Report of Decision Tree

Lin Sun
sun@unm.edu

1. How my code work:

In this project, I need to implement ID algorithm to classify the training samples and then use validation sampes to verify my final decision tree.

To build the decision tree, first, I need to read training samples from the file. For each sample, there is one Sample object instance associated to it. It records down the detailed information for that sample, that is, a piece of DNA sequence. Because there are 57 DNA positions for one DNA sequence, I just name each position as “pos” plus the position number. Each position in DNA sequence is treated as one attribute. At the end of each reading, I add a new attribute “isPromotor” to indicate if this sequence of DNA is a promotor.

Then I did recursive call to build decision tree. Each time when run method is called, it will check if the sample list for current node is pure, which means if the sample list only contains positive or negative sequences. If it is pure, just stop classification and mark it as a leaf node. Otherwise, keep classifying until the node sample list is pure or all attributes are used for classification. There are two methods I implemented to decide which node to use. The first one is to calculate gain value. When deciding which attribute to use for classification, I calculate the gain for each attribute not used so far, then choose the attribute with the higest gain for classification. Here I also implements chi-square testing, which will check if the attribute is significant or not. So just calculate chi-square value and compare with the standard value. If not significant, just skip it this time. Another method is to use misclassification error to decide the attribute. I calculate the misclassification for each attribute, and choose the attribute with the lowest misclassfication error. After classification, I prepare new child node and call run method in child node recursively to build the whole decision tree.

After the decision tree is built, I need to use validation samples to verify my decision tree. As previous said, first I need to read all samples as object instances from evaluation file. Then I will classify the samples using the decision tree I just built. Each time when I split the samples, I should check whether the list is pure. If the list is pure, it means that the current list can't be splitted any more. It can be marked as one class. Otherwise, I need to continue splitting the list until the list is pure or the node in the algorithm is a leaf node in decision tree.

After classification of validation samples, I need to calculate error rate for each leaf node using the equation as follow:

$$Err(t, D(t)) = \frac{|\{(\mathbf{x}, c(\mathbf{x})) \in D(t) : c(\mathbf{x}) \neq label(t)\}|}{|D(t)|} = 1 - \max_{c \in C} \frac{|\{(\mathbf{x}, c(\mathbf{x})) \in D(t) : c(\mathbf{x}) = c\}|}{|D(t)|}$$

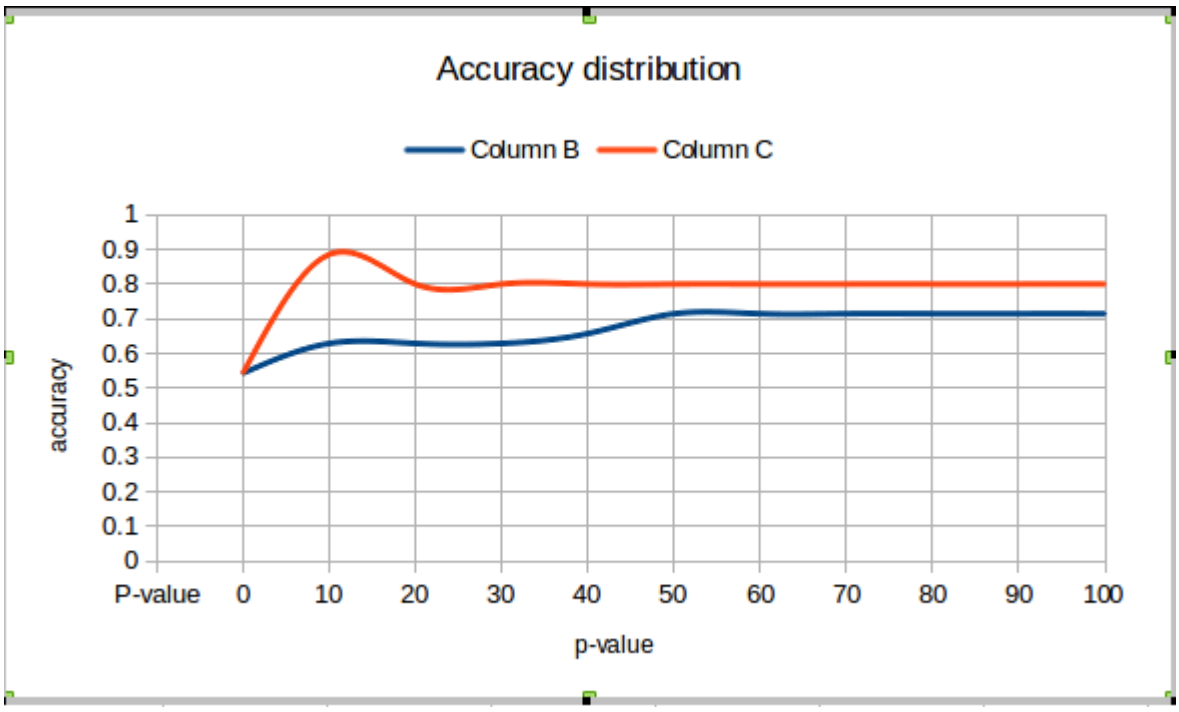
Then I can get the misclassification rate using the following equation.

$$Err(T, D) = \sum_{t \in leaves(T)} \frac{|D(t)|}{|D|} \cdot Err(t, D(t))$$

When I get misclassification rate, I can calculate the accuracy of my algorithm.

2. Accuracy under different setting:

I tried 2 kinds of splitting method. Attached is the reuslt graph for one try. The blue line is the description of accuracy along with confidence interval for chi-square test in misclassification method. The red line is the description of accuracy along with confidence interval for chi-square test in entropy calculation method. It seems that both methods converges when confidence interval increases.



For misclassification method, the accuracy reaches the maximum, 0.7143, when chi-square test confidence level is set to around 0.5 or higher than 0.5. At constrast, for entropy method, the accuracy

reaches maximum, 0.8857, when confidence level for chi-square testing is set around 0.1.

3. Which option to choose:

In this example, I prefer to using 10% confidence level and using entropy method for splitting the sample list, because I can get a higher accuracy for training samples. For entropy, it calculates whether the contribution of attribute to the classification, so here in this examples, it seems like more accurate compared with mis-classification method. For different confidence interval, entropy method always gets higher accuracy than mis-classification method. So I would prefer entropy to split the samples. The if I choose low confidence interval, it can keep splitting as possible as it can. So the decision tree is much more detailed, which may improve the accuracy.

4.How to improve accuracy:

To improve accuracy, I tried to add a new attribute called highpercentageacid, which records down the acid with highest frequency in the sequence. But the result shows that the accuracy is almost the same as before. In my opinion, that attribute is not significant or has low entropy, so it is filtered out all the time until the child list is pure. So that attribute is not used for splitting at all. Therefore, I should get the same result. So I drop that attribute and keep the original 57 attributes.

For the other way, I tried to add a new attribute called maximum acid percentage, which record down the acid subtype with maximum percentage. One acid subtype is c and g, the other one is a and t. And I got the same result as before. It is 0.8. So I give up this method.