Capstone Final Paper Outline: Rule Extraction from Random Forests

William Jardee willjardee@gmail.com
Physics
Montana State University
Bozeman, MT 59715, USA

Editor: N/A

1. Introduction

1. One of the most widely applied structures applied to machine learning problems is the decision tree because of its interpretability and ease of implementation.

2. Decision trees are a specific application of a broader topic in machine learning called inductive logic programming.

3. Our goal is to find high-level rules that explain the whole decision space by using unsupervised learning to search a condensed representation of the whole random forest.

2. Related Works

1. Random forests are an ensemble approach of decision trees that sacrifices global interpretability for increased accuracy. (Explain local/global interpretability vs. explainability. Bring up the fundamental idea of white/black box models)

2. Random forests explain the decision space better but remove the desired explainability that decision trees have, so numerous methods have been proposed to extract a singular tree from a forest. (Talk about tree extraction approaches, such as counterfactuals, bayesian methods, and logic algebra)

3. Unsupervised learning is the search of a space when the absolute classes or performance of input values are unknown, searching for trends in the data that can be used to assume relationships or similarity. (Explain PCA, t-SNE, spectral clustering. Either explain here or make another paragraph about community extraction from complex social networks.)

3. Algorithm

1. Since this method focuses specifically on extraction from a forest; there will be an assumption that the random forest has already been generated via popular models, such as **insert the specific ones that we use/try**. (This will be all the assumptions

of this algorithm - pretty much everything else that we do not have time to account for in the paper.)

2. Each potential path on a tree can be thought of as a logical rule in modus ponens form and can be manipulated such that (*add math*), providing a collection of nodes that are connected via or's. (Continue with the node generation/extraction and generation of graph. This will probably take more than one paragraph, but that will be assessed as written.)

3. As the nodes are being added to the graph, there will likely be clauses that don't line up, such as mismatch intervals and positive and negative nodes of the same clause. (Go on to how nodes will be generated and accounted for to decrease spacial complexity.)

4. Once the whole map has been generated, the collective characteristics can be seen, and some pre-pruning can happen to smooth over the noise in the graph.

5. To begin getting rules from the graph, a fuzzy spectral clustering can be applied to extract clusters of clauses. (Go on to stipulate what must be in a cluster, i.e., a class and general application).

6. The set of deduced rules will likely be extensive, and post-pruning is in order. (Talk about possible pruning methods, such as many-class-rule-removal and eigenvector decomposition.)

7. Once vectors that describe each cluster have been extracted; these vectors can be converted back into logical statements and manipulated until they represent logical rules again.

8. The design decisions given here have removed a couple of possible other characteristics that could be exploited that individuals implementing this algorithm may want to think about. (Talk about eigenvalue importance characterization, node combination to create higher-level clauses, etc.)

4. Testing

This section will be a little rougher since we have not gotten results yet.

1. To set a benchmark to test our new algorithm, a common rule extraction approach from ILP will be applied, as well as classic random forest, decision tree, and neural network methods. (Each will allow a different desirable characteristic to be tested - explain how.)

2. The datasets choice covers a range of applications of the algorithm, allowing the model's viability to be comprehensively analyzed. (Go into the dataset choice, size, and reason.)

3. The data set was first tested with a small, synthetic data set with and without noise to test the general functionality of the algorithm. (This will be the data set and results.)

4. Next, the data set was tested against a real-world data set known to have a complex decision basis that describes it. (Go into the dataset, results, and others' results.)

5. Finally, a large-scale data set will be tested to see how effectively our algorithm can scale. (Explain the cost of accuracy and explainability.)

6. It can be seen that our method... **Explain the results more**. (There will likely be multiple paragraphs for this section, but no data yet.)

5. Conclusion

1. In this paper, we introduced a method to extract high-level, overlapping, complex decision rules from random forests. (Give an overview of the approach.)

2. Through testing a wide breadth of problems and comparing to other common methods that offer different benefits, we found that this method **insert results**.

3. There are some limitations in the algorithm laid out here and further work should be dedicated to exploring expansions on this algorithm. (Go into what we did not have time to do. Talk about applying this approach to decision forests and other learning methods.)