Thomas M. Alisi
Jeff Leek
Data Analysis
10.3.13

ACTIVITY PREDICTION MODELS USING SAMSUNG ACCELEROMETER DATA

1. Introduction

Research in activity and gesture recognition is quite important nowadays [ref. 3], as it can help in several different fields, such as video surveillance, health, disease prevention and much more. Since the introduction of smartphones, and most importantly the accelerometers mounted in the devices, along with their wide adoption across all countries, plenty of data describing different gestures has become availble. This data can be used proficiently to monitor and predict activities of people carrying the device, to help them expoiting the goals fixed by a specific application that takes advantage of these values. As a matter of fact, a simplified version of the predictor described in this paper has been used to teach first aid techniques in a project financed by the Technology Strategy Board and lead by the Resuscitation Council in the UK [ref. 7]. In this paper we will show how a set of data can be used to train models and predict what activity is performed by a user producing comparable values through an accelerometer mounted on a mobile device.

2. Methods

Data collection

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. [ref. 1]

All of the columns of the data set (except the last two) represents one measurement from the Samsung phone. The variable subject indicates which subject was performing the tasks when the measurements were taken. The variable activity tells what activity they were performing. [ref. 2]

Exploratory analysis

Great help has been provided by the community in the initial approach to understanding how the acceleramoter data is structured, through a discussion around differences between total body acceleration, gravity, jerk and all related elements [ref. 4].

- 1st 40 columns contain data related to body acceleration along the 3 axes, this is taking into account the gravity vector
- columns 41-80 contain data related to gravity acceleration only along the 3 axes
- columns 81-120 contain data related to acceleration jerk for the body along the 3 axes (jerk is 3rd order derivative of position, equals 2nd order derivative of velocity, 1st order derivative of acceleration [ref. 5])

- further exploration shows that: columns 121-160 contain body gyroscope, 161-200 body gyroscope jerk, and up to column 265 other derivatives in the time domain, while from column 266 to column 554 the values transformed in the frequency domain, then columns 556 to 561 for the angles between accelerometer vectors and the last two columns dedicated to subject and activity labelings
- table of activities show how they are equally distributed across 6 items: laying, sitting, standing, walk, walkdown, walkup

laying	sitting	standing	walk	walkdown	walkup
1407	1286	1374	1226	986	1073

- table of subjects show how they are equally distributed across 21 people, for the purpose of this analysis, we are going to use subjects labeled 1, 3, 5, 6 for training sets and subjects 27, 28, 29, and 30 for tests.

1	3	5	6	7	8	11	14	15	16	17	19	21	22	23	25	26	27	28	29	30
347	341	302	325	308	281	316	323	328	366	368	360	408	321	372	409	392	376	382	344	383

A quick observation of the 1st 40 columns reveals that there are not empty values in the dataset, and a boxplot shows that they are all normalized in a [-1,1] interval, so it should not need removal of outliers. the same applies the all numeric values contained in the dataset. Calling the summary function also reveals that reserved R keywords are used for variable names, so to stay on a safe side, when creating the training and test set, they will be converted using R function make.names.

The activity field has been stored as character so it will be converted to factor in the training/validation/test sets. We subset the whole dataframe for training, validation and tests with the subjects we want to use, then ordering by activity, cleaning up variable names (using make.names in R) and removing the column related to subjects since we don't want it to appear in the model. The training set is made of 328 obs., validation 346 obs. and test 376 obs., we also create an extended training set made of 5867 observations to validate our models without bootstrap. All the observations are consistently made of 562 variables.

Statistical modeling

We are going to use the data from subjects 1,3,5,6 as training sets to create a range of regression models, using a mixture of kernel based functions (i.e. using trees, randomForests and support vector machines) taking extra care in balancing the bias / variance trade off in statistical modeling, as described in the analysis section of this paper. Particularly, given the nature of this dataset, differences in errors will be evaulated when using bootstrap techniques in favour of enlarging the dataset and creating models without sampling. A complete recap of statistical methods (regression models and bootstrap technique) mentioned in this paper can be found in [ref. 6]

3. Analysis

After plotting values from the training set, we can see clear clusters of activities: these plots show comparable patterns no matter the variable chosen for the y-axis [fig. 1a], indicating that it is worth creating a regression model that uses all the variables in the data set, we will then proceed with further exploration in order to reduce the complexity of our models. We load the required libraries and set a seed so as to generate a consistent set of training models, so that errors on validation and test will be always consistent.

