Faithful and Customizable Explanations of Black Box Models

Himabindu Lakkaraju[1], **Ece Kamar**[2], **Rich Caruana**[2], **Jure Leskovec**[3]
[1] Harvard University [2] Microsoft Research [3] Stanford University

Abstract

As predictive models increasingly assist human experts (e.g., doctors) in day-to-day decision making, it is crucial for experts to be able to explore and understand how such models behave in different feature subspaces in order to know if and when to trust them. To this end, we propose *Model Understanding through Subspace Explanations* (MUSE), a novel model agnostic framework which facilitates understanding of a given black box model by explaining how it behaves in subspaces characterized by certain features of interest. Our framework provides end users (e.g., doctors) with the flexibility of customizing the model explanations by allowing them to input the features of interest. The construction of explanations is guided by a novel objective function that we propose to simultaneously optimize for fidelity to the original model, unambiguity and interpretability of the explanation. More specifically, our objective allows us to learn, with optimality guarantees, a small number of compact *decision sets* each of which captures the behavior of a given black box model in unambiguous, well-defined regions of the feature space. Experimental evaluation with real-world datasets and user studies demonstrate that our approach can generate customizable, highly compact, easy-to-understand, yet accurate explanations of various kinds of predictive models compared to state-of-the-art baselines.

Introduction

The successful adoption of predictive models for real world decision making hinges on how much decision makers (e.g., doctors, judges) can understand and trust their functionality. Only if decision makers have a clear understanding of the behavior of predictive models, they can evaluate when and how much to depend on these models, detect potential biases in them, and develop strategies for further model refinement. However, the increasing complexity and the proprietary nature of predictive models employed today is making this problem harder (Ribeiro, Singh, and Guestrin 2016), thus, emphasizing the need for tools which can explain these complex black boxes in a faithful and interpretable manner.

Prior research on explaining black box models can be categorized as: 1) *Local* explanations, which focus on explaining individual predictions of a given black box classifier (Ribeiro, Singh, and Guestrin 2016; 2018; Koh and Liang 2017) and 2) *Global explanations*, which focus on explaining model behavior as a whole, often by summarizing complex models using simpler, more interpretable approximations such as decision sets or lists (Lakkaraju, Bach, and Leskovec 2016; Letham et al. 2015). In this paper, we focus on a new form of explanation that is designed to help end users (e.g., decision makers such as judges, doctors) gain deeper understanding of model behavior: *a differential explanation* that describes how the model logic varies across different subspaces of interest in a faithful and interpretable fashion. To illustrate, let us consider a scenario where a doctor is trying to understand a model which predicts if a given patient has depression or not. The doctor might be keen on understanding how the model makes predictions for different patient subgroups (See Figure 1 left). Furthermore, she might be interested in asking questions such as "how does the model make predictions on patient subgroups associated with different values of exercise and smoking?" and might like to see explanations customized to her interest (See Figure 1 right). The problem of constructing such explanations has not been studied by previous research aimed at understanding black box models.

Here, we propose a novel framework, *Model Understanding through Subspace Explanations* (MUSE), which constructs *global* explanations of black box classifiers which highlight their behavior in subspaces characterized by features of user interest. To the best of our knowledge, this is the first work to study the notion of incorporating user input when generating explanations of black box classifiers while successfully trading off notions of fidelity, unambiguity and interpretability. Our framework takes as input a dataset of instances with semantically meaningful or interpretable features (*e.g.* age, gender), and the corresponding class labels assigned by the black box model. It also accepts as an optional input a set of features that are of interest to the end user in order to generate explanations tailored to user preferences. Our framework then maps these inputs to a customized, faithful, and interpretable explanation which succinctly summarizes the behavior of the given model. We employ a two-level decision set representation, where the if-then clauses at the outer level describe the subspaces, and the inner if-then clauses explain the decision logic employed by the black box model within the corresponding subspace (See Figure 1 left). The two-level struc-

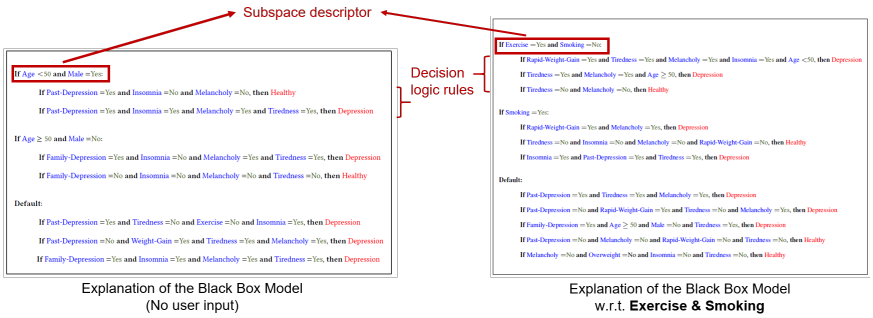


Figure 1: Explanations generated by our framework MUSE to describe the behavior of a 3-level neural network trained on depression dataset. MUSE generates explanation of the model without user input (left). It automatically selects features for defining subspaces by optimizing for fidelity, unambiguity, and interpretability. MUSE generates customized explanations based on the features of interest input by the end user - exercise and smoking (right).

ture which decouples the descriptions of subspaces from the decision logic of the model naturally allows for incorporating user input when generating explanations. In order to construct an explanation based on the above representation, we formulate a novel objective function which can jointly reason about various relevant considerations: fidelity to the original model i.e., mimicking the original model in terms of assigning class labels to instances, unambiguity in describing the model logic used to assign labels to instances, and interpretability by favoring lower complexity i.e., fewer rules and predicates etc. While exactly optimizing our objective is an NP-hard problem, we prove that our optimization problem is a non-normal, non-monotone submodular function with matroid constraints which allows for provably near optimal solutions.

We evaluated the fidelity and interpretability of the explanations generated by our approach on three real world datasets: judicial bail decisions, high school graduation outcomes, and depression diagnosis. Experimental results indicate that our approach can generate much less complex and high fidelity explanations of various kinds of black box models compared to state-of-the-art baselines. We also carried out user studies in which we asked human subjects to reason about a black box model's behavior using the explanations generated by our approach and other state-of-the-art baselines. Results of this study demonstrate that our approach allows humans to accurately and quickly reason about the behavior of complex predictive models.

