Project2

HanGyu Kang

April 20, 2019

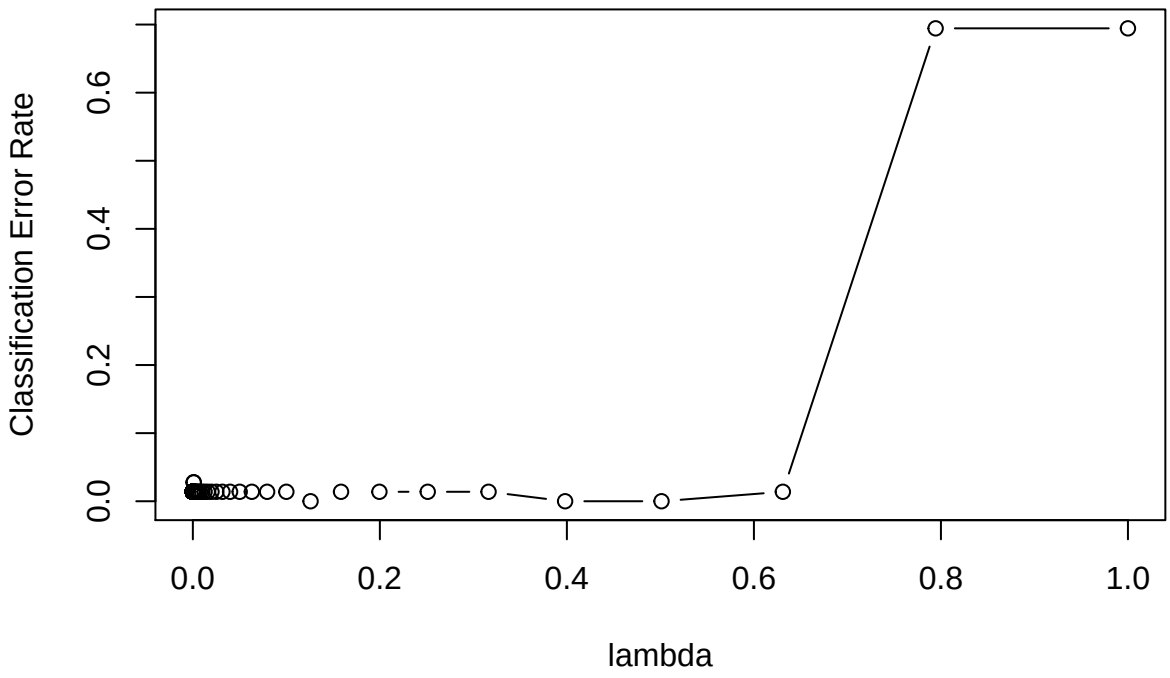
```
dat <- read.delim('wine.data', header=F, sep = ',')
colnames(dat)<- c("Class","Alcohol","Malicacid","Ash","AlcalinityofAsh","Magnesium","Totalphenols","Fla
dat$Class <- as.factor(dat$Class)

set.seed(4052)
train.ind <- sample(1:nrow(dat),0.6*nrow(dat))
train.dat <- dat[train.ind,]
test.dat <- dat[-train.ind,]
```

```
library(gbm)
```

```
## Warning: package 'gbm' was built under R version 3.4.4
```

```
## Loaded gbm 2.1.5
```

```
pow <- seq(-10, 0, by = 0.1)
lambdas <- 10^pow
test.boosting.error <- rep(NA, length(lambdas))
for (i in 1:length(lambdas)) {
  set.seed(4052)
  boost.dat <- gbm(Class~., data = train.dat, n.trees=1000, interaction.depth = 1, distribution = "mult
  test.pred <- apply(predict(boost.dat, test.dat, n.trees=1000), 1, which.max)
  test.boosting.error[i] <- 1-sum(diag(table(test.pred, test.dat$Class)))/sum(table(test.pred, test.dat$
}
plot(lambdas, test.boosting.error, type="b", xlab = "lambda", ylab="Classification Error Rate")
```




```
min.lambda <- lambdas[which.min(test.boosting.error)]
min.lambda
```

```
## [1] 0.1258925
```

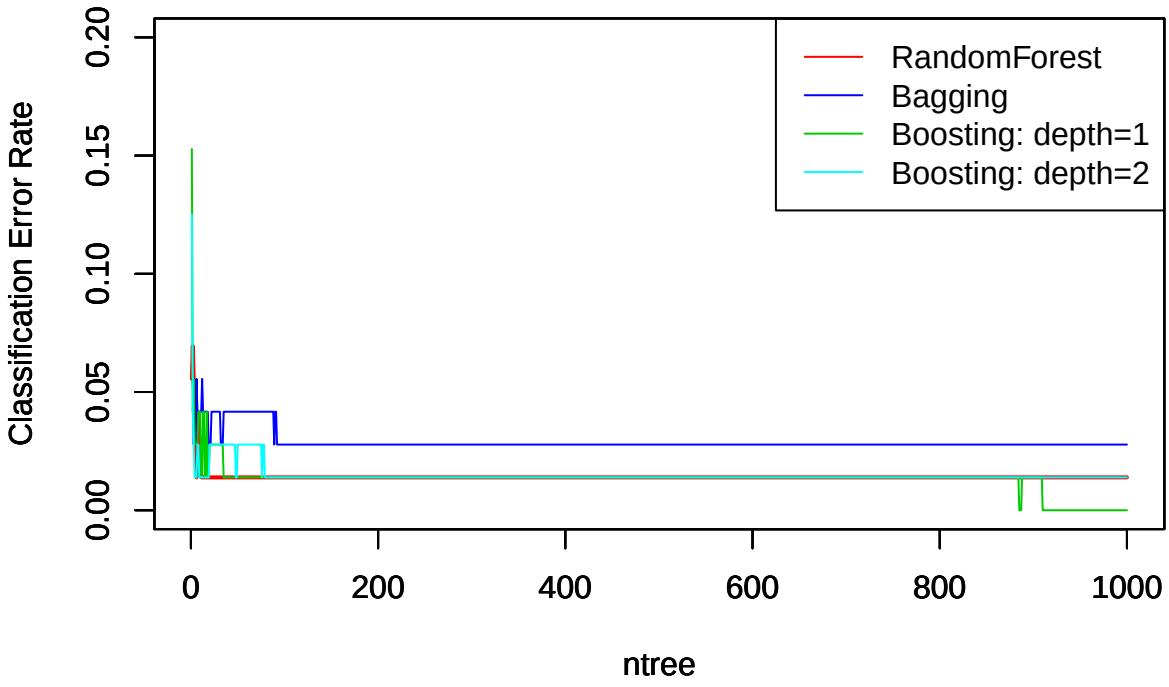
```
test.boosting.error1 <- rep(NA, 1000)
for (i in 1:1000) {
  set.seed(4052)
  boost.dat <- gbm(Class~., data = train.dat, n.trees=i, interaction.depth = 1, distribution = "multinor
  test.pred <- apply(predict(boost.dat, test.dat, n.trees=i), 1, which.max)
  test.boosting.error1[i] <- 1-sum(diag(table(test.pred, test.dat$Class)))/sum(table(test.pred, test.da
}

test.boosting.error2 <- rep(NA, 1000)
for (i in 1:1000) {
  set.seed(4052)
  boost.dat <- gbm(Class~., data = train.dat, n.trees=i, interaction.depth = 2, distribution = "multinor
  test.pred <- apply(predict(boost.dat, test.dat, n.trees=i), 1, which.max)
  test.boosting.error2[i] <- 1-sum(diag(table(test.pred, test.dat$Class)))/sum(table(test.pred, test.da
}

library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.4.4
```

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

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

```r
#randomforest
test.rf.error <- rep(NA, length=1000)
for(i in 1:1000){
  set.seed(4052)
  rf.dat = randomForest(Class~., data=train.dat, mtry = floor(sqrt(13)), ntree=i)
  yhat.rf = predict(rf.dat, newdata = test.dat)
  test.rf.error[i] <- 1-(sum(diag(table(yhat.rf, test.dat$Class))))/sum(table(yhat.rf, test.dat$Class))
  }

#bagging
test.bag.error <- rep(NA, length=1000)
for(i in 1:1000){
  set.seed(4052)
  bag.dat = randomForest(Class~., data=train.dat, mtry = 13, ntree=i)
  yhat.bag = predict(bag.dat, newdata = test.dat)
  test.bag.error[i] <- 1-(sum(diag(table(yhat.bag, test.dat$Class))))/sum(table(yhat.bag, test.dat$Class
  }

ntree <- 1:1000
plot(ntree, test.rf.error, type = 'l', ylab="Classification Error Rate", ylim = c(0, 0.20), col=2, lwd =
par(new=T)
plot(ntree, test.bag.error, type = 'l', ylab="Classification Error Rate", ylim = c(0, 0.20), col=4)
par(new=T)
plot(ntree, test.boosting.error1, type = 'l', ylab="Classification Error Rate", ylim = c(0, 0.20), col=
par(new=T)
plot(ntree, test.boosting.error2, type = 'l', ylab="Classification Error Rate", ylim = c(0, 0.20), col=
legend("topright", c("RandomForest","Bagging","Boosting: depth=1", "Boosting: depth=2"), lwd=c(1,1), co
```




```
#randomforest
set.seed(4052)
rf.dat = randomForest(Class~., data=train.dat, mtry = floor(sqrt(13)), ntree=1000)

yhat.rf = predict(rf.dat, newdata = test.dat)
table(yhat.rf, test.dat$Class)
```

```
##
## yhat.rf  1  2  3
##       1 22  0  0
##       2  0 26  0
##       3  0  1 23
```

```
1-mean(yhat.rf == test.dat$Class)
```

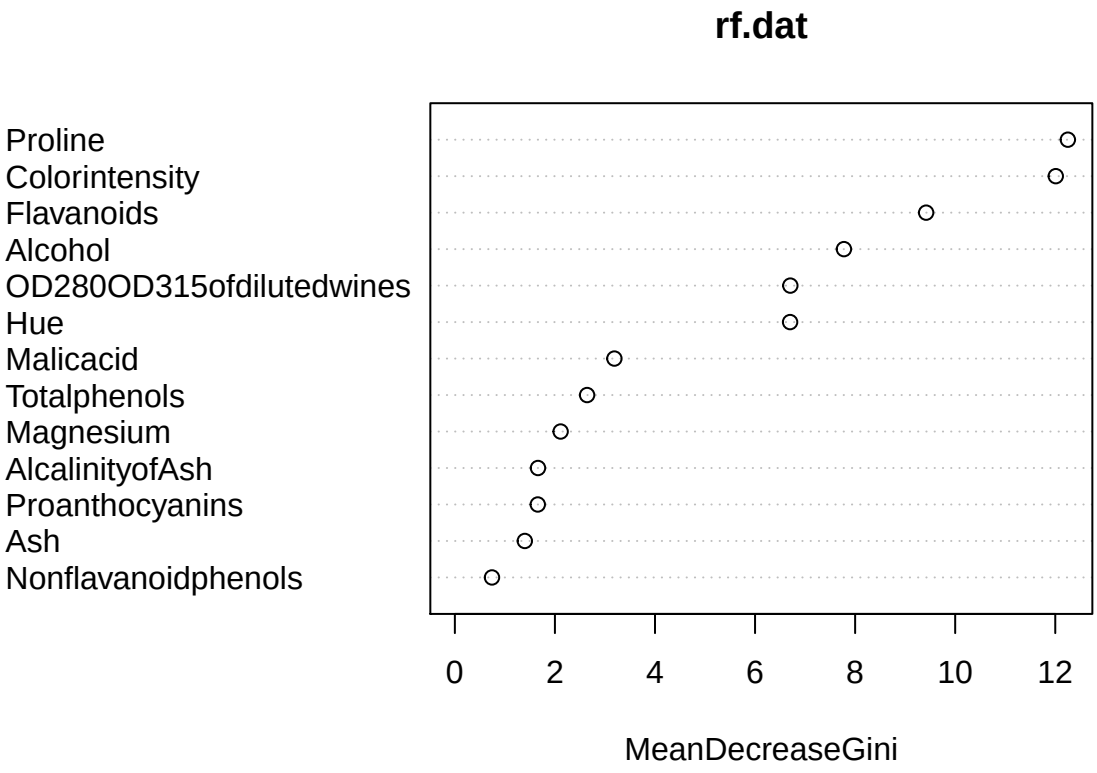
```
## [1] 0.01388889
```

```
importance(rf.dat)
```

```
##                         MeanDecreaseGini
## Alcohol                        7.7764714
## Malicacid                      3.1882303
## Ash                            1.3978761
## AlcalinityofAsh                1.6615111
## Magnesium                      2.1137782
## Totalphenols                   2.6429692
## Flavanoids                     9.4197108
## Nonflavanoidphenols            0.7417862
```

```
## Proanthocyanins                1.6572085
## Colorintensity                12.0104450
## Hue                            6.6999336
## OD280OD315ofdilutedwines       6.7048910
## Proline                       12.2514717
```

```
varImpPlot(rf.dat)
```



