Partially Interpretable Models with Guarantees on Coverage and Accuracy

Nave Frost NAFROST@EBAY.COM
eBay

Zachary C. Lipton ZLIPTON@CMU.EDU
Carnegie Mellon University and Abridge

Yishay Mansour MANSOUR.YISHAY@GMAIL.COM
Tel-Aviv University and Google

Michal Moshkovitz MICHAL.MOSHKOVITZ@IL.BOSCH.COM
Bosch Center for AI

Editors: Claire Vernade and Daniel Hsu

Abstract

Simple, sufficient explanations furnished by short decision lists can be useful for guiding stakeholder actions. Unfortunately, this transparency can come at the expense of the higher accuracy enjoyed by black box methods, like deep nets. To date, practitioners typically either (i) insist on the simpler model, forsaking accuracy; or (ii) insist on maximizing accuracy, settling for post-hoc explanations of dubious faithfulness. In this paper, we propose a hybrid *partially interpretable model* that represents a compromise between the two extremes. In our setup, each input is first processed by a decision list that can either execute a decision or abstain, handing off authority to the opaque model. The key to optimizing the decision list is to optimally trade off the accuracy of the composite system against coverage (the fraction of the population that receives explanations). We contribute a new principled algorithm for constructing partially interpretable decision lists, providing theoretical guarantees addressing both interpretability and accuracy. As an instance of our result, we prove that when the optimal decision list has length k, coverage c, and b mistakes, our algorithm will generate a decision list that has length no greater than $4k$, coverage at least $c/2$, and makes at most $4b$ mistakes. Finally, we empirically validate the effectiveness of the new model.

Keywords: Interpretability, Decision List, Partially Interpretable Models

1. Introduction

In many settings throughout society, it is desirable that decision-makers provide a set of *explanations* to justify any unfavorable decision. While the effort to pin down what broadly constitutes an explanation has devolved into a fraught philosophical exercise, we focus here on two desiderata. In our cases of interest, we would like to produce explanations that are *sufficient* and *simple*. Consider an explanation of the form: “you received an unfavorable decision U because of E”. For explanation E to be sufficient, it must be the case that whenever predicate E is satisfied, the unfavorable decision U is received. While the explanation doesn’t recommend precise actions to realize the positive decision in the future, they are still action-guiding in that they point to a condition that *must* change. Moreover, these explanations are only useful if the candidate conditions are sufficiently simple to parse and cover a substantial fraction of the population. The explanation “you were denied a loan because your current credit utilization is exactly 34.12%” is not very useful. But if the condition were "because your credit utilization is above 20%", you could use this information to plan.

Unfortunately, for many real-world problems, such simplicity-restricted rule lists cannot compete on utility against more expressive model classes. Thus, a tension emerges between the accuracy of a method and its amenability to such simple sufficient explanations. Many practitioners assert that we should always use the simpler methods in certain scenarios (Rudin, 2019; Letham et al., 2015; Lan et al., 2023). Others suggest *post hoc* heuristics for *"explaining"* black-box models, either pointing to "salient" features or approximating the complex model by a simple model for purposes of furnishing an explanation (Ribeiro et al., 2016; Lundberg and Lee, 2017; Ribeiro et al., 2018). In both cases, it's not clear what these ostensible explanations really mean.

In this paper, we propose to navigate this tension as a tradeoff between coverage and accuracy. We accomplish this by constructing *partially interpretable models*, where a simple model, amenable to generating the desired explanations, runs in the foreground, defaulting to a black-box model whenever it abstains. The key insight is that even when the expressivity of the black-box model is required to achieve high overall accuracy on the entire population, there may exist large subpopulations for which the simple model is fit to task. Whenever the simple model, instantiated here as a decision list, makes a decision, it generates a valid explanation. Whenever the simple model abstains, no explanation is given. Effectively, our approach partitions the population into an "easier" subset, for which we can have our cake and eat it, and a more difficult subset, for which accuracy comes at the cost of explainability. While such hybrid approaches may not be appropriate in all cases (e.g. when regulatory authorities mandate explanations for all), we believe both that there are cases when they are appropriate and characterizing this tradeoff is of general conceptual interest.

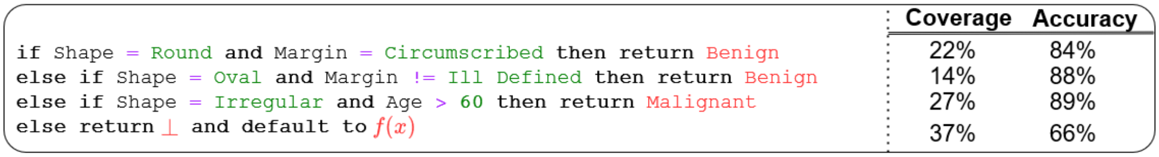
Specifically, in our proposed partially interpretable models, the "simple model" consists of a decision list and we are agnostic to the form of the black-box model f. We introduce a principled algorithm for learning the decision list given a fixed f and derive associated theoretical guarantees on coverage and accuracy. Decision lists operate by moving from one condition to another until a condition is satisfied, at which point an output is produced. In our setup, we add a fallback option, defaulting to f's prediction whenever none of the conditions are met. Note that the explanation E here consists of the set of all conditions tested, starting from the head of the list and going until the condition that was met.

As a special case, our algorithm can be applied in cases where we only require explanations of one side of the classification (say, the unfavorable decision). In such cases, the simplicity of explanations increases dramatically, because we can consider all conditions that are met (e.g., "your income was too low", "your credit utilization is too high") to each constitute a sufficient explanation. In other words, we need only return the conditions that were met, not the conjunction of all conditions that were tested. This special case is conceptually important given our motivating examples, where only unfavorable decisions require recourse.

Figure 1 illustrates an example of a partially interpretable model based on a decision list. By allowing the model to return f's choice, we combine f, the opaque model, with the decision list to create a partially interpretable model. This approach enhances the accuracy of decision lists without sacrificing interpretability altogether.

Partially interpretable decision list models, which modifies the opaque model f to enhance transparency, offers several advantages:

1. **Accuracy interpretability trade off.** Partially interpretable decision list models have the ability to showcase both ends of the spectrum in terms of interpretability and accuracy. On



```
if Shape = Round and Margin = Circumscribed then return Benign
else if Shape = Oval and Margin != Ill Defined then return Benign
else if Shape = Irregular and Age > 60 then return Malignant
else return ⊥ and default to f(x)
```

Coverage	Accuracy
22%	84%
14%	88%
27%	89%
37%	66%

Figure 1: An interpretable decision list model for mammogram classification. The model uses sets of simple rules that can classify samples as benign or malignant. If no rule matches a sample, i.e., $\perp$ is returned, a non-interpretable neural network f gives a prediction. The *test coverage and accuracy* of each rule are shown on the right. The three interpretable rules cover 63% of the test samples, and the whole model achieves 79% accuracy overall. The difference in test accuracy between the stand-alone neural network and the one that includes the partial interpretable model is less than 1%.

one hand, the decision list that consistently produces $\perp$ as its result, represents the opaque model which lacks interpretability. On the other hand, the decision list that never returns $\perp$, is a fully interpretable model, although it may come with a potential decrease in accuracy. Our approach empowers us to find a balance between these two extremes by selecting a partial interpretable model that best suits the given situation. This flexibility allows us to tailor the level of interpretability and accuracy according to our specific needs.

2. **Local advantage: the sufficiency property.** Partial interpretable decision list provides a local explanation for all examples it covers. This explanation is fully sufficient (Dasgupta et al., 2022), i.e., whenever the explanation holds, all examples that satisfy the explanation have the same prediction. This is in strike comparison to the popular post-hoc explanation method Anchors (Ribeiro et al., 2018). Anchors is a technique for providing a local explanation, which generates a rule as an explanation for a particular instance. The rule generated by Anchors is not entirely precise, so if the rule is applied, the label cannot be determined with complete certainty. Therefore, two users who are subject to the same rule may receive different predictions, making the rule an unreliable explanation. For example, suppose a user is refused a loan due to a salary of 15K. In that case, the explanation would not be satisfactory if another user received a loan with a salary of 10K. In summary, unlike post-hoc explanations such as Anchors that rely solely on high probability, partially interpretable models take a different approach by modifying the classifier f itself. This modification aims to provide fully sufficient explanations, effectively addressing the problem of unreliable rules.

3. **Global advantage: model (partial) transparency.** Black-box models, which provide limited transparency, have become the norm in modern machine learning, posing a problem in consequential scenarios. Fortunately, partial interpretable models enhance transparency. The procedure we suggest in this paper modifies a black-box model's predictions to ensure that a significant portion of the decision-making process becomes more transparent, granting us valuable insights into the internal mechanisms of the decision-making process.

1.1. Our results

This paper introduces and analyzes a new computationally efficient algorithm for finding partial interpretable decision lists. There are three main parameters used to quantify the quality of these models: length, coverage, and mistakes. Length ensures that the decision list is simple, coverage measures the probability that the decision list is used, and mistakes indicate the probability of misclassifications when the decision list is used. We prove that the new algorithm presented in this paper have good performance with respect to all three parameters compared to the optimal partially interpretable model. Namely, we prove the following.

Theorem 1 (main theorem, informal) *Suppose there is a partial interpretable decision list with length k, b mistakes and coverage c. Then, the output of Algorithm 1 is a decision list with at most $O(k)$ rules, coverage at least $\Omega(c)$, and at most $O(b)$ mistakes.*

In addition to the theoretical analysis, this paper also includes an empirical evaluation of the proposed algorithm. We test the algorithm on several datasets and compare its performance to other black-box models. Our experimental results demonstrate that the proposed algorithm is effective in producing partially interpretable models that achieve high accuracy while maintaining interpretability. Overall, the empirical evaluation provides further evidence of the practical usefulness of the proposed algorithm in real-world applications.

1.2. Related work

In this paper, we develop partially interpretable models with provable guarantees. Namely, we build an algorithm that returns a partially interpretable model having approximation bounds compared to the optimal model. Another advantage of our model is that its explanations are faithful, i.e., the model satisfies the sufficiency property (Dasgupta et al., 2022). The sufficiency property ensures that all instances meeting the conditions specified in an explanation share the same label. While prior research has focused on variants of partially interpretable models, none have been able to achieve an approximation algorithm that returns a model satisfying the sufficiency property. For further details, please refer to Table 1 and the next two paragraphs.

Table 1: Comparison with prior research related to partially interpretable models

Criterion Work	Approximation guarantees	Sufficiency
Ours	✓	✓
Lakkaraju et al. (2016)	✓	✗
Wang (2019)	✗	✗
Pan et al. (2020)	✗	✗
Rafique et al. (2020)	✗	✗
Wang and Lin (2021)	✗	✓
Ismail et al. (2022)	✗	✗
Ferry et al. (2023)	✗	✗

The only related work having approximation guarantees compared to the optimal model is Lakkaraju et al. (2016). This work constructs a decision set, which differs from a decision list as it includes a set of rules without a specific order. Additionally, they have a single objective that combines all desired model's parameters (e.g., coverage, accuracy, and complexity). Optimizing this objective does not ensure simultaneous guarantees for all parameters. In contrast, our approach guarantees the performance of each term individually. In simpler terms, our algorithm yields a solution with a small number of rules, a small number of errors, and a large coverage. Other theoretical works, such as Ferry et al. (2023), employ a PAC argument to bound accuracy but do not provide bounds on the coverage. Additionally, Wang and Lin (2021) establishes bounds on the properties of each rule individually, but it does not offer a bound compared to the optimal solution.

Next, we explore previous research in view of the sufficiency property. Lakkaraju et al. (2016) uses a decision set approach, wherein, if multiple rules cover an example, the decision defaults to a tie-breaking function. Consequently, it lacks the sufficiency property. For instance, consider two samples, x_1 covered solely by the rule r_1 with a resulting label of 0, and x_2 covered by both r_1 and r_2. Although the explanation provided by r_1 is applicable to x_2, it is directed to the tie-breaking function and may receive a label of 1. The remainder of Table 1 typically employs a gating mechanism to determine whether to use the black-box model or the interpretable model. However, a generic gating mechanism poses a challenge to the sufficiency property. For instance, the gating mechanism might steer x_1 towards the interpretable model, which predicts label 0 and provides explanation r. Yet, r might also cover x_2, but the gating mechanism may steer x_2 towards the black-box model, resulting in label 1. Consequently, the explanation r fails to be sufficient for label determination. Nonetheless, in Wang and Lin (2021) and in our own work, the gating mechanism is engineered to ensure the sufficiency property is maintained.

Other works diverge from the pursuit of constructing interpretable models or even those with partial interpretability. Instead, their focus lies in adding post-hoc explanations to an existing model prediction. There are various types of post-hoc explanations, including finding feature attributions (Lundberg and Lee, 2017; Chen et al., 2018; Senetaire et al., 2023), approximating the model (Ribeiro et al., 2016, 2018), returning counterfactual examples (Wachter et al., 2017; Mothilal et al., 2020; Deutch and Frost, 2019), or providing concept explanations (Kim et al., 2018; Espinosa Zarlenga et al., 2022). However, the use of post-hoc explanations poses many difficulties (Rudin, 2019), and their faithfulness is questionable (Adebayo et al., 2018; Dasgupta et al., 2022).

In a different line of research, various models exist for learning with abstention (Hendrickx et al., 2021), such as the mistake-bound model (Li et al., 2008; Rivest and Sloan, 1988), active learning (El-Yaniv and Wiener, 2010; Wiener and El-Yaniv, 2011), online settings (Cortes et al., 2018), and transductive learning (Goldwasser et al., 2020). The works of Sayedi et al. (2010); Zhang and Chaudhuri (2016) examined the tradeoff between making mistakes and abstaining. One of the main differences between our work, and the previous works, is that we limit the way that we can select the abstaining. Specifically, we limit the abstaining to be obtained by a partially interpretable decision list. This important restriction is a major challenge of our work.

2. Problem formulation

We denote the example domain by $\mathcal{X}$ and the label domain by $\mathcal{Y}$ and we fix a classifier $f : \mathcal{X} \to \mathcal{Y}$. In this paper,we focus on partially interpretable *decision lists* with a family of rules $\mathcal{R}$, where each rule $r \in \mathcal{R}$ is composed from a condition $\mathrm{cond} : \mathcal{X} \to \{\mathrm{True}, \mathrm{False}\}$ and a class label $o \in \mathcal{Y}$.

Definition 2 (partially interpretable decision list) *Given a classifier $f : \mathcal{X} \to \mathcal{Y}$, a partially interpretable decision list with family of rules $\mathcal{R}$ is a decision list, where the last condition leads to "I don't know" which defaults to the classifier f. More formally, the new model is defined by a series of rules $(r_1, o_1) \dots, (r_k, o_k) \in \mathcal{R} \times \mathcal{Y}$. Given an example x, it goes through the following process:*

- *if* $r_1(x)$ *is True return* o_1
- *else if* $r_2(x)$ *is True return* o_2
- *else if* ...
- *else return* $\perp$ *and default to* $f(x)$

For example, in Figure 1, r_1 is "`Shape = Round and Margin = Circumscribed`", o_1 is `Benign`, r_2 is "`Shape = Oval and Margin` $\neq$ `Ill Defined`", and o_2 is `Benign`.

There are four measures to quantify the quality of a partially interpretable decision list $DL : \mathcal{X} \to \mathcal{Y} \cup \{\perp\}$, which we describe next. The first three measures quantify the interpretability of the partially interpretable decision list, while the last one measures its performance.

