Lab1_Block2

gowtham

12/4/2019

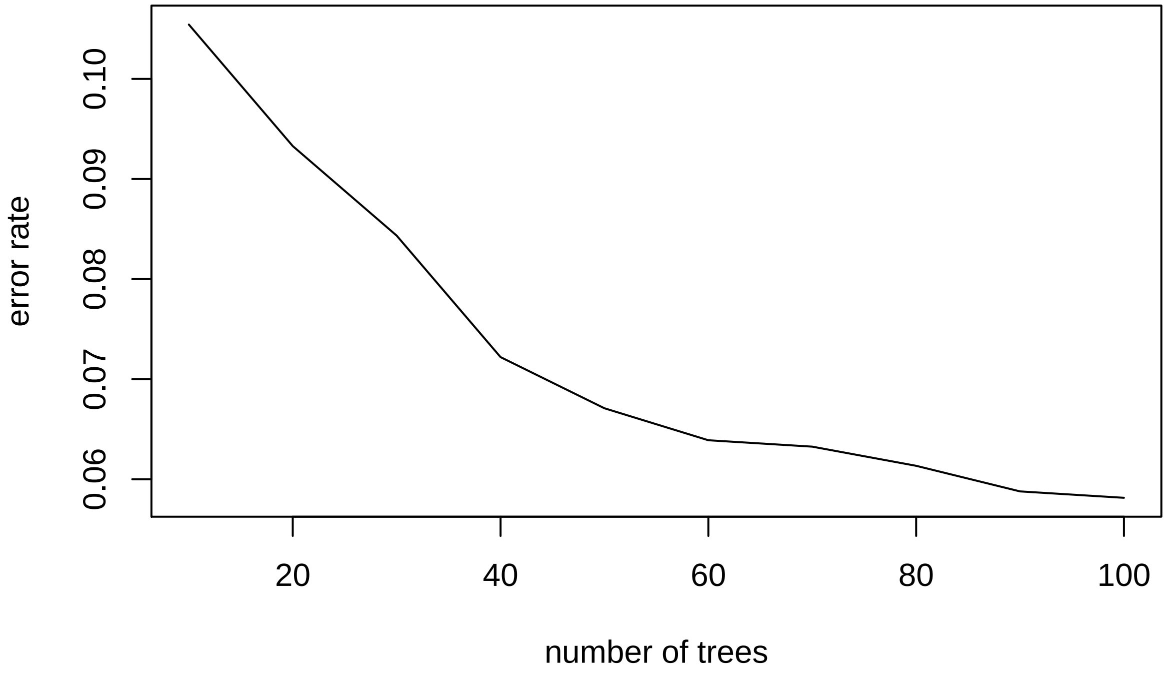
#1. Ensemble Methods

##1. Adaboost classification tree

```
## Loading required package: parallel
```

```
## Loading required package: stabs
```

