SYDDANSK UNIVERSITET

Random Forests and Decision Trees

Bjarki Sigurdsson
Abdulrahman Abdulrahim
Miguel de la Colina
Group 5

May 2, 2017

ABSTRACT

THE PURPOSE OF THE EXERCISE WAS TO GET FAMILIAR WITH RANDOM FORESTS AND DECISION TREES AND MANAGE TO GET A GRASP OF HOW THEY WORK AND HOW THEY FUNCTION WHEN USING REAL DATA-SETS IN THIS CASE THE CIPHERS.

1 DECISION TREES

In this section we will be working exclusively with decision trees finding optimal decision point, computing and visualizing one and also doing cross validation with the results that we manage to obtain.

1.1 OPTIMAL DECISION POINT

When using decision trees for classification with continuous variables, the common method involves, for each attribute, a threshold which maximizes information gain. Thresholding is a binary operation and thus does not generalize simply to multiclass problems such as this, but one workaround is to train multiple classifiers. Each of these is trained to classify a single class in a one vs. all manner. In computing the following results, we create a classifier for the 1 digit by computing the optimal decision point for each of the first five principal components. The method used for computing the threshold is simple: We sort on the current PC and iterate through the samples, testing each value as a threshold and computing the corresponding information gain.

Figure 1 shows information gain as a function of data matrix row number for each of the five PCs. The thresholds chosen were the maxima of these graphs, 0.1136, 0.0463, 0.0298, 0.0457 and 0.0726 for components 1-5, respectively. Note that some of these gains are greater than those for preceding PCs. This shows the influence of specific PCs on the classification of the 1 digit and varies for other classes.

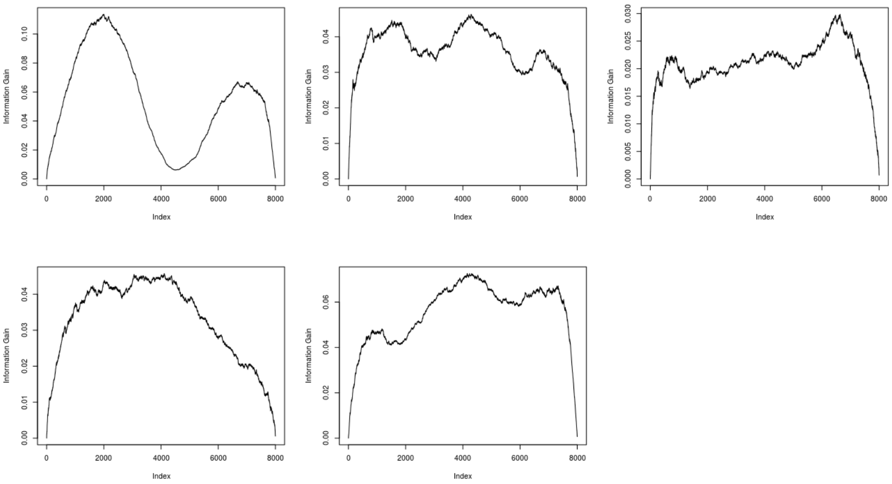


Figure 1: Decision Tree of person dependent set.

1.2 Compute Decision Tree

In the second part of the exercise we used the person dependent data-set and split it into training and test sets and using rpart in r we created a model for a decision tree and as we can see in figure 1 that is the outcome that came from the actual decision tree once we tested the test set we actually got an accuracy of .5042 and as we can see from the confusion matrix in figure 2 the diagonal is where most of the results are showing that it had pretty reasonable result accuracy.

