Motion Prediction

Lester Burgwardt

2/13/2021

Selecting Variables

This is a classification problem with five outcomes: A,B,C,D,E. It also has 152 variables not counting the outcome and record identifiers. We first investgated the option of reducing the number of variables to those most influential on the model. There was missing data though out the training set which was dealt with by using the na.action = na.pass option in the train function. We used a prediction with trees approach as it offered the most convenient means of separating the outcomes. Variables selection as made using the Recursive Partitioning And Regression Trees (rpart) as it was fast and although not as accurate as random forest it provided a quick indication as to which variables provided the most information for the classification problem.

Random subsampling from the training set was used to acheive cross validation for building of the rpart models and prediction. A subsample size of 2000 was used representing about 10% of the total dataset size.

A leave-one-out approach was used to select variables. Each variable was used individually in the construction of the rpart model. After the model was built using train the accuracy and kappa values that are of the out were checked for extermly low values indicating a failure in the creation of the model. A threshold of of 0.01 for kappa was used. If kappa for a variable fell below the threshold the model was not used and the selection process moved on to the next variable.

The next step in variable selection was to use the variable's rpart model in a prediction. For that the subsamples held back from the model creation were used. After prediction completed a confusion matrix was constructed using the R function confusionMatrix. This function reports accuracy and kappa. These values were recorded in a csv file.

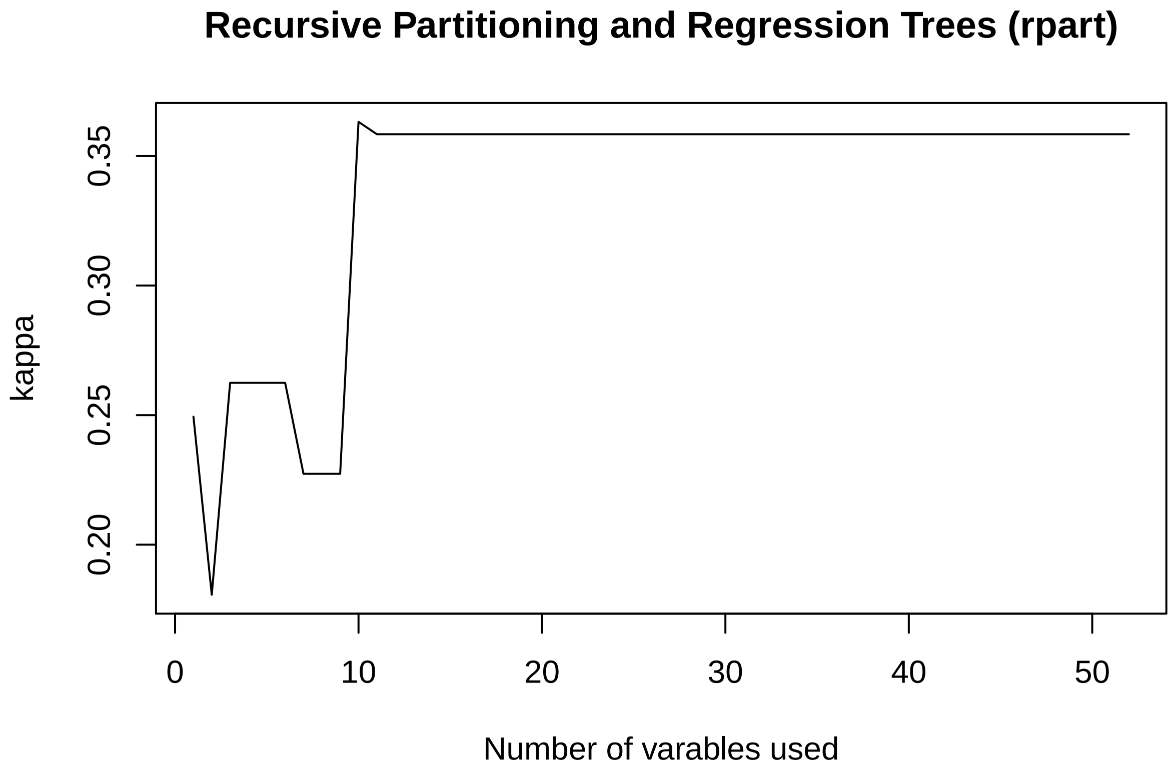
```
training_df <- read.csv(’pml-training.csv’, header = TRUE)
accFile = "D:/Coursera/DataScience/PracticalMachineLearning/CourseProject/acc5.csv"
#acc2.csv was done with 100 samples using train accuracy
#acc3.csv was done with 2000 samples using train accuracy, about 10% of the training set
#acc4.csv was done with 2000 samples using confusionMatrix accuracy - correct way to do this
#Write header
cat("varnumber,varname,accuracy,kappa\n",file=accFile, append=TRUE)

library(caret)

training_df <- subset(training_df, select=-c(X,user_name,raw_timestamp_part_1,raw_timestamp_part_2,
                                             cvtd_timestamp,new_window,num_window))

name_var = names(training_df)
Y = training_df$classe

for(k in 1:152) {
  i = sample(1:length(Y),2000)

  # random forest takes too long, Use rpart: Recursive Partitioning and Regression Trees
  fml = paste(name_var[153],"~", name_var[k], sep=" ")
  mod = train(as.formula(fml), method="rpart",
              data=training_df[-i,],
              na.action = na.pass)
  #print(mod$results[1,3])
  if (mod$results[1,3] < 0.01) {
    cat("Variable ",k,"(",name_var[k],") :",mod$results[1,2],mod$results[1,3],"\n")
    cat(k,",",name_var[k],",",mod$results[1,2],",",mod$results[1,3],"\n",
        sep="",file=accFile, append=TRUE)
    next}

  pred = predict(mod, training_df[i,])
  rpartAcc = confusionMatrix(as.factor(pred), as.factor(training_df[i,’classe’]))

  cat("Variable ",k,"(",name_var[k],") :",rpartAcc$overall[1],rpartAcc$overall[2],"\n")
  cat(k,",",name_var[k],",",rpartAcc$overall[1],",",rpartAcc$overall[2],"\n",
      sep="",file=accFile, append=TRUE)
}

varAcc_df = read.csv(accFile)
```

Ranking Variables

The csv file containing prediction accuracy and kappa was read back in to a data frame which was sorted based on the kappa value. A similar process as before of rpart model building and prediction was performed using a random subsample size of 2000. Except this time everytime a model was built and a prediction made using the variable with the highest kappa the next cycle added the next highest kappa ranked variable. As more variables were added prediction kappa would improve. Figure 1 shows this improvement. Prediction accuracy increased until the eleventh variable was added. At that point prediction kappa stopped increasing and remained at about 0.35. We allowed the process to continue until it failed with 52 variables in the model.

```
training_df <- read.csv(’pml-training.csv’, header = TRUE)
training_df <- subset(training_df, select=-c(X,user_name,raw_timestamp_part_1,raw_timestamp_part_2,
                                             cvtd_timestamp,new_window,num_window))
name_var = names(training_df)

accFileIn = "D:/Coursera/DataScience/PracticalMachineLearning/CourseProject/acc4.csv"
varAcc_df = read.csv(accFileIn)
varRanked_df = varAcc_df[order(varAcc_df$kappa,decreasing=TRUE),]

accFileOut = "D:/Coursera/DataScience/PracticalMachineLearning/CourseProject/rankedVarAcc2.csv"
cat("varnumber,accuracy,kappa\n",file=accFileOut, append=TRUE)

library(caret)

for(k in 1:152) {

  #Build up formula to use in train
  #Add variables by decreasing rank
  fml1 = paste(varRanked_df$varname[1:k],collapse="+")
  fml = paste(name_var[153],"~", fml1, sep=" ")
  #print(fml)

  mod = train(as.formula(fml), method="rpart",
              data=training_df[-i,],
              na.action = na.pass)
  cat("Ranked var set ",k,mod$results[1,2],mod$results[1,3],"\n")
  if (mod$results[1,3] < 0.01) {break}

  pred = predict(mod, training_df[i,])
  rpartAcc = confusionMatrix(as.factor(pred), as.factor(training_df[i,’classe’]))

  cat("Ranked var set ",k,rpartAcc$overall[1],rpartAcc$overall[2],"\n\n")
  cat(k,",",rpartAcc$overall[1],",",rpartAcc$overall[2],"\n",
      sep="",file=accFileOut, append=TRUE)

}
```

```
plotFileIn = "D:/Coursera/DataScience/PracticalMachineLearning/CourseProject/rankedVarAcc1.csv"
rpartResults_df = read.csv(plotFileIn)

plot(rpartResults_df$varnumber,rpartResults_df$kappa,
     type="l",
     main = ’Recursive Partitioning and Regression Trees (rpart)’,
     xlab = "Number of varables used",
     ylab = "kappa")
```



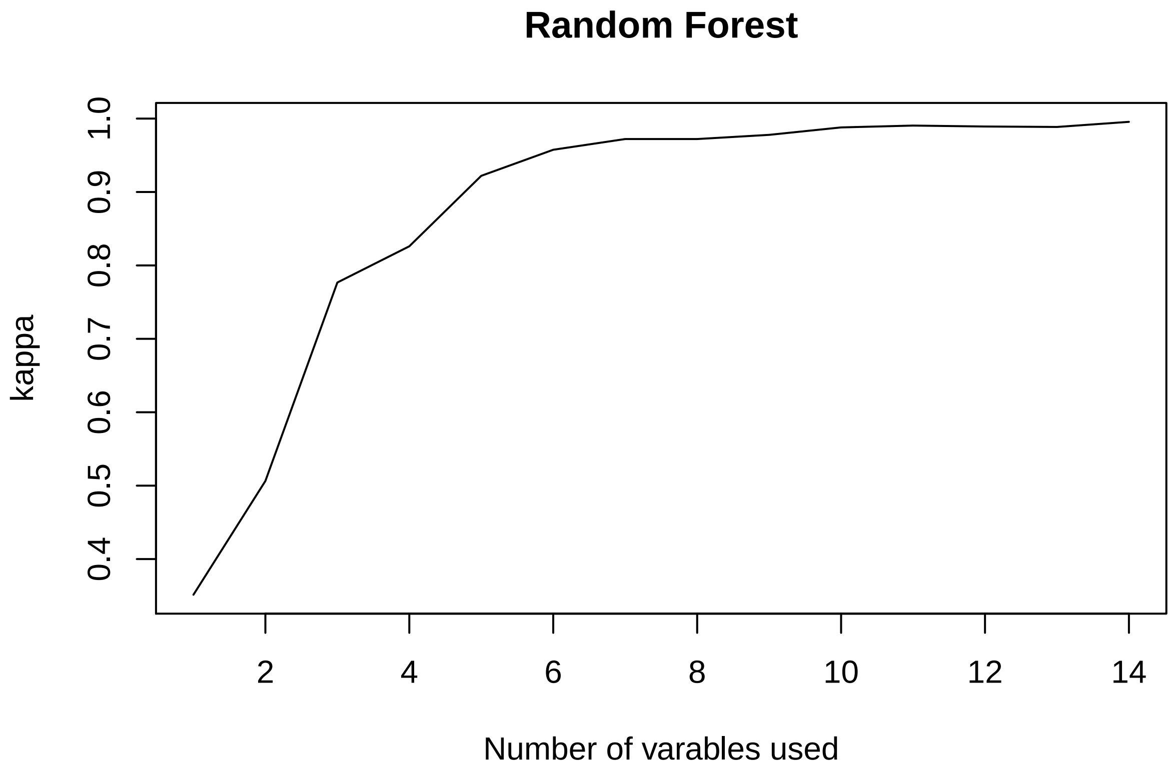
Test ranked variables on random forest

The final step in variable selection was to repeat the additive process using random forest prediction. We immediately saw an improvement in prediction kappa of 0.77 with the first variable over the 0.25 kappa observed using rpart. As variables were added to the model the prediction kappa became asymptotic after eight variables were added. Figure 2 shows the improvement with kappa as variables were added.

```
training_df <- read.csv(’pml-training.csv’, header = TRUE)
training_df <- subset(training_df, select=-c(X,user_name,raw_timestamp_part_1,raw_timestamp_part_2,
                                              cvtd_timestamp,new_window,num_window))
name_var = names(training_df)

accFileIn = "D:/Coursera/DataScience/PracticalMachineLearning/CourseProject/acc4.csv"
varAcc_df = read.csv(accFileIn)
varRanked_df = varAcc_df[order(varAcc_df$kappa,decreasing=TRUE),]

#accFileOut = "D:/Coursera/DataScience/PracticalMachineLearning/CourseProject/randomForestRankedAcc2.cs
#cat("varnumber,accuracy,kappa\n",file=accFileOut, append=TRUE)

library(caret)

for(k in 1:152) {

  #Build up formula to use in train
  #Add variables by decreasing rank
  fml1 = paste(varRanked_df$varname[1:k],collapse="+")
  fml = paste(name_var[153],"~", fml1, sep=" ")
  #print(fml)

  mod = train(as.formula(fml), method="rf",
              data=training_df[-i,],
              na.action = na.pass)
  cat("Ranked var set ",k,mod$results[1,2],mod$results[1,3],"\n")
  if (mod$results[1,3] < 0.01) {break}

  pred = predict(mod, training_df[i,])
  rpartAcc = confusionMatrix(as.factor(pred), as.factor(training_df[i,'classe']))



}
```

```
accFileIn = "D:/Coursera/DataScience/PracticalMachineLearning/CourseProject/randomForestRankedAcc1.csv"
rfResults_df = read.csv(accFileIn)

plot(rfResults_df$varnumber,rfResults_df$kappa,
     type="l",
     main = "Random Forest",
     xlab = "Number of varables used",
     ylab = "kappa")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): NAs introduced by coercion

## Warning in xy.coords(x, y, xlabel, ylabel, log): NAs introduced by coercion
```



Running Random Forest on the Test Set

Running predict on the test set using the 14 variable random forest model produced 100% accuracy.

```
Random Forest

17622 samples
14 predictor
5 classes: ‘A’, ‘B’, ‘C’, ‘D’, ‘E’

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 17622, 17622, 17622, 17622, 17622, 17622, ...
Resampling results across tuning parameters:

mtry Accuracy Kappa
2 0.9887797 0.9858111
8 0.9882297 0.9851170
14 0.9811163 0.9761248

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.

source(‘D:/Coursera/DataScience/PracticalMachineLearning/CourseProject/test.R’)
> pred
[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
>
```

```r
test_df <- read.csv('pml-testing.csv', header = TRUE)
test_df <- subset(test_df, select=-c(X,user_name,raw_timestamp_part_1,raw_timestamp_part_2,
                                             cvtd_timestamp,new_window,num_window))

testFileOut = "D:/Coursera/DataScience/PracticalMachineLearning/CourseProject/testResults1.csv"

library(caret)

pred = predict(mod, test_df)

pred_df <- data.frame(pred)
```