Homework

Human Activity Recognition Using Smartphones

10/30/2015

Due: Sunday, November 8.

Smart health and fitness monitoring devices (for example, created by FitBit, Nike, Adidas, Misfit Shine, ...) are very popular. They are also useful for analyzing person's daily physical activities and recommending health-enhancing exercises. At their core, how do these devices work? Before analyzing human activities, the first thing the device needs to do is to recognize which activities are being performed in the first place.

This experiment[1] monitors people carrying Samsung Galaxy smartphones. Measurements are collected using the phones' accelerometers and gyroscopes, while subjects perform 6 different activites:

- WALKING
- WALKING_UPSTAIRS
- WALKING_DOWNSTAIRS
- SITTING
- STANDING
- LAYING

Your task is to classify person's activity based on the phones recordings.

We have preprocessed data for you. To obtain the data in R, run:

```r
source(paste(
    "https://raw.githubusercontent.com/ChicagoBoothML/MachineLearning_Fall2015/master/",
    "Programming%20Scripts/UCI%20Human%20Activity%20Recognition%20Using%20Smartphones/R/ParseData.R",
    sep=""))
data <- parse_human_activity_recog_data()
```

The preprocessed data has been cleaned and normalized, so you are not expected to do any further cleaning.

Training data is located in

```
data$X_train
data$y_train
```

while the test data is located in

```
data$X_test
data$y_test
```

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

Your tasks:

1. Build a Neural Network model to classify the 6 activity patterns and report your Accuracy on the Test set
2. Build a tree-based model (or a few) to do the same thing. Compare accuracy on the test set with that of Neural Networks.

Note: We do not expect you to spent too much time on this exercise. Simply try to build a few models using the h2o package to see how the number of hidden units and number of epochs affect the performance. You can leave all the other parameters at their default values. If you have time, you could also try regularization techniques based on dropout or l1/l2 penalization. Same for boosting or random forests.

Data and additional information on the experiment can also be obtained from our git repository [2] or from the UCI repository[3].

[2] https://github.com/ChicagoBoothML/DATA___UCI___HumanActivityRecognitionUsingSmartphones
[3] http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones