BLOCK2_PART1

Jin Yan; Donwei Ni;

2022-12-09

EMSEMBLE METHOD

TASK ONE

The relevant code is added in Appendix

mean_mis_rate1 0.205101

mean_mis_rate10 0.122574

mean_mis_rate100 0.077882

var_mis_rate1 0.002805146

var_mis_rate10 0.0008313639

var_mis_rate100 0.0004518479

TASK TWO

The relevant code is added in Appendix

mean_mis_rate1 0.093459

mean_mis_rate10 0.010777

mean_mis_rate100 0.006265

var_mis_rate1 0.01690613

var_mis_rate10 0.0001199092

var_mis_rate100 2.553328e-06

TASK THREE

The relevant code is added in Appendix

mean_mis_rate1 0.213347

mean_mis_rate10 0.077935

mean_mis_rate100 0.034172

var_mis_rate1 0.01461418

var_mis_rate10 0.001286011

var_mis_rate100 1.389431e-05

TASK FOUR **1.What happens with the mean error rate when the number of trees in the random forest grows? Why?** From the above three situations, we can see that as the number of basic models increase, the mean error decreases. This is because when B(the number of of basic models) increases, the variance decreases according to the formula (7.2b) in MLFC.

2.The third dataset represents a slightly more complicated classification problem than the first one. Still, you should get better performance for it when using suffi-cient trees in the random forest. Explain why you get better performance

The smaller nodesize means that the decision tree tree is larger, in other words , the average of the predicted values from the decision tree in the random forest has smaller bias compared to the true values, and this will increase the variance of the distribution of the predicted values from the decision tree in the random forest. However, according to the relationship between the predicted values of the decision tree and the predicted values of the random forest, the effect of bagging can make the variance of the distribution of the predicted values of the random forest smaller, especially when B increases gradually. This is why, for the third data set, when B is 1, the prediction effect is not as good as the first set of predictions. This is because when B is 1 and the decision tree becomes complex, compared to the true value ,the bias of the mean of the distribution of the predicted values of the decision tree inside the random forest decreases and the variance increases, but the bagging effect of the random forest does not work well, so the prediction effect of the random forest is not very good. However, we notice that the effect of bagging is also improved when B increases. Accordingly, compared to the true value the bias of the mean value of the entire random forest prediction distribution is reduced. At the same time, the variance also decreases. In other words, the prediction accuracy of the individual random forest has been significantly improved. This also corresponds to the data.

```
set.seed(1234)
x1<-runif(1000)
x2<-runif(1000)
tedata<-cbind(x1,x2)
y<-as.numeric(x1<x2)
telabels<-as.factor(y)
test_data <- as.data.frame(tedata)
test_data <- dplyr::mutate(test_data,telabels)



x1<-runif(100)
x2<-runif(100)
trdata<-cbind(x1,x2)
y<-as.numeric(x1<x2)
trlabels<-as.factor(y)
train_data <- as.data.frame(trdata)
train_data <- dplyr::mutate(train_data,trlabels)

# QUESTION ONE PART ONE
mis_rate1 <- c()
for(i in 1:1000){

  Forest_1 <- randomForest::randomForest(trlabels~.,data = train_data, ntree = 1
                                         ,nodesize = 25, keep.forest = TRUE)
  pre <- predict(Forest_1,newdata = test_data)
  misclassification_rate <- mean(pre!=telabels)
  mis_rate1 <- c(mis_rate1,misclassification_rate)
}

mean_mis_rate1 <- mean(mis_rate1)
var_mis_rate1 <- var(mis_rate1)

# the mean and variable for misclassification rate when B = 1
mean_mis_rate1
```

```
var_mis_rate1


# QUESTION ONE PART TWO
mis_rate10 <- c()
for(i in 1:1000){

  Forest_10 <- randomForest::randomForest(trlabels~.,data = train_data, ntree =
                                        10, nodesize = 25, keep.forest = TRUE)
  pre <- predict(Forest_10,newdata = test_data)
  misclassification_rate <- mean(pre!=telabels)
  mis_rate10 <- c(mis_rate10,misclassification_rate)
}

mean_mis_rate10 <- mean(mis_rate10)
var_mis_rate10 <- var(mis_rate10)

# the mean and variable for misclassification rate when B = 10
mean_mis_rate10
var_mis_rate10




# QUESTION ONE PART THREE
mis_rate100 <- c()
for(i in 1:1000){

  Forest_100 <- randomForest::randomForest(trlabels~.,data = train_data,
                                ntree = 100, nodesize = 25, keep.forest = TRUE)
  pre <- predict(Forest_100,newdata = test_data)
  misclassification_rate <- mean(pre!=telabels)
  mis_rate100 <- c(mis_rate100,misclassification_rate)
}

mean_mis_rate100 <- mean(mis_rate100)
var_mis_rate100 <- var(mis_rate100)

# the mean and variable for misclassification rate when B = 10
mean_mis_rate100
var_mis_rate100


################################################################################
################################################################################

set.seed(1234)
x1<-runif(1000)
x2<-runif(1000)
tedata<-cbind(x1,x2)
y<-as.numeric(x1<0.5)
telabels<-as.factor(y)
```

```r
test_data <- as.data.frame(tedata)
test_data <- dplyr::mutate(test_data,telabels)

x1<-runif(100)
x2<-runif(100)
trdata<-cbind(x1,x2)
y<-as.numeric(x1<0.5)
trlabels<-as.factor(y)
train_data <- as.data.frame(trdata)
train_data <- dplyr::mutate(train_data,trlabels)




# QUESTION TWO PART ONE
mis_rate1 <- c()
for(i in 1:1000){

  Forest_1 <- randomForest::randomForest(trlabels~.,data = train_data, ntree = 1
                                         , nodesize = 25, keep.forest = TRUE)
  pre <- predict(Forest_1,newdata = test_data)
  misclassification_rate <- mean(pre!=telabels)
  mis_rate1 <- c(mis_rate1,misclassification_rate)
}

mean_mis_rate1 <- mean(mis_rate1)
var_mis_rate1 <- var(mis_rate1)

# the mean and variable for misclassification rate when B = 1
mean_mis_rate1
var_mis_rate1


# QUESTION two PART TWO
mis_rate10 <- c()
for(i in 1:1000){

  Forest_10 <- randomForest::randomForest(trlabels~.,data = train_data,
                                  ntree = 10, nodesize = 25, keep.forest = TRUE)
  pre <- predict(Forest_10,newdata = test_data)
  misclassification_rate <- mean(pre!=telabels)
  mis_rate10 <- c(mis_rate10,misclassification_rate)
}

mean_mis_rate10 <- mean(mis_rate10)
var_mis_rate10 <- var(mis_rate10)

# the mean and variable for misclassification rate when B = 10
mean_mis_rate10
var_mis_rate10
```

```r
# QUESTION tWO PART THREE
mis_rate100 <- c()
for(i in 1:1000){
  Forest_100 <- randomForest::randomForest(trlabels~.,data = train_data,
                              ntree = 100, nodesize = 25, keep.forest = TRUE)
  pre <- predict(Forest_100,newdata = test_data)
  misclassification_rate <- mean(pre!=telabels)
  mis_rate100 <- c(mis_rate100,misclassification_rate)
}

mean_mis_rate100 <- mean(mis_rate100)
var_mis_rate100 <- var(mis_rate100)

# the mean and variable for misclassification rate when B = 10
mean_mis_rate100
var_mis_rate100



################################################################################
################################################################################
################################################################################

set.seed(1234)
x1<-runif(1000)
x2<-runif(1000)
tedata<-cbind(x1,x2)
y<-as.numeric(((x1<0.5 & x2<0.5)|(x1>0.5 & x2>0.5)))
telabels<-as.factor(y)
test_data <- as.data.frame(tedata)
test_data <- dplyr::mutate(test_data,telabels)


x1<-runif(100)
x2<-runif(100)
trdata<-cbind(x1,x2)
y<-as.numeric(((x1<0.5 & x2<0.5)|(x1>0.5 & x2>0.5)))
trlabels<-as.factor(y)
train_data <- as.data.frame(trdata)
train_data <- dplyr::mutate(train_data,trlabels)

# QUESTION THREE PART ONE
mis_rate1 <- c()
for(i in 1:1000){

  Forest_1 <- randomForest::randomForest(trlabels~.,data = train_data,
                                ntree = 1, nodesize = 12, keep.forest = TRUE)
  pre <- predict(Forest_1,newdata = test_data)
  misclassification_rate <- mean(pre!=telabels)
  mis_rate1 <- c(mis_rate1,misclassification_rate)
}

mean_mis_rate1 <- mean(mis_rate1)
```

```r
var_mis_rate1 <- var(mis_rate1)

# the mean and variable for misclassification rate when B = 1
mean_mis_rate1
var_mis_rate1



# QUESTION THREE PART TWO
mis_rate10 <- c()
for(i in 1:1000){

  Forest_10 <- randomForest::randomForest(trlabels~.,data = train_data,
                                ntree = 10, nodesize = 12, keep.forest = TRUE)
  pre <- predict(Forest_10,newdata = test_data)
  misclassification_rate <- mean(pre!=telabels)
  mis_rate10 <- c(mis_rate10,misclassification_rate)
}

mean_mis_rate10 <- mean(mis_rate10)
var_mis_rate10 <- var(mis_rate10)

# the mean and variable for misclassification rate when B = 10
mean_mis_rate10
var_mis_rate10




# QUESTION THREE PART THREE
mis_rate100 <- c()
for(i in 1:1000){

  Forest_100 <- randomForest::randomForest(trlabels~.,data = train_data,
                                ntree = 100, nodesize = 12, keep.forest = TRUE)
  pre <- predict(Forest_100,newdata = test_data)
  misclassification_rate <- mean(pre!=telabels)
  mis_rate100 <- c(mis_rate100,misclassification_rate)
}

mean_mis_rate100 <- mean(mis_rate100)
var_mis_rate100 <- var(mis_rate100)

# the mean and variable for misclassification rate when B = 10
mean_mis_rate100
var_mis_rate100
```