Machine Learning Assignment

MBJ

Wednesday, August 20, 2014

Summary

Accelerometer data from 6 participants using barbells was used to build predictors describing the type of barbell lift performed. It was determined (of the two approached evaluated) that boosting trees provided the best performance on the cross-validation set of 0.96 accuracy.

Background

Devices such as Jawbone Up, Nike FuelBand, and Fitbit permit the collection of large quantities of data about personal activity relatively inexpensively using accelerometers. Data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants (courtesy of http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har) in which each participant was asked to perform barbell lifts correctly and incorrectly in 5 different ways was used. The data includes a large number of variables from accelerometers and the "classe" variable which indicates the type of lifts performed.

Class Description

A exactly according to the specification,

B throwing the elbows to the front,

C lifting the dumbbell only halfway,

D lowering the dumbbell only halfway and

E throwing the hips to the front.

Methods

The training data for this project was downloaded from:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv)

The test data for this project was downloaded from:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)

Credit for this data: http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har).

```
trainset.raw <- read.csv("pml-training.csv",header=TRUE,na.strings=c("", "NA", "NULL"))

# remove columns extraneous to the analysis
delvars = c("X","user_name","raw_timestamp_part_1","raw_timestamp_part_2","cvtd_timestamp","new
_window","num_window")
myvars <- names(trainset.raw) %in% delvars
trainset.edited <- trainset.raw[!myvars]

# number cases
nbrcases.edited = dim(trainset.edited)[1]
nbrcols.edited  = dim(trainset.edited)[1]

# remove variable with less than 5% availability
colprcntminbool = ((100.0*colSums(!is.na(trainset.edited)))/nbrcases.edited) > 5.0
trainset.edited <- trainset.edited[colprcntminbool]
nbrcols.trimmed  <- dim(trainset.edited)[2]

# compute the correlation matrix and find how many with strong correlation
fctr = c("classe")
myvars <- names(trainset.edited) %in% fctr
cortrain <- trainset.edited[!myvars]
cormatrix = cor(cortrain)
cid = which(cormatrix > 0.9,arr.ind=TRUE)
indx = c(1:(length(cid)[1]/2))
cid[!(cid[indx,1]==cid[indx,2]),]
```

```
##                  row col
## total_accel_belt   4   1
## accel_belt_y       9   1
## roll_belt          1   4
## accel_belt_y       9   4
## roll_belt          1   9
## total_accel_belt   4   9
## gyros_forearm_z   46  33
## gyros_dumbbell_z  33  46
```

```
# parition data
in.train <- createDataPartition(trainset.edited$classe, p=.60, list=FALSE)
train <- trainset.edited[ in.train,]
test  <- trainset.edited[-in.train,]
remove(trainset.edited)
```

The data was read (with na.strings options used), and edited by removing columns unrelated to the prediction task (e.g. time, date) and those columns with less than 5% of the rows with data resulting in a reduction of variables from 19622 to 53. The possibility of eliminating columns that were strong correlated was explored and it was determinted that only 4 sets of relationships had correlations greater than 0.9. After this the data was partitioned into training and cross-validation sets (60/40).

Two approaches were explored - linear discrimant analysis, and boosted trees. Random forests was considered and even run with subsets of the data since it is most likely a good performer on this data (given the large number of variables). However, given the long run time of this algorithm with the settings selected and the good performance of the other methods - it was dropped for this assignment.

Results

```
# all variables numeric
fol <- formula(classe ~ .)

# linear discriminant analysis
mod.lda <- train(classe ~ ., data=train, method="lda")
```

```
## Loading required package: MASS
```

```
## Warning: package 'MASS' was built under R version 3.1.0
## Warning: package 'e1071' was built under R version 3.0.3
```

```
pred.lda <- predict(mod.lda, test)
conf.lda = confusionMatrix(pred.lda, test$classe)
conf.lda
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1834  242  156   80   58
##          B   47  976  140   56  268
##          C  159  176  886  141  130
##          D  187   66  158  961  140
##          E    5   58   28   48  846
##
## Overall Statistics
##
##                Accuracy : 0.701
##                  95% CI : (0.691, 0.711)
##     No Information Rate : 0.284
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.622
##  Mcnemar's Test P-Value : <2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             0.822    0.643    0.648    0.747    0.587
## Specificity             0.905    0.919    0.906    0.916    0.978
## Pos Pred Value          0.774    0.656    0.594    0.636    0.859
## Neg Pred Value          0.927    0.915    0.924    0.949    0.913
## Prevalence              0.284    0.193    0.174    0.164    0.184
## Detection Rate          0.234    0.124    0.113    0.122    0.108
## Detection Prevalence    0.302    0.190    0.190    0.193    0.126
## Balanced Accuracy       0.863    0.781    0.777    0.832    0.782
```

```
# Stochastic Gradient Boosting
mod.gbm <- train(classe ~ ., data=train, method="gbm",verbose = FALSE)
```

```
## Loading required package: gbm
```

```
## Warning: package 'gbm' was built under R version 3.0.3
```

```
## Loading required package: survival
```

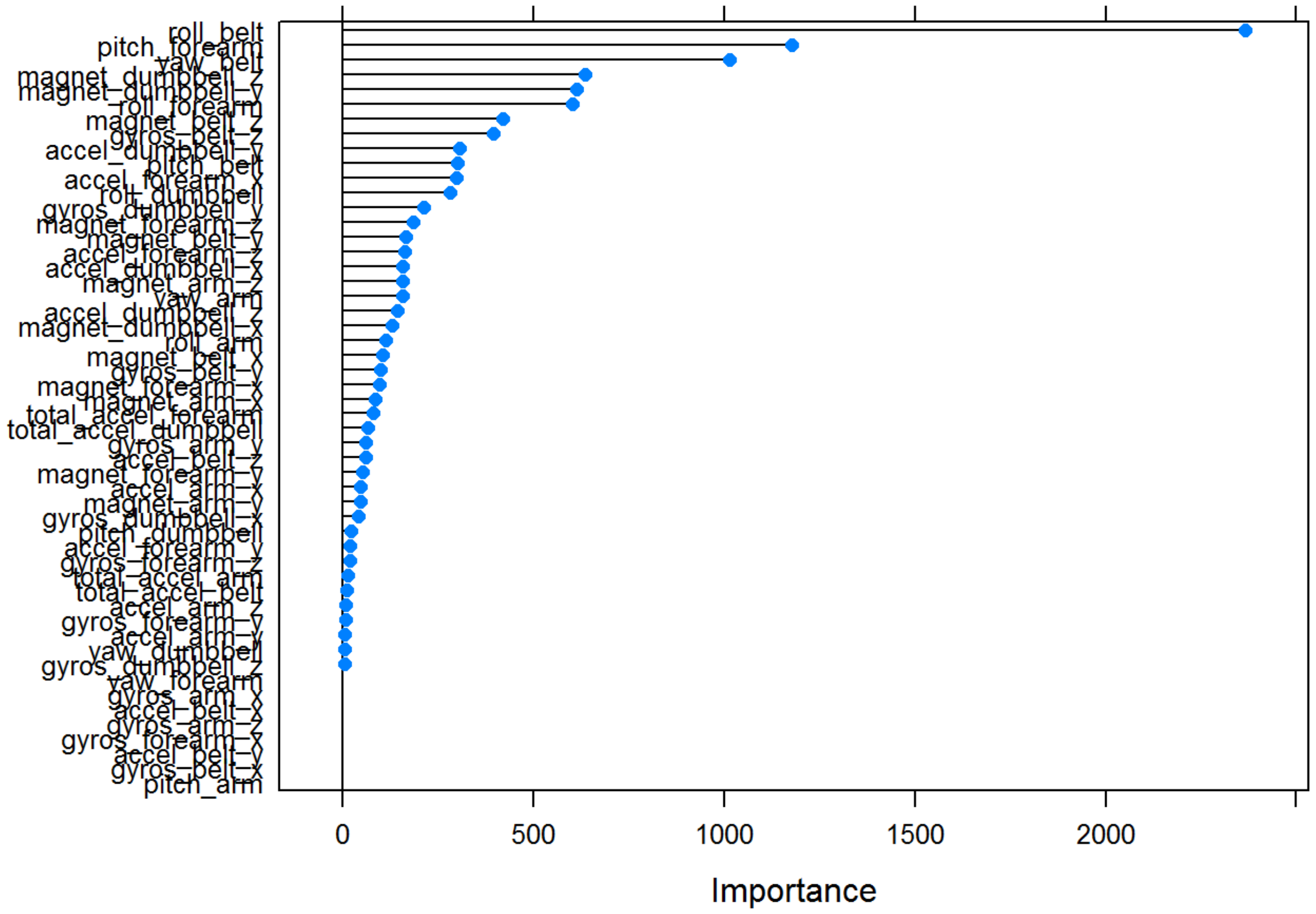
```
## Warning: package 'survival' was built under R version 3.1.0
## Loading required package: splines
##
## Attaching package: 'survival'
##
## The following object is masked from 'package:caret':
##
##     cluster
##
## Loading required package: parallel
## Loaded gbm 2.1
## Loading required package: plyr
```

```
## Warning: package 'plyr' was built under R version 3.1.0
```

```
pred.gbm <- predict(mod.gbm, test)
conf.gbm = confusionMatrix(pred.gbm, test$classe)
conf.gbm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2209   49    0    1    5
##          B   17 1438   52    8   13
##          C    5   29 1301   46   10
##          D    1    1   14 1224   18
##          E    0    1    1    7 1396
##
## Overall Statistics
##
##                Accuracy : 0.965
##                  95% CI : (0.96, 0.969)
##     No Information Rate : 0.284
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.955
##  Mcnemar's Test P-Value : 1.9e-12
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             0.990    0.947    0.951    0.952    0.968
## Specificity             0.990    0.986    0.986    0.995    0.999
## Pos Pred Value          0.976    0.941    0.935    0.973    0.994
## Neg Pred Value          0.996    0.987    0.990    0.991    0.993
## Prevalence              0.284    0.193    0.174    0.164    0.184
## Detection Rate          0.282    0.183    0.166    0.156    0.178
## Detection Prevalence    0.289    0.195    0.177    0.160    0.179
## Balanced Accuracy       0.990    0.967    0.969    0.973    0.983
```

```
vars.gbm = varImp(mod.gbm, scale = FALSE)
print(plot(vars.gbm))
```



The out-of-sample error/accuracy with the cross-validation set from confusion matrix gave very good results for the boosted tree approach. The top 3 variables of importance can be observed to be roll_belt, pitch_forearm and yaw_belt.

Conclusion

The linear discriminant analysis provides a cross-validation (CV) accuracy of approx. 0.7. The Stochastic Gradient Boosting provided a very good level of CV accuracy (0.96) of predicting the classes A-E from accelerometers measurements from the 6 participants. Lastly, the data was read in for the 20 test cases and predictions for the different approaches determined, with small differences between the two methods - with the gradient boosting matching the solution.

```
testdata = read.csv("pml-testing.csv", na.strings=c("", "NA", "NULL"))
predictions.lda <- predict(mod.lda, testdata)
predictions.lda
```

```
##  [1] B A B C C E D D A A D A B A E A A B B B
## Levels: A B C D E
```

```
predictions.gbm <- predict(mod.gbm, testdata)
predictions.gbm
```

```
##  [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
```