

Interpretable Decision Sets: A Joint Framework for Description and Prediction

Himabindu Lakkaraju
Stanford University
himalv@cs.stanford.edu

Stephen H. Bach
Stanford University
bach@cs.stanford.edu

Jure Leskovec
Stanford University
jure@cs.stanford.edu

ABSTRACT

One of the most important obstacles to deploying predictive models is the fact that humans do not understand and trust them. Knowing which variables are important in a model's prediction and how they are combined can be very powerful in helping people understand and trust automatic decision making systems.

Here we propose interpretable decision sets, a framework for building predictive models that are highly accurate, yet also highly interpretable. Decision sets are sets of independent if-then rules. Because each rule can be applied independently, decision sets are simple, concise, and easily interpretable. We formalize decision set learning through an objective function that simultaneously optimizes accuracy and interpretability of the rules. In particular, our approach learns short, accurate, and non-overlapping rules that cover the whole feature space and pay attention to small but important classes. Moreover, we prove that our objective is a non-monotone submodular function, which we efficiently optimize to find a near-optimal set of rules.

Experiments show that interpretable decision sets are as accurate at classification as state-of-the-art machine learning techniques. They are also three times smaller on average than rule-based models learned by other methods. Finally, results of a user study show that people are able to answer multiple-choice questions about the decision boundaries of interpretable decision sets and write descriptions of classes based on them faster and more accurately than with other rule-based models that were designed for interpretability. Overall, our framework provides a new approach to interpretable machine learning that balances accuracy, interpretability, and computational efficiency.

1. INTRODUCTION

When using machine learning to build models from data, often it is not sufficient for a model to simply be accurate at predicting on held-out test data. Many applications require that models be *interpretable*, meaning that humans can easily understand the information they contain [12, 20, 43, 47]. For classification models, this requirement means that humans can understand where the decision boundaries between classes are and why particular labels are

predicted for different data points [44]. When there are multiple classes to predict, it is also important to characterize all the classes, not just the common ones.

Interpretable models are needed in many domains because they bridge the gap between domain experts and data scientists. When domain experts need to make important decisions, learning from data can improve their results, but this requires the human to understand and trust the model. From medical diagnosis to decision making in the justice and education systems, the ability to interpret a model enables decision makers to critique, refine, and ultimately trust it based on their expertise [39]. As machine learning is applied to new societal and high-stakes problems, the need for interpretable models will only continue to grow in the future.

Learning interpretable models is challenging because interpretability and accuracy are generally two competing objectives, one favoring simplicity and generalization, the other favoring nuance and exception. Further, even quantifying interpretability is a challenge. A popular approach to interpretable models is rule-based models, such as decision trees [8, 42] and decision lists [45], because they strike a balance between the two objectives. Their benefit is that they are stated in terms of the input features, without relying on any latent variables or representations, and they use concise, logical rules to make interpretable predictions. Just being rule-based, however, is not sufficient; the structure connecting a set of rules is also an important factor in interpretability. For example, decision lists [5, 34] make a prediction whenever a rule is true. This restricted structure can be thought of equivalently as a list of if-then-else statements, and is considered more interpretable than a general decision tree because of its reduced complexity. However, decision lists still have drawbacks. The chaining of rules via if-then-else statements means that new rules can only cover increasingly narrow slices of the feature space. Even if the first few rules in the list are interpretable, each additional rule becomes less so because it only applies to more and more narrow situations, *i.e.*, a given rule only applies to those data points that do not satisfy any of the preceding rules. Understanding all of the conditions that must be satisfied before a rule can be applied is difficult, limiting the interpretability of decision lists. Further, in multi-class classification, most of the classes can only be described by more narrow rules because they appear further down the list. In effect, some classes are only described as alternatives after other classes have been considered, a serious problem for domain experts who want equally good rules for all the classes in the data.

Present work: Interpretable decision sets. Here we propose a new framework, called *interpretable decision sets* (Figure 1 (left)), for learning decision sets that are interpretable, accurate, and address the shortcomings of previous approaches [34, 35]. Decision sets take a different approach to structuring classification rules.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD '16, August 13-17, 2016, San Francisco, CA, USA

© 2016 ACM. ISBN 978-1-4503-4232-2/16/08...\$15.00

DOI: <http://dx.doi.org/10.1145/2939672.2939874>

```

If Respiratory-Illness=Yes and Smoker=Yes and Age $\geq$  50 then Lung Cancer
If Risk-LungCancer=Yes and Blood-Pressure $\geq$  0.3 then Lung Cancer
If Risk-Depression=Yes and Past-Depression=Yes then Depression
If BMI $\geq$  0.3 and Insurance=None and Blood-Pressure $\geq$  0.2 then Depression
If Smoker=Yes and BMI $\geq$  0.2 and Age $\geq$  60 then Diabetes
If Risk-Diabetes=Yes and BMI $\geq$  0.4 and Prob-Infections $\geq$  0.2 then Diabetes
If Doctor-Visits  $\geq$  0.4 and Childhood-Obesity=Yes then Diabetes

```

```

If Respiratory-Illness=Yes and Smoker=Yes and Age $\geq$  50 then Lung Cancer
Else if Risk-Depression=Yes then Depression
Else if BMI  $\geq$  0.2 and Age $\geq$  60 then Diabetes
Else if Headaches=Yes and Dizziness=Yes, then Depression
Else if Doctor-Visits $\geq$  0.3 then Diabetes
Else if Disposition-Tiredness=Yes then Depression
Else Diabetes

```

Figure 1: An interpretable decision set, our proposed approach (left), and a decision list [45] (right) learned from the same medical diagnosis dataset. Decision sets (left) are more comprehensible to humans because rules apply independently. In decision lists (right), rules implicitly depend on all the rules above it not being true. Thus, while the order of the rules in decision lists is crucial, it does not matter for decision sets.

Rather than organizing them in a hierarchy, decision sets are collections of if-then rules that can be considered in any order. Any data point not covered by any rule or covered by rules indicating different classes is assigned a default label. This non-hierarchical structure is the key to interpretability. Without any if-then-else structure to describe narrow cases, each rule in a decision set must be an accurate predictor in isolation. Humans can therefore read decision sets and understand how each rule works, one at a time.

Learning an interpretable decision set is challenging because it requires finding an accurate model within a space specifically designed for interpretability. To balance these competing goals, we introduce a learning objective that scores both accuracy, measured by the precision and recall of the rules; and interpretability, measured by the conciseness, coverage, and overlap of the rule set. An important part of this objective is favoring decision sets with rules that overlap as a little as possible. We show that exactly optimizing the objective is an NP-hard problem. However, we also show that the objective has a particular structure which allows for provably near-optimal solutions. In particular, we show that learning interpretable decision sets is an instance of a non-monotone sub-modular maximization problem. Such problems can be efficiently optimized with an algorithm that guarantees a near-optimal solution which will be at least $2/5$ of the optimal solution [18]. Thus, in contrast with previous rule-based methods that construct a model in a greedy, incremental way, our approach guarantees a globally near-optimal set of rules.

To ensure that our method can scale to big data, we pre-mine the space of rules that we will search via association rule mining [1] techniques, which massively reduces computation. As long as the pre-mined set of rules is sufficiently comprehensive, the expressive power of the decision set is equivalent to that of a decision tree. The result is that we can learn accurate and interpretable decision sets for data with hundreds of thousands of instances in a few minutes.

