Machine learning: Qualitative Activity Recognition of Weight Lifting Exercises

1. Get and clean data

We want to use data form accelerometers to quantify how well 6 participants do weight lifting exercises. Firstly we download and read the datas:

```
# Download data
if (!file.exists("./pml-training.csv")) {
URL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
destfile <- "./pml-training.csv"
download.file(URL, destfile)}
if (!file.exists("./pml-testing.csv")) {
URL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
destfile <- "./pml-testing.csv"
download.file(URL, destfile)}
# Read data
train_origin <- read.csv("pml-training.csv")
test_origin <- read.csv("pml-testing.csv")
train_origin[train_origin==""] <- NA
sum(is.na(train_origin))
```

```
## [1] 1921600
```

```
names(train_origin)[1:7]
```

```
## [1] "X"                    "user_name"            "raw_timestamp_part_1"
## [4] "raw_timestamp_part_2" "cvtd_timestamp"       "new_window"
## [7] "num_window"
```

Secondly we should clean our datas, for We can see there are 1921600 datas are "NA" or empty and the first seven variables are clearly irrelevant with the outcome "classe". Therefore we delete the first seven variables and the variables whose number of "NA" or empty values are above 90%.

```
train <- train_origin[,-1:-7]
test <- test_origin[,-1:-7]
train <- train[, colSums(is.na(train)) <= dim(train)[1]*0.1]
test <- test[, colSums(is.na(test)) <= dim(test)[1]*0.1]
```

2. Use random forests method to fit model

```
dim(train)
```

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

Considering the data set has many variables (53 variables), in order to get good accuracy, we choose random forests method to fit our model. **We expect the out of sample error should be below** 0.5%**.**

```
library(randomForest)
```

```
set.seed(100)
model <- randomForest(classe~.,data=train,ntree =100)
model
```

```
##
## Call:
##  randomForest(formula = classe ~ ., data = train, ntree = 100)
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.33%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 5576    2    1    0    1   0.0007168
## B   13 3781    3    0    0   0.0042139
## C    0   15 3405    2    0   0.0049679
## D    0    0   19 3194    3   0.0068408
## E    0    0    1    5 3601   0.0016634
```

The prediction values are as follow:

```
predict(model,test);
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  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
```

3. Conclusion

We can see that the OOB (out-of-bag) estimate of error rate, which is **the cross-validation error rate generated by the random forest algorithm**, is only 0.33%. We can believe our model satisfy our requirement (out of sample error $< 0.5\%$), for **OOB error rate is a good estimate for the out-of-sample error rate**. And the predictions are reasonalbe.