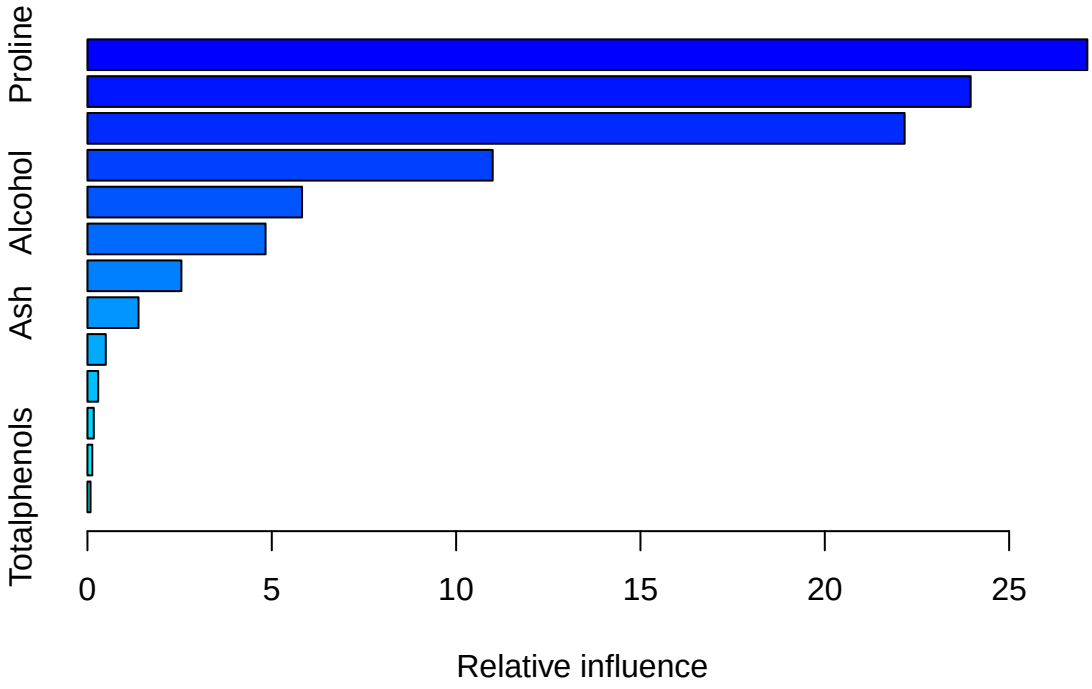

```
#boosting
set.seed(4052)
boost.dat <- gbm(Class ~ ., data = train.dat, n.trees = 1000, interaction.depth = 2, distribution = "mu
yhat.bt <- apply(predict(boost.dat, test.dat, n.trees=1000), 1, which.max)
table(yhat.bt, test.dat$Class)
```

```
##
## yhat.bt  1  2  3
##       1 22  0  0
##       2  0 26  0
##       3  0  1 23
```

```
1-mean(yhat.bt == test.dat$Class)
```

```
## [1] 0.01388889
```

```
summary(boost.dat)
```




```
##                                                 var     rel.inf
## Proline                                     Proline 27.12141299
## Flavanoids                               Flavanoids 23.95634875
## Colorintensity                     Colorintensity 22.16504803
## Hue                                             Hue 10.99221959
## Alcohol                                     Alcohol  5.82096275
## OD280OD315ofdilutedwines OD280OD315ofdilutedwines  4.83193584
## Malicacid                                 Malicacid  2.54861432
## Ash                                             Ash  1.38591614
## Magnesium                                 Magnesium  0.49754985
## AlcalinityofAsh                   AlcalinityofAsh  0.29197295
## Nonflavanoidphenols           Nonflavanoidphenols  0.17225799
## Proanthocyanins                   Proanthocyanins  0.13195814
## Totalphenols                         Totalphenols  0.08380265
```