**Лабораторная работа №7**

**“Бустинг”**

по дисциплине “Машинное обучение”

|  |  |
| --- | --- |
| Выполнил  студент гр. 33504/4 | Череповский Д.К. |
| Руководитель | Селин И.А. |

**Оглавление**

[**Первое задание:** 3](#_Toc511395922)

[**Второе задание:** 4](#_Toc511395923)

[**Третье задание:** 5](#_Toc511395924)

# **Первое задание:**

Исследуйте зависимость тестовой ошибки от количества деревьев в ансамбле для алгоритма adaboost.M1 на наборе данных Vehicle из пакета mlbench (обучающая выборка должна состоять из 7/10 всех прецедентов, содержащихся в данном наборе данных). Постройте график зависимости тестовой ошибки при числе деревьев, равном 1, 11, 21,…, 301, объясните полученные результаты.

**Код программы:**

#Unit 1

library(rpart)

library(mlbench)

library(adabag)

data(Vehicle)

l <- length(Vehicle[,1])

sub <- sample(1:l,l\*(7/10))

mfinal <- 11

maxdepth <- 5

error <- c()

for(i in seq(1,301,by=10))

{

Vehicle.adaboost <- boosting(Class ~.,data=Vehicle[sub,], mfinal=i, maxdepth=maxdepth)

Vehicle.adaboost.pred <- predict.boosting(Vehicle.adaboost, newdata=Vehicle[-sub, ])

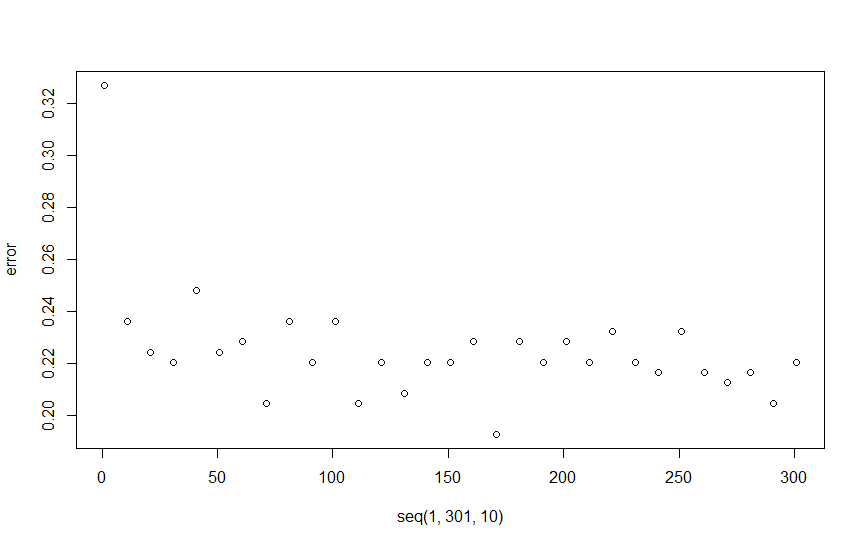
print(Vehicle.adaboost.pred$error)

error <- c(error,Vehicle.adaboost.pred$error)

}

plot(error, x = seq(1, 301, 10))

**Результаты:**



С увеличением максимального количества итераций количество ошибок уменьшается, но незначительно, это обусловлено тем, что увеличение количества итераций ведет к переобучению.

# **Второе задание:**

Исследуйте зависимость тестовой ошибки от количества деревьев в ансамбле для алгоритма bagging на наборе данных Glass из пакета mlbench (обучающая выборка должна состоять из 7/10 всех прецедентов, содержащихся в данном наборе данных). Постройте график зависимости тестовой ошибки при числе деревьев, равном 1, 11, 21,…, 201, объясните полученные результаты.

**Код программы:**

library(mlbench)

library(adabag)

data(Glass)

l <- length(Glass[,1])

sub <- sample(1:l,l\*(7/10))

mfinal <- 25

maxdepth <- 5

error <- c()

for(i in seq(1,201,by=10))

{

Glass.bagging <- bagging(Type ~.,data=Glass[sub,], mfinal=i, maxdepth=5)

Glass.bagging.pred <- predict.bagging(Glass.bagging, newdata=Glass[-sub, ])

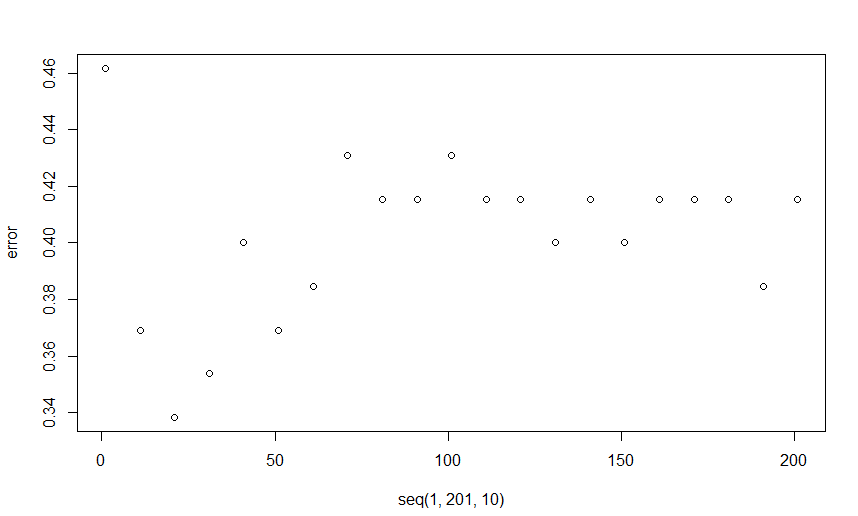
print(Glass.bagging.pred$error)

error <- c(error,Glass.bagging.pred$error)

}

plot(error, x = seq(1, 201, 10))

**Результаты:**



На графике видно, что ошибка с увеличением числа деревьев изменяется, но не значительно, так же влияет на переобучение.

# **Третье задание:**

Реализуйте бустинг алгоритм с классификатором K ближайших соседей. Сравните тестовую ошибку, полученную с использованием данного классификатора на наборах данных Vehicle и Glass, c тестовой ошибкой, полученной с использованием единичного дерева классификации.

**Код программы:**

#Unit 3

learn = function(train, iter)

{

error <- double(iter)

i = 1

classifiers <- list(iter)

while (i < iter) {

error[i] = 1

while (error[i] > 0.1) {

A\_train = train

n <- runif(1, 1, dim(A\_train)[1])

A\_rand <- A\_train[order(runif(n)),]

tmp <- A\_rand[1:n,]

classifier <- kknn(Class ~ ., tmp, A\_train)

tb <- table(A\_train$Class, classifier$fitted.values)

error[i] <- 1 - (sum(diag(tb)) / sum(tb))

}

classifiers[i] = list(tmp)

i = i + 1

}

return (classifiers)

}

run = function(frame, test)

{

answer <- list(dim(test)[1] + 1)

for (i in 1:length(frame)) {

tmp <- kknn(Class ~ ., data.frame(frame[[i]]), test)

answer[i] <- list(tmp$fitted.value)

}

prediction <- double(dim(test)[1])

for (j in 1:dim(test)[1]) {

check <- double(4)

for (i in 1:length(answer))

{

check[answer[[i]][j]] = check[answer[[i]][j]] + 1

}

prediction[j] = levels(factor(test$Class))[which(check == max(check))]

}

tb <- table(test$Class, prediction)

error <- 1 - (sum(diag(tb)) / sum(tb))

return (error)

}

library(kknn)

library(mlbench)

library(rpart)

library(tree)

library(maptree)

library(caret)

library(adabag)

data(Vehicle)

l <- length(Vehicle[, 1])

sub <- sample(1:l, 2 \* l / 3)

boost <- learn(Vehicle[sub, ], 20)

run(boost, Vehicle[-sub,])

Vehicle.rpart <- rpart(Class~.,data=Vehicle[sub,])

Vehicle.rpart.pred <- predict(Vehicle.rpart,newdata=Vehicle[-sub, ],type="class")

tb <- table(Vehicle.rpart.pred,Vehicle$Class[-sub])

error.rpart <- 1-(sum(diag(tb))/sum(tb))

print(error.rpart)

**Результаты:**

[1] 0.3368794

[1] 0.3865248

Расхождение в ошибках есть, но на мой взгляд в пределах нормы. Суть бустинга заключается в использовании слабых классификаторов. На мой взгляд использовать kknn в качестве классификатора для бустинга некорректно.