Human Activity Recognition

Alexander Alexandrov

Overview

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.

This project goal is to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants to predict the manner in which they did the exercise.

The data for this project comes from this source: http://groupware.les.inf.puc-rio.br/har

Exploratory Analysis

```
# Load necessary packages
library(data.table)
library(dplyr)
library(caret)
library(ggplot2)
library(randomForest)
```

Read training data.

```
# Data table is used due to performance purposes
setwd(".")
training <- fread("pml-training.csv")
str(training)
```

```
## Classes 'data.table' and 'data.frame':   19622 obs. of  160 variables:
##  $ V1                       : chr  "1" "2" "3" "4" ...
##  $ user_name                : chr  "carlitos" "carlitos" "carlitos" "carlitos" ...
##  $ raw_timestamp_part_1     : int  1323084231 1323084231 1323084231 1323084232 1323084232 1323084232 132308
##  $ raw_timestamp_part_2     : int  788290 808298 820366 120339 196328 304277 368296 440390 484323 484434 ...
##  $ cvtd_timestamp           : chr  "05/12/2011 11:23" "05/12/2011 11:23" "05/12/2011 11:23" "05/12/2011 11:23
##  $ new_window               : chr  "no" "no" "no" "no" ...
##  $ num_window               : int  11 11 11 12 12 12 12 12 12 12 ...
##  $ roll_belt                : num  1.41 1.41 1.42 1.48 1.48 1.45 1.42 1.42 1.43 1.45 ...
##  $ pitch_belt               : num  8.07 8.07 8.07 8.05 8.07 8.06 8.09 8.13 8.16 8.17 ...
##  $ yaw_belt                 : num  -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 ...
##  $ total_accel_belt         : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ kurtosis_roll_belt       : chr  "" "" "" "" ...
##  $ kurtosis_picth_belt      : chr  "" "" "" "" ...
##  $ kurtosis_yaw_belt        : chr  "" "" "" "" ...
##  $ skewness_roll_belt       : chr  "" "" "" "" ...
##  $ skewness_roll_belt.1     : chr  "" "" "" "" ...
##  $ skewness_yaw_belt        : chr  "" "" "" "" ...
##  $ max_roll_belt            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_picth_belt           : int  NA NA NA NA NA NA NA NA NA NA ...
```

```
##  $ max_yaw_belt           : chr  "" "" "" "" ...
##  $ min_roll_belt          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_pitch_belt         : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_yaw_belt           : chr  "" "" "" "" ...
##  $ amplitude_roll_belt    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ amplitude_pitch_belt   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ amplitude_yaw_belt     : chr  "" "" "" "" ...
##  $ var_total_accel_belt   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ avg_roll_belt          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ stddev_roll_belt       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ var_roll_belt          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ avg_pitch_belt         : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ stddev_pitch_belt      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ var_pitch_belt         : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ avg_yaw_belt           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ stddev_yaw_belt        : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ var_yaw_belt           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ gyros_belt_x           : num  0 0.02 0 0.02 0.02 0.02 0.02 0.02 0.02 0.03 ...
##  $ gyros_belt_y           : num  0 0 0 0 0.02 0 0 0 0 0 ...
##  $ gyros_belt_z           : num  -0.02 -0.02 -0.02 -0.03 -0.02 -0.02 -0.02 -0.02 -0.02 0 ...
##  $ accel_belt_x           : int  -21 -22 -20 -22 -21 -21 -22 -22 -20 -21 ...
##  $ accel_belt_y           : int  4 4 5 3 2 4 3 4 2 4 ...
##  $ accel_belt_z           : int  22 22 23 21 24 21 21 21 24 22 ...
##  $ magnet_belt_x          : int  -3 -7 -2 -6 -6 0 -4 -2 1 -3 ...
##  $ magnet_belt_y          : int  599 608 600 604 600 603 599 603 602 609 ...
##  $ magnet_belt_z          : int  -313 -311 -305 -310 -302 -312 -311 -313 -312 -308 ...
##  $ roll_arm               : num  -128 -128 -128 -128 -128 -128 -128 -128 -128 -128 ...
##  $ pitch_arm              : num  22.5 22.5 22.5 22.1 22.1 22 21.9 21.8 21.7 21.6 ...
##  $ yaw_arm                : num  -161 -161 -161 -161 -161 -161 -161 -161 -161 -161 ...
##  $ total_accel_arm        : int  34 34 34 34 34 34 34 34 34 34 ...
##  $ var_accel_arm          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ avg_roll_arm           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ stddev_roll_arm        : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ var_roll_arm           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ avg_pitch_arm          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ stddev_pitch_arm       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ var_pitch_arm          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ avg_yaw_arm            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ stddev_yaw_arm         : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ var_yaw_arm            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ gyros_arm_x            : num  0 0.02 0.02 0.02 0 0.02 0 0.02 0.02 0.02 ...
##  $ gyros_arm_y            : num  0 -0.02 -0.02 -0.03 -0.03 -0.03 -0.03 -0.02 -0.03 -0.03 ...
##  $ gyros_arm_z            : num  -0.02 -0.02 -0.02 0.02 0 0 0 0 -0.02 -0.02 ...
##  $ accel_arm_x            : int  -288 -290 -289 -289 -289 -289 -289 -289 -288 -288 ...
##  $ accel_arm_y            : int  109 110 110 111 111 111 111 111 109 110 ...
##  $ accel_arm_z            : int  -123 -125 -126 -123 -123 -122 -125 -124 -122 -124 ...
##  $ magnet_arm_x           : int  -368 -369 -368 -372 -374 -369 -373 -372 -369 -376 ...
##  $ magnet_arm_y           : int  337 337 344 344 337 342 336 338 341 334 ...
##  $ magnet_arm_z           : int  516 513 513 512 506 513 509 510 518 516 ...
##  $ kurtosis_roll_arm      : chr  "" "" "" "" ...
##  $ kurtosis_picth_arm     : chr  "" "" "" "" ...
##  $ kurtosis_yaw_arm       : chr  "" "" "" "" ...
##  $ skewness_roll_arm      : chr  "" "" "" "" ...
##  $ skewness_pitch_arm     : chr  "" "" "" "" ...
```

```
##  $ skewness_yaw_arm        : chr  "" "" "" "" ...
##  $ max_roll_arm            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_picth_arm           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_yaw_arm             : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_roll_arm            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_pitch_arm           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_yaw_arm             : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ amplitude_roll_arm      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ amplitude_pitch_arm     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ amplitude_yaw_arm       : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ roll_dumbbell           : num  13.1 13.1 12.9 13.4 13.4 ...
##  $ pitch_dumbbell          : num  -70.5 -70.6 -70.3 -70.4 -70.4 ...
##  $ yaw_dumbbell            : num  -84.9 -84.7 -85.1 -84.9 -84.9 ...
##  $ kurtosis_roll_dumbbell  : chr  "" "" "" "" ...
##  $ kurtosis_picth_dumbbell : chr  "" "" "" "" ...
##  $ kurtosis_yaw_dumbbell   : chr  "" "" "" "" ...
##  $ skewness_roll_dumbbell  : chr  "" "" "" "" ...
##  $ skewness_pitch_dumbbell : chr  "" "" "" "" ...
##  $ skewness_yaw_dumbbell   : chr  "" "" "" "" ...
##  $ max_roll_dumbbell       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_picth_dumbbell      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_yaw_dumbbell        : chr  "" "" "" "" ...
##  $ min_roll_dumbbell       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_pitch_dumbbell      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_yaw_dumbbell        : chr  "" "" "" "" ...
##  $ amplitude_roll_dumbbell : num  NA NA NA NA NA NA NA NA NA NA ...
##   [list output truncated]
##  - attr(*, ".internal.selfref")=<externalptr>
```

- Some of variables contain NAs or empty strings. Such variables should be considered for interpolation or should be removed from further analysis.
- Window can be used to aggregate and reduce training data.

The approach is simple. Start from simplified model on reduced data. Check accuracy. Move to more complicated model if necessary.

Data Cleaning

Exclude columns with lots of NAs. Such columns can't be interpolated so can't be useful in machine learning.

```
predictors.na.stat <- training[, colMeans(is.na(.SD) | .SD == "")]
# Less than 10% NAs
non.na.predictors <- names(training)[melt(predictors.na.stat) < 0.1]
```

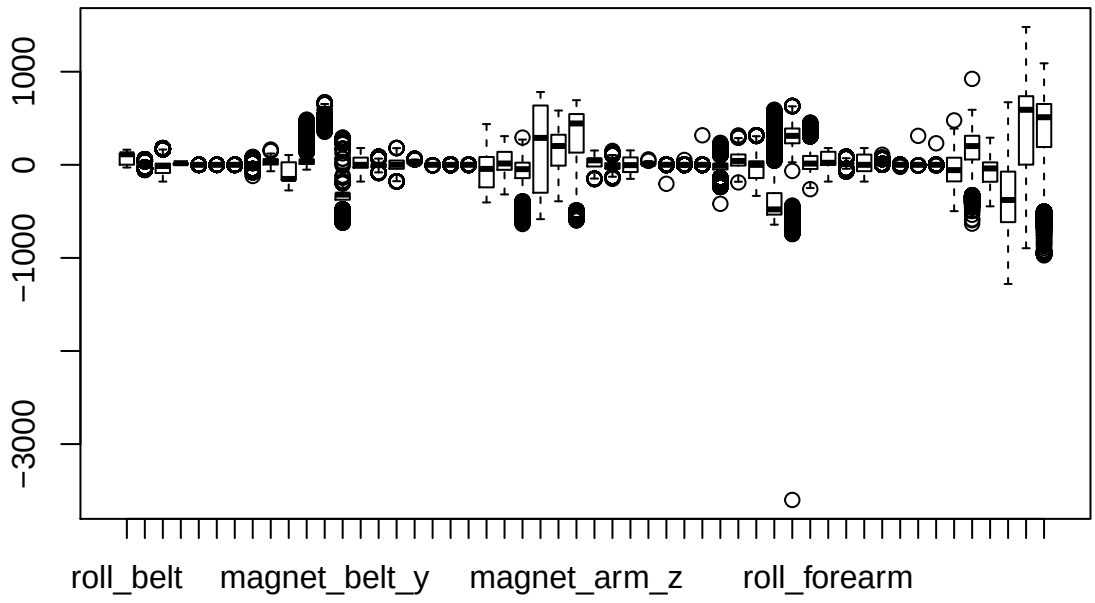
Also predictors with nonrelevant information should be removed too. User name and timestapms can't help to classify activity in common case.

```
nonrelevant.predictors <- c("V1", "user_name", "raw_timestamp_part_1", "raw_timestamp_part_2", "cvtd_ti
relevant.predictors <- non.na.predictors[!(non.na.predictors %in% nonrelevant.predictors)]
relevant.training <- training[, relevant.predictors, with=FALSE]
```

