Learning an Interpretable Model for Imbalanced Data via Submodular Optimization

1^{st} Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2^{nd} Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3^{rd} Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—Imbalanced classification is a challenging task in many real-world applications such as cancer screening and fraud detection due to the lack of data from the minority class. In particular, in many scenarios people are interested in the interpretable solution especially for the minority class, but state-of-the-art methods for the imbalanced problem rarely consider an interpretable approach. In this paper, we utilize decision rule sets which are highly interpretable logical models and enable them to deal with highly imbalanced data through the direct optimization of the F1 score subject to cardinality constraints. We propose a novel method that greedily adds the rule with maximal marginal gain, and design an efficient minorize-maximization (MM) approach to generate rules iteratively maximizing a non-monotone submodular lower bound. Compared with existing rule learning algorithms, our algorithm does not require rule pre-mining and can achieve better solution. Empirical studies demonstrate the superior performance of our algorithm to existing interpretable algorithms in terms of both F1 score and model interpretability.**

***Index Terms*—imbalanced classification, interpretability, submodular optimization, minorize-maximization**

I. Introduction

The class imbalance problem is pervasive and ubiquitous in many real-world applications [5, 17], such as cancer screening and financial fraud detection, where the proportion of the interesting class is much smaller than others. The minority class samples are hard to model due to the lack of data. For example, in cancer screening the number of patients with cancer is significantly less than that of healthy people. Meanwhile, misdiagnosis can lead to serious consequences, requiring a good trade-off between precision and recall. In particular, interpretability being a key property of trustworthy models is a natural requirement in these scenarios, as it enables understanding behind decision makings and adjustments when bias exists.

Motivations. Existing methods for imbalanced classification are roughly grouped into data-level techniques by re-sampling the data and algorithm-level methods by directly increasing the importance of minority samples in model training [17]. To train an interpretable model for the imbalanced data, combining the interpretable model with these techniques seems feasible. However, it raises two questions: (i) **Do re-sampling methods always work well for all kinds of datasets?** In fact, simple over-sampling the minority class samples or down-sampling the majority class samples may cause model overfitting or miss some vital information. Although some methods [4, 19, 24] propose to solve these issues, the noise data and the borderline data affects the performance, especially when the minority data is not well modeled in the sampling process. Then (ii) **How about training an interpretable model on imbalanced data without data re-sampling?** Until now, some interpretable models investigate the imbalanced problems from the *algorithm-level* point of view [11, 26]. However, these methods do not work well on large-scale data with high dimension features. In fact, building interpretable and effective models for imbalanced data remains a challenging task, where the difficulty lies in two aspects:

1) The first challenge arises from pursuing model interpretability and good performance simultaneously. On the one hand, highly imbalanced data will bias the classifiers against the minority class and more to the majority class. On the other hand, simplifying model complexity can induce more interpretability but may weaken the performance. Take the rule learning method as an example, constraining the number of features contained in each rule leads to a more interpretable model, but theses rules might be prone to omitting small clusters where interesting samples lie, which further reduces *recall*.
2) The second challenge comes from model scalability for real-world applications. Some recent work on interpretable algorithms [11, 26] suffers from poor scalability.

In this paper, we aim to train interpretable classifiers on highly imbalanced data via learning decision rule sets [9], which can be expressed in disjunctive normal forms (DNF, OR-of-ANDs), a kind of model with logical clauses to be understood easily. An example of DNF models with two conditions is "IF ($w > 80$ AND $h > 180$) OR ($w < 70$ AND $h < 170$) THEN $\hat{y} = 1$". Compared with the ensemble models like Random Forest (RF) [15], the rule set is more interpretable. As we only focus on building rules for the minority class (further called the positive class), which makes it an ideal interpretable model for imbalanced classification.

Technical Challenges. We summarize three main technical challenges to build effective and scalable rule set models for imbalanced datasets as follows.

(a) **How to balance *recall* and *precision* for positive samples?** As the number of positive samples is much less, it is difficult to learn the rules for the positive samples. In addition, optimizing recall may incur many false positives, which decreases the *precision*.

(b) **How to jointly generate the rules and select the appropriate ones?** Until now, many existing rule learning algorithms rely on mining the rules in advance (the so-called pre-mininig), and then select the satisfactory ones. The separated two-step approach suffers from suboptimal solution or high-computational cost of rule pre-mining [9, 34].

(c) **How to design an effective algorithm?** Many real-world scenarios of imbalanced classification such as fraud detection involve large-scale high-dimensional data. Thus, a scalable and efficient implementation is a must for these applications.

To address these technical challenges, we propose an algorithm called **RISO** (**R**ule set for **I**mbalanced data via **S**ubmodular **O**ptimization). Specifically, we summarize the highlights of our algorithms below.

- **Direct optimization of the F1 score.** Our objective is maximizing the F1 score directly, which is appropriate to measure the performance when data is highly skewed. We formulate the F1 score as a ratio of submodular functions and transform it to a difference of non-negative monotone submodular functions.
- **Unified optimization framework and efficient algorithm based on submodular optimization.** In RISO, we propose an efficient greedy algorithm to jointly generate the rules and select the satisfactory ones for the rule set, which are defined as the sub-problem and master-problem respectively in our algorithm. Fig. 1 illustrates the overall workflow to learn a rule set, where rule selection and rule generation is related by marginal gain in submodular optimization.
- **Automatic generation of candidate rules.** In our framework, we avoid resorting to other miners for rule pre-mining. Generating each rule is transformed to maximizing a submodular problem via a minorize-maximization (MM) approach instead. We utilize the submodularity and concavity of the functions and arrive at a tight lower bound of the original function. We treat the cardinality constraint as a matroid constraint and optimize the lower bound using a simple local search, which is guaranteed to converge to the nearly optimal solution.
- **Excellent empirical performance.** The proposed algorithm RISO outperforms state-of-the-art interpretable algorithms in highly imbalanced classification tasks empirically in terms of both classification performance and interpretability. In particular, our algorithm learns the least number of rules while achieving the same F1 score among all rule based methods, which demonstrates our advantage on model interpretability.

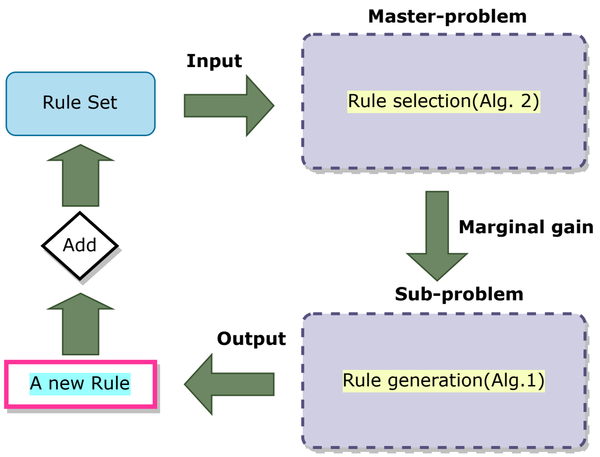


Fig. 1: Workflow of learning a rule set. Rule selection and rule generation is related by marginal gain in submodular optimization.

II. Related work

In imbalanced datasets, one of the classes includes much smaller number of samples than the other. Usually, the minority samples are of the primary interests to decision makers and the correct classification of minority samples is much more important than that of majority samples. Such situations often occur in medical diagnose, detection financial risk and so on. Typical methods often do not work well on the imbalanced datasets as they are often biased to the majority data but neglect the recall of minority data. This issue affects various kinds of classifiers including rule set models, which are the main focus of this paper. In this section we briefly review imbalanced classification and interpretable models. In the following, we call the minority class as positive class and the majority class negative class.

Imbalanced Classification. Previous approaches [13, 17, 30] to imbalanced classification problems mainly follow two paradigms: *data-level* approach and *algorithm-level* approach. Most algorithms resort to various data processing methods like data re-sampling [4, 35] to deal with imbalanced classification problem. Data re-sampling methods usually alter the training data distribution to balance the amount of positive and negative data. Typical techniques include over-sampling minority class samples and under-sampling majority class samples. Although both sampling approaches can adapt to any machine learning algorithms directly, they do not generate new information and over-sampling may cause model over-fitting. To deal with this, SMOTE [4] and its variants [19, 24] are proposed to create new samples by interpolating the minority class samples. Unfortunately, noises are also produced when artificially generating these samples. Our experiments on combining the baseline models with these data re-sampling methods demonstrate that data re-sampling methods do not always work well. The *algorithm-level* models [23, 31], often called cost-sensitive learning models, concentrate on adjusting the weights for samples from different classes in the loss function, where heavier penalties are assigned to samples from the minority class. Unlike data re-sampling, cost-sensitive learning requires specific modifications of existing algorithms. Some works focus on boosting multiple classifiers [12, 18] to address the imbalanced datasets and [6] proposed *SMOTEBoost* to improve the minority samples predicting by combining the boosting and data re-sampling.

Interpretable Models. Interpretable models are more comprehensible and human-readable than deep networks and complex ensemble models. The demand for model interpretability arises in many important decision-making situations such as medical diagnosis and safety supervision. White-box interpretable methods have been proposed for a long time in the literature [7]. A classical early algorithm is RIPPER [8], which adopts a greedy approach to select features and form conditions to cover samples from the interesting class. Recent methods [10, 20, 25, 32, 33, 36] usually take a two-step strategy to train a rule set: candidate rules generation (or pre-mining) and appropriate rules selection. These two stages are usually performed separately, and thus leading to suboptimal results such as precision loss incurred by missing rules in rule pre-mining.

To address this issue, methods without pre-mining have been proposed to unify the optimization procedure by formulating an integer program (IP) and solved via column generation [9] or submodular optimization [34]. This paper proposes an interpretable model not mining the rules in advance. Instead, it automates the rule generation (further called sub-problem) in the rule set generation (further called master problem) from the view of submodular optimization.

Imbalanced data often occur in the fields that require high interpretable models, but most research on imbalanced classification ignores interpretability. Until recently, some work [2, 11, 26] have been proposed. [11] proposes a box drawing classifier called Exact Boxes, i.e., a rule set, for class imbalance problems. However, the Exact Boxes algorithm solves a mixed integer programming problem applicable to small to medium sized datasets only. Another approximated version called Fast Boxes is proposed, but its performance on imbalanced data is not satisfying in practice. BRACID [26] proposes a bottom-up rule learning algorithm by treating all samples as hybrid representation rules first and then inducing rules from these single samples. However, BRACID suffers from poor scalability because bottom-up learning might generate vaster number of rules and be more sensitive to the noisy data. LIBRE [2] proposes to combine weak learners with a simple union to improve the interpretability, where each weak learner operates on a random subset of features. An early work is EXPLORE[29], which adopts a less greedy search for the minority rules reaching the threshold of certain supports and uses a standard sequential covering procedure to generate rules for majority class samples. In this way, EXPLORE generates more rules for minority, which helps classifying the unknown samples when voting. In this paper, our proposed algorithm devotes to training rule set models to solve the imbalanced classification problems by greedily solving a submodular maximization problem, learns the rules for the minority samples and realizes a good balance between the *recall* and *precision*.

III. Problem Formulation

Without loss of generality, in this paper we focus on binary classification problems with highly imbalanced data, where the number of samples of one class is much smaller than that of the other. Also we assume all features are binary features. Note that the categorical features can be easily binarized using one-hot encoding. For numeric features, here we use a sequence of thresholds to compare with the numerical features to generate binarized features, which is also known as bucketing strategy.

Formally, given a dataset $\mathcal{X} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ with a feature index set Γ of size d, where $\boldsymbol{x}_i \in \{0,1\}^d$ and $y_i \in \{0,1\}$ denote the binary feature vector and the label of the i-th sample, respectively. Assuming that a sample $(\boldsymbol{x}_i, y_i)$ is positive if $y_i = 1$ and negative if $y_i = 0$. We aim to learn an interpretable rule set from the dataset $\mathcal{X}$.

Definition 1 (**Rule**). *A rule r is a set of feature indices, i.e. r is a subset of Γ. A sample $(\boldsymbol{x}_i, y_i)$ is considered to be covered by a rule r if and only if $r \subseteq \{j \in \Gamma | x_{i,j} = 1\}$.*

Definition 2 (**Rule Set**). *A rule set s consists of multiple rules, and serves as a classifier, which classifies a sample as positive if the sample is covered by at least one rule in s, and as negative if there is no rule in s that covers it, i.e., predict the label of $\boldsymbol{x}_i$ as $\bigvee_{r \in s}(\bigwedge_{j \in r} x_{i,j})$.*

Let $\mathcal{X}_j$, $\mathcal{X}_r$ and $\mathcal{X}_s$ denote the set of samples covered by the j-th feature, the rule r and the rule set s, respectively. Note that in the following we may use different subscripts to distinguish sets of samples covered by different features/rules/sets. Thus, we can write $\mathcal{X}_j = \{i | \boldsymbol{x}_{i,j} = 1\}$, $\mathcal{X}_r = \{i | (\bigwedge_{j \in r} \boldsymbol{x}_{i,j}) = 1\}$, $\mathcal{X}_s = \{i | \bigvee_{r \in s}(\bigwedge_{j \in r} \boldsymbol{x}_{i,j}) = 1\}$. According to the relationships among the features, rules and rule sets, we get $\mathcal{X}_r = \cap_{j \in r} \mathcal{X}_j$ and $\mathcal{X}_s = \cup_{r \in s} \mathcal{X}_r$. Let $\mathcal{X}^+$ be the set of all positive samples in the dataset $\mathcal{X}$, we define the operation $\mathcal{A}^+$ as $\mathcal{A}^+ = \mathcal{A} \cap \mathcal{X}^+$ for any set $\mathcal{A}$, which returns the set of all positive samples in $\mathcal{A}$.

It is well-known that accuracy and error rate, the most frequently used metrics in classification evaluation, are not applicable in imbalanced classification as they are dominated by the majority class [17]. In this paper, we learn a rule set by directly maximizing the F1 score, which combines both precision and recall in classification evaluation. Formally, we first derive the F1 score of a given rule set s. Let $y_i \in \{0,1\}$ and $h_i \in \{0,1\}$ denote the true label and the prediction of the i-th sample, respectively. We can represent precision and recall as $\sum_i y_i h_i / \sum_i h_i$ and $\sum_i y_i h_i / \sum_i y_i$, respectively. As a result, the F1 score can be readily computed as

$$F1(s) = \frac{2 \sum_i y_i h_i}{\sum_i h_i + \sum_i y_i}. \tag{1}$$

Note that the term $\sum_i y_i h_i$ denotes the number of positive samples that are correctly classified by our model, and the term

$\sum_i h_i$ denotes the number of all samples that are predicted as positive by our model. Thus $\sum_i h_i$, $\sum_i y_i h_i$ and $\sum_i y_i$ can be rewritten as $|\mathcal{X}_s|$, $|\mathcal{X}_s^+|$ and $|\mathcal{X}^+|$, respectively. Formally, we formulate the F1 score as follows:

$$F1(s) = \frac{2|\mathcal{X}_s^+|}{|\mathcal{X}_s| + |\mathcal{X}^+|} = \frac{2|\cup_{r\in s}\mathcal{X}_r^+|}{|\cup_{r\in s}\mathcal{X}_r| + |\mathcal{X}^+|}. \quad (2)$$

In this paper, our goal is to learn an interpretable and simple rule set with maximal F1 score. Formally, we have the following optimization problem:

$$\begin{aligned} \max_{s\subseteq\Omega} \ & \frac{2|\cup_{r\in s}\mathcal{X}_r^+|}{|\cup_{r\in s}\mathcal{X}_r| + |\mathcal{X}^+|}, \\ \text{s.t.} \ & |s| \le K, \\ & \Omega = \{r||r| \le l, r\subseteq\Gamma\}. \end{aligned} \quad (3)$$

To ensure the interpretablity of rules, we consider both the number of rules and the length of rules. Specifically, we require all rule lengths are no more than l, i.e., $|r| \le l$, $\forall r \subseteq \Gamma$, where l is a predefined parameter, Γ is the feature index set with size d. Also we add a cardinality constraint $|s| \le K$ to ensure that there are no more than K rules in the rule set.

a) Discussion: As $|\cup_{r\in s}\mathcal{X}_r^+|$ and $|\cup_{r\in s}\mathcal{X}_r|$ denote the number of positive samples and all samples covered by the rule set s, respectively, both of them are monotone, non-negative and submodular. Hence, our optimization problem can be considered as maximizing a ratio of two submodular functions subject to cardinality constraints. Maximizing the ratio of two non-negative monotone submodular functions f and g has been investigated in the literature [3, 28] and several methods have been proposed. Among them, *GreedRatio* is a widely used one, which greedily adds the item with the highest ratio of marginal gains of f and g to the solution set. However, *GreedRatio* cannot be applied to our model as explained below. Specifically, consider adding a rule r' to the current rule set s, the marginal gains of the numerator and denominator of (3) are given respectively as

$$|\mathcal{X}_s^+ \cup \mathcal{X}_{r'}^+| - |\mathcal{X}_s^+| = |\mathcal{X}_{r'}^+ \setminus \mathcal{X}_s^+| \overset{(a)}{=} |\mathcal{X}_{r'}^+ \setminus \mathcal{X}_s|, \quad (4)$$

$$|\mathcal{X}_s \cup \mathcal{X}_{r'}| - |\mathcal{X}_s| = |\mathcal{X}_{r'} \setminus \mathcal{X}_s|, \quad (5)$$

where equation (a) comes from the fact that $\mathcal{X}_{r'}^+$ is a subset of $\mathcal{X}^+$. *GreedRatio* selects r' with the highest ratio below,

$$\begin{aligned} \frac{|\mathcal{X}_s^+ \cup \mathcal{X}_{r'}^+| - |\mathcal{X}_s^+|}{|\mathcal{X}_s \cup \mathcal{X}_{r'}| - |\mathcal{X}_s|} &= \frac{|\mathcal{X}_{r'}^+ \setminus \mathcal{X}_s|}{|\mathcal{X}_{r'} \setminus \mathcal{X}_s|} = \frac{|\mathcal{X}_{r'}^+ \setminus \mathcal{X}_s|}{|\mathcal{X}_{r'}^+ \setminus \mathcal{X}_s| + |\mathcal{X}_{r'}^- \setminus \mathcal{X}_s|} \\ &= \frac{1}{1 + \frac{|\mathcal{X}_{r'}^- \setminus \mathcal{X}_s|}{|\mathcal{X}_{r'}^+ \setminus \mathcal{X}_s|}}, \end{aligned} \quad (6)$$

where $|\mathcal{X}_{r'}^- \setminus \mathcal{X}_s| = |\mathcal{X}_{r'} \setminus \mathcal{X}_s| - |\mathcal{X}_{r'}^+ \setminus \mathcal{X}_s|$ denotes the number of negative samples newly covered by rule r'. Obviously, *GreedRatio* favors rules with the lowest $|\mathcal{X}_{r'}^- \setminus \mathcal{X}_s| / |\mathcal{X}_{r'}^+ \setminus \mathcal{X}_s|$, which makes it prefer the rule that covers 1 positive samples and 0 negative samples over the rule that covers 1000 positive samples and 3 negative samples. In other words, *GreedRatio* gives higher priority to the rule with higher precision while ignores its recall, leading to unsatisfying model generalization. Hence, we conclude that *GreedRatio* is inappropriate to be used directly in our model.

