Predicting Movement Based on Smartphone Accelerometer and Gyroscope Data

Introduction

Smart phones are typically equipped with accelerometers and gyroscopes (sensor signals) in order to pass phone orientation information to the user interface [1]. The data used in our project was collected from accelerometers of several different Samsung Galaxy SII Smartphones [2]. Using the data, we were tasked with creating a function to predict what activity a person was performing.

Being able to predict what a person is doing based on data from a phone has many different applications to phone usability and function. For instance, a phone developer may like to use information from the sensor signals to know if the user is lying in bed or walking up a flight of stairs and change functionality of the phone based on that information.

Methods

Data Collection

The raw data was provided by the University of Genova's Smartlab [3] and is hosted by the UCI Machine Learning Repository [2]. The course instructor took an extra step by aggregating the data into one file in order to easily read the data into R.

This aggregate data was downloaded on March 2, 2013 in an RDA format from an Amazon public host. It contained 7,352 observations from 30 different subjects and contained 562 variables.

The variables consisted of the following information:

- "Triaxial acceleration from the accelerometer (total acceleration) and the estimated body acceleration.
- Triaxial Angular velocity from the gyropscope.
- A 561-feature vector with time and frequency domain varaibles."[2]
- An activity label (walking, waling upstairs, waling downstairs, sitting, standing, or laying)
- A number indicating the subject performing the activity.

Exploratory Analysis

Analysis was performed using the R statistical programming language (version 2.15.3) in R Studio (version 0.97.320) [4]. Exploratory analysis was used to verify

the data had expected values, the values of the variables were within expected ranges, and that there were no missing values in the data.

Statistical Modeling

Several methods were initially used to generate a predictive model. But the one that proved to be the most accurate (when testing on the validation set, which is defined below) was Leo Breiman and Adele Cutler's random forest algorithm [4]. This algorithm creates multiple trees by bootstrapping both the data and the variables.

The model was then used with the predict function in R to create predictions based on data from the test set (again, defined below). A simple error rate was then calculated using the following formula:

Error Rate = (Number of incorrect predictions)/(Total number of Predictions).

Results

The data set was complete, meaning there were no missing values. All of the variables from the sensor signals had been normalized and so were expected to have normal summary statistics (mean, standard deviation, etc). This was confirmed using the built in summary function in R. The subject and activity variables both had expected values.

Some variable names were not unique so the data.frame() function was used to create unique variable names across the data set. The subject and activity variables were changed from character type to factor in order to pass these variables into the randomForests functions.

The data were then divided into three different groups:

A. A training set based on all of the observations from subjects 1, 3, 5, 6, 7, and 8,
B. A validation set containing all of the observations from subjects 11, 14, 15, and 16,
C. A test set containing all of the observations from subjects 27, 28, 29, and 30.

The data were divided by subject to ensure that the model did not generate predictions based on one individual's phone. After the division, the subject variable was omitted altogether from the training and validation sets to further ensure model accuracy.

We then ran the randomForest function on the training set with activity as the outcome variable and all other variables as predictors. This generated a forest of 500 trees and tested a selection of 23 variables at each split in the tree. This model had an out-of-bag estimate of error of 1.58%. In other words, the model miss-

categorized the observations that were "bootstrapped" out of the training set less than 2 out of every 100 times.

Finally, the model was used on the test set. This produced an error rate of 8.15%. In other words, the model was able to correctly predict the type of activity from the test set more than 9 out of 10 times.

To confirm the behavior of the randomForest function, we plotted several of the variables that were deemed "important" by the function. (The details about how the importance is described in the randomForest library documentation.) An example of one of these plots can be found in Figure 1, A. The plots clearly show grouping among the activities and can therefore confirm the reliability of the function.

Conclusion

Our model was fairly accurate at predicting activity given certain data from a phone's sensory signals. But we haven't discussed what would be deemed an acceptable level of error. To know would be considered acceptable we would have to know the purpose of the predictive model. For instance, if a phone developer were creating a game, correctly predicting a users activity 9 out of 10 times might not be good enough.

Figure 1, B shows where the errors from our prediction on the test set occurred. Most of the errors (68%) were produced while trying to predict whether a subject was sitting or standing. Creating a model that better differentiates these two actions would certainly increase the models overall accuracy and reliability. Perhaps another variable would need to be introduced in order to do so. Going further, we speculate that additional sensory signals such as light meters or microphones could be used to help predict activity.

Works Cited

[1]Wikipedia "Accelerometer" Page. URL: http://en.wikipedia.org/wiki/Accelerometer. Accessed 3/10/13.

[2]UCI Machine Learning Repository Website. URL: http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones. Accessed 3/10/13

[3] University of Geneva Smartlab website. URL: http://www.smartlab.ws/. Accessed 3/10/13/

[4] R Core Team (2012). "R. A Language and Environment for statistical computing." URL: http://www.R-project.org

[5]Random Forest Algorithm http://stat-www.berkeley.edu/users/breiman/RandomForests/cc_graphics.htm