, a globally consistent rule is an "open box" in the feature space that contains the observation we wish to explain and only other observations that are similarly labelled. `BinMinSetCover` tries to find a box with minimal

number of facets, while `BinMaxSupport` tries to find a box that simultanously maximizes support and minimizes the number of facets.

Figure 5 **Number of terms and support per algorithm (Algorithms `BinMinSetCover` and `BinMaxSupport` for binary datasets).**

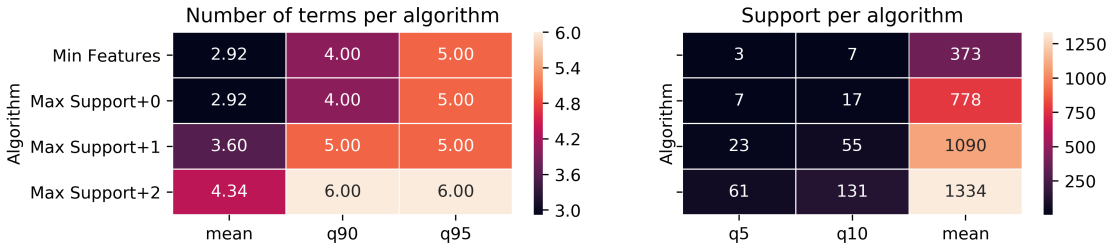
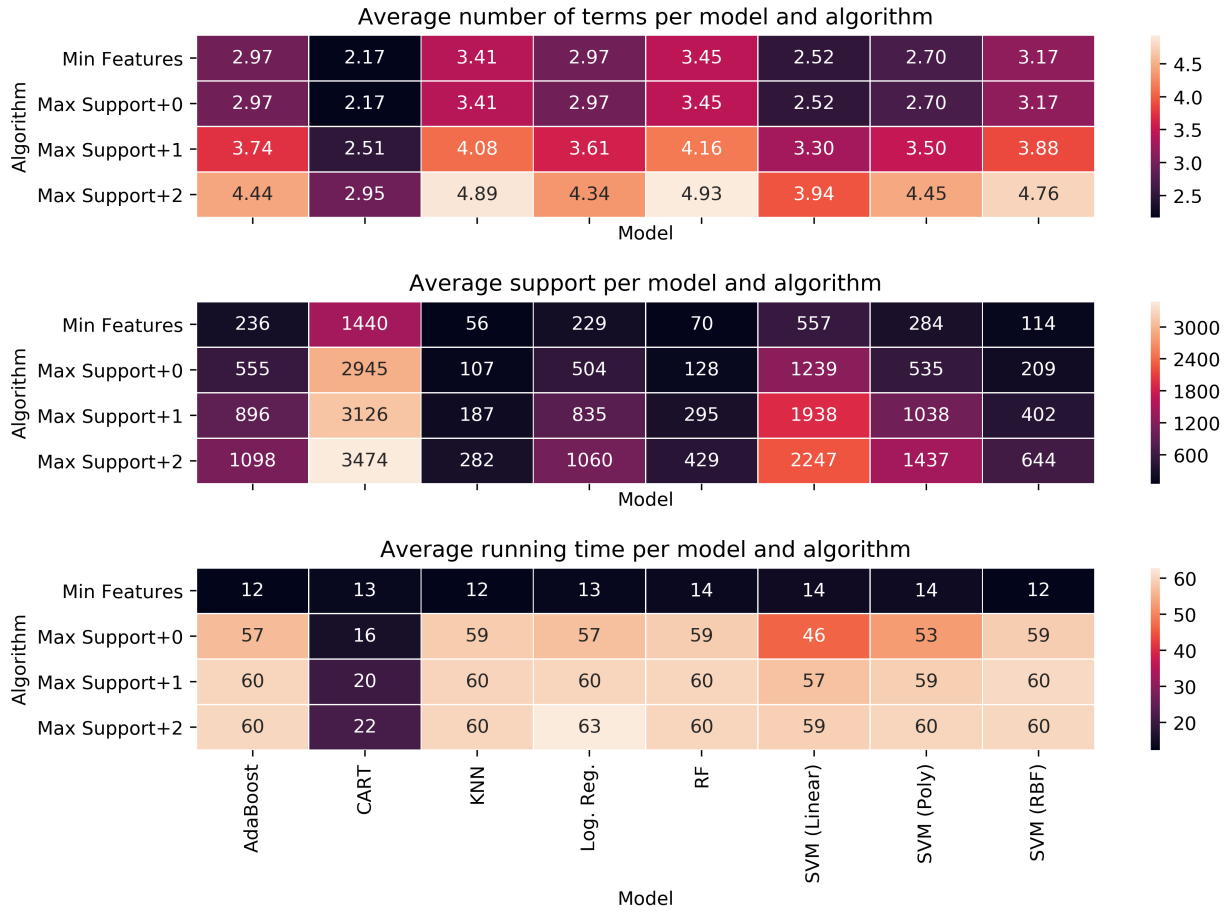


