COMS 4995 - Parallel Functional Programming - Final Project Proposal

Phan Anh Nguyen (pn2363) Azhaan Zahabee (az2641)

November 22, 2021

Background and Motivation

Decision Trees are a class of supervised machine learning algorithms that are often used to solve both regression and classification problems. At a high level, the Decision Tree algorithm takes in a set of input data with corresponding labels, greedily builds a lookup-tree where data is grouped at each level according to a 'best-feature' to split the data on, and recursively applies this splitting down the branches until a stop criterion is met. During test time, a new input is classified by following the decision tree nodes until a leaf is reached. The predicted label is that associated with the leaf that the input data ultimately lands on.

The choice of which feature to best split on is based on two concepts: **Entropy** and **Information Gain**. **Entropy** for a set of data that is split on a feature that has c classes measures the degree of 'impurity' associated with a set of data. It is defined as:

$$Entropy(Node) = \sum_{i=1}^{c} -p_i log_2(p_i),\ p_i = \text{proportion of data with label } i \tag{1}$$

Information gain measures the expected reduction in entropy caused by partitioning the examples/data according to an attribute/feature. It is defined as:

$$Gain(Node, A) = Entropy(Node) - \sum_{c \in A} \frac{\mid Node_c \mid}{\mid Node \mid} Entropy(Node_c) \tag{2}$$

From these two equations, we can identify the 'best attribute/feature' to split our data on by selecting the attribute that gives us the most information gain at any node.

For our implementation, we will be focusing on Decision Tree algorithms for classification problems as regresison-based problems require extra pre-processing in the form of data discretization, which we will not focus on in this project. Additionally, since our project focuses on the performance associated with building the decision tree, we will not be implementing any prediction-based optimizations such as early-stopping or pruning.

Parallelization Approach

We have identified two potential levels of parallelization that can be done

1. **Entropy** parallelization: A sequential implementation of the Entropy calculation for a node would use a for-loop over each possible class that the label can have. For a multi-class label dataset, the runtime could increase accordingly. We hope to paralellize this in our implementation of Entropy.
2. **Information Gain** parallelization: A sequential implementation of the Information Gain calculation for a node would use a for-loop over each possible attribute that we could split on and then choose to split the data on the attribute that results in the highest gain. In our implementation, we hope to parallelize this process.

References

We have consulted with the following references:

- *Intelligence, A Modern Approach (Fourth Edition)*, Stuart Russel, Peter Norvig

- *Parallel and Concurrent Programming in Haskell*, Simon Marlow
- COMS 4995 - Parallel Functional Programming Lecture Notes
- COMS 4701 - Artificial Intelligence Lecture Notes
- https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html