Related Work

Explaining Model Behavior: One approach for interpretability is learning predictive models which are human understandable (e.g., decision trees (Rokach and Maimon 2005), decision lists (Letham et al. 2015), decision sets (Lakkaraju, Bach, and Leskovec 2016), linear models, generalized additive models (Lou, Caruana, and Gehrke 2012)). Recent research focused on explaining individual predictions of black box classifiers (Ribeiro, Singh, and Guestrin 2016; Koh and Liang 2017). Ribeiro et. al.,'s approach of approximating global behavior of black box models through a collection of locally linear models create ambiguity as it does not clearly specify which local model applies to what part of the feature space. Global explanations can also be generated by approximating the predictions of black box models with interpretable models such as decision sets, decision trees. However, the resulting explanations are not suitable to answer deeper questions about model behavior (e.g., 'how the model logic differs across patient subgroups associated with various values of exercise and smoking?'). Furthermore, existing frameworks do not jointly optimize for fidelity, unambiguity, and interpretability.

Visualizing and Understanding Specific Models: The problem of visualizing how certain classes of models such as deep neural networks are making predictions has attracted a lot of attention in the recent past (Yosinski et al. 2015; Zintgraf et al. 2017). Zintgraf et. al. (Zintgraf et al. 2017) focused on visualizing how a deep neural network responds to a given input. Shrikumar et. al. (Shrikumar et al. 2016) proposed an approach to determine the important features of deep neural networks. Furthermore, there exist tools and frameworks to visualize the functionality of different classes of models such as decision trees (Teoh and Ma 2003), SVMs etc. (Jakulin et al. 2005). However, unlike our framework, these approaches are tailored to a particular class of models and do not generalize to any black box model.

Our Framework

Here, we describe our framework, Model Understanding through Subspace Explanations (MUSE), which is designed to address the problem of explaining black box models while highlighting their behavior w.r.t. specific subspaces of interest. As part of this discussion, we examine how to: (1) design a representation which enables us to not only construct faithful, unambiguous, and interpretable explanations but also readily incorporate user input for customization, (2) quantify the notions of fidelity, unambiguity, and interpretability in the context of the representation we choose, (3) formulate an optimization problem which effectively tradeoffs fidelity, unambiguity, and interpretability, (4) solve the optimization problem efficiently, and (5) customize explanations based on user preferences (See 3 for a sketch of MUSE workflow).

Our Representation: Two Level Decision Sets

The most important criterion for choosing a representation is that it should be understandable to decision makers who are not experts in machine learning, readily approximate complex black box models, and allow us to incorporate human input when generating explanations. We choose two level decision sets as our representation. The basic building block of this structure is a decision set, which is a set of if-then rules that are unordered. The two level decision set can be regarded as a set of multiple decision sets, each of which is embedded within an outer if-then structure, such that the inner if-then rules represent the decision logic employed by the black box model while labeling instances within the subspace characterized by the conditions in the outer if-then clauses. Consequently, we refer to the conditions in the outer if-then rules as *subspace descriptors* and the inner if-then rules as *decision logic rules* (See Figure 1). This two level nested if-then structure allows us to clearly specify how the model behaves in which part of the feature space. Furthermore, the decoupling of the subspace descriptors and the decision logic rules allows us to readily incorporate user input and describe subspaces that are of interest to the user in a compact fashion.

While the expressive power of two level decision sets is the same as that of other rule based models (e.g., decision sets/lists/trees), the nesting of if-then clauses in a two level decision set representation enables the optimization algorithm (more details later in this Section) to select *subspace descriptors* and *decision logic rules* such that higher fidelity to the original model can be obtained with minimal complexity thus resulting in more compact approximations compared to conventional decision sets (details in experiments section). In addition, two level decision set representation does not have the pitfalls associated with decision lists where understanding a particular rule requires reasoning about all the previously encountered rules because of the if-else-if construct (Lakkaraju, Bach, and Leskovec 2016).

Definition 1. A **two level decision set** $\mathcal{R}$ is a set of rules $\{(q_1, s_1, c_1)(q_2, s_2, c_2) \cdots (q_M, s_M, c_M)\}$ where q_i and s_i are conjunctions of $predicates$ of the form $(feature, operator, value)$ (eg., $age \geq 50$) and c_i is a class label. q_i corresponds to the subspace descriptor and (s_i, c_i) together represent the inner if-then rules (decision logic rules) with s_i denoting the condition and c_i denoting the class label (See Figure 1). A two level decision set assigns a label to an instance $\boldsymbol{x}$ as follows: if $\boldsymbol{x}$ satisfies exactly one of the rules i i.e., $\boldsymbol{x}$ satisfies $q_i \wedge s_i$, then its label is the corresponding class label c_i. If $\boldsymbol{x}$ satisfies none of the rules in $\mathcal{R}$, then its label is assigned using a default function and if $\boldsymbol{x}$ satisfies more than one rule in $\mathcal{R}$ then its label is assigned using a tie-breaking function.

In our experiments, we employ a default function which computes the majority class label (assigned by the black box model) of all the instances in the training data which do not satisfy any rule in $\mathcal{R}$ and assigns them to this majority label. For each instance which is assigned to more than one rule in $\mathcal{R}$, we break ties by choosing the rule which has a higher agreement rate with the black box model. Other forms of default and tie-breaking functions can be easily incorporated into our framework.

Quantifying Fidelity, Unambiguity, and Interpretability

To meaningfully describe the behavior of a given black box model, it is important to construct an explanation that is not only faithful to the original model but also unambiguous and interpretable. Below we explore each of these desiderata in detail and discuss how to quantify them w.r.t a two level decision set explanation $\mathcal{R}$ with M rules (See Definition 1), a black box model $\mathcal{B}$, and a dataset $\mathcal{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2 \cdots \boldsymbol{x}_N\}$ where $\boldsymbol{x}_i$ captures the feature values of instance i. We treat the black box model $\mathcal{B}$ as a function which takes an instance $\boldsymbol{x} \in \mathcal{D}$ as input and returns a class label.

Fidelity: A high fidelity explanation should faithfully mimic the behavior of the black box model. While different notions of fidelity can be defined, our metric of choice quantifies the disagreement between the labels assigned by the explanation and the labels assigned by the black box model. We define **disagreement($\mathcal{R}$)** as the number of instances for which the label assigned by the black box model $\mathcal{B}$ does not match the label c assigned by the explanation $\mathcal{R}$ (Table 1).

Unambiguity: An unambiguous explanation should provide unique deterministic rationales for describing how the black box model behaves in various parts of the feature space. To quantify this notion, we introduce two metrics: 1) **ruleoverlap($\mathcal{R}$)** which captures the number of additional rationales (beyond 1) provided by the explanation $\mathcal{R}$ for each instance in the data. Higher the value of this metric, higher the ambiguity of the explanation (Table 1). 2) **cover($\mathcal{R}$)** which captures the number of instances in the data that satisfy some rule in $\mathcal{R}$. Our goal here would be to minimize **ruleoverlap($\mathcal{R}$)** and maximize **cover($\mathcal{R}$)**. These two notions together ensure that the explanation we generate describes as much of the feature space as unambiguously as possible (Table 1).

Interpretability: Interpretability metric quantifies how easy it is to understand and reason about the explanation. While we choose an interpretable representation (e.g., two level decision sets), how interpretable the explanation is depends on its complexity (For example, a decisions set with many rules and high depth would not be interpretable for a user).

We quantify the interpretability of explanation $\mathcal{R}$ using the following metrics (Table 1): **size($\mathcal{R}$)** is the number of rules (triples of the form (q, s, c)) in the two level decision set $\mathcal{R}$. **maxwidth($\mathcal{R}$)** is the maximum width computed over all the elements in $\mathcal{R}$, where each element is either a condition of some decision logic rule s or a subspace descriptor q, and $width(s)$ is the number of predicates in the condition s. Similarly, $width(q)$ is defined as the total number of predicates of the subspace descriptor q. **numpreds($\mathcal{R}$)** counts the number of predicates in $\mathcal{R}$ including those appearing in both the decision logic rules and subspace descriptors. Note that the predicates of subspace descriptors are counted multiple times as a subspace descriptor q could potentially appear alongside multiple decision logic rules. **numdsets($\mathcal{R}$)**

Fidelity	$disagreement(\mathcal{R}) = \sum_{i=1}^{M} \lvert\{\boldsymbol{x} \mid \boldsymbol{x} \in \mathcal{D}, \boldsymbol{x} \text{ satisfies } q_i \wedge s_i, \mathcal{B}(\boldsymbol{x}) \neq c_i\}\rvert$
Unambiguity	$ruleoverlap(\mathcal{R}) = \sum_{i=1}^{M} \sum_{j=1, i\neq j}^{M} overlap(q_i \wedge s_i, q_j \wedge s_j)$ $cover(\mathcal{R}) = \lvert\{x \mid x \in \mathcal{D}, x \text{ satisfies } q_i \wedge s_i \text{ where } i \in \{1 \cdots M\}\}\rvert$
Interpretability	$size(\mathcal{R})$: number of rules (triples of the form (q, s, c)) in $\mathcal{R}$ $maxwidth(\mathcal{R}) = \max_{e \in \bigcup_{i=1}^{M} (q_i \cup s_i)} width(e)$ $numpreds(\mathcal{R}) = \sum_{i=1}^{M} width(s_i) + width(q_i)$ $numdsets(\mathcal{R}) = \lvert dset(\mathcal{R})\rvert$ where $dset(\mathcal{R}) = \bigcup_{i=1}^{M} q_i$ $featureoverlap(\mathcal{R}) = \sum_{q \in dset(\mathcal{R})} \sum_{i=1}^{M} featureoverlap(q, s_i)$

Table 1: Metrics used in the Optimization Problem

Algorithm 1 Optimization Procedure (Lee et al. 2009)

1: **Input:** Objective f, domain $\mathcal{ND} \times \mathcal{DL} \times \mathcal{C}$, parameter δ, number of constraints k
2: $V_1 = \mathcal{ND} \times \mathcal{DL} \times \mathcal{C}$
3: **for** $i \in \{1, 2 \cdots k+1\}$ **do** ▷ Approximation local search procedure
4: $X = V_i; n = |X|; S_i = \emptyset$
5: Let v be the element with the maximum value for f and set $S_i = v$
6: **while** there exists a delete/update operation which increases the value of S_i by a factor of at least $(1 + \frac{\delta}{n^4})$ **do**
7: **Delete Operation:** If $e \in S_i$ such that $f(S_i \backslash \{e\}) \geq (1 + \frac{\delta}{n^4}) f(S_i)$, then $S_i = S_i \backslash e$
8:
9: **Exchange Operation** If $d \in X \backslash S_i$ and $e_j \in S_i$ (for $1 \leq j \leq k$) such that
10: $(S_i \backslash e_j) \cup \{d\}$ (for $1 \leq j \leq k$) satisfies all the k constraints and
11: $f(S_i \backslash \{e_1, e_2 \cdots e_k\} \cup \{d\}) \geq (1 + \frac{\delta}{n^4}) f(S_i)$, then $S_i = S_i \backslash \{e_1, e_2, \cdots e_k\} \cup \{d\}$
12: **end while**
13: $V_{i+1} = V_i \backslash S_i$
14: **end for**
15: **return** the solution corresponding to $\max\{f(S_1), f(S_2), \cdots f(S_{k+1})\}$

counts the number of unique subspace descriptors (outer if-then clauses) in $\mathcal{R}$.

In a two-level decision set, subspace descriptors and decision logic rules have different semantic meanings i.e., each subspace descriptor characterizes a specific region of the feature space, and the corresponding inner if-then rules specify the decision logic of the black box model within that region. To make the distinction more clear, we minimize the overlap between the features that appear in subspace descriptors and those that appear in decision logic rules. To quantify this, we compute for each pair of subspace descriptor q and decision logic rule s, the number of features that occur in both q and s ($featureoverlap(q, s)$) and then sum up these counts. The resulting sum is denoted as **featureoverlap**($\mathcal{R}$).

Objective Function

We formulate an objective function that can jointly optimize for fidelity to the original model, unambiguity and interpretability of the explanation. We assume that we are given as inputs a dataset $\mathcal{D}$, labels assigned to instances in $\mathcal{D}$ by black box model $\mathcal{B}$, a set of possible class labels $\mathcal{C}$, a candidate set of conjunctions of predicates (Eg., Age $\geq$ 50 and Gender = Female) $\mathcal{ND}$ from which we can pick the subspace descriptors, and another candidate set of conjunctions of predicates $\mathcal{DL}$ from which we can choose the decision logic rules. In practice, a frequent itemset mining algorithm such as apriori (Agrawal, Srikant, and others) can be used to generate the candidate sets of conjunctions of predicates. If the user does not provide any input, both $\mathcal{ND}$ and $\mathcal{DL}$ are assigned to the same candidate set generated by apriori.