Figure 6 **Average number of terms, support and runtime (in seconds) for generating 1 rule per model and algorithm (Algorithms `BinMinSetCover` and `BinMaxSupport` for binary datasets).**



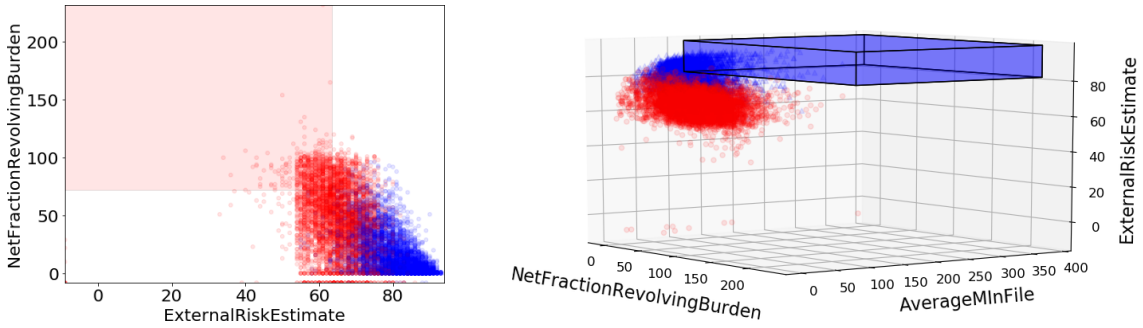
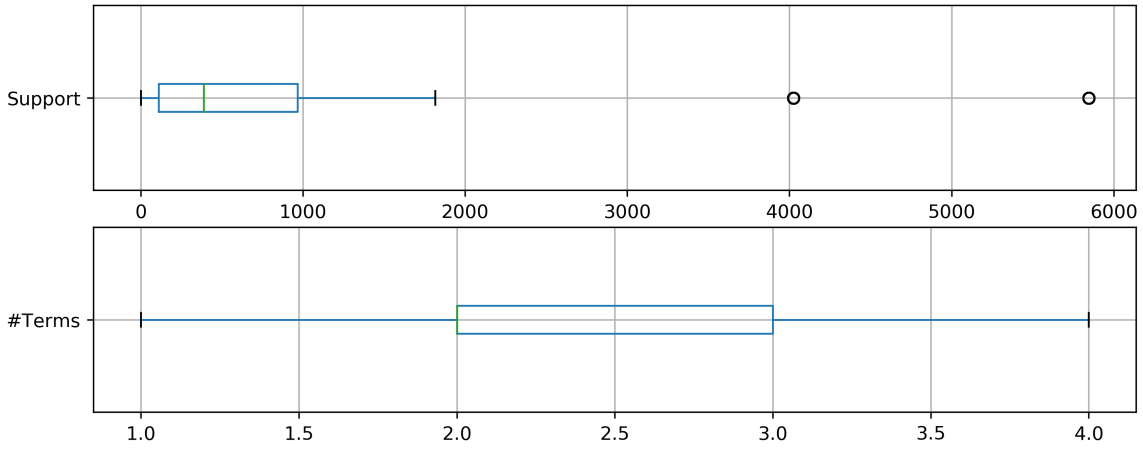


Figure 7 **Graphical illustration of two rules. These rules are summary-explanations of all the points they contain.**

4.3. Algorithms for continuous datasets

We apply the algorithm `ContMaxSupport` to the FICO dataset after adding binary features to indicate missing values ("Missing as binary"). We generate summary-explanations for various models and measure sparsity, support and running time. The results are given in Figures 8 and 9.

