Title: Predicting Subject Activity using Accelerometer and Gyroscopic Data from Samsung Galaxy II Smartphones

Introduction:

Accelerometers and Gyroscopes have recently been used in research studies to monitor daily activities in human subjects. These types of studies have attracted much interest in recent years in the fields such as the fitness field or health care. An interesting application is the study of human activities using motion sensors embedded in the Samsung Galaxy II smartphone mounted on a subject's waist. In this study, 21 subjects performed six different activities: Laying, Sitting, Standing, Walking, Walking downstairs and Walking upstairs. Accelerometer and gyroscopic data were recorded from the smartphone for each subject. The experiment was also video-taped to label the type of activity performed manually. The goal of this analysis is to build a prediction model that can predict a subject's activity based on the quantitative measurements recorded by the smartphone.

Methods:

Data collection

The dataset consists of recordings from 21 subjects performing six different activities while wearing a smartphone on their waist. The two motion sensors recorded a total of 561 measurement variables for the six different activities totaling 7,352 observations (including the activity type variable and subject variable). The 561 measurement variables can be categorized in the following "types" of variables:

- tBodyAcc: Linear Body Acceleration
- tGravityAcc: Gravity Acceleration
- tBodyAccJerk: Body Acceleration with Jerk Signal
- tBodyGyro: Body Gyroscope
- tBodyGyroJerk: Body Gyroscope with Jerk Signal
- fBodyAcc: Fast Fourier Transform (FFT) Linear Body Acceleration
- fBodyAccJerk: FFT Linear Body Acceleration with Jerk Signal
- fBodyGyro: FFT Body Gyroscope

For each of these types of variables, the dataset included the estimated variables: mean value **(mean)**, standard deviation **(std)**, median absolute deviation **(mad)**, largest value in array **(max)**, smallest value in array **(min)**, signal magnitude area **(sma)**, energy measure **(energy)**, interquartile range **(iqr)**, signal entropy **(entropy)**, auto-regression coefficients with Burg order equal to 4 **(arCoeff)**, correlation coefficient between two signals **(correlation)** in 3-axis X, Y, Z and their magnitudes. For the variables fBodyAcc, fBodyAccJerk, fBodyGyro, the dataset provided values for the index of the frequency component with largest magnitude **(maxInds)**, weighted average of the frequency components to obtain a mean frequency, **(meanFreq)**, skewness of the frequency domain signal **(skewness),** kurtosis of the frequency domain signal **(kurtosis)**, bandsenergy of a frequency interval within the 64 bins of the FFT of each window **(bandsEnergy).**

The data was normalized and bound between [-1,1]. Because of this normalization, the values of all variables are dimensionless quantities.

The dataset, in form of an *.rda file, was downloaded from the Data Analysis course website on **December 2, 2013** using the R programming language [1].

In predictive modeling analysis, it is recommended to partition [2] the dataset into non-overlapping Train and Test Sets. The split was performed on the basis of the subject ID. The Train Set contained subjects 1, 3, 5, and 6 with a total of 1315 observations and the Test Set contained subjects 27, 28, 29 and 30 with a total of 1485 observations (minimum requirements for this assignment).

Exploratory Analysis

Exploratory analysis was performed by building various predictive models and assessing the performance of each model by evaluating performance metrics like Residual Mean Deviance, Misclassification Error Rate, Accuracy, Sensitivity and Specificity. We then determined the model that can best predict the outcome, and which corresponding measurement variables (predictors) from the dataset are most important to predict outcome.

There were no missing data found in the dataset. A few transformations were performed on the downloaded dataset: a) the variable "Activity" was changed from character type to factor type, and b) the dataset was transformed into a dataframe objet to eliminate duplicate names and illegal characters in R.

Statistical Modeling

The overall performance of a model is based on its ability to correctly predict the outcome. A popular tool for evaluating the performance of a model is the Confusion Matrix [3]. The Confusion Matrix (from the "caret" package in R) visualizes how a model is predicting the outcome (the subject's activities) by calculating the error rates, i.e. correct classifications (True positive/True negative) and incorrect classifications (False Positive/True Negative) [3]. The following statistical performance measures can then be evaluated from these results:

- **Accuracy** = sum of the correct classifications/total number of classifications, describes the overall correctness of the model [3].
- **Sensitivity** = true positive rate, measures the proportion of actual positives which are correctly identified as such [4].
- **Specificity** = true negative rate, measures the proportion of negatives which are correctly identified as such [4].

The model with the optimum Accuracy, Sensitivity and Specificity will be considered the best model to predict the outcome. In practical, the predictive model is first built on the Train Set, then its true performance evaluated on the Test Set.

