Decision Trees

Taylor Berger

February 17, 2014

1 High Level Overview

My code was written in the functional language, Haskell. To get started, if you lack a Haskell compiler you will need to do the following (assuming you are using the 'yum' package manager):

```
sudo yum install haskell-platform
```

After you have the Haskell platform installed, or if you already had it, you will need to install the following extension (the CS machines on the moons server have the Haskell platform already installed, but you will need this cabal package):

```
cabal install list-extras
```

The program accepts two arguments. The first being the location of the file containing the training data. The second is the location of the file used for validation of the decision tree. The top 'data' and 'type' definitions are to help the readability of the function definitions. The decision tree a standard rose tree implementation with the nodes representing the index of the nucleotides in any given DNA string.

If you are familiar with Haskell, you can skip this section. There are type signatures on the end of most lines that will give the type of the data that was constructed with that line(these are designated by the ':: TYPE'). Above the function definitions are the type signature for that function. It explains the arguments the function takes and what the function returns.

2 Important Bits of Information:

Unfortunately, in the course of tarballing my program to turn it in last week, I accidentally overwrote my source file before I was able to commit it to my Github (https://github.com/teberger/haskell). Dr. Estrada instructed me to write the report from my memory. I was able to rewrite a small portion of the source file in a few hours, which is what I submitted with this report.

3 Accuracies and Performance

Using the formula for information gain, I am able to obtain an 85% accuracy rate with the training data provided. The tree that is learned by this model is fully grown, no pruning or decision making was made in accordance with the Chi Squared test.

Chi Squared testing. In the previous version of my program, the Chi Squared testing at the 95% level yielded somewhere around 90% accuracy when used with the information gain method. The misclassification impurity method yielded less accurate results, somewhere around 80% if my memory is correct.