1. **Length of decision list**—number of rules in the decision list, i.e., k, preferably be as small as possible.

2. **Coverage**—empirical coverage (or coverage for short) is the fraction of examples in the training data $x_1, \ldots, x_n$ that the decision list does not return $\perp$.

$$coverage(DL) := \frac{\sum_{i=1}^{n} \mathbb{I}(DL(x_i) \neq \perp)}{n} \in [0, 1].$$

   If the coverage is 0, then the model is simply the opaque model f. If the coverage is 1, then every prediction is handled by the decision list. Intermediate values allow us to trade off the accuracy of f and the interpretability of the decision list by compromising on the level of coverage.

3. **Mistakes**—empirical fraction of mistakes is the number of examples in the training data $(x_1, y_1) \ldots, (x_n, y_n)$ that the decision list returns a wrong answer, divided by the unnormalized coverage

$$mistake(DL) = \frac{\sum_{i=1}^{n} \mathbb{I}(DL(x_i) \neq \perp \wedge DL(x_i) \neq y_i)}{\sum_{i=1}^{n} \mathbb{I}(DL(x_i) \neq \perp)} \in [0, 1].$$

4. **Complexity of the family of rules** $\mathcal{R}$ — various methods exist for measuring its complexity, such as using $|\mathcal{R}|$ for finite classes or employing the VC dimension of $\mathcal{R}$ for infinite classes when $\mathcal{Y} = \{0, 1\}$. We mainly use $|\mathcal{R}|$ as the complexity measure in this paper.

3. Algorithm

This section presents Algorithm 1 that learns a partial interpretable decision list with provable guarantees (the proof is in the next section). The algorithm operates by selecting a new rule at each step, and once the intended coverage is achieved, it defaults to the opaque model, resulting in a partly decision list. At each step, the algorithm selects a rule that has the minimal ratio of mistakes to coverage, among all rules that are sufficiently large. The pseudocode for the algorithm can be found in Algorithm 1.

The algorithm consists of two components: initialization and the main part, which is the rule selection process. During the initialization phase, which occurs between Lines 5 and 11, various

Algorithm 1 Learning partial explainer

```
 1: **Input:** $X = \{(x_i, y_i)\}_{i=1}^{n}$: training data
 2:         $\ell$ : fraction covered to be considered large set ($\ell \in (0, 1)$)
 3:         $s$ : fraction of examples to collect before algorithm stops ($s \in (0, 1)$)
 4:         $\mathcal{R}$ : set of rules
 5: # Initialization
 6: $\forall r \in \mathcal{R}.\ \ C_r \leftarrow \{x_i \in X : r.\text{cond}(x_i) = \text{True}\}$
 7: $\forall r \in \mathcal{R}.\ \ B_r \leftarrow \{x_i \in X : r.\text{cond}(x_i) = \text{True} \wedge r.o_i \neq y_i\}$
 8: $t \leftarrow 1$
 9: $U_1 \leftarrow X$
10: $\mathit{coverage\_so\_far} \leftarrow 0$
11: $\mathit{rules} \leftarrow []$
12:
13: # Rule Selection
14: **while** $\mathit{coverage\_so\_far} < s$ **do**
15:     $r \leftarrow \text{argmin}_{r \in \mathcal{R} : |U_t \cap C_r| \geq \ell n} \frac{|U_t \cap B_r|}{|U_t \cap C_r|}$
16:     $\mathit{coverage\_so\_far} \leftarrow \mathit{coverage\_so\_far} + |U_t \cap C_r|/n$
17:     $U_{t+1} \leftarrow U_t \setminus C_r$
18:     $t \leftarrow t + 1$
19:     $rules.add(r)$
20: **end while**
21: $rules.add($'else return $\perp$'$)$
22: **return** $rules$
```

parameters are set up. These parameters include C_r, which represents the examples covered by rule r, and B_r, which represents the examples covered by rule r but produce incorrect labels, considered as mistakes. The main part of the algorithm, the rule selection process, continues until a sufficient number of examples, defined by the input s, have been covered. In each iteration, as seen in Line 15, a rule is selected that minimizes the ratio of mistakes to coverage among all rules with a size of at least ℓ, where ℓ is the input threshold. Once the rule selection process is complete, in Line 22, the algorithm returns the list of all chosen rules.

To calculate the total running time of the algorithm, we need to consider two main factors: the number of iterations and the time it takes to identify the relevant rule. The number of iterations, I, is bounded in the next section. At each iteration, we need to find a corresponding rule and update the set U based on the intersection of U_t with B_t and C_t. This process involves going through all rules and finding $|U_t \cap C_r|$ and $|U_t \cap B_r|$, which can take up to $|\mathcal{R}|$ times n in the worst case scenario, where $|\mathcal{R}|$ is the number of rules and n is the size of the training data. Therefore, the overall running time is $O(I \cdot |\mathcal{R}| \cdot n)$.

4. Provable guarantees

In the following section, we will provide provable guarantees on the outcome of Algorithm 1. We start by introducing the setting and main theorem. Then in Section 4.1, we bound the length of the obtained decision list and the empirical mistakes and coverage it achieves. Finally, in Section 4.2, we move to prove the generalization bound.

Throughout this section, let us focus on a decision list DL consisting of rules $R \subset \mathcal{R}$, which covers examples C from the training data, makes errors $B \subseteq C$, and has a size of $k = |R|$:

$$\textbf{if } r_1 \textbf{ then } y_1 \textbf{ else if } r_2 \textbf{ then } y_2 \textbf{ else if } r_3 \ldots,$$

where $r_1, r_2, r_3, \ldots \in \mathcal{R}$. Here, the empirical coverage is $c = |C|/n$ and the empirical mistakes is $b = |B|/|C|$. For the proof, we want to rewrite R as a set of disjoint rules. For this purpose, we define $\mathcal{R}^{(\leq k)}$ as the set of at most k conjunctions from $\mathcal{R}$. Then, the rules in $\mathcal{R}^{(\leq k)}$ corresponding to R are $r_1, \neg r_1 \wedge r_2, \neg r_1 \wedge \neg r_2 \wedge r_3, \ldots$, and we denote this set by $R^* \subseteq \mathcal{R}^{(\leq k)}$. Noticeably, the rules in R^* are disjoint. Thus, the total number of mistakes equals the summation of individual mistakes of rules in R^*.

Theorem 3 (main theorem) *Suppose there is a partial interpretable decision list with length k, b mistakes, coverage c, and with family of rules $\mathcal{R}$. Then for any $0 < \alpha < \beta < 1$ if one runs the algorithm with $\ell = \frac{\alpha}{k} \cdot c$ and $s = (1-\beta)c$ and a family of rules $\mathcal{R}^{(\leq k)}$, then the output is a decision list with at most $\frac{k}{\alpha}$ rules, coverage at least $(1-\beta)c$, and at most $b/(\beta - \alpha)$ mistakes.*

For the specific choice of $\ell = \frac{\alpha}{k}$ and $s = (1-\beta)c$, the number of iterations, I, is bounded by $I \leq \frac{k}{\alpha}$. Consequently, the total running time of the algorithm is $O(\frac{k}{\alpha} \cdot |\mathcal{R}^{(\leq k)}|)$. The running time is dominated by k , which is expected to be small if the decision list we are comparing ourselves to is indeed interpretable. We remark that if plugging $\alpha = 1/4$ and $\beta = 1/2$ in Theorem 3 we can immediately derive the statement made in the abstract.

Our algorithm has an interesting application where we only need explanations for one class, such as an unfavorable decision. In this scenario, all rules in $\mathcal{R}$ have the same classification outcome when used as input for Algorithm 1, and the same theoretical guarantees will apply. This approach is advantageous for interpretability, as each rule in the list can be applied independently of the rules that come before it in the list order.