To facilitate theoretical analysis, the metrics defined in Table 1 are expressed in the objective function either as non-negative reward functions or constraints. To construct non-negative reward functions, penalty terms (metrics in Table 1) are subtracted from their corresponding upper bound values ($\mathcal{P}_{max}$, $\mathcal{O}_{max}$, $\mathcal{O'}_{max}$, F_{max}) which are computed with respect to the sets $\mathcal{ND}$ and $\mathcal{DL}$.

$f_1(\mathcal{R}) = \mathcal{P}_{max} - numpreds(\mathcal{R})$, where $\mathcal{P}_{max} = 2 * \mathcal{W}_{max} * |\mathcal{ND}| * |\mathcal{DL}|$

$f_2(\mathcal{R}) = \mathcal{O}_{max} - featureoverlap(\mathcal{R})$, where $\mathcal{O}_{max} = \mathcal{W}_{max} * |\mathcal{ND}| * |\mathcal{DL}|$

$f_3(\mathcal{R}) = \mathcal{O'}_{max} - ruleoverlap(\mathcal{R})$, where $\mathcal{O'}_{max} = N \times (|\mathcal{ND}| * |\mathcal{DL}|)^2$

$f_4(\mathcal{R}) = cover(\mathcal{R})$

$f_5(\mathcal{R}) = \mathcal{F}_{max} - disagreement(\mathcal{R})$, where $\mathcal{F}_{max} = N \times |\mathcal{ND}| * |\mathcal{DL}|$

where W_{max} is the maximum width of any rule in either candidate sets. The resulting optimization problem is:

$$\mathcal{R} \subseteq \mathcal{ND} \times \mathcal{DL} \times \mathcal{C} \sum_{i=1}^{5} \lambda_i f_i(\mathcal{R}) \quad (1)$$

s.t. $size(\mathcal{R}) \leq \epsilon_1, maxwidth(\mathcal{R}) \leq \epsilon_2, numdsets(\mathcal{R}) \leq \epsilon_3$

$\lambda_1 \cdots \lambda_5$ are non-negative weights which manage the relative influence of the terms in the objective. These can be specified by an end user or can be set using cross validation (details in experiments section). The values of $\epsilon_1, \epsilon_2, \epsilon_3$ are application dependent and need to be set by an end user.

Theorem 1. *The objective function in Eqn. 1 is non-normal, non-negative, non-monotone, submodular and the constraints of the optimization problem are matroids.*

Proof. See Appendix. □

Optimization Procedure

While exactly solving the optimization problem in Eqn. 1 is NP-Hard (Khuller, Moss, and Naor 1999), the specific properties of the problem: non-monotonicity, submodularity, non-normality, non-negativity and the accompanying matroid constraints allow for applying algorithms with provable optimality guarantees. We employ an optimization procedure based on approximate local search (Lee et al. 2009) which provides the best known theoretical guarantees for this class of problems. More specifically, the procedure we employ provides an optimality guarantee of $\frac{1}{k+2+1/k+\delta}$ where k is the number of constraints and $\delta > 0$. In the case of our problem with 3 constraints, this factor boils down to $\sim 1/5$ approximation.

The pseudocode for the optimization procedure is in Algorithm 1. The solution set is initially empty (line 4) and then an element v with the maximum value for the objective function is added (line 5). This is followed by a sequence of delete and/or exchange operations (lines 6 – 12) until there is no other element remaining to be deleted or exchanged from the solution set. This entire process is repeated $k + 1$ times (line 13) and the solution set with the maximum value is returned as the final solution (line 15).

Incorporating User Input

A distinguishing characteristic of our framework is being able to incorporate user feedback to customize explanations. As Figure 1 demonstrates, customizing the explanation based on features of interest, namely exercise and smoking (Figure 1 right) makes it easier to understand how model logic varies for different values of these features. When a user inputs a set of features that are of interest to him, we simply restrict the candidate set of predicates $\mathcal{ND}$ from which subspace descriptors are chosen (See Objective Function) to comprise only of those predicates with features that are of interest to the user. This will ensure that the subspaces in the resulting explanations are characterized by the features of interest. Furthermore, the metric **featureoverlap**($\mathcal{R}$) and the term $f_2(\mathcal{R})$ of our objective function ensure that the features that appear in subspace descriptors do not appear in the decision logic rules there by creating a clear demarcation.

Experimental Evaluation

We begin this section by comparing our approach with state-of-the-art baselines on real-world datasets w.r.t the fidelity vs. interpretability trade-offs and unambiguity of the generated explanations. We then discuss the results of a user study that we carried out to evaluate how easy it is for humans to reason about the behavior of black box models using the explanations generated by our framework.

Datasets We evaluate our framework on the following real world datasets: 1) A dataset of **bail outcomes** collected from various U.S. courts during 1990-2009 (Lakkaraju, Bach, and Leskovec 2016) comprising of demographic information and details of past criminal records for about 86K defendants. Each defendant is assigned a class label based on whether he/she has been released on bail or locked up. 2) A dataset of about 21K high school **student performance** (Lakkaraju, Bach, and Leskovec 2016) records collected from a school district between 2012-2013 with various details such as grades, absence rates, suspension history. The class label of each student indicates if he/she graduated high school on time, dropped out, or encountered a delay in graduation. 3) **Depression diagnosis** dataset collected by an online health records portal comprising of medical history, symptoms, and demographic information of about 33K individuals. Each individual has either been diagnosed with depression or is healthy.

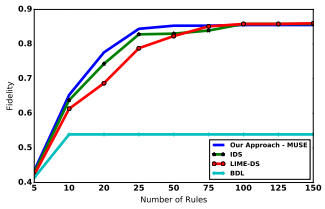
Baselines We benchmark the performance of our framework against the following baselines: 1) Locally interpretable model agnostic explanations (LIME) (Ribeiro, Singh, and Guestrin 2016) 2) Interpretable Decision Sets (IDS) (Lakkaraju, Bach, and Leskovec 2016) 3) Bayesian Decision Lists (BDL) (Letham et al. 2015). While IDS and BDL were developed as stand alone interpretable classifiers, we employ them to explain other black box models by treating the instance labels assigned by black box models as the ground truth labels. Since LIME approximates black box classifiers using multiple locally linear models, the approximations created by LIME and our approach have representational differences. To facilitate fair comparison, we construct a variant of LIME known as LIME-DS where each local model is a decision set (a set of if-then rules) instead of being a linear model.