We evaluate interpretable decision sets on three real-world classification problems: (1) diagnosing patients who have one of six different diseases, (2) identifying high-school students that are at risk of dropping out or not earning their diplomas on time, and (3) predicting which defendants are more likely to commit a violent or nonviolent crime if they are released on bail. These problems all require interpretable models because they involve important decisions by domain experts that have the potential to significantly affect people’s lives. They also are multi-class problems with a mix of common classes and rare but important ones. Experiments show that interpretable decision sets are more accurate classifiers than Bayesian decision lists [34], and come close to matching the accuracy of uninterpretable models like random forests. We analyze the

learned models using multiple ways of quantifying interpretability, and show that the decision sets have simpler, more interpretable decision boundaries. For example, we find that interpretable decision sets have up to 62% fewer rules and 63% shorter rules than state-of-the-art methods for learning decision lists [34, 35].

We also conduct a user study in which we ask participants to interpret models learned during our empirical evaluation. We compare decision sets and decision lists using two types of tasks: answering multiple-choice questions about the decision boundaries of the models and writing descriptions of classes based on them. We find that the structure of decision sets enables users to better understand the predictions they will make. Humans were 17% more accurate when answering multiple-choice questions about the decision boundaries of a decision set versus a decision list. They were also 22% faster at answering the questions. Further, we find that users interpret both decision sets and decision lists in similar ways, most often writing descriptions of classes as a set of possible conditions without any structure connecting them. This is correct for decision sets but not decision lists, so humans were three times more accurate given a decision set versus a decision list, and they used 74% fewer words and 71% less time to write their descriptions. Our results indicate that decision sets are a more natural and interpretable way of presenting class information to humans.

To summarize, we show that decision sets are a more interpretable way of organizing classification rules. They can be just as accurate as more complicated rule structures, but are easier for humans to understand. Users can interpret their decision boundaries, not just the mechanism that produced an individual prediction, in contrast with previous work. As machine learning and data mining are used more and more to address high-stakes problems, decision sets offer a new, more interpretable method for making important decisions using data.

2. RELATED WORK

Research on interpretable models has a long history [9, 12, 43, 44], but with the availability of ever larger datasets and the adoption of computer-based decision making in all parts of society, the need for interpretable machine learning models has only increased. Recent efforts have sought models that offer both high predictive accuracy and interpretability. For example, prototype vector machines [6], regression models [47, 49], generalized additive models [23, 36, 37, 55], Bayesian case models [27], mind-the-gap models [28], and scoring systems [50] have all been proposed as interpretable models.

Our work here is motivated by rule-based methods. One family of such methods starts with association rule mining methods [1]

to identify a set of rules and then sort them in order to identify the most predictive ones [54]. The resulting ordered rules are instances of decision lists [10, 11, 13, 45, 52, 56]. However, these methods focus only on maximizing classification accuracy and aim to achieve interpretability just by building the model from rules. Similarly, algorithms for problems such as subgroup discovery [24, 33, 40, 46], contrast set learning [3, 4, 30], and emerging pattern mining [17, 19] identify sets of rules to describe the relationships among variables and discover interesting patterns in the data. In contrast, our work explicitly defines an objective function that scores interpretability and accuracy, and by optimizing it, we find a globally near-optimal model.

Learning decision lists directly has been well studied [7, 15, 29, 51] and has been used for building parsimonious models [5, 34]. However, as we show in our user study, decision lists (Figure 1 (right)) have limited interpretability as rules are coupled by the “else” statements. In particular, reasoning about a rule in a decision list requires reasoning about all of the preceding rules as well, because a rule applies only when none of the preceding rules apply. For example, the third rule in Figure 1 (right): **If BMI ≥ 0.2 and Age ≥ 60 then Diabetes**, appears simple and easy to interpret. However, the rule only applies when the previous two rule conditions are false, so the correct interpretation of the third rule is actually: **If not (Respiratory-Illness=Yes and Smoker=Yes and Age ≥ 50) and not (Risk-Depression=Yes) and BMI ≥ 0.2 and Age ≥ 60 , then Diabetes**.

Interpretable decision sets (Figure 1 (left)) do not suffer from the above problem and thus have greater interpretability. Rules in decision sets are self-contained and apply individually, which means that no complicated reasoning about multiple rules is needed. As we show in Section 5.4, decision sets are much more interpretable to users and just as accurate.

Prior work has also considered learning unordered sets of rules, but has not considered how to minimize the overlap of the rules in order to maximize interpretability. One approach [14] modified the CN2 algorithm [15] for learning decision lists to produce rule sets using simple heuristics to grow the set greedily. In contrast, our approach finds globally near-optimal sets of rules that are on average six times smaller than a decision list trained on the same data, while also being much more easily understandable. Other methods attempt to construct sets with fewer rules [38, 48, 53], but again do not account for rule overlap. In contrast, interpretable decision sets also account for rule overlap and by optimizing a joint objective find a near-optimal set of rules.

3. INTERPRETABLE DECISION SETS

In this section, we define decision sets and formalize the properties that characterize them. These properties will form the basis of our approach to learning interpretable decision sets.

3.1 Decision Sets

We propose decision sets as a model class that can both accurately predict class labels and interpretably describe its decision boundaries. Decision sets are sets of classification rules, *i.e.*, the rules are *unordered* (see example in Figure 1). Although similar in appearance to decision lists, the rules crucially are not connected by “else” statements. Each rule is an independent classifier that can assign its label without regard for any other rules. The expressive power of decision sets, decision lists, and decision trees are all equivalent because the decision boundaries captured by decision trees and lists can also be represented by rules in decision sets.

We define decision sets using *itemsets*. An itemset s is a filter for data points, defined as a conjunction of one or more *predicates* of

Notation	Definition	Term
\mathcal{D}	Input set of data points $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$	Dataset
\mathbf{x}	Observed attribute values of a data point	
y	Class label of a data point	
\mathcal{C}	Set of class labels in \mathcal{D}	
p	(attribute, operator, value) tuple, <i>e.g.</i> , Age ≥ 50	Predicate
s	Conjunction of one or more predicates, <i>e.g.</i> , Age ≥ 50 and Gender = Female	Itemset
\mathcal{S}	Input set of itemsets	
r	Itemset-class pair (s, c)	Rule
\mathcal{R}	Set of rules $\{(s_1, c_1), \dots, (s_k, c_k)\}$	Decision set

Table 1: Summary of notation.

the form (attribute, operator, value), *e.g.*, $(x_1 > 5)$. Given attribute values \mathbf{x} , we say that \mathbf{x} *satisfies* s if all the predicates in s are true when evaluated on \mathbf{x} . A *rule* is a tuple (s, c) where s is an itemset and c is a class label. We now state a formal definition of decision sets.

Definition 1. A **decision set** \mathcal{R} is a set of rules of the form (s, c) where s is an itemset and c is a class label. A decision set assigns a class label c to attribute values \mathbf{x} as follows. If \mathbf{x} satisfies exactly one itemset s_i , then its class label is the corresponding c_i . If \mathbf{x} satisfies zero itemsets then its class label is assigned using a default label, and if \mathbf{x} satisfies more than one itemset, it is assigned a class using a tie-breaking function.

The choice of default class label and class tie-breaking function is up to the user. For our experiments, we report results with two simple choices. For data points that satisfy zero itemsets, we predict the majority class label in the training data, and for data points that satisfy more than one itemset, we predict using the rule with the highest F1 score on the training data. However, other choices of default class labels and class tie-breaking functions can be easily incorporated. For example, we could use the smallest minority class as the default label. This would be appropriate for a problem in which recall is very important, such as fraud detection. Similarly, one could break ties using a majority vote for data points that satisfy more than one itemset. Regardless of a user’s choice, the default label and tie-breaking function tend to be applied infrequently. Across the datasets in our experiments, either was used from 14% to 22% of the time.

3.2 Properties of Interpretable Decision Sets

We now define the properties of decision sets that we will optimize in order to balance interpretability and accuracy. We have already taken a significant step towards interpretability by choosing to learn decision sets. As we will show in our experimental evaluation, their if-then structure without any connecting “else” state-

ments enables users to easily reason about the decision boundaries of classes. However, we still need to define the properties we optimize to find the best decision set for a particular dataset. Note that many of these properties are defined with respect to a given dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$, where each \mathbf{x}_i is a tuple of attribute values and y_i is the corresponding label. When we use these properties to learn interpretable decision sets in Section 4, they will be computed with respect to the training data. We group the properties based on whether they relate to interpretability or accuracy.

3.2.1 Interpretability

Decision sets are naturally interpretable because of their simplicity, but there is still a cognitive limit on how complex a model can be while still also being understandable. We capture the interpretability of a decision set by defining four natural metrics: *size*, *length*, *cover*, and *overlap*. Minimizing *size* encourages decision sets with a small number of rules. Minimizing *length* captures the notion that interpretable rules are short and concise. We use *cover* to denote how many datapoints satisfy the itemset of a rule, which is necessary for defining subsequent metrics. Finally, minimizing *overlap* encourages each rule to cover an independent part of the feature space.

We shall next describe each of the four metrics that when combined produce interpretable decision sets.

Size. First, we consider the size of the decision set itself. The fewer the rules in a decision set, the easier it is for a user to understand all of the conditions that correspond to a particular class label.

Definition 2. $\text{size}(\mathcal{R})$ is the number of rules in a decision set \mathcal{R} .

Length. Second, we consider the size of rules in a decision set. Logical functions are generally easy for humans to understand [16], so it is natural to use them to specify itemsets. However, if the number of predicates in the itemset of a rule is too large, it will lose its natural interpretability. We use the term *length* to measure the size of a rule.

Definition 3. $\text{length}(r)$ for some rule $r = (s, c)$ is the number of predicates in the itemset s .

We will use these two properties to measure the conciseness of a decision set, a key component of its interpretability.

However, conciseness alone is not sufficient for an interpretable decision set. Interpretability also requires providing the user with a clear description of decision boundaries for all the classes in the data.

Cover. The first step in assessing the clarity of a decision set's description is measuring which points in the data set it covers, which we define on a per-rule basis.

Definition 4. $\text{cover}(r)$ for a rule $r = (s, c)$ is the set of data points in \mathcal{D} with attribute values \mathbf{x} that satisfy the itemset s .

Note that whether a point \mathbf{x}_i is in $\text{cover}(r)$ does not depend on the observed label y_i . Measuring cover will allow us to reason about how the data set is divided by rules.

Overlap. We measure whether the decision boundaries of a decision set are clearly defined, via the overlap of rules:

Definition 5. $\text{overlap}(r, r')$ for rules $r = (s, c)$ and $r' = (s', c')$ is the set of data points that satisfy both s and s' :

$$\text{overlap}(r, r') = \text{cover}(r) \cap \text{cover}(r').$$

Overlap is a very important property for decision sets. Anytime that two rules overlap at a data point, then the tie-breaking function must be invoked to make a prediction and interpretability is reduced. Therefore, we will seek rules that overlap as little possible, ensuring that users can understand where the decision boundaries of the model are. While metrics accounting for conciseness have been studied in the context of interpretable rule-based models [34, 53], the property of overlap has not been previously considered.

3.2.2 Accuracy

A decision set must be an accurate predictor of the class labels. We can measure a decision set's overall accuracy by evaluating it on test data, but we also need to measure the accuracy of its rules individually during learning. We define two properties to measure per-rule accuracy.

Definition 6. $\text{correct-cover}(r)$ for a rule $r = (s, c)$ is the set of data points in \mathcal{D} which satisfy s and belong to class c :

$$\text{correct-cover}(r) = \{(\mathbf{x}, y) \in \text{cover}(r) \mid y = c\}.$$

Definition 7. $\text{incorrect-cover}(r)$ for a rule $r = (s, c)$ is the set of data points which satisfy s and do not belong to class c :

$$\text{incorrect-cover}(r) = \text{cover}(r) \setminus \text{correct-cover}(r).$$

Defining these properties enables us to reason about how each rule contributes to the precision and recall of the entire decision set. If $|\text{correct-cover}(r)|$ is high and $|\text{incorrect-cover}(r)|$ is low, then the rule r is precise and contributes towards a higher precision for the decision set. (How much it contributes depends on $|\text{cover}(r)|$.) On the other hand, having a high recall requires many data points to be correctly covered. We say that a point (\mathbf{x}, y) is correctly covered by a rule r if $(\mathbf{x}, y) \in \text{correct-cover}(r)$.

4. LEARNING DECISION SETS

In this section we present our framework for learning decision sets that jointly maximizes their interpretability and predictive accuracy. We assume that we are given a training set \mathcal{D} , a set of itemsets \mathcal{S} , and a set of possible class labels \mathcal{C} . Our goal is to find a decision set that makes accurate predictions with clearly described decision boundaries by balancing the properties introduced in Section 3.2. We search over subsets of $\mathcal{S} \times \mathcal{C}$ (itemset and class label pairs) to define our decision set. Note that we do not assume a particular structure for the itemsets \mathcal{S} . In practice, we follow a two step approach where we first use frequent itemset mining [1] to extract itemsets \mathcal{S} and then apply our method to select a set of rules that form an interpretable decision set.

4.1 Objective

Our algorithm maximizes a joint objective that scores decision sets based on both how interpretable and accurate they are. To facilitate analysis, we always construct the terms of the objective as non-negative reward functions which we want to maximize.

Interpretability. First, we favor decision sets with a smaller number of rules:

$$f_1(\mathcal{R}) = |\mathcal{S}| - \text{size}(\mathcal{R})$$

where $|\mathcal{S}|$ is the total number of itemsets provided as input, which is an upper bound on the size of the decision set.

Second, we favor a decision set with fewer predicates in its rules:

$$f_2(\mathcal{R}) = L_{\max} \cdot |\mathcal{S}| - \sum_{r \in \mathcal{R}} \text{length}(r)$$

where L_{\max} is the maximum length of any rule that can be constructed from an itemset in \mathcal{S} , i.e.,

$$L_{\max} = \max_{r \in \mathcal{S} \times \mathcal{C}} \text{length}(r) .$$

Next, we introduce a pair of objective terms to favor decision sets with rules that do not overlap in the feature space:

$$f_3(\mathcal{R}) = N \cdot |\mathcal{S}|^2 - \sum_{\substack{r_i, r_j \in \mathcal{R} \\ i \leq j \\ c_i = c_j}} \text{overlap}(r_i, r_j)$$

and

$$f_4(\mathcal{R}) = N \cdot |\mathcal{S}|^2 - \sum_{\substack{r_i, r_j \in \mathcal{R} \\ i \leq j \\ c_i \neq c_j}} \text{overlap}(r_i, r_j) .$$

Recall that N is the number of points in the data set and therefore an upper bound on the overlap of two rules. Further, there could be a maximum of $|\mathcal{S}|^2$ pairs of rules in a given decision set. Overlapping rules blur the decision boundaries of the model, and with less overlap, it is easier for users to see which attribute values lead to particular class labels. We include two terms, one for rules with the same class and one for rules with different classes, so that we can weight them individually. We also include a term to encourage the decision set to have at least one rule that predicts each class:

$$f_5(\mathcal{R}) = \sum_{c' \in \mathcal{C}} \mathbb{1}(\exists r = (s, c) \in \mathcal{R} \text{ such that } c = c') .$$

(The indicator function $\mathbb{1}$ is 1 whenever the condition it takes as an argument is true and 0 otherwise.) Characterizing all of the classes in the data is important to users. Often, rare classes are important, such as a rare disease, and this term balances the need to cover them with the need for conciseness.