IV. RISO Algorithm

Most greedy approaches solve the submodular function maximization problems by designing an item selection strategy based on the **marginal gain**. However, the marginal gain of the denominator in (2) is only related to the term $|\mathcal{X}_s|$. Hence, any greedy method maximizing the F1 score based on the marginal gain of the denominator in (2) will share the same strategy with methods maximizing $|\mathcal{X}_s^+|/|\mathcal{X}_s|$, i.e., the precision. As a result, it fails to optimize the recall of the classifier. To address this issue, we apply a nonlinear transformation on the F1 score to involve the effect of $|\mathcal{X}^+|$ in the marginal gain. We use logarithm function as the nonlinear transformation due to its submodular preserving property. The resulting objective function is rewritten as

$$\max_{s\subseteq\Omega} \ \log(|\bigcup_{r\in s}\mathcal{X}_r^+|) - \log(|\bigcup_{r\in s}\mathcal{X}_r| + |\mathcal{X}^+|), \ \text{s.t.} \ |s| \le K. \quad (7)$$

Let $G(s) \triangleq \log(|\cup_{r\in s}\mathcal{X}_r^+|)$ and $C(s) \triangleq \log(|\cup_{r\in s}\mathcal{X}_r| + |\mathcal{X}^+|)$. As logarithm function is non-decreasing and concave, both $G(s)$ and $C(s)$ are non-negative monotone submodular functions [2]. Consequently, the objective function can be viewed as a difference between two submodular functions.

In this paper, we propose a greedy method, named **RISO** (Rule set for Imbalanced data via Submodular Optimization), to maximize the aforementioned function subject to cardinality constraints. RISO greedily adds the rule r with maximal marginal gain to the rule set until some termination conditions are met. Obviously, it is impractical to evaluate marginal gains of all possible rules especially when the number of candidate rules is large. Meanwhile, generating candidate rules is usually based on some predefined metrics and thresholds, whose parameters if improperly defined would significantly degrade the performance of the method. Hence, we propose to find the optimal rule by solving one optimization problem, unifying the process of generating candidate rules and selecting the optimal one. Following subsection provides the details of the rule generation process, and the overall rule learning method, i.e. RISO, is introduced at section IV-B.

A. Proposed Efficient Rule Generation Method

In this subsection, we introduce our rule generation method. At the tth iteration of our rule learning process, we aim to find a rule r leading to the highest F1 score increase if added to the current rule set s. We learn the desired rule by solving an optimization problem, where a distorted marginal gain of a rule is used as the objective function by introducing a parameter α. Particularly, we consider the following general problem,

$$\max_{|r|\le l} \quad \alpha G(r|s) - C(r|s), \quad (8)$$

where $G(r|s) \triangleq G(s\cup\{r\}) - G(s)$ and $C(r|s) \triangleq C(s\cup\{r\}) - C(s)$ denote the marginal gains of G and C by adding r to

s. We note that different with the original marginal gain, we add a coefficient α for $G(r|s)$, and we will show its critical effect on adaptively adjusting the trade-off between precision and recall in next subsection. As s is independent with r, we would like to find a rule r such that following function is maximized, i.e.,

$$\max_{|r|\leq l} \quad \alpha \log(|\mathcal{X}_r^+ \cup \mathcal{X}_s^+|) - \log(|\mathcal{X}_r \cup \mathcal{X}_s| + |\mathcal{X}^+|). \tag{9}$$

Since a rule r is a set of feature indices, as discussed previously, finding an optimal rule is equivalent to finding a set of features that maximizes the following function,

$$\begin{aligned} W(r) =& \alpha \log(|(\cap_{j\in r}\mathcal{X}_j)^+ \cup \mathcal{X}_s^+|) \\ &- \log(|(\cap_{j\in r}\mathcal{X}_j) \cup \mathcal{X}_s| + |\mathcal{X}^+|). \end{aligned} \tag{10}$$

Note that directly maximizing $W(r)$ is difficult. Although $|(\cap_{j\in r}\mathcal{X}_j)^+ \cup \mathcal{X}_s|$ is a supermodular function, the presence of logarithm function makes the property of $\log|(\cap_{j\in r}\mathcal{X}_j)^+ \cup \mathcal{X}_s|$ non-trivial. In our algorithm, $W(r)$ is maximized using MM algorithm, which iteratively increases the value of the objective function by maximizing a tight lower bound. We propose a proper lower bound of $W(r)$ by finding a lower bound of $\log|(\cap_{j\in r}\mathcal{X}_j)^+ \cup \mathcal{X}_s^+|$ and an upper bound of $\log(|(\cap_{j\in r}\mathcal{X}_j) \cup \mathcal{X}_s| + |\mathcal{X}^+|)$ separately.

To find a lower bound of $\log(f(r))$, where $f(r) = |(\cap_{j\in r}\mathcal{X}_j)^+ \cup \mathcal{X}_s|$, it suffices to find a lower bound of $f(r)$ due to the non-decreasing property of the logarithm function. Nevertheless, the lower bound of $f(r)$ may be negative which makes the whole function undefined. To address this issue, we first define a new function, ψ, whose domain is extended to $(-\infty, +\infty)$, while still maintaining the monotone property of the logarithm function. We keep the function as its original value when $x \geq 1$, and when $x < 1$, we make the approximation by using the first-order Taylor expansion at the point $x = 1$, hence leading to

$$\psi(x) = \begin{cases} \log(x) & x \geq 1, \\ x - 1 & x < 1. \end{cases} \tag{11}$$

It is easy to check that $\log(f(r))$ and $\psi(f(r))$ share the same function value for all r except the case when $(\cap_{j\in r}\mathcal{X}_j)^+ = \emptyset$ and $s = \emptyset$, which corresponds to zero F1 score. Rule sets with zero F1 score can never be optimal as one can always construct another rule set with positive F1 score by selecting K positive samples and treating them as rules. As a result, the optimal solution that maximizes $\log(f(r))$ is same as that of $\psi(f(r))$, which makes $\psi(f(r)))$ a tractable replacement to $\log(f(r))$.

Next we present the solution of finding a tight lower bound of $f(r)$. Notice that it is equivalent to finding a tight upper bound of the submodular function $-f(r)$, since $f(r)$ is supersubmodular. Motivated by the modular upper bounds of submodular functions presented in [16, 27], two proper lower bounds of $f(r)$ are given as

$$\begin{aligned} L^1_{f,r^{(t)}}(r) \triangleq& f(r^{(t)}) - \sum_{j\in r^{(t)}\setminus r} f(j|r^{(t)} \setminus \{j\}) + \sum_{j \in r\setminus r^{(t)}} f(j|\emptyset) \\ \leq& f(r), \ \forall\ r \subseteq \Gamma, \end{aligned} \tag{12}$$

$$\begin{aligned} L^2_{f,r^{(t)}}(r) \triangleq& f(r^{(t)}) - \sum_{j\in r^{(t)}\setminus r} f(j|\Gamma \setminus \{j\}) + \sum_{j \in r\setminus r^{(t)}} f(j|r^{(t)}) \\ \leq& f(r), \ \forall\ r \subseteq\ \Gamma, \end{aligned} \tag{13}$$

where $r^{(t)}$ denotes the current estimation of r. These two inequalities hold for all possible $r^{(t)}$, and the equality is achieved when $r = r^{(t)}$.