Partial decision trees as benchmarks. Up until now, we have been using decision lists as a benchmark to compare our algorithm's results. However, some might argue that decision trees would be a better benchmark due to their wide usage for interpretability and self-explanatory nature. Interestingly, decision trees are a subtype of decision lists, as shown in Blum (1992). This demonstrates the broad applicability of our results, as they also apply to decision trees benchmarks. The reason for this is due to the reduction from a decision tree to a decision list. Specifically, a decision tree having k leaves and depth m is equivalent to a decision list with length k, where the depth m represents the complexity of the rules' family. This equivalence is proved by converting each path from the root to a leaf into a rule for the decision list, see Blum (1992) for more details.

4.1. Proof ideas

The proof analyzes the various parameters of the decision list, including its coverage, mistakes, length, and the runtime of the algorithm. It bounds these parameters with respect to the optimal partially interpretable decision list model. Bounding the mistakes is the most challenging aspect which will be the focus of this section. Furthermore, it is imperative to prove the algorithm is well-defined, by establishing the existence of a rule that can be consistently chosen in Line 15. Subsequently, we will demonstrate the primary steps involved in bounding the mistakes and establishing the well-defined nature of the algorithm.

Denote the uncovered training example at iteration t by U_t. The algorithm only chooses *large-covering rules*, where a rule j at time t is large-covering if its coverage, $|C_j \cap U_t|$, is at least

$$|C_j \cap U_t| \geq \frac{\alpha}{k} \cdot cn.$$

We denote the set of all large-covering rules at iteration t that are in R^* as L_t. The first step in the proof is to show that for each iteration t, the union of all the large-covering rules is at least $(\beta - \alpha)|C|$. Namely,

Claim 4 *If one runs Algorithm 1 with $s = (1-\beta)c$ and $\ell = \frac{\alpha}{k} \cdot c$, then for each iteration t, it holds that*

$$Z_t := | \cup_{j \in L_t} C_j \cap U_t| \geq (\beta - \alpha)|C|. \qquad (1)$$

The claim reveals that if β exceeds α, Z_t is greater than zero, indicating that $|L_t|$ is also greater than zero. Whenever $|L_t| > 0$, it must contain at least one rule, making the instruction in Line 15 well-defined. Therefore, one implication of this observation is that the algorithm is well-defined whenever β is greater than α.

We now proceed to the difficult task of bounding the mistakes. At each time t, we look at the current mistake-coverage ratio, where for rule j the ratio is equal to

$$\frac{|B_j \cap U_t|}{|C_j \cap U_t|}.$$

To bound this term we take average weight of all rules in L_t, weighted by their coverage and normalized by Z_t. We use the disjointedness of rules in L_t and the probabilistic method to establish that the ratio of mistakes to coverage at each time t is restricted by the inequality:

$$\frac{|B_t \cap U_t|}{|C_t \cap U_t|} \leq \frac{|B|}{(\beta - \alpha)|C|}. \qquad (2)$$

To obtain the desired result, we sum over all iterations. Additional information is in the appendix.

4.2. Generalization

Up until now, our primary focus has been on minimizing empirical coverage and mistake. Known techniques can leverage these results to bound the generalization error for finite classes. Indeed the class of decision list is finite for finite rule set. Specifically, if the decision lists are of length k and over the rule set $\mathcal{R}$, the number of decision lists is at most $|\mathcal{R}|^k$.

Bounding the generalization error for coverage can be achieved through the use of Hoeffding's inequality and union bound, as detailed in the appendix. By leveraging the fact that our algorithm focuses solely on rules that have sufficient coverage, we are able to establish a generalization bound for the mistakes.

More formally, for a decision list DL in a class $\mathcal{H}$ and n training examples, denote by $c = \Pr_x(DL(x) \neq \perp)$ the true coverage, by $c' = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(DL(x_i) \neq \perp)$ the empirical coverage, by $m = \Pr_{x,y}(DL(x) \neq \perp \wedge DL(x) \neq y)/c$ the true mistakes, and by $m' = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(DL(x_i) \neq \perp \wedge DL(x_i) \neq y_i)/c'$ the empirical mistakes. Then,

Claim 5 *Assume that $c' \geq \gamma$. For any $\epsilon, \delta \in (0,1)$ if there are $n \geq \frac{16k}{\epsilon^2\gamma^2} \log \frac{2|\mathcal{R}|}{\delta}$ examples, then with probability at least $1-\delta$*

$$|c - c'| < \epsilon, \quad \textit{and} \quad \left|m' - m\right| \leq \epsilon$$

5. Experiments

In the following section we empirically evaluate the performance of partial interpretable decision lists on real datasets.

5.1. Datasets, setting, models, and hyperparameters

In Table 2, we provide details on the training of partial interpretable decision lists and black-box models over 13 datasets (further information in Appendix C.1).

Partial interpretable decision lists. The rules are conjunctions of thresholds, every rule condition takes the form of $\bigwedge_{j=1}^{r} x_{i_j} \diamond \theta_j$, where i_j is some feature, $\theta_j \in \mathbb{R}$ is some threshold, and $\diamond$ is some operator $\leq, \geq$, or $=$. Algorithm 1 consists of three hyperparameters: (i) ℓ, a minimum size for the rule coverage considered, (ii) $r^* = \max_{r \in \mathcal{R}} |r|$, the maximum number of conjunctions allowed in the rules, where $|r|$ is the number of conjunctions in the rule r and (iii) the desired coverage. For each dataset, we conducted a grid search to determine the optimal hyperparameters. The values of ℓ considered were $\{0.01, 0.02, 0.05, 0.1, 0.2\}$, while r^* was chosen from $\{1, 2, 3\}$. We performed 3-fold cross-validation for each dataset to select the best values of ℓ and r^*, while maximizing the area under the accuracy vs coverage curve (see Figure 2 and Appendix C.4 for more details on how to calculate the area under the curve). The best performing hyperparameters are presented in Table 2.

To attain the desired balance between accuracy and interpretability in our partially interpretable model, we must minimize the total error that includes examples where the model defaults to the black box model. This is achieved by selecting the appropriate number of rules m such that the total validation error of the partially interpretable model is not greater than the validation error of the black box model plus a threshold value of 0.005. We take the largest m that satisfies this criterion, ensuring that our model is partially interpretable while maintaining high accuracy and coverage.

Black-box model. A grid search was also conducted for each dataset to select the best type of a black-box model, along with its hyperparameters. The evaluated black-box model types included neural network (NN), gradient boosted trees (GBT), support vector machine (SVM), and a decision tree (DT). Similar to the partial interpretable decision lists, a 3-fold cross-validation was performed to identify the best performing model, and the configuration with the highest mean accuracy was chosen. The selected black-box model types are shown in Table 2. Further information about the hyperparameters can be found in Appendix C.2.

5.2. Results

Table 3 illustrates the performance of the partial interpretable decision lists on the test data. This performance is then compared to the test performance of the best opaque model. In Appendix C.5 we also compared it to a baseline which based on decision trees. The experiment was conducted 10 times, using different splits for training and testing.

For each dataset, we provide the average accuracy of the opaque model, along with its standard deviation. We also present the accuracy of the partial interpretable decision list. Additionally, to

Table 2: Dataset properties along with best performing hyperparameters.

Dataset	n	d	Black-box	ℓ	r^*
heart	270	20	SVM	0.02	2
ionosphere	351	34	SVM	0.1	1
breastcancer	683	9	SVM	0.2	1
diabetes	769	8	SVM	0.2	1
mammo	961	14	NN	0.1	3
careval	1728	15	NN	0.02	1
spambase	4601	57	GBT	0.1	2
compasbin	6907	12	NN	0.1	2
mushroom	8124	113	DT	0.2	1
ficobin	10459	17	GBT	0.05	2
adult	32561	36	GBT	0.05	2
bank	41188	57	GBT	0.2	1
bank2	41188	63	GBT	0.05	1

assess the level of interpretability, we disclose the number of interpretable decision rules utilized by the model, as well as the average percentage of test samples covered by these rules, along with their standard deviation.

Across the 13 datasets, the partial interpretable decision list demonstrates a relatively comparable accuracy to the black-box model. The largest decrease in accuracy, observed in the spambase dataset, is only 0.011, while the average (and median) decrease across all datasets is merely 0.002.

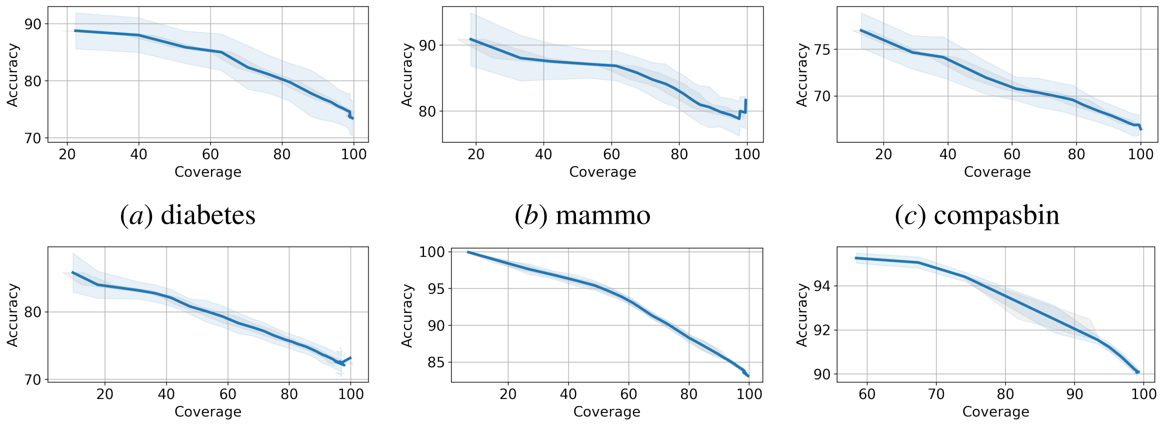
In terms of interpretability, the coverage and the number of rules are the important factors to consider. The heart dataset yields the least interpretable model, covering only 12.4% of the samples with simple decision rules. However, across all datasets, a significant number of partial interpretable decision lists exhibit high interpretability, with an average coverage of 68.6% over the entire dataset set and a median coverage of 80.1%. When it comes to the number of rules, on 10 out of 13 datasets, the number of rules did not exceed 15.

During the execution of Algorithm 1, interpretable decision rules are accumulated iteratively. The process begins with rules that have a low ratio of training mistakes to covered points. As the algorithm progresses, the selected rules tend to have higher ratios until the algorithm reaches its conclusion. Consequently, as the interpretable decision rules cover more samples, their accuracy tends to deteriorate. Figure 2 demonstrates this phenomenon across multiple datasets (the remaining datasets are in Appendix C.3) by showcasing the relationship between the test accuracy and coverage of the interpretable decision rules alone. The figure specifically focuses on samples that are covered by the interpretable decision rules and does not include samples that were defaulted to the opaque model. The curves in the figure are generated by altering the number of decision rules selected by the algorithm.

Furthermore, comparing these curves with the accuracy of the black-box models (as presented in Table 3) highlights that, for large fraction of the samples, the interpretable decision rules alone can achieve comparable performance to an opaque model.

Table 3: Partial interpretable model performance compared to the best black-box model.

Dataset	Black-box model accuracy	Partial interpretable model accuracy	Rules coverage	Number of rules
heart	.821 ± .041	.819 ± .036	.124 ± .071	1
ionosphere	.946 ± .022	.948 ± .020	.175 ± .028	1
breastcancer	.961 ± .007	.957 ± .012	.801 ± .085	4
diabetes	.759 ± .031	.756 ± .038	.626 ± .030	4
mammo	.796 ± .023	.791 ± .026	.953 ± .034	15
careval	.998 ± .003	.998 ± .003	.541 ± .023	2
spambase	.956 ± .005	.944 ± .006	.906 ± .010	20
compasbin	.668 ± .012	.666 ± .013	.999 ± .001	15
mushroom	1.00 ± 0.00	1.00 ± 0.00	.265 ± .003	1
ficobin	.723 ± .014	.721 ± .013	.984 ± .029	28
adult	.840 ± .004	.835 ± .004	.896 ± .026	21
bank	.899 ± .002	.898 ± .003	.991 ± .004	9
bank2	.917 ± .002	.911 ± .003	.666 ± .035	14



(*a*) diabetes (*b*) mammo (*c*) compasbin

(*d*) ficobin (*e*) adult (*f*) bank

Figure 2: Accuracy vs coverage curve. Performance of interpretable rules without defaulting to the opaque model, comparing the fraction of covered points and the corresponding accuracy achieved. The curves represent the effect of increasing the number of rules used, based on the average of 10 random runs with standard deviation in light blue.

6. Conclusion

The paper focuses on partially interpretable decision lists as a hybrid model that offers both performance and interpretability advantages. The paper developed a new algorithm for learning partially interpretable decision lists. It also provided provable guarantees with respect to both interpretability and accuracy compared to the optimal partially interpretable decision list. The algorithm was experimentally evaluated on 13 different datasets and was found to have good accuracy and interpretability compared to black-box models, indicating that the proposed algorithm is effective.

A few questions surrounding the use of partial decision lists prompt further research. Exploring the scenarios where it is possible to formally prove the advantages of *optimal* partially interpretable decision lists in terms of accuracy and interpretability. From the experimental side, can we develop efficient algorithms capable of rapidly pruning any bad rules that may negatively impact performance. By addressing these questions, future research can contribute to the advancement of decision-making models that effectively balance accuracy and interpretability.

Acknowledgments

The authors thank the Simons Institute for the Theory of Computing, as this work was initiated during a visit of some of the authors to the institute. Z.L gratefully acknowledges the NSF (FAI 2040929 and IIS2211955), UPMC, Highmark Health, Abridge, Ford Research, Mozilla, the PwC Center, Amazon AI, JP Morgan Chase, the Block Center, the Center for Machine Learning and Health, and the CMU Software Engineering Institute (SEI) via Department of Defense contract FA8702-15-D-0002, for their generous support of ACMI Lab's research. Y.M would like to thank the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 882396), by the Israel Science Foundation, Tel Aviv University Center for AI and Data Science (TAD), and the Yandex Initiative for Machine Learning at Tel Aviv University.

References

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

Avrim Blum. Rank-r decision trees are a subclass of r-decision lists. *Information Processing Letters*, 42(4):183–185, 1992.

Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *International conference on machine learning*, pages 883–892. PMLR, 2018.

Corinna Cortes, Giulia DeSalvo, Claudio Gentile, Mehryar Mohri, and Scott Yang. Online learning with abstention. In *International conference on machine learning*, pages 1059–1067. PMLR, 2018.

Sanjoy Dasgupta, Nave Frost, and Michal Moshkovitz. Framework for evaluating faithfulness of local explanations. In *International Conference on Machine Learning*, pages 4794–4815. PMLR, 2022.

Daniel Deutch and Nave Frost. Constraints-based explanations of classifications. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 530–541. IEEE, 2019.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(5), 2010.

Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, et al. Concept embedding models: Beyond the accuracy-explainability trade-off. *Advances in Neural Information Processing Systems*, 35:21400–21413, 2022.

Julien Ferry, Gabriel Laberge, and Ulrich Aïvodji. Learning hybrid interpretable models: Theory, taxonomy, and methods. *arXiv preprint arXiv:2303.04437*, 2023.

Shafi Goldwasser, Adam Tauman Kalai, Yael Kalai, and Omar Montasser. Beyond perturbations: Learning guarantees with arbitrary adversarial test examples. *Advances in Neural Information Processing Systems*, 33:15859–15870, 2020.

Kilian Hendrickx, Lorenzo Perini, Dries Van der Plas, Wannes Meert, and Jesse Davis. Machine learning with a reject option: A survey. *arXiv preprint arXiv:2107.11277*, 2021.

Aya Abdelsalam Ismail, Sercan Ö Arik, Jinsung Yoon, Ankur Taly, Soheil Feizi, and Tomas Pfister. Interpretable mixture of experts for structured data. *arXiv preprint arXiv:2206.02107*, 2022.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.

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 international conference on knowledge discovery and data mining*, pages 1675–1684, 2016.

Hui Lan, Tomas M Bosschieter, Zifei Xu, Benjamin Lengerich, Harsha Nori, Kristin Sitcov, Ian Painter, Vivienne Souter, and Rich Caruana. Understanding risk factors for shoulder dystocia using interpretable machine learning. *American Journal of Obstetrics & Gynecology*, 228(1): S753, 2023.

Benjamin Letham, Cynthia Rudin, Tyler H McCormick, and David Madigan. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. 2015.

Lihong Li, Michael L Littman, and Thomas J Walsh. Knows what it knows: a framework for self-aware learning. In *Proceedings of the 25th international conference on Machine learning*, pages 568–575, 2008.

Jimmy Lin, Chudi Zhong, Diane Hu, Cynthia Rudin, and Margo Seltzer. Generalized and scalable optimal sparse decision trees. In *International Conference on Machine Learning*, pages 6150–6160. PMLR, 2020.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Michal Moshkovitz, Yao-Yuan Yang, and Kamalika Chaudhuri. Connecting interpretability and robustness in decision trees through separation. In *International Conference on Machine Learning*, pages 7839–7849. PMLR, 2021.

Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 607–617, 2020.

Danqing Pan, Tong Wang, and Satoshi Hara. Interpretable companions for black-box models. In *International conference on artificial intelligence and statistics*, pages 2444–2454. PMLR, 2020.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Hassan Rafique, Tong Wang, Qihang Lin, and Arshia Singhani. Transparency promotion with model-agnostic linear competitors. In *International Conference on Machine Learning*, pages 7898–7908. PMLR, 2020.

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

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Ronald L Rivest and Robert H Sloan. Learning complicated concepts reliably and usefully. In *AAAI*, pages 635–640, 1988.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

Amin Sayedi, Morteza Zadimoghaddam, and Avrim Blum. Trading off mistakes and don't-know predictions. *Advances in Neural Information Processing Systems*, 23, 2010.

Hugo Henri Joseph Senetaire, Damien Garreau, Jes Frellsen, and Pierre-Alexandre Mattei. Explainability as statistical inference. In *International Conference on Machine Learning*, pages 30584–30612. PMLR, 2023.

Berk Ustun and Cynthia Rudin. Optimized risk scores. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1125–1134, 2017.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

Tong Wang. Gaining free or low-cost interpretability with interpretable partial substitute. In *International Conference on Machine Learning*, pages 6505–6514. PMLR, 2019.

Tong Wang and Qihang Lin. Hybrid predictive models: When an interpretable model collaborates with a black-box model. *The Journal of Machine Learning Research*, 22(1):6085–6122, 2021.

Yair Wiener and Ran El-Yaniv. Agnostic selective classification. *Advances in neural information processing systems*, 24, 2011.

Chicheng Zhang and Kamalika Chaudhuri. The extended littlestone's dimension for learning with mistakes and abstentions. In *Conference on Learning Theory*, pages 1584–1616. PMLR, 2016.

Appendix A. Proof of main theorem

Proof of Theorem 3 Denote the uncovered training example at iteration t by U_t. For each rule j denote by C_j the training examples the rule covers and $B_j \subseteq C_j$ the covered examples that they are mistake. A claim that will be useful throughout this section is that the union of rules in DL with large enough coverage, is large. I.e., for $\alpha \in (0,1)$, the large enough rules at time t are

$$L_t = \left\{ j \in R^* : |U_t \cap C_j| \geq \frac{\alpha}{k}|C| \right\},$$

and their cover size is $Z_t := |\cup_{j \in L_t} U_t \cap C_j|$. Since C_j in R^* are disjoint $Z_t = \sum_j |U_t \cap C_j|$. Claim 7 shows that the large enough rules L_t cover most of the examples, i.e., at least $\beta - \alpha$ of the examples in C that are not covered up until time t.

Claim 6 *If one runs Algorithm 1 with $s = (1-\beta)c$, then at each time t, it holds that*

$$|U_t \cap C| \geq \beta |C|$$

Proof Assume by contradiction that there is iteration t with $|U_t \cap C| < \beta|C|$, then the examples taken so far are at least $(1-\beta)|C|$ examples in C, thus, *coverage_so_far* $\geq (1-\beta)|C|$, so algorithm will not enter the while loop in the t-th iteration. ■

Claim 7 *If one runs Algorithm 1 with $s = (1-\beta)c$ and $\ell = \frac{\alpha}{k}c$, then at each time t, it holds that*

$$Z_t := |\cup_{j \in L_t} U_t \cap C_j| \geq (\beta - \alpha)|C|. \tag{3}$$

Proof The coverage of all the rules in the decision list R that are not in L_t is at most $k \cdot \frac{\alpha}{k}|C| = \alpha|C|$. The coverage $|\cup_{j \in R^*} U_t \cap C_j|$ is equal to $|U_t \cap C|$, as the union of C_j is C. By the previous lemma this is at least $\beta|C|$. Putting the two observations together we get that $Z_t \geq \beta|C| - \alpha|C|$, which is what we wanted to show. ■

The challenging part of proving Theorem 3 is bounding the number of mistakes, which we prove in the following lemma. Specifically, we show that if we run the algorithm with $\ell = \alpha/k \cdot c$ and $s = (1-\beta)c$, for some parameters $0 < \alpha < \beta < 1$), then the number of mistake can increase by at most a factor of about $1/(\beta-\alpha)$.

Lemma 8 *Suppose there is a partial interpretable decision list with length k, b mistakes, coverage c, and with family of rules $\mathcal{R}$. Then for any $0 < \alpha < \beta < 1$ if one runs the algorithm with $\ell = \frac{\alpha}{k} \cdot c$, $s = (1-\beta)c$ and a family of rules $\mathcal{R}^{(\leq k)}$, the output of the algorithm is a decision list with at most $b/(\beta - \alpha)$ mistakes.*

Proof The proof contains several steps that prove the claim.

Bound on the ratio mistakes to coverage at each iteration. The first step in the proof will be to show that at each iteration, the ratio of mistakes to relative cover is bounded. The bound is the total number of mistakes to coverage. More precisely, we will prove that at each time t

$$\frac{|U_t \cap B_t|}{|U_t \cap C_t|} \leq \frac{|B|}{(\beta - \alpha)|C|}, \tag{4}$$

where C_t are the examples covered by the rule chosen at time t and $B_t \subseteq C_t$ are the examples covered with mistake by this rule, at time t.

Any rule j at time t have mistakes-to-coverage ratio $\frac{|U_t \cap B_j|}{|U_t \cap C_j|}$. Take a weighted average of mistakes-to-coverage ratio among all rules in L_t in the following way. The weight of rule $j \in L_t$ is $|U_t \cap C_j|/Z_t$ (recall that Z_t is defined in Equation 3). Hence, among all large enough rules, the average ratio of mistakes to size is equal to

$$\sum_j \frac{|U_t \cap B_j|}{|U_t \cap C_j|} \frac{|U_t \cap C_j|}{Z_t} = \sum_j \frac{|U_t \cap B_j|}{Z_t} \leq \sum_j \frac{|U_t \cap B_j|}{(\beta - \alpha)|C|},$$

where the last inequality follows from Claim 7. From the disjointness of R^*, the weighted-average of mistakes to size is bounded by

$$\frac{|U_t \cap B|}{(\beta - \alpha)|C|} \leq \frac{|B|}{(\beta - \alpha)|C|}.$$

From the probabilistic method there must be a rule in L_t that satisfies this ratio. Since the algorithm takes a large set with the best mistake-to-coverage ratio, for all t we have that $|U_t \cap B_t|/|U_t \cap C_t| \leq |B|/(\beta - \alpha)|C|$.