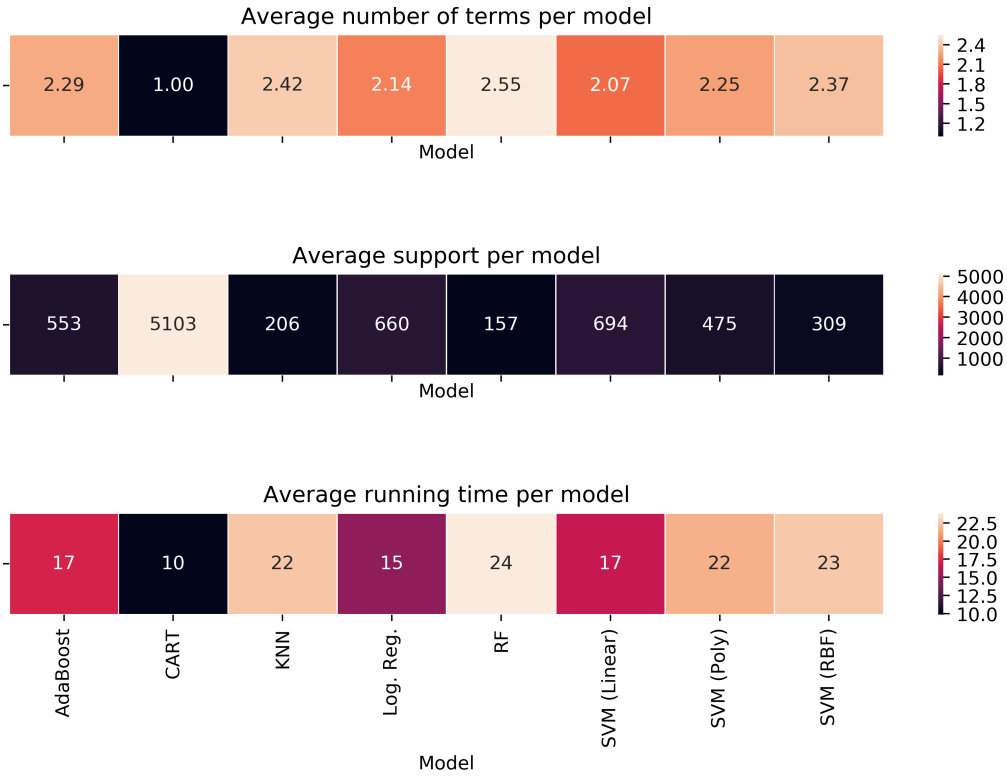
Figure 8 **Distribution of the number of terms and support (across all models; Algorithm `ContMaxSupport` for continuous datasets).**



Note. 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.

Similarly to the binary datasets, we find that the resulting summary-explanations 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 8). While there is some variability between different models, overall the resulting summary-explanations are good across models (Figure 9). Moreover, we find that the running time is quite fast; without any time limit, the rules were obtained in less than 30

Figure 9 **Average number of terms, support and runtime (in seconds) per model (Algorithm `ContMaxSupport` for continuous datasets).**



seconds (in contrast to the algorithms for binary datasets where a time limit of 1 minute was set for solving the IPs).

Our main conclusion is that the proposed algorithms for generating summary-explanations from binary and continuous datasets work well and can be used in practice. Moreover, discretization is not necessary, and in fact, the algorithms for continuous datasets seems to work faster and generate better rules, suggesting that discretization should be avoided when possible.

5. Conclusions and Future Work

This work studies summary-explanations for predictions made by ML models, which are globally consistent, thus avoiding scenarios where an explanation offered to one customer can be contradicted. 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 practically short times. 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 techniques developed in this work can also be leveraged to provide other forms of explanations:

- **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 way.
- **Counterfactual explanations.** One could generate recommendations for what the user could do to increase his or her 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 which improving may reverse the prediction made by the model (see also Wachter et al. (2017), Guidotti et al. (2018)).
- **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.

We leave these directions for future research.

References

Angwin, Julia, Jeff Larson, Surya Mattu, Lauren Kirchner. 2016. Machine bias. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

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

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

Bertsimas, Dimitris, John N Tsitsiklis. 1997. *Introduction to linear optimization*, vol. 6. Athena Scientific Belmont, MA.

Brennan, Tim, William Dieterich, Beate Ehret. 2009. Evaluating the predictive validity of the COMPAS risk and needs assessment system. *Criminal Justice and Behavior* **36**(1) 21–40.

Chen, Chaofan, Kangcheng Lin, Cynthia Rudin, Yaron Shaposhnik, Sijia Wang, Tong Wang. 2018. An interpretable model with globally consistent explanations for credit risk. *arXiv preprint arXiv:1811.12615* .

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

Craven, Mark, Jude W Shavlik. 1996. Extracting tree-structured representations of trained networks. *Advances in neural information processing systems*. 24–30.

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

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

Flores, Anthony W., Christopher T. Lowenkamp, Kristin Bechtel. 2016. False positives, false negatives, and false analyses: A rejoinder to "Machine bias: There's software used across the country to predict future criminals". *Federal probation* **80**(2).

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

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

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

Larson, J., S. Mattu, L. Kirchner, J. Angwin. 2016. How we analyzed the COMPAS recidivism algorithm. Tech. rep., ProPublica.

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

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

Ribeiro, Marco Tulio, Sameer Singh, Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 1135–1144.

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

Setiono, Rudy, Huan Liu. 1996. Symbolic representation of neural networks. *Computer* **29**(3) 71–77.

Su, Guolong, Dennis Wei, Kush R Varshney, Dmitry M Malioutov. 2016. Learning sparse two-level boolean rules. *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 1–6.

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

Wachter, Sandra, Brent Mittelstadt, Chris Russell. 2017. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harvard Journal of Law & Technology* **31**(2) 2018.

Wexler, Rebecca. 2017. 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* .

Williamson, David P, David B Shmoys. 2011. *The design of approximation algorithms*. Cambridge university press.