CSL603 Machine Learning
LAB 1

Abhishek Chowdhry
2015CSB1002

September 2017

Introduction

In this lab we are trying to predict the sentiment of a movie review. We will be implementing ID3 algorithm to do so. We fill first use some training data to train the classifier and then we will use this classifier to classify some testing data and we will check its accuracy. Our decision tree will use some general words appearing in movie reviews as attributes. We will be using several methods to increase the accuracy of our classifier like early stopping, pruning, decision forests, etc. We will also see the effects of adding noise to the training data set on the accuracy of the decision tree on the testing data set. We will be using the Large Movie Review data set from Stanford for running our experiments.

1 Preprocessing

1.1 Sampling of Data

The data set given to us is large and so we have generated testing, validation and training data sets each containing 1000 observations having equal number of positive and negative instances. In this way we will be able to reproduce the results of our experiments. The training and the validation data sets have been generated from the instances contained in the train directory by random sampling such that no instance is chosen twice. The testing data set has been generated from the instances contained in the train directory through a similar method.

1.2 Selecting attributes

The total number of attributes(words) given to us in the data is large so we have taken a subset of these attributes(first 5000 words in the file) for our purpose. These attributes have been saved in a file in the preprocessing step and for all the further experiments these set of words have been used.

2 Learning a Decision Tree

In this part, we implemented the ID3 algorithm and used it to generate a decision tree on the training set generated in the preprocessing step. For constructing the tree, at every node we choose that attribute from the remaining attributes which gives the maximum information gain. Then the instances in which the selected attribute is present one or more times are pushed onto the left child of the node and the instances in which the selected attribute is present zero number of times is pushed onto the right child. After the decision tree was created, we used it to classify the instances of the testing set and check the accuracy of our decision tree. The results are given below:

Number of nodes in the tree: **251**
Number of terminal nodes in the tree: **126**
Height of the generated tree: **44**
Accuracy of the decision tree on the training data set: **100%**
Accuracy of the decision tree on the testing data set: **66.2%**

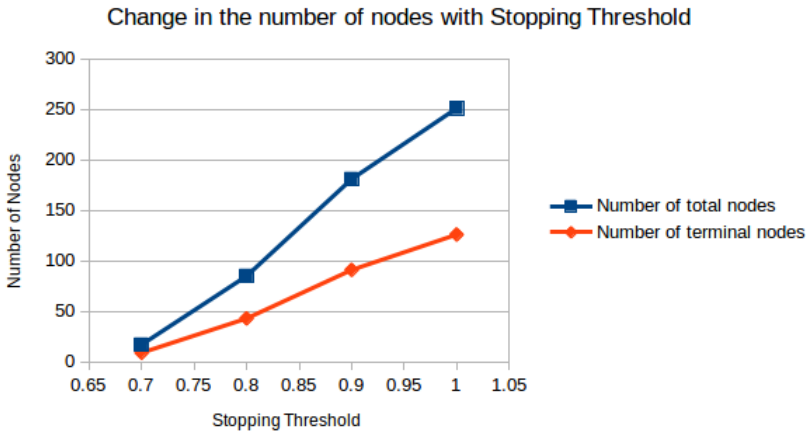
The above data shows the height of the tree and the number of nodes present in the tree. The accuracy of the learned decision tree on the training set is 100% which is justifiable because the training set was used to create the decision tree. It can also be seen that the accuracy of the decision tree on the testing set is less than 100%. This is also justifiable because the decision tree tries to predict the label of a new instance on the basis of the instances through which it has been trained(training set) and so it cannot be always 100% accurate.

2.1 Effects of Early Stopping

We tried early stopping on the tree by labeling a node and making it a leaf as soon as the the probability of the negative/positive instances at that node became greater than a given threshold. On using different threshold for early stopping different observations were made. Some of them are given below.

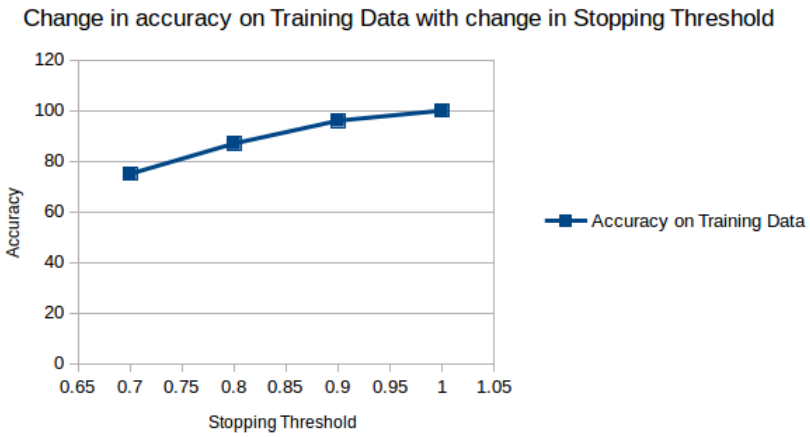
2.1.1 Number of terminal nodes in the Decision tree

As we decreased the threshold for early stopping, the number of terminal nodes in the created decision tree decreased. This is because if the early stopping threshold would have been higher, then the nodes at which we made a leaf(due to early stopping) would have given rise to more leaf nodes(at least one leaf node) instead of just one. So the number of nodes would have been greater than or equal to the current number of nodes if the stopping threshold was higher. This observation can be seen in the graph given below:



2.1.2 Accuracy on the training set

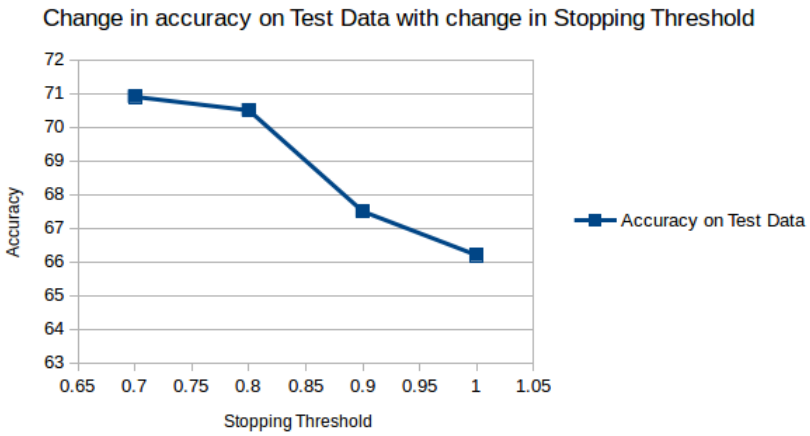
As we decrease the threshold for early stopping, the accuracy on the training set decreases. This is because the nodes at which we make a leaf due to early stopping contain both positive and negative instances. But if the probability of positives/negatives is greater than a given threshold, we assign it a given label(positive/negative). So some of the instances in this node are assigned a wrong label. As a result, on searching for those instances(which are a part of the training set), we get the wrong label. Due to this reason accuracy on the training set decreases as the stopping threshold decreases. This can be seen in the graph given below:



2.1.3 Accuracy on the test set

As we decrease the threshold for early stopping, the accuracy on the test set **generally** increases. This is because early stopping prevents overfitting over the the training data. As the stopping threshold decreases, the extent of early stopping

increases and hence extent of overfitting decreases. As a result accuracy on the test set increases. This can be seen in the graph given below:



2.2 Frequently Used Attributes

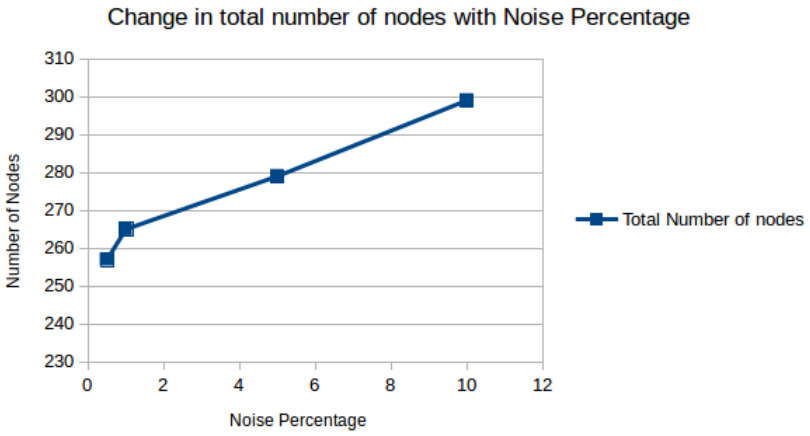
After constructing four trees with four different stopping thresholds, we found out the most commonly used attributes by doing a BFS over all the four trees and counting the number of occurrences of different attributes. Then we printed a list of most commonly used attributes in the decreasing order of their number of occurrences in all the four trees combined.

