ML

Saturday, November 17, 2014

Background

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 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, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. 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). This background paragraph was taken from the Practical Machine Learning Course Project page by Jeff Leek, PhD, Roger D. Peng, PhD, Brian Caffo, PhD

```
#Read files into R
Train <- read.csv("c:/rWork/pml-training.csv", header=T,
                  stringsAsFactors = FALSE)
Test <- read.csv("c:/rWork/pml-testing.csv", header=T,
                 stringsAsFactors = FALSE)
#Set options to no scientific notations and four digits
options(scipen=999, digits=4)
#Activate caret and rpart libraries
library("caret", lib.loc="~/R/win-library/3.1")
```

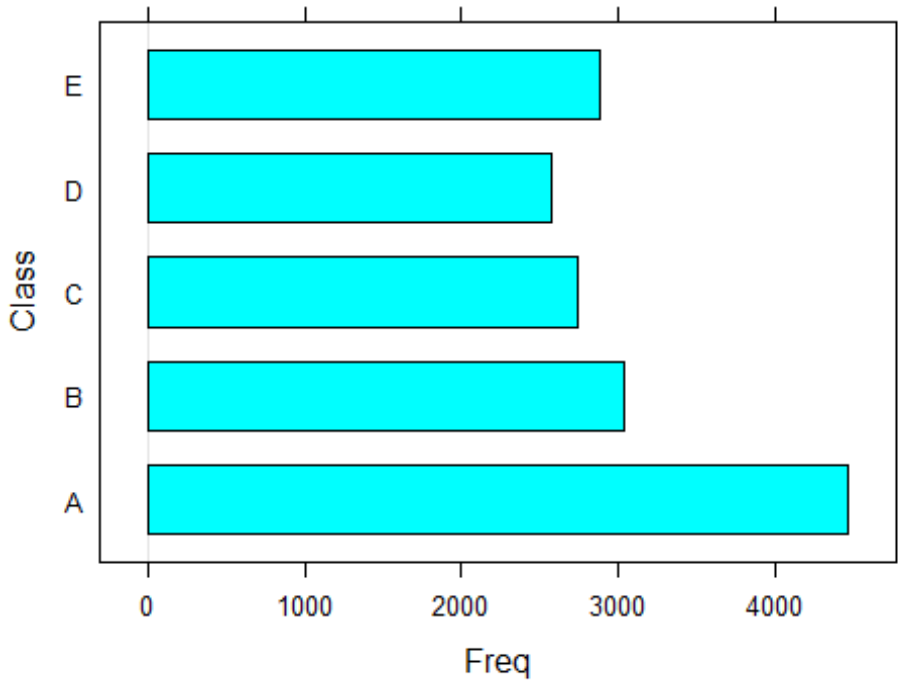
```
## Loading required package: lattice
## Loading required package: ggplot2
```

```
library("rpart", lib.loc="~/R/win-library/3.1")
#Separate Training and Test data
inTrain <- createDataPartition(y=Train$classe, p=.8, list=FALSE)
training <- Train[inTrain,]
trainingTest <- Train[-inTrain,]
dim(training); dim(trainingTest)
```

```
## [1] 15699   160
```

```
## [1] 3923  160
```

```
#display the classe variable graphically
barchart(training$classe, horiz=TRUE, ylab="Class")
```




```
#Fit classe to selected variables
```

Cross-validation

Cross-validation was used by segmenting the Training dataset into two parts: a smaller training dataset(80%) to be used to build the model and a testing dataset(20%) to be used to validate the model. The goal of cross-validation was to estimate the expected level of fit of the model to a data set that is independent of the data that was used to train the model. It was used to predict the manner in which the exercises were performed as described in the previous paragraph.

Sample error

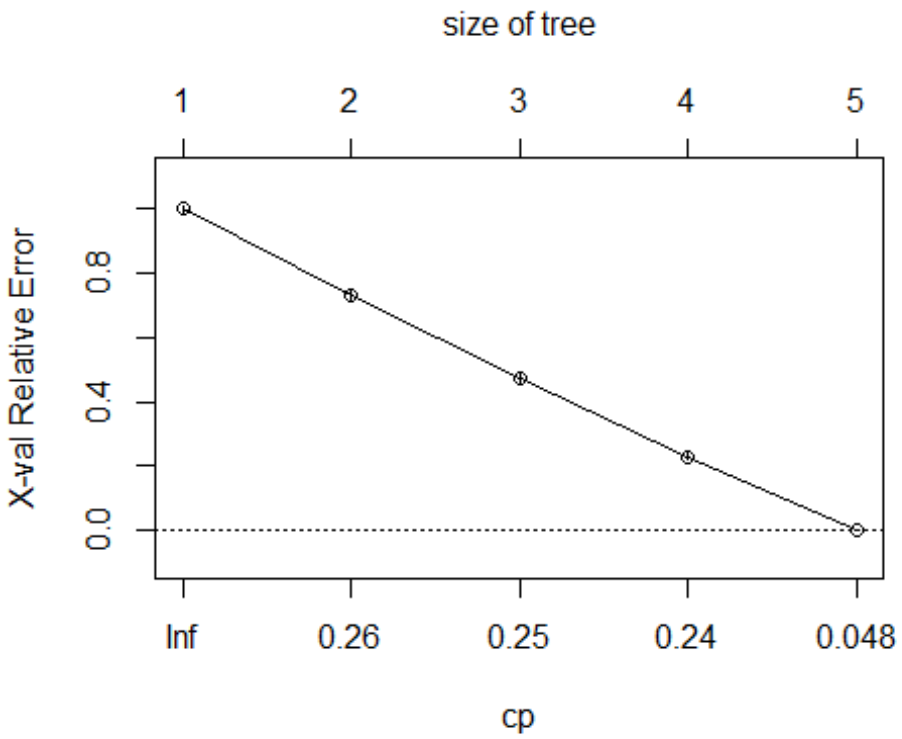
Sample error is reduced by increasing the sample size, in this case 80% of the available data. Out-of-sample error was addressed in the same way by providing 20% of the available data when testing the model.

```
modFit <- rpart(classe~X+accel_arm_z+cvtd_timestamp+gyros_dumbbell_y+
                    magnet_belt_y+total_accel_forearm, method="class",
                data=trainingTest)
```

```r
#Predict the fitted rpart object
pred <- predict(modFit, type = "prob")
# display the results
prcp <- printcp(modFit)

##
## Classification tree:
## rpart(formula = classe ~ X + accel_arm_z + cvtd_timestamp +
gyros_dumbbell_y +
##     magnet_belt_y + total_accel_forearm, data = trainingTest,
##     method = "class")
##
## Variables actually used in tree construction:
## [1] X
##
## Root node error: 2807/3923 = 0.72
##
## n= 3923
##
##     CP nsplit rel error  xerror   xstd
## 1 0.27      0      1.00 1.00000 0.0101
## 2 0.26      1      0.73 0.72996 0.0111
## 3 0.24      2      0.47 0.47346 0.0106
## 4 0.23      3      0.23 0.22978 0.0083
## 5 0.01      4      0.00 0.00071 0.0005
```

```
# visualize cross-validation results
plcp <- plotcp(modFit)
```




```
#detailed summary of splits
summ <- summary(modFit)

## Call:
## rpart(formula = classe ~ X + accel_arm_z + cvtd_timestamp +
gyros_dumbbell_y +
##     magnet_belt_y + total_accel_forearm, data = trainingTest,
##     method = "class")
##   n= 3923
##
##       CP nsplit rel error    xerror      xstd
## 1 0.2704      0    1.0000 1.0000000 0.0100670
## 2 0.2569      1    0.7296 0.7299608 0.0111456
## 3 0.2437      2    0.4727 0.4734592 0.0105608
## 4 0.2291      3    0.2291 0.2297827 0.0082705
## 5 0.0100      4    0.0000 0.0007125 0.0005037
##
## Variable importance
##                  X      cvtd_timestamp       magnet_belt_y
##                 58                  29                   6
##    gyros_dumbbell_y         accel_arm_z total_accel_forearm
##                  4                   2                   2
##
## Node number 1: 3923 observations,    complexity param=0.2704
```

```
##   predicted class=A  expected loss=0.7155  P(node) =1
##     class counts:  1116   759   684   643   721
##    probabilities: 0.284 0.193 0.174 0.164 0.184
##   left son=2 (1116 obs) right son=3 (2807 obs)
##   Primary splits:
##       X                < 5580   to the left,  improve=998.90, (0 missing)
##       cvtd_timestamp   splits as  LLLRLLRRLLRLLRLLRLLR, improve=594.90, (0
missing)
##       magnet_belt_y    < 556.5  to the right, improve=181.40, (0 missing)
##       gyros_dumbbell_y < 0.57   to the left,  improve= 99.09, (0 missing)
##       accel_arm_z      < -203.5 to the right, improve= 42.40, (0 missing)
##   Surrogate splits:
##       cvtd_timestamp      splits as  LLRRLLRRLRRLRRLRRLRR, agree=0.884,
adj=0.591, (0 split)
##       total_accel_forearm < 13.5   to the left,  agree=0.730, adj=0.051,
(0 split)
##
## Node number 2: 1116 observations
##   predicted class=A  expected loss=0  P(node) =0.2845
##     class counts:  1116     0     0     0     0
##    probabilities: 1.000 0.000 0.000 0.000 0.000
##
## Node number 3: 2807 observations,    complexity param=0.2569
##   predicted class=B  expected loss=0.7296  P(node) =0.7155
##     class counts:     0   759   684   643   721
##    probabilities: 0.000 0.270 0.244 0.229 0.257
##   left son=6 (759 obs) right son=7 (2048 obs)
##   Primary splits:
##       X                < 9379   to the left,  improve=738.80, (0 missing)
##       cvtd_timestamp   splits as  -LLR-LRR-LR-LR-LR-LR, improve=485.40, (0
missing)
##       magnet_belt_y    < 555.5  to the right, improve=151.80, (0 missing)
##       gyros_dumbbell_y < 0.57   to the left,  improve= 62.64, (0 missing)
##       accel_arm_z      < -196.5 to the right, improve= 22.18, (0 missing)
##   Surrogate splits:
##       cvtd_timestamp   splits as  -LRR-LRR-LR-RR-RR-RR, agree=0.834,
adj=0.387, (0 split)
##       accel_arm_z      < -514   to the left,  agree=0.732, adj=0.009, (0
split)
##       gyros_dumbbell_y < -0.745 to the left,  agree=0.730, adj=0.001, (0
split)
##
## Node number 6: 759 observations
##   predicted class=B  expected loss=0  P(node) =0.1935
##     class counts:     0   759     0     0     0
##    probabilities: 0.000 1.000 0.000 0.000 0.000
##
## Node number 7: 2048 observations,    complexity param=0.2437
##   predicted class=E  expected loss=0.6479  P(node) =0.522
##     class counts:     0     0   684   643   721
```

```
##    probabilities: 0.000 0.000 0.334 0.314 0.352
##   left son=14 (1327 obs) right son=15 (721 obs)
##   Primary splits:
##       X                < 16010  to the left,  improve=701.00, (0 missing)
##       cvtd_timestamp   splits as  --LR--LR-LR-LR-LR-LR, improve=370.60, (0
missing)
##       magnet_belt_y    < 580.5  to the right, improve=153.40, (0 missing)
##       gyros_dumbbell_y < 0.62   to the left,  improve= 88.91, (0 missing)
##       accel_arm_z      < -196.5 to the right, improve= 27.64, (0 missing)
##   Surrogate splits:
##       cvtd_timestamp      splits as  --LR--LR-LL-LR-LR-LL, agree=0.791,
adj=0.406, (0 split)
##       magnet_belt_y       < 577.5  to the right, agree=0.765, adj=0.333,
(0 split)
##       gyros_dumbbell_y    < 0.57   to the left,  agree=0.722, adj=0.209,
(0 split)
##       accel_arm_z         < -218.5 to the right, agree=0.672, adj=0.069,
(0 split)
##       total_accel_forearm < 53.5   to the left,  agree=0.652, adj=0.012,
(0 split)
##
## Node number 14: 1327 observations,    complexity param=0.2291
##   predicted class=C  expected loss=0.4846  P(node) =0.3383
##     class counts:     0     0   684   643     0
##    probabilities: 0.000 0.000 0.515 0.485 0.000
##   left son=28 (684 obs) right son=29 (643 obs)
##   Primary splits:
##       X                < 12800  to the left,  improve=662.90, (0 missing)
##       cvtd_timestamp   splits as  --LR--R--LR-LR-RR-LR, improve=226.00, (0
missing)
##       magnet_belt_y    < 554    to the right, improve= 32.81, (0 missing)
##       gyros_dumbbell_y < 0.425  to the left,  improve= 18.40, (0 missing)
##       accel_arm_z      < 154.5  to the left,  improve= 12.05, (0 missing)
##   Surrogate splits:
##       cvtd_timestamp      splits as  --LR--R--LR-LR-LR-LR, agree=0.788,
adj=0.563, (0 split)
##       gyros_dumbbell_y    < -0.025 to the right, agree=0.573, adj=0.120,
(0 split)
##       magnet_belt_y       < 554    to the right, agree=0.560, adj=0.092,
(0 split)
##       accel_arm_z         < -0.5   to the left,  agree=0.539, adj=0.048,
(0 split)
##       total_accel_forearm < 46.5   to the left,  agree=0.532, adj=0.034,
(0 split)
##
## Node number 15: 721 observations
##   predicted class=E  expected loss=0  P(node) =0.1838
##     class counts:     0     0     0     0   721
##    probabilities: 0.000 0.000 0.000 0.000 1.000
##
```

```
## Node number 28: 684 observations
##   predicted class=C  expected loss=0  P(node) =0.1744
##     class counts:     0     0   684     0     0
##    probabilities: 0.000 0.000 1.000 0.000 0.000
##
## Node number 29: 643 observations
##   predicted class=D  expected loss=0  P(node) =0.1639
##     class counts:     0     0     0   643     0
##    probabilities: 0.000 0.000 0.000 1.000 0.000
```