Lab2 block2 Ensemble methods 732A95 ML

Anton Persson antpe404

9 december 2016

Lab2a block 2

I started off by looking at the data to get a feel for it, and how a tree would possibly look.

Assignement 2.1

I used 2/3 of the data as training set and 1/3 as testing set, as instructed. Since the bagging method guarantees that the bagged error is at most the same error as the average individual errors, I did compute the average errors of the bagged models. The code for the computations is shown below.

```
#Assignment 2.
bodyfat<-read.csv2("data/bodyfatregression.csv", sep=";", header=T)
library(tree)

#2.1
set.seed(1234567890)
#sampling<-sample(1:nrow(bodyfat))
bodyfat_samplad<-bodyfat[sample(1:nrow(bodyfat)),]
bodyfat_tr<-bodyfat_samplad[1:73,] #Traningsset
bodyfat_te<-bodyfat_samplad[74:110,] #testset

#The upper bound
felen_upper<-integer(0)
set.seed(1234567890)
for ( i in 1:100){
  saf<-sample(1:nrow(bodyfat_tr), replace=T)
  bodyfat_bag<-bodyfat_tr[saf,]
  fat_tree<-tree(formula=Bodyfat_percent~., data=bodyfat_bag, split="deviance")
  fitsen<-predict(fat_tree, newdata=bodyfat_te)
  felen_upper[i]<-mean((fitsen-bodyfat_te$Bodyfat_percent)**2)
}

upperbound<-mean(felen_upper)
```

My result is that the upper bound of the squared error of the bagging regression tree is 37.103.

Assignment 2.2

The code to repeat the same task but with cross validation (3 folds) instead of a hold out test data set is shown below.

```r
folds<-3
baggingar<-100
set.seed(1234567890)
folds_data<-suppressWarnings(split(bodyfat, 1:folds))

alla_fel<-matrix(0, nrow=folds, ncol=baggingar)

for (i in 1:folds){
  training<-folds_data[-i]
  del1_train<-data.frame(training[1])
  colnames(del1_train)<-colnames(bodyfat)
  del2_train<-data.frame(training[2])
  colnames(del2_train)<-colnames(bodyfat)
  training<-rbind(del1_train, del2_train)
  testing<-data.frame(folds_data[i])
  colnames(testing)<-colnames(bodyfat)

  for (j in 1:baggingar){

  urval<-sample(1:nrow(training), replace=T)
  bodyfat_bag<-training[urval,]
  fat_tree<-tree(formula=Bodyfat_percent~., data=bodyfat_bag, split="deviance")
  fitsen<-predict(fat_tree, newdata=testing)
  alla_fel[i, j]<-mean((fitsen-testing$Bodyfat_percent)**2)
}
}

upperbound_2<-mean(alla_fel)
```

The results I receive from the code above says that the upper bound when using three folds CV is 40.53.

Assignment 2.3

I assume that it's supposed to be *trees* instead of *tree* in the instructions, i.e. plural. However, I would return a list of all trees created by the bagging regression tree, but with all data used as training data. The code for that is presented below.

```
trees_fulldataset<-list() #empty list to place the trees in.
set.seed(1234567890)

for ( i in 1:100){
  saf<-sample(1:nrow(bodyfat), replace=T)
  bodyfat_bag<-bodyfat[saf,]
  fat_tree<-tree(formula=Bodyfat_percent~., data=bodyfat_bag, split="deviance")
  trees_fulldataset[[i]]<-fat_tree
}
```

As seen in the code above, I put every single tree in different elements of a list. The list trees_fulldataset thus contains 100 trees. I present one of them, the third tree, just to show how what it looks like.