Experimental Setup We generate explanations of multiple classes of models: deep neural networks, gradient boosted trees, random forests, decision trees, SVM. Due to space constraints, we present results with a deep neural network of 5 layers in this section, however our findings generalize to other model classes. Our optimization problem has the following parameters $\lambda_1 \cdots \lambda_5$ (scaling coefficients) and $\epsilon_1 \cdots \epsilon_3$ (constraint values). We employed a simple tuning procedure to set these parameters (details in Appendix). We set other parameters as follows: $\epsilon_1 = 20$, $\epsilon_2 = 7$, and $\epsilon_3 = 5$. Support threshold for Apriori algorithm was set to 1%.

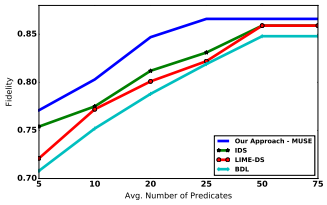
Experimentation with Real World Data

Analyzing the Tradeoffs between Fidelity and Interpretability To understand how effectively different approaches trade-off fidelity with interpretability, we plot fidelity vs. various metrics of interpretability (as outlined in the previous section) for explanations generated by our framework (without user input regarding features of interest) and other baselines. We define fidelity as the fraction of instances in the data for which the label assigned by the explanation is the same as that of the black box model prediction. Figures 2a and 2b show the plots of fidelity vs. number of rules (*size*) and fidelity vs. average number of predicates (ratio of *numpreds* to *size*) respectively for the explanations constructed using MUSE, LIME-DS, IDS, and BDL. These results correspond to explanations of a 5 layer deep neural network trained on the depression diagnosis data. Similar results were observed with other data sets and black box model types (See Appendix). It can be seen from Figure 2a that our framework (MUSE) and IDS achieve the best trade-offs between fidelity and number of rules. Furthermore, Figure 2b shows that our framework MUSE significantly outperforms all the other baselines when trading off fidelity with average number of predicates per rule. For instance, at an average width of 10 predicates per rule, explanations generated by MUSE already reach a fidelity of about 80% whereas explanations output by other approaches require at least 20 predicates per rule to attain this level of fidelity (Figure 2b). These results demonstrate that the explanations produced by MUSE provide significantly better trade-offs of fidelity vs. complexity compared to other state-of-the-art baselines.

Evaluating Unambiguity of Explanations We can readily evaluate the unambiguity of approximations constructed by our approach MUSE, IDS, BDL using two of the metrics outlined in previous section, namely, *ruleoverlap* and *cover*. Note that decision list representation by design achieves the optimal values of zero for *ruleoverlap* and N for *cover* since each else-if clause ensures that every instance satisfies a single rule in the list and else clause ensures that no instance is left uncovered. We found that the approximations generated using IDS and our approach also result in low values of *ruleoverlap* (between 1% and 2%) and high values for *cover* (95% to 98%). LIME is excluded from this comparison since it does not even specify which local model is applicable to what part of the feature space.



(a) Number of Rules



(b) Avg. Number of Predicates

Approach	Human Accuracy	Avg. Time (in secs.)
MUSE (No customization)	94.5%	160.1
IDS	89.2%	231.1
BDL	83.7%	368.5
MUSE (Customization)	98.3%	78.3

(c) Results of User Study

Figure 2: Evaluating our framework MUSE: a & b) Fidelity vs. interpretability trade Offs for a 5-layer neural network trained on depression diagnosis data. c) Results of user study.

Evaluating Human Understanding of Explanations

Here, we report the results of three user studies that were designed to evaluate the ease with which users can understand and reason about the behavior of black box models using our explanations. The explanations that we showed to users have been constructed by approximating a 5 layer deep neural network trained on depression diagnosis data.

Comparing rule based approximations (task 1) We designed an online user study with 33 participants, where each participant was randomly presented with the explanations generated by one of the following approaches: 1) our approach MUSE 2) IDS 3) BDL. Participants were asked 5 questions, each of which was designed to test their understanding of the model behavior (as depicted by the explanation) in different parts of feature space. An example question is: *Consider a patient who is female and aged 65 years. Based on the approximation shown above, can you be absolutely sure that this patient is Healthy? If not, what other conditions need to hold for this patient to be labeled as Healthy?* These questions closely mimic decision making in real-world settings where decision makers such as doctors, judges would like to reason about model behavior in certain parts of the feature space. The answers to these questions could be objectively judged as right or wrong based on the decision logic encoded by the explanation. Based on this, we computed the accuracy of the answers provided by users. We also recorded the time taken to answer each question and used this to compute the average time spent (in seconds) on each question. Figure 2c (top) shows the results obtained using explanations from MUSE (without customization), IDS, and BDL. It can be seen that user accuracy associated with our approach was higher than that of IDS, BDL. In addition, users were about 1.5 and 2.3 times faster when using our explanations compared to those constructed by IDS and BDL respectively.

Customizing Explanations We measured the benefit obtained when the explanation presented to the user is customized w.r.t to the question the user is trying to answer. For example, imagine the question above now asking about a patient who smokes and does not exercise. Whenever a user is asked this question, we showed him/her an explanation where exercise and smoking appear in the subspace descriptors (See Figure 1 (right)) thus simulating the effect of the user trying to explore the model w.r.t these features. We recruited 11 participants for this study and we asked each of these participants the same 5 questions as those asked in task 1. Table 2c (bottom row) shows the results of our model customized to the question being answered. In comparison to the results given in the first study, it can be seen that the time taken to answer questions is almost reduced in half compared to the setting where we showed users the same explanation (which is not customized to the question being asked) each time. In addition, answers are also more accurate, thus, demonstrating that allowing users to explore the model behavior from different perspectives can be very helpful in reasoning about its behavior in different parts of the feature space.

Comparing our approach with LIME (task 2) In the final study, our goal was to carry out the comparison outlined in task 1 between our approach and LIME. However, preliminary discussions with few test subjects revealed that the ill-defined subspace notions of LIME make it almost impossible to answer questions of the form mentioned above. We therefore carried out an online survey where we showed each participant explanations generated using our model and LIME, and asked them which explanation would they prefer to use to answer questions of the form mentioned above. We recruited 12 participants for carrying out this survey and they unanimously preferred using explanations generated by our approach to reason about the model behavior.

Conclusions & Future Work

In this paper, we propose MUSE, a framework for explaining black box classifiers by highlighting how they make predictions in subspaces characterized by features of user interest. An interesting research direction would be to combine our framework with ongoing efforts on extracting interpretable features from images. For example, superpixels (Ribeiro, Singh, and Guestrin 2016) output by intermediate layers of deep neural networks can be fed into our framework to enable explanations of image classifiers. Furthermore, the notions of fidelity, interpretability and unambiguity that we outline in this work can be further enriched. For instance, we could imagine certain features being more easy to understand than others in which case we can associate costs with features, and choose explanations with smaller costs (and more interpretable features). Our optimization framework can easily incorporate these newer notions as long as they satisfy the properties of non-negativity and submodularity.

References

Agrawal, R.; Srikant, R.; et al. Fast algorithms for mining association rules.

Jakulin, A.; Možina, M.; Demšar, J.; Bratko, I.; and Zupan, B. 2005. Nomograms for visualizing support vector machines. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 108–117. ACM.

Khuller, S.; Moss, A.; and Naor, J. S. 1999. The budgeted maximum coverage problem. *Information Processing Letters* 70(1):39–45.

Koh, P. W., and Liang, P. 2017. Understanding black-box predictions via influence functions. *arXiv preprint arXiv:1703.04730*.

Lakkaraju, H.; Bach, S. H.; and Leskovec, J. 2016. Interpretable decision sets: A joint framework for description and prediction. In *KDD*.

Lee, J.; Mirrokni, V. S.; Nagarajan, V.; and Sviridenko, M. 2009. Non-monotone submodular maximization under matroid and knapsack constraints. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, 323–332. ACM.

Letham, B.; Rudin, C.; McCormick, T. H.; Madigan, D.; et al. 2015. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics* 9(3):1350–1371.

Lou, Y.; Caruana, R.; and Gehrke, J. 2012. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 150–158. ACM.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. 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*, 1135–1144. ACM.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2018. Anchors: High-precision model-agnostic explanations.

Rokach, L., and Maimon, O. 2005. Top-down induction of decision trees classifiers-a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 35(4):476–487.

Shrikumar, A.; Greenside, P.; Shcherbina, A.; and Kundaje, A. 2016. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*.

Teoh, S. T., and Ma, K.-L. 2003. Paintingclass: interactive construction, visualization and exploration of decision trees. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 667–672. ACM.

Yosinski, J.; Clune, J.; Nguyen, A.; Fuchs, T.; and Lipson, H. 2015. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*.

Zintgraf, L. M.; Cohen, T. S.; Adel, T.; and Welling, M. 2017. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*.

Appendix

Proof for Theorem

Statement: *The optimization problem in Eqn. 1 is non-normal, non-negative, non-monotone, submodular with matroid constraints.*

Proof In order to prove that the objective function in Eqn. 1 is non-normal, non-negative, non-monotone, and submodular, we need to prove the following:

- Prove that any one of the terms in the objective is non-normal
- Prove that all the terms in the objective are non-negative
- Prove that any one of the terms in the objective is non-monotone
- Prove that all the terms in the objective are submodular

Non-normality Let us choose the term $f_1(\mathcal{R})$. If f_1 is normal, then $f_1(\emptyset) = 0$. Let us check if this holds true:

It can be from the definition of f_1 that $f_1(\emptyset) = \mathcal{P}_{max}$ because the numpreds metric would be 0 in this case as there are no rules in the empty set. This also implies $f_1(\emptyset) \neq 0$. Therefore f_1 is non-normal and consequently the entire objective is non-normal.

Non-negativity The functions f_1, f_2, f_3, f_5 are non-negative because first term in each of these is an upper bound on the second term. Therefore, each of these will always have a value ≥ 0. In the case of f_4 which encapsulates the cover metric which is the number of instances which satisfy some rule in the approximation. This metric can never be negative by definition. Since all the functions are non-negative, the objective itself is non-negative.

Non-monotonicity Let us choose the term $f_1(\mathcal{R})$. Let us consider two approximations $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{A} \subseteq \mathcal{B}$. If f_1 is monotonic then, $f_1(\mathcal{A}) \leq f_1(\mathcal{B})$. Let us see if this condition holds:

Based on the definition of $numpreds$ metric, it is easy to note that

$$numpreds(\mathcal{A}) \leq numpreds(\mathcal{B})$$

This is because $\mathcal{B}$ has at least as many rules as that of $\mathcal{A}$. This implies the following:

$$-numpreds(\mathcal{A}) \geq -numpreds(\mathcal{B})$$

$$\mathcal{P}_{max} - numpreds(\mathcal{A}) \geq \mathcal{P}_{max} - numpreds(\mathcal{B})$$

$$f_1(\mathcal{A}) \geq f_1(\mathcal{B})$$

This shows that f_1 is non-monotone and therefore the entire objective is non-monotone.

Submodularity Let us go over each of the terms in the objective and show that each one of those is submodular.

Let us choose the term $f_1(\mathcal{R})$. Let us consider two approximations $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{A} \subseteq \mathcal{B}$. If f_1 is submodular then, $f_1(\mathcal{A} \cup e) - f_1(\mathcal{A}) \geq f_1(\mathcal{B} \cup e) - f_1(\mathcal{B})$ where $e = (q, s, c) \notin \mathcal{B}$

Let x be the number of predicates in the rule $e = (q, s, c)$. This implies that when e is added to either $\mathcal{A}$ or $\mathcal{B}$, the value of the numpred metric increases by x i.e.,

$$f_1(\mathcal{A} \cup e) - f_1(\mathcal{A}) = x = f_1(\mathcal{B} \cup e) - f_1(\mathcal{B})$$

This implies that the function f_1 is modular and consequently submodular.

Let us choose the term $f_2(\mathcal{R})$. Let us consider two approximations $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{A} \subseteq \mathcal{B}$. If f_2 is submodular then, $f_2(\mathcal{A} \cup e) - f_2(\mathcal{A}) \geq f_2(\mathcal{B} \cup e) - f_2(\mathcal{B})$ where $e = (q, s, c) \notin \mathcal{B}$

By definition, $featureoverlap(\mathcal{A}) \leq featureoverlap(\mathcal{B})$ because $\mathcal{B}$ has at least as many rules as $\mathcal{A}$.

Let $featureoverlap(\mathcal{A}) = x$ and $featureoverlap(\mathcal{B}) = x + \epsilon$ where $\epsilon \geq 0$. When we add e to $\mathcal{A}$, let $featureoverlap(\mathcal{A} \cup e) = y$, then $featureoverlap(\mathcal{B} \cup e) = y + \epsilon + \epsilon'$ where ϵ' denotes the feature overlap between the e and the rules that exist in $\mathcal{B}$ but not in $\mathcal{A}$ Therefore, $\epsilon' \geq 0$.

$$f_2(\mathcal{A} \cup e) - f_2(\mathcal{A}) = \mathcal{O}_{max} - y - \mathcal{O}_{max} + x = x - y$$

$$f_2(\mathcal{B} \cup e) - f_2(\mathcal{B}) = \mathcal{O}_{max} - y - \epsilon - \epsilon' - \mathcal{O}_{max} + x + \epsilon = x - y - \epsilon'$$

This implies that

$$f_2(\mathcal{A} \cup e) - f_2(\mathcal{A}) \geq f_2(\mathcal{B} \cup e) - f_2(\mathcal{B})$$

Therefore, f_2 is submodular.

f_3 has a very similar structure to f_2 and it can be shown that it is submodular by following analogous steps as above.

f_4 is the cover metric which denotes the number of instances that satisfy some rule in the approximation. This is clearly a diminishing returns function i.e., more additional instances in the data are covered when we add a new rule to a smaller set compared to a larger set. Therefore, this is submodular.

Consider the function f_5, the metric disagreement is additive / modular which means each time a rule is added, the value of disagreement is simply incremented by the number of data points incorrectly labeled by this rule. Since the metric disagreement is modular, the function f_5 is also modular which implies submodularity.

Constraints: A constraint is a matroid if it has the following properties: 1) $\emptyset$ satisfies the constraint 2) if two sets A and B satisfy the constraint and $|A| < |B|$, then adding an element $e \in B, e \notin A$ to A should result in a set that also satisfies the constraint. It can be seen that these two conditions hold for all our constraints. For instance, if an approximation B has $\leq \epsilon_1$ rules and approximation A has fewer rules than B, then the set resulting from adding any element of B to the smaller set A will still satisfy the constraint. Similarly, maxwidth and numdsets satisfy the aforementioned properties too.

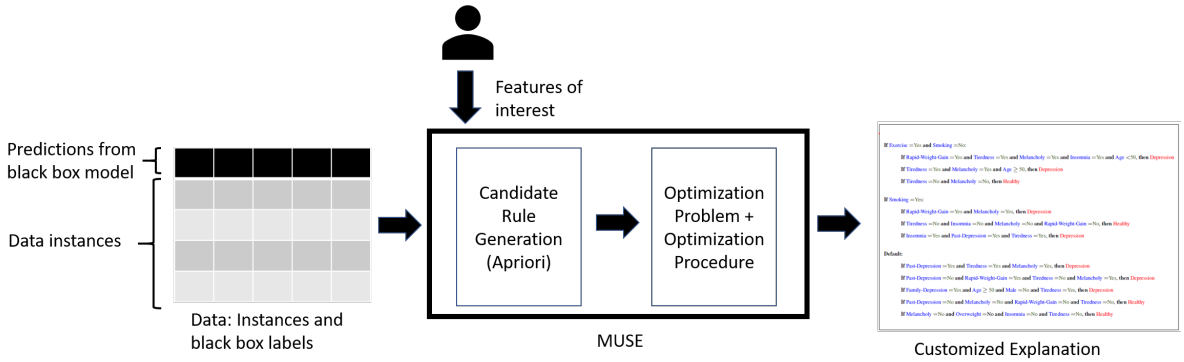


Figure 3: Algorithmic flow of MUSE approach: MUSE takes data, black box model predictions and user's features of interest. It outputs customized explanations.

Experiments & Results

Parameter Tuning We set aside 5% of the dataset as a validation set to tune these parameters. We first initialize the value of each λ_i to 100. We then carry out a coordinate descent style approach where we decrement the values of each of these parameters while keeping others constant until one of the following conditions is violated: 1) less than 95% of the instances in the validation set are *covered* by the resulting approximation 2) more than 5% of the instances in the validation set are *covered* by multiple rules in the approximation 3) the labels assigned by the approximation do not match those of the black box for more than 85% of the instances in the validation set.

We set the λ parameters of IDS in the same way as discussed above. BDL has three hyperparameters: 1) α which is the dirichlet prior on the distribution of the class labels. We set this to 1. 2) λ is the prior on the number of rules in the decision list and we set it to the same value as that ϵ_1 in our approach 3) η is the prior on average number of predicates per rule and we set to the same value as that of ϵ_2 in our approach. Our approach, IDS, and BDL take as input candidate sets of conjunctions of predicates. These candidate sets are generated using Apriori algorithm () with a support threshold of 1% which ensures that each conjunction holds true for at least 1% of the instances in the data.

Evaluating Interpretability of Customized Explanations To evaluate the interpretability of our explanations when features of interest are input by end users, we performed a set of experiments in which we simulate user input by randomly subsampling features of interest. Note that the baselines IDS, BDL or LIME are not designed to handle end user input when generating explanations. To benchmark the performance of our approach with user input, we construct variants of the baselines IDS and BDL where we first generate subspaces and then run IDS and BDL independently on instances belonging to each of these subspaces. Subspaces are generated by enumerating every possible combination of values of the randomly chosen subset of features (e.g., exercise = yes and smoking = yes, exercise = yes and smoking = no, exercise = no and smoking = yes, exercise = no and smoking = no). When averaged across 100 runs (where we randomly subsample features of interest for each run), we found that at the same level of fidelity, our explanations have about 22.02% and 38.39% fewer rules compared to those produced by variants of IDS and BDL respectively. Our framework also generated explanations with 17.53% and 26.28% decrease in the average width of rules compared to the variants of IDS and BDL respectively while maintaining the same level of fidelity. These results clearly highlight the importance of jointly optimizing the discovery of subspaces, and the corresponding decision logic rules so that the resulting explanation is both faithful to the original model and interpretable.