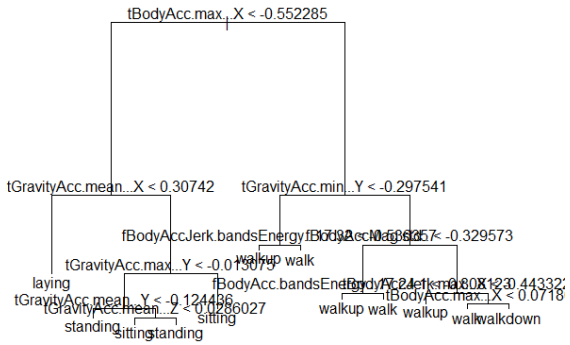
Results:

1. Model using Logistic Regression to predict subject activity as outcome:

A model was constructed using a Logistic Regression function including all variables in the Train Set to predict the outcome variable Activity. The algorithm did not converge due to the large number of variables in the model.

2. Model using Classification Tree to predict subject activity as outcome:

Next, a model was constructed using a Classification Tree to predict the outcome Activity. The Tree model was built on the Train Set including all variables. The result of the fit is shown in **Figure 1** below (**Figure 1 + caption** also available in separate document):



The result shows a Tree with 12 terminal nodes and the following 10 most important variables used in the Tree construction:

1	"tBodyAcc.max...X"
2	"tGravityAcc.max...Y"
3	"tGravityAcc.mean...Z"
4	"fBodyAccJerk.bandsEnergy...17.32"
5	"fBodyAcc.bandsEnergy...17.24.1"
6	"tGravityAcc.mean...X"
7	"tGravityAcc.mean...Y"
8	"tGravityAcc.min...Y"
9	"fBodyAccMag.std.."
10	"tBodyAccJerk.max...X"

The summary of the Tree model gives:

- Residual Mean Deviance (measure of fit quality) = 0.157
- Misclassification Error Rate = 0.023.

- Calculated Accuracy of the Tree model <u>on the Train Set</u> = **97.7%.**

Such a high accuracy for a model is not surprising since our model was optimized to the Train Set. The Confusion Matrix for the Tree model on the Test Set gives the following performance results:

Accuracy	0.8195
95% CI	(0.799,0.838)
p value	< 2.2e -16

Statistics by class

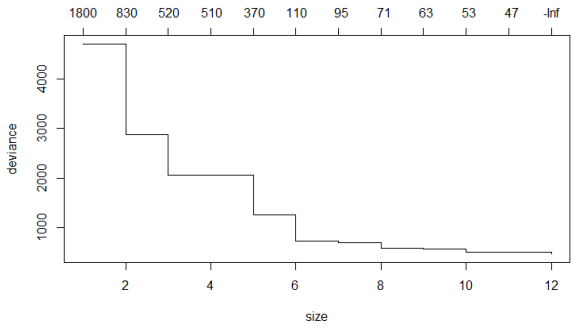
Class:	laying	siting	standing	walk	walkdow	walkup
Sensitivity	1	0.724	0.7105	0.9223	0.7863	0.7707
Specificity	1	0.9291	0.9433	0.9695	0.9872	0.9547

The Accuracy of our Tree model on the Test Set is only ~**82%.** This drop in performance is possibly due to over fitting, or using too small a dataset, and/or using too many correlated variables. Based on the Sensitivity and Specificity measures, our Tree model can best predict the activities "laying" and "walk". Our model also predicts that the variable "tBodyAccMax..X" accounts for more than 50% of the variability in predicting the outcome.

To further understand the drop in performance of our model, we applied the technique of "pruning" [5] to our Tree, to see if reducing the size of the Tree might improve the accuracy of the model. We tried pruning our original Tree with 12 nodes to a Tree with 9 nodes. The results of the Pruned Tree on the Train Set gave a Residual Mean Deviance equal to 0.282 and a Misclassification Error Rate equal to 0.033.

When applied on the Test Set, the Accuracy of the Pruned Tree model is only ~**78%**. The lower performance of the Pruned Tree over the full Tree suggests that the original 12 nodes Tree was not too large (i.e. did not over fit) and a smaller size Tree probably does not capture all the complexity in the dataset.

Cross-validation is another technique that can help us understand the poor performance of our original Tree model. Cross-validation predicts the fit of a model to a hypothetical data set which is different from the dataset used to build the model [6]. Using the cv.tree() function in R, we can estimate the deviance measure between various model sizes. The results from the cross-validation test is shown in the Figure below:



You can see that as the model increases in size, the deviance decreases and there is no significant deviance for a 9 nodes Tree (or Pruned Tree) vs a 12 nodes Tree (original Tree). Consequently, our original 12 nodes Tree was not too large. Further testing is needed to determine if the size of the Train and/or Test Sets's could impact the performance of our Tree model.

