Title: Classification of human activities using Smartphones

Introduction

UCI Machine Learning Repository [1] provides a dataset with measurements recorded by a group of 30 volunteers (19-48 years old). 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, the phone captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz – ie. recorded 50 events per second in the 3 dimensional plane [2].

“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.” [3]

Purpose of this analysis is to find and evaluate a classifier that will, based on the features of the dataset, classify observations and assign them into the correct class.

Methods

The dataset contains 7352 observations of 563 variables. All but two (subject, activity) are numeric and normalized into the range of [-1, 1]. Gyroscope measures orientation on the x-y-z axis, accelerometer speed – together they provide information on linear acceleration (moving forward/backward), angular velocity (turning), jerk (sensation of sudden increase/decrease in speed - and jerk [4]). Dataset contains a number of 'derived' attributes for each of the main measurements: such as acceleration on X axis and and its derived values: average, minimum, maximum, standard deviation (and many others). These derived variables record measurements in the time-window (ie. number of events is per certain time-windows is represented by the average per the time-window; these events overlap in time). Number of filters has been applied to the dataset by the original authors. There are no missing values, however certain features re-state already recorded data. Thus care must be taken to exclude these values from the linear regression because they are not independent.

The dataset was divided into a training and testing subsets. The test set contains measurements for subjects with id 27, 28, 29, 30 (1485 observations). The rest is reserved for the training set (7351 obs). The two datasets do not overlap. I report a two metrics to compare the performance of the classification function, Root mean square error [5] and F-Score [6]

Analysis

The exploratory analysis of the subject 1 hinted on a strong linear relationship between activities and the certain measurements suitable for linear regression [10] (e.g. maximum values for acceleration in the X axis, ie. moving forward/backward, are the sure sign that separates walking from sitting etc). But for a number of features, the relationship was less then clear.

Because of a high number of features, I have first tried to apply the step() function of the R software [7, 8] to select the most important features (by removing the linearly dependent tBody... measurements and using the Fourier transform [9] values in their places – as they seemed to separate activities with higher margins).

Yet this process did not produce satisfactory results, therefore I have manually reviewed plots of the features for the 1st subject and selected 16 features to use for a linear regression model (they produced a statistically highly significant model, on alpha level <0.01; however the model was not covering all important features and was able to classify correctly 62% of the test set cases).

```
glm(formula = activity ~ tBodyAcc.max.X + tBodyAcc.energy.X +
    tGravityAcc.mean.X + tGravityAcc.mean.Y + tGravityAcc.max.Y +
    tGravityAcc.energy.X + fBodyAccJerk.mean.X + fBodyAccJerk.mean.Y +
    fBodyAccJerk.energy.X + tBodyGyro.mean.Y + tBodyGyro.max.Y +
    tBodyGyro.energy.Y + tBodyGyroJerkMag.mean, family = "gaussian",
    data = trainSet)
```

Since there are possible outliers in the data (ie. activity of laying contains high values for jerk, and even acceleration – I suppose that means people turn in beds much more than they fall from chairs). I have tried to apply robust linear regression to deal with the possible outliers. I have also attempted to use the automated selection of features again (from the original set of hand-picked values), with these results:

```
Hand-picked features, linear regression : 62% observations correctly classified
Hand-picked features – robust linear regression: 66.6%
Features pruned by AIC: 62%
Robust linear regression: 64.9%
```

The robust linear regression indeed improved results (the automatic selection of features, on the other hand, did decrease the performance – but I must repeat the features were selected from the already hand-picked values). The main sources of errors was that laying(1) was often confused with sitting (2) and walking with walking down (5) – with added confusion between classification between walking up and down as the contingency table shows:

```
          1   2   3   4   5   6
  1 251  42   0   0   0   0
  2   0 178  86   0   0   0
  3   0  40 243   0   0   0
  4   0   0   0  87 141   1
  5   0   0   0   1 149  50
  6   0   0   0  13 122  81
```

I have therefore tried decision trees [11], first selecting from all features of the dataset and letting the tree to choose the most important. Afterwards, I have removed from the selection all features that started with “tBody” (this means the tree algorithm worked with the transformed Fourrier data values

instead of 'raw' measurements) and all features marked “bandsEnergy” were removed as well - these contained detailed measurements (sort of detailed view of a certain range of values). The performance improved slightly:

```
decision tree on all feature: 86.7% observations explained correctly
decision tree after certain confounders were removed: 87.2%
```

These results are much better than the linear regression model that I was able to guess from 'eyeballing' of plots of several hundred features, yet further improvements should be possible (ie. I didn't use smoothing of values or explored reasons for misclassified classes – decision trees are very sensitive to adjustments and order of features).

But the time constraints forced me to proceed to the most promising classification method. I used an R package e1071 package which provides an interface to popular LIBSVM package which provides Support Vector Machine functionality [12]. Using default SVM I was able to correctly classify 96.8% of test cases. Which, in comparison with previous methods, seems more than satisfactory.

The performance of the classification methods for each of the 6 tasks can be compared using a F-measure in the **Plot 1**. Finally, I provide here the root mean square error for each of the method (the lower score means better accuracy). The error is computed as:

rmse = sqrt(mean((real-class - observed-class)^2))

Where the classes can be values: 1-6

```
Linear regression (hand-picked features): 0.6287179
Robust linear regression (hand-picked features): 0.6019049
Linear regression (automatic selection of features): 0.6249579
Robust Linear regression (automatic selection of features): 0.6080271
Decision tree (from all features): 0.4813001
Decision tree (after certain features removed): 0.4457053
Support Vector Machine classification: 0.177904
```

Conclusions

The SVM classification shows the best performance on this dataset, but we should remind ourselves that the dataset already contains values that were properly normalized – therefore SVM can choose the features that best distinguish classes. However, by the nature of SVM it is difficult to know which of the features were the most important.
I have not (due to tight schedule) conducted multi-fold cross-validation. Thus the performance, as reported, can be slightly higher or lower on a next set of data. But we can reasonably expect that SVM outperforms all of the other discussed methods.
By repeating the study, using bootstrap techniques, I can try to evaluate with higher precision confidence intervals for this classifier. Also it should be said that SVM is computationally expensive, if the lower precision of the decision tree (currently ~87%) is acceptable, then it can likely be improved. Especially if we concentrate on the most often confounded pairs (laying-sitting; walk-walkdown; walk-walkup).

References:

[1] UCI Machine Learning Repository
[http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones]

[2] Cartesion Coordinate System [http://en.wikipedia.org/wiki/Cartesian_coordinate_system]

[3] 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

[4] Jerk [http://en.wikipedia.org/wiki/Jerk_(physics)]

[5] Root mean square error - [http://en.wikipedia.org/wiki/Root-mean-square_deviation]
[6] F Score - [http://en.wikipedia.org/wiki/F_score]
[7] Venables, W. N. and Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth edition. Springer.
[8] R Core Team (2012). ”R: A language and environment for statistical computing.” URL: http://www.R-project.org
[9] Fourier transformation - [http://en.wikipedia.org/wiki/Fourier_transform]

[10] Dobson, A. J. (1990) An Introduction to Generalized Linear Models. London: Chapman and Hall.

[11] Breiman L., Friedman J. H., Olshen R. A., and Stone, C. J. (1984) Classification and Regression Trees. Wadsworth.

[12] Chang, C.-C. & Lin, C.-J. (2001). LIBSVM: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm,