CodeBook

Rebecca Kimmel

July 24, 2015

## Acquiring the data set

The original files are located at: <https://d396qusza40orc.cloudfront.net/getdata%2Fprojectfiles%2FUCI%20HAR%20Dataset.zip>

After downloading the zip file, the files need to be unzipped into the working directory.

## Reading the data into the function

The files are read into data frames by the function as follows:

* y\_test.txt
* subject\_test.txt
* X\_test.txt
* y\_train.txt
* subject\_train.txt
* X\_train.txt

The data frames are then appended together to create one large frame.

The column labels of subject and activity are then updated to be more readable. The data dictionary files are then read in:

* activity\_labels.txt
* features.txt

The columns in the resulting features table is then scanned and selected into a subset dataframe to contain only those labels that contain mean() and std(). This is the dataset in which we are interested.

Using this subset of names, we extract all of the data from the large dataframe that was initially created, followed by adding the descriptive activity labels, and subject ID.

## Tidying the Data

The dataframe is then aggregated and the mean is computed for each feature data column. This operation is then followed by ordering the data by subject followed by the activity each subject performed. The feature labels are then updated to be more readable.

## Results

Results of the tidy data set are displayed and written to the file tidydata.txt in the working directory.

# run\_analysis() Code

#----------- Read the data files ----------------------------------  
   
 print("Reading Test Data ....")  
   
 dataActivityTest <- read.table ("y\_test.txt" ,header = FALSE)  
 dataSubjectTest <- read.table ("subject\_test.txt", header = FALSE)  
 dataFeaturesTest <- read.table ( "X\_test.txt" ,header = FALSE)  
   
 print ("Reading Training Data ....")  
   
 dataActivityTrain <- read.table ("y\_train.txt" ,header = FALSE)  
 dataSubjectTrain <- read.table ("subject\_train.txt", header = FALSE)  
 dataFeaturesTrain <- read.table ( "X\_train.txt" ,header = FALSE)  
   
 #------------ Merge the Test and Training Data together -------------  
   
 print ("Merging Training and Test Data together ....")  
   
 dataSubject <- rbind(dataSubjectTrain, dataSubjectTest)  
 dataActivity <- rbind(dataActivityTrain, dataActivityTest)  
 dataFeatures <- rbind(dataFeaturesTrain, dataFeaturesTest)  
   
 #------------ Add more readable labels ------------------------------  
   
 print ("Updating labels for Subject and Subject Activities ....")  
   
 names(dataSubject) <- c("Subject")  
 names(dataActivity) <- c("Activity ID")  
   
 print ("Reading Activity and Feature definition files ....")  
   
 activityLabels <- read.table ("activity\_labels.txt",header = FALSE)  
 names(activityLabels) <- c("Activity ID", "Activity")  
   
 dataFeaturesNames <- read.table("features.txt",head=FALSE)  
 names(dataFeatures) <- dataFeaturesNames$V2  
   
 print ("Combining Subject, Activity, with Feature Data.")  
   
 dataCombine <- cbind(dataSubject, dataActivity)  
 Data <- cbind(dataFeatures, dataCombine)  
   
 print("Creating Subset of data containing only mean and std feature data.")  
   
 subFeaturesNames <- dataFeaturesNames$V2[grep("mean\\(\\)|std\\(\\)",   
 dataFeaturesNames$V2)]  
   
   
 selectedNames <- c(as.character(subFeaturesNames),"Subject", "Activity ID")  
 Data <- subset(Data, select=selectedNames)  
 Data <- mutate(Data, Activity = activityLabels[Data$`Activity ID` ,2])  
 Data$`Activity ID` <- NULL   
   
 #---------- Tidy the data up by aggregating on mean then ordering it ------  
   
 print ("Tidying the Data that contains the average (mean) of each variable.")  
 print ("Ordered by Subject and Activity ....")  
   
 tidyData <- aggregate(. ~Subject + Activity, Data, mean)  
 tidyData <- tidyData[order(tidyData$Subject, tidyData$Activity), ]  
   
 names(tidyData) <- gsub("^t", "time", names(tidyData))  
 names(tidyData) <- gsub("^f", "frequency", names(tidyData))  
 names(tidyData) <- gsub("Acc", "Accelerometer", names(tidyData))  
 names(tidyData) <- gsub("Gyro", "Gyroscope", names(tidyData))  
 names(tidyData) <- gsub("Mag", "Magnitude", names(tidyData))  
 names(tidyData) <- gsub("BodyBody", "Body", names(tidyData))  
 names(tidyData) <- gsub("\\(|\\)", "", names(tidyData))  
   
 #---------- View the data and write it out to a file --------------------  
   
 View(tidyData, "Results of Samsung Data")  
  
 write.table(tidyData, file = "tidydata.txt",row.name=FALSE)  
 print("Results have been saved and are located in ./tidydata.txt")

# Additional Information about the Samsung Data

Jorge L. Reyes-Ortiz, Davide Anguita, Alessandro Ghio, Luca Oneto. Smartlab - Non Linear Complex Systems Laboratory DITEN - Università degli Studi di Genova. Via Opera Pia 11A, I-16145, Genoa, Italy. [activityrecognition@smartlab.ws](mailto:activityrecognition@smartlab.ws) www.smartlab.ws \*\*\*

The experiments 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. Using its embedded accelerometer and gyroscope, we captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz. The experiments have been video-recorded to label the data manually. The obtained dataset has been randomly partitioned into two sets, where 70% of the volunteers was selected for generating the training data and 30% the test data.

The sensor signals (accelerometer and gyroscope) were pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (128 readings/window). The sensor acceleration signal, which has gravitational and body motion components, was separated using a Butterworth low-pass filter into body acceleration and gravity. The gravitational force is assumed to have only low frequency components, therefore a filter with 0.3 Hz cutoff frequency was used. From each window, a vector of features was obtained by calculating variables from the time and frequency domain. See 'features\_info.txt' for more details.

# For each record it is provided:

* Triaxial acceleration from the accelerometer (total acceleration) and the estimated body acceleration.
* Triaxial Angular velocity from the gyroscope.
* A 561-feature vector with time and frequency domain variables.
* Its activity label.
* An identifier of the subject who carried out the experiment.

# This project uses the following files:

* 'README.txt'
* 'features\_info.txt': Shows information about the variables used on the feature vector.
* 'features.txt': List of all features.
* 'activity\_labels.txt': Links the class labels with their activity name.
* 'train/X\_train.txt': Training set.
* 'train/y\_train.txt': Training labels.
* 'test/X\_test.txt': Test set.
* 'test/y\_test.txt': Test labels.

The following files are available for the train and test data. Their descriptions are equivalent.

* 'train/subject\_train.txt': Each row identifies the subject who performed the activity for each window sample. Its range is from 1 to 30.

# Notes:

* Features are normalized and bounded within [-1,1].
* Each feature vector is a row on the text file.

# Feature Selection

The features selected for this database come from the 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. Then they were filtered using a median filter and a 3rd order low pass Butterworth filter with a corner frequency of 20 Hz to remove noise. Similarly, the acceleration signal was then separated into body and gravity acceleration signals (tBodyAcc-XYZ and tGravityAcc-XYZ) using another low pass Butterworth filter with a corner frequency of 0.3 Hz.

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 were calculated using the Euclidean norm (tBodyAccMag, tGravityAccMag, tBodyAccJerkMag, tBodyGyroMag, tBodyGyroJerkMag).

Finally a Fast Fourier Transform (FFT) was applied to some of these signals producing fBodyAcc-XYZ, fBodyAccJerk-XYZ, fBodyGyro-XYZ, fBodyAccJerkMag, fBodyGyroMag, fBodyGyroJerkMag. (Note the 'f' to indicate frequency domain signals).

These signals were used to estimate variables of the feature vector for each pattern:  
'-XYZ' is used to denote 3-axial signals in the X, Y and Z directions.

The feature variables used for this project are in **bold** and have been assigned new names.

|  |  |
| --- | --- |
| - tBodyAcc-XYZ | **timeBodyAccelerometer-XYZ** |
| - tGravityAcc-XYZ | **timeGravityAccelerometer-XYZ** |
| - tBodyAccJerk-XYZ | **timeBodyAccelerometerJerk-XYZ** |
| - tBodyGyro-XYZ | **timeBodyGyroscope-XYZ** |
| - tBodyGyroJerk-XYZ | **timeBodyGyroscopeJerk-XYZ** |
| - tBodyAccMag | **timeBodyAccelerometerMagnitude** |
| - tGravityAccMag | **timeGravityAccelerometerMagnitude** |
| - tBodyAccJerkMag | **timeBodyAccelerometerJerkMagnitude** |
| - tBodyGyroMag | **timeBodyGyroscopeMagnitude** |
| - tBodyGyroJerkMag | **timeBodyGyroscopeJerkMagnitude** |
| - fBodyAcc-XYZ | **frequencyBodyAccelerometer-XYZ** |
| - fBodyAccJerk-XYZ | **frequencyBodyAccelerometerJerk-XYZ** |
| - fBodyGyro-XYZ | **frequencyBodyGyroscope-XYZ** |
| - fBodyAccMag | **frequencyBodyAccelerometerMagnitude** |
| - fBodyAccJerkMag | **frequencyBodyAccelerometerJerkMagnitude** |
| - BodyGyroMag | **frequencyBodyGyroscopeMagnitude** |
| - fBodyGyroJerkMag | **frequencyBodyGyroscopeJerkMagnitude** |

The set of variables that were estimated from these signals are:

|  |  |
| --- | --- |
| - **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. |

Additional vectors obtained by averaging the signals in a signal window sample. These are used on the angle() variable:

gravityMean tBodyAccMean tBodyAccJerkMean tBodyGyroMean tBodyGyroJerkMean

For more information about this dataset contact: [activityrecognition@smartlab.ws](mailto:activityrecognition@smartlab.ws)

# License:

Use of this dataset in publications must be acknowledged by referencing the following publication [1]

[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

This dataset is distributed AS-IS and no responsibility implied or explicit can be addressed to the authors or their institutions for its use or misuse. Any commercial use is prohibited.

Jorge L. Reyes-Ortiz, Alessandro Ghio, Luca Oneto, Davide Anguita. November 2012.