Interpreting Blackbox Models via Model Extraction

Osbert Bastani
University of Pennsylvania
obastani@seas.upenn.edu

Carolyn Kim
Stanford University
ckim@cs.stanford.edu

Hamsa Bastani
University of Pennsylvania
hamsab@wharton.upenn.edu

ABSTRACT

The ability to interpret machine learning models has become incredibly important as machine learning is increasingly used to inform consequential decisions. We propose an approach to interpreting complex, blackbox models by constructing *global explanations* that summarize their reasoning process. In our approach, a global explanation is a decision tree that approximates the blackbox model. As long as the decision tree is a good approximation, then the reasoning process of the decision tree mirrors that of the blackbox model. We devise a novel algorithm for extracting decision tree explanations that actively samples new training points to avoid overfitting. We evaluate our algorithm on a random forest to predict diabetes risk, and a learned control policy for the cart-pole problem. Compared to several baselines, the decision trees extracted by our algorithm are substantially more accurate and are equally or more interpretable based on a user study. Finally, we describe several insights we derived based on our interpretations, including a causal issue that we validated with a physician.

ACM Reference Format:
Osbert Bastani, Carolyn Kim, and Hamsa Bastani. 2018. Interpreting Blackbox Models via Model Extraction. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 27 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Machine learning has revolutionized our ability to use data to inform critical decisions, such as medical diagnosis [8, 18, 30], bail decisions for defendants [16, 17] and the design of aircraft collision avoidance systems [27]. At the same time, machine learning models have been shown to exhibit unexpected defects when deployed in the real world, such as causality issues (i.e., inability to distinguish causal effects from correlations) [8, 22], fairness (i.e., internalizing prejudices present in training data) [13, 15], and covariate shift (i.e., differences in the training and test distributions) [6, 25].

Interpretability is a promising approach to address these challenges [12, 24], since it enables data scientists to diagnose issues and verify correctness of machine learning models by providing insight into the model's reasoning [7, 21, 23, 29, 33]. There are three possible approaches to interpretability. First, we can demand that an interpretable model such as a decision tree or rule list be deployed in production [7, 21, 29, 33]. While this approach ensures that the data scientist can manually validate the deployed model, it often means sacrificing some amount of accuracy. Alternatively, we can use a very accurate model such as a random forest in production, but provide a *local explanation* for each prediction made by the classifier [23]. For example, the local explanation may indicate which covariates had the most influence on the model's prediction. The drawback of this approach is that the data scientist now has to independently validate every prediction that is made by the model, or at the very least validate explanations for a large number of predictions. However, many production models are intended to be used automatically, in which case this approach would be impractical.

In this paper, we propose a third approach, which is to extract a *global explanation* that enables the data scientist to interpret the overall reasoning process performed by the model. As long as the global explanation is a good approximation of the model and the data scientist validates the global explanation, then we can deploy a high performing model while simultaneously reducing the possibility of unexpected defects. To maximize the flexibility of our approach, we treat the given model as a *blackbox*, i.e., our approach only requires the ability to run the model on a chosen input, and does not depend on the internal structure of the model. By making minimal assumptions about the model we aim to interpret, we can apply our approach to a broad range of models.

We first have to choose a suitable representation for global explanations. In contrast to local explanations, which are often fairly simple, a global explanation is necessarily complex since it must be able to capture the internal structure of a highly nonlinear function. We use decision trees as global explanations, since decision trees are nonparametric and can compactly represent complex functions.

Then, we require an algorithm for constructing decision trees that are good global explanations. We propose a *model extraction* algorithm that extracts a decision tree closely approximating the given blackbox model. The key challenge is that decision trees are traditionally hard to learn since they tend to overfit the data. Especially in situations where there is very little data available, growing large decision trees is difficult because the data very quickly becomes diluted among the many paths in the tree. To overcome this difficulty, we can leverage the ability to generate arbitrarily large amounts of training data by sampling new inputs and labeling them using the blackbox model. In particular, our algorithm uses *active sampling* to generate inputs that flow down a given path in the decision tree, and then uses these newly generated training points to avoid overfitting. We prove that by generating a sufficient amount of data, the extracted tree converges, implying that it avoids overfitting since the sampling error goes to zero.

We evaluate the accuracy and interpretability of the extracted decision tree explanations on blackbox models trained on two datasets. First, we use a random forest trained to predict diabetes risk. Second, to demonstrate the flexibility of our blackbox approach, we use a control policy for the cart-pole problem trained using reinforcement learning [1]. This model is essentially a table lookup, so

Conference'17, July 2017, Washington, DC, USA
© 2018 Copyright held by the owner/author(s).
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

there is no internal structure that can guide the construction of a global interpretation. In both cases, we show that the decision trees are substantially more accurate than several baselines, including both rule lists and decision trees learned using other algorithms. On the diabetes risk dataset, the F_1 score of our decision tree is 0.31, versus that of the best baseline is only 0.24, and on cart-pole, the F_1 score of our decision tree is 0.94, whereas that of the best baseline is only 0.88. Second, we perform a user study where we ask data scientists to compute counterfactuals, identify risky subpopulations, and identify model defects. This study shows that our decision trees are equally or more interpretable than rule lists and decision sets. Finally, we describe a number of insights we gained by studying our interpretations; in particular, we discovered a causal issue that was validated by a physician. In summary, our contributions are:

- We propose an approach to interpreting the global behavior of blackbox machine learning models (Section 3)
- We devise a model extraction algorithm for constructing global explanations; our algorithm uses an active sampling strategy to avoid overfitting (Section 4).
- We prove that the decision tree learned by our algorithm converges asymptotically, i.e., it avoids overfitting (Section 5).
- We evaluate our algorithm on two datasets, and show that the global explanations constructed by our algorithm are simultaneously far more accurate and equally or more interpretable than several baselines (Section 6).

2 RELATED WORK

Interpretable models. There has been a long history of learning models such as decision trees and rule lists that are interpretable by humans. This approach can be used when accuracy is less important than ensuring that a model does not contain unexpected defects, such as judicial decision making algorithms [17]. For example, axis-aligned decision trees learned using CART [3] are considered highly interpretable, though their accuracy is often lacking. This concern has been tackled by learning rule lists [21, 33] or decision sets [19] that resemble decision trees yet are non-greedy, enabling them to learn more accurate models without sacrificing interpretability.

More closely related to our work, [4] learns decision trees by sampling new points and labeling them using a random forest. However, they use a naïve rejection sampling strategy to sample new inputs, whereas our active sampling strategy directly targets paths most in need of additional data. As we show, this active sampling strategy enables us to substantially improve accuracy. There have also been approaches focused on extracting decision trees from specific model families such as random forests [10, 31, 32], enabling them to leverage domain-specific knowledge about the internal structure of the model; in contrast, our approach is fully blackbox, enabling it to work with any model family.

Interpretable model families based on sparse linear models have also been proposed. LASSO uses L_1 regularization to enforce coefficient sparsity [28]. Alternatively, [29] proposes supersparse linear integer models, which are sparse linear models where the coefficients are integer valued, thus resembling risk-scoring systems constructed manually by humans for applications such as medical diagnosis or criminal rescidivism; [16] extends this approach to classification problems with binary features. Relatedly, [7] proposes generalized additive models, which are linear combinations of arbitrarily complex single-feature models.

Finally, our model extraction algorithm is closely related to *model compression* [5], where a larger, more complex model is used to guide the training of a smaller, more efficient or interpretable model. Our algorithm can be thought of as a model compression algorithm that uses active sampling to train a decision tree.

Local explanations. Another proposed approach is to use the blackbox model, but generate interpretations for every prediction. Such an approach is necessary in applications where sacrificing even a small amount of accuracy can be undesirable, yet every prediction must be interpretable. For example, this category includes critical medical diagnosis tasks where the doctor must be certain that the prediction is correct, yet reducing accuracy is undesirable since lives may be on the line. As an example, given a new test point x, [23] generates an interpretation for the prediction $f(x)$ by fitting an interpretable model locally around x and using it as the explanation for the prediction. Similarly, [26] proposes a method for computing feature relevance scores for deep neural networks. These approaches can help the user understand a specific prediction, but they cannot help understand the model as a whole, making it less useful for diagnosing problems with the model itself.

Global explanations. Our paper takes the third approach, i.e., it gives an explanation of the overall reasoning process performed by the model across all possible predictions. We believe that this approach is useful for a large fraction of data science tasks, where the model is to be used automatically, yet the data scientist wants some assurance that the model is free from unexpected defects. For example, consider a medical prediction task where the goal is to predict patient risk and propose interventions such as offering preventative education, scheduling follow-up visits, or screening patients for potential diseases; this approach can help the data scientist identify potential causal issues in the model. Alternatively, for reinforcement learning tasks, this approach can help the data scientist determine whether a learned control policy generalizes to a new environment. Past techniques have focused on identifying influential features. The *relative influence* scores the contribution of each feature in tree-based models such as random forests [14]. Similarly, [9] uses the Shapley value to quantify the influence of each feature. In our evaluation, we show that these approaches cannot help understand more complex reasoning performed by the model, as is needed to understand the dependence of a model on potentially non-causal features.

In concurrent work, [20] extracts global explanations in the form of decision sets. In contrast, we focus on the case where a large, relatively complex explanation is needed. We compare to their approach in our evaluation, and show that the large decision trees extracted using our algorithm are as interpretable as their decision sets while achieving higher accuracy relative to the blackbox model.

3 PROBLEM FORMULATION

Our algorithm learns axis-aligned decision trees [3]. We start by establishing notation. An *axis-aligned constraint* is a constraint $C = (x_i \leq t)$, where $i \in [d] = \{1, ..., d\}$ and $t \in \mathbb{R}$, where d is the dimension of the input space. More general constraints can be built from existing constraints using negations $\neg C$, conjunctions $C_1 \wedge C_2$, and disjunctions $C_1 \vee C_2$. The *feasible set* of C is $\mathcal{F}(C) = \{x \in \mathcal{X} \mid x \text{ satisfies } C\}$.

A *decision tree* T is a binary tree. An *internal node* $N = (N_L, N_R, C)$ of T has a left child node N_L and a right child node N_R, and is labeled with an axis-aligned constraint $C = (x_i \leq t)$. A *leaf node* $N = (y)$ of T is associated with a label $y \in \mathcal{Y}$. We use N_T to denote the root node of T. The decision tree is interpreted as a function $T : \mathcal{X} \to \mathcal{Y}$ in the usual way. More precisely, a leaf node $N = (y)$ is interpreted as a function $N(x) = y$, an internal node $N = (N_L, N_R, C)$ is interpreted as a function $N(x) = N_L(x)$ if $x \in \mathcal{F}(C)$, and $N(x) = N_R(x)$ otherwise. Then, $T(x) = N_T(x)$. For a node $N \in T$, we let C_N denote the conjunction of the constraints along the path from the root of T to N. More precisely, C_N is defined recursively: for the root N_T, we have $C_{N_T} = \text{True}$, and for an internal node $N = (N_L, N_R, C)$, we have $C_{N_L} = C_N \wedge C$ and $C_{N_R} = C_N \wedge \neg C$.

Then, given a training set $X_{\text{train}} \subseteq \mathcal{X}$ and blackbox access to a function $f : \mathcal{X} \to \mathcal{Y}$, our goal is to learn a decision tree $T : \mathcal{X} \to \mathcal{Y}$ that approximates f. We focus on the case $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = [m]$ (i.e., classification); our approach easily generalizes to the case where $\mathcal{X}$ contains categorical dimensions, and to the case $\mathcal{Y} = \mathbb{R}$ (i.e., regression). For classification, we measure performance using accuracy relative to f on a held out test set, i.e., $\frac{1}{|X_{\text{test}}|} \sum_{x \in X_{\text{test}}} \mathbb{I}[T(x) = f(x)]$. For binary classification, we use F_1 score, and for regression, we use mean-squared error.

4 DECISION TREE EXTRACTION

We give an overview of our decision tree extraction algorithm in Figure 1. At a high level, given a training set $X_{\text{train}} \subseteq \mathcal{X}$ and a blackbox model $f : \mathcal{X} \to \mathcal{Y}$, our algorithm first estimates a distribution $\mathcal{P}$ over the input space $\mathcal{X}$ by fitting a Gaussian mixture model to X_{train}. Then, our algorithm uses $\mathcal{P}$ to generate new samples $x \sim \mathcal{P}$, and it computes the corresponding label $y = f(x)$. In particular, our algorithm uses an active sampling strategy to sample points from $\mathcal{P}$ that are relevant to the node in the decision tree that the algorithm is currently trying to estimate. Finally, our algorithm uses this newly generated data to fit a decision tree. Our decision tree learning algorithm is greedy, both for scalability and because it is a natural fit for interpretability, since more relevant features occur higher in the tree. The decision trees extracted by our algorithm can be pruned using cross-validation, as in CART [3].

In addition, we theoretically characterize how our algorithm avoids overfitting the initial training set X_{train}. To do so, we define the *exact greedy decision tree* T^*, which is intuitively the greedy decision tree extracted from f given infinitely many samples from $\mathcal{P}$. In Section 5, we show that the decision tree extracted by our algorithm asymptotically converges to T^*.

Input distribution. Our algorithm constructs a distribution $\mathcal{P}$ over $\mathcal{X}$ by fitting a mixture of axis-aligned Gaussian distributions to the training data using expectation maximization. It has parameters $\phi \in \mathbb{R}^k$ defining a categorical distribution over $[K]$, and parameters $\mu \in \mathbb{R}^{Kd}$ and $\Sigma \in \mathbb{R}^{Kd^2}$, where $\mu_j \in \mathbb{R}^d$ and $\Sigma_j \in \mathbb{R}^{d^2}$ (where Σ_j is diagonal), for $j \in [K]$, defining the jth Gaussian distribution in the mixture. To sample $x \sim \mathcal{P}$, we first sample $j \sim \text{Categorical}(\phi)$ and then sample $x \sim \mathcal{N}(\mu_j, \Sigma_j)$.

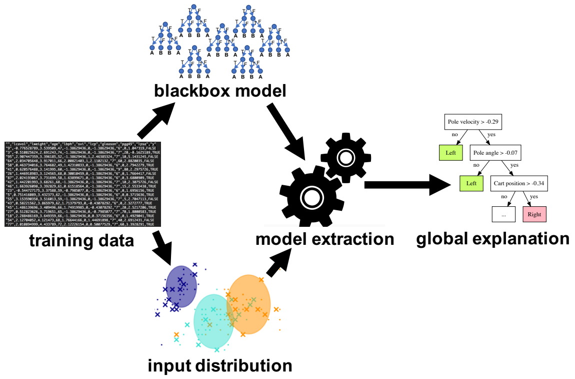


Figure 1: Overview of our decision tree extraction algorithm.

Exact greedy decision tree. We define the *exact greedy decision tree* T^* of size k, which is based on CART [3]. It is initialized to a tree with a single leaf node $N_{T^*} = (y)$, where y is the majority label according to $\mathcal{P}$. The construction then proceeds iteratively—at each iteration, we choose a leaf node $N = (y)$ in the current tree and replace it with an internal node $N' = (N_L, N_R, C)$, where $N_L = (y_L)$ and $N_R = (y_R)$ are leaf nodes, and $C = (x_{i^*} \leq t^*)$ where

$$(i^*, t^*) = \operatorname*{arg\,max}_{i \in [d],\, t \in \mathbb{R}} G(i, t), \tag{1}$$

where the gain

$$\begin{aligned} G(i,t) = &- H(f, C_N \wedge (x_i \leq t)) \\ &- H(f, C_N \wedge (x_i > t)) + H(f, C_N) \end{aligned} \tag{2}$$

$$H(f, C) = \left(1 - \sum_{y \in \mathcal{Y}} \Pr_{x \sim \mathcal{P}}[f(x) = y \mid C]^2\right) \cdot \Pr_{x \sim \mathcal{P}}[C]$$

uses the (weighted) Gini impurity H (which can be replaced with other metrics such as entropy or MSE). The leaf node labels are

$$\begin{aligned} y_L &= \operatorname*{arg\,max}_{y \in \mathcal{Y}} \Pr_{x \sim \mathcal{P}}[f(x) = y \mid C_N \wedge (x_i \leq t)] \\ y_R &= \operatorname*{arg\,max}_{y \in \mathcal{Y}} \Pr_{x \sim \mathcal{P}}[f(x) = y \mid C_N \wedge (x_i > t)]. \end{aligned} \tag{3}$$

The node chosen to be replaced is that with the highest potential gain (2) in the current tree.

To construct the exact greedy decision tree of size k, this process is repeated $k - 1$ times. If at any iteration, the gain (2) is zero, then the process is terminated (so T^* may have fewer than k nodes).

Estimated greedy decision tree. Given $n \in \mathbb{N}$, our algorithm constructs a greedy decision tree in the same way as the construction of the exact greedy decision tree, except that (2) and (3) are estimated using n i.i.d. samples $x \sim \mathcal{P} \mid C_N$ each. Recall that at each iteration, we have to select the node N with the highest potential gain to replace. Our algorithm does so by estimating the gain for all nodes; these estimates are constructed using a separate set of i.i.d. samples to ensure unbiasedness of the tree parameters.

Next, we describe how our algorithm samples $x \sim \mathcal{P} \mid C$, where C is a conjunction of axis-aligned constraints:

$$C = (x_{i_1} \leq t_1) \wedge \ldots \wedge (x_{i_k} \leq t_k) \wedge (x_{j_1} > s_1) \wedge \ldots \wedge (x_{j_h} > s_h).$$

Some inequalities in C may be redundant. First, for two constraints $x_i \le t$ and $x_i \le t'$ such that $t \le t'$, the first constraint implies the second, so we can discard the latter. Similarly, for two constraints $x_i > s$ and $x_i > s'$ such that $s \ge s'$, we can discard the latter. Second, given two constraints $x_i \le t$ and $x_i > s$, we can assume that $t \ge s$; otherwise C is unsatisfiable, so the gain (2) would have been zero and the algorithm would have terminated. In summary, we can assume C contains at most one inequality $(x_i \le t)$ and at most one inequality $(x_i > s)$ per $i \in [d]$, and if both are present, then the two are not mutually exclusive. For simplicity, we assume C contains both inequalities for each $i \in [d]$:

$$C = (s_1 \le x_1 \le t_1) \wedge \ldots \wedge (s_d \le x_d \le t_d).$$

Now, recall that $\mathcal{P}$ is a mixture of axis-aligned Gaussians, so it has probability density function

$$\begin{aligned} p_{\mathcal{P}}(x) &= \sum_{j=1}^{K} \phi_j \cdot p_{\mathcal{N}(\mu_j, \Sigma_j)}(x) \\ &= \sum_{j=1}^{K} \phi_j \prod_{i=1}^{d} p_{\mathcal{N}(\mu_{ji}, \sigma_{ji})}(x_i), \end{aligned}$$

where $\sigma_{ji} = (\Sigma_j)_{ii}$. The conditional distribution is

$$\begin{aligned} p_{\mathcal{P}|C}(x) &\propto \sum_{j=1}^{K} \phi_j \prod_{i=1}^{d} p_{\mathcal{N}(\mu_{ji}, \sigma_{ji})|C}(x_i) \\ &= \sum_{j=1}^{K} \phi_j \prod_{i=1}^{d} p_{\mathcal{N}(\mu_{ji}, \sigma_{ji})|(s_i \le x_i \le t_i)}(x_i). \end{aligned}$$

Since the Gaussians are axis-aligned, the unnormalized probability of each component is

$$\begin{aligned} \tilde{\phi}'_j &= \int \phi_j \prod_{i=1}^{d} p_{\mathcal{N}(\mu_{ji}, \sigma_{ji})|(s_i \le x_i \le t_i)}(x_i) dx \\ &= \phi_j \prod_{i=1}^{d} \left(\Phi\left(\frac{t_i - \mu_{ji}}{\sigma_{ji}} \right) - \Phi\left(\frac{s_i - \mu_{ji}}{\sigma_{ji}} \right) \right), \end{aligned}$$

where Φ is the cumulative density function of the standard Gaussian distribution $\mathcal{N}(0, 1)$. Then, the normalization constant is $Z = \sum_{j=1}^{K} \tilde{\phi}'_j$, and the component probabilities are $\tilde{\phi} = Z^{-1}\tilde{\phi}'$. Finally, to sample $x \sim \mathcal{P} \mid C$, we sample $j \sim \text{Categorical}(\tilde{\phi})$, and

$$x_i \sim \mathcal{N}(\mu_{ji}, \sigma_{ji}) \mid (s_i \le x_i \le t_i) \quad \text{(for each } i \in [d]\text{)}.$$

We use standard algorithms for sampling truncated Gaussian distributions to sample each x_i.

5 THEORETICAL GUARANTEES

We show that our decision tree extraction produces a decision tree that is close to the exact greedy tree for sufficiently large n. A related result is [11], but their analysis is limited to discrete features, for which convergence is much easier to analyze. We give proofs in Appendix A of our extended paper [2]. We begin by describing our assumptions. First, we make mild assumptions about the distribution $\mathcal{P}$.

Assumption 1. The probability density function $p(x)$ of the distribution $\mathcal{P}$ over $\mathcal{X}$ is continuous, bounded (i.e., $p(x) \le p_{\max}$), and has bounded domain (i.e., $p(x) = 0$ for $|x| > x_{\max}$).

To satisfy this assumption, we can truncate the Gaussian mixture models used by our algorithm to $\mathcal{X} = \{x \in \mathbb{R}^d \mid \|x\|_\infty \le x_{\max}\}$, for some $x_{\max} \in \mathbb{R}$. Intuitively, for reasonably large $x_{\max}$, this modification should not affect either the exact greedy tree or the approximate greedy tree by very much, since Gaussian distributions have exponential tails. Our next assumption says that the exact greedy tree is well defined:

Assumption 2. The maximizers (i^*, t^*) in (1), and y_L and y_R in (3) are unique.

In other words, there are no nodes where the Gini impurity for two different choices of branch are exactly tied; such a tie is very unlikely in practice. We now define the notion in which the extracted tree converges to the exact tree. For simplicity, we additionally assume that we are learning complete decision trees of depth D.

Definition 5.1. Let T, T' be complete decision trees of depth D. For $\epsilon > 0$, we say T is an ϵ approximation of T' if $\Pr_{x \sim \mathcal{P}}[T(x) = T'(x)] \ge 1 - \epsilon$. Let T^* denote the exact greedy decision tree that is complete of depth D. For any $\epsilon, \delta > 0$, we say T is (ϵ, δ) exact if

$$\Pr[T \text{ is an } \epsilon \text{ approximation of } T^*] \ge 1 - \delta,$$

where the randomness is taken over the training samples $x \sim \mathcal{P}$.

