Weightlifting Data Prediction Algorithm

Lucas McLaughlin

Saturday, September 26, 2015

Introduction

This project uses the Weight Lifting Exercise dataset from research done regarding human activity recognition.[1] The researchers write the following in their abstract:

- “Research on activity recognition has traditionally focused on discriminating between different activities, i.e. to predict ‘which’ activity was performed at a specific point in time. The quality of executing an activity, the ‘how (well)’, has only received little attention so far, even though it potentially provides useful information for a large variety of applications.”

In their experiment, they used sensors placed on the glove, arm, belt, and asked participants to do certain weightlifting exercizes first, with flawless technique, and then subsequently in various incorrect ways.

My prediction algorithm is built on this data. If given sensor information from a new participant, my algorithm tells the user whether or not they are doing the exercize correctly, or if they are doing it incorrectly, in what way they are doing it incorrectly.

Getting and Cleaning the Data

The data can be downloaded as a csv file here.

After some initial exploration, I chose to discard the variables containing summary statistics. These summary statistics calculate things like means and standard deviations for certain windows of time. Since the data we were given to test our algorithm on contains only single observations, and not time series observations, I chose to build my model not using the summary statistics. I also discarded the ID variables such as participant names and window ID number. The following script does this:

```
finalTest <- read.csv("./pml-testing.csv")
data <- read.csv("./pml-training.csv")
names <- names(finalTest[, colSums(is.na(finalTest)) == 0])
rawData <- data
data <- data[, c(names[-length(names)], "classe")] #discard summary statistics
data <- data[, -1:-7] #discard names, timestamps, window info
```

The raw dataset had 160 variables. The cleaned up dataset only has 53. This comes out to 13 variables per each of the 4 sensors and then the outcome variable. These are all that is needed to make an accurate model.

Partitioning the Data

I chose to split my data 70% into a training set and 30% into a testing set. I then took 30% of the **training** set and reserved it for cross validation. I chose to split on the outcome variable. Here’s the code:

```
inTrain <- createDataPartition(data$classe, p = 0.7, list = F)
bigTraining <- data[inTrain, ]
testing <- data[-inTrain, ]

inTrain <- createDataPartition(bigTraining$classe, p = 0.7, list = F)
training <- bigTraining[inTrain, ]
cv <- bigTraining[-inTrain, ]
```

So in the end, the training set comprised 49% of the data, the cross-validation set comprised 21% of the data, and the testing set comprised 30% of the data.

Building the Model

I began by trying several different models on the training set and testing them on the cross validation set to get a sense of their accuracy. I discovered that data did not have a strong linear relationship to the outcome variable and after experimentation, discovered that decision trees and random forests performed well. I decided on a random forest for my final model and used the randomForest package to build it. Here is the model built on the combined training and cross validation set:

```
rf <- randomForest(classe ~ ., data = bigTraining, ntree = 1000)
```

When applied to the testing set, it has an accuracy of 99.59% with a 95% CI of (99.39%, 99.74%). My estimated out of sample error is 0.41%.

Conclusion

As stated previously, my prediction model only considers sensor values at instances of time. It has a high degree of accuracy doing this, but it has its limitations. If one were to apply this model to a new participant who was doing a weight lifting exercise in order to tell him whether or not he was doing it correctly, it would have its challenges. For example, for what instance of time would you run this model? Right when the individial begins the exercise? Somewhere in the middle? How would you detect that? A better model would use information from snapshots of time, process them, and then advise the participant.

All things considered, though, this model has a high degree of accuracy and performs very well.

References

[1] Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human '13) . Stuttgart, Germany: ACM SIGCHI, 2013.

Link: http://groupware.les.inf.puc-rio.br/public/papers/2013.Velloso.QAR-WLE.pdf

Loading [Contrib]/a11y/accessibility-menu.js