3 Adding Noise

In this part we tried to analyze the effects of adding noise to the training data set before creating the decision tree. Noise was added by switching the labels of a selected percentage of randomly chosen instances in the training data set. We corrupted training data sets by adding different percentages of noises and analyzed them.

3.1 Number of nodes in the Decision Tree

As the percentage of noise was increased, we observed that the number of nodes in the learned decision tree increased. This could be due to the fact that noise can confuse learning algorithms resulting the formation of a long and complex decision tree. This happens because the algorithm tries to fit every training instance(including noisy instances) into the model description. As a result we get increased number of total nodes in the learned decision tree. This can be seen in the graph below:



From the data we obtained(given below) we also observed that the addition of noise generally decreases the accuracy on the test set. This may not always be true. But in our case the addition of noise mostly decreases the accuracy of the decision tree on the test set.

Noise(%)	0.5	1.0	5.0	10.0
Accuracy	66.6	64.7	64.2	62.7

4 Post Pruning

In this part we first made the decision tree using the instances in the training set and found its accuracy on the testing set. We then used post pruning technique using the validation set and pruned the tree recursively to increase its accuracy on the validation set. Then we used the new pruned decision tree and found out its accuracy on the testing set. The results are given below:

4.1 Tree Before Pruning

Accuracy of the decision tree on the training data set: **100%**
Accuracy of the decision tree on the validation data set: **68.7%**
Accuracy of the decision tree on the testing data set: **66.2%**

4.2 Tree After Pruning

Accuracy of the decision tree on the training data set: **80.7%**
Accuracy of the decision tree on the validation data set: **73.4%**
Accuracy of the decision tree on the testing data set: **73.2%**

Form the above observations we can see that **the accuracy of the tree on the validation set increases** after pruning. This is obvious because we when we start pruning the tree recursively, we prune the tree only if its accuracy on the validation set increases.

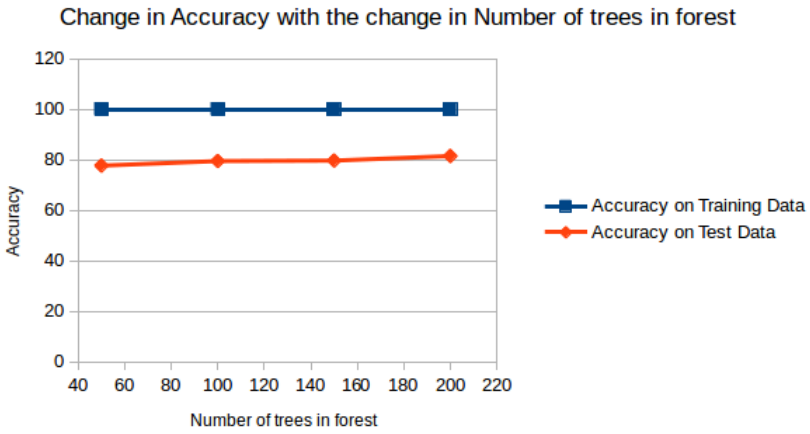
We also observe that **the accuracy of the tree on the training set decreases**. This can be explained by the fact that when we prune a node, it contains both positive and negative instances from the training set. But after pruning we assign it a given label(positive/negative) based on the majority instances on that node. So some of the instances in this node are assigned a wrong label. As a result, on searching for those instances(which are a part of the training set), we get the wrong label. Due to this reason accuracy of the tree on the training set decreases after pruning the tree with the validation set.

From the data, we also observe that t**he accuracy of the tree increases on the testing set aft**er pruning. This is generally true because pruning reduces the overfitting on the training set thereby increasing the accuracy of the tree on the testing set.

We also observe that **pruning reduces the number of nodes in the tree**. This is obvious because we are cutting a part of the tree.

5 Decision Forest

In this part instead of generating one decision tree, we generated an entire forest(collection of trees). For each tree of the forest we randomly chose a set of 500 attributes from the 5000 attributes we selected in the preprocessing step. Then we used these 500 attributes and the entire training set to produce one tree. We produces multiple trees and then we tested the accuracy of this forest on the training set and the testing set. The label predicted by the entire forest is the majority label of all the labels predicted by all the trees in the forest. We increased the number of trees in the forests and the calculated the accuracy of the forest over the training and the testing set and the results are given in the graph below:



We can observe from the graph that the as the number of trees in the decision forest increase, the accuracy of the forest over the testing data set also increases. This is justifiable because decision forests help to minimize overfitting over averaging over multiple trees. So as the number of trees increases, the overfitting on the training set decreases, and as a result, in the **general** case the accuracy of the forest over the testing data increases.