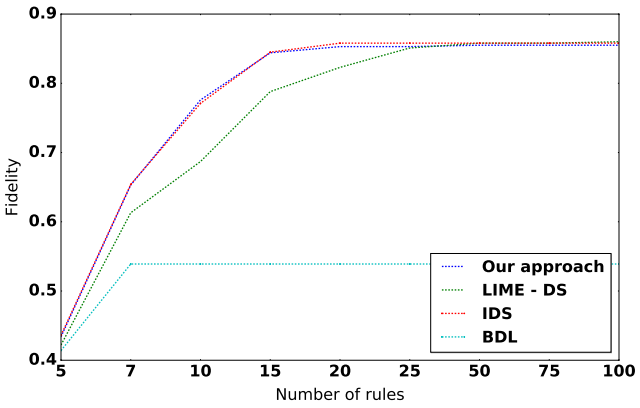


Figure 4: Number of rules vs. fidelity on bail data when approximating a deep neural network with 10 layers

Dataset	# of Data Points	Features	Classes
Bail Outcomes	86,152	gender, age, current offense details, past criminal behavior of defendants	Released on Bail, Not Released
Student Performance	21,713	gender, age, grades, absence rates & tardiness behavior recorded through grades 6 to 8, suspension/withdrawal/transfer history of students	Graduated on Time, Delayed Graduation Dropped Out
Depression Diagnosis	33,458	current ailments, age, BMI, gender, smoking habits, medical history, family medical history of patients	Depression, Healthy

Table 2: Summary of datasets.

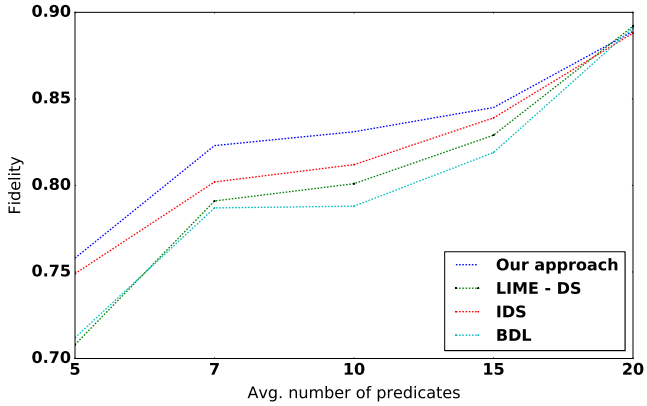


Figure 5: Avg. number of predicates vs. fidelity on bail data when approximating a deep neural network with 10 layers

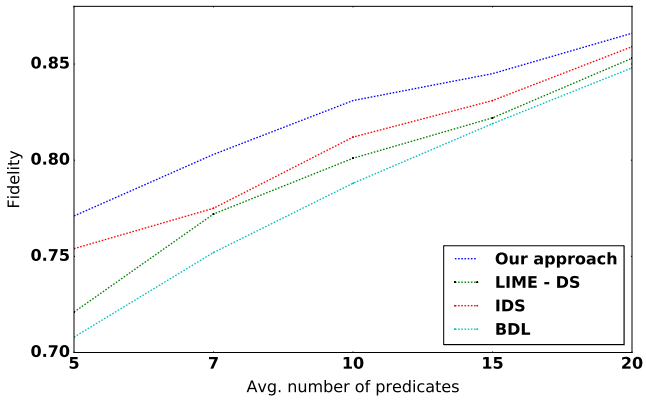


Figure 7: Avg. number of predicates vs. fidelity on bail data when approximating gradient boosted trees (100)

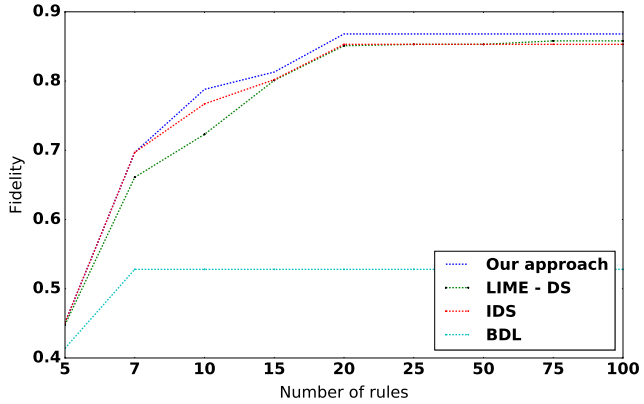


Figure 6: Number of rules vs. fidelity on bail data when approximating gradient boosted trees (100)

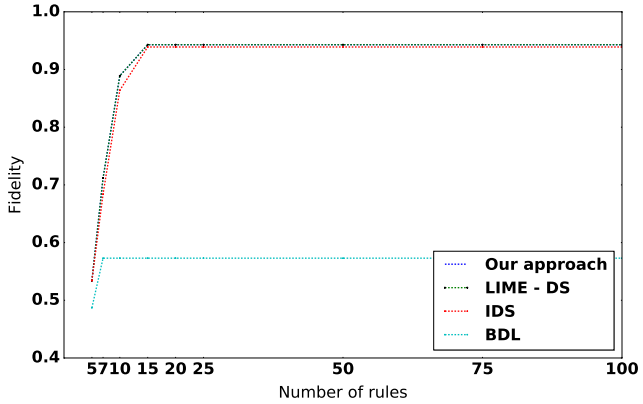


Figure 8: Number of rules vs. fidelity on education data when approximating SVM

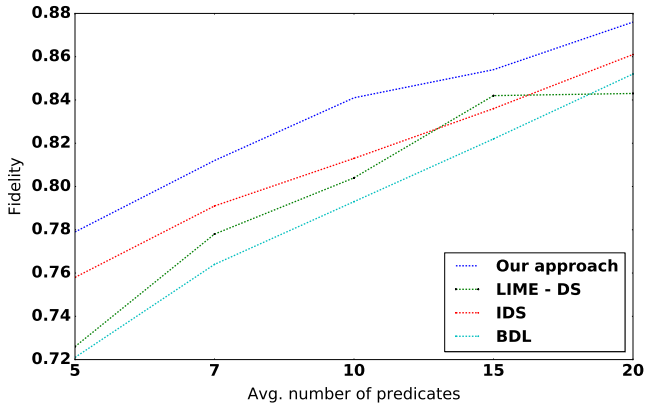


Figure 9: Avg. number of predicates vs. fidelity on education data when approximating SVM