EECS 349 PS2 Report

By Jiaming Li

1. The decision tree is represented as a list, in which every node takes form of [best_attribute, best_split_value, less_than_value_node, more_than_value_node] or a binary integer (0 or 1). When it is a binary integer, it represents that all of the instances that belongs to this leaf are pure (they belong to one category).

2. Examples are represented by a matrix. The matrix has as many rows as the number of attributes. Each column is an example. We chose this design because the matrix in Python is stored in row-major order. If we want to create a decision tree, we want to access the information in each attribute. Therefore it makes more sense to store data of each type of attribute in the same row.

3. For each node, we calculate the information gain (minimum entropy) for a split in each attribute. We pick the attribute with largest information gain. For each node in the decision tree, we store several information: the attribute we chose to split the node, the threshold for that attribute, the left node, right node, and parent node.

4. For the missing attributes, we replace them with the average of all other attributes of the same type. So for example we replace "?" in winning rate with 0.5 if the average winning rate is 0.5

5. We search the lowest point of entropy by calculating the entropy on both side of a point and moving to the side with lower entropy. The termination for each node split is when we reach a local minimum in entropy. The whole training process terminates when the height of the tree reaches the MAX_HEIGHT constant defined in the script.

6. The Boolean formula of the unpruned tree of maximum height3 (7 nodes) is the following:

(((numinjured<1.37)and((oppnuminjured<1.41)and(numinjured>0.54)))or((numinjured>1.37)and((oppnuminjured<2.26)or((oppnuminjured<2.26)and(numinjured>2.47)))))

7. The Boolean formula is true iff it is satisfied. That is, if an instance fall into one of the leaves labeled as "1". For example: if numinjured<1.37 and oppnuminjured<1.41 and numinjured>0.54, then the formula is satisfied and the tree predicts the result to be 1.

8. The pruning is done based on one principle: If using the majority of the given data's result as a classifier is better than splitting the node, then we should prune this node and all nodes that follows.

9. Maximum height = 3 (In this case pruned and unpruned tree are the same, but higher trees are harder to understand. Therefore we chose this height as an example)

(((numinjured<1.37)and((oppnuminjured<1.41)and(numinjured>0.54)))or((numinjured>1.37)and((oppnuminjured<2.26)or((oppnuminjured<2.26)and(numinjured>2.47)))))

10. The number of splits for a maximum height 9 pruned tree has about 200 splits. The unpruned tree has about 400 splits.

11. For the unpruned tree with maximum height 9, the accuracy is 0.917. The pruned tree with maximum height 9 is 0.922.

12. The learning curve graphs are shown in the appendix. There are a statistically significant difference between pruned and unpruned trees. The unpruned tree clearly overfits the training data when maximum height=30.

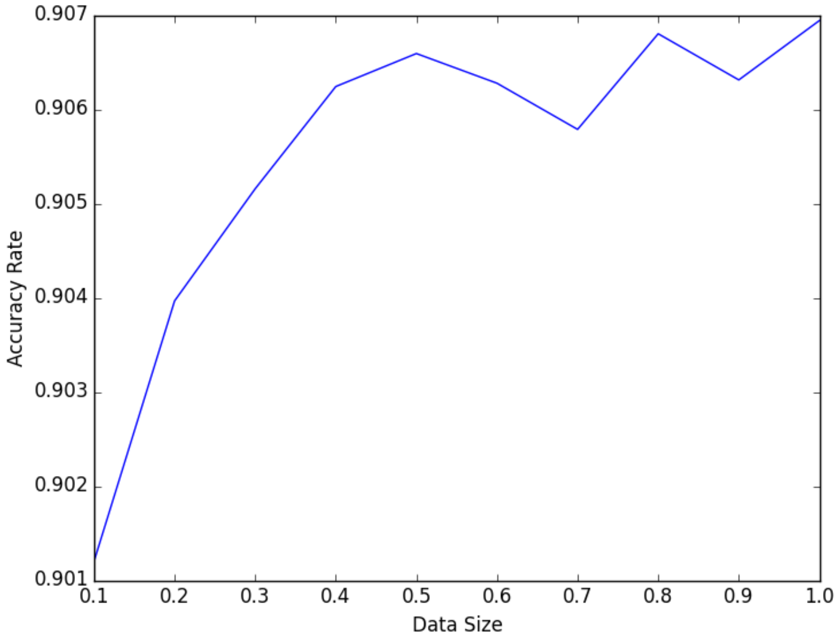
13. The tree with maximum height 9 is likely to perform well. It performs similarly on training and validation data with highest accuracy that we could get.

