HW3.5 - P2

9075858895, Yahn-Chung Chen

a) [3] Encoding a single chess board position into 35 features takes time. Using the results of running the information gain heuristic at the root of the tree on chess\_training.txt, which feature would you recommend removing from the feature vectors to minimize negative impact on accuracy? Why?

• Information gain:

Spcop: 0.00000

Since the information gain of Spcop is 0, it seems to useless for classifying the data. Therefore, I will recommend removing this feature to minimize the negative impact on accuracy.

b) [3] What is the percentage *decrease* in *error rate* between the unpruned decision tree’s performance on the test set and the pruned tree’s performance on the test set. Justify your answer by referencing both figures.

• Testing set accuracy

Unpruned decision’s tree: 0.86538

Pruned decision’s tree: 0.91827

• Error rate decrease:

0.91827 – 0.86538 = 0.05289

The reason leads to the lower accuracy for unpruned decision’s tree is overfitting. The two deep trees, one ending with Wkpos and another ending with Wtoeg, seems to overfit the data and also generate some redundant leaves with no example passed in.

c) [4] Explain why overfitting can be positively correlated to the number of the nodes in decision trees built on the same training data. Why does the pruned tree overfit less?

Overfitting will create some redundant branches to fit the training data well; however, this kind of branch is useless for interpreting the pattern of the data set. You will see some longest branches for classifying really small amount of data and leaving tons of leaf nodes with no data passed in. This condition leads to a sky high accuracy on the training set and relatively lower accuracy on the testing set. Pruning the nodes caused redundant branches can avoid passing data to a misleading tree and prevent making an error.