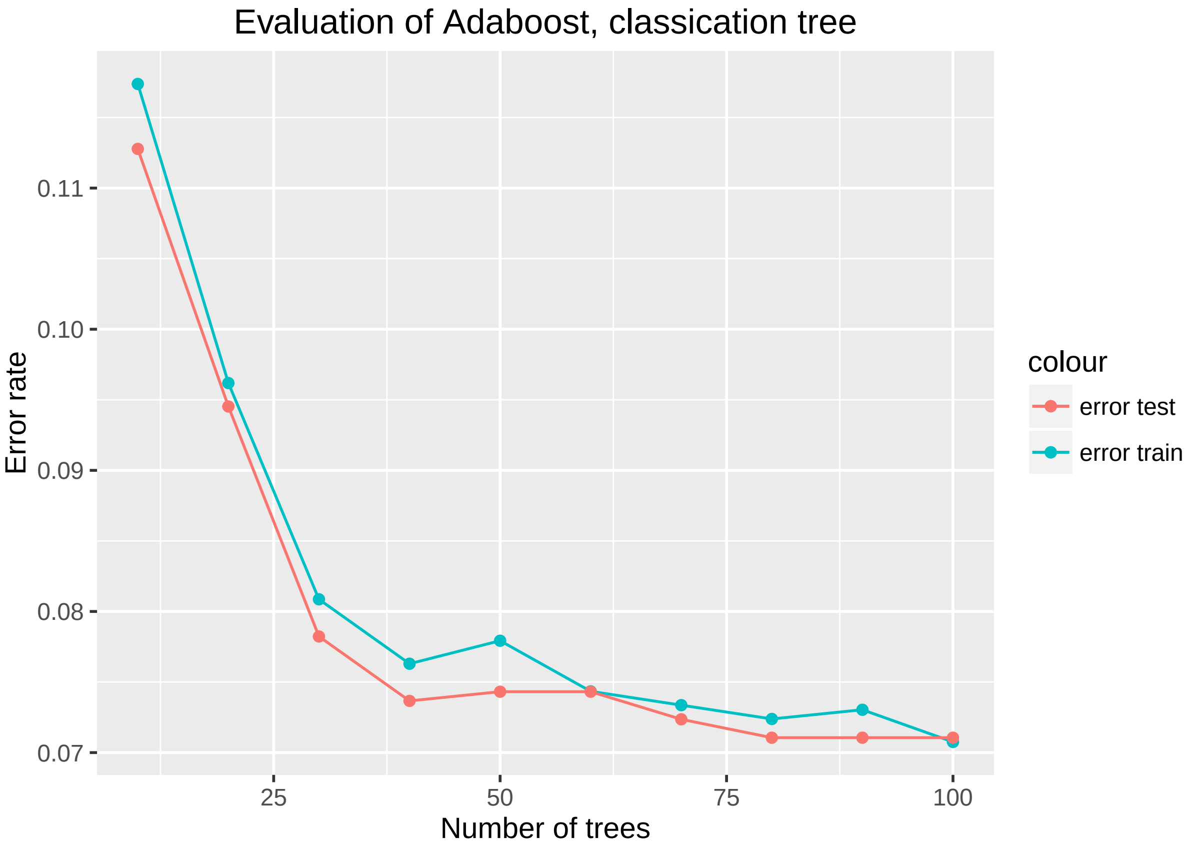
```
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 110 10880.00 22.90
##    2) Waist_cm < 95.45 63  2378.00 16.21
##      4) Waist_cm < 84.05 23   465.70 11.57 *
##      5) Waist_cm > 84.05 40  1132.00 18.88
##       10) Weight_kg < 86.65 26   425.50 20.69 *
##       11) Weight_kg > 86.65 14   461.50 15.50
##         22) Waist_cm < 90.5 6    58.83 10.83 *
##         23) Waist_cm > 90.5 8   174.00 19.00 *
##    3) Waist_cm > 95.45 47  1893.00 31.87
##      6) Waist_cm < 104.8 25   627.80 28.08
##       12) Weight_kg < 83.5 5   107.20 32.60 *
##       13) Weight_kg > 83.5 20   393.00 26.95 *
##      7) Waist_cm > 104.8 22   497.30 36.18
##       14) Waist_cm < 109.2 17   323.10 34.76 *
##       15) Waist_cm > 109.2 5    24.00 41.00 *
```

Assignment 4

The first task is the evualute the Adaboost algorithm and it's performance in classification trees. As in assignment 2, I use 2/3 as training data and 1/3 as test data. The required plot is shown below the code that produces it.

```
library(mboost)
library(randomForest)
library(ggplot2)

spam<-read.csv2("data/spambaselab2b2.csv", sep=";", header=T)
spam$Spam<-as.factor(spam$Spam)
set.seed(1234567890)
spam_samplad<-spam[sample(1:nrow(spam)), ]
```

```
spam_tr<-spam_samplad[1:round((2/3)*nrow(spam)), ]
spam_te<-spam_samplad[-(1:round((2/3)*nrow(spam))), ]

sekvens<-seq(10,100, 10)
training_errors<-integer()
test_errors<-integer()
index<-1
for (i in sekvens){
  modellen_ct<-blackboost(Spam~., data=spam_tr, family=AdaExp(),  control=boost_control(mstop=i))

  tejbell_train<-table(pred=predict(modellen_ct, newdata= spam_tr, type="class"), truth=spam_tr$Spam)
  training_errors[index]<-1-sum(diag(tejbell_train))/sum(tejbell_train)

  tejbell_test<-table(pred=predict(modellen_ct, newdata= spam_te, type="class"), truth=spam_te$Spam)
  test_errors[index]<-1-sum(diag(tejbell_test))/sum(tejbell_test)
  index<-index+1
}

plotredo_ct<-data.frame(cbind(sekvens,training_errors, test_errors))

number_of_trees_plot<-ggplot(data=plotredo_ct)+geom_point(aes(x=sekvens, y=training_errors, col="error t
  geom_line(aes(x=sekvens, y=training_errors, col="error train"))+
  geom_point(aes(x=sekvens, y=test_errors, col="error test"))+
  geom_line(aes(x=sekvens, y=test_errors, col="error test"))+xlab("Number of trees")+
  ylab("Error rate")+ggtitle("Evaluation of Adaboost, classication tree")

number_of_trees_plot
```


Evaluation of Adaboost, classication tree
Error rate
0.11
0.10
0.09
0.08
0.07
25
50
75
100
Number of trees
colour
error test
error train