Title: Predicting cellphone user activity based on sensors data

Introduction

With more than 192 million smartphones in the US in 2016[1], there is an explosion of mobile applications and we are only starting to realize their potential. The large number of sensors embedded in these devices (GPS, accelerometer, compass, light sensor, etc) gives us access to large datasets that open new perspective in many different fields. Mobile Health[2] for example is a rapidly growing domain with life-saving applications such as detecting when an elderly person falls on the ground[3]; or coaching the user to make sure he moves enough every day[4].

In this paper, we look at data captured on Samsung Galaxy S3 phones[5] and show that it is possible to predict the current activity of the user using the data of the phone sensors, with an accuracy superior to 90%.

Methods

Collecting and checking the data

We downloaded the data from the Coursera DataAnalytics Class page during the winter 2013 session (https://spark-public.s3.amazonaws.com/dataanalysis/samsungData.rda).
To load the data and perform our predictive analysis, we used the R Programming Language[6].
The dataset consists of 7352 observations of 563 variables.
The activity variable in the dataset which measures the activity that the user was performing at the time of the observation was transformed into a categorical variable. It is the outcome variable for our study.
The subject variable measures which subject was performing the activity for this observation.
The 561 other variables consist of the raw values captured by the sensor at the time of the observation and normalized, as well as new variables calculated from the sensor raw data. Those new variables give statistical information on the sensor data (mean, standard deviation, etc), analysis in the frequency domain as well as some basic decomposition (acceleration is decomposed into gravity and body acceleration for example).
We had to rename some of those variables to facilitate processing but we did not alter the data.
A first analysis of the data showed no missing values and all the values were in the expected normalized range.

Preparation of the dataset for a predictive analysis

The observations of subjects 27, 28, 29 and 30 were separated from the rest of the data, to be used as our final test set. Those observations were never used to train the model or compare different models. This data set was used only once, with our final model to get an estimate of the error rate of our final model. The final error reported is based on the misclassification error rate of our final model used to predict the activity of the observations in this dataset.
The rest of the data was split into a training set (2/3) and a cross-validation set (1/3. The training set is the one we used to build different models. The cross-validation set is the set we used to compare

different models and get an estimate of our misclassification error rate for each of them.

Predictive modeling

We chose a "brute-force" approach and ran different models against all the variables. Our approach is inspired by the Caret R package[7] but we proceeded manually to test and tune different models.

In this approach the main challenge is to make sure that we are not over-fitting the training data. To reduce the risk of over fitting, we used bootstrapping. Some of the models (such as those generated by the randomforest[8] algorithm) already include bootstrapping but when the model we used did not (such as models generated by the tree method), we manually bootstrapped the model.

Cross-validation

Cross-validation is the second step to ensure that the model does not over-fit the training data. We used cross-validation on the dedicated cross-validation data set to choose the final model.

Results

We used four different predictive models: tree, rpart, random forest and support vector machines.

Figure 1 shows the results of our different models against the cross-validation set. The misclassification error rate is also reported against our training dataset.
The model that gives the best results on the cross-validation set is the one built with Support Vector Machines[9]. The error rate on the cross-validation rate is only 8.62%.

It is interesting to remark that the random forest model had a perfect (0%) error rate on the training set which is an obvious sign of over-fitting the data. On the cross-validation set this model gets an error rate of 10.18% which is worse than the model built with support vector machines.

Reporting the error rate of our final model

We select the model built with support vector machine because it is the best one after cross-validation. When we apply this model to the official test data, the error rate we get is 4.85%.

Conclusions

Using a relatively large dataset and a brute-force approach we were able to create a predictive model that is able to predict the activity of a subject based on information captured by her or his smartphone sensors with an accuracy superior to 95%. This number seems to be well in the range of similar studies[10].
One obvious shortcoming of our brute-force approach is that we did not look for specific features in the dataset. Our results may be improved by a careful selection of features to include in the model.

We believe that the model could be further improved with larger datasets as predictions usually are improved with more data[11]. Another way to improve our model would be to include more activities in the research, such as driving, running, or using a computer.

References:

[1] Emarketer (2012): Number of smartphone users in the US URL: http://www.statista.com/statistics/201182/forecast-of-smartphone-users-in-the-us/

[2] Wikipedia: http://en.wikipedia.org/wiki/MHealth

[3] Ivo C Lopes, Binod Vaidya, Joel J P C Rodrigues (2012): Sensorfall an accelerometer based mobile application URL: http://www.researchgate.net/publication/228407330_Sensorfall_an_accelerometer_based_mobile_application

[4] Jawbone Inc.: "Jawbone Up": https://jawbone.com/up

[5] 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

[6]R Core Team (2012). "R: A language and environment for statistical computing." URL: http://www.R-project.org

[7] Max Kuhn (2008) – "Journal of Statistical Software: Building Predictive Models in R using the caret package". URL: http://www.jstatsoft.org/v28/i05/paper

[8] Leo Breiman and Adele Cutler - "Random Forests". URL: http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

[9] Chang, Chih-Chung and Lin, Chih-Jen: "LIBSVM: a library for Support Vector Machines". URL: http://www.csie.ntu.edu.tw/~cjlin/libsvm

[10] Jennifer R. Kwapisz, Gary M. Weiss, Samuel A. Moore - "Activity Recognition using Cell Phone Accelerometers". URL: http://www.cis.fordham.edu/wisdm/public_files/sensorKDD-2010.pdf

[11] Peter Norvig - "The Unreasonable Effectiveness of Data". URL: http://www.youtube.com/watch?v=yvDCzhbjYWs