Theorem 5.2. For any $\epsilon, \delta > 0$, there exists $n \in \mathbb{N}$ such that the decision tree extracted using n samples per node is (ϵ, δ) exact.

6 EVALUATION

We show that the decision trees extracted by our algorithm outperform baseline interpretable models in terms of accuracy relative to the blackbox model, which we refer to as *fidelity*. Higher fidelity ensures that the structure of the global explanation more closely matches the structure of the interpretable model. We compare to decision trees learned using CART [3] and the born-again algorithm [4], rule lists [21, 34], and decision sets [19]. Second, we perform a user study to show how our decision trees are equally or more interpretable than rule lists and decision sets. Finally, we analyze the decision tree explanations extracted using our algorithm, including a causal issue we discovered with the diabetes risk classifier that we have validated with a physician. We give tables with numerical results in Appendix C of our extended paper [2].

6.1 Datasets

We base our evaluation on two blackbox machine learning models; see Table 1 for a summary of the datasets.

Diabetes risk prediction. Our first dataset is a database of patient electronic medical records (EMRs) obtained from a leading EMR provider. The goal is to predict whether a patient has high or low risk for type II diabetes; the presence of a diagnosis of type II diabetes is used as a proxy for diabetes risk. From each patient's EMR data, we construct a number of primarily categorical patient-specific features, including ICD-9 diagnosis codes, prescribed medications, and demographic features. Then, we train a random forest to predict diabetes risk, using 70% of the data for training and 30%

Dataset	Task	# Features	Outcomes	# Training	# Test	Blackbox Model	Blackbox Performance
diabetes risk	classification	384	{high risk, low risk}	404	174	random forest	$F_1 = 0.24$
cart-pole [1]	reinforcement learning	4	{left, right}	100	100	control policy	reward = 200.0

Table 1: Summary of the datasets used in our evaluation.

for testing. Predicting risk is a challenging task, since the dataset is very imbalanced—only 11.8% of the patients in the dataset have been diagnosed with diabetes. Thus, we balance the training set; however, the performance metrics we report are all on an unbalanced test set. To interpret the random forest, we use our algorithm to extract a decision tree. In particular, we fit a Gaussian mixture model $\mathcal{P}$ using the same training data used to estimate the random forest; we then use our algorithm to extract a decision tree, sampling 1000 new training points per node.

Our dataset contains patients from multiple providers. A major problem in healthcare is variation in treatments and ICD-9 diagnosis coding practices across providers; to better understand these variations, we train different models for each provider and then compare their interpretations. For most of our evaluation, we focus on data from the largest provider, which contains the EMRs of 578 patients. We additionally use data from another large provider, which contains the EMRs of 402 patients.

Cart-pole control. Our second dataset consists of samples from the cart-pole control problem [1], where the goal is to balance a pole on top of a cart. We can model this problem as a Markov decision process (MDP), where the state includes the positions and velocities of the cart and the pole. We estimate the MDP transition probabilities and rewards using a large number of random samples, and use value iteration to compute the optimal control policy.

To interpret the control policy, we use our algorithm to extract a decision tree. In particular, we sample 100 training points by executing the control policy, and use these points to fit a Gaussian mixture model $\mathcal{P}$. We use a small training set since our focus is on the small data setting; in practice, obtaining training data for real-world control problems is typically very expensive. We then use our algorithm to extract a decision tree, sampling 200 new training points per node. We use fewer samples compared to the diabetes risk dataset, since the dimension of the inputs in the cart-pole dataset is substantially smaller, making it much easier to avoid overfitting the initial training set. Finally, we sample an additional 100 test points to evaluate the performance of our extracted tree.

6.2 Fidelity

First, we evaluate the fidelity of the decision trees extracted from the blackbox model by our algorithm, which measures the accuracy of the decision tree relative to the blackbox model f. More precisely, the fidelity of our decision tree is the F_1 score on the training set $\tilde{X}_{\text{train}} = \{(x, f(x)) \mid x \in X_{\text{train}}\}$, where X_{train} is the original training set. Achieving high fidelity is important, because it ensures that the insights obtained from the extracted decision tree actually hold for the blackbox model. In particular, we show that the decision trees extracted using our algorithm achieve higher fidelity than a number of baselines, including two existing decision tree learning algorithms, as well as algorithms for learning rule lists and decision sets. All fidelity results reported in this section are the median over 20 random splits of the dataset into training and test sets.

Comparison to other decision trees. First, we compare the fidelity of the decision trees extracted using our algorithm to those extracted using CART [3] and the born-again algorithm [4]. For CART, we train a decision tree using the training set $\tilde{X}_{\text{train}}$. The born-again algorithm is similar to ours—it takes as input a distribution over training points; then, it samples new inputs and labels them using the blackbox model f to enlarge the initial training set. In contrast to our algorithm, which uses an active sampling strategy, the born-again algorithm uses rejection sampling to generate new data. To enable a fair comparison, we use the same Gaussian mixture model $\mathcal{P}$ used by our decision tree extraction algorithm in the born-again algorithm. Furthermore, for the born-again algorithm, we use the same total number of samples that our algorithm uses, i.e., 1000 per node in the case of the diabetes risk dataset, and 200 per node for the cart-pole dataset.

In Figure 2, we show for (a) the diabetes risk dataset, and (b) the cart-pole dataset, the fidelity of our decision trees (black, solid) compared to those trained using CART (red, dotted) and the born-again algorithm (blue, dashed), for a range of sizes, where size equals the total number of internal and leaf nodes in the decision tree. As can be seen, our algorithm outperforms the two baselines in every case. Particularly, for larger decision trees, the additional training data generated by our active sampling algorithm greatly reduces the chances that the decision tree overfits the initial training data. Because the born-again algorithm uses rejection sampling, it is unable to generate a substantial number of new training points at deeper levels of the tree, where the input distribution $\mathcal{P}$ quickly thins out, demonstrating the value of our active sampling strategy.

Comparison to other model families. Next, we compare the fidelity of our decision trees to that of rule lists [33] and decision sets [19, 20] (based on the largest size decision trees for each dataset). We train rule lists using the algorithm proposed in [34], and we train decision sets using the algorithm proposed in [19], both using the original implementation provided by the authors. Unlike our algorithm, the baseline algorithms require us to bin continuous features beforehand; for the diabetes risk dataset, we bin age into seven categories, and for the cart-pole dataset, we use the same bins as the original MDP. The rule list learning algorithm scales to both of our datasets, but the decision set learning algorithm only scales to cart-pole. We found that the decision set learning algorithm does not scale well to datasets with many features; the datasets in their evaluation have tens of features [19], whereas our diabetes risk dataset has hundreds of features. In contrast, the rule list learning algorithm is designed to scale to large datasets (though it is still not as fast as our algorithm since it is non-greedy).

In Figure 2 (c), we compare the fidelity of our algorithm to that of the baselines. In both cases, the performance of our decision

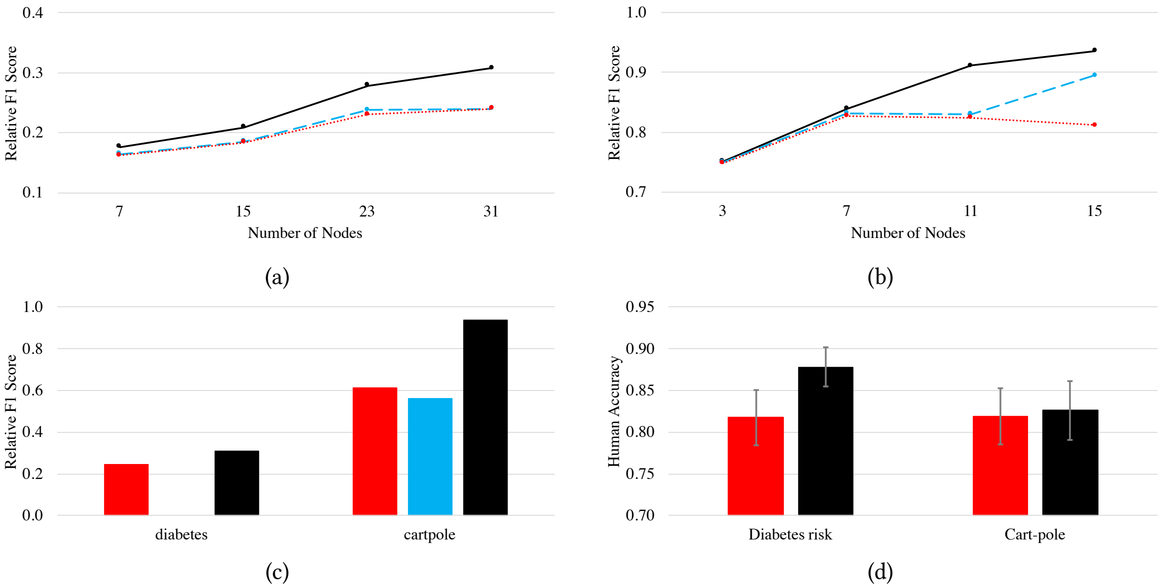


Figure 2: Fidelity on (a) the diabetes risk dataset, and (b) the cart-pole dataset, of decision trees learned using CART (red, dotted), the born-again algorithm (blue, dashed), and our algorithm (black, solid). (c) Fidelity of rule lists (red), decision sets (blue), and our decision trees (black). (d) User response accuracy for the baseline rule list or decision set (red) and our decision trees (black).

tree learning algorithm substantially exceeds the baselines. We found the performance results on the cart-pole dataset to be especially surprising. We believe the difference comes because we have to bin continuous features, and all features in the cart-pole dataset are continuous. Thus, whereas the inputs to the decision tree are 4 dimensional, the inputs to the rule list and decision set are 28 dimensional. This difference makes it easier for these models to substantially overfit the small training set, even compared to algorithms such as CART that do not generate extra training data.

6.3 Interpretability for Diabetes Risk

We performed a user study to evaluate the interpretability of the decision trees learned using our algorithm. The goal of our approach is to enable data scientists familiar with machine learning to understand and validate the blackbox that they train. Thus, we recruited 46 graduate students with a background in machine learning to participate in our study. Each participant answered questions intended to test their understanding of various interpretable models; we asked them to skip a question if they were unable to determine the answer in 1-2 minutes. We show images of our user study interface in Appendix D of our extended paper [2]. We randomized the order of the models and corresponding questions. First, we describe the part of our user study focused on the diabetes risk classifier.

Interpretations. We compare the following two interpretations:

- A decision tree with 31 nodes extracted using our algorithm; a simplified version is shown in Figure 3 (a).
- A rule list trained using [34], shown in Figure 3 (b).

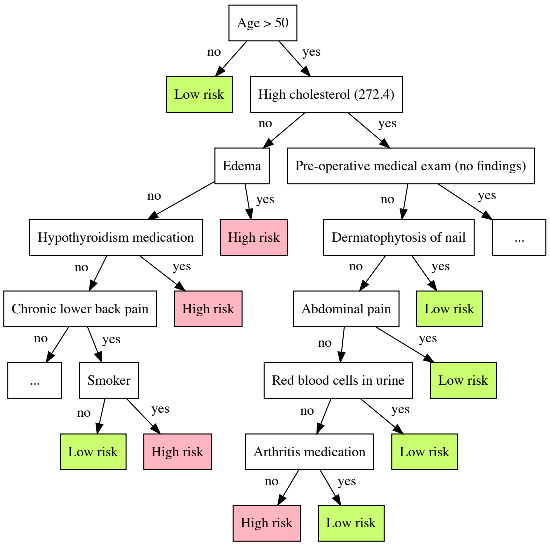
We pre-processed each interpretation by replacing technical feature names with non-technical descriptions.

Questions. We designed five questions to test whether the user could understand a model. We adapted each of these five questions to each of the two models interpreting the random forest. In particular, we adapted the possible answers to each question to fit the specific structure of the model so that there was a single correct answer. Two examples of questions are shown in Figure 5; the variants on the left are for the decision tree, and those on the right are for the rule list. The models used in the study were chosen randomly among the models we generated.

We show two example questions in Figure 5. The first question tests whether the user can determine how the model classifies a given patient. The second question tests whether the user is able to identify the subpopulation for which "Smoker" is a relevant feature; we believe that enabling users to understand these subpopulation-level effects is a major benefit of global explanations.

Results. We show results from our user study in Figure 2 (d) (averaged across questions). Users responded equally or more accurately when using the decision tree, despite the fact that the decision tree was larger than the rule list; this effect was significant ($p = 0.02$ using a paired t-test with 46 samples). Furthermore, for each question, a majority of users answered correctly, so we believe our questions were fair. Finally, we timed two users; in both cases, they responded faster for the decision tree (on average, 44 sec. per decision tree question and 57 sec. per rule list question).

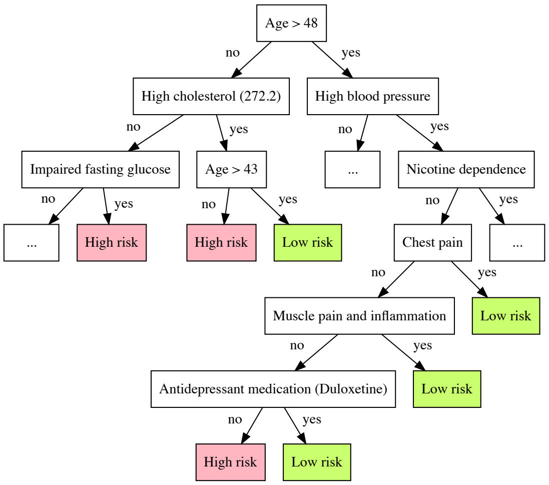
Difficulty with conditional structure. We found that people had particular difficulty understanding the conditional structure of the decision list. In particular, one of our questions required users to determine that only the first three lines of the model were relevant for patients taking arthritis medication. However, this proved challenging for users, who answered correctly only 65% of the time. In



(a)

if Age $<$ 41 **then** Low risk
else if Moderate/severe pain medication (tramadol) **then** High risk
else if Arthritis medication (etodolac) **then** Low risk
else if High cholesterol **and** Smoker **then** High risk
else if High blood pressure **then** High risk
else if Age $<$ 53 **then** Low risk
else if Restless legs syndrome **then** Low risk
else if not High cholesterol **then** Low risk
else High risk

(b)



(c)

Figure 3: Global explanations of diabetes risk classifier: (a) decision tree, (b) rule list, and (c) decision tree for classifier trained on data from an alternate provider.

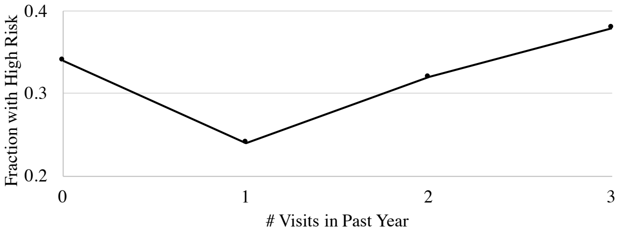


Figure 4: The fraction of patients diagnosed with diabetes among those who visited a doctor n times in past year, for $n \in \{0, 1, 2, 3\}$, conditioned on being over 50 years old and having high cholesterol.

contrast, for the decision tree, users were able to correctly answer a similar question 91% of the time. This finding mirrors previous work showing that reasoning about long sequences of if-then-else rules can be difficult for humans [19].

6.4 Discussion of the Diabetes Risk Classifier

Variations across providers. We can use explanations to understand differences in classifiers trained on data from different providers. In Figure 3 (c), we show a decision tree trained on data from an alternate provider, which contained EMRs for 402 patients. Based on these explanations, we noticed a number of differences in how diagnoses were reported. For example, there are several ICD-9 codes corresponding to high cholesterol; for the original provider, 30% of patients were diagnosed with 272.4 ("unspecified hyperlipidemia"), whereas only 2% of patients were diagnosed with 272.2 ("mixed hyperlipidemia"). In contrast, for the alternate provider, 19% of patients were diagnosed with 272.2, and 18% of patients were diagnosed with 272.4. As another example, for the alternate provider, 10% of patients were diagnosed with "Impaired fasting glucose", which appears in Figure 3 (c). In contrast, for the original provider, only 1% of patients have this diagnosis; indeed, this feature never shows up in explanations extracted for classifiers trained on data from the original provider. Understanding such covariate shifts among the patient populations for different providers can aid data scientists in feature engineering and in adapting existing models to new providers.

Dependence on previous doctor visits. One notable feature of the decision tree shown in Figure 3 (a) is the subtree rooted at the node labeled "Dermatophytosis of nail". In this subtree, if the patient has any of the diagnoses listed, then the decision tree classifies the patient as having low risk for diabetes, but if they do not have any of these diagnoses, then they are classified as high risk. We verified that this effect is present (and quite strong) in the data. Furthermore, this effect occurs across providers—in particular, it also occurs in subtree rooted at "Chest pain" in Figure 3 (c).

This effect is surprising since, in an interview with a physician, we learned that these diagnoses have no known relationship to diabetes risk. As a consequence, this effect is most likely non-causal. After examining the decision tree, the physician suggested a plausible explanation—patients who have these diagnoses are more likely to have visited a doctor at least once in the past year, upon which the doctor may have recommended pre-diabetic interventions that

Consider patients over 50 years old who are otherwise healthy and are not taking any medications. According to the decision tree, are these patients at a high risk for diabetes?

- Yes
- No

Smoking is known to increase risk of diabetes, so the local hospital has started a program to help smokers quit smoking. According to the decision tree, which patient subpopulation should we target in this program if we want to reduce diabetes risk?

- Patients over 50 years old who have high cholesterol
- Patients over 50 years old who have chronic lower back pain
- Patients over 50 years old who have high cholesterol, edema, chronic lower back pain, and who take medication for hypothyroidism

Consider patients over 53 years old who are otherwise healthy and are not taking any medications. According to the rule list, are these patients at a high risk for diabetes?

- Yes
- No

Smoking is known to increase risk of diabetes, so the local hospital has started a program to help smokers quit smoking. According to the rule list, which patient subpopulation should we target in this program if we want to reduce diabetes risk?

- Patients over 41 years old
- Patients over 41 years old who have high cholesterol
- Patients over 41 years old who have high cholesterol, and take medication for arthritis

Figure 5: Examples of questions asked in our user study on the diabetes risk classifier, for our decision tree (left) and for the rule list (right).

reduced the patient's risk for diabetes. In contrast, patients who have not visited a doctor in the past year may not have realized they were at high risk for diabetes, especially since this subtree is conditioned on patients who are over 50 years old and have high cholesterol, which are both known risk factors for diabetes.

In Figure 4, we plot the relationship between the number of doctor visits in the past year and incidence of diabetes (restricted to patients over 50 years old who have high cholesterol). As the physician suspected, we find a surprising V-shaped effect—while diabetes risk typically increases with the number of doctor visits (since the patient is likely to be sicker), the risk is actually higher for patients with no doctor visits compared to patients with a single doctor visit. This finding suggests that our interpretation was valuable for the domain expert to form and test hypotheses about unexpected defects in the classifier.

As pointed out in prior work, awareness of such a non-causal effect is important [8]. In particular, it is likely that many of the patients in this subpopulation have already received pre-diabetic interventions. For these patients, the classifier may incorrectly decide to discontinue these pre-diabetic interventions, and such a decision could be dangerous for the patient. In the case of diabetes risk prediction, the patient's doctor would likely review the recommendations made by the classifier, thus mitigating this issue; however, for models intended to be used automatically, it is important that a data scientist identify these kinds of issues.

We note that influence scores alone are insufficient to tease out such an effect, since they do not examine patient subpopulations. In particular, the effect we described only applies to the subpopulation of patients that are at least 50 years old and have high cholesterol. For example, the correlation of "Abdominal pain" with high risk of diabetes is 8.1×10^{-3}; however, conditioned on age greater than 50 and having high cholesterol, the correlation is -9.8×10^{-2}. Indeed, none of the features in this subtree appear in the top 40 relative influence scores for the random forest.

Non-monotone dependence on age. Note that age appears twice in the decision tree in Figure 3 (c). Typically, younger patients are at lower risk for diabetes; however, conditioned on being less than 48 years old and having high cholesterol, the classifier predicts higher risk for younger patients. While we cannot be certain of the cause, there are a number of possible explanations. For example, it may be the case that a diagnosis of high cholesterol in younger patients is abnormal and therefore much more indicative of high diabetes risk. Alternatively, doctors may be more likely to urge older patients with high cholesterol to take preventative measures to reduce diabetes risk.

This structure demonstrates how the decision tree can capture non-monotone dependencies on continuous features such as age. In contrast, non-monotone dependencies cannot be captured by influence scores. Rule lists can also capture such a dependence, but their restricted structure makes it more difficult to understand the effect—for example, to reason about the relationship between the first and sixth rules in the rule list in Figure 3 (b), we have to reason about the four intermediate rules as well.

6.5 Interpretability for Cart-pole

Next, we describe the part of our user study focused on the cart-pole control policy.

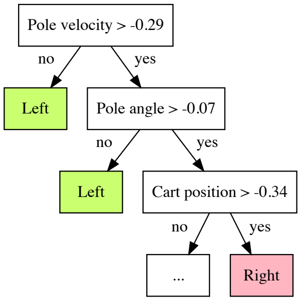
Interpretations. We compare the following two interpretations:

- A decision tree with 15 nodes extracted using our algorithm; a simplified version is shown in Figure 6 (a).
- A decision set trained using [19], shown in Figure 6 (b).

Questions. We designed three questions to test whether the user could understand a model. Similar to the questions regarding the interpretations of the diabetes risk classifier, we adapt each of these questions to each of the two models interpreting the control policy. In addition, one of the questions tests whether the user can compute how the model classifies a given state, and the remaining two questions test whether the user can reason about potential symmetries of the model. For example, one of these questions asks:

> In theory, the action taken should not depend on the position of the cart. Does the decision tree satisfy this property?

Results. We show results from our user study in Figure 2 (d) (averaged across questions as before). As can be seen, users responded equally or more accurately when using the decision tree, despite the fact that the decision tree was somewhat larger than the decision sets. In this case, the overall effect was not significant, but nevertheless the decision tree was not less interpretable than the decision set. As before, for each question, a majority of users answered correctly, so we believe our questions were fair. Finally, we timed two users; one responded faster for the decision tree, and the other responded faster for the decision set (on average, 32 sec. per decision tree question and 35 sec. per decision set question).



(a)

if $0.29 <$ Pole angle < 0.86 **then** Left
if $-1.29 <$ Cart velocity < -0.43 **then** Left
if $0.43 <$ Cart velocity < 1.29 **then** Left
if $-0.34 <$ Cart position < 0.34 **then** Right
default Left

(b)

Figure 6: Global explanations of the cart-pole policy: (a) decision tree, and (b) decision set.

6.6 Discussion of the Cart-Pole Control Policy

Translation invariance. We expect that the motion of the cart-pole should be invariant to translating the cart position. However, it is easy to see from both the decision tree and the decision set in Figure 6 that the learned policy does not exhibit this symmetry. This asymmetry likely arises because the MDP simulation always starts from the same initial position. Thus, the cart position is highly correlated with its velocity, the control policy can use the two interchangeably to predict what action to take. Understanding this bias in the control policy is important because it may not generalize well if the initial position changes.

Reflection invariance. We also expect the motion of the cart-pole to be invariant to reflection across the y-axis (i.e., flip left and right). However, the models in Figure 6 do not exhibit this symmetry. This asymmetry likely arises because in the MDP simulation, the pole is always initially falling toward the left. As a consequence, to maximize performance, the control policy focuses on stopping the pole from falling toward the left, which requires moving the cart toward the left. As before, the control policy may not generalize well if we change the initial direction in which the pole is falling.

7 CONCLUSION

We have proposed an approach for interpreting blackbox models based on decision tree extraction, and shown how it can be used to interpret both random forests and control policies. Important directions for future work include devising algorithms for model extraction using more expressive input distributions, and developing new ways to gain insight from the extracted decision trees.

REFERENCES

[1] Andrew G Barto, Richard S Sutton, and Charles W Anderson. 1983. Neuron-like Adaptive elements that can solve difficult learning control problems. *IEEE transactions on systems, man, and cybernetics* (1983).

[2] Osbert Bastani, Kim Carolyn, and Hamsa Bastani. 2018. Interpreting Blackbox Models via Model Extraction. *arXiv:1705.08504* (2018).

[3] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. 1984. *Classification and regression trees.* CRC press.

[4] Leo Breiman and Nong Shang. 1996. Born again trees. *University of California, Berkeley, Berkeley, CA, Technical Report* (1996).

[5] Cristian Bucilua, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *KDD.*

[6] J Quiñonero Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. 2009. Dataset shift in machine learning. (2009).

[7] Rich Caruana, Yin Lou, and Johannes Gehrke. 2012. Intelligible Models for Classification and Regression. In *KDD.*

[8] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *KDD.*

[9] Anupam Datta, Shayak Sen, and Yair Zick. 2016. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *IEEE Symposium on Security and Privacy.*

[10] Houtao Deng. 2014. Interpreting tree ensembles with intrees. *arXiv:1408.5456* (2014).

[11] Pedro Domingos and Geoff Hulten. 2000. Mining high-speed data streams. In *KDD.*

[12] Finale Doshi-Velez and Been Kim. 2017. A Roadmap for a Rigorous Science of Interpretability. *arXiv:1702.08608* (2017).

[13] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *ITCS.*

[14] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[15] Moritz Hardt, Eric Price, Nati Srebro, et al. 2016. Equality of opportunity in supervised learning. In *NIPS.*

[16] Jongbin Jung, Connor Concannon, Ravi Shroff, Sharad Goel, and Daniel Goldstein. 2017. Simple rules for complex decisions. *arXiv:1702.04690* (2017).

[17] Jon Kleinberg, Himabindu Lakkaraju, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. 2017. *Human decisions and machine predictions.* Technical Report. National Bureau of Economic Research.

[18] Igor Kononenko. 2001. Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in medicine* (2001).

[19] Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. 2016. Interpretable decision sets: A joint framework for description and prediction. In *KDD.*

[20] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. 2017. Interpretable & Explorable Approximations of Black Box Models. In *FAT/ML.*

[21] Benjamin Letham, Cynthia Rudin, Tyler H McCormick, David Madigan, et al. 2015. Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics* 9, 3 (2015), 1350–1371.

[22] Judea Pearl. 2009. *Causality.* Cambridge university press.

[23] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why Should I Trust You?: Explaining the Predictions of Any Classifier. In *KDD.*

[24] Cynthia Rudin. 2014. Algorithms for interpretable machine learning. In *KDD.*

[25] Hidetoshi Shimodaira. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference* 90, 2 (2000), 227–244.

[26] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning Important Features Through Propagating Activation Differences. In *ICML.*

[27] Selim Temizer, Mykel Kochenderfer, Leslie Kaelbling, Tomas Lozano-Pérez, and James Kuchar. 2010. Collision avoidance for unmanned aircraft using Markov decision processes. In *AIAA guidance, navigation, and control conference.* 8040.

[28] Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* (1996), 267–288.

[29] Berk Ustun and Cynthia Rudin. 2016. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning* 102, 3 (2016), 349–391.

[30] Gilmer Valdes, José Marcio Luna, Eric Eaton, Charles B Simone, et al. 2016. MediBoost: a Patient Stratification Tool for Interpretable Decision Making in the Era of Precision Medicine. *Scientific Reports* 6 (2016).

[31] Anneleen Van Assche and Hendrik Blockeel. 2007. Seeing the Forest through the Trees. In *ILP.*

[32] Gilles Vandewiele, Olivier Janssens, Femke Ongenae, Filip De Turck, and Sofie Van Hoecke. 2016. GENESIM: genetic extraction of a single, interpretable model. *arXiv:1611.05722* (2016).

[33] Fulton Wang and Cynthia Rudin. 2015. Falling Rule Lists. In *AISTATS.*

[34] Hongyu Yang, Cynthia Rudin, and Margo Seltzer. 2017. Scalable Bayesian Rule Lists. In *ICML.*

A PROOFS OF MAIN RESULTS

In this section, we give a proof of Theorem 5.2.

A.1 Proof of Main Lemmas

In this section, we state and prove the main lemmas required to prove Theorem 5.2. First, a key definition in our proof is the *gap* of the exact greedy decision tree.

Definition A.1. Let $\mathcal{P}$ be a distribution over $\mathbb{R}$, and let $F(t)$ be an unnormalized cumulative distribution function for $\mathcal{P}$. We say $g : \mathbb{R} \to \mathbb{R}$ is (ϵ', δ') *gapped* according to $\mathcal{P}$ if it has a unique maximizer $t^* = \arg\max_{t\in\mathbb{R}} g(t)$, and for every $t \in \mathbb{R}$ such that $|F(t) - F(t^*)| > \epsilon'$, we have $g(t^*) > g(t) + \delta'$.

Now, we prove the following main lemma, which is essentially the inductive steps needed to prove our main results.

Lemma A.2. Let $\mathcal{X} \subseteq \mathbb{R}^d$, and let $\mathcal{P}$ and $\tilde{\mathcal{P}}$ be distributions over $\mathcal{X}$, and let $p(x)$ and $\tilde{p}(x)$ be unnormalized probability density functions for $\mathcal{P}$ and $\tilde{\mathcal{P}}$, respectively. The function $\tilde{p}$ depends on a parameter $n \in \mathbb{N}$; in applications of this lemma, $\tilde{p}$ is an estimate of p computed by our algorithm on n samples. Assume that for any $\epsilon, \delta > 0$, there exists $n \in \mathbb{N}$ such that $\|p - \tilde{p}\|_1 \le \epsilon$ with probability at least $1 - \delta$.

Furthermore, let $i \in [d]$, and define

$$G(t) = -\sqrt{\sum_{s\in\{\pm 1\}} \sum_{y\in\mathcal{Y}} \frac{g_{s,y}(t)^2}{h_s(t)}}$$

$$\tilde{G}(t) = -\sqrt{\sum_{s\in\{\pm 1\}} \sum_{y\in\mathcal{Y}} \frac{\tilde{g}_{s,y}(t)^2}{\tilde{h}_s(t)}}$$

$$\hat{\tilde{G}}(t) = -\sqrt{\sum_{s\in\{\pm 1\}} \sum_{y\in\mathcal{Y}} \frac{\hat{\tilde{g}}_{s,y}(t)^2}{\tilde{h}_s(t)}},$$

where

$$h_s(t) = \int \mathbb{I}[s \cdot x_i \le t] \cdot p(x) dx$$

$$\tilde{h}_s(t) = \int \mathbb{I}[s \cdot x_i \le t] \cdot \tilde{p}(x) dx$$

$$g_{s,y}(t) = \int \mathbb{I}[f(x) = y] \cdot \mathbb{I}[s \cdot x_i \le t] \cdot p(x) dx$$

$$\tilde{g}_{s,y}(t) = \int \mathbb{I}[f(x) = y] \cdot \mathbb{I}[s \cdot x_i \le t]] \cdot \tilde{p}(x) dx$$

$$\hat{\tilde{g}}_{s,y}(t) = \frac{Z}{n} \sum_{i=1}^{n} \mathbb{I}[f(x^{(k)}) = y] \cdot \mathbb{I}[s \cdot x_i^{(k)} \le t],$$

where $x^{(1)}, ..., x^{(n)}$ are i.i.d. samples from $\tilde{\mathcal{P}}$, and where

$$Z = \int \tilde{p}(x) dx$$

is a normalization constant. Now, let

$$t^* = \arg\max_{t\in\mathbb{R}} G(t)$$

$$\tilde{t}^* = \arg\max_{t\in\mathbb{R}} \hat{\tilde{G}}(t),$$

and let

$$p'(x) = p(x) \cdot \mathbb{I}[x_i \le t^*]$$

$$\tilde{p}'(x) = \tilde{p}(x) \cdot \mathbb{I}[x_i \le \tilde{t}^*]$$

be unnormalized probability density functions for $\mathcal{P} \mid (x_i \le t^*)$ and $\tilde{\mathcal{P}} \mid (x_i \le \tilde{t})$, respectively.

Assume that $\|h_s\|_\infty \ge \gamma > 0$ for each $s \in \{\pm 1\}$. For any $\epsilon', \delta' > 0$, there exists $n \in \mathbb{N}$ such that

$$\|p' - \tilde{p}'\|_1 \le \epsilon'$$

with probability at least $1 - \delta'$.

Proof. Let $\epsilon', \delta' > 0$ be arbitrary. First, we show that the L_∞ norm of each $g = g_{s,y}$ is bounded with high probability. By Lemma B.3, we have

$$\|g - \hat{\tilde{g}}\|_\infty \leq \|g - \tilde{g}\|_\infty + \|\tilde{g} - \hat{\tilde{g}}\|_\infty \leq \|g - \tilde{g}\|_\infty + \frac{4 \log n}{\sqrt{n}}$$

with probability at least $1 - 2n^{-3/2}$. Furthermore, by Lemma B.2 and by our assumption, for any $\epsilon, \delta > 0$, there exists $n \in \mathbb{N}$ such that

$$\|g - \hat{\tilde{g}}\|_\infty \leq \epsilon + \frac{4 \log n}{\sqrt{n}}$$

with probability at least $1 - 2n^{-3/2} - \delta$. Similarly, for $h = h_s$, by Lemma B.2 and by our assumption, using the same (ϵ, δ), we have $\|h - \tilde{h}\|_\infty \leq \epsilon$. In particular, it follows that $\|\tilde{h}\|_\infty \geq \gamma - \epsilon$. Next, let

$$\begin{aligned} g(t) &= \operatorname*{arg\,max}_{s \in \{\pm 1\}, y \in \mathcal{Y}} \frac{g_{s,y}(t)}{h_s(t)} \\ \hat{\tilde{g}}(t) &= \operatorname*{arg\,max}_{s \in \{\pm 1\}, y \in \mathcal{Y}} \frac{\hat{\tilde{g}}_{s,y}(t)}{\tilde{h}_s(t)}. \end{aligned}$$

Then, note that

$$\begin{aligned} \left\| \frac{1}{\sqrt{h_s}} - \frac{1}{\sqrt{\tilde{h}_s}} \right\|_\infty = \left\| \frac{\sqrt{\tilde{h}_s} - \sqrt{h_s}}{\sqrt{h_s \cdot \tilde{h}_s}} \right\|_\infty &= \left\| \frac{\tilde{h}_s - h_s}{\sqrt{h_s \cdot \tilde{h}_s}} \right\|_\infty \cdot \left\| \sqrt{\tilde{h}_s} + \sqrt{h_s} \right\|_\infty^{-1} \\ &\leq \frac{\epsilon}{\sqrt{\gamma \cdot (\gamma - \epsilon)}} \cdot \left(\frac{1}{\sqrt{\gamma - \epsilon}} + \frac{1}{\sqrt{\gamma}} \right) \\ &\leq \frac{\epsilon}{2(\gamma - \epsilon)^{3/2}}, \end{aligned}$$

from which it follows that

$$\begin{aligned} \left\| \frac{g_{s,y}}{\sqrt{h_s}} - \frac{\hat{\tilde{g}}_{s,y}}{\sqrt{\tilde{h}_s}} \right\|_\infty &= \left\| \frac{g_{s,y}}{\sqrt{h_s}} - \left(\frac{1}{\sqrt{\tilde{h}_s}} + \frac{1}{\sqrt{h_s}} - \frac{1}{\sqrt{h_s}} \right) \cdot \hat{\tilde{g}}_{s,y} \right\|_\infty \\ &\leq \left\| \frac{g_{s,y} - \hat{\tilde{g}}_{s,y}}{\sqrt{h_s}} \right\|_\infty + \left\| \left(\frac{1}{\sqrt{\tilde{h}_s}} - \frac{1}{\sqrt{h_s}} \right) \cdot \hat{\tilde{g}}_{s,y} \right\|_\infty \\ &\leq \frac{\epsilon}{\sqrt{\gamma}} + \frac{4 \log n}{\sqrt{\gamma} \cdot \sqrt{n}} + \frac{\epsilon}{2(\gamma - \epsilon)^{3/2}} \\ &\leq \frac{2\epsilon}{(\gamma - \epsilon)^{3/2}} + \frac{4 \log n}{\sqrt{\gamma} \cdot \sqrt{n}}. \end{aligned}$$

Therefore, letting $m = |\mathcal{Y}|$, we have

$$\begin{aligned}
&\|G - \hat{\tilde{G}}\|_\infty \\
&\le \left\| \sqrt{\sum_{s\in\{\pm 1\}} \sum_{y\in\mathcal{Y}} \frac{g_{s,y}^2}{h_s}} - \sqrt{\sum_{s\in\{\pm 1\}} \sum_{y\in\mathcal{Y}} \frac{\hat{\tilde{g}}_{s,y}^2}{\tilde{h}_s}} \right\|_\infty \\
&= \left\| \sum_{s\in\{\pm 1\}} \sum_{y\in\mathcal{Y}} \frac{g_{s,y}^2}{h_s} - \sum_{s\in\{\pm 1\}} \sum_{y\in\mathcal{Y}} \frac{\hat{\tilde{g}}_{s,y}^2}{\tilde{h}_s} \right\|_\infty \cdot \left\| \sqrt{\sum_{s\in\{\pm 1\}} \sum_{y\in\mathcal{Y}} \frac{g_{s,y}^2}{h_s}} + \sqrt{\sum_{s\in\{\pm 1\}} \sum_{y\in\mathcal{Y}} \frac{\hat{\tilde{g}}_{s,y}^2}{\tilde{h}_s}} \right\|_\infty^{-1} \\
&\le \sum_{s\in\{\pm 1\}} \sum_{y\in\mathcal{Y}} \left\| \frac{g_{s,y}^2}{h_s} - \frac{\hat{\tilde{g}}_{s,y}^2}{\tilde{h}_s} \right\|_\infty \cdot \|g + \hat{\tilde{g}}\|_\infty^{-1} \\
&\le \sum_{s\in\{\pm 1\}} \sum_{y\in\mathcal{Y}} \left\| \frac{g_{s,y}}{\sqrt{h_s}} - \frac{\hat{\tilde{g}}_{s,y}}{\sqrt{\tilde{h}_s}} \right\|_\infty \\
&\le \frac{4m\epsilon}{(\gamma - \epsilon)^{3/2}} + \frac{8m \log n}{\sqrt{\gamma} \cdot \sqrt{n}}.
\end{aligned}$$

Now, let $\mathcal{P}_i$ be the marginal probability distribution of $\mathcal{P}$ along dimension i, let

$$p_i(t) \propto \int p(x_1, ..., x_{i-1}, t, x_{i+1}, ..., x_d) dx_1 ... dx_{i-1} dx_{i+1} ... dx_d$$

be an unnormalized marginal probability density function for $\mathcal{P}_i$, and let

$$F_i(t) = \int_{-\infty}^{t} p_i(t') dt'$$

be the corresponding unnormalized cumulative distribution function.

Next, let $\epsilon'' > 0$ be arbitrary; we choose its value later. By Lemma B.1, for any $\epsilon'' > 0$, there exists $\delta_G(\epsilon'') > 0$ such that G is $(\epsilon'', \delta_G(\epsilon''))$ gapped according to the function p_i.

Recall that we have assumed that ϵ and δ can be made arbitrarily close to zero by taking n to be sufficiently large. In other words, there exists $n \in \mathbb{N}$ such that both of the following hold:

$$\frac{4m}{(\gamma - \epsilon)^{3/2}} \cdot \epsilon + \frac{8m \log n}{\sqrt{\gamma} \cdot \sqrt{n}} \le \frac{\delta_G(\epsilon'')}{2} \tag{4}$$

$$\delta + \frac{2}{n^{3/2}} \le \delta', \tag{5}$$

in which case

$$\|G - \hat{\tilde{G}}\|_\infty \le \frac{\delta_G(\epsilon'')}{2}$$

with probability at least $1 - \delta'$. Then, by Lemma B.4, we have

$$|F_i(\tilde{t}^*) - F_i(t^*)| \le \epsilon'',$$

so by Lemma B.5, we have

$$\|p' - \tilde{p}'\|_1 \le \epsilon + \epsilon''.$$

Thus, the claim follows taking $\epsilon' = \epsilon + \epsilon''$. □

Corollary A.3. Assume the same setup as in Lemma A.2. Assume that $\gamma \ge 2\epsilon$, and suppose that

$$\delta_g(\epsilon'') \ge \kappa\epsilon''$$

for all $\epsilon'' > 0$ and some constant $\kappa > 0$. For any $\epsilon', \delta' > 0$, we have $\|p' - \tilde{p}\|_1 \le \epsilon'$ with probability at least $1 - \delta'$ as long as $\|p - \tilde{p}\|_1 \le \epsilon$ with probability at least δ, where

$$\begin{aligned}
\epsilon &\le \left(1 + \frac{4m}{(\gamma/2)^{3/2}} \cdot \frac{4}{\kappa}\right)^{-1} \cdot \epsilon' \\
\delta &\le \frac{1}{2} \cdot \delta',
\end{aligned}$$

and $n \geq 3$ satisfies

$$\epsilon \geq \frac{2(\gamma/2)^{3/2}}{\sqrt{\gamma}} \cdot \frac{\log n}{\sqrt{n}}$$
$$\delta \geq \frac{2}{n^{3/2}}.$$

Proof. The claim follows from the proof of Lemma A.2 by choosing

$$\epsilon'' = \frac{4m}{(\gamma/2)^{3/2}} \cdot \frac{4}{\kappa} \cdot \epsilon$$

in (4). □

A.2 Proof of Theorem 5.2

Proof. Let $\epsilon, \delta > 0$ be arbitrary; we need to show that there exists $n \in \mathbb{N}$ such that T is an ϵ approximation of T^* with probability at least $1 - \delta''$.

First, for each leaf node $N \in \text{leaves}(T^*)$ of the exact tree, let $\phi(N)$ be the corresponding leaf in the learned tree T. We assume that sufficiently many samples are taken so that the estimates of the following information are correct with probability at least $1 - \frac{\delta}{4k(d+m)}$ (where $m = |\mathcal{Y}|$):

- For each internal node, the dimension $i \in [d]$ along which to branch.
- For each leaf node, the label y assigned to that leaf node.

First, we describe how to ensure that the optimal dimension $i \in [d]$ along which to branch is chosen for each of the $\frac{k}{2}$ internal nodes in T. In particular, by Assumption 2, there exists $\Delta > 0$ such that for every $i \neq i^*$ and for every t, we have

$$G(i^*, t^*) > G(i, t) + \Delta.$$

Let $G_i(t) = G(i, t)$. As long as $\|G - \hat{\tilde{G}}\|_\infty \leq \frac{\Delta}{2}$ for each $i \in [d]$, then the optimal dimension is selected. Lemma A.2 already shows that $\|G - \hat{\tilde{G}}\|_\infty$ is arbitrarily small for $n \in \mathbb{N}$ sufficiently large.

Next, we describe how to ensure that the optimal label y is selected for each leaf node. In particular, labels are selected according to (3). To simplify notation, let

$$p_y = \int \mathbb{I}[f(x) = y] \cdot \mathbb{I}[C_N] \cdot p(x)dx$$

be the unnormalized version of (3), which can equivalently be used to select the optimal label (since the normalization factor is a constant across all $y \in \mathcal{Y}$). Our goal is to choose the optimal label

$$y^* = \arg\max_{y \in \mathcal{Y}} p_y$$

for a leaf node $N = (y^*)$ in T^*. By Assumption 2, there exists a gap $\Delta > 0$ such that for every $y \neq y^*$, we have $p_{y^*} \geq p_y + \Delta$, so it suffices to show that $|p_y - \hat{\tilde{p}}_y| \leq \frac{\Delta}{2}$, where $\hat{\tilde{p}}_y$ is our estimate of p_y, as before, with two kinds of error. In particular, for each $y \in \mathcal{Y}$, we can break down the error of our estimate $\hat{\tilde{p}}_y$ into

$$|p_y - \hat{\tilde{p}}_y| \leq |p_y - \tilde{p}_y| + |\tilde{p}_y - \hat{\tilde{p}}_y|,$$

where

$$\tilde{p} = \int \mathbb{I}[f(x) = y] \cdot \mathbb{I}[C_{\phi(N)}] \cdot p(x)dx.$$

We assume that N satisfies $\text{Pr}_{x \sim \mathcal{P}}[C_N] \geq \gamma$, so we have $\|p_N - \tilde{p}_N\|_1 \leq \epsilon'$. Then, we have

$$|p_y - \tilde{p}_y| = \left| \int \mathbb{I}[f(x) = y] \cdot (\mathbb{I}[C_N] - \mathbb{I}[C_{\phi(N)}]) \cdot p(x)dx \right| \leq \|p_N - p_{\phi(N)}\|_1 \leq \epsilon'.$$

For the estimation error, by Hoeffding's inequality, we have

$$\text{Pr}_{x^{(1)},\ldots,x^{(n)} \sim \mathcal{P}} \left[|\hat{\tilde{p}}_y - \tilde{p}_y| \geq \frac{\Delta}{4} \right] \leq 2 \exp\left(-\frac{2n\Delta^2}{16} \right).$$

Taking $\epsilon' \leq \frac{\Delta}{4}$, the total error $|p_y - \hat{\tilde{p}}_y| \leq \frac{\Delta}{2}$. Taking a union bound over $y \in \mathcal{Y}$, this inequality holds with probability $2m \exp\left(-\frac{2n\Delta^2}{16}\right)$, where $m = |\mathcal{Y}|$. Since this probability is exponentially decreasing in n, we can easily bound it by $\frac{\delta}{2}$.

