Human Activity Recognition Analysis

H Hansen

4/14/2017

Executive Summary

This is a project for the Machine Learning Course for the Data Science certificate program. The goal of the project was to develop a machine learning alorgythm that would correctly identify exersizes that work perfomed properly. The data set consists of a the following:

"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)."(Velloso, 2013)

Devies such as gyros and accelerameters measured key movments of the participant and weight. These measurements were deivideed into a training set and test test. Models were fitted against the classe viariable, which defined wether the movement was perfectly perforect or contained some error. The clasification models chosen were a classification tree (i.e CART/rpart), random forest, boost, and bagging.

The CART model performed poorly. The other models had better out of sample error(OSE) and performance. The the end, the ndom forest, boost, and bagging performed equally, althought techinically the boost model had the best accuracy statitics.

Data Processing

The data summary indicated that some variable were not direct measures variable. These were identifed and moved. Additianally, a training and valiation set were used for cross validation. The validation set will be used to measure OSE.

```
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

```
#read in data
train <- read.csv("/Users/hunterhansen/OneDrive/coursera/8-Machine\ Learning/pml-training.csv", stringsAsFactors=FALSE)
test <-  read.csv("/Users/hunterhansen/OneDrive/coursera/8-Machine\ Learning/pml-testing.csv", stringsAsFactors=FALSE)

# summarize data
summ<- function(x){
dim(x)
sapply(x, class)
summary(x)
```

```
str(x)
}

dim(train);  dim(test)

## [1] 19622   160

## [1]  20 160

## preprocessing

# change classe to factor
train$classe<- as.factor(train$classe)

## clean the data

#take out non-activity data

training<- train[,c(8:ncol(train))]

# calculate % of NA by col
boo<- apply(training, 2, function(col)sum(is.na(col))/length(col))

#find all col w a lot of NA
result <- matrix()
for (i in 1:length(boo)){
  if (boo[i] > 0){
    result <- c(result,names(boo[i]))
  }
}

#delte all col with high NA %
foo<- training[ , -which(names(training) %in% result)]

# deleted all col with class char

final<- foo[,-which(sapply(foo, class) == "character")]
##final <- final[,c(3:ncol(final))]
training <- final

#take out non-activity data
testing<- test[,c(8:ncol(test))]

# calculate % of NA by col
boo<- apply(testing, 2, function(col)sum(is.na(col))/length(col))

#find all col w a lot of NA
result <- matrix()
for (i in 1:length(boo)){
  if (boo[i] > 0){
```

```r
    result <- c(result,names(boo[i]))
  }
}

#delte all col with high NA %
foo<- testing[ , -which(names(testing) %in% result)]

testing <- foo

# create training and validation sets
inTrain <- createDataPartition(y=training$classe, p=0.7, list=FALSE)
train <- training[inTrain,]
val <- training[-inTrain,]
```

Building the Models

Since the outcome variable (i.e classe) was a factor variable, a classifcation model was selcted. Using confusion matrixies, the models were evaluated. Finally all were compared, using the accuracy statistics. While the boost model preformed best.

```r
## build models
#1 classification tree
ModelZero <- train(classe ~ .,data=train, method="rpart")
```

```
## Loading required package: rpart
```

```r
plot(ModelZero$finalModel, uniform=TRUE, main="Classification Tree")
text(ModelZero$finalModel, use.n=TRUE, all=TRUE, cex=.8) # plot 1
```

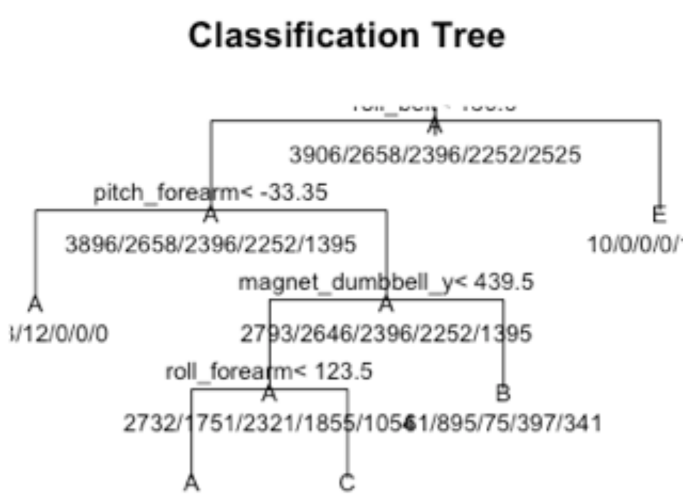


Fig. 1 Classifcation Tree for ModelUno