Lemma 1. *Both $L^1_{f,r^{(t)}}(r)$ and $L^2_{f,r^{(t)}}(r)$ are modular and non-increasing functions.*

Proof. As $L^1_{f,r^{(t)}}(r)$ and $L^2_{f,r^{(t)}}(r)$ share similar structure, then we only prove the results on $L^1_{f,r^{(t)}}(r)$, and that of $L^2_{f,r^{(t)}}(r)$ can be proved similarly. For $L^1_{f,r^{(t)}}(r)$, the marginal gain of adding element j into r is

$$\begin{aligned} L^1_{f,r^{(t)}}(j|r) =& L^1_{f,r^{(t)}}(r \cup \{j\}) - L^1_{f,r^{(t)}}(r) \\ =& \begin{cases} f(j|r^{(t)} \setminus \{j\}) & j \in r^{(t)} \\ f(j|\emptyset) & j \notin r^{(t)} \end{cases}. \end{aligned} \tag{14}$$

Note that $L^1_{f,r^{(t)}}(j|r)$ is independent with r and non-positive due to $f(r)$ being a non-increasing function. Then we can conclude that $L^1_{f,r^{(t)}}(r)$ is a modular and non-increasing function. □

Since we have obtained two lower bounds of $f(r)$, the lower bounds of $\psi(f(r))$ can be readily obtained as $\psi(L^1_{f,r^{(t)}}(r))$ and $\psi(L^2_{f,r^{(t)}}(r))$.

Lemma 2. *Both $\psi(L^1_{f,r^{(t)}}(r))$ and $\psi(L^2_{f,r^{(t)}}(r))$ are submodular functions.*

Proof. We only provide proof for function $\psi(L^1_{f,r^{(t)}}(r))$, as the result on $\psi(L^2_{f,r^{(t)}}(r))$ can be arrived at similarly. We first prove that the function $g(x) = \psi(x + \Delta) - \psi(x)$ is non-increasing for all non-negative Δ. The derivative of $g(x)$ can be computed as

$$g'(x) = \begin{cases} \frac{1}{x+\Delta} - \frac{1}{x} & x \geq 1 \\ \frac{1}{x+\Delta} - 1 & x + \Delta \geq 1, x < 1 \\ 0 & x + \Delta < 1 \end{cases}. \tag{15}$$

It is clear that $g'(x)$ is non-positive, which implies that $g(x)$ is a non-increasing function. In other words, we have $\psi(x + \Delta) - \psi(x) \leq \psi(y + \Delta) - \psi(y)$ for all $x \geq y$. According to Lemma 1, we have $L^1_{f,r^{(t)}}(r \cup \{i\}) \geq L^1_{f,r^{(t)}}(r \cup \{i, j\})$ and $L^1_{f,r^{(t)}}(r)$ is a modular function. Then let $x = L^1_{f,r^{(t)}}(r \cup \{i\})$,

$y = L^1_{f,r^{(t)}}(r \cup \{i,j\})$, and $\Delta = L^1_{f,r^{(t)}}(r) - L^1_{f,r^{(t)}}(r \cup \{i\}) = L^1_{f,r^{(t)}}(r \cup \{j\}) - L^1_{f,r^{(t)}}(r \cup \{i,j\}) \geq 0$, we have

$$\begin{aligned}\psi(x+\Delta)-\psi(x) =& \psi(L^1_{f,r^{(t)}}(r)) - \psi(L^1_{f,r^{(t)}}(r \cup \{i\})\\ \leq& \psi(y+\Delta)-\psi(y)\\ =& \psi(L^1_{f,r^{(t)}}(r \cup \{j\})) - \psi(L^1_{f,r^{(t)}}(r \cup \{i,j\}),\end{aligned}$$

which shows that $\psi(L^1_{f,r^{(t)}}(r))$ is submodular. □

Next, we show how to find an upper bound of $\log(g(r))$, where $g(r) = |(\cap_{j \in r} \mathcal{X}_j) \cup \mathcal{X}_s| + |\mathcal{X}^+|$. Similar to $f(r)$, $g(r)$ is also a supermodular function and we proceed to find a tight lower bound of $-g(r)$, as [16, 27] also present a permutation-based method to find lower bounds for submodular functions. However, it is also mentioned in [16] that to certify an optimum of such a lower bound, one needs to examine about $O(m)$ permutations which is critical to the algorithm, where m is the size of the ground set, i.e. in our setting, $|\Gamma|$. Hence, in this paper, we employ another strategy to find an upper bound of $\log(g(r))$ by utilizing the concavity of logarithm functions. As $\log(x) \leq \log(x_0) + \frac{1}{x_0}(x - x_0)$, then a tight upper bound of $\log(g(r))$ is readily given as, which holds for any $r^{(t)}$,

$$\log(g(r)) \leq \log(g(r^{(t)})) + \frac{1}{g(r^{(t)})}(g(r) - g(r^{(t)})). \quad (16)$$

Since such an upper bound removes the non-linearity of the logarithm function, it shares the same property of $g(r)$, which is supermodular. We combine the bounds obtained above, and derive two tight lower bounds of $W(r)$ as follows,

$$W(r) \geq \alpha\psi(L^1_{f,r^{(t)}}(r)) - \frac{g(r)}{g(r^{(t)})} = V^1_{\alpha,r^{(t)}}(r), \quad (17)$$

$$W(r) \geq \alpha\psi(L^2_{f,r^{(t)}}(r)) - \frac{g(r)}{g(r^{(t)})} = V^2_{\alpha,r^{(t)}}(r). \quad (18)$$

The problem of maximizing $W(r)$ is translated to maximizing $V^1_\alpha(r|r^{(t)})$ and $V^2_\alpha(r|r^{(t)})$.

Lemma 3. *Both $V^1_\alpha(r|r^{(t)})$ and $V^2_\alpha(r|r^{(t)})$ are non-monotone submodular functions.*

Proof. Both $L^1_{f,r^{(t)}}(r)$ and $L^2_{f,r^{(t)}}(r)$ are submodular, and $g(r)$ is supermodular, so both $V^1_\alpha(r|r^{(t)})$ and $V^2_\alpha(r|r^{(t)})$ are submodular functions. It is obvious that $L^1_{f,r^{(t)}}(r)$, $L^2_{f,r^{(t)}}(r)$ and $g(r)$ are all non-increasing functions, then $V^1_\alpha(r|r^{(t)})$ and $V^2_\alpha(r|r^{(t)})$ are both non-monotone functions. □

Maximizing a non-monotone submodular function subject to cardinality constraints has been extensively studied in the literature. Specifically, RISO maximizes $V^1_\alpha(r|r^{(t)})$ and $V^2_\alpha(r|r^{(t)})$ by using a simple local search method. As shown in [21], by identifying the cardinality constraint as a matroid constraint, the local search method can provide at least 1/4-approximation to the optimum.

Lemma 4. *The local search method can provide at least 1/4-approximation to the optimum of $V^1_\alpha(r|r^{(t)})$ and $V^2_\alpha(r|r^{(t)})$.*

Proof. See the Theorem 2.6 of [21]. □

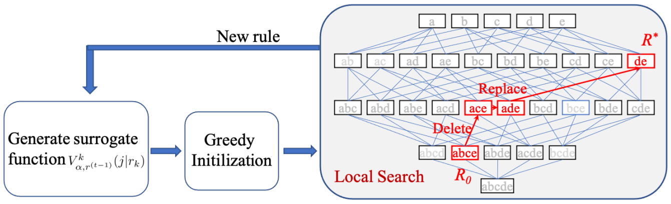


Fig. 2: Framework of Algorithm 1.

Algorithm 1: GenerateRule

```
Input  : $l$: Maximal size of a rule.
         $\alpha$: trade-off parameter.
Output: Rule $r$.
1  Initialize $r_0$ by GreedRatio;
2  for $t = 1, 2, ...$ do
3      $r_k \leftarrow \emptyset$, $k \in \{1, 2\}$;
4      for $i = 1, 2, ..., l$ do
5          $j^* \leftarrow \arg\max_{j \in \Gamma} V^k_{\alpha,r^{(t-1)}}(j|r_k)$;
6          $r_k \leftarrow r_k \cup \{j^*\}$;
7      end
8      repeat
9          $\tilde{r} \leftarrow r_k$;
10         for $i \in r_k$ do
11             Define $V(j) = V_{\alpha,r^{(t-1)}}(j|r_k \setminus \{i\})$;
12             $j^* \leftarrow \arg\max_{j \in \Gamma \cup \{\emptyset\}} V(j)$;
13             if $V(j^*) > 0$ then $r_k \leftarrow (r_k \setminus \{i\}) \cup \{j^*\}$;
14         end
15     until $\tilde{r} = r_k$;
16     $r^{(t)} \leftarrow \arg\max_{r \in \{r_1, r_2\}} W(r)$;
17     if $r^{(t)} = r^{(t-1)}$ then Return $r^{(t)}$;
18 end
```

Formally, Algorithm 1 summarizes our MM-based method for learning an optimal rule. The lines 4-7 present the initialization of each MM iteration, where we insert l features to an empty set greedily. Local search is then performed to refine the solution, as stated in lines 8-15. The replacement operation, which replaces a feature in r with another feature in Γ if the objective function can be improved, is performed in the local search procedure only. To perform the deletion operation, we extend Γ by including the empty set, as shown in line 12 of Algorithm 1, so that a feature in r can be deleted by replacing it with the empty set. To conclude, we have presented an efficient rule generating method, which iteratively improves the objective and is guaranteed to converge to a local optimum of the objective function. Our method illustrated in Fig. 3 only involves the set operation and is hence a computational efficient method. Note that the computational complexity of our method can be further improved by permitting early stopping, i.e, terminating the local search if no significant improvement is achieved by replacing features.