Next, let N be a node in the exact tree T^*, and let $\phi(N)$ be its corresponding node in the learned tree T. Let $\mathcal{P} \mid C_N$ be the distribution over points that flow to N in T^*, let $\mathcal{P} \mid C_{\phi(N)}$ be the distribution over points that flow to $\phi(N)$ in T, and let

$$\begin{aligned} p_N(x) &= p(x) \cdot \mathbb{I}[C_N] \\ p_{\phi(N)}(x) &= p(x) \cdot \mathbb{I}[C_{\phi(N)}] \end{aligned}$$

be their respective unnormalized probability density functions.

We prove by induction that for each node $N \in \text{leaves}(T^*)$, one of the following holds:

- The premise of Lemma A.2 holds, i.e., for any $\epsilon', \delta' > 0$, there exists $n \in \mathbb{N}$ such that $\|p_N - p_{\phi(N)}\|_1 \le \epsilon'$ with probability at least $1 - \delta'$.
- A negligible amount of probability mass flows to N, i.e., $\Pr[C_N] \le \gamma = \frac{\epsilon}{k}$.

For the root node N_{T^*} of T^*, the hypothesis of Lemma A.2 holds for $\mathcal{P} = \tilde{\mathcal{P}}$, since then $\|p - \tilde{p}\|_1 = 0$ with probability 1.

For any node in N', suppose that the inductive hypothesis holds for its parent N. If a negligible amount of probability mass flows to N, then the same is true of N'. Otherwise, the premise of Lemma A.2 holds for N. Consider the gain (2) used to construct the branch $C_N = (x_{i^*} \le t^*$ for N. First, it is easy to check that the maximum t^* of the function $G(t)$ in Lemma A.2 equals the maximizer of the gain—the only difference between $G(t)$ is the square root (which is monotone) and a normalization constant $\int p(x)dx$ (since the probability density function $p(x)$ in Lemma A.2 may be unnormalized).

Therefore, either the conclusion of Lemma A.2 holds, in which case we have shown the inductive hypothesis, or $\|h_s(t)\|_\infty \le \gamma$ for some $s \in \{\pm 1\}$. But

$$\gamma \ge \|h_s(t)\|_\infty = \int \mathbb{I}[C_N] \cdot p(x)dx = \Pr[C_N],$$

so again, the inductive hypothesis holds.

Now, note that

$$\begin{aligned} |\mathbb{I}[C_N] - \mathbb{I}[C_{\phi(N)}]| &= (\mathbb{I}[C_N] - \mathbb{I}[C_{\phi(N)}])^2 \\ &= \mathbb{I}[C_N] + \mathbb{I}[C_{\phi(N)}] - 2 \cdot \mathbb{I}[C_N] \cdot \mathbb{I}[C_{\phi(N)}] \\ &= (1 - \mathbb{I}[C_{\phi(N)}]) \cdot \mathbb{I}[C_N] + \mathbb{I}[C_{\phi(N)}] - \mathbb{I}[C_N] \cdot \mathbb{I}[C_{\phi(N)}] \\ &\ge (1 - \mathbb{I}[C_{\phi(N)}]) \cdot \mathbb{I}[C_N]. \end{aligned}$$

Let L be the set of leaves N in T^* for which $\Pr_{x \sim \mathcal{P}}[C_N] \ge \gamma = \frac{\epsilon}{k}$, i.e., at least γ fraction of the probability mass flows to N, and let L' be the remaining leaves. Then, the total error of the decision tree is

$$\begin{aligned} \Pr_{x \sim \mathcal{P}}[T(x) \ne T^*(x)] &= \sum_{N \in \text{leaves}(T^*)} \Pr_{x \sim \mathcal{P}}[T(x) \ne T^*(x) \mid C_N] \cdot \Pr_{x \sim \mathcal{P}}[C_N] \\ &\le \sum_{N \in L} \Pr_{x \sim \mathcal{P}}[\neg C_{\phi(N)} \mid C_N] \cdot \Pr_{x \sim \mathcal{P}}[C_N] + \sum_{N \in L'} \Pr_{x \sim \mathcal{P}}[C_N] \\ &\le \sum_{N \in L} \Pr_{x \sim \mathcal{P}}[\neg C_{\phi(N)} \mid C_N] \cdot \Pr_{x \sim \mathcal{P}}[C_N] + |L'| \cdot \gamma \\ &= \sum_{N \in L} \int (1 - \mathbb{I}[C_{\phi(N)}]) \cdot \mathbb{I}[C_N] \cdot p(x)dx + |L'| \cdot \gamma \\ &\le \sum_{N \in L} \int |\mathbb{I}[C_N] - \mathbb{I}[C_{\phi(N)}]| \cdot p(x)dx + |L'| \cdot \gamma \\ &\le \sum_{N \in L} \int |\mathbb{I}[C_N] - \mathbb{I}[C_{\phi(N)}]| \cdot p(x)dx + |L'| \cdot \gamma \\ &= \sum_{N \in L} \|p_N - \tilde{p}_N\|_1 + |L'| \cdot \gamma \\ &\le |L| \cdot \epsilon' + |L'| \cdot \gamma \\ &\le \epsilon, \end{aligned}$$

where in the last step, we choose $\epsilon' = \frac{\epsilon}{k}$. This inequality holds with probability at least $1 - \frac{\delta}{2} - k \cdot \delta'$, by taking a union bound, including over the probability $1 - \frac{\delta}{2}$ that the label choices described above hold. Therefore, the result follows taking $\epsilon' = \frac{\epsilon}{2k}$ and $\delta' = \frac{\delta}{4k}$. □

B PROOFS OF TECHNICAL LEMMAS

In this section, we prove the technical lemmas required for our proofs of Lemma B.1 and Theorem 5.2.

B.1 Proof of Existence of a Gap

We prove that a gap exists for reasonably behaved functions.

Lemma B.1. Let $g : \mathbb{R} \to [0, 1]$ be a continuous function with bounded support, and assume that its global maximizer

$$t^* = \arg\max_{t \in \mathbb{R}} g(t)$$

is unique. Then, for any $\epsilon' > 0$, there exists $\delta' > 0$ such that g is (ϵ', δ') gapped.

Proof. Let $t_{\max}$ be a bound on the support of g, i.e., $g(t) = 0$ if $|t| > t_{\max}$. Let $\epsilon' > 0$ be arbitrary, and let

$$A_{\epsilon'} = \{t \in \mathbb{R} \mid |t| \leq t_{\max} \text{ and } |F(t) - F(t^*)| \geq \epsilon'\}.$$

Note that $A_{\epsilon'}$ is a compact set, so g achieves its maximum on $A_{\epsilon'}$, i.e.,

$$t^*_{\epsilon'} = \arg\max_{t \in A_{\epsilon'}} g(t).$$

Then, the result follows for

$$\delta' = \frac{g(t^*) - g(t^*_{\epsilon'})}{2} > 0.$$

Note that we divide by 2 since the inequality in Definition A.1 is strict. □

B.2 Bound on Intrinsic Error

The following lemma enables us to bound the intrinsic error $\|g - \tilde{g}\|_\infty$ given that the distributions have probability density functions bounded in L_1 norm.

Lemma B.2. Let $\mathcal{X} \subseteq \mathbb{R}^d$, and let $\mathcal{P}$ and $\tilde{\mathcal{P}}$ be distributions over $\mathcal{X}$, and let $p(x)$ and $\tilde{p}(x)$ be unnormalized probability density functions for $\mathcal{P}$ and $\tilde{\mathcal{P}}$, respectively, that satisfy $\|p - \tilde{p}\|_1 \leq \epsilon$. Let $\beta : \mathcal{X} \times \mathbb{R} \to [0, 1]$ be any function, and let

$$g(t) = \int \beta(x, t) \cdot p(x) dx$$
$$\tilde{g}(t) = \int \beta(x, t) \cdot \tilde{p}(x) dx.$$

Then, we have

$$\|g - \tilde{g}\|_\infty \leq \epsilon.$$

Proof. Note that

$$\begin{aligned}\|g - \tilde{g}\|_\infty = \sup_{t \in \mathbb{R}} |g - \tilde{g}| &\leq \sup_{t \in \mathbb{R}} \int \beta(x, t) |p(x) - \tilde{p}(x)| dx \\ &\leq \sup_{t \in \mathbb{R}} \int |p(x) - \tilde{p}(x)| dx \\ &\leq \epsilon,\end{aligned}$$

as claimed. □

B.3 Bound on Estimation Error

The following lemma enables us to bound the estimation error $\|\tilde{g} - \hat{\tilde{g}}\|_\infty$.

Lemma B.3. Let $\mathcal{X} \subseteq \mathbb{R}^d$, let $\mathcal{P}$ be a distribution over $\mathcal{X}$, and let $p(x)$ be an unnormalized probability density function for $\mathcal{P}$. Let $\alpha : \mathcal{X} \to [0, 1]$ be an arbitrary function, let $i \in [d]$, and let

$$g(t) = \int \alpha(x) \cdot \mathbb{I}[x_i \leq t] \cdot p(x) dx.$$

Let $x^{(1)}, ..., x^{(n)}$ be i.i.d. samples from $\mathcal{P}$, and let

$$\hat{g}(t) = \frac{Z}{n} \sum_{i=1}^{n} \alpha(x^{(k)}) \cdot \mathbb{I}[x_i^{(k)} \leq t]$$

be the empirical estimate of g on these points, where

$$Z = \int p(x) dx$$

is a normalization constant. Then, we have

$$\mathrm{Pr}_{x^{(1)},\ldots,x^{(n)}\sim\mathcal{P}}\left[\|g-\hat{g}\|_{\infty}\geq\frac{4\log n}{\sqrt{n}}\right]\leq\frac{2}{n^{3/2}},$$

for any $n \geq 3$.

Proof. First, we define points $t_0, t_1, \ldots, t_{\sqrt{n}} \in \mathbb{R}$ that divide $\mathbb{R}$ into $\sqrt{n}$ intervals according to the marginal probability density function $p_i(t)$ along dimension i (for convenience, we assume n is a perfect square), which has corresponding cumulative distribution function $F_i(t)$. Then, we choose t_j to satisfy

$$t_j \in F_i^{-1}\left(\frac{j\cdot Z}{\sqrt{n}}\right).$$

For convenience, we choose $t_0 = -\infty$ and $t_{\sqrt{n}} = \infty$, which satisfy the condition. Now, for each $j \in [\sqrt{n}]$, let $I_j = (t_{j-1}, t_j]$. Note that these intervals cover $\mathbb{R}$, i.e., $\mathbb{R} = I_1 \cup \ldots \cup I_{\sqrt{n}}$.

Then, we can decompose the quantity $\|g - \hat{g}\|_{\infty}$ into three parts:

$$\begin{aligned}
\|g-\hat{g}\|_{\infty} &= \sup_{t\in\mathbb{R}} |g(t)-\hat{g}(t)| \\
&= \sup_{j\in[\sqrt{n}]}\sup_{t\in I_j} |g(t)-\hat{g}(t)| \\
&\leq \sup_{j\in[\sqrt{n}]}\sup_{t\in I_j} \left\{|g(t)-g(t_j)| + |g(t_j)-\hat{g}(t_j)| + |\hat{g}(t_j)-\hat{g}(t)|\right\} \\
&\leq \sup_{j\in[\sqrt{n}]}\sup_{t\in I_j} |g(t)-g(t_j)| + \sup_{j\in[\sqrt{n}]} |g(t_j)-\hat{g}(t_j)| + \sup_{j\in[\sqrt{n}]}\sup_{t\in I_j} |\hat{g}(t_j)-\hat{g}(t)|.
\end{aligned}$$

We show that each of these three parts can be made arbitrarily small with high probability by taking n sufficiently large.

First, consider the term $\sup_{j\in[\sqrt{n}]}\sup_{t\in I_j}|g(t)-g(t_j)|$. Since $t \leq t_j$, we have $\mathbb{I}[x_i \leq t] \leq \mathbb{I}[x_i \leq t_j]$. Thus, for all $t \in I_j$, we have

$$\begin{aligned}
|g(t)-g(t_j)| &= \left|\int \alpha(x)\cdot(\mathbb{I}[x_i\leq t]-\mathbb{I}[x_i\leq t_j])\cdot p(x)dx\right| \\
&\leq \int(\mathbb{I}[x_i\leq t_j]-\mathbb{I}[x_i\leq t])\cdot p(x)dx \\
&= F_i(t_j)-F_i(t) \\
&\leq n^{-1/2},
\end{aligned}$$

where the last inequality follows from the definition of t_j and the fact that $t \in I_j$.

Second, consider the term $\sup_{j\in[\sqrt{n}]}|g(t_j)-\hat{g}(t_j)|$. Note that each $Z\cdot\alpha(x^{(k)})\cdot\mathbb{I}[x_i^{(k)}\geq t]$ is a random variable in $[0, 1]$. Therefore, by the Hoeffding inequality, we have

$$\mathrm{Pr}_{x^{(1)},\ldots,x^{(n)}\sim\mathcal{P}}\left[|g(t_j)-\hat{g}(t_j)|\geq\frac{\log n}{n}\right]\leq e^{-2(\log n)^2}\leq\frac{1}{n^2}$$

for $n \geq 3$. By a union bound, this inequality holds for every $j \in [\sqrt{n}]$ with probability $n^{-3/2}$.

Third, consider the term $\sup_{j\in[\sqrt{n}]}\sup_{t\in I_j}|\hat{g}(t_j)-\hat{g}(t)|$. We first show that for every $j \in [\sqrt{n}]$, the interval I_j contains at most $n^{1/2}\log n$ of the points $x^{(1)}, \ldots, x^{(n)}$ with high probability. By the definition of the points t_j, the probability that a single randomly selected point $x^{(k)}$ falls in I_j is $n^{-1/2}$ (since the points t_j were constructed according to the cumulative distribution function F_i):

$$M = \mathbb{E}_{x\sim\mathcal{P}}\left[\mathbb{I}[x\in I_j]\right] = \mathrm{Pr}_{x\sim\mathcal{P}}[x\in I_j] = \frac{1}{\sqrt{n}}.$$

Then, the fraction of the n points $x^{(k)}$ that fall in the interval I_j is

$$\hat{M} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}[x^{(k)}\in I_j].$$

Note that each $\mathbb{I}[x^{(k)} \in I_j]$ is an random variable in $[0, 1]$, so by Hoeffding's inequality, we have

$$\mathrm{Pr}_{x^{(1)},\ldots,x^{(n)}\sim\mathcal{P}}\left[|\hat{M}-M|\geq\frac{\log n}{\sqrt{n}}\right]\leq e^{-2(\log n)^2}\leq\frac{1}{n^2}.$$

Now, note that each point $x^{(k)}$ in I_j can increase the value of $|\hat{g}(t_j) - \hat{g}(t)|$ by at most n^{-1}. Since there are $n \cdot \hat{M}$ points $x^{(k)}$ in I_j, the total increase is bounded by $\hat{M}$, i.e.,

$$\Pr_{x^{(1)},\ldots,x^{(n)}\sim\mathcal{P}}\left[\sup_{t\in I_j} |\hat{g}(t_j) - \hat{g}(t)| \geq \frac{2\log n}{\sqrt{n}}\right] \leq \frac{1}{n^2}.$$

As before, by a union bound, this inequality holds for every $j \in [\sqrt{n}]$ with probability $n^{-3/2}$.

Putting these three results together, we can conclude that for $n \geq 3$, we have

$$\Pr_{x^{(1)},\ldots,x^{(n)}\sim\mathcal{P}}\left[\|g - \hat{g}\|_\infty \geq \frac{4\log n}{\sqrt{n}}\right] \leq \frac{2}{n^{3/2}},$$

as claimed. □

B.4 Bound on Error of Maximizers

The following lemma enables us to bound the maximizer of two functions that are close in L_∞ norm.

Lemma B.4. Let $\mathcal{P}$ be a probability distribution over $\mathbb{R}$, and let $F(t)$ be an cumulative distribution function for $\mathcal{P}$. Suppose that $g : \mathbb{R} \to \mathbb{R}$ is (ϵ, δ) gapped according to $\mathcal{P}$, and $h : \mathbb{R} \to \mathbb{R}$ satisfies $\|g - h\|_\infty \leq \frac{\delta}{2}$. Let

$$t_g^* = \arg\max_{t\in\mathbb{R}} g(t)$$
$$t_h^* = \arg\max_{t\in\mathbb{R}} h(t).$$

Then, we have $|F(t_g^*) - F(t_h^*)| \leq \epsilon$.

Proof. By definition of the L_∞ norm, we have

$$g(t_g^*) - h(t_g^*) \leq \frac{\delta}{2}$$
$$h(t_h^*) - g(t_h^*) \leq \frac{\delta}{2}.$$

Combining these gives

$$g(t_g^*) - g(t_h^*) \leq \delta + h(t_g)^* - h(t_h^*) \leq \delta,$$

where the second inequality follows because t_h^* is a maximizer of h. Since g is (ϵ, δ) gapped, and since t_g^* is the maximizer of g, this inequality implies the claim. □

B.5 Bound on Error of Probabiliy Density Functions

The following lemma enables us to bound the L_1 error of the estimated probability density function.

Lemma B.5. Let $\mathcal{X} \subseteq \mathbb{R}^d$, let $\mathcal{P}$ and $\tilde{\mathcal{P}}$ be distributions over $\mathcal{X}$, and let $p(x)$ and $\tilde{p}(x)$ be unnormalized probability density functions for $\mathcal{P}$ and $\tilde{\mathcal{P}}$, respectively, satisfying $\|p - \tilde{p}\| \leq \epsilon$. Let $i \in [d]$, let $\mathcal{P}_i$ be the marginal distribution of $\mathcal{P}$ along dimension i, let F_i be an unnormalized cumulative distribution function for $\mathcal{P}_i$, and let $t, \tilde{t} \in \mathbb{R}$ satisfying $|F_i(t) - F_i(\tilde{t})| \leq \epsilon'$. Let $i \in [d]$, and let

$$p'(x) = p(x) \cdot \mathbb{I}[x_i \leq t]$$
$$\tilde{p}'(x) = \tilde{p}(x) \cdot \mathbb{I}[x_i \leq \tilde{t}]$$

be the unnormalized probability density functions for $\mathcal{P} \mid (x_i \leq t)$ and $\tilde{\mathcal{P}} \mid (x_i \leq \tilde{t})$, respectively. Then, we have

$$\|p' - \tilde{p}'\|_1 \leq \epsilon + \epsilon'.$$

Proof. Assume without loss of generality that $t \le \tilde{t}$. Then, we have

$$\begin{aligned}
\|p' - \tilde{p}'\|_1 &= \int |p'(x) - \tilde{p}'(x)| dx \\
&= \int |p(x) \cdot \mathbb{I}[x_i \le t] - \tilde{p}(x) \cdot \mathbb{I}[x_i \le \tilde{t}]| dx \\
&= \int |p(x) \cdot \mathbb{I}[x_i \le t] - (p(x) + \tilde{p}(x) - p(x)) \cdot \mathbb{I}[x_i \le \tilde{t}]| dx \\
&\le \int p(x) \cdot |\mathbb{I}[x_i \le t] - \mathbb{I}[x_i \le \tilde{t}]| dx + \int |\tilde{p}(x) - p(x)| \cdot \mathbb{I}[x_i \le \tilde{t}] dx \\
&\le \int p(x) \cdot |\mathbb{I}[x_i \le t] - \mathbb{I}[x_i \le \tilde{t}]| dx + \epsilon \\
&= \int p(x) \cdot \mathbb{I}[t \le x_i \le \tilde{t}] dx + \epsilon \\
&= |F_i(\tilde{t}) - F_i(t)| + \epsilon \\
&\le \epsilon' + \epsilon,
\end{aligned}$$

as claimed. □

C NUMERICAL RESULTS

Fidelity. The following table shows the fidelity (F_1 score relative to the blackbox model) of different interpretations:

Blackbox Model	Interpretation	Fidelity
diabetes risk classifier	our decision tree (7 nodes)	0.176
diabetes risk classifier	our decision tree (15 nodes)	0.209
diabetes risk classifier	our decision tree (23 nodes)	0.278
diabetes risk classifier	our decision tree (31 nodes)	0.308
diabetes risk classifier	CART decision tree (7 nodes)	0.162
diabetes risk classifier	CART decision tree (15 nodes)	0.183
diabetes risk classifier	CART decision tree (23 nodes)	0.231
diabetes risk classifier	CART decision tree (31 nodes)	0.240
diabetes risk classifier	born-again decision tree (7 nodes)	0.164
diabetes risk classifier	born-again decision tree (15 nodes)	0.185
diabetes risk classifier	born-again decision tree (23 nodes)	0.238
diabetes risk classifier	born-again decision tree (31 nodes)	0.240
diabetes risk classifier	rule list	0.246
diabetes risk classifier	decision set	–
cart-pole control policy	our decision tree (3 nodes)	0.752
cart-pole control policy	our decision tree (7 nodes)	0.839
cart-pole control policy	our decision tree (11 nodes)	0.912
cart-pole control policy	our decision tree (15 nodes)	0.936
cart-pole control policy	CART decision tree (3 nodes)	0.749
cart-pole control policy	CART decision tree (7 nodes)	0.827
cart-pole control policy	CART decision tree (11 nodes)	0.824
cart-pole control policy	CART decision tree (15 nodes)	0.812
cart-pole control policy	born-again decision tree (3 nodes)	0.750
cart-pole control policy	born-again decision tree (7 nodes)	0.832
cart-pole control policy	born-again decision tree (11 nodes)	0.830
cart-pole control policy	born-again decision tree (15 nodes)	0.894
cart-pole control policy	rule list	0.612
cart-pole control policy	decision set	0.560

Interpretability. The following table shows the accuracy of user responses in our user study:

Blackbox Model	Interpretation	User Response Accuracy
diabetes risk classifier	our decision tree (31 nodes)	0.878
diabetes risk classifier	rule list	0.817
cart-pole control policy	our decision tree (15 nodes)	0.826
cart-pole control policy	decision set	0.819