Bounding total number of mistakes. From the last step, Equation 4, we know that at every time t,

$$|U_t \cap B_t| \leq \frac{|B| \cdot |U_t \cap C_t|}{(\beta - \alpha)|C|}$$

Summing over all times we get a bound on total number of mistakes

$$\sum_t |U_t \cap B_t| \leq \frac{|B|}{|C|(\beta - \alpha)} \sum_t |U_t \cap C_t|$$

The total coverage of the decision list we build is $\sum_t |U_t \cap C_t|$. Thus, the mistake is bounded by

$$\frac{\sum_t |U_t \cap B_t|}{\sum_t |U_t \cap C_t|} \leq \frac{|B|}{|C|} \cdot \frac{1}{\beta - \alpha}$$

■

Now we are ready to prove the main theorem.

Proof (of Theorem 3)

Taking a rule at each step. An immediate implication of the Claim 7, is that at each step there is at least one rule in L_t and thus, the algorithm, and specifically Line 15, is well-defined.

Bounding number of rules. By construction, at each time t, the coverage is large enough. I.e., for all t, $|U_t \cap C_t| \geq \alpha|C|/k$. The algorithm stops once it took at least $(1-\beta)|C|$ examples. Thus after at most $\frac{k(1-\beta)}{\alpha} \leq \frac{k}{\alpha}$ iterations the algorithm stops.

Running time. We proved that the number of iterations is bounded by $\frac{k}{\alpha}$. At each iteration we go over all rules, achieving the desired running time of $O(|\mathcal{R}^{(\leq k)}| \cdot \frac{k}{\alpha})$.

Coverage. Once the algorithm stops, the coverage is at least $(1-\beta)c$. ■

■

Appendix B. Generalization

Proof (of Claim 5) The bound on the coverage for a specific hypothesis follows from Hoeffding's inequality, see the next claim. We can use union bound over all decision lists to get the required bound on the coverage.

As for the mistakes, from Hoeffding's inequality, we know that for every hypothesis in the class it holds that $|mc - m'c'| \leq \epsilon'$, $|c - c'| \leq \epsilon'$.

$$\begin{aligned}
|m - m'| &= \left|\frac{mc}{c} - \frac{m'c'}{c'}\right| \leq \left|\frac{m'c' + \epsilon'}{c' - \epsilon'} - \frac{m'c'}{c'}\right| \\
&= \left|\frac{m'(c')^2 + \epsilon' c' - m'c'(c' - \epsilon')}{(c' - \epsilon')c'}\right| \\
&= \left|\frac{\epsilon' c' + \epsilon' m' c'}{(c' - \epsilon')c'}\right| \leq \frac{2\epsilon' c'}{(c' - \epsilon')c'} = \frac{2\epsilon'}{c' - \epsilon'} \leq \frac{4\epsilon'}{\gamma} \leq \epsilon
\end{aligned}$$

where the second inequality follows from $m'c' \leq c'$, the third from $\epsilon' \leq c'/2$ and $c' \geq \gamma$ and the last from taking $\epsilon' = \frac{\epsilon\gamma}{4}$. ■

For a specific decision list the empirical and true are similar:

Claim 9 *For any hypothesis $h : \mathcal{X} \to \mathcal{Y}$, $a \in \mathcal{Y}$, for any $\epsilon, \delta \in (0,1)$, for $n \geq \frac{1}{\epsilon^2} \log \frac{1}{\delta}$, with probability at least $1 - \delta$*

$$\left|\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}_{h(x)\neq a} - \Pr[h(x) \neq a]\right| \leq \epsilon$$

The claim follows immediately from Hoeffding's inequality.

Fact 10 (Hoeffding's inequality) *For $X_1, \ldots, X_n \in [0,1]$ independent random variables*

$$\Pr\left(\left|\frac{1}{n}\sum_i \mathbb{E}[X_i] - \frac{1}{n}\sum_i X_i\right| > \epsilon\right) < \exp(-2n\epsilon^2)$$

Appendix C. Additional experimental information

C.1. Datasets

We utilized a collection of 13 datasets for our experimental evaluation, which were sourced from Moshkovitz et al. (2021). All the datasets were obtained from publicly available repositories, including:

- https://github.com/yangarbiter/interpretable-robust-trees
- https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/
- https://github.com/ustunb/risk-slim
- https://github.com/chenhongge/RobustTrees
- https://github.com/Jimmy-Lin/GeneralizedOptimalSparseDecisionTrees/tree/master/experiments/datasets

These datasets have been previously used in research on interpretable models (Ustun and Rudin, 2017; Lin et al., 2020; Moshkovitz et al., 2021). In line with the approach described in Moshkovitz et al. (2021), we applied the same scaling and preprocessing methods to these datasets. All features were scaled to the range $[0, 1]$ using the formula $(x - \text{min})/(\text{max} - \text{min})$, where x represents the feature value, and min and max denote the minimum and maximum values of the respective feature across the entire dataset.

Classification tasks All 13 datasets in our study involve binary classification tasks. Below is a list of the specific tasks along with their objectives:

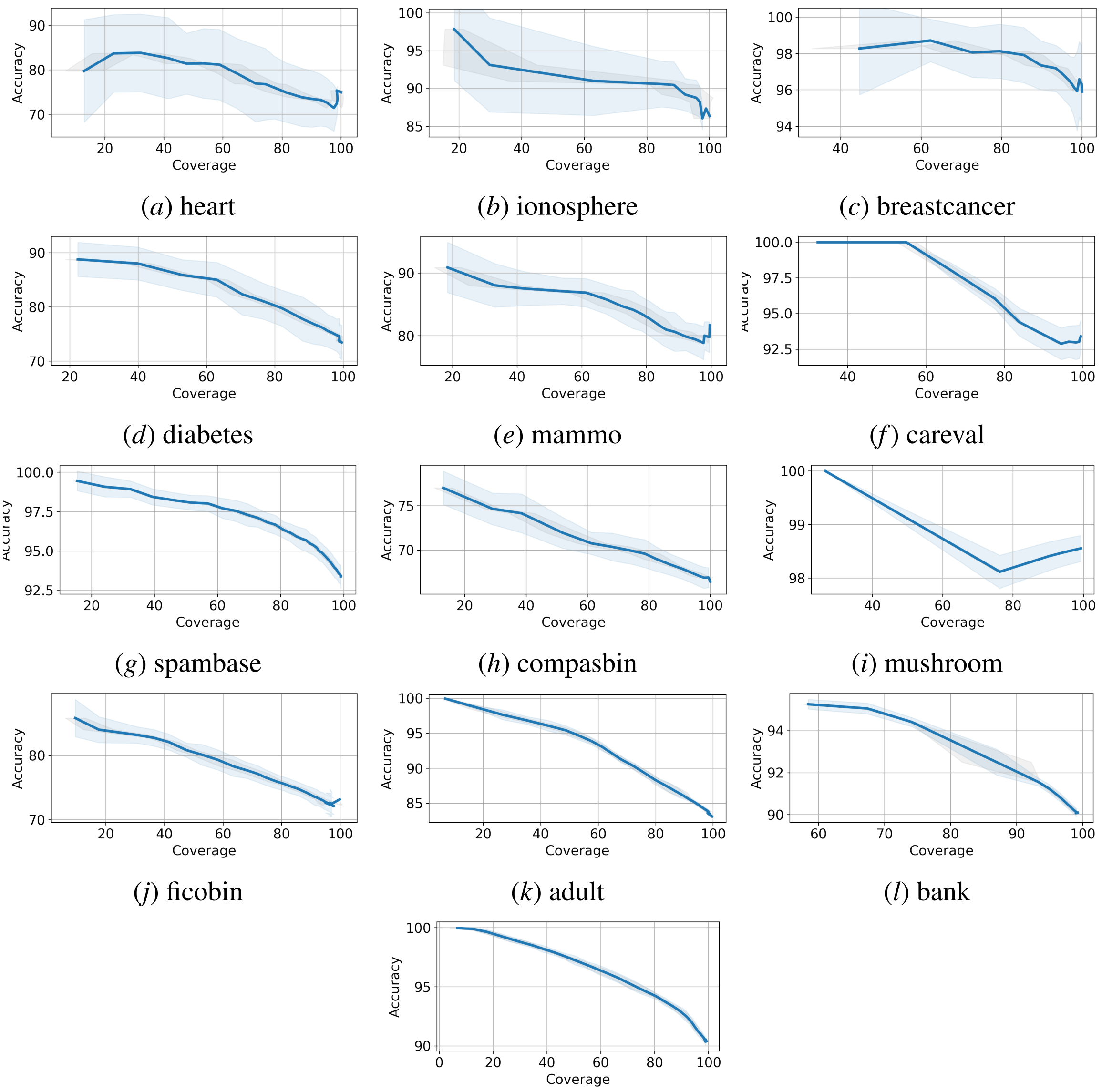
- heart: Detect the presence of heart disease in a patient.
- ionosphere: Identify whether the radar data shows evidence of any structure in the ionosphere.
- breastcancer: Identify whether the given sample is benign.
- diabetes: Predict the presence of diabetes in the patients within the dataset.
- mammo: Predict whether the sample from mammography is malignant.
- careval: Evaluate cars.
- spambase: Predict whether an email is a spam.
- campasbin: Determine if a convicted criminal will re-offend.
- mushroom: Determine whether the mushroom is poisonous.
- ficobin: Predict a person's credit risk.
- adult: Predict whether the person's income is greater than 50,000.
- bank and bank2: Determine if the client opens a bank account after a marketing call.

C.2. Black-box hyperparameters

For the black-box training we have executed a grid search over the following combinations of hyperparameters from `scikit-learn` (Pedregosa et al., 2011).

1. NN
   - $$\begin{aligned} hidden\_layer\_sizes \in &\{[10],[10,10],[10,10,10],[10,10,10,10]\}\cup \\ &\{[100],[100,100],[100,100,100],[100,100,100,100]\}\cup \\ &\{[1000],[1000,100],[1000,200,100]\} \end{aligned}$$
   - $alpha \in \{0.00005, 0.0001, 0.0002\}$
   - $batch\_size \in \{auto, 8, 16\}$
   - $solver \in \{lbfgs, adam\}$
   - $max\_iter \in \{200, 500\}$
2. GBT
   - $learning\_rate \in \{0.01, 0.05, 0.1, 0.15, 0.2\}$
   - $n\_estimators \in \{10, 100, 250, 500, 1000, 2000\}$
   - $max\_depth \in \{3, 5, 7, 9, 11\}$
3. SVM
   - $kernel \in \{linear, rbf\}$
   - $C \in \{0.1, 0.2, 0.5, 1.0, 2.0, 5.0\}$
   - $gamma \in \{scale, auto, 1e-5, 5e-5, 1e-4, 5e-4, 1e-3, 1e-2\}$
4. DT
   - $criterion \in \{gini, entropy, log\_loss\}$
   - $max\_depth \in \{None, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 20\}$
   - $min\_samples\_split \in \{2, 5, 10, 0.1, 0.05, 0.01\}$
   - $min\_impurity\_decrease \in \{0.0, 0.01, 0.02, 0.05, 0.1, 0.2\}$

C.3. Accuracy vs coverage curves



(*a*) heart

(*b*) ionosphere

(*c*) breastcancer

(*d*) diabetes

(*e*) mammo

(*f*) careval

(*g*) spambase

(*h*) compasbin

(*i*) mushroom

(*j*) ficobin

(*k*) adult

(*l*) bank

(*m*) bank2

Figure 3: Accuracy vs coverage curve. Performance of interpretable rules without defaulting to the opaque model, comparing the fraction of covered points and the corresponding accuracy achieved. The curves represent the effect of increasing the number of rules used, based on the average of 10 random runs with standard deviation in light blue.

C.4. Area under the curve for coverage versus accuracy plot

In order to calculate the area under the curve for coverage versus accuracy, we start by constructing a partial explainer (Algorithm 1) with perfect coverage by setting the algorithm's parameter s to 1. Let us denote the number of learned rules as m. Then, for each $i \in \{1, \dots, m\}$, we evaluate the validation accuracy and coverage of the list containing rules $1, \dots, i$ — forming the accuracy versus coverage curve. Ultimately, we compute the area beneath this curve.

C.5. Partially interpretable models comparison

Next we compare between the performance of two partial interpretable models. One model is our proposed solution, employing decision lists, while the other is a baseline utilizing a decision tree. In the decision tree-based approach, we initially learn a complete decision tree for the classification task, subsequently resorting to a non-interpretable black-box model for low accuracy leaves.

In the same fashion as we determine the number of rules in our partially interpretable model, we also choose the leaves in the decision tree-based approach, which default to the black-box model. This selection minimizes the number of leaves that default to the black-box model while guaranteeing that the overall validation error of the partially interpretable model remains within a threshold of 0.005 compared to the validation error of the black box model.

Table 4 presents the mean±std accuracy and coverage for both approaches, along with the number of rules utilized by each model (excluding those defaulting to the black-box model). Our partial interpretable decision list outperforms the decision tree-based approach in terms of accuracy and coverage across 7 out of the 13 datasets. More specifically, in 6 of these datasets, both accuracy and coverage surpass those of the competitor. In the remaining dataset, spambase, while accuracy is slightly lower, coverage is significantly higher. Conversely, the decision tree-based approach performs better on 4 datasets, albeit in 3 of them (adult, bank, and bank2), it does so at the expense of substantially increased rule complexity, which compromises model interpretability. In the remaining 2 datasets, both methods yield similar results.

Table 4: Compare our partially interpretable model, which utilizes decision lists, with the standard decision tree model, wherein low accuracy leaves default to a black-box model. Rows highlighted in green indicate superior performance of our model, while those in red signify the decision tree-based approach's superiority. In the remaining rows, the methods exhibit comparable performance.

	Ours			Decision tree		
Dataset	Accuracy	Coverage	# of rules	Accuracy	Coverage	# of rules
heart	.819 ± .036	.124 ± .071	1	.809 ± .049	.132 ± .085	1
ionosphere	.948 ± .020	.175 ± .028	1	.935 ± .025	.101 ± .227	4
breastcancer	.957 ± .012	.801 ± .085	4	.956 ± .009	.737 ± .242	3
diabetes	.756 ± .038	.626 ± .030	4	.760 ± .032	.167 ± .151	3
mammo	.791 ± .026	.953 ± .034	15	.790 ± .025	.893 ± .137	4
careval	.998 ± .003	.541 ± .023	2	.991 ± .005	.018 ± .005	10
spambase	.944 ± .006	.906 ± .010	20	.951 ± .005	.032 ± .012	32
compasbin	.666 ± .013	.999 ± .001	15	.665 ± .014	.998 ± .004	107
mushroom	1.00 ± 0.00	.265 ± .003	1	1.000 ± .000	1.000 ± .000	15
ficobin	.721 ± .013	.984 ± .029	28	.718 ± .008	.857 ± .071	43
adult	.835 ± .004	.896 ± .026	21	.837 ± .005	.904 ± .033	151
bank	.898 ± .003	.991 ± .004	9	.899 ± .002	.997 ± .005	23
bank2	.911 ± .003	.666 ± .035	14	.913 ± .001	1.000 ± .000	32