3. Model using Random Forest to predict subject activity as outcome:

In order to improve the accuracy of our predictive model, we tried to build a model using the technique called "Random Forest" [7] to predict the outcome. Random Forest is a technique known to be accurate, easy to use and robust. The Random Forest model was first built on the Train Set using the default randomForest() function from the random Forest package in R. When applied on the Test Set, the Random Forest model gave the following performance results:

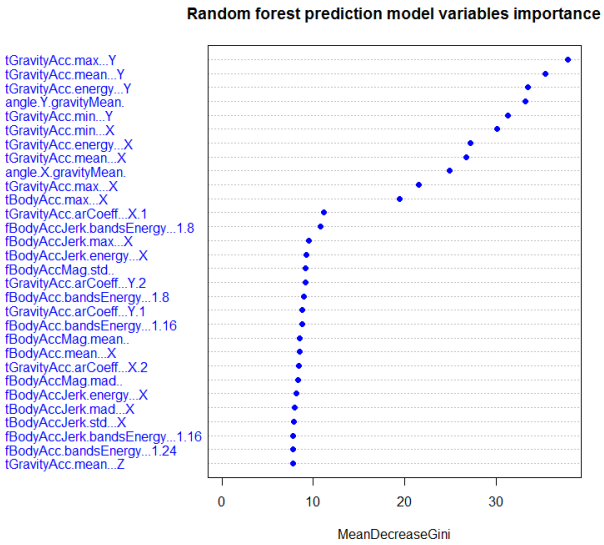
Accuracy	0.928
95% CI	(0.9143,0.9412)
p value	< 2.2e -16

Statistics by class

Class:	laying	siting	standing	walk	walkdow	walkup
Sensitivity	1	0.8819	0.8377	0.9955	0.9061	0.97
Specificity	1	0.9675	0.9746	0.9945	0.9945	0.9829

The Accuracy for our Random Forest model is ~**93%,** an increase of more than 10% over the Tree model. Based on the Sensitivity and Specificity measures, our Random Forest model can best predict the activities "laying" and "walk".

The variables of most importance from the Random Forest model are given in **Figure 2** below (**Figure 2 + caption** also available in separate document)



We can see that the variables of most importance derived from the Random Forest model are different than the ones obtained with the Tree model. This is not surprising since Random Forest techniques model the data using an ensemble of Trees that are calculated on random subsets of data using random selected predictors for each split [8]. This allows the model to better evaluate the contribution and importance of each predictor in determining the outcome. The consequence is an overall more accurate model.

4. Confounders

When using the Random Forest model in section 3, all arguments for that function were set to default. As a consequence, the number of variables randomly sampled as candidates at each split, (argument 'mtry' in randomForest() function) is the square root of the number of variables in the dataset, i.e. sqrt(561)=23. We found that reducing the mtry number from 23 to 15 further improved the accuracy of our RandomForest model to **93.5%** (~1% increase)**,** which seems to suggest there may be correlation among the variables and/or confounding variables in the dataset, and eliminating some of these can further increase the overall accuracy of the model [9]. Further testing is required to determine which variables in particular are confounding variables in this particular dataset.

Conclusions

The goal of this analysis was to build a predictive model that can predict what activity a subject is performing based on data recorded by motion sensors embedded in a smartphone. The predictive model's performance was evaluated using Mean Residual Deviance, Misclassification Error Rate and Accuracy. A subset of the dataset was partitioned into non overlapping Train Set and Test Set. The model was first built on the Train Set and later evaluated against the Test Set. Our analysis focused on 3 types of predictive models: Logistic Regression model, Classification Tree model and Random Forest Model. A comparison of the performance of each model against

the Test Set showed that the Random Forest model (Accuracy ~ 93%) outperformed the Tree model (Accuracy~78%) in predicting the outcome "activity". Based on Sensitivity and Specificity measures, we found that the Random Forest model can best predict the activities "laying" and "walk". We also found that the accuracy of the Random Forest model can further be improved by reducing the amount of correlation or confounding variables in the model. Correlation, size and subsetting of the dataset appear to be the biggest factors affecting and limiting the predictive accuracy of our models.

References

[1] http://www.r-project.org/.

[2] http://www.statistics.com/index.php?page=glossary&term_id=751.

[3] http://en.wikipedia.org/wiki/Confusion_matrix.

[4] http://en.wikipedia.org/wiki/Sensitivity_and_specificity.

[5] http://en.wikipedia.org/wiki/Pruning_%28decision_trees%29.

[6] http://en.wikipedia.org/wiki/Cross-validation_%28statistics%29.

[7] http://en.wikipedia.org/wiki/Random_forest.

[8] http://www.stanford.edu/~stephsus/R-randomforest-guide.pdf.

[9] http://en.wikipedia.org/wiki/Confounding.