14. Some of the features are real numbers and some are integers. Also sometimes the features are always positive and sometimes they are negative. If we could map them all to real numbers from 0 to 1 and distribute them evenly, it would make the decision-tree building much easier. Because we don't have to worry about sudden change of the entropy in a narrow interval if all of them are relatively evenly distributed.

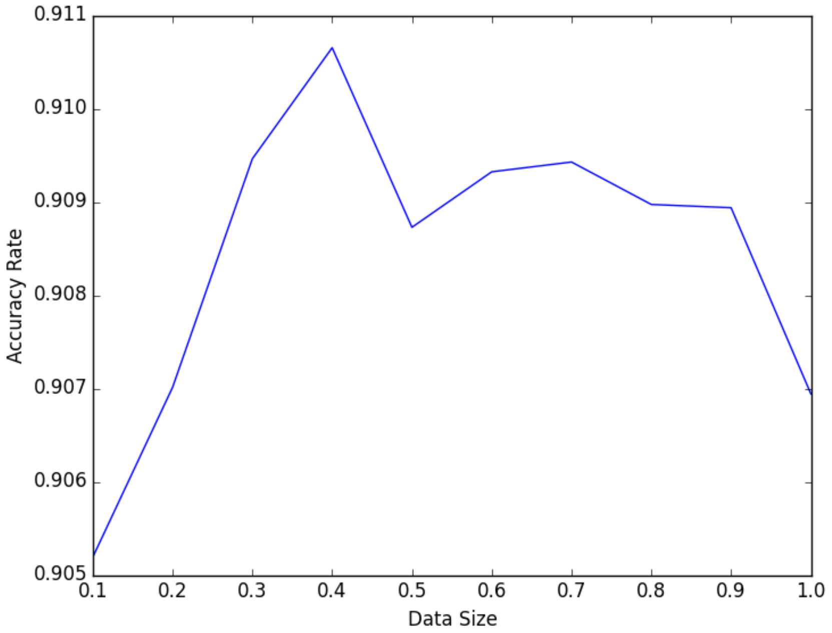
15. Jerry Li: splitting; Annie Fu: pruning; Ivy Zheng: everything else.

Appendix

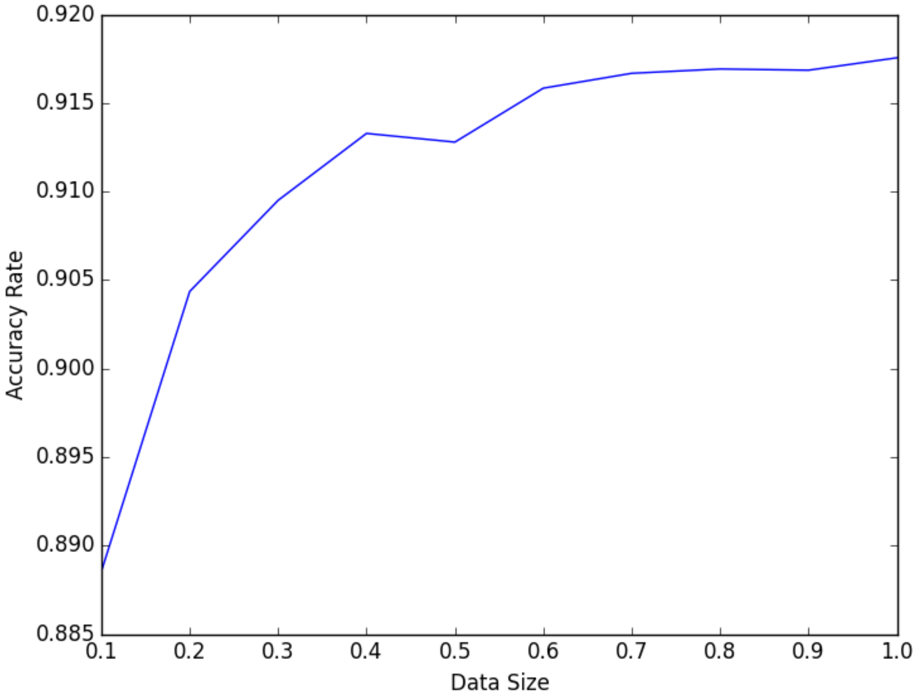
a) Height = 7 unpruned tree



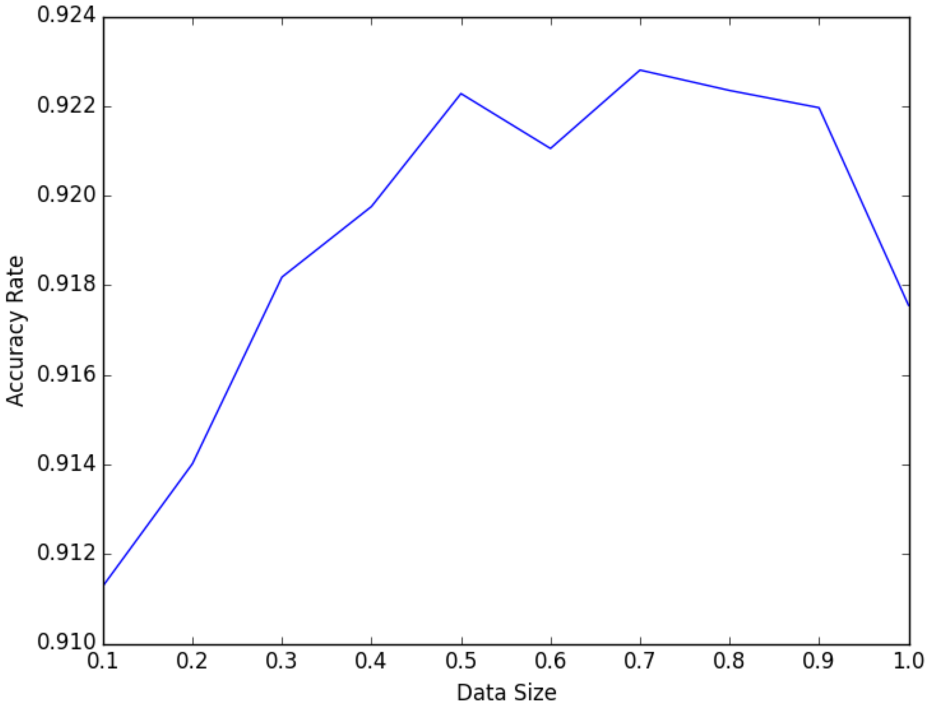
b) Height = 7 pruned tree



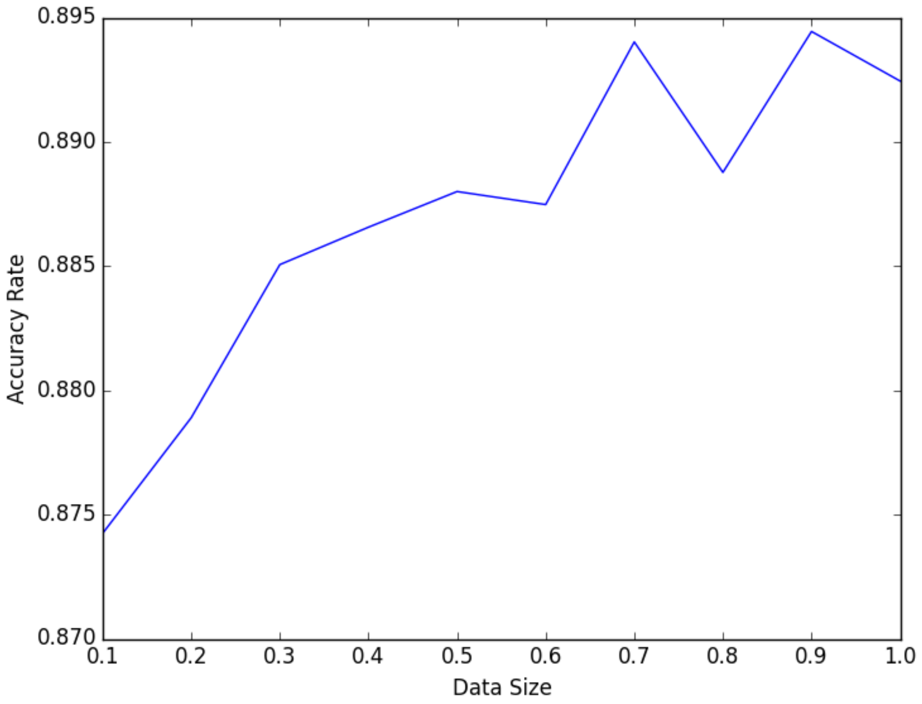