B. Proposed rule set learning framework

In this subsection, we present the framework of RISO whose working flow is shown in Fig. 1. Recall that as the objective (7) is a difference of two submodular functions, RISO is based on the method *DistortedGreedy* [14], for maximizing the difference between a non-negative monotone submodular function and a modular function. We will show that by introducing the notation curvature, *DistortedGreedy* is applicable to our problem. We first define the curvature γ of $C(s)$

$$\gamma = 1 - \min_{r\in\Omega} \frac{C(r|\Omega \setminus \{r\})}{C(r|\emptyset)} \tag{19}$$

to measure the closeness of $C(s)$ to a modular function, and γ is unknown in advance.

Algorithm 2: Rule Set for Imbalanced Data Set

```
Input : Training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, cardinality $K$,
        maximal size of a rule $l$.
Output: Rule set $s$.
1 Let $s_0 \leftarrow \emptyset$;
2 for $i = 0, 1, ..., K-1$ do
3     $\alpha_i \leftarrow (1 - \frac{\gamma}{K})^{K-(i+1)}$;
4     $r^* \leftarrow$ GenerateRule$(l, \alpha_i)$;
5     if $\alpha_i.G(r^*|s) - C(r^*|s) > 0$ then
6         $s_{i+1} \leftarrow s_i \cup \{r^*\}$;
7     end
8 end
9 Return $s_{K-1}$.
```

The complete procedure for rule set learning is summarized in Algorithm 2. Given the training dataset, the maximal number of rules and the limitation on the length of rules, and by initializing the rule set as an empty set, we iteratively add a rule r^* to the set by invoking Algorithm 1, which is stated in line 4 of Algorithm 2. Similar to *DistortedGreedy*, we adaptively update the trade-off between $G(r|s)$ and $C(r|s)$, denoted as α, as stated in line 3.

In early stages, a small value of α is adopted to select rules with higher *precision*. The value of α is gradually increased to improve the *recall* of the rule set. In other words, RISO tends to select the rules with higher *precision* in the early iteration steps and focuses on the rules with higher *recall* later. The rationale behind the α updating strategy is that when first focusing on the rules with high *precision* and low *recall*, RISO can achieve higher precision and later improve the recall by including more rules. However, if the rules with high *recall* but low *precision* are given more priority in early stages, it is difficult to eliminate the effect of the false positive samples.

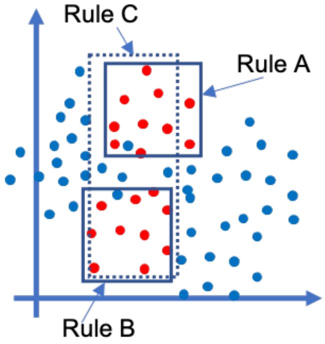


Fig. 3: Example of the proposed rule selection strategy.

To better illustrate this, we show a toy example in Fig. 3. Given a dataset with 20 positive and 100 negative samples, and our goal is to select 2 rules from 3 candidate rules, namely rules A, B and C, where rule A covers 10 positive and 1 negative samples, rule B covers the rest 10 positive samples which are not covered by rule A and 1 additional negative sample, and rule C covers 18 positive samples and 5 negative samples. We first discuss the scenario that we replace the α update strategy in line 3 of Algorithm 2 with $\alpha_i = 1$. At the first iteration for rule A, $|\mathcal{X}^+| = 20$, $|\cup_{r\in s} \mathcal{X}_r| = 11$, $|\cup_{r\in s} \mathcal{X}_r^+| = 10$, then the marginal gain of rule A is $\log(10/(20+11)) \approx \log(0.31)$. Similarly, the marginal gains of rule B and rule C are given as $\log(10/(20+11)) \approx \log(0.31)$, $\log(18/(20+18+5)) \approx \log(0.42)$, respectively. In this scenario, RISO will select rule C in the first iteration and rule B(or A) in the second iteration. Finally, RISO constructs a rule set which covers 20 positive samples and 6 negative samples. However, with the proposed α update strategy in line 3 of Algorithm 2, at the first iteration (corresponding to $K = 2$, $i = 0$, $\gamma = 1$, and $\alpha = 0.5$), the marginal gains of rule A, B and C are given as $0.5 \times \log(10) - \log(31) (\approx \log(0.102))$, $0.5 \times \log(10) - \log(31) (\approx \log(0.102))$ and $0.5 \times \log(18) - \log(43) (\approx \log(0.099))$, respectively. Then RISO will return a better rule set that consists of rule A and rule B, which covers only 2 negative samples.

The following theorem provides a theoretical guarantee for the proposed method.

Theorem 1. *If r^* in line 4 of Algorithm 2 is the optimal solution of problem (9), Algorithm 2 will return a rule set s satisfying*

$$G(s) - C(s) \geq (1 - e^{1-\gamma})G(s^*) - C(s^*), \ |s| \leq K, \tag{20}$$

where s^* represents the optimal solution of problem (7).

Proof. Define $q \triangleq \frac{1-\gamma}{K}$ and $\alpha_i \triangleq (1-q)^{K-(i+1)}$. Obviously, $\alpha_{-1} = (1-q)^K \leq e^{\gamma-1}$ and $\alpha_{K-1} = (1-q)^0 = 1$. We then define

$$\Phi_i(T) \triangleq \alpha_{i-1} G(T) - C(T), \tag{21}$$

$$\Psi_i(S_i, e) \triangleq \max(0, \alpha_i G(e|S_i) - C(e|S_i)). \tag{22}$$

Then we have

$$\begin{aligned}
&\Phi_{i+1}(S_{i+1}) - \Phi_{i+1}(S_i) \\
=&\alpha_i G(S_{i+1}) - C(S_{i+1}) - \alpha_{i-1} G(S_i) + C(S_i) \\
=&\alpha_i G(S_{i+1}) - (1-q)\,\alpha_i G(S_i) - C(e|S_i) \\
=&\alpha_i (G(S_{i+1}) - G(S_i)) + q\alpha_i G(S_i) - C(e|S_i) \\
\geq&\Psi_i(S_i, e_i) + q\alpha_i G(S_i).
\end{aligned} \tag{23}$$

Thus, we can give a lower bound of $\Psi_i(S_i, e_i)$ as follows:

$$\begin{aligned}
&\Psi_i(S_i, e_i) \\
\geq &\alpha_i G(e_i|S_i) - C(e_i|S_i) \\
\geq &\frac{1}{K}\sum_{e \in OPT} \alpha_i G(e|S_i) - C(e|S_i) \\
\geq &\frac{1}{K}\alpha_i(G(OPT) - G(S_i)) - \frac{1}{K}\sum_{e \in OPT} C(e|S_i) \qquad (24) \\
\geq &\frac{1}{K}\alpha_i(G(OPT) - G(S_i)) - \frac{1}{K(1-\gamma)}C(OPT), \qquad (25)
\end{aligned}$$

where (24) follows from the submodularity and monotone of $G(S)$, i.e.,

$$\sum_{e \in OPT} G(e|S_i) \geq G(OPT \cup S_i) - G(S_i) \geq G(OPT) - G(S_i).$$

And (25) follows from the definition of γ, i.e.,

$$\frac{C(e|OPT \setminus e)}{C(e|\emptyset)} \geq \frac{C(e|\Omega \setminus e)}{C(e|\emptyset)} \geq 1 - \gamma, \qquad (26)$$

and

$$\begin{aligned}
\sum_{e \in OPT} C(e|S_i) \leq &\sum_{e \in OPT} C(e|\emptyset) \\
\leq &\sum_{e \in OPT} \frac{1}{(1-\gamma)} C(e|OPT \setminus e) \\
\leq &\frac{1}{(1-\gamma)} C(OPT). \qquad (27)
\end{aligned}$$

Next we prove the bound of *DistortedGreedy* as follows. Note that the following equation holds:

$$\Phi_0(S_0) = \alpha_{-1}G(\emptyset) - C(\emptyset) = 0, \qquad (28)$$

$$\Phi_K(S_K) = \alpha_{k-1}G(S_K) - C(S_K) = G(S_K) - C(S_K). \qquad (29)$$

Then we have

$$\begin{aligned}
G(S_K) - C(S_K) = &\Phi_K(S_K) - \Phi_0(S_0) \\
= &\sum_{i=0}^{K-1} \Phi_{i+1}(S_{i+1}) - \Phi_i(S_i) \\
\geq &\sum_{i=0}^{K-1} \Psi_i(S_i, e_i) + q\alpha_i G(S_i) \\
\geq &\sum_{i=0}^{K-1} (1-\gamma)\Psi_i(S_i, e_i) + q\alpha_i G(S_i) \qquad (30) \\
\geq &\sum_{i=0}^{K-1} (1-\gamma)(\frac{1}{K}\alpha_i(G(OPT) - G(S_i)) \\
&- \frac{1}{K(1-\gamma)}C(OPT)) + q\alpha_i G(S_i) \\
\geq &\sum_{i=0}^{K-1} q\alpha_i G(OPT) - \frac{1}{K}C(OPT) \\
\geq &\left(1 - (1-q)^K\right) G(OPT) - C(OPT) \\
\geq &(1 - e^{\gamma - 1})G(OPT) - C(OPT) \qquad (31)
\end{aligned}$$

where (30) follows from $\Psi_i(S_i, e_i) \geq 0$ and $0 \leq (1-\gamma) \leq 1$. This completes the proof. □

Theorem 1 shows that although a greedy updating procedure is employed, Algorithm 2 can still return a decent rule set once the problem (9) is solved optimally. Notice that, since the curvature γ is usually unknown in advance, we run the proposed Algorithm 2 multiple times with different γ and return the rule set with best performance.

V. Experiments

In this section, we perform experiments on benchmark datasets to study the performance of RISO in comparison with the state-of-the-art imbalanced classification and interpretable algorithms.

A. Experimental Setup

Data: Algorithms are tested on 11 public imbalanced datasets. Specifically, *FICO-binary* is from the Fair Isaac credit dataset and the rest are from the UCI machine learning repository [1]. Since in this paper we focus on the binary imbalanced classification while most of these raw datasets contain multiple classes of data, we label one class samples as the positive and the rest samples as the negative to build the binary imbalanced datasets. For those binary datasets such as *FICO-binary*, we build binary imbalanced datasets by down-sampling the positive samples from the raw datasets by a factor greater than 10. We binarize all features as in [9], which is also implemented in IBM AIX360 [1]. Specifically, categorical features are binarized using the standard one-hot encoding, such that a categorical feature u with a enumerated value u_i is binarized into $u = u_i$ and $u \neq u_i$. A numerical feature v is binarized as a set of comparison features $\{v < z_i, v \geq z_i\}$, where $\{z_i\}_{i=0}^{9}$ denote deciles of feature v. It is worth noting that the data features are not binarized for CART and RF in our experiments. To measure the imbalance degree of the dataset, we compute the imbalance ratio (IR) [17], the ratio of the number of negative to the number of positive samples, for all datasets. We note that the imbalanced ratios on all datasets are greater than 7.5 and the maximum of them is reached 47.6. The statistics of all tested datasets are summarized in Table I, which includes the number of binarized features after processing, the number of samples and positive samples, the imbalanced ratios and the class names considered as the positive one from the raw datasets.

Baseline Algorithms: We compare several state-of-the-art interpretable classification algorithms, including 1) an incremental pruning based method RIPPER[2]; 2) a column generation method CG[3] [9] minimizing classification error as well as the model complexity by formulating an integer program; 3) a submodular optimization based method IDRS [34] which maximizes the classifying accuracy of the learnt rule set. Note that in IDRS, there are corresponding weight parameters for

[1] https://archive.ics.uci.edu/ml/index.php

[2] https://github.com/imoscovitz/wittgenstein

[3] https://github.com/Trusted-AI/AIX360

TABLE I: Details of 11 imbalanced datasets.

Dataset	#features	#samples	#positive samples	IR	positive class name
gas	2304	13910	1641	7.5	3
ecoli	94	336	35	8.6	imU
abalone	132	4177	335	9.7	1
flare	66	1066	95	10.2	1
FICO-binary	34	5884	425	12.8	1
shuttle	126	43500	2458	16.7	5
car	42	1728	69	24.0	good
cardio	344	2129	81	25.3	1.0
yeast	106	1484	51	28.1	ME2
nursery	52	12960	330	38.3	1
anuran	380	7195	148	47.6	Scinax

imbalanced data; 4) Fast Boxes [11], which solves a mixed integer program to classify the imbalanced data. The result of the BRACID [26] is not included since it fails to terminate in several hours on most datasets.

Besides those interpretable algorithms, we also include two widely used classification algorithms CART and random forest (RF) to compare the classification performance. We use the implementations of the scikit-learn[4] in our experiment. To study how the data sampling methods affect the imbalanced classification performance, we combine the baseline models (not developed for imbalanced data) with SMOTE implemented in [22]. Furthermore, we combine CART with SMOTEN, RandomOverSampler and RandomUnderSampler which all implemented by [22] to do experiments more specifically and report the results in Fig. 5. Note that for IDRS combined with SMOTE, both classes are supposed to have weight one.

Parameter Tuning: We tune parameters for all methods by 10-fold cross-validation. Specifically, for our model, we choose the parameter K (number of rules) from $\{8, 12, 16, 20, 25\}$, and limit the length of a rule to 6 for better interpretability, i.e., $l = 6$. For RIPPER, we tune the pruning ratio of the training data in $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. For IDRS, we tune the penalty for negative samples in the loss function with $\beta_0 \in \{1, 8, 10, 12\}$, the penalty for rule diversity with $\beta_2 \in \{0.5, 0.1, 0.01\}$, the model complexity with $\lambda \in \{0.01, 0.1, 1, 4, 8, 16, 64\}$ and the maximal number of rules with $K \in \{8, 16, 32\}$. We fix the penalty for uncovered positive samples as $\beta_1 = 1.0$. For CG, we tune the cost of each clause λ_0 and the additional cost for each literal λ_1 with $\lambda_0 \in \{0.001, 0.0001, 0.0005\}$ and $\lambda_1 \in \{0.0001, 0.0005, 0.0008\}$. The beam search width in CG is chosen from $\{15, 20\}$. For CART and RF, the number of samples at each leaf node is tuned from 1 to 100.

B. Experimental Results

Here we summarize all experimental results obtained from 10-fold stratified cross-validation implemented in scikit-learn with *random_state=42* and *shuffle=True*.

Performance: We summarize the F1 scores of all baseline methods in Table II. For models not specifically designed for imbalanced datasets, although some of them support sample weights, we report the performance of all models after combining with the SMOTE implemented by [22] at the bottom of each line in Table II for consistency. To further compare these interpretable methods, we report recall/precision/F1 score of rule-based methods, including RISO, IDRS, RIPPER, and CG without SMOTE in Fig. 4.

From the results shown in Table II and Fig. 4, we summarize our observations as follows:

1) RISO achieves the highest F1 scores among all interpretable algorithms on 8 imbalanced datasets. For the rest datasets, the difference between our method and the best one is marginal. Note that the standared errors might not be ignored as the distributions of imbalanced datasets can vary greatly over different folds. Comparing with other rule based models, RISO performes best in the average ranking of 10 fold stratified cross validation experiments.
2) The re-sampling methods bring significant benefit not for all datasets and all models. This is because the effects of noise and borderline samples generated cannot be eliminated. More experiments about the data re-sampling methods are demonstrated in Fig. 5.
3) RISO has obvious advantages to other rule based methods, especially For highly imbalanced datasets (IR$\geq$20).
4) Some rule based methods such as RIPPER and CG fail to handle imbalanced datasets. IDRS is relatively robust to the data imbalance by tuning the sample weights in the loss function, but still worse than our method.
5) RISO achieves higher *recall* on 8 datasets than RIPPER and CG as shown in Fig. 4. When compared with IDRS, RISO can reach a higher *precision* at the comparable *recall*.

Interpretablity: For interpretable methods, the average number of rules learned by the corresponding white-box methods are reported in Table III. Fast Boxes is not available on *nursery*, *shuttle* and *gas* because it ran beyond our time limit. It can be observed that RISO typically learns more rules than other rule based methods on most datasets, which confirms our claim that methods based on accuracy maximization may miss small positive clusters, which in turn weakens the performance. CART usually learns more rules on most datasets than rule based models.

In addition, our algorithm is more interpretable under the same classification performance, which is concluded by varying the number of rules N on the datasets and training the corresponding models. We illustrate the specific F1 score and the number of rules on dataset *car* in Fig.6a. It is observed that, in general, the training F1 score increases as the rule number N increases. Our method learns the least number of rules to achieve the same F1 score among all methods. For CART and RF, we consider each path from the root to the leaf as one rule. RF obviously contains much more rules because there

[4] https://scikit-learn.org/

TABLE II: Mean test F1 score(%, standard error in parentheses). Imbalanced datasets are sorted in an ascending order by IR. For IDRS, RIPPER, CG and CART, we report the experiment result of the models *without SMOTE* (reported at the top of each line) and *with SMOTE* (at the bottom of each line) on each dataset. **Bold**: Best overall interpretable models.

