Practical Machine Learning - Human Activity Recognition

Tatjana TD

Sunday, March 22, 2015

This is a report on classification of activity based on human activity recognision data. The source of this data is http://groupware.les.inf.puc-rio.br/har. On this website, it is said that *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)

For the 6 participants data was collected from accelerometers on the belt, forearm, arm and dumbell. The participants perform barbell lifts and they perform it correctly and incorrectly. The investigated question is, if we can predict in which fashion the barbell lifts are performed at a specific point in time.

In this report, the following analysis steps are performed.
exploratory analysis, preprocessing
feature creation, selection
model building
prediction accuracy
accuracy of prediction on test data set

```
library(caret)
download.file("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv", "trainingdata.csv
train <- read.csv("trainingdata.csv", header=T )
## validation data set
download.file("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv", "testdata.csv", qu
testglobal <- read.csv("testdata.csv", header=T )
```

Exploratory analysis

The training dataset contains N_1 =19622 observations and the test / validation dataset has only N_2= 20 observations. The six participants have more or less similar number of data rows.

```
table(train$user_name)
```

```
##
##   adelmo carlitos  charles   eurico   jeremy    pedro
##     3892     3112     3536     3070     3402     2610
```

```
table(train$classe ,train$user_name)
```

```
##
##     adelmo carlitos charles eurico jeremy pedro
##   A   1165      834     899    865   1177   640
##   B    776      690     745    592    489   505
```

```
##   C    750      493     539    489    652   499
##   D    515      486     642    582    522   469
##   E    686      609     711    542    562   497
```

```
str(train$classe)
```

```
##  Factor w/ 5 levels "A","B","C","D",..: 1 1 1 1 1 1 1 1 1 1 ...
```

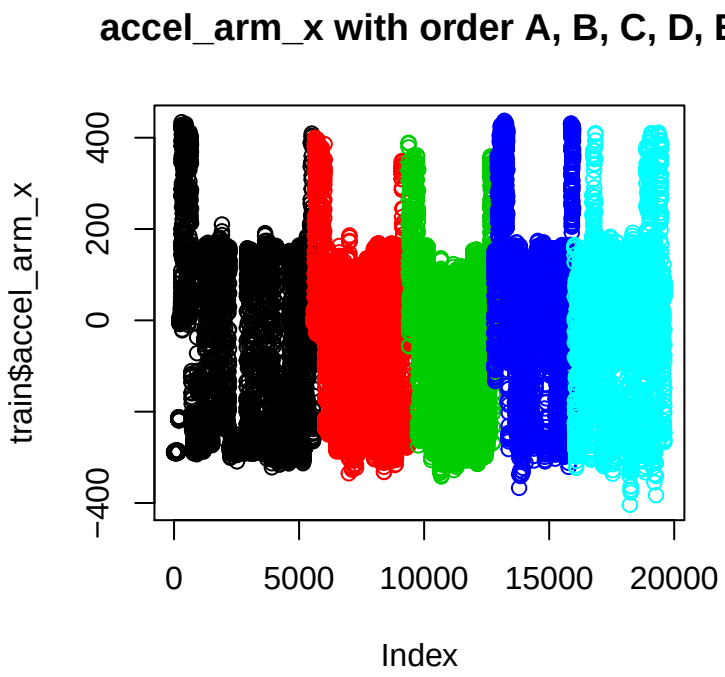
```
table(train$cvtd_timestamp, train$user_name)
```

```
##
##                   adelmo carlitos charles eurico jeremy pedro
##   02/12/2011 13:32    177        0       0      0      0     0
##   02/12/2011 13:33   1321        0       0      0      0     0
##   02/12/2011 13:34   1375        0       0      0      0     0
##   02/12/2011 13:35   1019        0       0      0      0     0
##   02/12/2011 14:56      0        0     235      0      0     0
##   02/12/2011 14:57      0        0    1380      0      0     0
##   02/12/2011 14:58      0        0    1364      0      0     0
##   02/12/2011 14:59      0        0     557      0      0     0
##   05/12/2011 11:23      0      190       0      0      0     0
##   05/12/2011 11:24      0     1497       0      0      0     0
##   05/12/2011 11:25      0     1425       0      0      0     0
##   05/12/2011 14:22      0        0       0      0      0   267
##   05/12/2011 14:23      0        0       0      0      0  1370
##   05/12/2011 14:24      0        0       0      0      0   973
##   28/11/2011 14:13      0        0       0    833      0     0
##   28/11/2011 14:14      0        0       0   1498      0     0
##   28/11/2011 14:15      0        0       0    739      0     0
##   30/11/2011 17:10      0        0       0      0    869     0
##   30/11/2011 17:11      0        0       0      0   1440     0
##   30/11/2011 17:12      0        0       0      0   1093     0
```

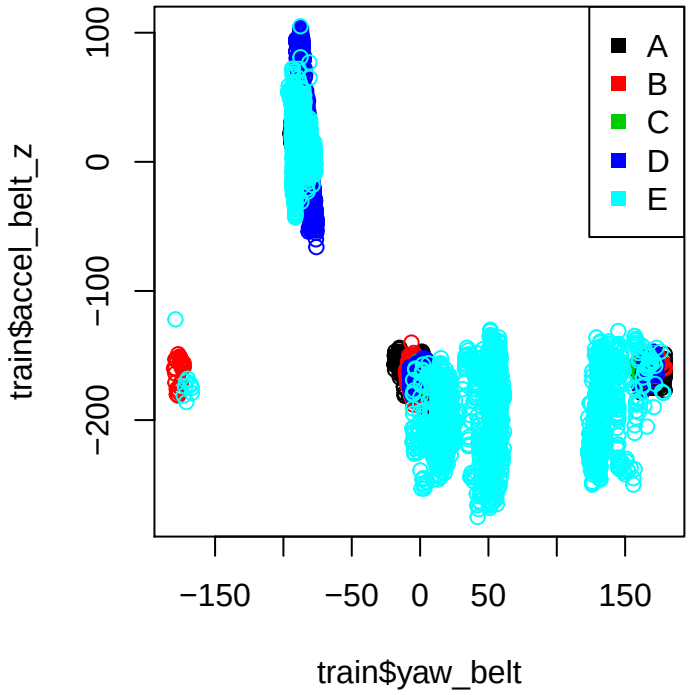
```
length(unique(train$cvtd_timestamp))
```

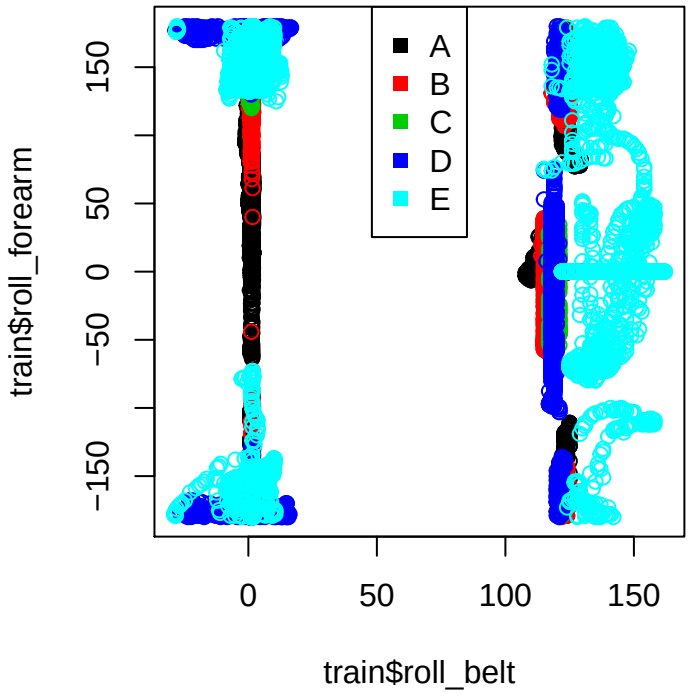
```
## [1] 20
```

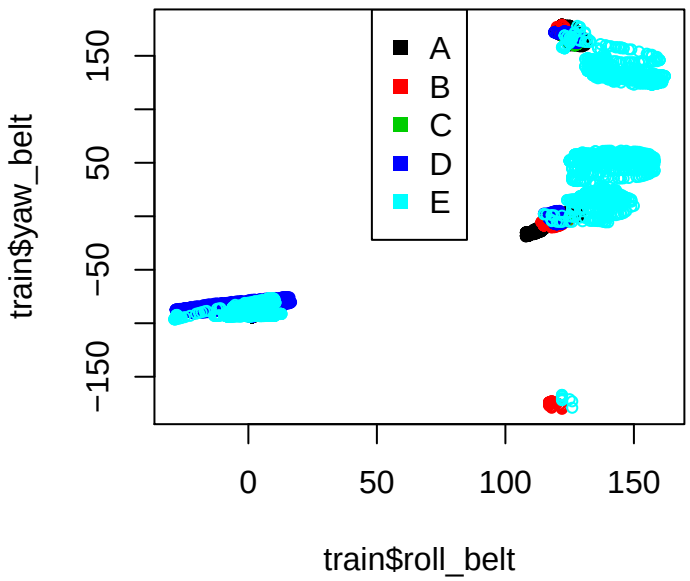
```
plot(train$accel_arm_x, col=train$classe, main="accel_arm_x with order A, B, C, D, E")
```



The data is sorted by time and the participant performed the different barbell lifting ways subsecuently. There are variables (e.g. cvtd_timestamp) in the data set that suggests a time series. However, for each of the 6 participants there are only 3 or 4 different time points and they are all on the same day in the same hour. Observations were made on 3 or 4 consecutive minutes, therefore the time effect cannot be very strong.


train$accel_belt_z
100
0
−100
−200
−150
−50
0
50
150
train$yaw_belt
A
B
C
D
E


train$roll_forearm
150
50
0
−50
−150
0
50
100
150
train$roll_belt
A
B
C
D
E



Preprocessing

To reduce the number of predictors the function **nearZeroVar** was applied. The variables with a small variance that got the value TRUE in the last column of the nsv data frame are removed from further considerations. Also, variables with a high amount of missing values are deleted.
Furthermore, by looking of figure 2 it can be seen that the time determined the activity. The participants conducted the lifting exercises in a certain order, but that is not the right information for predicting the type of activity. So all time variables were removed.

```
nsv <- nearZeroVar(train, saveMetrics=TRUE)
head(nsv, 10)
```

```
##                        freqRatio percentUnique zeroVar   nzv
## X                          1.000     100.00000   FALSE FALSE
## user_name                  1.101       0.03058   FALSE FALSE
## raw_timestamp_part_1       1.000       4.26562   FALSE FALSE
## raw_timestamp_part_2       1.000      85.53155   FALSE FALSE
## cvtd_timestamp             1.001       0.10193   FALSE FALSE
## new_window                47.330       0.01019   FALSE  TRUE
## num_window                 1.000       4.37264   FALSE FALSE
## roll_belt                  1.102       6.77811   FALSE FALSE
## pitch_belt                 1.036       9.37723   FALSE FALSE
## yaw_belt                   1.058       9.97350   FALSE FALSE
```

```
# through out variables with near zero variance
newnames <- rownames(nsv)[!nsv$nzv]; length(newnames)
```

```
## [1] 100
```

```
train <- train[, newnames]; dim(train)
```

```
## [1] 19622   100
```

```
#### removing variables with more than 90% missing values
missvars <- apply(train, 2, function(x) sum(is.na(x))/length(x))
w <- which(missvars > .9);
newnames <- names(missvars[-w]); length(newnames)
```

```
## [1] 59
```

```
newnames <- newnames[-c(1:6)] # also remove time variables 3:5
train <- train[, newnames]
names(train)
```

```
##  [1] "roll_belt"            "pitch_belt"           "yaw_belt"
##  [4] "total_accel_belt"     "gyros_belt_x"         "gyros_belt_y"
##  [7] "gyros_belt_z"         "accel_belt_x"         "accel_belt_y"
## [10] "accel_belt_z"         "magnet_belt_x"        "magnet_belt_y"
## [13] "magnet_belt_z"        "roll_arm"             "pitch_arm"
## [16] "yaw_arm"              "total_accel_arm"      "gyros_arm_x"
## [19] "gyros_arm_y"          "gyros_arm_z"          "accel_arm_x"
## [22] "accel_arm_y"          "accel_arm_z"          "magnet_arm_x"
## [25] "magnet_arm_y"         "magnet_arm_z"         "roll_dumbbell"
## [28] "pitch_dumbbell"       "yaw_dumbbell"         "total_accel_dumbbell"
## [31] "gyros_dumbbell_x"     "gyros_dumbbell_y"     "gyros_dumbbell_z"
## [34] "accel_dumbbell_x"     "accel_dumbbell_y"     "accel_dumbbell_z"
## [37] "magnet_dumbbell_x"    "magnet_dumbbell_y"    "magnet_dumbbell_z"
## [40] "roll_forearm"         "pitch_forearm"        "yaw_forearm"
## [43] "total_accel_forearm"  "gyros_forearm_x"      "gyros_forearm_y"
## [46] "gyros_forearm_z"      "accel_forearm_x"      "accel_forearm_y"
## [49] "accel_forearm_z"      "magnet_forearm_x"     "magnet_forearm_y"
## [52] "magnet_forearm_z"     "classe"
```

After these cleaning steps, only 53 predictor variables are left.

For the cross validation, inside the training data 70 % of the data rows are randomly selected for training and the rest for testing the model.

```
set.seed(44944)
inTrain <- createDataPartition(y=train$classe, p=0.7, list=FALSE)
training <- train[inTrain, ]
testing <- train[-inTrain,]
dim(training); dim(testing)
```

```
## [1] 13737    53
```

```
## [1] 5885   53
```

Feature selection

It is not easy to say until now which of the 53 predictor variables are really important for prediction. Therefore, a first small random sample is used to train a random forest model and then to look at the variable importance.

```
inSelect <- sample(1:nrow(training), 1000, replace=FALSE)
modfit <-  train(y=training$classe[inSelect], x=training[inSelect, -53],   trControl=trainControl(metho
best <- varImp(modfit)
tab <- best$importance;    or <-order(tab$Overall, decreasing = TRUE)
tab$names <- rownames(tab)
tab <- tab[or,]
varnames <- tab$names[1:25] # first best 25 predictors
varnames
```

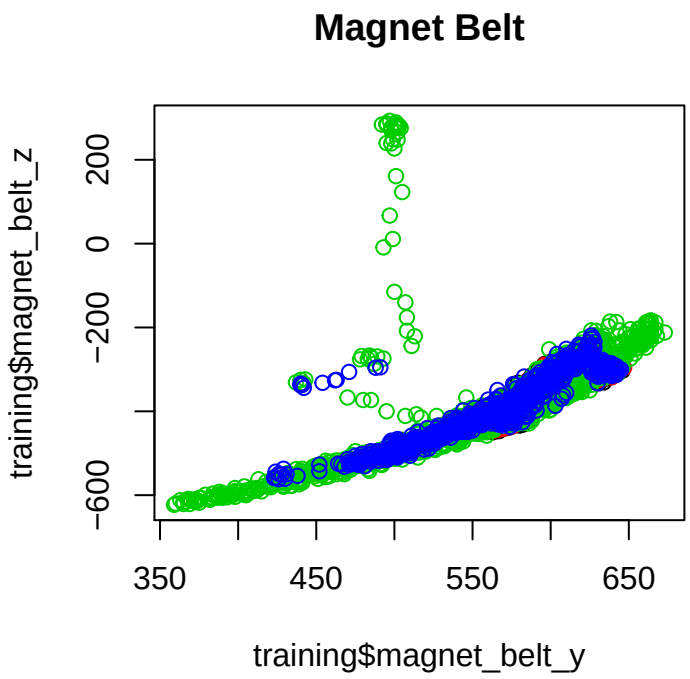
```
##  [1] "roll_belt"            "magnet_dumbbell_z"    "pitch_forearm"
##  [4] "yaw_belt"             "magnet_dumbbell_y"    "roll_dumbbell"
##  [7] "magnet_dumbbell_x"    "roll_forearm"         "magnet_belt_y"
## [10] "pitch_belt"           "accel_belt_z"         "accel_dumbbell_y"
## [13] "magnet_belt_z"        "accel_forearm_x"      "accel_dumbbell_x"
## [16] "accel_arm_x"          "accel_dumbbell_z"     "pitch_dumbbell"
## [19] "yaw_dumbbell"         "magnet_forearm_z"     "magnet_forearm_x"
## [22] "magnet_arm_x"         "total_accel_dumbbell" "gyros_dumbbell_y"
## [25] "gyros_belt_z"
```

Some variables suggest that they measure similar, like *magnet_belt_y* and *magnet_belt_z*. They correlate with almost 0.8.

```
cor(training$magnet_belt_y, training$magnet_belt_z)
```

```
## [1] 0.7756
```

```
plot(training$magnet_belt_y, training$magnet_belt_z, col=train$classe, main="Magnet Belt")
```



From the figure 4 it can be seen, that only one (*magnet_belt_z*) of the two variables would suffice.

model building

```
modfit <-  train(y=training$classe, x=training[, varnames],   trControl=trainControl(method="cv", numbe
modfit
```

```
## Random Forest
##
## 13737 samples
##    25 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
##
## Summary of sample sizes: 9157, 9160, 9157
##
## Resampling results
##
##   Accuracy  Kappa  Accuracy SD  Kappa SD
##   0.9889    0.986  0.001282     0.001621
##
## Tuning parameter 'mtry' was held constant at a value of 5
##
```

```r
pr <- predict(modfit, newdata=testing[,varnames])
confusionMatrix(table(pr, testing$classe))
```

```
## Confusion Matrix and Statistics
##
##
## pr     A    B    C    D    E
##   A 1674    8    0    0    0
##   B    0 1123   10    0    0
##   C    0    8 1015    9    5
##   D    0    0    1  955    1
##   E    0    0    0    0 1076
##
## Overall Statistics
##
##                Accuracy : 0.993
##                  95% CI : (0.99, 0.995)
##     No Information Rate : 0.284
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.991
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             1.000    0.986    0.989    0.991    0.994
## Specificity             0.998    0.998    0.995    1.000    1.000
## Pos Pred Value          0.995    0.991    0.979    0.998    1.000
## Neg Pred Value          1.000    0.997    0.998    0.998    0.999
## Prevalence              0.284    0.194    0.174    0.164    0.184
## Detection Rate          0.284    0.191    0.172    0.162    0.183
## Detection Prevalence    0.286    0.193    0.176    0.163    0.183
## Balanced Accuracy       0.999    0.992    0.992    0.995    0.997
```

prediction on test data set

```
##    problem_id predTest
## 1           1        B
## 2           2        A
## 3           3        B
## 4           4        A
## 5           5        A
## 6           6        E
## 7           7        D
## 8           8        B
## 9           9        A
## 10         10        A
## 11         11        B
## 12         12        C
## 13         13        B
## 14         14        A
```

```
## 15          15        E
## 16          16        E
## 17          17        A
## 18          18        B
## 19          19        B
## 20          20        B
```