Coursera Practical Machine Learning Course Project

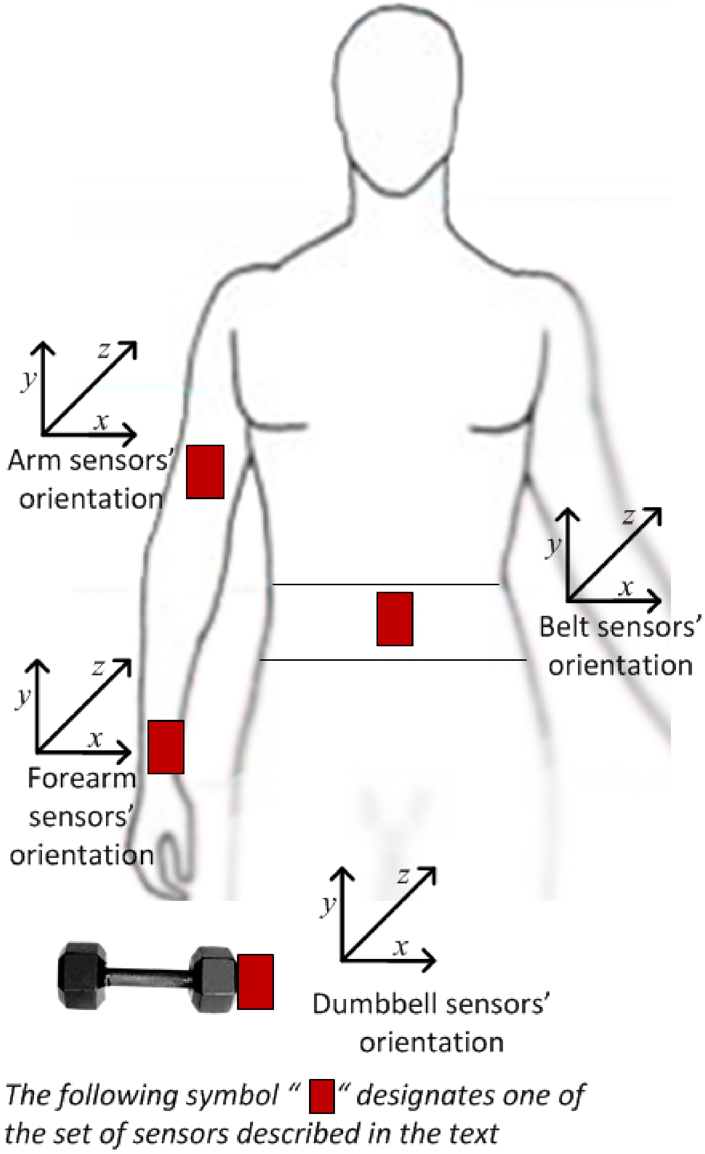
Atul Verma

May 13, 2017

Background and Introduction

Using devices such as *Jawbone Up, Nike FuelBand*, and *Fitbit* 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, we will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participant They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. The five ways are 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). Only Class A corresponds to correct performance. The goal of this project is to predict the manner in which they did the exercise, i.e., Class A to E. More information is available from the website here: http://groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).

The Figure below shows the locations of accelerometers (taken from the publication).

image: 
y
z
x
Arm sensors' orientation
y
z
x
Belt sensors' orientation
y
z
x
Forearm sensors' orientation
y
z
x
Dumbbell sensors' orientation
The following symbol " 🟥 " designates one of the set of sensors described in the text

Data

The training and test data for this project were obtained from the following websites:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

```r
# load data locally
training <- read.csv("pml-training.csv", na.strings = c("NA", ""))
testing <- read.csv("pml-testing.csv", na.strings = c("NA", ""))
dim(training)
```

```
## [1] 19622   160
```

```
dim(testing)
```

```
## [1]  20 160
```

It was determined that training dataset contains 19622 observations and 160 variables, and the testing data set contains 20 observations and the same variables as the training set.

Data Preparation

The columns with missing values (predictors) were deleted. The variables that don't make intuitive sense to be included in the prediction - X, user_name, raw_timestamp_part_1, raw_timestamp_part_2, cvtd_timestamp, new_window and num_window were also removed from further analysis.

```
set.seed(1)
training <- training[, colSums(is.na(training)) == 0]
testing <- testing[, colSums(is.na(testing)) == 0]
trainData <- training[, -c(1:7)]
testData <- testing[, -c(1:7)]
dim(trainData)
```

```
## [1] 19622    53
```

```
dim(testData)
```

```
## [1] 20 53
```

The cleaned data sets trainData and testData both contained 53 columns with the same first 52 variables and the last column namely, classe. The trainData and TestData have 19622 and 20 rows, respectively.

Prepare the data for training

The function *createDataPartition* of the caret package was used to split the data into a training and a cross-validation data set.

```
library(caret); library(rattle); library(rpart); library(rpart.plot)
library(randomForest); library(repmis)

set.seed(10)
inTrain <- createDataPartition(y = trainData$classe, p = 0.7, list = FALSE)
# The index inTrain is used to split the data.
training <- trainData[inTrain, ]
# data set for cross validation
valid <- trainData[-inTrain, ]
```

Model Selection

As many as 52 columns (possible variables) were present in the data sets and it was determined that the use of prediction algorithms such as **classification trees** and **random forest** could be useful. The power of **Random forest** in terms of handling a large data set with higher dimensionality was the key factor to try out the random forest algorithm. It can handle large number of input variables and identify most significant variables so it is considered as one of the dimensionality reduction methods. Further, the model outputs Importance of variable, which can be a very handy feature on some known data set.

5–Fold Cross–Validation

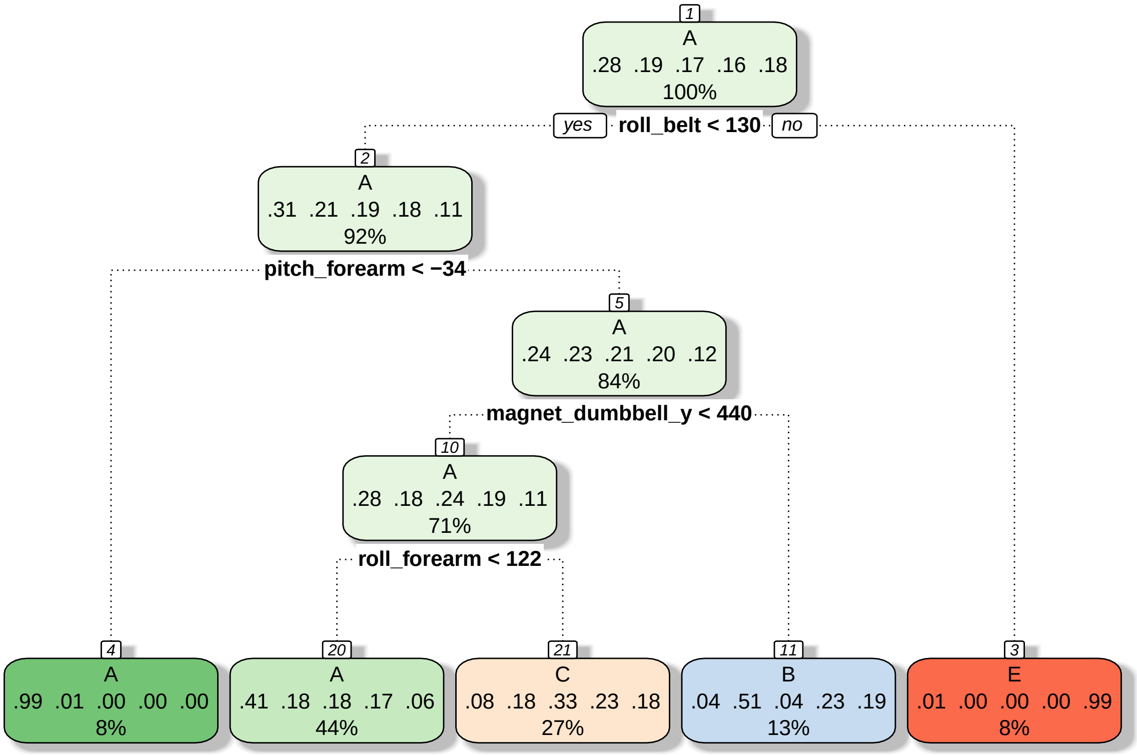
Here, we randomly split the data into 5 distinct blocks of roughly equal size.

- Step 1 We leave out the first block of data and fit a model.
- Step 2 This model is used to predict the held-out block
- Step 3 We continue this process until we've predicted all 5 held–out blocks

The final performance is based on the hold-out predictions. Please note that K= 5 or 10 are most commonly used default parameters for cross validation.

Model I - Classification Trees

```
control <- trainControl(method = "cv", number = 5)
fit_rpart <- train(classe ~ ., data = training, method = "rpart",
                   trControl = control)
fancyRpartPlot(fit_rpart$finalModel)
```



Rattle 2017−May−15 01:05:40 Atul

```
# predict outcome using the validation set
predict_rpart <- predict(fit_rpart, valid)
conf_rpart <- confusionMatrix(valid$classe, predict_rpart)
print(accuracy_rpart <- conf_rpart$overall[1])
```

```
## Accuracy
## 0.499915
```

It was concluded that **classification tree does not predict the outcome** *classe* well enough as refected by the poor accuracy rate of ~ 0.5.

Model II - Random Forest

Random Forest algorithm was attempted next.

```
fit_rf <- train(classe ~ ., data = training, method = "rf",
                   trControl = control)
print(fit_rf, digits = 4)
```

```
## Random Forest
##
## 13737 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 10989, 10989, 10990, 10990, 10990
## Resampling results across tuning parameters:
##
##   mtry  Accuracy  Kappa
##    2    0.9906    0.9881
##   27    0.9909    0.9885
##   52    0.9857    0.9819
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 27.
```

An excellent accuracy was achieved. Next, the outcome from the validation set was predicted. The function *confusionMatrix* was use to calculate the accuracy of the prediction.

```
# predict outcomes using validation set
valid_rf <- predict(fit_rf, valid)
# Show prediction result
(conf_rf <- confusionMatrix(valid$classe, valid_rf))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1673    1    0    0    0
##          B    7 1125    6    1    0
##          C    0    8 1015    3    0
##          D    0    1    8  954    1
##          E    0    1    1    2 1078
##
## Overall Statistics
##
##                Accuracy : 0.9932
##                  95% CI : (0.9908, 0.9951)
##     No Information Rate : 0.2855
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9914
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9958   0.9903   0.9854   0.9938   0.9991
## Specificity            0.9998   0.9971   0.9977   0.9980   0.9992
## Pos Pred Value         0.9994   0.9877   0.9893   0.9896   0.9963
## Neg Pred Value         0.9983   0.9977   0.9969   0.9988   0.9998
## Prevalence             0.2855   0.1930   0.1750   0.1631   0.1833
## Detection Rate         0.2843   0.1912   0.1725   0.1621   0.1832
## Detection Prevalence   0.2845   0.1935   0.1743   0.1638   0.1839
## Balanced Accuracy      0.9978   0.9937   0.9916   0.9959   0.9991
```

```r
(accuracy_rf <- conf_rf$overall[1])
```

```
##  Accuracy
## 0.9932031
```

```r
# percent accuracy is calculated below
percent_accuracy <- round(accuracy_rf *100, 2)
percent_serror <- round((1 - accuracy_rf)*100,2)
```

As shown by the the accuracy, 99.32 %, random forest appears to be an excellent algorithm to model the given dataset. Random forests chooses a subset of predictors at each split and *decorrelate* the trees. This leads to high accuracy, although this algorithm is sometimes difficult to interpret and computationally inefficient. It may be possible to further tune the model to cut down on the computation time. However, no such attempts were made at this point.

Prediction on the Testing Set

Finally, we have used the above derived model to predict the outcome from the testing dats set.

```r
pred <- predict(fit_rf, testData)

# convert predictions to character vector
pred_ch <- as.character(pred)
# create function to write predictions to files
pml_write_files <- function(x) {
    n <- length(x)
    for(i in 1:n) {
        filename <- paste0("problem_id_", i, ".txt")
        write.table(x[i], file=filename, quote=F, row.names=F, col.names=F)
    }
}
# create prediction files to submit
pml_write_files(pred_ch)
```

Prediction Results

The model was used on the testData set and prediction results are presented below:

```
# Show prediction result
print(pred)
```

```
##  [1] B A B A A E D B A A B C B A E E A B B B
## Levels: A B C D E
```

Out-of-sample error

Please note that the accuracy of prediction is 99.32 %. Hence, the **out-of-sample error, (100-accuracy)** is 0.68 %.

Variable Importance

```
vi <- varImp(fit_rf)$importance
vi[head(order(unlist(vi), decreasing = TRUE), 10L), , drop = FALSE]
```

```
##                     Overall
## roll_belt         100.00000
## pitch_forearm      59.32376
## yaw_belt           54.32732
## magnet_dumbbell_z  43.90825
## magnet_dumbbell_y  42.76146
## roll_forearm       42.22110
## pitch_belt         41.95788
## accel_dumbbell_y   21.35141
## magnet_dumbbell_x  16.45689
## roll_dumbbell      16.44099
```

The top ten variables are listed above in the table.

The source of the data

The assignment is based on data of weight lifting exercises. It has been published: 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.