Human Activity Recognition with Random Forest

S Smolenski

Introduction

In this project, I consider the Weight Lifting Exercises dataset from [this] (http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har#literature) website. The dataset was collected by using Microsoft Kinect to develop a way to recognize incorrect from correctly performed exercises. According to the website "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)."

Model Building

To classify a given exercise as belonging to one of class A - E, I train a Random Forest model and perform a brief scan over parameter space to determine the values of ntree and mtry that give the highest accuracy.

For reproducibility, I set the rng seed:

```
set.seed(1234)
```

Preprocessing

The raw data set contains several columns which are primarily empty, primarily NA, or contain many mathematical errors (ie "kurtosis_picth_belt" contains many "#DIV/0!" errors). I omit these columns from the dataset and focus on training the model entirely on the more informative parameters.

```
data <- read.csv("pml-training.csv")[,c(8:11,37:49,60:68,84:86,102,113:124,140,151:160)]
test <- read.csv("pml-testing.csv")[,c(8:11,37:49,60:68,84:86,102,113:124,140,151:160)]
```

To perform the scan over parameter space, I create one cross validation set on which to test models for highest accuracy.

```
inTrain <- createDataPartition(y=data$classe, p=.70, list=FALSE)
train <- data[inTrain,]
crossVal <- data[-inTrain,]
```

Furthermore, I eliminate any predictors which have near zero variance, as they will not have strong predictive power and including them in the model will only waste time and processing power. I also drop the first four columns, which contain information about the time at which the exercise was performed and the user who performed the exercise, none of which are of particular interest.

```
zeroVar <- nearZeroVar(train)
train <- train[-c(zeroVar,1:7)]
crossVal <- crossVal[-c(zeroVar,1:7)]
test <- test[-c(zeroVar,1:7)]
```

Model training

Given the limited processing power and time available for this analysis project, I chose to scan over a very limited region of parameter space for the best parameters:

```
N=c(300, 500, 700)
M=c(10,20)
A=0
Nbest=0
Mbest=0

for(n in N){
    for(m in M){
        ffit <- randomForest(classe~., data=train, ntree=n, mtry=m)
        pred <- predict(ffit, crossVal)
        acc<-confusionMatrix(pred, crossVal$classe)$overall['Accuracy']

        if(acc > A) {
            A=acc
            Nbest=n
            Mbest=m
            bestfit=ffit
        }
    }
}
```

Following the scan, we find that the best parameters are:

```
## n =  500
##  m =  10
```

And the model is parameterized by:

```
##
## Call:
##  randomForest(formula = classe ~ ., data = train, ntree = n, mtry = m)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 10
##
##         OOB estimate of  error rate: 1.06%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 3899    6    0    0    1 0.001792115
## B   30 2619    8    1    0 0.014672686
## C    0   20 2373    3    0 0.009599332
## D    1    1   55 2190    5 0.027531083
## E    0    5    2    8 2510 0.005940594
```

Summary and Conclusions

We have trained a random forest model on the Weight Lifting Exercise dataset and achieved an accuracy on the cross validation set of

```
##  Accuracy
## 0.9925234
```