Dataset(*IR*)	RISO	IDRS	RIPPER	CG	Fast Boxes	CART	RF

gas(*7.5*)	**98.1**(0.3)
	/
		96.3(1.5)
		95.9(0.4)
			96.6(0.9)
			97.8(1.7)
				90.9(2.0)
				97.7(0.9)
					N/A
					/
						95.9(1.2)
						96.1(1.3)
							98.3(0.8)
							98.8(0.6)

ecoli(*8.6*)	**64.9**(12.8)
	/
		62.6(15.1)
		58.4(14.9)
			51.9(33.8)
			52.9(33.7)
				55.3(26.0)
				62.1(26.0)
					60.9(12.1)
					/
						59.8(16.4)
						53.9(11.7)
							62.3(15.2)
							61.9(18.2)

abalone(*9.7*)	36.7(4.2)
	/
		33.5(2.9)
		31.9(3.0)
			19.6(9.2)
			14.7(9.0)
				7.8(5.9))
				29.2(6.1)
					20.4(0.7)
					/
						37.1(3.6)
						30.6(6.3)
							49.3(5.1)
							41.2(4.3)

flare(*10.2*)	**44.3**(8.5)
	/
		40.2(7.1)
		35.3(7.2)
			14.0(12.8)
			18.7(12.6)
				29.7(14.7)
				37.9(14.5)
					26.8(4.3)
					/
						38.5(6.2)
						34.7(8.6)
							45.7(6.1)
							36.9(10.5)

FICO-binary(*12.8*)	**28.9**(5.4)
	/
		28.2(2.6)
		21.0(2.4)
			9.7(7.9)
			7.9(8.0)
				2.1(3.1)
				26.2(4.3)
					14.3(0.3)
					/
						24.2(1.8)
						20.3(3.6)
							26.6(2.5)
							21.0(3.3)

shuttle(*16.7*)	99.9(0.3)
	/
		99.3(0.5)
		99.1(0.5)
			77.3(2.5)
			99.1(2.3)
				99.0(0.5).
				99.3(0.6)
					N/A
					/
						99.9(0.1)
						99.9(0.1)
							99.9(0.1)
							99.9(0.2)

car(*24.0*)	87.6(10.8)
	/
		79.5(19.1)
		63.6(20.3)
			64.5(14.1)
			61.5(14.3)
				70.9(15.9)
				68.9(16.0)
					82.4(7.7)
					/
						88.4(5.9)
						88.9(9.0)
							86.5(7.5)
							89.9(8.3)

cardio(*25.3*)	**83.8**(10.4)
	/
		78.4(13.7)
		60.2(13.7)
			81.0(9.7)
			79.3(9.8)
				75.7(14.1)
				66.5(13.9)
					65.4(6.9)
					/
						73.3(13.2)
						73.0(11.1)
							82.8(13.3)
							86.5(7.1)

yeast(*28.1*)	**41.3**(18.9)
	/
		32.2(18.4)
		30.3(18.4)
			20.7(16.7)
			28.8(16.8)
				25.1(21.6)
				28.4(21.4)
					17.5(3.2)
					/
						26.5(12.7)
						32.3(11.7)
							39.9(18.4)
							36.9(17.0)

nursery(*38.3*)	**100.0**(0.)
	/
		98.6(2.0)
		98.6(1.8)
			94.3(3.3)
			89.5(3.3)
				88.7(4.6)
				74.9(4.7)
					N/A
					/
						98.6(2.0)
						95.9(2.5)
							94.9(2.8)
							95.8(2.4)

anuran(*47.6*)	**93.3**(5.1)
	/
		85.7(6.6)
		78.9(6.7)
			86.2(8.9)
			84.8(8.9)
				85.3(12.6)
				77.6(13.0)
					64.0(6.3)
					/
						82.1(10.8)
						87.5(6.9)
							91.6(4.5)
							92.8(3.7)

TABLE III: Average number of rules (standard error in parentheses) learned for interpretable methods.

Dataset	RISO	RIPPER	IDRS	CG	FastBoxes	CART
ecoli	7.8(0.4)	1.9(0.7)	3.0(1.0)	3.5(1.5)	2.0(0.)	7.2(8.5)
abalone	19.2(3.9)	4.9(1.8)	11.2(4.4)	7.4(1.5)	11.8(2.4)	73.3(58.3)
flare	12.5(1.4)	1.8(0.8)	6.0(1.6)	10.1(3.5)	2.0(0.)	36.7(11.7)
gas	10.1(0.2)	13.0(1.7)	7.4(3.3)	4.0(0.)	N/A	77.6(4.7)
FICO-binary	20.0(4.4)	2.9(0.7)	7.2(2.2)	6.2(1.1)	16.2(4.9)	39.1(6.4)
shuttle	8.0(0.)	9.4(2.5)	5.6(2.2)	2.0(0.)	N/A	4.5(0.8)
car	7.9(0.3)	6.0(1.2)	6.8(1.6)	9.7(5.1)	6.4(2.5)	35.0(4.9)
cardio	9.6(2.0)	6.4(1.3)	6.1(2.1)	6.4(2.2)	3.4(0.9)	37.9(4.4)
yeast	7.8(0.6)	3.2(1.0)	11.8(4.8)	8.8(4.6)	2.2(0.6)	39.8(11.8)
nursery	9.0(0.8)	16.0(0.9)	9.3(1.7)	9.7(0.8)	N/A	65.1(4.3)
anuran	6.2(0.9)	7.1(1.0)	3.2(0.4)	1.0(0.)	2.0(0.)	33.8(3.7)

are many trees in the model.

Scalability: Note that the computational complexity of our algorithm is dominated by the local search procedure in Algorithm 1, which is linear to the number of features. To show it empirically, we conduct experiments on *anuran* and *cardio* datasets to demonstrate the relationship between model training time and the number of features. Specifically, we train the model and summarize the training time by randomly sampling P features in Fig.6b. It is shown that the training time increases linearly as the dimension increases. Hence, our algorithm is scalable in terms of feature dimension, making it an ideal tool to process large-scale imbalanced data.

C. Case Study

We show one rule set learned on dataset *car* as an example in Table IV. The rule set in Table IV is to predict whether the car is good. The learned rules imply that a car with low buying, low maint, both persons and doors not equal to 2, big lug_boot and median safety is good. These rules can help the decision makers classify a sample distinctly.

TABLE IV: An example of learned rule set.

dataset: car, 8 rules in the rule set
buying != high AND buying != vhigh AND maint == low
buying == low AND maint != high AND maint != vhigh
buying == low AND maint != high AND maint != vhigh
buying != high AND buying != vhigh AND maint == low
buying == med AND maint == low AND doors != 2
buying == med AND maint == low AND persons != 2
buying == low AND maint == med AND doors != 4
buying == low AND maint == med AND persons != 2

VI. Conclusion

In this paper, we propose an interpretable and effective rule set learning algorithm RISO to directly optimize the F1 score, which is particularly applicable for highly imbalanced classification. Our experimental results demonstrate the superior classification performance and interpretability of RISO in comparison with existing methods. In addition, the good scalability of RISO makes it an ideal tool to handle large-scale datasets in many real-world scenarios involving imbalanced datasets.

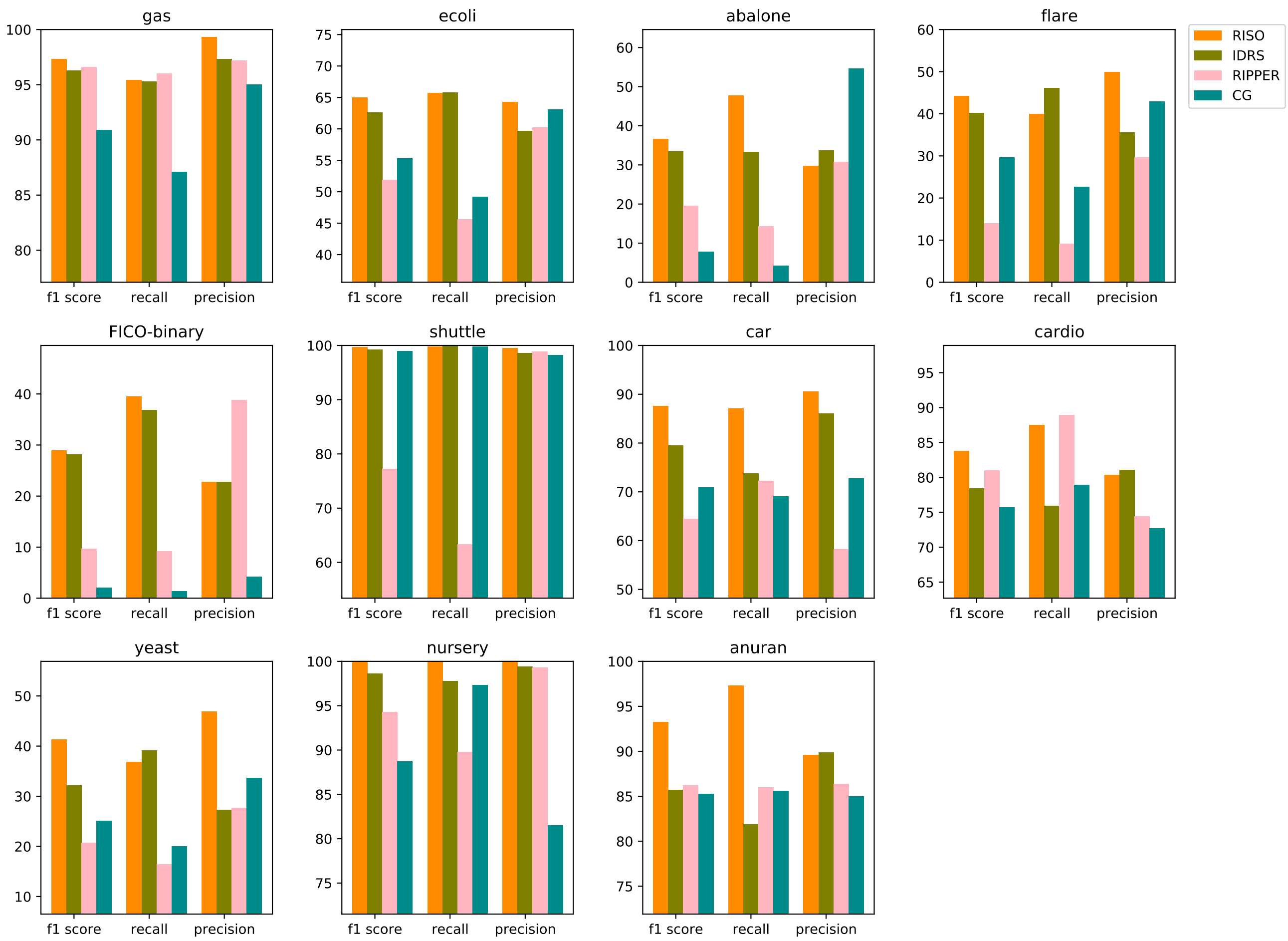


Fig. 4: Mean test F1 score, recall and precision(%) on 11 imbalanced datasets for rule-based methods.