We start creating 3 different models using the basic training set: tree, random forest and svm.

	laying	sitting	standing	walk	walkdown	walkup	class.error
laying	55	0	0	0	0	0	0.00000
sitting	0	46	4	0	0	0	0.08000
standing	0	3	54	0	0	0	0.05263
walk	0	0	0	62	1	1	0.03125
walkdown	0	0	0	1	47	1	0.04082
walkup	0	0	0	0	2	51	0.03774

Random forest with 500 trees with bootstrap shows an error < 4% in activity classification, while tree splits show that first variables used are all in the time domain. Now we can do some prediction on the **validation test** to see how the models behave, obtaining the table below.

random forest	**svm**	**tree**
0.1763	0.2486	0.2139

We obtain 17% error using random forest, 25% using svm and 21% using trees. Given that a random forest training is natively executed using boostrap technique, it is reasonable to obtain a lower error and does not seem that the model is overfitting the data, returning a moderate error but still allowing a decent bias.

Larger errors using svm and tree are probably given by the nature of the reduced dataset, so in order to obtain values that are comparable to a random forest using bootstrap, we enlarge the training set using a greater number of subjects (in this case we have a training set made of 5867 observations of 562 variables) and we check the predictors again. Printing the tree now shows how the first split are done using a mixture of frequency, time and angle variables: this supports the idea that all variables contribute to the decision, no matter their specific domain. Now check predictions of the new models based on the **larger training set**.

svm	**tree**
0.1156	0.1994

Using tree and svm on larger dataset significantly reduces the error, svm in particular benefits of roughly 10% reduction.

Now it's time to validate data on our **test set** using the random forest model we already had and the new tree and svm models based on larger training set.

random forest	svm	tree
0.08356	0.04582	0.1375

Validating results on test set show that random forest (using bootstrap) and svm (using a larger data set) produce comparable results, with error < 10%, while the simple tree in this case is slightly worse. The intersting part of using a tree is that it can be easily examined performing a k-fold cross validation on the tree [fig. 1b] to see how it can be optimized.

We see a significant reduction of misclassifications for a depth larger than 6, this can help creating a simplified version of the tree in order to achieve comparable error with less variables. We can then validate our pruned tree against the test set, obtaining a value of 0.1482: this is slightly higher (~ 1%) than the full model, but the reduction of the number of variables is much greater, allowing to select the significant variables for a correct classifcation [fig. 1c].

4. Conclusions

The exploratory analysis revealed that a high number of variables present in these observations define values in different domains, being time, freqency and angles. Further analysis reveals a clear clustering of activities around specific values of acceleration across all 3 axes [fig. 1a]

Prediction defined as outcomes for activities with all the variables as covariates and using random forest based on bootstrap techniques to define the data set confirms to be comparable to ordinary tree and svm models trained on a larger population.

Exploring the tree models reveals that it's possible to subset the number of variables used for modeling, significantly reducing the complexity of the model [fig. 1b and 1c].

Training the random forest on the large dataset would increase the computing time of a large scale, and would probably provide an overfitted model. Potential problems of this analysis are given by the size of the dataset and further analysis should be carried on in case we need to build models related to more specific activities (such as the CPR mentioned in the introduction of this paper)

A. References

1. data collection description, page accessed on march 8th http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones

2. data analysis assignment n.2, page accessed throughout the days of the assignement until march 11th https://class.coursera.org/dataanalysis-001/human_grading/index

3. Activity Recognition using Cell Phone Accelerometers, Jennifer R. Kwapisz, Gary M. Weiss, Samuel A. Moore, Department of Computer and Information Science http://www.cis.fordham.edu/wisdm/public_files/sensorKDD-2010.pdf

4. about accelerometers, gyroscopes, and relevant elements in the data set, https://class.coursera.org/dataanalysis-001/forum/thread?thread_id=2771

5. Jerk in physics, page accessed on 8th march, http://en.wikipedia.org/wiki/Jerk_(physics)

6. Elements of statistical learning http://www-stat.stanford.edu/~tibs/ElemStatLearn/

7. Lifesaver, adoption of gesture prediction on tablets, https://life-saver.org.uk/