Data Analysis Assignment 2

Predicting activity from smartphone data

Introduction

A study [1] was carried out to record detailed data from Samsung smartphones that were being worn by test subjects undertaking various activities (walking, walking upstairs, walking downstairs, sitting, standing, and laying). Details of the data set and the study are published on the Machine Learning Repository website [2].

The purpose of this analysis is to build a model for prediction of what kind of activity a subject is performing based on the quantitative measurements from the phone.

Methods

Data Collection

The data for this analysis were downloaded from the Coursera Data Analysis file share [3], on 1 March 2013.

Initial Exploration

Exploratory analysis was performed by examining tables and plots of the observed data, but due to the large number of variables it was decided to uncover the most important variables via automated processes, rather than by visually inspecting plots.

The data set consists of 7352 observations of 563 different variables. One of these variables, “subject”, denotes which of the 21 subjects the measurements are from. Another variable, “activity”, denotes the type of activity the subject is undertaking whilst the phone is recording measurements. The remaining 561 variables are quantitative measurements recorded by the phones’ accelerometers and gyroscopes. The readme.txt [4] accompanying the full data set describes the measurements as being normalized and bounded between -1 and 1.

From exploratory analysis it was determined that there were no missing values and it was confirmed the quantitative variables lay between -1 and 1. There were no data that looked to be obvious mistakes, but it was difficult to ascertain what constituted abnormal data in this context. What was slightly unusual was the absence of particular subjects. The original study involved 30 subjects, but the data set only contained 21. The variable names in the data set were transformed using the `make.names()` function because they include dashes and brackets which can sometimes confuse R functions.

Study Design

The data set was divided up into three subsets. The Test set consisted of the data from 7 subjects (including subjects 27, 28, 29, and 30, as stipulated in the assignment prompt), and was set aside until the testing of the final model. The Training set consisted of data from 7 subjects (including subjects 1, 3, 5, and 6, as stipulated in the assignment prompt). The Validation set consisted of the remaining 7 subjects and was used to cross validate models generated from the training data.

The error rate for the assessment of model effectiveness was chosen in advance to be the misclassification rate, i.e. the number of times the model failed to classify an observation correctly, divided by the overall number of predictions the model made.

Model Building

The initial step was to build a classification tree [5], supplying all variables to the algorithm. The following 9 variables were initially identified as producing the best splits for the tree using the defaults:

"fBodyAccJerk.std...X", "tGravityAcc.mean...X", "tGravityAcc.mean...Y", "tGravityAcc.max...Y", "tBodyAcc.max...X", "tGravityAcc.arCoeff...Y.1", "tGravityAcc.min...X", "tBodyAccJerk.max...X"

One of the benefits of using a classification tree is that the results are readily interpretable, for example it can be seen which are the important variables and how they affect the decision tree. Figure 1 shows a visualisation of the tree. A larger version of Figure 1 is also supplied separately.

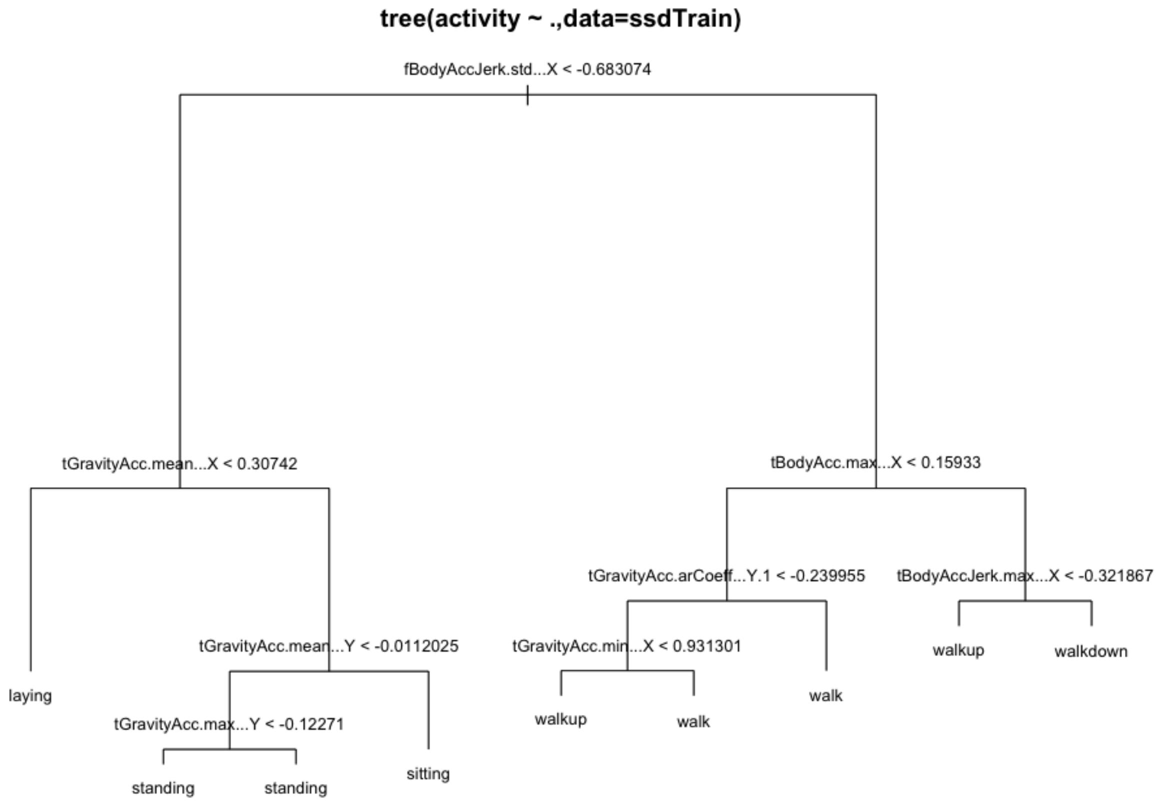

tree(activity ~ .,data=ssdTrain)
fBodyAccJerk.std...X < -0.683074
tGravityAcc.mean...X < 0.30742
tBodyAcc.max...X < 0.15933
tGravityAcc.arCoeff...Y.1 < -0.239955
tBodyAccJerk.max...X < -0.321867
walkup
walkdown
tGravityAcc.mean...Y < -0.0112025
tGravityAcc.min...X < 0.931301
laying
walk
tGravityAcc.max...Y < -0.12271
walkup
walk
sitting
standing
standing

Figure 1. Tree diagram for initial tree classification using all variables.

It is interesting to note that no variables with a Z-axis component featured in the model, and also none of the “bandsEnergy” variables from the gyroscope were considered important.

The misclassification rate for the training set was quite good, at **6.8%** (151 misclassified out of 2220). However, when the model was applied to the validation set the misclassification rate rose to **19.0%** (471 misclassified out of 2474), suggesting that the model is overfitting the training set.

However, with so many variables, it is difficult to work with a simple tree model such as this, and reducing the model to just using 9 of the 561 variables is throwing away potentially useful information. A single tree is susceptible to overfitting the training data, and it would take lots of manual effort to prune or modify the tree with so many variables. For these reasons the next step was to investigate using Random Forests [6] to find a more accurate model.

Random Forest Model

A model was built using all the variables in the training set. However, because of the bootstrapping techniques used in construction of the random forest model, the usual validation dataset was not needed, and it was decided to combine the Training and Validation data sets together in order to produce the model (training on 14 subjects rather than 7).

The code used was as follows:

```
rf.trainVal <- randomForest(activity ~ ., data=ssdTrainVal)
```

The resultant model produced a misclassification rate of **1.4%** (66 misclassified out of 4694).

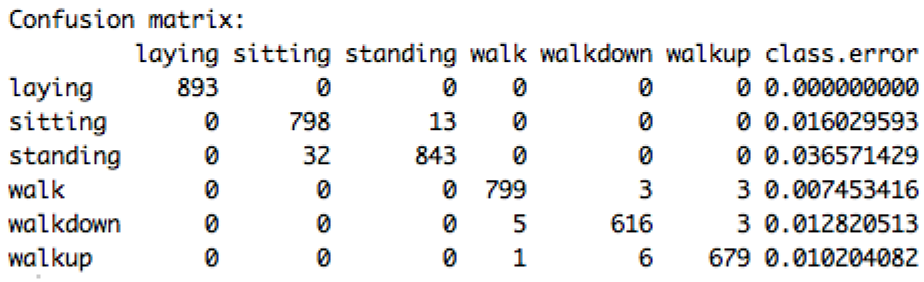
Results

The final model was then applied to the Test dataset, which had been held back from building and validating the model until the end. The final misclassification rate was **9.0%** (471 misclassified out of 2658 observations).

The analysis shows that random forests can produced a moderately accurate prediction model without a lot of investigation and configuration, and that it out-performs a single tree approach (at least without significant modification), particularly when predicting outcomes for new data.

Conclusions

The analysis shows that with relatively little investment in time, a random forest approach to this problem produced a pretty good predictive model (accurately predicting on the test set 91% of the time). The main area where the model fell short was in differentiating between the subject sitting or standing, which is understandable given the relatively small differences in movement a person would make during these activities. The confusion matrix from the training data illustrates this, with 45 misclassifications between both activities.



```
Confusion matrix:
         laying sitting standing walk walkdown walkup class.error
laying      893       0        0    0        0      0 0.000000000
sitting       0     798       13    0        0      0 0.016029593
standing      0      32      843    0        0      0 0.036571429
walk          0       0        0  799        3      3 0.007453416
walkdown      0       0        0    5      616      3 0.012820513
walkup        0       0        0    1        6    679 0.010204082
```

Figure 2. Confusion matrix from random forest training data, showing misclassifications between sitting and standing.

Given more time it would have been interesting to investigate ways of differentiating between the activities that caused trouble for the model, perhaps combining multiple models to overcome this.

References

1. Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz. Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine. International Workshop of Ambient Assisted Living (IWAAL 2012). Vitoria-Gasteiz, Spain. Dec 2012

2. UCI Machine Learning Repository study website, URL: http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones

3. Coursera Assignment 2 source data. URL: https://spark-public.s3.amazonaws.com/dataanalysis/samsungData.rda

4. Full study data, including readme.txt. URL: http://archive.ics.uci.edu/ml/machine-learning-databases/00240/UCI%20HAR%20Dataset.zip

5. Classification and regression trees. Breiman L., Friedman J. H., Olshen R. A., and Stone, C. J. (1984) Classification and Regression Trees. Wadsworth.

6. Random Forests. Breiman, L. (2001), Random Forests, Machine Learning 45(1), 5-32.