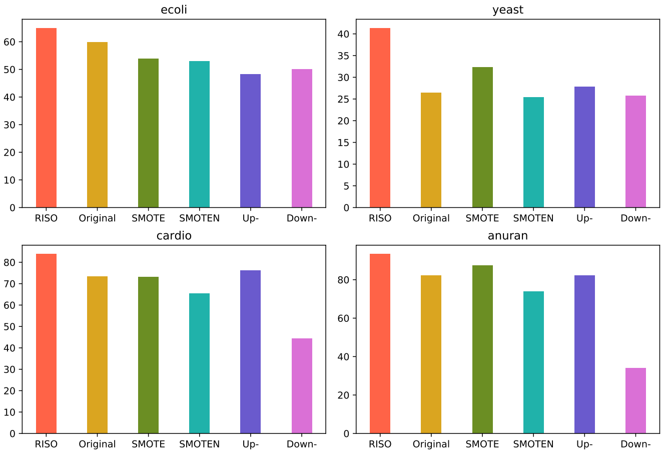
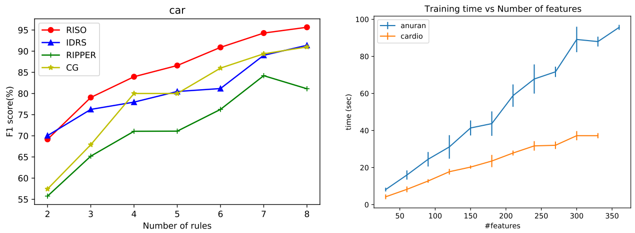


Fig. 5: Comparison of different data re-sampling methods applied to **CART** on *ecoli*, *yeast*, *cardio*, *anuran* datasets. The "Original" represents CART without data re-sampling and the "SMOTE", "SMOTEN", "Up-", "Down-" respectively represents SMOTE, SMOTEN, RandomOverSampler and RandomUnderSampler method implemented in package [22].



(a) Interpretability

(b) Scalability

Fig. 6: Interpretability and Scalability Study.

References

[1] Vijay Arya, Rachel K. E. Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Q. Vera Liao, Ronny Luss, Aleksandra Mojsilović, Sami Mourad, Pablo Pedemonte, Ramya Raghavendra, John Richards, Prasanna Sattigeri, Karthikeyan Shanmugam, Moninder Singh, Kush R. Varshney, Dennis Wei, and Yunfeng Zhang. One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques. *arXiv preprint arXiv:1909.03012*, 2019.

[2] Francis Bach et al. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends® in Machine Learning*, 6(2-3):145–373, 2013.

[3] Wenruo Bai, Rishabh Iyer, Kai Wei, and Jeff Bilmes. Algorithms for optimizing the ratio of submodular functions. In *Proceedings of the 33rd ICML*, volume 48, pages 2751–2759. PMLR, 2016.

[4] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 2002.

[5] Nitesh V Chawla, Nathalie Japkowicz, and Aleksander Kotcz. Editorial: Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, 6(1):1–6, 2004.

[6] Nitesh V Chawla, Aleksandar Lazarevic, Lawrence O Hall, and Kevin W Bowyer. SMOTEBoost: Improving prediction of the minority class in boosting. In *European conference on principles of data mining and knowledge discovery*, pages 107–119. Springer, 2003.

[7] Peter Clark and Tim Niblett. The CN2 induction algorithm. *Machine Learning*, 3(4), 1989.

[8] William W Cohen. Fast effective rule induction. In *Machine learning proceedings 1995*. Elsevier, 1995.

[9] Sanjeeb Dash, Oktay Gunluk, and Dennis Wei. Boolean decision rules via column generation. In *Advances in Neural Information Processing Systems*, 2018.

[10] Jerome H Friedman and Bogdan E Popescu. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3):916–954, 2008.

[11] Siong Thye Goh and Cynthia Rudin. Box drawings for learning with imbalanced data. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 333–342, 2014.

[12] Hongyu Guo and Herna L Viktor. Learning from imbalanced data sets with boosting and data generation: the databoost-im approach. *ACM SIGKDD Explorations Newsletter*, 6(1):30–39, 2004.

[13] Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73:220–239, 2017.

[14] Chris Harshaw, Moran Feldman, Justin Ward, and Amin Karbasi. Submodular maximization beyond non-negativity: Guarantees, fast algorithms, and applications. In *Proceedings of the 36th ICML*, volume 97, pages 2634–2643. PMLR, 2019.

[15] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, 2009.

[16] S Jegelka and J Bilmes. Submodularity beyond submodular energies: Coupling edges in graph cuts. In *Proceedings of the 2011 IEEE Conference on CVPR*, 2011.

[17] Justin M. Johnson and Taghi M. Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(27), 2019.

[18] Mahesh V Joshi, Vipin Kumar, and Ramesh C Agarwal. Evaluating boosting algorithms to classify rare classes: Comparison and improvements. In *Proceedings 2001 IEEE international conference on data mining*, pages 257–264. IEEE, 2001.

[19] György Kovács. Smote-variants: A python implementation of 85 minority oversampling techniques. *Neurocomputing*, 366:352–354, 2019.

[20] Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD*, KDD '16, page 1675–1684, 2016.

[21] Jon Lee, Vahab S Mirrokni, Viswanath Nagarajan, and Maxim Sviridenko. Maximizing nonmonotone submodular functions under matroid or knapsack constraints. *SIAM Journal on Discrete Mathematics*, 23(4), 2010.

[22] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017.

[23] Victoria López, Sara Del Río, José Manuel Benítez, and Francisco Herrera. Cost-sensitive linguistic fuzzy rule based classification systems under the mapreduce framework for imbalanced big data. *Fuzzy Sets and Systems*, 258:5–38, 2015.

[24] Tomasz Maciejewski and Jerzy Stefanowski. Local neighbourhood extension of smote for mining imbalanced data. In *2011 IEEE symposium on CIDM*, 2011.

[25] Graziano Mita, Paolo Papotti, Maurizio Filippone, and Pietro Michiardi. Libre: Learning interpretable boolean rule ensembles. In *Proceedings of the 23rd AISTATS*, 2020.

[26] Krystyna Napierala and Jerzy Stefanowski. BRACID: a comprehensive approach to learning rules from imbalanced data. *Journal of Intelligent Information Systems*, 39(2):335–373, 2012.

[27] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978.

[28] Pierre Perrault, Jennifer Healey, Zheng Wen, and Michal Valko. On the approximation relationship between optimizing ratio of submodular (RS) and difference of submodular (DS) functions. *arXiv preprint arXiv:2101.01631*, 2021.

[29] Jerzy Stefanowski and Szymon Wilk. Extending rule-based classifiers to improve recognition of imbalanced classes. In *Advances in Data Management*, pages 131–154. Springer, 2009.

[30] Yanmin Sun, Andrew KC Wong, and Mohamed S Kamel. Classification of imbalanced data: A review. *International journal of pattern recognition and artificial intelligence*, 23(04):687–719, 2009.

[31] Benjamin X Wang and Nathalie Japkowicz. Boost-

ing support vector machines for imbalanced data sets. *Knowledge and information systems*, 25(1), 2010.

[32] Tong Wang and Cynthia Rudin. Learning optimized or's of and's. *arXiv preprint arXiv:1511.02210*, 2015.

[33] Tong Wang, Cynthia Rudin, Finale Doshi-Velez, Yimin Liu, Erica Klampfl, and Perry MacNeille. A Bayesian framework for learning rule sets for interpretable classification. *Journal of Machine Learning Research*, 18(70):1–37, 2017.

[34] Fan Yang, Kai He, Linxiao Yang, Hongxia Du, Jingbang Yang, Bo Yang, and Liang Sun. Learning interpretable decision rule sets: A submodular optimization approach. In *Advances in Neural Information Processing Systems*, volume 34, pages 27890–27902, 2021.

[35] Show-Jane Yen and Yue-Shi Lee. Cluster-based undersampling approaches for imbalanced data distributions. *Expert Systems with Applications*, 36(3):5718–5727, 2009.

[36] Guangyi Zhang and Aristides Gionis. Diverse rule sets. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, KDD '20, page 1532–1541, 2020.