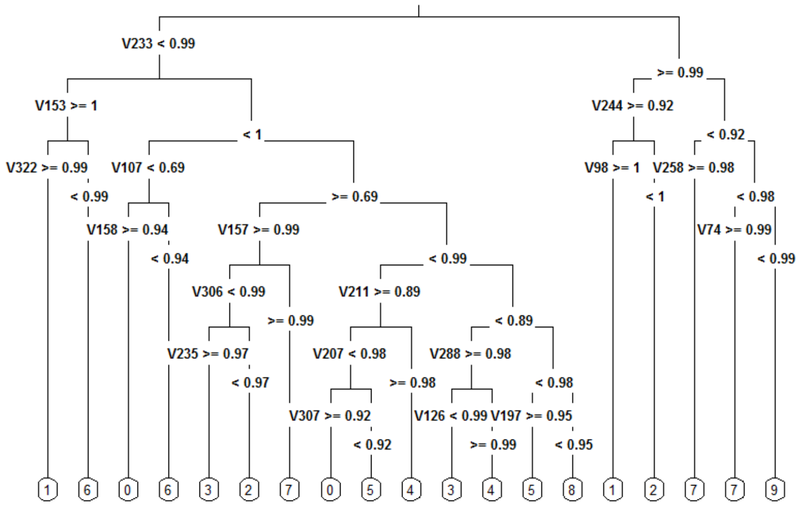
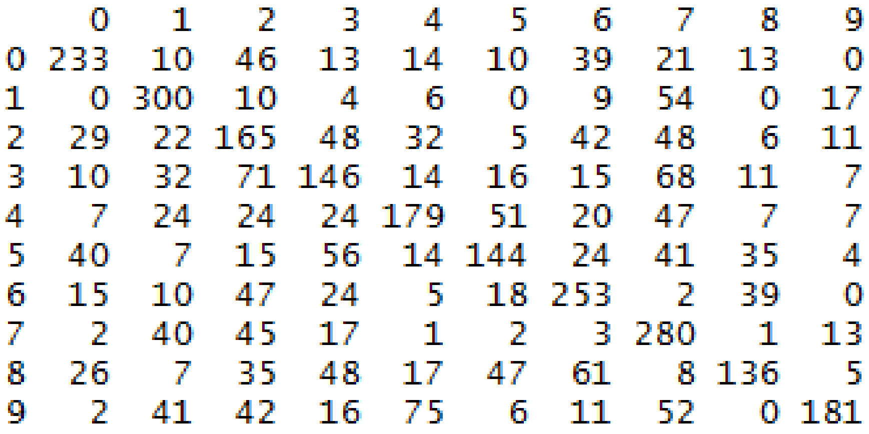


Figure 2: Decision Tree of person dependent set.



```
     0   1   2   3   4   5   6   7   8   9
0  233  10  46  13  14  10  39  21  13   0
1    0 300  10   4   6   0   9  54   0  17
2   29  22 165  48  32   5  42  48   6  11
3   10  32  71 146  14  16  15  68  11   7
4    7  24  24  24 179  51  20  47   7   7
5   40   7  15  56  14 144  24  41  35   4
6   15  10  47  24   5  18 253   2  39   0
7    2  40  45  17   1   2   3 280   1  13
8   26   7  35  48  17  47  61   8 136   5
9    2  41  42  16  75   6  11  52   0 181
```

Figure 3: Confusion matrix of test set results and actual classes.

1.3 Cross-Validation

Now we made cross validation for the dependent set using 10% as test set and the other 90% as training and we can see on figure 3 that most of the results are around .53 showing a very consistent trend no matter which test and training sets we are using

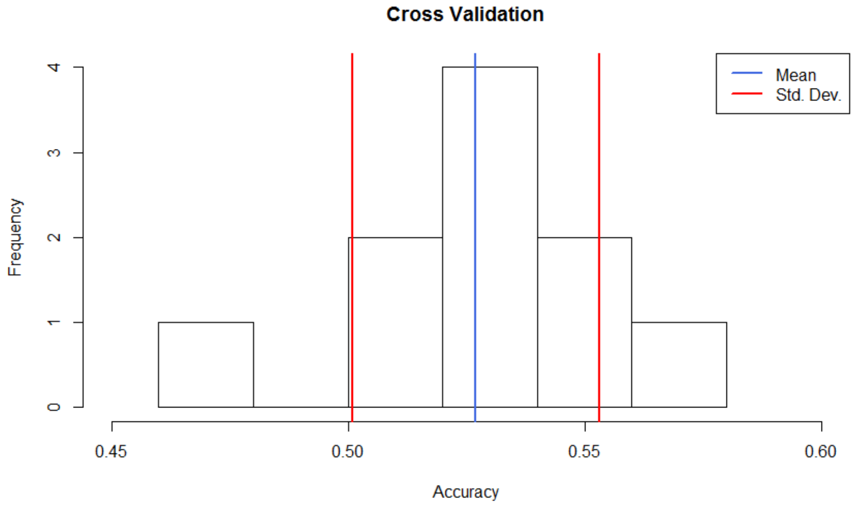


Figure 4: Decision Tree of person dependent set.

2 Random Forests

Random forests are robust classifiers which use the majority vote of multiple decision trees for classification. This method uses bootstrap aggregation to circumvent the overfitting inherent to decision trees.

There are two main parameters to tune when constructing a decision tree, namely the total number of trees and their depth. A higher total number of trees tends to reduce overall classification error with diminishing results. In other words, adding trees reduces the error but the error will eventually converge. The tree depth contributes to the classifier's level of overfitting. Deeper trees tend to overfit data so if the data is noisy, then pruning is recommended and often implemented by simply limiting the tree depth.