c) Height=9 unpruned tree



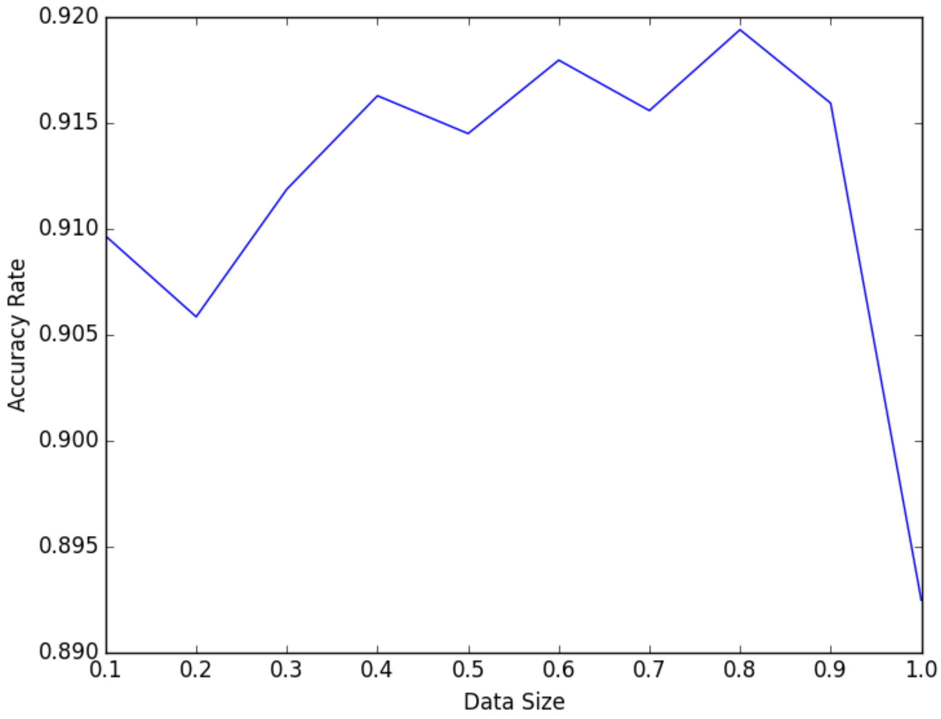
d) Height=9 pruned tree



e) Height=30 unpruned tree



f) Height = 30 pruned tree


Accuracy Rate
Data Size