08ML_Course_Project

12/14/2017

Project Summary

This project evaluates accelerometer data and two human activity recognition models (stochastic gradient boosting and random forest packages in R) to predict how well a weight lifting exercise is performed.

Weight Lifting Exercises Dataset

Six healthy male participants aged between 20-28 years were asked to perform unilateral dumbbell biceps curls correctly and incorrectly in 5 different ways:

Class	Activity
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
E	throwing the hips to the front

Data from accelerometers on the belt, forearm, arm, and dumbell are in the raw training (trdata) and testing (tsdata) sets. The traning data also includes the “classe” factor variable for activity classification.

Additional information: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har)

Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human ’13) . Stuttgart, Germany: ACM SIGCHI, 2013.

Data Preprocessing

```
#load training and test data as data frames
library(data.table); library(caret)
trdata <- read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv")
tsdata <- read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv")

#viasualize missing values
#library(Amelia)
#missmap(trdata, col=c("black", "grey"), legend=FALSE)

#remove zero covariates and variables with significant NAs
tr <- trdata[,-nearZeroVar(tsdata)]
ts <- tsdata[,-nearZeroVar(tsdata)]

#remove irrelevant variables (first six: X, user_name, raw_timestamp_part_1, raw_timestamp_part_2, cvtd_timestamp, num_window)
tr <- tr[,-(1:6)]
ts <- ts[,-(1:6)]

#partition training data (tr) for training and testing
inTrain <- createDataPartition(y=tr$classe, p=0.7, list=FALSE)
training <- tr[inTrain,] #dim 13737x53
testing <- tr[-inTrain,] #dim 5885x53
```

Model Training

Stochastic gradient boosting and random forest (gbm and randomForest packages) models were chosen because they work well with large classification data sets. Feature selection and k-fold cross-validation (5-fold) were performed with train() and trainControl() functions in caret package.

```
library(gbm); library(randomForest); set.seed(3433)

# train the data using Stochastic Gradient Boosting (gbm) & Random forest (rf)
tc <- trainControl(method="cv", 5)
garbage <- capture.output( #suppress gbm iteration print output
gbm.fit <- train(classe ~., method="gbm", data=training, trControl=tc))
gbm.fit
```

```
## Stochastic Gradient Boosting 
## 
## 13737 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E' 
## 
## No pre-processing
## Resampling: Cross-Validated (5 fold) 
## Summary of sample sizes: 10989, 10990, 10991, 10989, 10989 
## Resampling results across tuning parameters:
## 
##   interaction.depth  n.trees  Accuracy   Kappa    
##   1                   50      0.7522033  0.6858290
##   1                  100      0.8198307  0.7719451
##   1                  150      0.8531718  0.8141765
##   2                   50      0.8573205  0.8192396
##   2                  100      0.9057291  0.8806968
##   2                  150      0.9290233  0.9101715
##   3                   50      0.8982317  0.8711376
##   3                  100      0.9419820  0.9265887
##   3                  150      0.9604720  0.9499923
## 
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
## 
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were n.trees = 150,
##  interaction.depth = 3, shrinkage = 0.1 and n.minobsinnode = 10.
```

```
rf.fit <- train(classe ~., method="rf", data=training, trControl=tc)
rf.fit
```

```
## Random Forest
##
## 13737 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 10991, 10990, 10989, 10989, 10989
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9909729  0.9885802
##   27    0.9892985  0.9864622
##   52    0.9804171  0.9752265
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

Model Testing

Based on the test set prediction accuracies, expected out-of-sample errors are less than 0.5% for both models.

```
# use the models to predict the results on the testing set
gbm.pred <- predict(gbm.fit, testing)
rf.pred <- predict(rf.fit, testing)

# model accuracy comparison on the test set
gbm.acc <- confusionMatrix(testing$classe, gbm.pred)$overall[1] #0.9647887
rf.acc <- confusionMatrix(testing$classe, rf.pred)$overall[1] #0.9975716
c(gbm=gbm.acc, rf=rf.acc)
```

```
## gbm.Accuracy  rf.Accuracy
##    0.9646559    0.9915038
```

Model Testing on Final Testing Data

Random forest model had higher accuracy on the test set (testing) than the gbm model and was chosen for the final testing on 20 quiz sample data (ts). All answers were correct.

```
# use models to predict results on the validation data set
rf.pred.ts <- predict(rf.fit, ts)
rf.pred.ts
```

```
##  [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
```