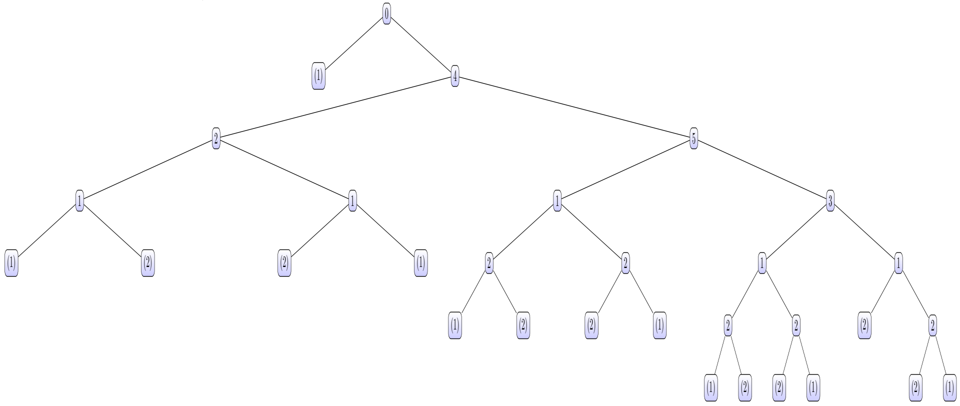
Artificial Intelligence Methods, exercise 3

Rendell Cale

April 25, 2018

Implementing and testing decision-tree-learning I implemented the decision-tree-learning algorithm in python, trained it and ran on the test data. The final tree turned out to have 31 nodes (when counting leaves which are decisions). A visualization of the tree is given below.



The internal node numbers indicate what attribute one should look at. If the attribute is 1, go to the left subtree and if it is 2 go to the right subtree.

Testing it on the test data gave an accuracy of 92%, and we get the same when we build the tree multiple times. It doesn't necessarily have to yield the same tree though, since the pluralityValue function uses randomness to break ties.

When we build the tree with the random importance function, the trees change each time and have wildly different accuracies. I ran the algorithm and tested it ten thousand times to gather statistics on the performance and got the following results.

```
———Tree statistics with random importance———
        Times run: 10000
        Average tree size (biggest/smallest): 275.64 (667.00/9.00)
        Size std dev: 103.34
        Accuracy avg (best/worst): 0.78 (1.00/0.36)
        Accuracy std dev: 0.11
———Tree statistics with information gain importance———
        Tree size: 31
```

```
Accuracy avg: 0.78
```

From the results above we see that the information gain based approach has alot better performance on average than the random importance. By performance I mean mainly that the average tree size is smaller (31 vs 275) since both actually get same accuracy on average, 78%. Note however that the random based approach got between 100% and 36%, which is a big spread. Accuracy of 36% means that the decision tree is wrong more than right. Since there are only two alternatives it would then be better to do the opposite!

Conclusion Information gain importance is in better in that it is more consistent and tends to produce trees which are several orders of magnitude smaller. Since it uses the data to build to structure of the tree, it will probably also contain more information about the structure of the data.