Predicting Human Activity Recognition (HAR) Using Smartphone Data

DSC383W Data Science Capstone Mini-Project

Instructor: Prof. Ajay Anand

Author: Fengyi Zhao

02/08/2019

Introduction

Human Activity Recognition (HAR) is the problem of classifying sequences of accelerometer data recorded into known well-defined movements. Movements are often normal indoor activities such as standing, sitting, jumping, and going up stairs. Sensors are often located on the subject such as a smartphone or vest and often record accelerometer data in three dimensions (x, y, z). The idea is that once the subject's activity is recognized and known, an intelligent computer system can then help. This project starts with data preprocessing, follows with data analysis and ends with modeling and prediction.

Data Collection and Preprocessing

The data collected consists of 561 different features generated from the raw accelerometer and gyroscope signals. Experiments were carried out with a group of 30 volunteers within an age bracket of 19-48 years while wearing a smartphone (Samsung Galaxy) on the waist. Each person performed sic activities: walking, walking upstairs, walking downstairs, sitting, standing, and lying.

The dataset given by the UCI Machine Learning Repository consists train, test files, and HAR Dataset. The test files are partly used in data visualization, which will be illustrated below. This project mainly focuses on applying analytics and models on the HAR Dataset file. For each record in the dataset 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, and an identifier of the subject who carried out the experiment. 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 data points/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 acceleration signal was then separated into body and gravity acceleration signals. From each window, a vector of features was obtained by calculated variables from the time and frequency domain.

Data Visualization

The following sample data visualization are based on the test datasets.

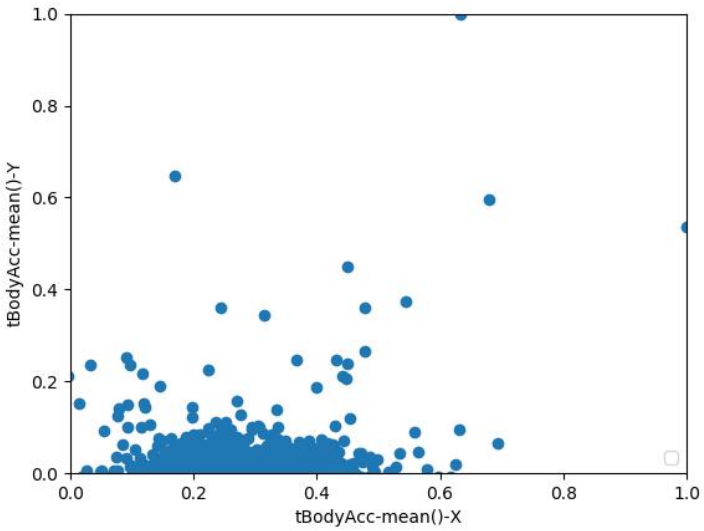

Fig 1: The Scatterplot of Mean Body Acceleration in the xy-direction

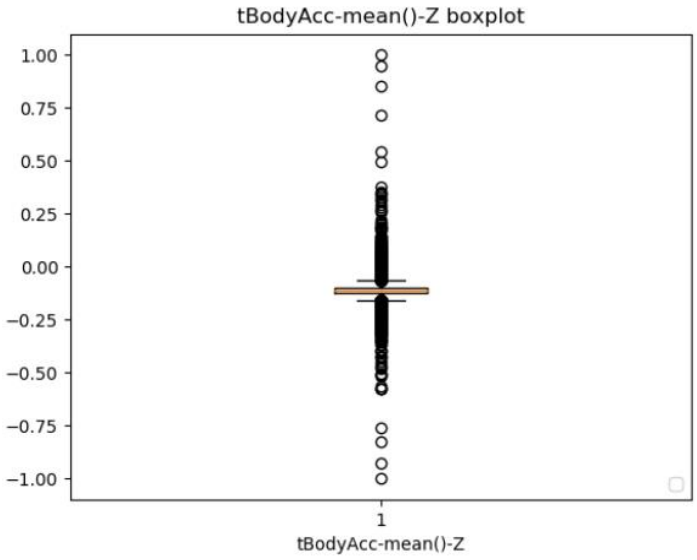

Fig 2: The Boxplot of Mean Body Acceleration in the z-direction

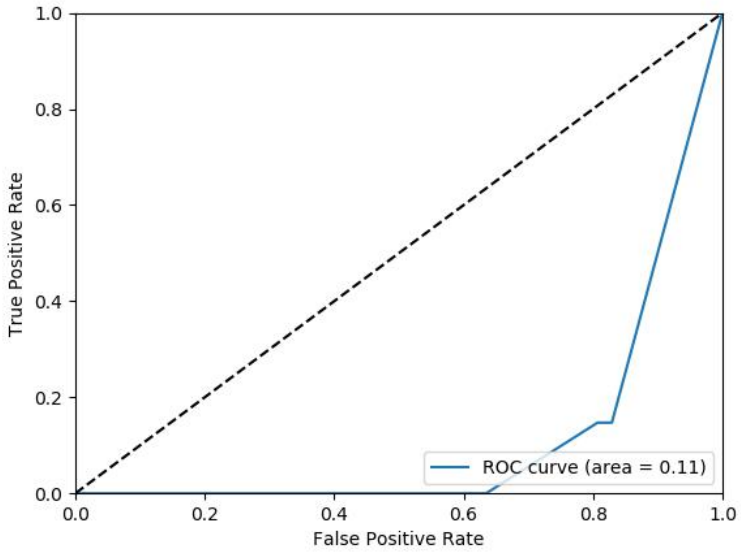

Fig 3: The ROC Curve

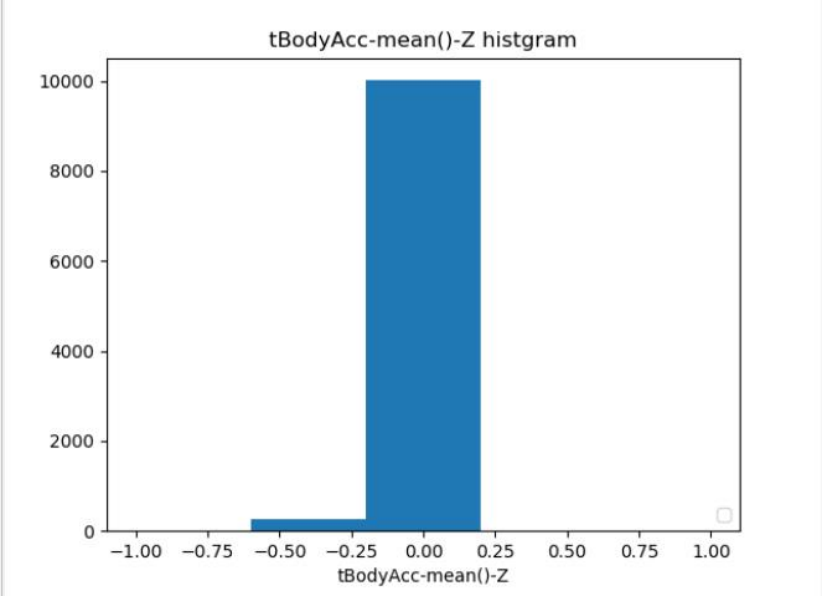

Fig 4: The Histogram of Mean Body Acceleration in z-direction

Methods and Statistical Modeling

There are two main parts in this project. The first object is to develop two predictive models using all 561 features. For each model, report accuracy, confusion matrix, and comment on the differences in accuracy between models (if applicable). The

project chose to use Lasso and random forest. Secondly, for one of the chosen models, vary the number of features used in the prediction (e.g. from 100 to 561), and compute the resulting accuracy. Moreover, determine the number of features required to obtain 8-%, 90% accuracy. The results will be presented and discussed in the following sessions. In the dataset provided by UCI Machine Learning Repository, there were no missing values. 561 features were numeric and target feature was categorical. The task wasa classification task. The goal of this project is to use datasets and to conduct a comparative study on different advanced regression algorithms.

The models used in this project are lasso and random forest. In statistics and machine learning, lasso (least absolute shrinkage and selection operator; also lasso or LASSO) is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it produces. Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the class (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. The random forest model is also used to fulfill the second objective.

Result and Conclusion

By using Random Forest model, the accuracy is 0.36036647438072617, and the confusion matrix is as followed in table 1:

Actual/Prediction	Walking downstairs	Walking	Sitting	Laying	Walking upstairs	Standing
Walking downstairs	438	0	58	0	0	0
Walking	321	116	33	0	0	2

Sitting	153	14	253	0	0	0
Laying	0	0	0	177	0	314
Walking upstairs	1	0	0	13	0	518
Standing	3	0	0	431	49	54

Table 1: The Confusion Matrix of Random Forest Model Application

By using Lasso, the accuracy is 0.3525619273837801, and the confusion matrix is as followed in table 2:

Actual/Prediction	Walking downstairs	Walking	Sitting	Laying	Walking upstairs	Standing
Walking downstairs	0	5	490	1	0	0
Walking	0	0	463	8	0	0
Sitting	0	0	420	0	0	0
Laying	0	0	0	29	462	0
Walking upstairs	0	0	0	73	459	0
Standing	0	0	28	331	178	0

Table 2: The Confusion Matrix of Lasso Regression Model Application

According to the above results, the accuracy of using random forest model is higher than the accuracy using lasso regression model. Therefore, the data performed better with random forest model.

Then by using random forest model and varying the number of features used in the prediction from 100 to 561, the program computes the accuracy accordingly. Sample outputs are as followed:

```
number of features:100, rf_scores:0.3488293179504581
confusion Matrix: (left labels: y_true, up labels: y_pred):
labels   [1, 2, 3, 4, 5, 6]
[[460   0  36   0   0   0]
 [339  61  71   0   0   0]
 [138   4 278   0   0   0]
 [  0   0   0 175   0 316]
 [  1   0   0  13   0 518]
 [  3   0   0 431  49  54]]
- * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- * --
number of features:101, rf_scores:0.3525619273837801
confusion Matrix: (left labels: y_true, up labels: y_pred):
labels   [1, 2, 3, 4, 5, 6]
[[423   0  73   0   0   0]
 [334  50  87   0   0   0]
 [ 80   4 336   0   0   0]
 [  0   0   0 177   0 314]
 [  1   0   0  13   0 518]
 [  3   0   0 436  45  53]]
- * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- * --
number of features:102, rf_scores:0.36443841194435017
confusion Matrix: (left labels: y_true, up labels: y_pred):
labels   [1, 2, 3, 4, 5, 6]
[[471   0  25   0   0   0]
 [311 108  52   0   0   0]
 [135  17 268   0   0   0]
 [  0   0   0 174   0 317]
 [  1   0   0  11   0 520]
 [  3   0   0 432  49  53]]
- * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- * --
number of features:103, rf_scores:0.3651170682049542
confusion Matrix: (left labels: y_true, up labels: y_pred):
labels   [1, 2, 3, 4, 5, 6]
[[455   0  41   0   0   0]
 [302  86  83   0   0   0]
 [108   9 303   0   0   0]
 [  0   0   0 177   0 314]
 [  1   0   0  12   0 519]
 [  3   0   0 430  49  55]]
- * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- * --
```

The data performed the best when 462 features were selected.

Python Libraries and Reference

[1]https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones

[2]https://en.wikipedia.org/wiki/Random_forest

[3]https://machinelearningmastery.com/how-to-load-and-explore-a-standard-human-activity-recognition-problem/

[4]https://github.com/ani8897/Human-Activity-Recognition/blob/master/reports/report.pdf

[5]https://en.wikipedia.org/wiki/Lasso_(statistics)

[6] Travis E, Oliphant. **A guide to NumPy**, USA: Trelgol Publishing, (2006).

[7] John D. Hunter. **Matplotlib: A 2D Graphics Environment**, Computing in Science & Engineering, **9**, 90-95 (2007), DOI:10.1109/MCSE.2007.55 (publisher link)

[8] Wes McKinney. **Data Structures for Statistical Computing in Python**, Proceedings of the 9th Python in Science Conference, 51-56 (2010) (publisher link)

[9] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay. **Scikit-learn: Machine Learning in Python**, Journal of Machine Learning Research, **12**, 2825-2830 (2011) (publisher link)

[9]https://www.scipy.org/citing.html

[10]https://towardsdatascience.com/human-activity-recognition-har-tutorial-with-keras-and-core-ml-part-1-8c05e365dfa0

[11]https://www.kaggle.com/uciml/human-activity-recognition-with-smartphones/discussion