Quarter 2 Project Proposal:

A Dynamic Heuristic Approach to Decision Tree Forests

Group Members:  
Jacob Dipasupil, Petr Kisselev, Nikhil Alladi

# 

# 1 Motivation

In this paper, we aim to tackle two aspects of the standard random forest approach that we have identified as being problematic or in need of improvement. First, it is not surprising based on the stochastic nature of the random forest model that there is often a lack of consistency in results, with high run-to-run variance. This is traditionally offset with large sample values, but increasing the amount of samples, and therefore generated trees, naturally leads to unnecessary computational bloat for little improvement. Second, we believe that while Random Forests (RFs) generally provide an overall performance uplift in comparison to single-tree models, there is performance being left on the table as a result of the uniformly distributed random selection of model features and instance samples.

# **2 Method**

This project intends to expand upon the machine learning model of random forests. Random forests are predictive models that combine an ensemble of decision trees to create an aggregated prediction to avoid overfitting. Currently, as the name suggests, random forests incorporate randomness into their tree-making, in the way that they sample data and features to build the tree on.

Our project focuses specifically on the feature selection part of this process. Currently, the method of choosing features is a random sample without replacement. Our project aims to replace this with a heuristic that evaluates the performance of the decision tree at each iteration. The concept is as follows: we train a decision tree on training data, then evaluate its performance on validation data. We then analyze the tree and find the least effective attribute split(s), and replace those attributes with ones previously unchosen. We can then repeat this until our performance decreases, and assemble a “forest” of the last N decision trees.

This method is likely to perform better on datasets of notably high dimensionality. This allows the method to flourish without being constrained to a minimal set of replacement attributes.

# **3 Intended Experiments**

We will compare the performance of the dynamic semi-random forest with decision trees and ordinary random forest. We will benefit most from having many different datasets of varying (but mostly high) dimensionality, size, and feature dependence so we can discover the strong and weak points of our model. Since overfitting is the main problem we are trying to avoid, we will focus on running the models on a datasets with high dimensionality and observe different performance metrics, such as F1-score, ROC area, accuracy, precision, etc. Consistently higher performance for certain datasets can indicate to us which types of datasets our model performs better on. We compare our model with decision trees to see if we can improve on the overfitting problem, and with random forest to see if our new approach improves upon it.

The following dataset(s) are possible datasets that this method could be applied to with positive effect, due to their high dimensionality.

**Dataset choices:**

* <https://www.kaggle.com/datasets/hhs/health-insurance-marketplace>
* <https://catalog.data.gov/dataset/usgs-water-quality-data-for-the-nation-national-water-information-system-nwis>

There has already been previous research, but we believe that our approach significantly extends the approaches previously attempted. While this work discusses weighting the distribution, it does not take into account the dynamic framework that we will introduce. The importance of a dynamically adjusting distribution is clear, the model can correct for initial poor performance as it learns.

**Previous approaches:**

* <https://pmc.ncbi.nlm.nih.gov/articles/PMC3912194>