Accuracy. First, to encourage precision, we favor rules with small incorrect-cover sets:

$$f_6(\mathcal{R}) = N \cdot |\mathcal{S}| - \sum_{r \in \mathcal{R}} |\text{incorrect-cover}(r)| .$$

Second, to encourage recall, we favor decision sets that correctly cover data points with at least one rule:

$$f_7(\mathcal{R}) = \sum_{(\mathbf{x}, y) \in \mathcal{D}} \mathbb{1}(|\{r | (\mathbf{x}, y) \in \text{correct-cover}(r)\}| \geq 1) .$$

Full Objective. We can now state our full learning objective:

$$\arg \max_{\mathcal{R} \subseteq \mathcal{S} \times \mathcal{C}} \sum_{i=1}^7 \lambda_i f_i(\mathcal{R}) \quad (1)$$

where $\lambda_1, \dots, \lambda_7$ are non-negative weights that scale the relative influence of the terms. They may be application-dependent or fit with cross-validation, as we do in our experiments.

4.2 Optimizing the Objective

We now describe our approach to optimizing objective (1). Since the objective scores decision sets on a rich variety of properties, its optimization is non-trivial. However, it has a structure that can be exploited for optimizing it approximately with good theoretical guarantees.

Theorem 1. *The function $\sum_{i=1}^7 \lambda_i f_i(\mathcal{R})$ is non-negative, non-normal, non-monotone, and submodular on $\mathcal{P}(\mathcal{S} \times \mathcal{C})$, the power set of $\mathcal{S} \times \mathcal{C}$.*

Algorithm 1 Smooth Local Search (SLS) [18]

```

1: Input: Objective  $f$ , domain  $X = \mathcal{S} \times \mathcal{C}$ , parameters  $\delta$  and  $\delta'$ 
2:
3:  $A = \emptyset$ 
4:  $\text{OPT} = f(\Phi_X(X, 0))$ 
5: for each element  $x \in X$  do
6:   Estimate  $\mathbb{E}[f(\Phi_X(A, \delta) \cup x)] - \mathbb{E}[f(\Phi_X(A, \delta) \setminus x)]$  within an
   error of  $\frac{1}{|X|^2} \text{OPT}$ 
7:   Call this estimate  $\tilde{\omega}_{A, \delta}(x)$ 
8: end for
9: for each element  $x \in X \setminus A$  such that  $\tilde{\omega}_{A, \delta}(x) > \frac{2}{|X|^2} \text{OPT}$  do
10:    $A = A \cup x$ 
11:   Goto Line 5
12: end for
13: for each element  $x \in A$  such that  $\tilde{\omega}_{A, \delta}(x) < \frac{-2}{|X|^2} \text{OPT}$  do
14:    $A = A \setminus x$ 
15:   Goto Line 5
16: end for
17: return  $\Phi_X(A, \delta')$ 

```

Proof (Sketch). Non-negative functions and submodular functions are each closed under addition and multiplication with non-negative scalars. Each term $f_i(\mathcal{R})$ is non-negative by construction. $f_1(\mathcal{R})$ and $f_2(\mathcal{R})$ are modular and therefore submodular. The remaining terms $f_i(\mathcal{R})$, $i = 3, \dots, 7$, are submodular. Since $\lambda_i \geq 0$, objective (1) is non-negative and submodular.

To show that the objective is non-normal and non-monotone, it suffices to show that f_i is non-normal and non-monotone for some i . Since $f_3(\emptyset) = N \cdot |\mathcal{S}|^2$, $f_3(\mathcal{R})$ is non-normal. Since $f_3(\mathcal{R}_1) \geq f_3(\mathcal{R}_2)$ for any $\mathcal{R}_1 \subseteq \mathcal{R}_2$, $f_3(\mathcal{R})$ is non-monotone. Therefore, objective (1) is non-normal and non-monotone. \square

The full proof is provided in the extended version of this paper [32]. Since our objective is to maximize a submodular function, an NP-hard problem [26], we will approximate the solution. The best known algorithm for maximizing a non-negative, non-normal, non-monotone, and submodular function is smooth local search (SLS) [18].

Smooth Local Search (SLS). The SLS algorithm finds a decision set \mathcal{R} by sampling elements in $X = \mathcal{S} \times \mathcal{C}$ with different probabilities based on some underlying set $A \subseteq X$. Elements (or rules) in the set A are chosen as a function of our objective. The decision set \mathcal{R} thus obtained is a *smoothed* local optimum. Before we describe SLS in detail, we first define the sampling procedure which forms the core subroutine of the SLS algorithm.

Definition 8. Let $\Phi_X(A, \delta)$ denote a subset of X sampled with bias δ on some set $A \subseteq X$ according to the following procedure: each element $x \in X$ is sampled independently with probability $p = (1 + \delta)/2$ if $x \in A$ and with probability $p = (1 - \delta)/2$ if $x \notin A$.

SLS is presented in Algorithm 1. We first initialize $A = \emptyset$ (line 3) and obtain an estimate of the optimal value of the objective function f by computing $\text{OPT} = f(\Phi_X(X, 0))$ (line 4). Note that $\Phi_X(X, 0)$ denotes a set of elements randomly sampled from X , i.e., each element in X is chosen with probability $1/2$. Then, we estimate the effect of each element $x \in X$ by computing the difference in the objective value when x is added to $\Phi_X(A, \delta)$ and when x is subtracted from it (lines 5–8). We compute these estimates by repeatedly sampling from $\Phi_X(A, \delta)$ until the standard error for the difference in scores is less than $1/|X|^2 \cdot \text{OPT}$ [22]. After computing these estimates, we look for an $x \in X$ to add to A by checking if the estimate for some $x \in X$ is greater than $2/|X|^2$.

OPT (line 9). If such an element is found, we add it to A , and start over (lines 10–11). If no such element is found, we then look for an $x \in X$ to remove from A by checking if the estimate for x is less than $-2/|X|^2 \cdot \text{OPT}$ (line 13). If such an element is found, we remove it from A and start over (lines 14–15). When elements can no longer be added or removed from A , SLS returns $\Phi_X(A, \delta')$, a random subset of X sampled with bias δ' on A (line 17).

SLS will terminate in polynomial time and produce a solution of guaranteed quality. If we run SLS for two choices of parameters, $(\delta = 1/3, \delta' = 1/3)$ and $(\delta = 1/3, \delta' = -1)$, the better of the two solutions has expected value at least $(2/5 - o(1))f^*$, where f^* is the optimal value of f over X [18]. To construct a decision set that maximizes Equation 1, we run SLS twice with the two aforementioned choices of parameters and return the better of the two solutions according to objective (1).

5. EXPERIMENTAL EVALUATION

In this section, we discuss the detailed experimental evaluation of interpretable decision sets. First, we analyze their classification performance. Second, we propose metrics for quantifying interpretability and use them to evaluate interpretable decision sets. For each of these experiments, we compare interpretable decision sets with various state-of-the-art baselines. Third, we also present a detailed ablation study where we explore the effects of various components of our objective function on the predictive accuracy and interpretability of the resulting decision sets. Lastly, we conclude this section by presenting the results of a user study which we carried out to understand how easy it is for people to interpret our models.

Dataset Description. We analyzed three real-world datasets from diverse domains such as the judicial system, education, and medicine. These domains rely heavily on human decision making, and hence would benefit a great deal from the design of predictive models which are interpretable.

Our first dataset is a sample of **bail outcomes** collected from several state courts in the US between 1990–2009 [25]. This dataset consists of past criminal records, demographic attributes, and other details of about 86K defendants who were released on bail (Table 2). Whenever a defendant is released on bail, there are four possible outcomes: (1) the defendant is not arrested while out on bail and appears for further court dates (No Risk), (2) the defendant fails to appear for further court dates (FTA), (3) the defendant commits a non-violent crime (NCA) and, (3) the defendant commits a violent crime (NVCA) when released on bail. Our goal is to predict these outcomes.

Our second dataset consists of **student performance** records for about 21K students who were set to graduate in 2012 and 2013 from schools in a medium-sized district on the east coast [31]. Grades, demographic attributes, absence rates, suspensions, and withdrawals were tracked from grade 6 to 8 for each of these students. The target prediction is whether a student graduated high school on time or not or dropped out mid way (Table 2).

Lastly, our third dataset comprises **medical diagnosis** records of about 150K patients collected by a web-based electronic health record company [41]. This dataset documents information about patients who are suffering from asthma, diabetes, depression, lung cancer and rare blood cancers such as leukemia and myelofibrosis. Several attributes such as current medical conditions and symptoms, medical history, and age are recorded for each of the 150K patients (Table 2).

Baselines for Comparison. We compared interpretable decision sets with the following state of the art interpretable rule-based sys-

tems for classification: Bayesian Decision Lists (BDL) [34], CN2 [15], and Classification Based on Associations (CBA) [35]. CBA was designed to bridge the gap between association rule mining and classification and thus focuses mostly on optimizing for predictive accuracy. CN2 is a classic decision set learning algorithm which does not explicitly account for conciseness or overlap of the rules. On the other hand, BDL, a more recently proposed framework optimizes for conciseness alongside predictive accuracy when generating the decision lists. In addition to these baselines, we also compare our approach with several other standard classification models such as logistic regression, random forests, gradient boosting, and decision trees.

Parameter Selection. The formulation that we proposed for learning decision sets in Section 4 has parameters $\lambda_1 \dots \lambda_7$. We set aside 5% of our data as a validation set to estimate these parameters. We searched the parameter space using coordinate ascent to find parameters that produced a decision set with the highest AUC on the validation set and also satisfied some simple bounds on the interpretability metrics: *Fraction Overlap* ≤ 0.10 , *Fraction Uncovered* ≤ 0.15 , *Avg. Rule Length* ≤ 10 , *Number of Rules* ≤ 15 and *Fraction Classes* = 1.0. Each of the aforementioned metrics are discussed in detail in Section 5.2. See the extended version of this paper [32] for additional details.

The baseline model BDL has three hyperparameters: α , λ , and η . The hyperparameter α controls the Dirichlet prior on the distribution of labels and is chosen so that the prior is uniform. λ controls the prior on the number of the rules in the decision list and η controls a prior on the average number of predicates per rule in the decision list. In order to set λ and η , we experimented with all the values in the range $\{2 \dots 25\}$ and picked those which resulted in a decision list that was both interpretable as well as accurate according to the metrics defined in Sections 5.1 and 5.2. Our approach IDS as well as the baselines BDL and CBA utilize the Apriori algorithm [2] for generation of candidate itemsets. The Apriori algorithm has a support threshold parameter ϵ which ensures that the candidate itemsets are present in at least ϵ data points. We set the value of this parameter to 1% of the total dataset size. In the case of CN2, we set the maximum rule length to 10. In the case of tree-based models, we stop splitting a node when the number of data points per leaf falls below 1% of the total dataset size.

5.1 Classification Performance

We evaluated the classification performance of our model on three different prediction tasks: (1) predicting the behavior of a defendant who was released on bail; (2) predicting if a student will not graduate high school on time or drop out; (3) predicting which of the six different diseases a patient is likely to be diagnosed with. We evaluated the performance of our approach as well as the baselines using standard metrics such as AUC ROC and F1-score via 10-fold cross validation.

In order to compute the AUC metric, we need to estimate the probability that a given data point d belongs to a particular class c . In the context of rule-based frameworks such as ours, such probabilities can be easily computed as follows: Let r be the rule which determined the class label of data point d . The probability that data point d belongs to some class c is the precision of rule r . That is, the fraction of data points classified by r to class c that truly belong to c . Since our prediction tasks are multi-class, we used *micro-averaged* AUC [21]. We binarized the classes in the data using a one vs. all strategy and computed the AUC for each class. Micro-averaged AUC is the average of these per-class AUCs.

Table 3 records the AUC scores for our framework and other baselines. It can be seen that the ensemble models comprising 50

Dataset	# of Data Points	Features	Classes
Bail Outcomes	86,152	gender, age, current offense details, past criminal behavior of defendants	No Risk, FTA, NCA, NVCA
Student Performance	21,713	gender, age, grades, absence rates & tardiness behavior recorded through grades 6 to 8, suspension/withdrawal/transfer history of students	Graduated on Time, Delayed Graduation, Dropped Out
Medical Diagnosis	150,743	current ailments, age, BMI, gender, smoking habits, medical history, family medical history of patients	Asthma, Diabetes, Depression, Lung Cancer, Leukemia, Myelofibrosis

Table 2: Summary of datasets.

Method	AUC ROC		
	Bail	Student	Medical
IDS	69.78	75.12	61.19
BDL	67.18	72.54	59.18
CBA	70.68	76.02	63.03
CN2	71.02	76.36	64.78
DT	70.08	75.31	63.28
GB (50 trees)	71.23	77.18	64.21
RF (50 trees)	70.87	77.12	63.92
LR	66.39	73.76	60.02

Table 3: Area Under the ROC curve for all the datasets. IDS: Interpretable Decision Sets (our method); BDL: Bayesian Decision Lists; CBA: Classification Based on Association; DT: Decision Tree; GB: Gradient Boosting; RF: Random Forests; LR: Logistic Regression.

trees exhibit higher AUC scores compared to other models. Further, our IDS framework, CBA, CN2, and decision trees exhibit comparable performance on bail outcomes and student performance data while our model slightly under-performs on the medical diagnosis data (based on paired t-tests at 95% CI). BDL and logistic regression turn out to be the worst performing baselines across all the datasets. It is interesting to note that in terms of classification accuracy our framework performs on par with models such as CBA and decision trees which optimize exclusively for predictive accuracy.

5.2 Quantifying Interpretability

We define metrics for evaluating the interpretability of rule-based models. We then use these metrics to compare and contrast the interpretability of our framework with that of other baselines. As discussed in Section 3.2, a set of rules is considered interpretable if (1) the rules in the set describe non-overlapping feature spaces, (2) most (ideally all) data points are covered by some rule in the set, (3) the set comprises a small number of rules and each of the rules is concise, and (4) the rules in the set describe most (ideally all) of the classes in the data.

Below, we present a list of interpretability metrics designed based on the aforementioned criteria. We first define these metrics for decision sets and then explain how they can be generalized to decision lists (i.e., if-then-else rules used by BDL and CBA methods).

Fraction Overlap. This metric captures the extent of overlap between every pair of rules of a decision set \mathcal{R} . Smaller values on this metric signify higher interpretability.

$$\text{Fraction Overlap}(\mathcal{R}) = \frac{2}{|\mathcal{R}| \cdot (|\mathcal{R}| - 1)} \sum_{\substack{r_i, r_j \in \mathcal{R} \\ i < j}} \frac{\text{overlap}(r_i, r_j)}{N}$$

This metric has a lower bound of 0.0 which corresponds to zero overlap between every pair of rules in \mathcal{R} and an upper bound of

1.0 which corresponds to a scenario where all the data points are covered by all the rules in \mathcal{R} . This metric takes a value zero for any decision list because the if-else if structure of a decision list ensures that a rule in the list applies only to those data points which have not been covered by any of the preceding rules.

Fraction Uncovered. This metric computes the fraction of data points which are not covered by any rule in a decision set \mathcal{R} :

$$\text{Fraction Uncovered}(\mathcal{R}) = 1 - \frac{1}{N} \left| \bigcup_{r \in \mathcal{R}} \text{cover}(r) \right|.$$

This metric assumes a minimum value of 0.0 when all the data points are covered by some rule in \mathcal{R} and it takes a maximum value of 1.0 when no data point is covered by any rule in \mathcal{R} which could imply that \mathcal{R} is empty. In the case of decision lists, this metric can be computed as the fraction of those data points which are covered by the final else clause of the list.

Avg. Rule Length. The metric captures the average number of predicates a human reader must parse to understand a rule in a decision set \mathcal{R} . It is computed as the mean length of the rules in \mathcal{R} :

$$\text{Avg. Rule Length}(\mathcal{R}) = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \text{length}(r).$$

When computing this metric for a decision list, it is important to account for the fact that a rule r_i in the list applies only to those data points which are not covered by any of the preceding rules $r_{i-1}, r_{i-2} \dots r_1$. This implies that the data points covered by rule r_i satisfy the pattern $s_i \wedge \neg s_{i-1} \wedge \neg s_{i-2} \wedge \dots \wedge \neg s_1$ where $s_i \dots s_1$ denote the itemsets corresponding to $r_i \dots r_1$ respectively. This also means that a human reader who is trying to parse the rule r_i must go through all the predicates of the preceding rules. Therefore, the length of a rule r_i in a decision list is computed as $\text{length}(r_i) + \text{length}(r_{i-1}) + \dots + \text{length}(r_1)$.

Number of Rules. This is the number of rules in a decision set \mathcal{R} denoted by $\text{size}(\mathcal{R})$ (Refer to the notation in Section 3.2.) It is straightforward to see how this metric applies to decision lists. When counting the number of rules in a decision list, we exclude the final else clause.

Fraction of Classes. This metric measures what fraction of the class labels in the data are predicted by at least one rule in a decision set \mathcal{R} :

$$\text{Frac. of Classes}(\mathcal{R}) = \frac{1}{|\mathcal{C}|} \sum_{c' \in \mathcal{C}} \mathbb{1}(\exists r = (s, c) \in \mathcal{R} \mid c = c').$$

This metric has a lower bound of 0.0 which corresponds to the scenario where no class is described by the decision set \mathcal{R} which could possibly mean that \mathcal{R} is empty. The maximum value that this metric can take is 1.0 which corresponds to the case where every class

	Frac. Overlap	Frac. Uncovered	Rule Length	Num. Rules	Frac. Classes
IDS	0.09	0.13	3.17	12	1.00
BDL	0.00	0.18	8.46	11	0.67
CBA	0.00	0.14	8.60	32	1.00
CN2	0.12	0.14	9.78	38	1.00

Table 4: Quantitative evaluation of model interpretability for different methods based on the medical diagnosis dataset. IDS: Interpretable Decision Sets (our method); BDL: Bayesian Decision Lists; CBA: Classification Based on Association.

is described by some rule in \mathcal{R} . This metric can be easily generalized to decision lists. When computing the number of classes described by a decision list, we do not consider the final else clause.

Results. We evaluated the interpretability of our model and the baselines using all the aforementioned metrics. Results for the medical diagnosis data are presented in Table 4. The overlap among the rules produced by our approach is much smaller than the overlap among the rules generated by CN2. On the other hand, there is zero overlap among the rules produced by BDL and CBA and, as discussed earlier in this section, this is due to their if-then-else structure. The fraction of data points left uncovered by IDS is smaller than that of the other baselines. The average rule length of the decision set produced by IDS is about three times smaller than that of the decision sets or lists output by the baselines. The number of rules in the decision set produced by our approach is comparable to that of BDL, where as CBA and CN2 generate models which have almost thrice the number of rules. In terms of accounting for all the classes, the rules generated by our framework, CN2, and CBA explain all the classes in the data. On the other hand, BDL produces a decision list which leaves out two of the less common but very important classes corresponding to rare types of blood cancers.

5.3 Ablation Studies

We also explore the impact of various components of our objective function on the interpretability and predictive accuracy of the resulting decisions sets. To this end, we experiment with various ablations of our model obtained by excluding some key components from the objective function one at a time.

Our first ablated model called *No Precision* is obtained by excluding the term which encourages precision, $f_6(\mathcal{R})$, from the learning objective. Similarly, another ablated model, which we refer to as *No Recall*, is learned by excluding the term which encourages recall, $f_7(\mathcal{R})$ from the objective. Other ablations, namely, *No Overlap*, *No Conciseness*, and *No Class* are obtained by removing the terms which favor each of these aspects, $f_3(\mathcal{R})$ and $f_4(\mathcal{R})$, $f_1(\mathcal{R})$ and $f_2(\mathcal{R})$, and $f_5(\mathcal{R})$, respectively.

We quantitatively evaluate the predictive power and interpretability of these ablated models. Results for medical diagnosis data are presented in Table 5. It can be seen that *No Precision* and *No Recall* exhibit poor predictive performance demonstrating the effect of excluding the precision and recall terms from the objective. It is interesting to note that decision set produced by the *No Precision* model leaves a large fraction of data points uncovered. This is because the precision term of our objective function ensures the coverage of data points and removing this term from the objective leaves the learning algorithm with no incentive to find a decision set which covers all the data points. Analogous observations can be made about other ablation models. Removing the overlap constraints from the objective (*No Overlap*) results in an increase in the fraction overlap metric and excluding the class coverage constraint

	AUC	Frac. Overlap	Frac. Uncov.	Rule Length	Num. Rules	Frac. Classes
Full IDS	61.19	0.09	0.13	3.17	12	1.00
No Prec.	51.26	0.09	0.19	3.19	12	1.00
No Recall	53.38	0.10	0.14	3.18	11	1.00
No Overlap	61.02	0.16	0.14	3.54	11	1.00
No Conc.	63.64	0.04	0.13	6.88	14	1.00
No Class	59.28	0.01	0.15	3.09	10	0.83

Table 5: Removing components of the objective function of our Decision Sets algorithm on the Medical Diagnosis dataset.

from the objective in the *No Class* model results in a decision set which does not describe a rare type of blood cancer called Myelofibrosis. Not accounting for conciseness (*No Conciseness*) produces rules which are about twice as long compared to those output by the full learning objective. These results demonstrate that each term in objective (1) contributes to the overall goal of learning accurate and interpretable decision sets.

5.4 Human Judgment of Interpretability

Our goal is to produce rule-based models which are interpretable as well as accurate. While most prior research [35, 34] used the term interpretable to loosely refer to any rule-based model since it can be parsed by humans, we proposed a much stronger notion of interpretability. We call a rule-based model interpretable if a human can understand the model’s decision boundaries by looking at the rules. There can be no better judges than humans to evaluate this notion of interpretability. Therefore, we performed a user study to determine how well humans are able to understand and describe the patterns of various classes in the data by looking at the rules produced by our framework.

User Study Design. We designed an online user study where each participant was presented with either a decision set produced by our framework or a decision list generated by the baseline BDL. We also evaluated the interpretability of BDL in the same setting so that we could compare and contrast the two approaches. In order to carry out this study, we recruited a set of 47 students who were taking a data mining course at Stanford. These students were familiar with concepts such as logical conjunctions, disjunctions and if-then/if-then-else structures. Each user was asked a series of descriptive and multiple-choice questions based on the rules shown to them. These questions were designed to test the user’s understanding of the decision boundaries of the classes in the data. Each user was presented with twelve questions, out of which ten were multiple choice questions and two were descriptive. Each user was allowed to participate in the study only once. Therefore, every student who participated in the study either answered questions based on a decision set or a decision list, but not both. We also recorded the time taken to answer every single question for each individual user. We used the decision sets and lists learned from the medical diagnosis data for this study. Figure 2 shows a screenshot of the interface used in the study.

Descriptive Questions. Each user was asked two descriptive questions. Each of the questions required the user to explain in plain language all the characteristics of a patient suffering from a particular disease based on the rules presented. For instance, if a user were presented with the model in Figure 1 (right), an example question would be: *Please write a short paragraph describing the characteristics of Depression patients based on the rules provided above.* These questions gauge how easy it is for a user to understand and describe the decision boundaries of the model.

Descriptive Question

In this question, you will see a set of rules which characterize various diseases. These rules have been generated by a machine learning model to explain the properties of patients suffering from the corresponding diseases. Please take a look at the rules and answer the question below.

Here, you will be asked to write a paragraph describing the properties of patients with a specific disease based on the given rules. Below, we provide an example which can help you understand how to write a short description given a rule.

Example:

Rule: If **Overweight** = False and **Smoking** = False, then **Healthy**

Description: People who do not smoke and do not have any weight problems are healthy.

Rules generated by a machine learning model "M1"

```
If Allergies = True and Smoking = True and Irregular-Heartbeat-Symptoms = True, then Asthma
If Allergies = True and Past-Respiratory-Illness = True and High-Body-Temperature = True, then Asthma
If Smoking = True and Overweight = True and Age >= 60, then Diabetes
If Family-History-Diabetes = True and Overweight = True and Has-Frequent-Infections = True, then Diabetes
If Frequently-Visited-Doctor = True and Childhood-Obesity = True and Past-Respiratory-Illness = True, then Diabetes
If Family-History-Depression = True and Past-Depression-Issues = True and Gender = Female, then Depression
If Overweight = True and Insurance-Coverage = False and High-Blood-Pressure = True, then Depression
If Past-Respiratory-Illness = True and Age >= 50 and Smoking = True, then Lung Cancer
If Family-History-LungCancer = True and Allergies = True and High-Blood-Pressure = True, then Lung Cancer
```

Question:

Please write a short paragraph describing the characteristics of **Depression** patients based on the rules provided above. Please use plain English language to write your description. Feel free to use multiple sentences to explain a single rule.

Your Answer:

Figure 2: A screenshot of the interface for the user study. In this example, the user is asked to write a paragraph describing a class based on the provided decision set.

Multiple Choice Questions. Each user was asked ten questions which could be answered with “true” or “false.” Each question provided a small subset of attribute values or symptoms associated with some hypothetical patient and asked the user if the information provided was sufficient to conclude that the patient suffered from a particular disease. For instance, if a user were presented with the model in Figure 1 (left), an example question would be: *Given a patient with the following attributes, **Respiratory-Illness** = Yes and **Smoker** = Yes, can you be absolutely sure that this patient suffers from **Lung Cancer**?* These questions measure if the rules provided to the users enable an easy understanding of the decision boundaries. Further, these questions closely resemble several real-world settings where decisions need to be made based on partial information and decision makers typically evaluate if the available information is sufficient to make an accurate decision.

Evaluation. The multiple choice questions that we asked users always had unambiguous answers. On the other hand, we had to set some rough guidelines to evaluate the correctness of the descriptive answers. An important guideline was that a right answer to a descriptive question should correctly explain all the rules characterizing a particular class while respecting the structure guided by the conjunctions, disjunctions and if-then/if-then-else statements.

Each descriptive answer was examined by two independent evaluators and tagged as right or wrong based on the guidelines highlighted above and their individual judgment. The interevaluator agreement was 89.36%. We excluded from our analysis those responses on which the evaluators did not agree.

We selected a list of metrics to evaluate the results: (1) **Human Accuracy**, which denotes the fraction of correct answers; (2) **Avg. Time Spent**, which denotes the average time spent in seconds to answer a question; (3) **# of Words**, which represents the average number of words used to write a descriptive answer. When computing metrics (2) and (3) for descriptive answers, we only considered those responses which were labeled as correct by both the evaluators.

Task	Metrics	IDS	BDL
Descriptive	Human Accuracy	0.81 (0.06)	0.17 (0.06)
	Avg. Time Spent (secs.)	113.4 (8.29)	396.86 (47.93)
	# of Words	31.11 (1.13)	120.57 (16.84)
Multiple Choice	Human Accuracy	0.97 (0.01)	0.82 (0.02)
	Avg. Time Spent (secs.)	28.18 (1.54)	36.34 (3.81)

Table 6: Results of a user study comparing interpretability of Interpretable Decision Sets (IDS) and Bayesian Decision Lists (BDL). Numbers in parentheses are standard errors.

Results. Table 6 presents the results of the user study. About 81% of the class descriptions written based on interpretable decision sets were accurate. In comparison, only 17% of the descriptions which were written based on BDLs were accurate.

The key to providing right answers to the descriptive questions was to correctly parse the conjunctions, disjunctions and the associated if-then/if-then-else statements. The most common mistakes users made when answering descriptive questions pertaining to our decision sets framework were: (1) explaining just one rule when there were multiple rules associated with a particular class, and (2) ignoring some predicate when explaining a rule. In the case of decision lists, the most common mistakes were: (1) not accounting for the negations induced by preceding rules when explaining a particular rule, (2) using other class labels when describing a particular class (for instance, we found answers such as “If a patient has not been diagnosed with lung cancer, then he/she is likely to have depression.”), and (3) explaining just one rule when there were multiple rules associated with a class. These additional mistake patterns introduced by the if-then-else structure of decision lists indicate that they are less interpretable than decision sets.

The average time taken to write a description based on a decision list is thrice as much as the time taken to write a description based on a decision set. In addition, the descriptions based on interpretable decision sets had 74% fewer words compared to the descriptions based on BDLs. This indicates that users find it much easier to describe classes based on the decision sets.

Our results even demonstrate that in the case of multiple choice questions, people were more accurate and much faster in answering questions based on our decision sets in comparison with those based on decision lists. This result confirms our hypothesis that users find it much simpler to reason about decision boundaries when presented with interpretable decision sets.

6. CONCLUSIONS

In this paper, we presented a method for learning interpretable decision sets. We formulated the problem of learning decision sets via an objective function that simultaneously optimizes for accuracy and interpretability. We proved that our objective is a non-monotone submodular function which can be efficiently optimized with a $2/5$ approximation guarantee. Our experiments demonstrated that decision sets are as accurate as other state-of-the-art methods but are much more interpretable. Our user study demonstrated that humans can reason much more accurately about the decision boundaries of a decision set than those of a decision list. Humans were also dramatically better at describing a decision set’s decision boundaries in words. There are multiple, interesting lines of future work to explore. It would be useful to extend decision sets to a regression setting. Another important direction is to incorporate human-in-the-loop feedback when learning interpretable decision sets so that human experts can help train and correct a model as it is learned.

Acknowledgements

This research has been supported in part by NSF CNS-1010921, IIS-1149837, NIH BD2K, ARO MURI, DARPA XDATA, DARPA SIMPLEX, Stanford Data Science Initiative, Boeing, Lightspeed, SAP, and Volkswagen. HL is funded by a Robert Bosch Stanford Graduate Fellowship. The authors would like to thank the students of the data mining course at Stanford for participating in the user study and Yilun Wang for helping with the evaluations. The authors are also grateful to Austin Benson, Marinka Zitnik, Bruno Abrahao, Rok Susic, and the anonymous reviewers for providing insightful feedback.

7. REFERENCES

- [1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *SIGMOD*, 1993.
- [2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Vldb*, 1994.
- [3] P. J. Azevedo. Rules for contrast sets. *Intelligent Data Analysis*, 14(6):623–640, 2010.
- [4] S. D. Bay and M. J. Pazzani. Detecting change in categorical data: Mining contrast sets. In *KDD*, 1999.
- [5] D. Bertsimas, A. Chang, and C. Rudin. Ordered rules for classification: A discrete optimization approach to associative classification. Operations Research Center Working Paper OR 386-11, MIT, 2011.
- [6] J. Bien and R. Tibshirani. Classification by set cover: The prototype vector machine. arXiv:0908.2284 [stat.ML], 2009.
- [7] A. Blum. On-line algorithms in machine learning. In A. Fiat and G. J. Woeginger, editors, *Online Algorithms: The State of the Art*, volume 1442 of *Lecture Notes in Computer Science*, chapter 14, pages 306–325. 1998.
- [8] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and Regression Trees*. Chapman and Hall/CRC, 1984.
- [9] L. C. Briand, V. R. Brasili, and C. J. Hetmanski. Developing interpretable models with optimized set reduction for identifying high-risk software components. *IEEE Transactions on Software Engineering*, 19(11):1028–1044, 1993.
- [10] B. Bringmann and A. Zimmermann. The chosen few: On identifying valuable patterns. In *ICDM*, 2007.
- [11] B. Bringmann and A. Zimmermann. One in a million: picking the right patterns. *Knowledge and Information Systems*, 18(1):61–81, 2009.
- [12] J. B. Carroll. An analytical solution for approximating simple structure in factor analysis. *Psychometrika*, 18(1):23–38, 1953.
- [13] H. Cheng, X. Yan, J. Han, and C.-W. Hsu. Discriminative frequent pattern analysis for effective classification. In *ICDE*, 2007.
- [14] P. Clark and R. Boswell. Rule induction with CN2: Some recent improvements. In *European Working Session on Machine Learning*, 1991.
- [15] P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3(4):261–283, 1989.
- [16] N. Cruz, J. Baratgin, M. Oaksford, and D. E. Over. Bayesian reasoning with ifs and ands and ors. *Frontiers in psychology*, 6, 2015.
- [17] G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In *KDD*, 1999.
- [18] U. Feige, V. S. Mirrokni, and J. Vondrák. Maximizing non-monotone submodular functions. *SIAM J. on Computing*, 40(4):1133–1153, 2011.
- [19] M. García-Borroto, J. F. Martínez-Trinidad, and J. A. Carrasco-Ochoa. A new emerging pattern mining algorithm and its application in supervised classification. In *KDD*, 2010.
- [20] S. Guillaume. Designing fuzzy inference systems from data: An interpretability-oriented review. *IEEE Transactions on Fuzzy Systems*, 9(3):426–443, 2001.
- [21] D. J. Hand and R. J. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning*, 45(2):171–186, 2001.
- [22] J. Hartline, V. Mirrokni, and M. Sundararajan. Optimal marketing strategies over social networks. In *WWW*, pages 189–198, 2008.
- [23] T. J. Hastie and R. J. Tibshirani. *Generalized Additive Models*. Chapman and Hall/CRC, 1990.
- [24] F. Herrera, C. J. Carmona, P. González, and M. J. Del Jesus. An overview on subgroup discovery: foundations and applications. *Knowledge and information systems*, 29(3):495–525, 2011.
- [25] K. L. Jordan and T. L. Freiburger. The effect of race/ethnicity on sentencing: Examining sentence type, jail length, and prison length. *J. of Ethnicity in Criminal Justice*, 13(3):179–196, 2015.
- [26] S. Khuller, A. Moss, and J. S. Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999.
- [27] B. Kim, C. Rudin, and J. Shah. The Bayesian case model: A generative approach for case-based reasoning and prototype classification. In *NIPS*, 2014.
- [28] B. Kim, J. Shah, and F. Doshi-Velez. Mind the gap: A generative approach to interpretable feature selection and extraction. In *NIPS*, 2015.
- [29] A. R. Klivans and R. A. Servedio. Toward attribute efficient learning of decision lists and parities. *J. of Machine Learning Research*, 7:587–602, 2006.
- [30] P. Kralj, N. Lavrač, D. Gamberger, and A. Krstajić. Contrast set mining for distinguishing between similar diseases. *Artificial Intelligence in Medicine*, pages 109–118, 2007.
- [31] H. Lakkaraju, E. Aguiar, C. Shan, D. Miller, N. Bhanpuri, R. Ghani, and K. L. Addison. A machine learning framework to identify students at risk of adverse academic outcomes. In *KDD*, 2015.
- [32] H. Lakkaraju, S. H. Bach, and J. Leskovec. Interpretable decision sets: A joint framework for description and prediction. Technical report, Stanford InfoLab, 2016.
- [33] N. Lavrač, B. Kavšek, P. Flach, and L. Todorovski. Subgroup discovery with CN2-SD. *J. of Machine Learning Research*, 5:153–188, 2004.
- [34] B. Letham, C. Rudin, T. H. McCormick, and D. Madigan. Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. *Annals of Applied Statistics*, 9(3):1350–1371, 2015.
- [35] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *KDD*, 1998.
- [36] Y. Lou, R. Caruana, and J. Gehrke. Intelligible models for classification and regression. In *KDD*, 2012.
- [37] Y. Lou, R. Caruana, J. Gehrke, and G. Hooker. Accurate intelligible models with pairwise interactions. In *KDD*, 2013.
- [38] D. Malioutov and K. Varshney. Exact rule learning via boolean compressed sensing. In *ICML*, 2013.
- [39] D. Nauck and R. Kruse. Obtaining interpretable fuzzy classification rules from medical data. *A.I. in Medicine*, 16(2):149–169, 1999.
- [40] P. K. Novak, N. Lavrač, and G. I. Webb. Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. *J. of Machine Learning Research*, 10:377–403, 2009.
- [41] H. Y. Ong, D. Wang, and X. S. Mu. Diabetes prediction with incomplete patient data. Technical report, 2014.
- [42] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
- [43] W. Revelle and T. Rocklin. Very simple structure: An alternative procedure for estimating the optimal number of interpretable factors. *Multivariate Behavioral Research*, 14(4):403–414, 1979.
- [44] G. Ridgeway, D. Madigan, T. Richardson, and J. O’Kane. Interpretable boosted naive Bayes classification. In *KDD*, 1998.
- [45] R. L. Rivest. Learning decision lists. *Machine Learning*, 2(3):229–246, 1987.
- [46] D. Rodríguez, R. Ruiz, J. C. Riquelme, and J. S. Aguilar-Ruiz. Searching for rules to detect defective modules: a subgroup discovery approach. *Information Sciences*, 191:14–30, 2012.
- [47] H. Schielzeth. Simple means to improve the interpretability of regression coefficients. *Methods in Ecology and Evolution*, 1(2):103–113, 2010.
- [48] G. Su, D. Wei, K. R. Varshney, and D. M. Malioutov. Interpretable two-level Boolean rule learning for classification. arXiv:1511.07361, 2015.
- [49] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. of the Royal Statistical Society. Series B*, 58(1):267–288, 1996.
- [50] B. Ustun and C. Rudin. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 102(3):1–43, 2015.
- [51] L. G. Valiant. Projection learning. *Machine Learning*, 37(2):115–130, 1999.
- [52] J. Wang and G. Karypis. Harmony: Efficiently mining the best rules for classification. In *SDM*, 2005.
- [53] T. Wang, C. Rudin, F. Doshi-Velez, Y. Liu, E. Klampfl, and P. MacNeille. Or’s of and’s for interpretable classification, with application to context-aware recommender systems. arXiv:1504.07614, 2015.
- [54] S. Wadyan. Review and comparison of associative classification data mining approaches. *International J. of Computer, Electrical, Automation, Control and Information Engineering*, 8(1):34–45, 2014.
- [55] S. Wood. *Generalized Additive Models: An Introduction with R*. Chapman and Hall/CRC, 2006.
- [56] X. Yin and J. Han. Cpar: Classification based on predictive association rules. In *SDM*, 2003.