1) (9.8/12) What is overfitting? When does overfitting occur with the C4.5 decision tree algorithm? How can one detect overfitting?

Overfitting occurs when a classifier becomes too specific to the training data and fails to predict well on new examples from the same distribution.

With C4.5 overfitting occurs when the tree grows too large.

Overfitting can be detected by evaluating the performance on a set of labeled examples separate from the examples used for training. This can either be a completely separate “test set” or one of the folds used for k-fold crossvalidation.

2) (10.6/16) Give brief descriptions of the two major types of strategies to combat overfitting in decision trees. You do not need to give equations or details, just a high-level description that distinguishes the two and gives advantages and disadvantages of each.

The two major types of strategies are early stopping and pruning.

For early stopping, one needs to constantly monitor the performance of the tree on a separate “test set”. It’s best to stop when (or a little after) the performance on the test set starts to go down. An advantage is that it avoids the extra computation of computing a large tree then eliminating branches. A disadvantage is that the data in the test set is not used to build the tree.

The other major approach is to build a tree that’s too large, then prune branches using some method such as Reduced Error Pruning or confidence intervals. Reduced Error Pruning requires a test set to determine whether the total errors among the children of a node are worse than the errors at the node itself. The confidence interval approach doesn’t require a test set but relies on estimates coming from the training set. This uses all the data to build the tree, but the error estimates are less reliable.

There are various hybrids, such as determining the stopping point using k-fold crossvalidation instead of holding out the test set, than building the tree using the whole data set but stopping at this point.

3) (9.1/12) Give a high-level description of SplitInfo. Don’t give the equation, just explain its purpose, what type of quantity it measures, and what information about nodes, attributes and labels you need in order to calculate it.

SplitInfo measures the entropy of examples when splitting a node by a given attribute. It depends only on the number of examples at each child node of the split, not on their labels. It is used to compensate for the unfair advantage given to “wider” splits, by dividing the Gain by the SplitInfo.