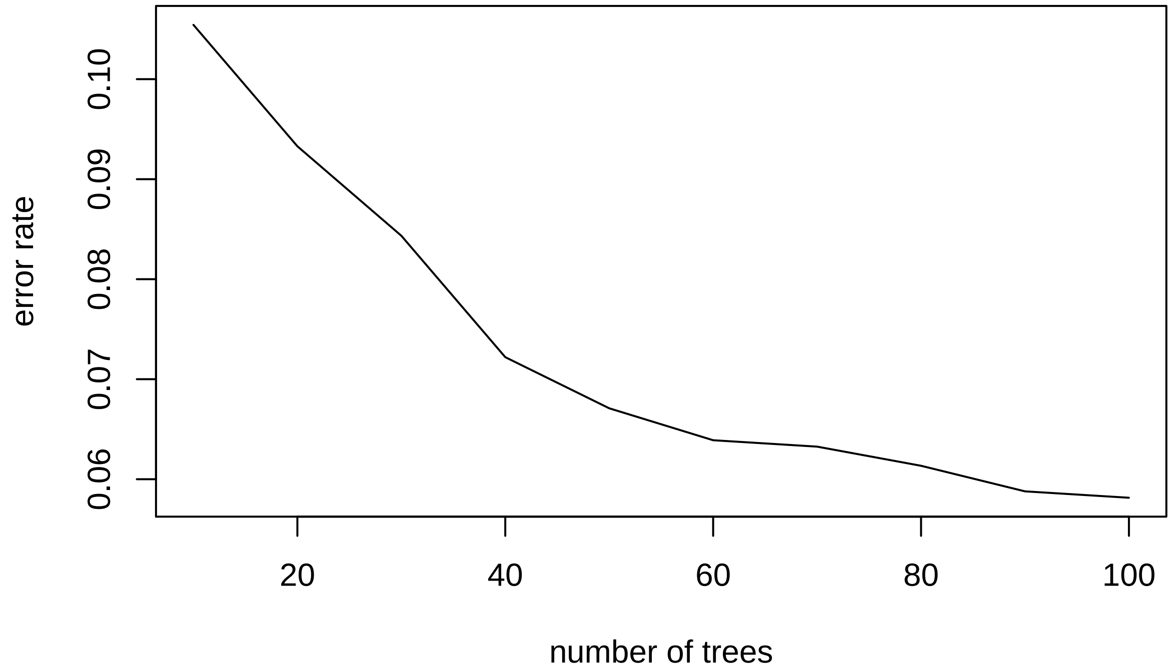
```
## This is mboost 2.9-1. See 'package?mboost' and 'news(package  = "mboost")'
## for a complete list of changes.
```



##2 Random Forest

```
## randomForest 4.6-14
```

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



Note: Both of these algorithms give similar accuracy in prediction over the given data set. # Code Appendix

```
knitr::opts_chunk$set(echo = TRUE)

sp<- read.csv2("spambase.csv") #load data
sp$Spam<-as.factor(sp$Spam)
nrow=dim(sp)[1]
library(mboost)
set.seed(12345)                         # divide as train & test data
train=sample(1:nrow, floor(nrow*0.66))
sp.test = sp[train,]
sp.train = sp[-train,]
mis.cal.vec<-seq(from=1.0, to=10, by=1.0)
error.vec<-c(1,2,3,4,5,6,7,8,9,10)
total.tree<-c(1:100)
number.tree<-total.tree[lapply(total.tree, "%%",10)==0]
i=1
while (i<=length(number.tree)) {
 tree.data=blackboost(formula = Spam~., data = sp.train, family =AdaExp(), control =boost_control(numbe
  tree.pred=predict(tree.data,newdata=sp.train,type="class")
    tab=table(sp.train$Spam,tree.pred)
    t1=tab[2,2]
    t2=tab[1,1]
    t3=tab[1,2]
    t4=tab[2,1]
  mis.cal.vec[i]<-(t4+t3)/sum(tab)
```

```
   i=i+1

}


plot(x=number.tree,y=mis.cal.vec,xlab = "number of trees", ylab = "error rate", type = "l")
sp<- read.csv2("spambase.csv") #load data
sp$Spam<-as.factor(sp$Spam)
nrow=dim(sp)[1]
library(randomForest)
set.seed(12345)                     # divide as train & test data
train=sample(1:nrow, floor(nrow*0.66))
sp.test = sp[train,]
sp.train = sp[-train,]
mis.cal.vec2<-seq(from=1.0, to=10, by=1.0)
error.vec<-c(1,2,3,4,5,6,7,8,9,10)
total.tree<-c(1:100)
number.tree1<-total.tree[lapply(total.tree, "%%",10)==0]
i=1
while (i<=length(number.tree1)) {
 tree.data=randomForest(formula = Spam~., data = sp.train, family =binomial(), ntree=number.tree1[i])
  tree.pred1=predict(tree.data,newdata=sp.train,type="class")
    tab=table(sp.train$Spam,tree.pred1)
    t1=tab[2,2]
    t2=tab[1,1]
    t3=tab[1,2]
    t4=tab[2,1]
  mis.cal.vec2[i]<-(t4+t3)/sum(tab)


   i=i+1

}

plot(x=number.tree,y=mis.cal.vec,xlab = "number of trees", ylab = "error rate", type = "l")
```