Project 8 - Qualitative Activity Recognition

Yi Lin

May 5, 2018

Background

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 is to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants to predict the quality of a activity so that people can correct their activity and improve their health.

Data Exploration

First of all, load the training and testing data and necessary R libraries.

```
library(mlbench)
library(caret)
library(parallel)
library(doParallel)
train_raw = read.csv("~/Git/myWD/course 8/pml-training.csv")
test_raw = read.csv("~/Git/myWD/course 8/pml-testing.csv")
c(dim(train_raw), dim(test_raw))
```

```
## [1] 19622   160    20   160
```

160 variables represents 4 sensors with eight features on the Euler angles (roll, pitch and yaw). But if we check the NA values of training data, only 406 rows are complete, which is really small size compared to total 19,622 records. If we check NA of each variable, we found 67 is pretty empty, each has 19,216 VAs. Then we decide to only keep those variable with none NA values.

```
sum(complete.cases(train_raw))
```

```
## [1] 406
```

```
na_col <- apply(train_raw, 2, FUN = function(x) length(x[is.na(x)]))
# names(na_col[na_col > 0])  ## 67 columns have 19,216 NA rows
# names(na_col[na_col == 0])  ## 93 columns have full data

train_clean <- train_raw[, names(na_col[na_col == 0])]
```

data source: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har).

Furthermore we only keep the sensor's measurements. And then transform those "factor" columns to "numeric" ones.

```
train <- train_clean[, c(seq(9,92))]    ## 84 cols
# sum((sapply(train, class) == 'factor'))  ## 33 cols
factor_col <- names(train[(sapply(train, class) == 'factor') == TRUE])
train[, factor_col] <- sapply(train[, factor_col], as.numeric)
```

Feature Selection

There are many ways to selection features. We are going to use the filter method based on correlation, using 0.5 as cutoff absolute correlation. It turns out we can remove 62 redundant variables, and keep only 22.

```
set.seed(7)
corMatrix <- cor(train)
highlyCor <- findCorrelation(corMatrix, cutoff = .5)  ## 62 variables
```

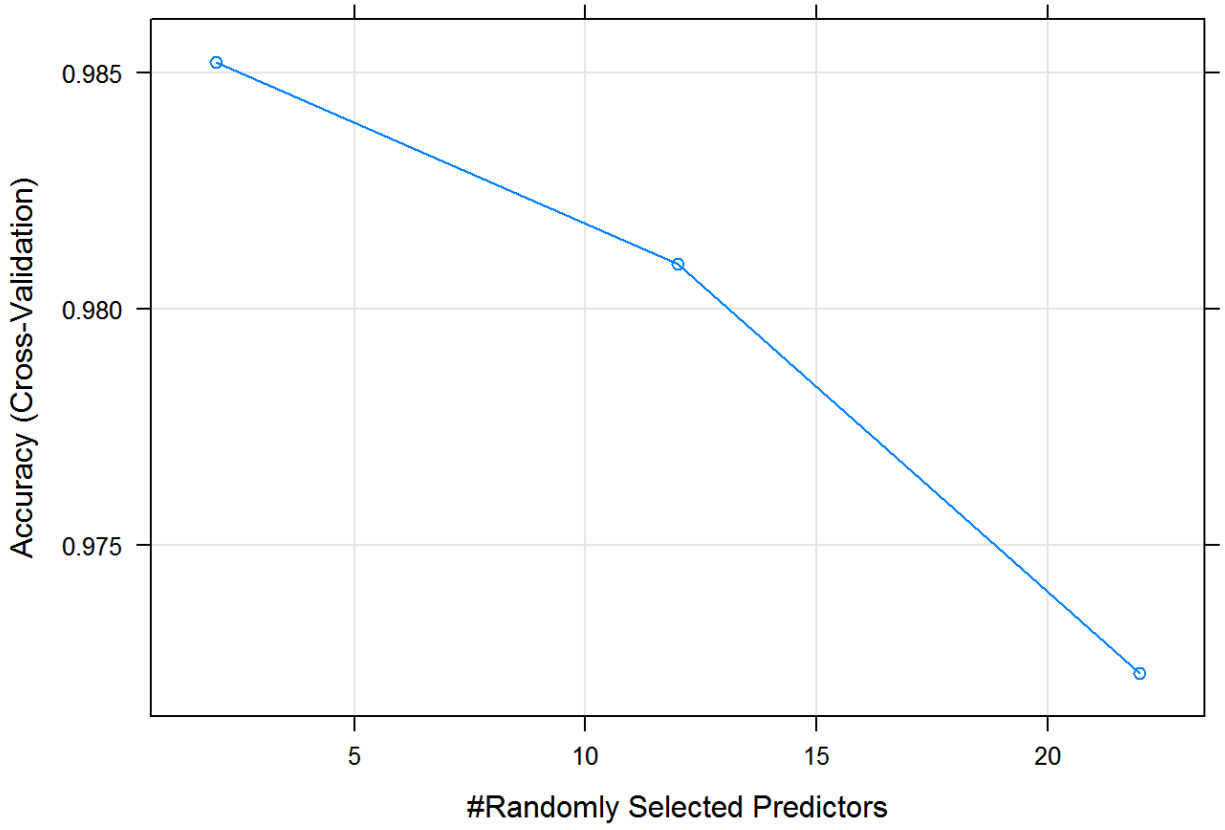
Modeling

Since there are 5 classes of activity pattern, we choose random forest method to train the model. Before doing so we have to complete the data set to include the tagged class of each mesurement.

```
data <- cbind(train[, -highlyCor], train_clean[, 93]);
colnames(data)[23] <- c("classe")
```

We use 5-fold cross-validation. On top of that, we enable parallel capacity to improve the training performance. The Accuracy (average) is 98.52%. It is pretty amazing! We also print a plot to show which predictors is more important in predicting the quality of an activity.

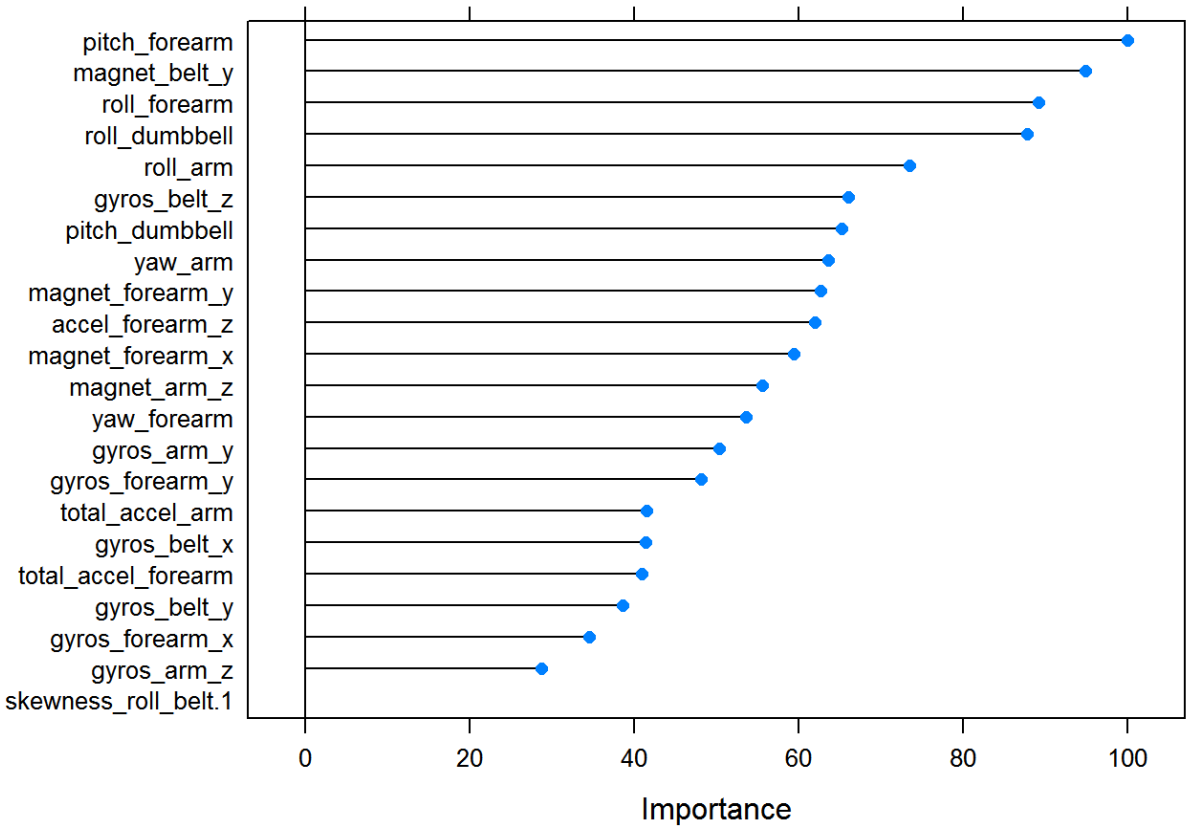
```
cluster <- makeCluster(detectCores() - 1) # convention to leave 1 core for OS
registerDoParallel(cluster)

set.seed(7)
fitControl <- trainControl(method = 'cv', number = 5, allowParallel = TRUE)
fit <- train(classe ~ ., data = data, method = 'rf', trControl = fitControl)

stopCluster(cluster)
registerDoSEQ()

plot(fit)
```




```
confusionMatrix.train(fit)
```

```
## Cross-Validated (5 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##            Reference
## Prediction    A    B    C    D    E
##          A 28.3  0.3  0.0  0.0  0.0
##          B  0.1 19.0  0.2  0.0  0.0
##          C  0.0  0.1 17.1  0.5  0.0
##          D  0.0  0.0  0.1 15.8  0.1
##          E  0.0  0.0  0.0  0.0 18.3
##
##  Accuracy (average) : 0.9852
```

```
plot(varImp(fit))
```



Prediction and Condlusion

The model looks great! Let's perform this on our test data. First of all, we have to pre-process the data in order to apply the model. It turns out that the 1st variable of test data is empty, so I imputed it with the mean of the same varable in train data set.

```
test_raw = read.csv("~/Git/myWD/course 8/pml-testing.csv")
test <- test_raw[, colnames(data[, 1:22])]
test[, 1] = mean(data$skewness_roll_belt.1)

predict(fit, test)
```

```
##  [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
```

The outcome is awesome. I fed it to the QUiz and got 100% correction.Now we are comfortable to conclude that the model we created performs well, and can be used to predict the quality of weight lifts.