Human Activity Recognition Using Smartphone Data

Introduction:

Samsung Galaxy S II is a Smartphone with built in accelerometer (Engadget, 2013) and gyroscope (mobile88, 2013). In this study, I used "Human Activity Recognition Using Smartphones Dataset" (UCI, 2013) to build a model. I want our model to recognize the type of activity (walking, walking upstairs, walking downstairs, sitting, standing, laying) the person was doing based on his/her 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz, as recorded by the accelerometer and gyroscope he/she is wearing on the wrist.

Methods:

Data Collection

The "Human Activity Recognition Using Smartphones Dataset" consisted of "experiments (that) have been carried out with a group of 30 volunteers within an age bracket of 19-48 years. Each person performed six activities (walking, walking upstairs, walking downstairs, sitting, standing, laying) wearing a Smartphone (Samsung Galaxy S II) on the waist" (UCI, 2013). However, we would only be using a subset of the data, consisting of data from subject 1,3,5,6,7,8,11,14,15,16,17,19,21,22,23,25-30 (21 subjects in total) Furthermore, "features are normalized and bounded within [-1,1]" (UCI, 2013). This meant that we would not expect extreme values among the features. The data that we used had been pre processed by Professor Jeff Leek from John Hopkins University to include:

- Subject - which subject was performing the tasks when the measurements were taken.
- Activity - what activity they were performing.

It consisted of 7352 observations with 563 features. In general, the features can be summarized as follow (UCI, 2013):

- Accelerometer and gyroscope 3-axial raw signals tAcc-XYZ and tGyro-XYZ. These time domain signals (prefix 't' to denote time) were captured at a constant rate of 50 Hz.
- Similarly, the acceleration signal was then separated into body and gravity acceleration signals (tBodyAcc-XYZ and tGravityAcc-XYZ).
- Subsequently, the body linear acceleration and angular velocity were derived in time to obtain Jerk signals (tBodyAccJerk-XYZ and tBodyGyroJerk-XYZ). Also, the magnitude of these three-dimensional signals was calculated using the Euclidean norm (tBodyAccMag, tGravityAccMag, tBodyAccJerkMag, tBodyGyroMag, tBodyGyroJerkMag).

In addition, the set of features that were estimated from these signals were (UCI, 2013):

- mean(): Mean value
- std(): Standard deviation
- mad(): Median absolute deviation
- max(): Largest value in array
- min(): Smallest value in array
- sma(): Signal magnitude area
- energy(): Energy measure. Sum of the squares divided by the number of values.
- iqr(): Interquartile range
- entropy(): Signal entropy
- arCoeff(): Autorregresion coefficients with Burg order equal to 4
- correlation(): correlation coefficient between two signals
- maxInds(): index of the frequency component with largest magnitude
- meanFreq(): Weighted average of the frequency components to obtain a mean frequency
- skewness(): skewness of the frequency domain signal
- kurtosis(): kurtosis of the frequency domain signal
- bandsEnergy(): Energy of a frequency interval within the 64 bins of the FFT of each window.
- angle(): Angle between to vectors.

Exploratory Analysis

Exploratory analysis was performed by examining tables and plots of the observed data using R (R, R Project, 2013). By observing these tables and plots, I identified transformations to perform on the raw data. Exploratory analysis was used to (1) identify missing values (if any), (2) verify the quality of the data, and (3) determine the features to be used in the model. There were no missing data identified. As the data was already normalized and bounded within [-1,1], there were no extreme values in the data set. I removed all special characters like brackets and commas from the feature name and renamed the features that had the exact same name (mostly the bandsEnergy features) as they might affect the modeling code at a later stage. I also converted the activity and subject features into factor variable to help R in the data modeling stage.

