Active Decision Set Induction For Model Interpretation

Jialin Lu[1]

[1]Simon Fraser University

June 3, 2019

Abstract

In the context of model-agnostic global interpretation of machine learning classifier, we consider an *active* algorithm to induce an interpretable decision set which faithfully approximates the input-output mapping of the blackbox classifier while being interpretable for human users to understand. We consider the problem as an optimization problem, and *actively query* the classifier to generate a *faithful* and *interpretable* global explanation. Borrowing ideas for multi-arm bandits, we identify an optimal set of high-precision faithful decision rules with the minimum number of necessary queries. Furthermore, several properties make our algorithm especially useful in practice, such as being model-agnostic and efficient in a fixed-budget setting. We demonstrate first in a synthetic dataset which shows how clever active querying helps find better solutions than previous methods, and then in a sentiment classification model where the induced rule set could be directly examined by human and finally, a genomics application where the induced rule set is further examined by biologists. A quantitive comparison on 20 datasets is also provided.

1 Introduction

Advanced Machine learning models with high predictive performance such as DNNs are generally hard to understand and interpret, but in certain cases, we are especially interested in interpreting what has been learned by the model. This model interpretation problem is important and useful for various cases such as model debugging (van den Berg, Ramabhadran, and Picheny 2017), causality issues (Kim and Bastani 2019; Bastani, Kim, and Bastani 2017) and trusting in critical decision making (Ribeiro, Singh, and Guestrin 2016; Guidotti et al. 2018), and itself is a interesting problem for knowledge discovery, where we wish to identify interesting input-output patterns by properly interpreting a well-trained classifier model. To be clear, we define model interpretation as learning an interpretable model, specifically a decision set consisting of "IF-THEN" rules, that approximates the input-output mapping of a trained classifier model. We refer this interpretable model as the *explanation* of the classifier model, and we say this interpretable model is *faithful* if it approximates the classifier model with high precision.

(write some more about biological applications that complex models)

We tackle model interpretation with an active approach, where our objective is to describe the global behavior of a trained classifier as a compact set of rules. Compared with standard supervised settings (Novak, Webb, and Wrobel 2009; van Leeuwen and Knobbe 2012; Lakkaraju, Bach, and Leskovec 2016), our objective is to extract rules from a classifier ***model*** instead of a labeled ***dataset***. The key ingredient of model interpretation problem is that we are not only given a dataset to learn from, but also an oracle which we could actively generate new instances to query, and use the query results to guide the process of rule induction.

We view this **active query** as the key to faithful model interpretation which, however, is not well addressed in previous approaches. Prior research could be categorized as 1) *Passive* model interpretation, which does not directly explains a classifier model but rather explains a dataset labeled by the classifier model (Lakkaraju et al. 2017; Ming, Qu, and Bertini 2018; Sushil, Suster, and Daelemans 2018; Lakkaraju et al. 2019) which is only faithful to the labeled dataset and is not guaranteed to be faithful to the classifier. A straightforward way to improve faithfulness is to sample an arbitrary large dataset (Ming, Qu, and Bertini 2018; Di Castro and Bertini 2019) or even exhaustively (Paçaci et al. 2019), which becomes inevitably intractable in most real-world domains.

2) *Active* model interpretation, in the sense that these methods actively query the classifier model with new synthetic instances to determine fine-grained decision boundary and thus are capable of generating faithful explanations. We further divide into *bottom-up* and *top-down* methods. *Bottom-up* methods leverage the recent success of local explanation approaches(Ribeiro, Singh, and Guestrin 2016; Ribeiro, Singh, and Guestrin 2018; Guidotti et al. 2018), which aims at explaining the prediction of a single instance, and constitute local explanations into a global one(Ribeiro, Singh, and Guestrin 2016; Ribeiro, Singh, and Guestrin 2018; Pedreschi et al. 2018; Puri et al. 2017). Generally, active query is utilized to generate a local explanation per instance, and then a generalization procedure is applied on all the local explanations, during which the classifier is put aside and not queried anymore. It becomes doubtful whether the generalization is faithful or global if it could not be checked and refined by querying the classifier model. Another line of work(Craven 1996; Bastani, Kim, and Bastani 2017), which we describe as *top-down* methods, works by hierarchically partitioning the feature space based on a decision tree structure. Whenever a node is being split, new instances are queried to determine the best split condition. However, as a inherent drawback of decision tree, the greedy splitting scheme results in sub-optimal (and often long) rules and since the algorithm does not explicitly consider interpretability but only precision, the resulted rules are generally less interpretable.

We argue that active querying the classifier model should simultaneously guide the whole process of model interpretation and we should also be aware of the interpretability-faithfulness trade-off.

We view model interpretation as a optimization problem and tackle it with the ***Active Decision Set induction from model (***Let us think of a good name***)*** approach inspired from compact decision set learning and multi-arm bandits. Our algorithm actively queries the classifier model to generate faithful explanations and are guaranteed to have high precision with a minimum number of necessary queries. By formulate interpretability and faithfulness explicitly as objective, our algorithm uses a stochastic local search algorithm to efficiently find an optimal rule set even in complex domains while ensuring the faithfulness and interpretability of identified explanations. More practically, our algorithm is a model-agnostic approach and could be applied to any classifiers that predict categorical value.

(This paragraph should talk about experimental results in brief. and is left intentionally empty for now)

We summarize the contributions as follows

- **Our insight into the problem**: we demonstrate, in thought and empirical experiments, the importance of active query as the key to global model interpretation.
- **Our algorithm is good for solving the problem**: we propose **XXXX**(the name) to generate *faithful* and *interpretable* global explanations given a classifier model.
- **Our approach works efficiently**: It reaches an optimal decision set with the minimum number of necessary active queries to the original classifier model. This is useful where exhaustive enumeration of input space is infeasible and where querying the classifier is expensive such as deep neural networks

(DNNs).

- **Our approach is very practical**: 1) it is model-agnostic and could be applied to any classifier. 2) efficient and useful in experiments balabala˜˜
- **Our experimental results**: (Say something about the experimental result, empty for now)

2 Background and related work

We argue that model interpretation is not only necessary but ubiquitously common in scientific research. It has been a common practice for decades in various domains and models from, on the simplest end of the spectrum, examining the coefficients in a logistic regression model, to interpreting the millions of parameters in a DNN. Complex data patterns underlying the system-of-interest, which are either not directly transparent or simply **intractable** for human, could be first accurately captured by sophisticated machine learning models and then discovered by carefully inspecting the learned model. In practice deploying directly an interpretable model like a decision tree or table often sacrifices predictive accuracy, indicating the underlying knowledge is not learned, though this trade-off might not be necessarily true as criticized by (Rudin 2018). But generally, the current situation is that complex blackbox models are kept being developed and deployed and there is an urgent need for model interpretation. We refer interesting readers to a survey (Guidotti et al. 2018).

2.1 Related work

Model interpretation has been a popular topic recently. We will not cover model-specific approaches such as extracting rules from a deep neural network(Fu 1994), SVM(Núñez, Angulo, and Català 2002) as well as approaches that assume a probabilistic output(de Fortuny and Martens 2015) which we also consider as inside-model information. We only consider a model-agnostic case where classifiers take an input instance and output a discrete categorical value.

As we discuss briefly in 1, prior research on model-agnostic interpretation could be categorized as either *passive* or *active* methods.

1) *Passive* model interpretation: in the sense that these methods that do not directly explain a classifier model but rather explain a dataset labeled by the classifier model (Lakkaraju et al. 2017; Ming, Qu, and Bertini 2018; Paçaci et al. 2019; Sushil, Suster, and Daelemans 2018; Di Castro and Bertini 2019). We argue that this line of work, however, is not able to generate faithfully explanations, simply because the fine-grained decision boundary of the classifier could not be revealed from the given dataset. A straightforward idea to complement this drawback is to sample an arbitrary large dataset (Ming, Qu, and Bertini 2018) or exhaustively (Paçaci et al. 2019) at the expense of computational cost. But this trick becomes infeasible for many real-world high-dimension applications, where enumeration of input space becomes intractable, not even mentioning the expense for querying a complex classifier like DNNs.

2) *active* model interpretation:

We further divide active methods into *bottom-up* and *top-down* methods. ***Bottom-up* methods** leverage the recent success of active query-based local explanation approaches(Ribeiro, Singh, and Guestrin 2016; Ribeiro, Singh, and Guestrin 2018; Guidotti et al. 2018), which aims at explaining the prediction of a single instance by generating and querying a neighbourhood *local* dataset, and constitute local explanations into a global one, by instance selection (Ribeiro, Singh, and Guestrin 2016; Ribeiro, Singh, and Guestrin 2018), agglomerative merging (Pedreschi et al. 2018) or generic algorithms (Puri et al. 2017). SP-LIME(Ribeiro, Singh, and Guestrin 2016) (SP short for 'submodular pick'), SP-Anchor(Ribeiro, Singh, and Guestrin 2018) selects several representative local explanations to constitute a global one considering a maximum coverage of different features. (Puri et al. 2017) propose an algorithm to first generate explanation per instance, and then use a generic algorithm to evolve an optimal set of explanations. (Pedreschi et al. 2018) proposes a

general local-to-global framework to first generate a local explanation for each instance and then iteratively merge similar explanations following an agglomerative hierarchical clustering approach, and finally outputs a optimal set of explanations measured by Bayesian information criteria or minimum description length. But these methods only utilize *active* query in the local explanation level, whatever procedure followed becomes a separate optimization problem, left the classifier aside. The generalization of explanations becomes rather a designed procedure, and without querying the classifier it is doubtful that the generalization would still be faithful to the model.

Another line of work(Craven 1996; Bastani, Kim, and Bastani 2017), which we describe as ***top-down* methods**, works by hierarchically partitioning the feature space into fine-grained regions where the predictions of a rule and that of the original classifier have a high agreement, using a decision tree structure. It borrows algorithm from standard decision tree learning like (Breiman et al. 1984) but differs in the node split procedure. Whenever a node is being split, new instances are queried to determine the best split condition to improve faithfulness. However, we identify two major weakness that is inherited from the decision tree algorithm: first, the tree grows in a greedy splitting manner, producing only locally optimal (and often very long) rules. Second, decision trees are generally considered to be less interpretable compared with decision sets, because algorithm is designed to consider only accuracy, instead of interpretability issues such as the number of rules, rule length, rule coverage, etc. Compared with decision sets, decision trees are generally considered to be less interpretable, as even carefully pruned and post-processed ones may still be "too cumbersome, complex, and inscrutable to provide insight into the domain"(Fürnkranz, Gamberger, and Lavrač 2012; Salzberg 1994).

(I remove the weakness of fixed number of query, since it is not a major drawback)

2.2 But why this new algorithm?

Before we introduce our approach in 4, perhaps a more concrete question is why don't we adopt some State-of-the-art methods for supervised rule induction such as (Lakkaraju, Bach, and Leskovec 2016; Su et al. 2016; Yang, Rudin, and Seltzer 2017; Angelino et al. 2017) and use it for model interpretation as done in (Lakkaraju et al. 2017; Lakkaraju et al. 2019). A two-fold answer is that we think these methods are not suitable for model interpretation because 1) these are passive methods and we need active query to ensure a faithful approximation of the original classifier. 2) They use association rule mining to reduce the space of all possible candidate rules. The rules will not be modified at the condition level during the algorithm, instead a rule is removed or added into the decison set. Pre-mining rules are especially important to scale these optimization-based approaches by reducing the search space, but does not make sense for model interpretation: A faithful explanation does not have to be a frequent pattern or statistically significant, as measured in the dataset. A more fine-grained modification to the conditions of a rule is necessary for faithful approximation.

To make a new algorithm suitable for decision set learning for model intepretation, we design our new approach to overcome the two aforementioned disadvantages:

1) passive methods -> active methods

2) we still use pre-mined rules (to reduce the search space and make it scale), but we allow the algorithm to query the classifier and modify the antecedents of a rule to faithfully approximate the classifier. This makes the search of our algorithm more flexible to find faithful explanations, but still efficient and scalable.

Note that there are some tricks to make the search efficient. But I am not sure if I should talk about it here, maybe just in the approach section.

To summarize, compared with previous approach,

- We actively query the classifier simultaneously the search process to ensure we find faithful (high-precision) explanations. And we do it efficiently.
- We only need a set of data instances X, and do not assume a data distribution $P(X)$ is given. Since it requires extra efforts to estimate density $P(X)$ from X as in (Bastani, Kim, and Bastani 2017; Ming, Qu, and Bertini 2018; Di Castro and Bertini 2019), and for most of the time in practice, we are only given a dataset X.
- We adapt the necessary number of new instances to query. This could avoid wasting computation cost when unnecessary.

3 Problem setting

3.1 Definitions, Notations, etc

Input of our algorithm:

a tuple (X, D_x, D_y, f, t)

- an instance set $X = \{x\}^n = \{(a_1, a_2, ..., a_m)\}^n$ of in total n instances and each instance has m features (attributes). $a \in dom(a)$ could be either categorical or continuous.
- the domain of input space: $D_x = \{dom(a_i)\}^m$ and $x \in D_x$.
- the domain of output space, all possible categories: $D_y = \{0, 1\}$ and $y \in D_y$.
- a pre-trained classifier model $f : D_x \mapsto D_y$
- a target category t: we wish to interpret the model for this class.

For simplicity and also to avoid tricky problem of conflict of rules, we only consider a binary classification case $D_y = \{0, 1\}$ and we only consider to find rules for the target (positive) class $t = 1$. This setting could also be viewed as learning disjunctive normal forms (DNF) and could be easily extended to multi-class classification, by setting the target class $t = 1$ and treat other classes as 0 .

Output of our algorithm:

A set of rules: $S = \{r_1, r_2, ...\}$

A Rule $r = \mathbf{c} \rightarrow t$ has two parts, an antecedent and a consequent. The antecedent $\mathbf{c} = \bigwedge_{i=1}^{I} c_i = c_1 \wedge c_2 \wedge ... \wedge c_I$ is a conjunction of I conditions and the consequent t is the target class, as we mentioned above we only consider $t = 1$. For a given instance x, $r(x) = 1$ if x satisfies the antecedent of this rule and else $r(x) = 0$.

3.2 Problem definition

3.2.1 Intuitively about the objective

The output, a rule set S should have some good property. Essentially three point:

- to be interpretable: the rules should be *general* and not too-specific to make it. A rule should be short (small number of rule conditions, since human could handle at most 7 ± 2 cognitive entities at one time (Miller 1956)). And it should cover a large volume of the input space (the input domain D_x) in order to be a good rule.
- to be faithful: high precision on approximating the original classifier.

- to be global: defined as high recall. Note that the decision set serves as a global explanation of the classifier, it should have a high recall. It should cover a large volume of input space and as many instances as possible.

3.2.2 Interpretability:

- length of rule. Average length, or the max length of rule
- number of rules.
- space-coverage of a single rule. It is defined as the probability of that it applies to a instance $x \in D_x$, specifically defined as $cov(r) = E_{x \sim P(x)}[r(x)]$. However, since we do not know the probability of x, we then assume a uniform distritbuion and defined it more concretely as the fraction of input domain covered by the rule: $cov(r) = cov(\mathbf{c} \rightarrow t) = \frac{|\mathbf{c}|}{|D_x|}$ (I still need to re-write this coverage definition later, and I give an example first) . For example if$r = (1 \leq a_1 \leq 4) \wedge (a_2 \in [\text{'grey'}, \text{'blue'}]) \rightarrow t$ and the given input domain is defined as $D_x = (a_1 \in [0, 10], a_2 \in [\text{'grey'}, \text{'blue'}, \text{'red'}])$ then $cov(r) = \frac{4-1}{10-0} * \frac{2}{3} = 0.2$.
- instance-coverage of a single rule: the number of instances covered by a single rule.
- space-overlap: overlapped space of rules, anytime two or more rules overlaps, the
- instance-overlap:overlapped instances

3.2.3 Faithfulness:

- define as expected precision: the probability of a instance covered by this rule being classified by the original model as the target class t. $\hat{prec}(r) = E_{\{x|r(x)=1, x \sim P(x)\}}[\mathbb{1}_{f(x)=t}]$

3.2.4 Globalness

- define as recall: the space coverage and instance-coverage of the entire decision set. .The space coverage of a decision set is the volume of the union of covered space of each instance $cov(S) = cov(\bigcup r)$.

Above we talked about the three point (interpretability and faithfulness)and now we talked about the objective for the rule set. The objective is a combination of seven separate function, as in (Lakkaraju, Bach, and Leskovec 2016).

$$\max Q(S) = \sum ...$$

However, we want to emphasize one critical difference of our objective and objective for supervised tasks. And that is the precision term. As for a supervised task, the precision could be computed directly on the dataset as $E_{\{x|r(x)=1, x \sim X\}}[\mathbb{1}_{f(x)=t}]$, in our case the precision could only be estimated as expectation precision $\hat{prec}(r) = E_{\{x|r(x)=1, x \sim P(x)\}}[\mathbb{1}_{f(x)=t}]$. In order to measure it we need to sample instances (not necessarily from the true datasets) and query the classifier, while the rest of objective could be computed directly.

This major difference raises a critical issue: on the one hand we want to estimate the objective – in particular the expected precision – as precise as possible, while on the other hand, since evaluating the expected precision requires expensive queries to the classifier, we cannot afford exhaustive sampling. We need a clever way to handle this, as we will discuss in 4.4.2 using methods for Best Arm Identification (BAI) in the multi-arm bandits setting.

4 Approach

4.1

4.2 Approach

We will give a brief outline of algorithm and then details on implementations.

Basically, we formulate the approach as a local search problem, we initialize the decision set as empty and in each iteration, we generate a set of actions (modify the current decision set) and apply an action to improve the objective. The choice of which action to apply is a mix of greedy step and random step in order to escape from local maximum.

For a greedy step, we measure the increase of objective of each action and choose the best. Note that we utilize **active query** when we measure the objective to determine the best action.

For random step, we randomly select an action or use random restart.

We first initialize a decision set S. At each step t while a termination is not met, we generate a set of action $A(S_t)$ of the current S_t. The generation of a candidate action could be done by apply several actions, such as modifying a rule(add/remove/modify a condition), remove one rule or add a new rule. After the actions is generated, we have two options, whether to exploit (choosing the best), or explore (choosing a random action).

4.3 Action generation

there are many actions,

- add a new rule
- remove a rule
- select a rule, and change one of the condition (remove a condition, add a new condition or modify the value of a condition).

This action generation could be done in many ways... and during the implementation and experiments, I feel that this is a key part of challenge. There are many ways to do it, I am still experimenting a good way.

In some Bayesian rule learning methods, they assign a probability to actions and sample them. There are also many pruning techniques. There is a huge space here to improve scalability (which is not our focus)

4.4 Applying action

4.4.1 Random step (Exploration)

randomly select an action and update the current S or re-start (initialize S) randomly.

4.4.2 Greedy step (Exploitation)

Here we utilize active query to find the best solution. We using a so-called best arm identification for multi-arm bandits to adaptively query the classifier and determine the best action.

Algorithm 1 The overall algorithm of active model interpretation for learning a decision set from the classifier

INPUT: ϕ - the objective function, f - the classifier model, X - the instance set, (D_x, D_y) the input and output domain, t - the target class (default to 1),
OUTPUT: $S = \{r\}$ - the decision set

procedure MAIN
  $S_0 \leftarrow \emptyset$ *or a random initialization*
for $t = 1, 2, \ldots$:
  $A \leftarrow GenerateActionSet(S_t, X)$.
  if explore **then**
    pick a random $a \in A$. ▷ Choose a random action
  else
    $a \leftarrow BestAction(A, S, f)$ ▷ Identify the best action by active query
  $S_t \leftarrow a(S_{t-1})$
  $S_{max} \leftarrow (\phi(S_t), \phi(S_{max})$.
  return $S* \leftarrow \underset{S \in \{S_0, S_1, \ldots, S_t\}}{\text{argmin}} \phi(S)$
procedure GENERATEACTIONSET(S,X)
  return A
procedure BESTACTION(A,S,X)
  return a^*

Figure 1: This is a caption

As we talked briefly in, 4.2, the objective function, in particular the expected precision term, cannot be computed directly but only estimated as an integral over all instances satisfying this rule. The naive way to measure objective increase of these actions based on the predicted label of the current dataset is not going to fully describe the classifier and leads to a faithful approximation. For estimating the expected precision and then determine the best action of best objective, we need to actively generate new samples and query it.

We think solving this greedy step as a best arm identification problem in the *pure-exploration* multi-arm bandits setting is a natural way to do it.

In the context of multi-arm bandits, there is a set of arms and a user could sequentially pull one of the arms to get reward. The reward of each arm is attached to a fixed but unknown distribution. While in standard settings the objective is to maximize the cumulative sum of rewards, in the best arm identification setting, we are try to identify the best arm with highest estimated reward given minimum number of pulls.

In our problem, and each arm is a candidate action and pull an arm means to generate a new sample of that action and query it and update the expected precision estimation. We then determine the top-1 best arm with minimum number of necessary queries. This is also called pure-exploration setting, the algorithm works by constructing a lower and upper bound for each arm, it then keep sampling (pulling the arm) and keep remove arms that has no chances of being the best arm.

This is called successive rejection, one of the most simple best arm identification method, and it is still considered as powerful and parameter-free.

And then we apply this action to update the decision set S

4.5 Theoretical issues

Apparently there are two issues:

One is the issue of local search.

And one is for the bandit problem for best action using active query. This seems to be a natural extension of pure-exploration bandit for best arm identification.

4.6 Practical Issues

4.7 Generation of actions

The generation of candidate actions is a challenging part of the algorithm, but there are previous methods for decision set learning which we could adopt.

Previously, this generation of actions is based on Monte-Carlo Tree Search. But it is also okay to use some Bayesian methods (from Bayesian rule set learning).

And there are also several efficient pruning technique, which is common in rule learning methods.

4.7.1

5 Experiment and Evaluation

- a synthetic dataset, showing the drawback of previous approach
- sentiment classification from tweets, where the induced explanation could be directly examined by human
- a genomics dataset, we demonstrate our applicability in real-world complex domains and the induced explanation should be usefully, by human biologist examination.
- a qualitative evaluation for performance and efficiency
- user test

5.1 Quantitative evaluation

do quantitative evaluation on UCI datasets (tabular data)

5.1.1 Performance

- interpretability
- faithfulness

5.1.2 Efficiency

- check the number of queries
- check running time
- anytime-interruption in fixed-budget setting. see what happened -> anticipated result is that it should find one or two most

6 Discussion and future direction

- interactive usages.
- possible option for customization (as in (Lakkaraju et al. 2017; Lakkaraju et al. 2019)) by feature preference ranking, controlled condition, or user-defined feature combinations.

6.1 Extended Customization

- **feature preference ranking**: search feature with an order, or only search a given set of features of interest,
- **controlled condition**: probe the machine behavior in a user-defined condition (for example, explaining a lung-cancer predictor, where we want to understand the machine prediction for non-smokers, we could fix the search for "smoking=False"
- **user-defined feature combinations**: x1 times x2 ...

7 Conclusion

References

Berg, Ewout van den, Bhuvana Ramabhadran, and Michael Picheny. 2017. "Training Variance and Performance Evaluation of Neural Networks in Speech". In *2017 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*. IEEE. https://doi.org/10.1109/icassp.2017.7952564.

Kim, Carolyn, and Osbert Bastani. 2019. "Learning Interpretable Models with Causal Guarantees". *CoRR* abs/1901.08576. http://arxiv.org/abs/1901.08576.

Bastani, Osbert, Carolyn Kim, and Hamsa Bastani. 2017. "Interpreting Blackbox Models via Model Extraction". *CoRR* abs/1705.08504. http://arxiv.org/abs/1705.08504.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?". In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 16*. ACM Press. https://doi.org/10.1145/2939672.2939778.

Guidotti, Riccardo, Anna Monreale, Franco Turini, Dino Pedreschi, and Fosca Giannotti. 2018. "A Survey Of Methods For Explaining Black Box Models". *CoRR* abs/1802.01933. http://arxiv.org/abs/1802.01933.

Novak, Petra Kralj, Geoffrey I. Webb, and Stefan Wrobel. 2009. "Supervised Descriptive Rule Discovery: A Unifying Survey of Contrast Set, Emerging Pattern and Subgroup Mining". *Journal of Machine Learning Research*.

Leeuwen, Matthijs van, and Arno Knobbe. 2012. "Diverse Subgroup Set Discovery". *Data Mining and Knowledge Discovery* 25 (2): 208–42. https://doi.org/10.1007/s10618-012-0273-y.

Lakkaraju, Himabindu, Stephen H. Bach, and Jure Leskovec. 2016. "Interpretable Decision Sets: A Joint Framework for Description and Prediction". In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1675–84. KDD '16. New York, NY, USA: ACM. https://doi.org/10.1145/2939672.2939874.

Lakkaraju, Himabindu, Ece Kamar, Rich Caruana, and Jure Leskovec. 2017. "Interpretable & Explorable Approximations of Black Box Models". *CoRR* abs/1707.01154. http://arxiv.org/abs/1707.01154.

Ming, Yao, Huamin Qu, and Enrico Bertini. 2018. "RuleMatrix: Visualizing and Understanding Classifiers with Rules". *CoRR* abs/1807.06228. http://arxiv.org/abs/1807.06228.

Sushil, Madhumita, Simon Suster, and Walter Daelemans. 2018. "Rule Induction for Global Explanation of Trained Models". *CoRR* abs/1808.09744. http://arxiv.org/abs/1808.09744.

Lakkaraju, Himabindu, Ece Kamar, Rich Caruana, and Jure Leskovec. 2019. "Faithful and Customizable Explanations of Black Box Models". *AIES*.

Di Castro, Federica, and Enrico Bertini. 2019. "Surrogate Decision Tree Visualization Interpreting and Visualizing Black-Box Classification Models with Surrogate Decision Tree". *CEUR Workshop Proceedings* 2327 (January).

Paçaci, Görkem, David Johnson, Steve McKeever, and Andreas Hamfelt. 2019. "Why Did You Do That?: Explaining Black Box Models with Inductive Synthesis". *CoRR* abs/1904.09273. http://arxiv.org/abs/1904.09273.

Ribeiro, Marco Túlio, Sameer Singh, and Carlos Guestrin. 2018. "Anchors: High-Precision Model-Agnostic Explanations". In *AAAI*.

Guidotti, Riccardo, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. 2018. "Local Rule-Based Explanations of Black Box Decision Systems". *CoRR* abs/1805.10820. http://arxiv.org/abs/1805.10820.

Pedreschi, Dino, Fosca Giannotti, Riccardo Guidotti, Anna Monreale, Luca Pappalardo, Salvatore Ruggieri, and Franco Turini. 2018. "Open the Black Box Data-Driven Explanation of Black Box Decision Systems". *CoRR* abs/1806.09936. http://arxiv.org/abs/1806.09936.

Puri, Nikaash, Piyush Gupta, Pratiksha Agarwal, Sukriti Verma, and Balaji Krishnamurthy. 2017. "MAGIX: Model Agnostic Globally Interpretable Explanations". *CoRR* abs/1706.07160. http://arxiv.org/abs/1706.07160.

Craven, Mark William. 1996. "Extracting Comprehensible Models from Trained Neural Networks". PhD thesis, The University of Wisconsin - Madison.

Rudin, Cynthia. 2018. "Please Stop Explaining Black Box Models for High Stakes Decisions". *ArXiv e-Prints*, November, arXiv:1811.10154.

Fu, LiMin. 1994. "Rule Generation from Neural Networks". *IEEE Transactions on Systems Man, and Cybernetics* 24 (8): 1114–24. https://doi.org/10.1109/21.299696.

Núñez, Haydemar, Cecilio Angulo, and Andreu Català. 2002. "Rule Extraction from Support Vector Machines.". In *Esann*, 107–12.

Fortuny, Enric Junque de, and David Martens. 2015. "Active Learning-Based Pedagogical Rule Extraction". *IEEE Transactions on Neural Networks and Learning Systems* 26 (11): 2664–77. https://doi.org/10.1109/tnnls.2015.2389037.

Breiman, Leo, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. 1984. *Classification And Regression Trees*. Routledge. https://doi.org/10.1201/9781315139470.

Fürnkranz, Johannes, Dragan Gamberger, and Nada Lavrač. 2012. *Foundations of Rule Learning*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-75197-7.

Salzberg, Steven L. 1994. "C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers Inc., 1993". *Machine Learning* 16 (3): 235–40. https://doi.org/10.1007/bf00993309.

Su, Guolong, Dennis Wei, Kush R. Varshney, and Dmitry M. Malioutov. 2016. "Interpretable Two-Level Boolean Rule Learning for Classification". *ArXiv e-Prints*, June, arXiv:1606.05798.

Yang, Hongyu, Cynthia Rudin, and Margo Seltzer. 2017. "Scalable Bayesian Rule Lists". In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, 3921–30. ICML'17. JMLR.org. http://dl.acm.org/citation.cfm?id=3305890.3306086.

Angelino, Elaine, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. 2017. "Learning Certifiably Optimal Rule Lists". In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 35–44. KDD '17. New York, NY, USA: ACM. https://doi.org/10.1145/3097983.3098047.

Miller, George A. 1956. "The Magical Number Seven plus or Minus Two: Some Limits on Our Capacity for Processing Information.". *Psychological Review* 63 (2): 81–97. https://doi.org/10.1037/h0043158.