```r
pd <- predict(ModelZero, val)
```

```
cfm<- confusionMatrix(pd, val$classe)
cfm$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1548  463  471  440  157
##          B   20  391   33  171  145
##          C  102  285  522  353  279
##          D    0    0    0    0    0
##          E    4    0    0    0  501
```

```
#2 random forest model
set.seed(123)
ModelUno<- train(classe ~ ., method = "rf",     data = train, importance = T,     trControl = trainControl(method = "
cv", number = 3))
```

```
## Loading required package: randomForest
```

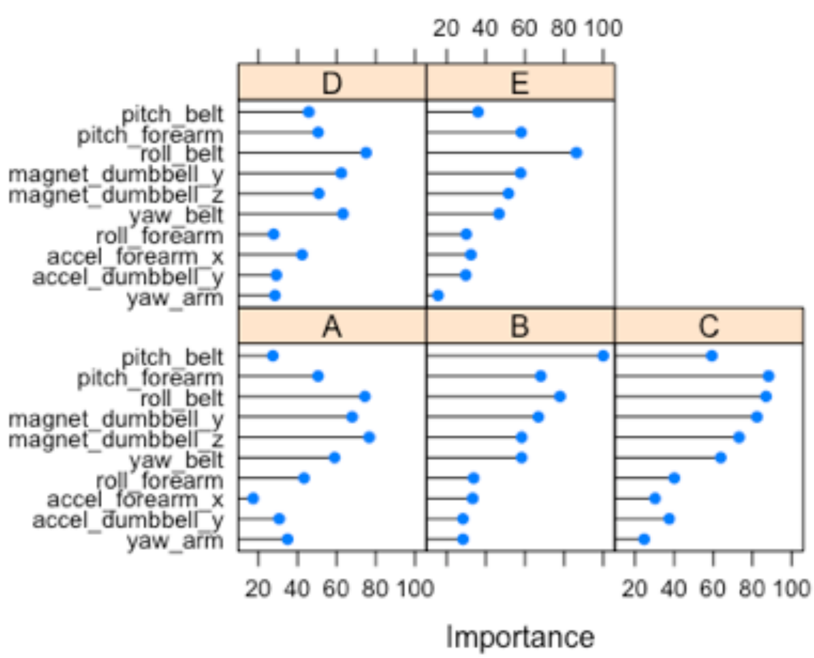
```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
v <- varImp(ModelUno)
plot(v, top = 10) # plot
```

Fig. 2 Importance Vriables for ModelUno



```r
# out of sample error
pd1<- predict(ModelUno, val)
cfm1<- confusionMatrix(pd1, val$classe)

# 3 bagging model
ModelDos <- train(classe ~ .,data=train,method="treebag")
```

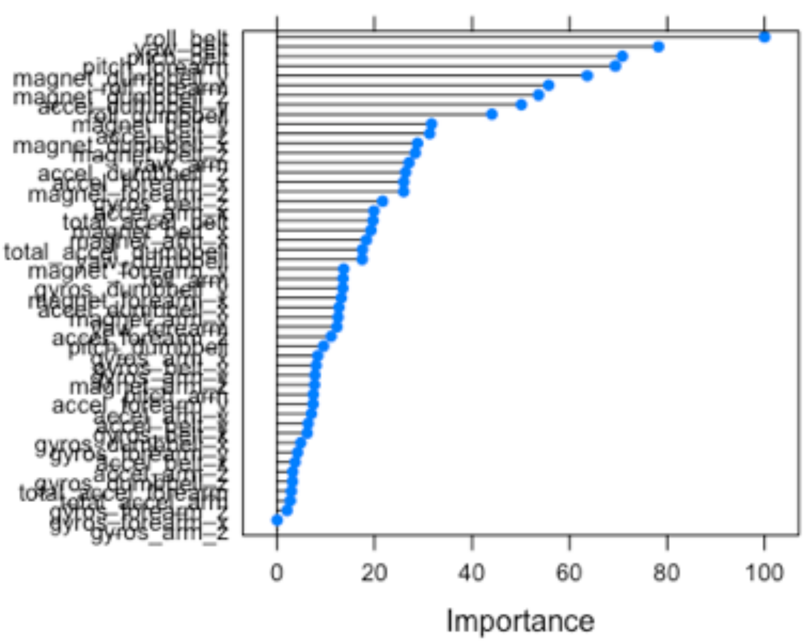
```
## Loading required package: ipred

## Loading required package: plyr

## Loading required package: e1071
```

```r
pd2 <- predict(ModelDos, val)
cfm2 <- confusionMatrix(pd2, val$classe)
cfm2$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##      0.9891249      0.9862446      0.9861337      0.9916151      0.2844520
## AccuracyPValue  McnemarPValue
##      0.0000000      0.3834745
```

```r
plot(varImp(ModelDos)) #plot 3
```

Fig. 3 Importance Variables for ModelDos



```r
## 4 boosting model
ModelTres <- train(classe ~ ., method = "gbm",  data = train, verbose = F, trControl = trainControl(method = "cv",
number = 3))
```

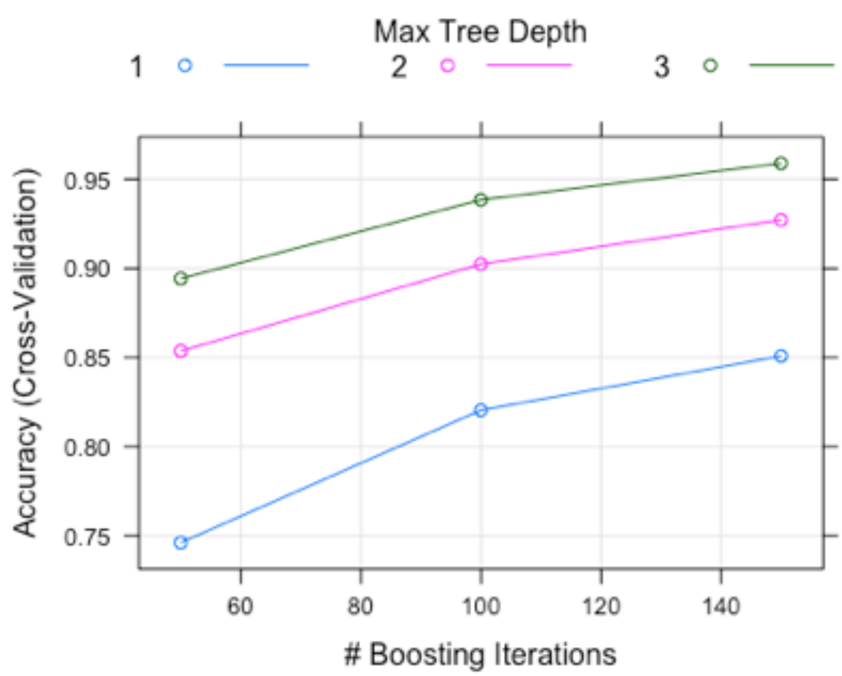
```
## Loading required package: gbm
```

```
## Loading required package: survival

##
## Attaching package: 'survival'

## The following object is masked from 'package:caret':
##
##     cluster

## Loading required package: splines

## Loading required package: parallel

## Loaded gbm 2.1.3
```

```
pd3 <- predict(ModelTres, val)
cfm3 <- confusionMatrix(pd3, val$classe)
cfm3$overall
```

```
##       Accuracy         Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##   9.639762e-01   9.544323e-01   9.588945e-01   9.685910e-01   2.844520e-01
## AccuracyPValue  McnemarPValue
##   0.000000e+00   8.063757e-11
```

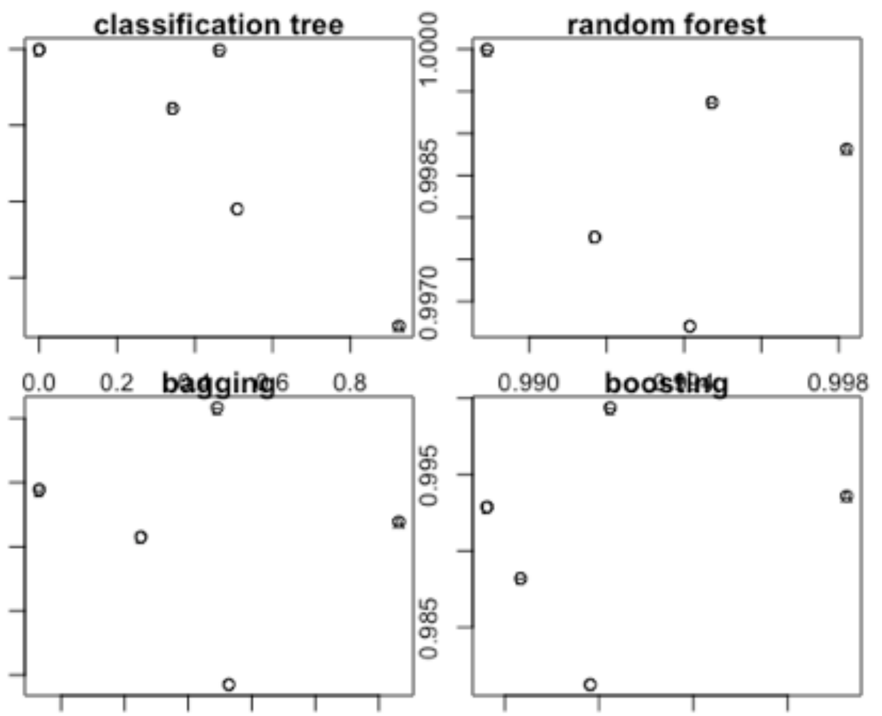
```
plot(ModelTres) # plot 4
```



Fir. 4 Boosting Itterations for ModelTres

```
# select  prediction  model

selectit <- data.frame(tree=cfm$overall[1],   rf=cfm1$overall[1], bagging=cfm2$overall[1],  boosting=cfm3$overall
[1])
```

```
#plot model comparison
par(mfrow=c(2,2))
lables=LETTERS[1:5]
cex= 0.7

plotit <- function(x,y){
  par(mar=c(1,1,1,1))
  plot(x$byClass, main=y)
  text(x$byClass[,1], x$byClass[,2], labels=lables, cex=cex )

}

plotit(cfm,"classification tree")
plotit(cfm1, "random forest")
plotit(cfm2, "bagging")
plotit(cfm3,"boosting")
```



Firg. 5. Modle Comparision

selectit

```
##              tree        rf   bagging  boosting
## Accuracy 0.5033135 0.9940527 0.9891249 0.9639762
```

```
#prediction
results <- as.character(predict(ModelUno, test))
results2 <- as.character(predict(ModelDos, test))
results3 <- as.character(predict(ModelTres, test))

list<- c(results,results2, results3)

equalR <- identical(results, results2)
```

```
equalR <- c(equalR, identical(results2, results3))
equalR <- c(equalR, identical(results, results3))
equalR
```

```
## [1] TRUE TRUE TRUE
```

Conclusion

This project demonstrates that machine learning can be used to build classifcaiton models for complex data. A simple classification tree, failed to preform very well, with only about 50% accuraction. Application of other models, random forest, boosting, bagging lead to better results do to averaging of trees methods. The main question of the project asks, can machine learning develop models to predict accurately the quality of activity as opposed to quantity. The results suggest this is likely.

Reference

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.

Read more: http://groupware.les.inf.puc-rio.br/har#ixzz4eKrNXrNM