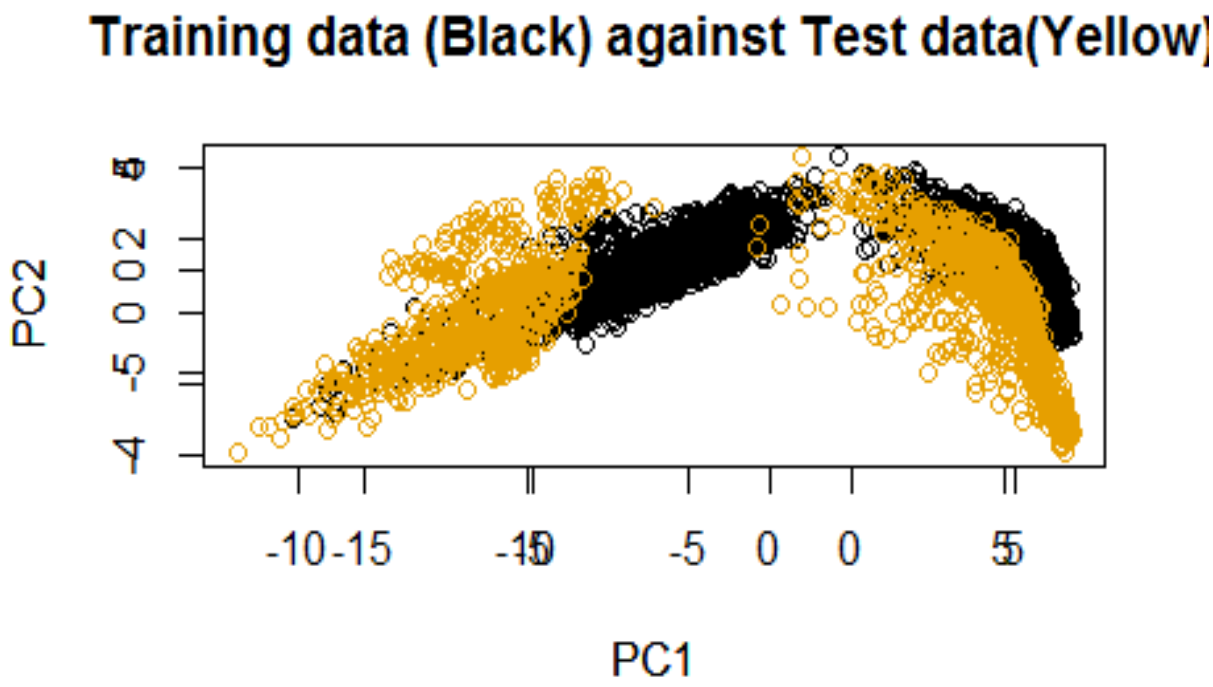
Next, I divided the data into training (subject 1,3,5,6,7,8,11,14,15,16,17,19,21,22,23,25,26) and test data set (subject 27,28,29,30) I noticed that within the training set, subject 25 had the highest observation count of 409, 45% more observations compared to subject 8 which had the lowest observation count of 281. To prevent any individual subjects from skewing the data (in

case some subjects had weird moving habits or their device was not tuned correctly), I did a random sampling across the training data again, this time ensuring every subject in the training data had equal counts of observation, 281. Finally, I further subsetted ¼ of the training data to be validation data via one last random sampling. The final data distribution is as follows:

- Training: 3582 observations, 562 features
- Validation: 1195 observations, 561 features
- Test: 1485 observations, 561 features

One last test I needed to perform was to check if the training data and test data had similar distribution. This is important as most modeling methods like decision trees, K-nearest neighbors etc works best if training and test data have similar distribution. By using Principal Components Analysis (StatSoft, 2013), I summarized the complex data set of over 500 features into 2 principal components that represent about 90% of the data distribution. Fig1 below shows the plot of the first two principal components of training data (black) against first two principal components of test data (yellow). As we can see, the two data sets had similar trends and overlapped each other. This meant that training data and test data had similar distribution.

Fig1: PCA Training Data against Test Data



Statistical Modeling

I decided not to do feature selection and to include as much data as possible to train the model. This is in line with Google engineers' paper on the unreasonable effectiveness of data (Alon Halevy, 2010), which stated that having more data almost always beat having a smarter or more complex algorithm. Since training data and test data had similar distribution, I decided to make use of random forest for our modeling (Kaggle, 2013). Random forest is basically a collection of decision trees (scikit, 2013) and voting is employed to decide the final classification of each data input. If we input test data into a random forest with 100 trees, each tree will work on a subset of the test data and produce different classification. The final classification will be based on the most popular classification across all the 100 trees (hence voting). Random forest was chosen for its simplicity, speed (could be parallel processed across Multicore machines) and reasonable accuracy from other studies (Segal, 2003). I decided that simple, single validation was sufficient and cross validation was not needed as long as I set a large enough number of trees and subsample the data well in the random forest; both techniques should result in a sufficiently unbiased random forest.