Further Exploratory Analysis

Just before averaging by window outliers should be considered.

```
boxplot(relevant.training[, -c("num_window", "classe"), with=FALSE])
```



According to this plot a lot of predictors contains ourliers (black circles). For example *Box-Cox* can be used to reduce outliers influence.

Final Data Cleaning

Average measurements by window to reduce original data set.

```
cleaned.training <- relevant.training[, lapply(.SD, mean), by=c("num_window", "classe")]
# Exclude window variable
cleaned.training <- cleaned.training[, -"num_window", with=FALSE]
```

Data Preprocessing

Exclude covariate predictors by means of correlation matrix

```
cleaned.measurements <- cleaned.training[, -"classe", with=FALSE]
predictors.cor <- abs(cor(cleaned.measurements))
predictors.cor[upper.tri(predictors.cor, diag=TRUE)] <- 0

# Correlation threshold is 0.8
predictors.cor.coords <- which(predictors.cor > 0.8, arr.ind=TRUE)
predictors.cor.coords.x <- unique(predictors.cor.coords[, "col"])
predictors.cor.coords.y <- unique(predictors.cor.coords[, "row"])

covariate.predictors.indices <- if (length(predictors.cor.coords.x) > length(predictors.cor.coords.y)) {
  predictors.cor.coords.x
} else {
  predictors.cor.coords.y
}

covariate.predictors <- names(cleaned.measurements)[covariate.predictors.indices]

reduced.training <- cleaned.training[, -covariate.predictors, with=FALSE]
```

Random Forest

This method is simple enough to get started and powerful to fit nonlinear case.

```
set.seed(1234)
rf.model.fit <- train(classe ~ ., data=reduced.training, method="rf", ntree=50, trainControl="cv", numbe
rf.model.fit
```

```
## Random Forest
##
## 858 samples
##  37 predictor
##   5 classes: 'A', 'B', 'C', 'D', 'E'
##
## Pre-processing: Box-Cox transformation (4)
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 858, 858, 858, 858, 858, 858, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa      Accuracy SD  Kappa SD
```

```
##    2    0.8402007  0.7973712  0.01411196   0.01781284
##   19    0.8356789  0.7916735  0.01922636   0.02427836
##   37    0.8206720  0.7725927  0.02667905   0.03368141
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 2.
```

Conclusion

The accuracy is greater than *0.8*. So this model works practically fine. To avoid overfitting this model can be choosen as final.

Testing

```
testing <- fread("pml-testing.csv")
predicted <- predict(rf.model.fit, newdata=testing)
cbind(testing[, c("user_name", "cvtd_timestamp"), with=FALSE],
      data.frame(classe=predicted))
```

```
##     user_name   cvtd_timestamp classe
##  1:     pedro 05/12/2011 14:23      C
##  2:    jeremy 30/11/2011 17:11      A
##  3:    jeremy 30/11/2011 17:11      A
##  4:    adelmo 02/12/2011 13:33      A
##  5:    eurico 28/11/2011 14:13      A
##  6:    jeremy 30/11/2011 17:12      E
##  7:    jeremy 30/11/2011 17:12      D
##  8:    jeremy 30/11/2011 17:11      B
##  9:  carlitos 05/12/2011 11:24      A
## 10:   charles 02/12/2011 14:57      A
## 11:  carlitos 05/12/2011 11:24      C
## 12:    jeremy 30/11/2011 17:11      C
## 13:    eurico 28/11/2011 14:14      B
## 14:    jeremy 30/11/2011 17:10      A
## 15:    jeremy 30/11/2011 17:12      E
## 16:    eurico 28/11/2011 14:15      E
## 17:     pedro 05/12/2011 14:22      A
## 18:  carlitos 05/12/2011 11:24      B
## 19:     pedro 05/12/2011 14:23      B
## 20:    eurico 28/11/2011 14:14      B
```