Computational Statistics (732A90) Lab2

Anubhav Dikshit(anudi287) and Thijs Quast(thiqu264)

23 January 2019

Contents

Question 1: Optimizing a model parameter

1. Import this file to R and add one more variable LMR to the data which is the natural logarithm of Rate. Afterwards, divide the data into training and test sets by using the following code:

```
data <- read.csv2("mortality_rate.csv")
data$LMR <- log(data$Rate)

n=NROW(data)
set.seed(123456)
id=sample(1:n, floor(n*0.5))
train = data[id,]
test = data[-id,]
```

2. Write your own function myMSE() that for given parameters lambda and list pars containing vectors X, Y, Xtest, Ytest fits a LOESS model with response Y and predictor X using loess() function with penalty lambda (parameter enp.target in loess()) and then predicts the model for Xtest. The function should compute the predictive MSE, print it and return as a result. The predictive MSE is the mean square error of the prediction on the testing data. It is defined by the following Equation (for you to implement): predictive MSE = (1/length(test)) * sum (Ytest[i] - fYpred(X[i]))^2 where fYpred(X[i]) is the predicted value of Y if X is X[i]. Read on R's functions for prediction so that you do not have to implement it yourself.

```
myMSE <- function(X, Y , Xtest, Ytest, lambda){

model <- loess(Y ~ X, enp.target=lambda)
predicted <- predict(model, Xtest , se = TRUE)$fit
n <- length(Ytest)

for(i in 1:n){
answer_mse <- (1/n) * sum(Ytest[i] - predicted[i])^2
}
return(answer_mse)
}
```

3. Use a simple approach: use function myMSE(), training and test sets with response LMR and predictor Day and the following lambda values to estimate the predictive MSE values: lambda = 0.1, 0.2, 0.3,. . . ,40

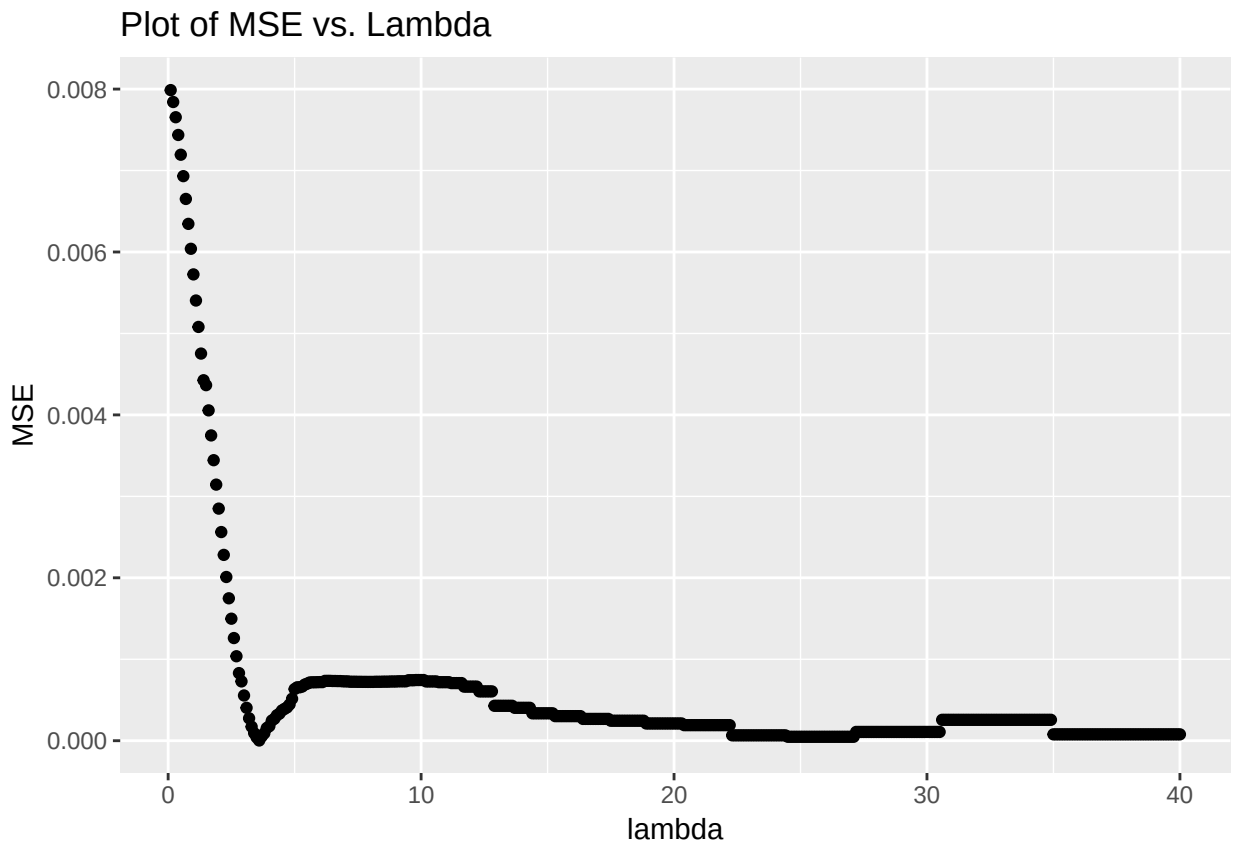
```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
X <- train %>% select(c(Day)) %>% as.matrix()
Y <- train %>% select(c(LMR)) %>% as.matrix()

Xtest <- test %>% select(c(Day)) %>% as.matrix()
Ytest <- test %>% select(c(LMR)) %>% as.matrix()

final <- NULL
for(lambda in seq(from = 0.1, to = 40, by = 0.1)){
temp <- myMSE(X, Y , Xtest, Ytest, lambda=lambda)
temp <- cbind(temp, lambda)
final <- rbind(temp, final)
}

colnames(final) <- c("MSE", "lambda")
```

4. Create a plot of the MSE values versus lambda and comment on which lambda value is optimal. How many evaluations of myMSE() were required (read ?optimize) to find this value?

```
library(ggplot2)

ggplot(data=data.frame(final), aes(x = lambda, y=MSE)) + geom_point() +
  ggtitle("Plot of MSE vs. Lambda")
```



5. Use optimize() function for the same purpose, specify range for search [0:1; 40] and the accuracy 0:01. Have the function managed to and the optimal MSE value? How many myMSE() function evaluations were required? Compare to step 4.

```
optimsied_value <- optimize(myMSE, X=X, Y=Y , Xtest=Xtest, Ytest=Ytest, lambda,
                            lower = 0.1, upper= 40, tol=0.01, maximum = FALSE)

summary(optimsied_value)
```

```
##           Length Class  Mode
## minimum   1      -none- numeric
## objective 1      -none- numeric
```

6. Use optim() function and BFGS method with starting point lambda = 35 to and the optimal value. How many myMSE() function evaluations were required (read ?optim)? Compare the results you obtained with the results from step 5 and make conclusions.

```
#optim(c(35,50),myMSE, X=X, Y=Y , Xtest=Xtest, Ytest=Ytest, method = c("BFGS"))
```

Appendix

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

data <- read.csv2("mortality_rate.csv")
data$LMR <- log(data$Rate)

n=NROW(data)
set.seed(123456)
id=sample(1:n, floor(n*0.5))
train = data[id,]
test = data[-id,]




myMSE <- function(X, Y , Xtest, Ytest, lambda){

model <- loess(Y ~ X, enp.target=lambda)
predicted <- predict(model, Xtest , se = TRUE)$fit
n <- length(Ytest)

for(i in 1:n){
answer_mse <- (1/n) * sum(Ytest[i] - predicted[i])^2
}
return(answer_mse)
}

library(dplyr)

X <- train %>% select(c(Day)) %>% as.matrix()
```

```r
Y <- train %>% select(c(LMR)) %>% as.matrix()

Xtest <- test %>% select(c(Day)) %>% as.matrix()
Ytest <- test %>% select(c(LMR)) %>% as.matrix()

final <- NULL
for(lambda in seq(from = 0.1, to = 40, by = 0.1)){
temp <- myMSE(X, Y , Xtest, Ytest, lambda=lambda)
temp <- cbind(temp, lambda)
final <- rbind(temp, final)
}

colnames(final) <- c("MSE", "lambda")

library(ggplot2)

ggplot(data=data.frame(final), aes(x = lambda, y=MSE)) + geom_point() +
  ggtitle("Plot of MSE vs. Lambda")

optimsied_value <- optimize(myMSE, X=X, Y=Y , Xtest=Xtest, Ytest=Ytest, lambda,
                            lower = 0.1, upper= 40, tol=0.01, maximum = FALSE)

summary(optimsied_value)


#optim(c(35,50),myMSE, X=X, Y=Y , Xtest=Xtest, Ytest=Ytest, method = c("BFGS"))
```