To determine the optimal settings for random forest, especially the "mtry" variable which affects the number of sampling features and hence the subsample size of data for each tree to sample, I trained the random forest multiple times with different "mtry" and number of trees. I then compared the accuracy rate of the different random forest models using the validation data and decided that "mtry" = $\sqrt[3]{}$(number of features) with around 2001 trees (odd number of trees help break tie votes) seems to return the best result. By making use of R's foreach (R, foreach, 2013) and doSNOW (R, doSNOW, 2013) libraries for Multicore processing, I was able to model a random forest with 2001 trees in about 3 minutes. I also stored the relative importance of each feature. Importance measures the mean decrease in Gini index (accuracy) of the model should the feature be removed. Since none had totally 0 importance, I decided to keep all the features to maximize accuracy. Table 1 below shows the top 10 features.

Table1: Feature Importance Table

No.	Features	importance
1	tGravityAccminX	54.1553598
2	tGravityAccmeanX	54.1138604
3	tGravityAccmaxX	51.6929082
4	tGravityAccenergyX	51.5891865
5	angleXgravityMean	51.2201409
6	angleYgravityMean	49.0953730
7	tGravityAccmaxY	49.0706419
8	tGravityAccmeanY	47.5035991
9	tGravityAccminY	46.8805115
10	tGravityAccenergyY	35.6867821

Reproducibility

All analyses performed in this manuscript could be reproduced using the R markdown file smartphoneAnalysis.Rmd (RStudio, 2013) (available on request). To reproduce the exact results presented in this manuscript, the analysis must be performed on the same data set (available on request).

Benchmark

Our baseline of comparison was a random model, where we randomly assigned a class label to the test data set. The random model had an accuracy rate of 14.81% and we would expect our random forest model to perform much better.

Results:

Table2: Confusion Matrix

	actualValues					
predictedValues	laying	sitting	standing	walk	walkdown	walkup
laying	293	0	0	0	0	0
sitting	0	231	10	0	0	0
standing	0	33	273	0	0	0
walk	0	0	0	228	3	1
walkdown	0	0	0	0	188	0
walkup	0	0	0	1	9	215

Accuracy : 0.9616 || 95% CI : (0.9506, 0.9708) || P-Value [Acc > NIR] : < 2.2e-16

As seen from table2 above, I obtained an overall accuracy of 96.16% with my random forest model. We had a very small P value and much higher accuracy than our benchmark, which signaled that our model was significantly better than random guessing. The accuracy of the model was between 95.06% and 97.08% at 95% confidence interval. Finally, we saw that most of the activities obtained close to 100% prediction; except for subjects who were actually sitting, as there was more than 10% chance that we would incorrectly predict them as standing.

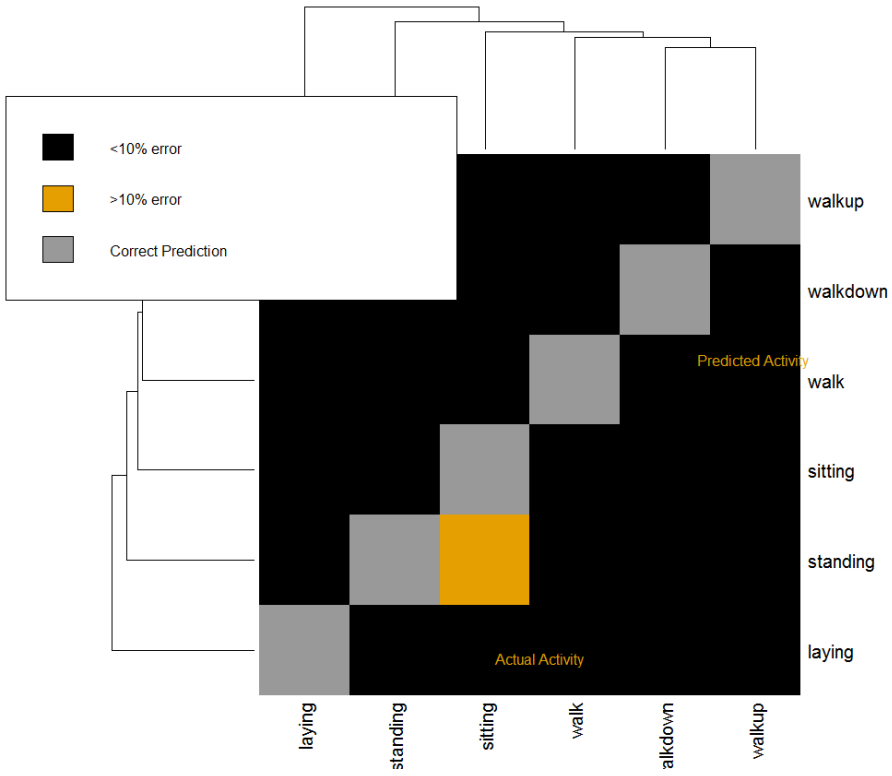
Table3 Error Rate:

