Incorporating Expert Knowledge into Rule Learning via Reinforcement Learning

Congjing Zhang

A thesis

submitted in partial fulfillment of the

requirements for the degree of

Master of Science

University of Washington

2023

Committee:

Shuai Huang, Chair

Chaoyue Zhao

Prashanth Rajivan

Program Authorized to Offer Degree:

Industrial & Systems Engineering

University of Washington

Abstract

Incorporating Expert Knowledge into Rule Learning via Reinforcement Learning

Congjing Zhang

Chair of the Supervisory Committee:
Professor Shuai Huang
Industrial & Systems Engineering

Rule learning algorithms have great interpretability compared with other machine learning models. They also express strong power in discovering interactions between diverse variables. However, the performance of rule learning is greatly limited by the quality and volume of available training data. We conduct a literature review to show that the combination of humans with machine learning algorithms is a solution to these problems. Thus, in this thesis, we propose an integration method of expert knowledge and rule learning via reinforcement learning (ERRL) which automatically involves an expert in the rule generation step of RuleFit. We split a node of decision tree in each time step by using the framework of the Markov Decision Process. Then we incorporate human knowledge with reinforcement learning by shaping rewards based on the expert responses to the chosen action. In an empirical evaluation, we train the ERRL model on a simulated dataset with binary variables. We show that incorporating expert knowledge can improve the classification accuracy compared with ERRL model without humans involved. ERRL model provides rule learning with an effective way to generate rules by iteratively and automatically engaging humans in the learning process.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Number | Page

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to begin by expressing my deepest gratitude to my advisor, Professor Shuai Huang, for his unwavering guidance and support throughout my master's years. He has been a compassionate and supportive mentor, fostering an environment that has allowed me to grow professionally. Professor Huang's expertise and encouragement have been instrumental in shaping my academic journey and the completion of this master's thesis.

I am also immensely grateful to my thesis committee members, Professor Chaoyue Zhao and Professor Prashanth Rajivan, for their valuable insights, constructive feedback, and continued support throughout the development of this project. Their expertise has been invaluable in refining my research and ensuring its rigor and relevance.

I would like to extend my sincere appreciation to my collaborators over the years, Professor Mei Li and Jun Song. Their insights greatly contributed to my work. I am truly fortunate to have had the opportunity to collaborate with them.

To the faculty and staff of the ISE department, Sheila Prusa and Jennifer Tsai, I am grateful for your assistance in navigating the administrative aspects of my research.

I am particularly indebted to my parents, Xufei Zhang and Anping Zhang. You have been a pillar of strength, providing me with endless encouragement and companionship throughout this process. I also need to mention my colleagues and friends, Ryan Lin, Xinyi Zhao, and all my friends at UW. Thank you for your constant support, understanding, and belief in my abilities.

DEDICATION

To my parents.

Chapter 1

INTRODUCTION

1.1 Motivation

Rule learning has been a major pillar of machine learning (ML) and artificial intelligence (AI). Compared with other ML/AI methods, it is commonly recognized for its remarkable interpretability [1]. But over the years, its development has been relatively slow, in contrast to other methodologies such as kernel methods and neural networks. One main challenge is its computational complexity: learning the optimal set of decision rules from data is known to be NP-hard [2] since the number of possible combinations of rules grows superexponentially with the number of variables in the dataset [3]. For this reason, rule learning has been largely relying on efficient heuristics. Nonetheless, rule learning is an essential method for making sense of data. One of its particular merits is that it can identify interactions between different variables in a complex dataset that is high-dimensional and has mixed types of variables.

From the 1970s to the 1990s, heuristic algorithms and logic induction approaches were the most used methods in rule learning. For example, the PRISM algorithm developed by Cendrowska employed a modified version of the information content heuristic to assess the quality of rules [4]. However, since PRISM required learning a rule set for each class (i.e., for classification problems), the resulting rule-based classifiers tended to be very complex. One of the first inductive logic approaches for rule learning is Quinlan's FOIL system [5]. It generated first-order rules that captured the underlying semantics by finding definitions of relations one by one and utilizing other relations as background knowledge. Nonetheless, the system depended on learning Horn clauses from data.

Since the beginning of the 21st century, there has been an explosion of efficient tree-based algorithms such as random forests [3] and sparse regularization models such as Least Absolute Shrinkage Selection Operator (LASSO) [6], the RuleFit method [7] adeptly brought them together. RuleFit initially builds a wide-ranging catalog of rules using a tree-based model and subsequently utilizes LASSO to choose a minimal set of rules that can obtain a predictive accuracy as good as the random forest model. Although RuleFit enjoys its computational efficiency, it is still a heuristic approach that leads to suboptimal solutions, i.e., it can discover predictive rules but it does not guarantee the discovered rules are the best ones. On the other hand, the quality and volume of available training data also greatly restrict the performance of any rule learning algorithm including RuleFit. To overcome these limitations, we propose a human-in-the-loop approach to help rule learning, since humans can compensate for these problems by leveraging prior knowledge. Particularly, we hypothesize that expert knowledge can provide a range of benefits for rule learning models such as data augmentations and computational guidance for better optimal solutions. Expert-Augmented Machine Learning (EAML) [8] has demonstrated the promising prospect of the combination of RuleFit and expert knowledge. EAML designed a novel regularization method by penalizing unreliable rules which have a large difference between clinicians' insights with the empirical results from RuleFit. Still, to the best of our knowledge, there has been no systematic approach to automatically integrate expert knowledge into rule learning. The main focus of this thesis is to devise an innovative model that can aid in surmounting this difficulty.

1.2 Research objectives

This thesis will start with a systematic review of the literature and summarize the potential lines of frameworks that can integrate human intelligence into machine learning algorithms such as the RuleFit. We will develop a preliminary model and algorithm by formulating tree learning within the framework of the Markov Decision Process (MDP) and integrating human knowledge with reinforcement learning (RL). We will evaluate the proposed model using simulation studies. We expect that our model can have higher classification accuracy than the model without expert knowledge when the maximum classification accuracy of the dataset or expert accuracy changes.

1.3 Organization of the thesis

This thesis is organized according to the following chapters: Chapter 2 will start with a systematic literature review of rule learning and expert knowledge integration in AI/ML. Chapter 3 will present our modeling framework to incorporate expert knowledge into RuleFit via RL. Chapter 4 will present a preliminary experimental evaluation of our proposed method. Chapter 5 concludes the thesis with the conclusion and future research.

Chapter 2

REVIEW OF THE STATE-OF-THE-ART

2.1 Integration of human knowledge into AI/ML

Integration of domain knowledge with AI/ML has been the subject of extensive research in recent years. Generally, there are a few major schools of approach:

- **The Bayesian framework:** It is natural to adopt the Bayesian framework to integrate human knowledge with AI/ML models, particularly for some ML models that already possess an inherent structure that allows them to incorporate human knowledge in the form of probabilistic relations. For example, for a Bayesian network (BN), encoded expert knowledge can help estimate the parameters of the conditional distribution [9, 10]. Another method for learning BN based on Monte Carlo simulations produced a posterior probability distribution for all potential parent sets of each variable based on the assumption that the ordering of nodes was known. Expert responses regarding whether some arcs really existed were used to decrease the entropy of the probability distribution [11]. In the ecological field, to evaluate the effects of grazing on birds, Kuhnert et al. and Martin et al. proposed a Bayesian generalized linear mixed model where the random effects associated with different grazing levels were assumed to follow a normal distribution [12, 13]. The random effects were modeled to allow for a shift in the mean which was reflected by knowledge from multiple experts and precision which indicated the similarity of expert knowledge. The Bayesian framework can also flexibly model a variety of data formats. For example, pairwise comparison data obtained from an expert can be continuously and automatically acquired to maximally reduce model uncertainty when searching for the structure of a Bayesian network [14]. Furthermore, in the biological field, gene relations decided by experts can determine the optimal Bayesian classification by building a set of constraints [15]. If the existing knowledge indicates $gene_1$ with $X_1 = 1$ is regulated by $gene_2$ with $X_2 = 0$ and $gene_3$ with $X_3 = 1$, the constraint can be $P(X_1 = 1|X_2 = 0, X_3 = 1) = 1$.

- **Reinforcement learning:** RL algorithms [16] consist of the interactions between the agent and environment, which is also a natural framework to characterize a human-in-the-loop process [17]. There are two common methods for human knowledge to be integrated into the RL model training. One is to shape rewards based on human responses. For example, the TAMER was proposed to allow a human to give the scalar reward signals to show their feedback on the observed actions [18]. Assigning greater positive weights to significant indicators based on the human judgment can shape the reward function of the MDP formulation [19]. Besides reward shaping, policy shaping is another method to include human-in-the-loop by making humans directly influence the policy. Griffith et al. proposed a method, Advise, that obtained expert feedback on the chosen action and changed the policy by labeling optimal actions [20]. Later, Advise was evaluated by taking human error into account and combining human feedback with the learning process [21].

- **Ad-hoc integration approaches:** Some particular ML models or specific domains inspire the development of some ad-hoc approaches to integrate human knowledge into the learning process. Descriptive and intuitive knowledge that is often expressed by plain language can be preprocessed to have more numerical formats for ML models [22]. For example, qualitative coding can be used to organize qualitative data by grouping them based on common characteristics to identify patterns within the data [23]. This method assigns inferential labels to chunks of data, which facilitates the incorporation of empirical human knowledge in social science into ML. During the postprocessing stage, qualitative knowledge can be used to organize and structure the code system in the natural language processing (NLP) field [24]. Also, in the biological field, to reduce the time spent on annotating multiple nuclear images, expert knowledge can be used to guide the learning process based on an Online Random Forest [25]. The task of image annotation was split into three levels where expert user input including final classification correction can refine the model.

2.2 The RuleFit framework

While RuleFit is not directly used in our proposed method in this thesis, it is helpful to introduce its general framework of learning rules, since our proposed method, if successful, will be integrated into the RuleFit framework (i.e., to replace the purely data-driven decision tree learning with our human-in-the-loop tree learning method) to achieve rule learning.

The RuleFit method proposed by Friedman and Popescu in 2008 [7] is a rule learning algorithm to create a set of interpretable decision rules. Each rule consists of a few variables and their ranges. A rule is formatted as follows:

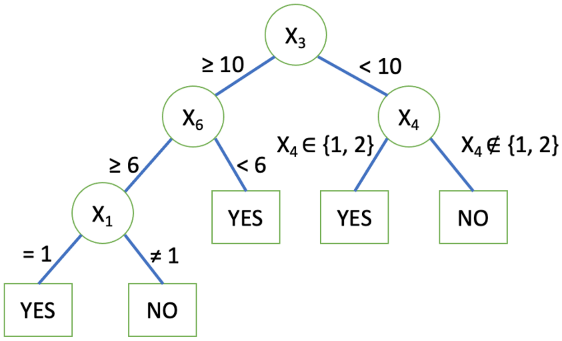
$$r(\mathbf{x}) = \prod_{j=1}^{p} 1_{v_j}(x_j). \tag{2.1}$$

It is the product of some indicator functions. $\mathbf{x}$ is an observation with p-dimensional variables in the dataset and x_j is the j-th element of $\mathbf{x}$. v_j represents an interval for continuous variables and a specified subset for discrete variables. $r(\mathbf{x})$ is equal to 1 when all conditions in the rule are met, otherwise the value of $r(\mathbf{x})$ is 0.

The challenge is to identify these rules from data. To do so, the RuleFit has two steps: rule generation and rule pruning. In the rule generation step, it fits a tree-based method, such as the random forest, to the dataset. According to the resulting tree ensemble model, we can dissemble the trees and extract a large set of rules. In the rule pruning step, we use these rules as input variables to a sparse learning model, such as LASSO [6], to select a subset of rules which remains high predictive accuracy. The details are shown below.

Step 1: rule generation

In this step, a decision tree learning method splits the data into different subsets by sequentially identifying variables and their cutoff values to enable the splitting. This process will



Rules	Conditions
r_1	$X_3 \geq 10$ & $X_6 \geq 6$ & $X_1 = 1$
r_2	$X_3 \geq 10$ & $X_6 \geq 6$ & $X_1 \neq 1$
r_3	$X_3 \geq 10$ & $X_6 < 6$
r_4	$X_3 < 10$ & $X_4 \in \{1, 2\}$
r_5	$X_3 < 10$ & $X_4 \neq \{1, 2\}$

Figure 2.2.1: An example decision tree and its corresponding rule list

result in a decision tree and the tree learning method follows a certain optimality principle or metric (such as information gain or prediction accuracy) to guide the splitting process. Figure 2.2.1 depicts an example decision tree and a set of candidate rules we can extract from it. Every rule is extracted from a path that goes from the root node to each leaf node in the decision tree. It includes a set of conditions that are used to split each node in the path. For example, if the data sample meets the conditions in r_4, the prediction is "YES".

However, a decision tree model can only classify each data sample based on one rule, neglecting the possibility that each observation may meet other rules with different variables and interactions. In other words, it employs an exclusive set of rules, which is quite limited to represent the diversity of the dataset and the interaction patterns of the variables. Random forest as an ensemble learning method can make up for the shortcoming. It builds a tree for each bootstrapped dataset. Thus, we can obtain diverse rule lists built on different subgroups of the training data. It gives us a chance to find other meaningful rule patterns that a data sample may meet.

Step 2: rule pruning

While the adoption of random forest can greatly expand the potential list of rules, an inevitable consequence is that there will be some low-quality rules (e.g., low predictive power) and many rules will be redundant. To further select a smaller set of rules that can sufficiently represent the data population while also maintaining high prediction accuracy, RuleFit proposed to use sparse learning methods such as LASSO [6] to conduct the rule pruning. LASSO is an efficient high-dimensional variable selection method for trimming large rule lists by using all rules built in the first step as predictors. It is a sparse linear model commonly used in bioinformatics and systems biology areas [26]. The formulation of LASSO is as follows:

$$\min_{\boldsymbol{\beta}} \|y - \mathbf{R}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 . \tag{2.2}$$

Here, $\mathbf{R} = [r_1, r_2, ..., r_d]$ is the set of rules generated in the rule generation step. It is used to predict the output variable y. The squared error part in 2.2, $\|y - \mathbf{R}\boldsymbol{\beta}\|_2^2$, measures the model fit. The sum of the absolute values of all elements in $\boldsymbol{\beta}$ is the L1-norm penalty part $\|\boldsymbol{\beta}\|_1$. It measures the complexity of the sparse regression model. λ is the penalty parameter that helps to balance the model fitness and model complexity, i.e., an increase in λ leads to a sparser estimate for $\boldsymbol{\beta}$. We can find optimal λ via cross-validation in RuleFit. Many algorithms are able to solve the optimization problem such as proximal gradient algorithms [27].

Chapter 3

INCORPORATING EXPERT KNOWLEDGE INTO RULEFIT VIA REINFORCEMENT LEARNING

To enable the integration of expert knowledge into rule learning, in this thesis, we will focus on the first step of RuleFit, the rule generation step. In fact, our approach is to develop a huamn-in-the-loop decision tree learning method based on the MDP and RL. RL algorithms can help identify the optimal decision tree by maximizing the long-term benefit, i.e., the cumulative reward based on different state-action pairs, obtained by interactions between an expert and the decision tree learning system. What's more, RL can enable an iterative process to improve policy based on previous expert responses. To do so, first, we formulate the decision tree building process as an MDP model. In each time step, the RL model selects an action to split a chosen node into two child nodes. Then the expert gives an opinion on whether the action is good or not. The basic reward is information gain after splitting the parent node in each time step. The final reward is influenced by expert feedback on the action. Finally, we use A2C [28] to find the optimal policy for decision tree building. We demonstrate the value of the novel model using a synthetic dataset consisting of different variables and observations.

It is worthy of mentioning that, our proposed method ERRL, an integration of expert knowledge and rule learning via reinforcement learning, is not the first attempt to develop an RL framework for decision tree learning. The typical approach to building a decision tree is to use a greedy algorithm. Its main idea is to split a parent node into child nodes based on a variable that the algorithm identifies as the best choice for the specific split. However, the greedy algorithm usually leads to suboptimal solutions due to its nature as a local search method. And because of this, it also makes the structure of the decision tree unstable and changes dramatically even with small changes in some samples. To overcome these issues, researchers have been trying to formulate the tree learning problem within RL frameworks. For example, in 2003, Pyeatt proposed an RL technique to split nodes over time, which retains a record of Q-value changes for each leaf, and partitions when this record signifies two distinct distributions [29]. An extension of this method was to conduct a reset operation to locally relearn the obsolete parts in the decision tree [30]. It served as a solution to the concept changes caused by modifications in the properties of the underlying population in a data stream. Similarly, Reinforcement Learning Trees (RLT) suggested splitting the variable that would lead to the greatest future improvement in later splits, rather than choosing the variable with the largest marginal effect from the immediate split [31].

Recently, decision tree is also built based on the framework of MDP where RL can be further incorporated. For example, Preda designed an MDP where each state included subsets of data [32]. An action can be taken in the form of a feature-subset pair which means it chose a feature to classify one of the subsets within the data samples. Furthermore, Reinforcement Learning-based Decision Trees (RLDT) was proposed to learn the optimal policy [33]. It formulated the state as combinations of feature-value pairs that represent the paths in decision trees. The action was performed by adding an extra feature to the policy or reporting the classification label of the example. Another approach is to use deep reinforcement learning (DRL). DRL with a hybrid action space was utilized to determine the optimal building strategy of decision tree [34]. This idea was extended to be used with a decentralized partial observable MDP (PDMDP) solved by Hybrid SAC algorithm [35, 36].

Given all these developments, still, there are some limitations that we aim to address. The performance of decision tree built based on RL is still limited as a pure data-driven approach, so the learning result depends on the quality and quantity of available training data. It can lead to meaningless rules or rules only useful for a particular dataset, which can be avoided if we can solicit input from experts. Thus, we propose ERRL to involve expert knowledge in RuleFit, especially in the rule generation step, by first building decision tree based on the framework of MDP and RL. In each time step, the policy will choose an action to split the parent node into two child nodes and the expert can give their response to the action. In the following parts, we will define the state, action, and reward of our decision tree building procedure.

State

We denote the set of observations in training data as ϵ which includes n observations. Let the set of states be S. A state $s \in S$ is a set as Eq.3.1 where $s_i \cap s_j = \emptyset$, $\forall i, j \in \{0, 1, ..., m\}, i \neq j, m \leq n$ and $\cup_{i=0}^{m} s_i = \epsilon$. If all samples from $s_i, \forall i \in \{0, 1, ..., m\}$ are classified in the same category, then the state s is defined as a final state. Each subset s_i in the state s is a partition of the data samples. In each time step, after taking an action, the state s will change to s' in Eq.3.2 where $s_{i_{d_1}} \in v_i, s_{i_{d_2}} \notin v_i$ and $s_{i_{d_1}} \cup s_{i_{d_2}} = s_i$. v_i is a criterion to split s_i. It is an interval for continuous variables, while for discrete variables it is a specified subset.

$$s = \{s_0, s_1, ..., s_m\} \tag{3.1}$$

$$s' = \{s_0, s_1, ..., s_{i-1}, s_{i_{d_1}}, s_{i_{d_2}}, s_{i+1}, ..., s_m\} \tag{3.2}$$

Action

Denote the action space as $A(s)$. Let $Vari$ be the set of variables, $va \in Vari$ be each variable and Val be the set of threshold values corresponding with variables in the dataset. $A(s)$ can be written as shown in Eq.3.3. An action is $\mathbf{a} = (va, e_{va}, i) \in A(s)$, which indicates that the variable va with threshold value e_{va} is used to further split the subset of samples s_i. The number of candidate threshold values for each variable can be user-specified. We can discretize continuous variables to get their threshold values. For each action, we define $v_i = \{x_{va} \in \mathbb{R} | x_{va} \geq e_{va}\}$ where $\mathbf{x}$ is an observation with p-dimensional variables and x_{va} is the value of the variable va in the observation $\mathbf{x}$.

$$A(s) = Vari \cdot Val \cdot \{0, 1, ..., m\} \tag{3.3}$$

Reward

We use information gain [37] as a basic reward $r(s, \mathbf{a})$ in our method. In this thesis, we assume y is binary, i.e., we only have two classes. The entropy function $Entropy(X)$ (as shown in Eq.3.4) measures the degree of impurity of data samples in a given node. The proportion of samples in the node that belongs to the class j is denoted as P_j. The reward function $r(s, \mathbf{a})$ (as shown in Eq.3.5) measures the reduction of the degree of impurity if the parent node is split into the child nodes.

$$Entropy(X) = \sum_{j=1}^{2} -P_j \log_2 P_j \tag{3.4}$$

$$r(s, \mathbf{a}) = Entropy(s_i) - \sum_{j=1}^{2} \frac{|s_{i_{d_j}}|}{|s_i|} Entropy(s_{i_{d_j}}) \tag{3.5}$$

In each time step, the expert will give an evaluation of the chosen action. If they think the action is critical to the decision tree building, they will answer “YES”. Otherwise, the answer will be “NO”.

$$f(s, \mathbf{a}) = \begin{cases} c & \text{if } answer = \text{“YES”} \\ -c & \text{if } answer = \text{“NO”} \end{cases} \tag{3.6}$$

Let $f(s, \mathbf{a})$ (as shown in Eq.3.6) be the shaping reward based on the expert's response and $r_f(s, \mathbf{a})$ (as shown in Eq.3.7) be the composite reward used for learning. c is a constant which can be decided by users.

$$r_f(s, \mathbf{a}) = r(s, \mathbf{a}) + f(s, \mathbf{a}) \tag{3.7}$$

Chapter 4

EXPERIMENTAL EVALUATION

4.1 Design of the numerical studies

We simulate a dataset to demonstrate our model ERRL. The dataset has 100 observations with 10 predictor variables and 1 outcome variable. To give a basic view of ERRL, we make all these variables binary with 0 and 1. We simulate 10 predictor variables as binomial distribution with a probability equal to 0.5, while the probability parameter of binomial distribution changes for the outcome variable to regulate noise in the dataset. We will compare the classification accuracy of ERRL and ERRL without expert knowledge in the following sections.

4.2 Training details

We choose A2C [28] as our baseline RL algorithm. The number of decision nodes is set as 4. Thus, in ERRL, there are total 3 effective actions that should be chosen. In our simplified case, all variables are binary, which makes e_{va} in action always be 1. Figure 4.2.1 shows the framework of ERRL method. In the beginning, $s = \{s_0\}$, which means all observations in the dataset belong to s_0. The first action is randomly chosen in RL. Thus, if i in action is larger than the maximum index of the subset in the current state, we set the reward as negative infinity to deny this action. The first effective action, i.e., i in action equal to 0, is $(Var3, 1, 0)$ chosen by RL. Then the state changes to $s = \{s_0, s_1\}$. Observations whose $Var3$ is equal to 1 belong to s_0, otherwise they belong to s_1. Reward is calculated based on information gain of the splitting and expert response. If the expert thinks the action is a good action, then reward will add 0.001. Otherwise, reward will minus 0.001. Afterward, the learning process moves to the next time step. Likewise, the second action $(Var5, 1, 1)$

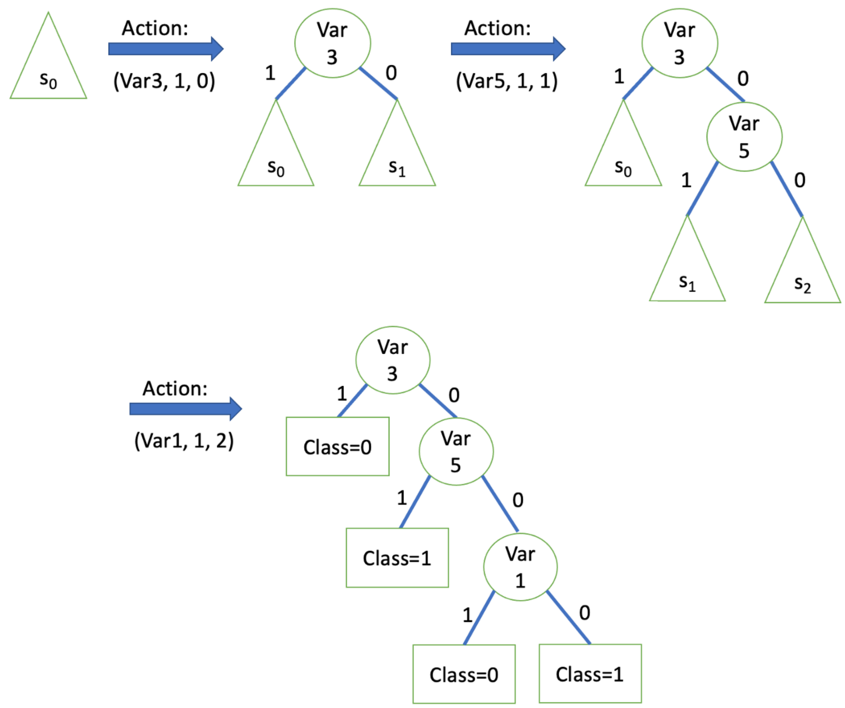


Figure 4.2.1: The framework of the ERRL method

classifying s_1 makes the state $s = \{s_0, s_1, s_2\}$ and we can get a new reward. Finally, after three effective actions have been chosen, the iteration ends.

4.3 Results

Figure 4.3.1 shows the relation between the improved classification accuracy which is calculated based on ERRL with expert and ERRL without expert knowledge and the percentage of the noise in the dataset. We use the maximum classification accuracy that the dataset can achieve to show the percentage of noise. In this setting, we assume the expert can provide perfect knowledge without faults. ERRL can improve classification accuracy when the noise in the dataset is relatively small. The simulated expert responses are based on the designed decision tree, which makes them kind of misleading when the designed tree cannot perform well. When there is almost no noise in the dataset, ERRL without expert knowledge can also classify observations in an efficient way, which makes the improved accuracy decrease.

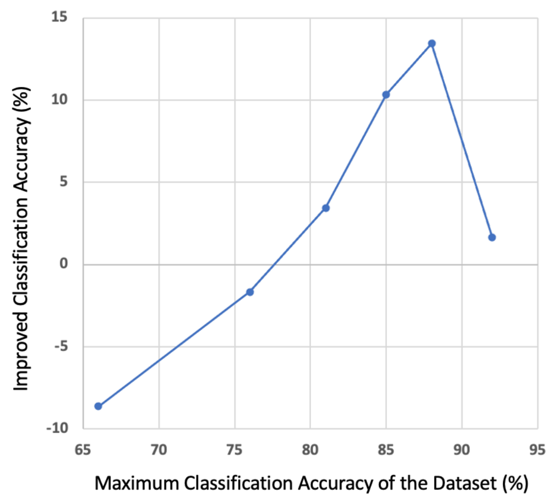


Figure 4.3.1: The relation between the improved classification accuracy and maximum classification accuracy of the dataset

We use the dataset whose maximum classification accuracy is 88% to study how expert accuracy influences classification accuracy. When expert accuracy is 100%, it means the expert is flawless. An accuracy of 0% means the expert gives totally wrong responses. As shown in Figure 4.3.2, when the accuracy of expert knowledge in ERRL increases, the classification accuracy of ERRL with expert also increases greatly. When the expert gives many wrong responses, it will not hurt the performance of ERRL in this simplified case.

4.4 Conclusion

We conclude that the ERRL model with expert knowledge involved can improve the classification accuracy especially when the dataset has relatively small number of noise outcomes. Choosing experts who can provide highly accurate answers can also improve the model performance on accuracy. According to the results of the simplified case, the ERRL model has the potential to be a powerful and efficient machine learning model that can involve an

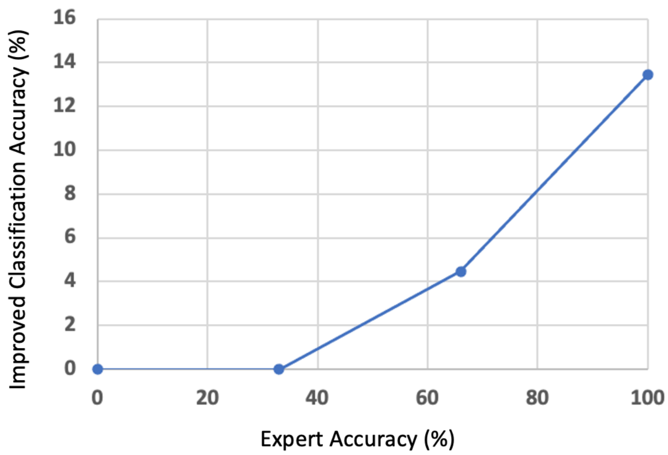


Figure 4.3.2: The relation between the improved classification accuracy and expert accuracy

expert in rule generation.

Chapter 5

CONCLUSION AND FUTURE RESEARCH

The contributions of this thesis are three folds. First, we present a literature review showing the value of integrating human knowledge into AI/ML algorithms. Second, we develop a basic methodological framework of the ERRL method to incorporate expert knowledge into the rule generation step of RuleFit based on the framework of MDP and RL. Specifically, we formulate an MDP model with data samples as state, variable-value-subset combinations as action, and information gain and expert responses as reward. Then we build decision tree based on this model which is solved by the A2C RL algorithm. Our third contribution is a proof-of-concept experiment study: Experiments on a simulated simplified dataset show that involving expert knowledge can improve the classification performance compared with ERRL without expert knowledge.

Future work on the ERRL model has several paths. First, we need to do more experiments to test our model performance in different aspects. For example, we can measure the stability of the ERRL model by comparing the number of unchanged rules when the size of the training dataset changes. Second, we will compare the ERRL method with other decision tree algorithms, such as ID3 [37] and C4.5 [38] to get a broad view of ERRL performance. Third, the rule pruning step in RuleFit needs to be considered. We focus on the first step in RuleFit to generate effective rules in this thesis. In the future, the whole RuleFit process should be included to demonstrate the model's proficiency. Finally, experiments on a more complex simulated dataset and real-world dataset need to be conducted. It can test the model performance based on both discrete and continuous variables simultaneously. In this thesis, we only choose binary variables, which does not require the model to learn the threshold value in action. To get a deeper understanding of our model performance, we will test on complex simulated datasets, such as data with different categories and imbalanced datasets. We will also implement our method on real-world datasets such as type 1 diabetes [39, 40, 41], depression [42, 43, 44], surgical site infection [45, 46], Alzheimer's disease [47, 48], sepsis [49, 50], and ADHD [51].

BIBLIOGRAPHY

[1] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[2] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[3] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[4] Jadzia Cendrowska. Prism: An algorithm for inducing modular rules. *International Journal of Man-Machine Studies*, 27(4):349–370, 1987.

[5] J. Ross Quinlan. Learning logical definitions from relations. *Machine learning*, 5:239–266, 1990.

[6] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[7] Jerome H Friedman and Bogdan E Popescu. Predictive learning via rule ensembles. *The annals of applied statistics*, pages 916–954, 2008.

[8] Efstathios D Gennatas, Jerome H Friedman, Lyle H Ungar, Romain Pirracchio, Eric Eaton, Lara G Reichmann, Yannet Interian, José Marcio Luna, Charles B Simone, Andrew Auerbach, et al. Expert-augmented machine learning. *Proceedings of the National Academy of Sciences*, 117(9):4571–4577, 2020.

[9] M Julia Flores, Ann E Nicholson, Andrew Brunskill, Kevin B Korb, and Steven Mascaro. Incorporating expert knowledge when learning bayesian network structure: a medical case study. *Artificial intelligence in medicine*, 53(3):181–204, 2011.

[10] Andrés R Masegosa and Serafín Moral. An interactive approach for bayesian network learning using domain/expert knowledge. *International Journal of Approximate Reasoning*, 54(8):1168–1181, 2013.

[11] Andrés Cano, Andrés R Masegosa, and Serafín Moral. A method for integrating expert knowledge when learning bayesian networks from data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(5):1382–1394, 2011.

[12] Tara G Martin, Petra M Kuhnert, Kerrie Mengersen, and Hugh P Possingham. The power of expert opinion in ecological models using bayesian methods: impact of grazing on birds. *Ecological applications*, 15(1):266–280, 2005.

[13] Petra M Kuhnert, Tara G Martin, Kerrie Mengersen, and Hugh P Possingham. Assessing the impacts of grazing levels on bird density in woodland habitat: a bayesian approach using expert opinion. *Environmetrics: The official journal of the International Environmetrics Society*, 16(7):717–747, 2005.

[14] Cao Xiao, Yan Jin, Ji Liu, Bo Zeng, and Shuai Huang. Optimal expert knowledge elicitation for bayesian network structure identification. *IEEE Transactions on Automation Science and Engineering*, 15(3):1163–1177, 2018.

[15] Shahin Boluki, Mohammad Shahrokh Esfahani, Xiaoning Qian, and Edward R Dougherty. Incorporating biological prior knowledge for bayesian learning via maximal knowledge-driven information priors. *BMC bioinformatics*, 18(14):61–80, 2017.

[16] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[17] Andreas Holzinger. Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Informatics*, 3(2):119–131, 2016.

[18] W Bradley Knox and Peter Stone. Interactively shaping agents via human reinforcement: The tamer framework. In *Proceedings of the fifth international conference on Knowledge capture*, pages 9–16, 2009.

[19] Yeping Hu, Alireza Nakhaei, Masayoshi Tomizuka, and Kikuo Fujimura. Interaction-aware decision making with adaptive strategies under merging scenarios. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 151–158. IEEE, 2019.

[20] Shane Griffith, Kaushik Subramanian, Jonathan Scholz, Charles L Isbell, and Andrea L Thomaz. Policy shaping: Integrating human feedback with reinforcement learning. *Advances in neural information processing systems*, 26, 2013.

[21] Thomas Cederborg, Ishaan Grover, Charles L Isbell Jr, and Andrea Lockerd Thomaz. Policy shaping with human teachers. In *IJCAI*, pages 3366–3372, 2015.

[22] Changyu Deng, Xunbi Ji, Colton Rainey, Jianyu Zhang, and Wei Lu. Integrating machine learning with human knowledge. *Iscience*, 23(11):101656, 2020.

[23] Nan-Chen Chen, Margaret Drouhard, Rafal Kocielnik, Jina Suh, and Cecilia R Aragon. Using machine learning to support qualitative coding in social science: Shifting the focus to ambiguity. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 8(2):1–20, 2018.

[24] Kevin Crowston, Eileen E Allen, and Robert Heckman. Using natural language processing technology for qualitative data analysis. *International Journal of Social Research Methodology*, 15(6):523–543, 2012.

[25] Florian Kromp, Inge Ambros, Tamara Weiss, Dominik Bogen, Helena Dodig, Maria Berneder, Teresa Gerber, Sabine Taschner-Mandl, Peter Ambros, and Allan Hanbury. Machine learning framework incorporating expert knowledge in tissue image annotation. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 343–348. IEEE, 2016.

[26] Tong Tong Wu, Yi Fang Chen, Trevor Hastie, Eric Sobel, and Kenneth Lange. Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics*, 25(6):714–721, 2009.

[27] Jun Liu, Jianhui Chen, and Jieping Ye. Large-scale sparse logistic regression. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 547–556, 2009.

[28] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.

[29] Larry D Pyeatt. Reinforcement learning with decision trees. In *21 st IASTED International Multi-Conference on Applied Informatics*, pages 26–31, 2003.

[30] Christopher Blake and Eirini Ntoutsi. Reinforcement learning based decision tree induction over data streams with concept drifts. In *2018 IEEE International Conference on Big Knowledge (ICBK)*, pages 328–335. IEEE, 2018.

[31] Ruoqing Zhu, Donglin Zeng, and Michael R Kosorok. Reinforcement learning trees. *Journal of the American Statistical Association*, 110(512):1770–1784, 2015.

[32] Mircea Preda. Adaptive building of decision trees by reinforcement learning. *Proceedings of the 7th WSEAS*, pages 34–39, 2007.

[33] Abhinav Garlapati, Aditi Raghunathan, Vaishnavh Nagarajan, and Balaraman Ravindran. A reinforcement learning approach to online learning of decision trees. *arXiv preprint arXiv:1507.06923*, 2015.

[34] Guixuan Wen and Kaigui Wu. Building decision tree for imbalanced classification via deep reinforcement learning. In *Asian Conference on Machine Learning*, pages 1645–1659. PMLR, 2021.

[35] Guixuan Wen and Kaigui Wu. Building decision forest via deep reinforcement learning. *arXiv preprint arXiv:2204.00306*, 2022.

[36] Olivier Delalleau, Maxim Peter, Eloi Alonso, and Adrien Logut. Discrete and continuous action representation for practical rl in video games. *arXiv preprint arXiv:1912.11077*, 2019.

[37] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1:81–106, 1986.

[38] J Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[39] Ying Lin, Xiaoning Qian, Jeffrey Krischer, Kendra Vehik, Hye-Seung Lee, and Shuai Huang. A Rule-Based Prognostic Model for Type 1 Diabetes by Identifying and Synthesizing Baseline Profile Patterns. *PLOS ONE*, 9:e91095, 2014.

[40] Mona Haghighi, Suzanne Bennett Johnson, Xiaoning Qian, Kristian F. Lynch, Kendra Vehik, Shuai Huang, and The TEDDY Study Group. A Comparison of Rule-based Analysis with Regression Methods in Understanding the Risk Factors for Study Withdrawal in a Pediatric Study. *Scientific Reports*, 6:30828, 2016.

[41] Kai He, Shuai Huang, and Xiaoning Qian. Early detection and risk assessment for chronic disease with irregular longitudinal data analysis. *Journal of biomedical informatics*, 96:103231, 2019.

[42] Ying Lin, Shuai Huang, Gregory E. Simon, and Shan Liu. Data-based Decision Rules to Personalize Depression Follow-up. *Scientific Reports*, 8, 2018.

[43] Aven Samareh, Yan Jin, Zhangyang Wang, Xiangyu Chang, and Shuai Huang. Detect depression from communication: how computer vision, signal processing, and sentiment analysis join forces. *IISE Transactions on Healthcare Systems Engineering*, 8(3):196–208, 2018.

[44] Ying Lin, Shuai Huang, Gregory E Simon, and Shan Liu. Analysis of depression trajectory patterns using collaborative learning. *Mathematical biosciences*, 282:191–203, 2016.

[45] Ziyu Jiang, Randy Ardywibowo, Aven Samereh, Heather Evans, William Lober, Xiangyu Chang, Xiaoning Qian, Zhangyang Wang, and Shuai Huang. A roadmap for automatic surgical site infection detection and evaluation using user-generated incision images. *Surgical infections*, 20(7):555–565, 2019.

[46] Chuyang Ke, Yan Jin, Heather Evans, Bill Lober, Xiaoning Qian, Ji Liu, and Shuai Huang. Prognostics of surgical site infections using dynamic health data. *Journal of Biomedical Informatics*, 65:22–33, January 2017.

[47] Yan Jin, Yi Su, Xiao-Hua Zhou, Shuai Huang, and Alzheimer's Disease Neuroimaging Initiative. Heterogeneous multimodal biomarkers analysis for alzheimer's disease via bayesian network. *EURASIP Journal on Bioinformatics and Systems Biology*, 2016:1–8, 2016.

[48] Ying Lin, Shan Liu, and Shuai Huang. Selective sensing of a heterogeneous population of units with dynamic health conditions. *IISE Transactions*, 50(12):1076–1088, 2018.

[49] Ying Wu, Shuai Huang, and Xiangyu Chang. Understanding the complexity of sep-

sis mortality prediction via rule discovery and analysis: a pilot study. *BMC medical informatics and decision making*, 21(1):1–15, 2021.

[50] Ameer Hamza Shakur, Shuai Huang, Xiaoning Qian, and Xiangyu Chang. Survfit: doubly sparse rule learning for survival data. *Journal of biomedical informatics*, pages 1–15, 2021.

[51] Ameer Hamza Shakur, Tianchen Sun, Ji-Eun Kim, and Shuai Huang. A rule-based exploratory analysis for discovery of multimodal biomarkers of adhd using eye movement and eeg data. *IISE Transactions on Healthcare Systems Engineering*, pages 1–15, 2022.