Prediction of Human Weight Lifting Performance

Albert de Roos

20 mei 2016

Summary

we are interested in predicting whether an fitness exercise is correctly executed and we use dumbell training as an example. Usinf socalled “wearables”" in the form of accelerometers and accelerometers placed on the body, we gathered date for a set of correctly and incorrectly dumbbell exercises and set out to predict based on the data whether an exercise was performed correctly. We tried 2 machine -learning algorithms, random and rpart to predict an we saw that random forest yielded the highest accuracy score on the train and test set. Random forest gave the highest accuracy (0.99 on training and test set). The validation set of 20 exercises had a 100% accuracy as indicate by the online validation. We conclude that, using simple wearables like accelerometers and gyroscopes, we can predict whtehr a dumbell exercise is executed in the correct (specified) way.

Background

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement - a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, our goal was to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. We set out to predict whether an exercise was performed corerectly and incorrectly. More information is available from [here](http://groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).

Loading the datasets

The training data for this project are available from the website. The test data are available here. The data for this project come from this source. There were no problems encountered in loading the data with headers.

```
train_raw <- read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv", header = 
validation_raw <- read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv", heade
```

Exploring the datasets

Six young health participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions: exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E). Class A corresponds to the specified execution of the exercise, while the other 4 classes correspond to common mistakes.

We see that we have a lot of columns with a lot of NA and some columns contain DIV/0 and those are not very useful for prediction. In the validation set, we see that a lot of columns are not used (all NAs) so they

will be of no value for the prediction anyway. It seems therefore a good choice to only use the columns that contain valid numbers from the validation set and subset both in a new datasets. In the validation set, there are also no rows where 'new timestamp' = 'yes' and those rows contain DIV/0 so we filtered out the rows that contained a new timestamp ('yes'). The columns 1 to 7 also were irrelevant for the prediction as they contained information about the exercise itself such as data and time and person who executed the exercise. The person might be a predictor, which would mean that the meaurements are very person-specific which could be further be analyzed in case of low accuracy in the out-of-set data.

```
usecolumns <- which(!is.na(validation_raw[1,]))
train_clean <- train_raw %>% select(usecolumns) %>% filter(new_window == "no") %>% select(-(1:7))
valdidation_clean <- validation_raw %>% select(usecolumns) %>% filter(new_window == "no") %>% select(-(
```

Train the set

We will use the new training set to train the data and get accuracy data on the test set we created. Afterwards, we can predict the values of the class in our validation set of 20 samples. First we create the train and test set, and we will try 2 different machine learning algorithms, randomForest and rpart. Since we are trying to predict categories of data instead of a continuous value, the random forest and rpart seem more suitable for categorization.

Create test and train set

```
in.train <- createDataPartition(y=train_clean$classe, p=0.6, list=FALSE)
training <- train_clean[in.train, ]
testing <-  train_clean[-in.train,]
```

Train using rpart

```
rpartFit <- rpart(classe ~ ., data=training)
prediction <- predict(rpartFit, testing, type = "class")
cm <- confusionMatrix(prediction, testing$classe)
cm$overall[1]  #0.7065331 not very good
```

```
##  Accuracy
## 0.7205882
```

```
cm$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1999  331   82  227   66
##          B   50  811   65   30   77
##          C   54  114 1015  171  185
##          D   46  119   92  681   52
##          E   39  112   86  149 1031
```

Train using random_forest

```
rfFit <- randomForest(classe ~ ., data=training)
rfprediction <- predict(rfFit, testing, type = "class")
rfcm <- confusionMatrix(rfprediction, testing$classe)
rfcm$overall[1]  #0.9923217 pretty good
```

```
##  Accuracy
## 0.9949245
```

```
rfcm$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 2182    7    0    0    0
##          B    6 1478    2    0    0
##          C    0    2 1338   20    0
##          D    0    0    0 1237    1
##          E    0    0    0    1 1410
```

Conclusion

We see that the random forst method yield an accuracy of .992 which is pretty good. The confusion matrix shows that some predictions are missed but overll is the performance quite good. On the validation set ('testing' from the raw data sat), which was checked with the Coursera quiz, we had an overall scope of 20/20!