Statistics by Class:

	laying	sitting	standing	walk	walkdown	walkup
Sensitivity	1.0000	0.8750	0.9647	0.9956	0.9400	0.9954
Specificity	1.0000	0.9918	0.9725	0.9968	1.0000	0.9921
Pos Pred Value	1.0000	0.9585	0.8922	0.9828	1.0000	0.9556
Neg Pred Value	1.0000	0.9735	0.9915	0.9992	0.9907	0.9992
Prevalence	0.1973	0.1778	0.1906	0.1542	0.1347	0.1455
Detection Rate	0.1973	0.1556	0.1838	0.1535	0.1266	0.1448
Detection Prevalence	0.1973	0.1623	0.2061	0.1562	0.1266	0.1515

To be precise, we could see from table3 above the actual error rates of our model. The sensitivity value for sitting (0.8750) was the lowest across all 6 classes while we had universally good specificity value across all models. Prevalence is the rate of actual occurrence of each class, while detection rate is the rate of predicted of occurrence of each class. As the value of prevalence was close to the value of detection rate across all 6 classes, this was another sign that we had a fairly accurate model.

Fig2: Confusion Heatmap



By plotting the sensitivity values on a heatmap, we arrived at figure 2 above. As we had already discussed, the weakness of our model are in subjects who were actually sitting, as there was more than 10% chance that we would incorrectly predict them as standing.

Confounders

It might be possible that this relatively large error margin of sitting was caused by confounders. After all, the main variable that would help us differentiate between standing and sitting would probably be the gyroscope and accelerometer's Z variables. The Z variables could easily be fooled by a subject sitting on a high chair or high surface; hence an important confounding variable might be the height of the surface where the subject was sitting on.

Conclusions:

This random forest model, with its high accuracy between 95.06% and 97.08% at 95% confidence interval and small P value proved to be fairly accurate in predicting the activity of subjects, although the model did had a relatively significant error margin of 10% in misclassifying a sitting person as standing. To further increase its accuracy in future modeling, we may need more data from different subjects to train the model. We should also find out the specific features that could help us differentiate between sitting and standing and upweigh those factors. Furthermore, while this random forest model worked well on the test data set that had similar distribution pattern as the training data set, we might had expected lower accuracy rates should we had tested the model on test data that were more different from the training data set. Finally, scaling the model may also be an issue as the current training data only consists of a small 3500 observations; we may need to consider feature selection to drop the less important features if we need to scale the model in future.

Bibliography

- Alon Halevy, P. N. (2010). *The Unreasonable Effectiveness of Data*. Retrieved from http://www.csee.wvu.edu/~gidoretto/courses/2011-fall-cp/reading/TheUnreasonable%20EffectivenessofData_IEEE_IS2009.pdf
- Engadget. (2013). *Engineer Guy shows how a phone accelerometer works, knows what's up and sideways*. Retrieved from http://www.engadget.com/2012/05/22/the-engineer-guy-shows-how-a-smartphone-accelerometer-works/
- Kaggle. (2013). *Random Forests*. Retrieved from https://www.kaggle.com/wiki/RandomForests
- mobile88. (2013). *Gyroscope In Smartphones*. Retrieved from http://www.mobile88.com/news/read.asp?file=/2012/4/21/20120421165938
- R. (2013). *doSNOW*. Retrieved from http://cran.r-project.org/web/packages/doSNOW/index.html
- R. (2013). *foreach*. Retrieved from http://cran.r-project.org/web/packages/foreach/index.html
- R. (2013). *R Project*. Retrieved from http://www.R-project.org
- RStudio. (2013). *Using R Markdown with RStudio*. Retrieved from http://www.rstudio.com/ide/docs/authoring/using_markdown
- scikit. (2013). *Decision Trees*. Retrieved from http://scikit-learn.org/dev/modules/tree.html
- Segal, M. R. (2003). *Machine Learning Benchmarks and Random Forest Regression*. Retrieved from http://www.epibiostat.ucsf.edu/biostat/cbmb/publications/bench.rf.regn.pdf
- StatSoft. (2013). *How to Reduce Number of Variables and Detect Relationships, Principal Components and Factor Analysis*. Retrieved from http://www.statsoft.com/textbook/principal-components-factor-analysis/
- UCI. (2013). *Human Activity Recognition Using Smartphones Dataset*. Retrieved from http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones