Non-para_exercise

ChoiTaeYoung

2019 10 6

Contents

1 연습문제1

1.1 문제

- 다음과 같은 단순 회귀분석 모형을 가정하자.

$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i, i = 1, 2, \cdots, n.(x_1, \cdots, x_n$:non-random)

$\epsilon_1, \cdots, \epsilon_n$:서로 확률적으로 독립

$E(\epsilon_i) = 0, Var(\epsilon_i) = \sigma^2 (i = 1, \cdots, n)$

위와 같은 회귀분석 모형에서 자료가 $(x_1, y_1), \cdots, (x_n, y_n)$로 주어져 있을 때 최소제곱방법에 의한 β_0와 β_1에 대한 추정량은 각각

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}, \hat{\beta}_1 = \frac{S_{xy}}{S_{xx}} \tag{1}$$

로 주어지고 이 추정량들에 대해 아래와 같은 사실이 알려져 있다.

$$E(\hat{\beta}_0) = \beta_0, E(\hat{\beta}_1) = \beta_1 \tag{2}$$

1. $x_1, \cdots, x_n$은 이미 주어져 있다.
2. 표준정규분포로부터 오차 ϵ_i의 값 n개를 생성 $(\epsilon_i \sim N(0, 1))$
3. 이미 주어진 x_i와 2. 에서 발생시킨 오차를 이용하여 $y_i = 1 + 3x + \epsilon_i, i = 1, \cdots, n$ $(\beta_0 = 1, \beta_1 = 3)$를 만족시키는 종속변수 y의 관측치 $y_1, \cdots, y_n$를 얻는다.
4. 이미 주어진 x_i와 생성된 종속변수 관측치 y_i를 이용하여 최소제곱 추정량 $\hat{\beta}_0^{(1)}, \hat{\beta}_1^{(1)}$를 계산한다.
5. 2,3,4를 500번 반복하여 β_0, β_1의 추정량을 각각 500개씩 얻는다.

1.1.1 n=100일 때 위의 1~5를 수행하는 R프로그램을 작성하시오.

```r
x <- runif(100,0,100)
n <- length(x)
e <- rnorm(n,0,1)
y <- 1+ 3*x + e

data <- cbind(x,y)
betahat0 <- c()
betahat1 <- c()

k <- 500 #number of loop
  for (i in 1:k){
  datam<-data.frame(data)
  mse<-lm(y~x,data=datam)
  betahat0<-append(betahat0,mse$coefficients[1])
  betahat1<-append(betahat1,mse$coefficients[2])
  }
mean(betahat0)
```

```
## [1] 1.022471
```

```r
mean(betahat1)
```

```
## [1] 3.000271
```

2 연습문제2

2.1 문제

- 프로그램을 이용하여, 비모수 회귀모형$Y = m(X) + \epsilon$에서 k-nn 추정법의 실제 성능을 아아보는 시뮬레이션을 한다. 이때 일반적으로 적절한 k값을 정하는 것은 회귀함수를 잘 추정하는 데 있어서 매우 중요한 문제이며, 이를 위해 다음의 두 가지 기준을 고려하고자 한다.

$$RSS(k^*) = \sum_{i=1}^{n} (y_i - \hat{m}(x_i; k^*))^2 \tag{3}$$

$$TE(k^*) = \sum_{i=1}^{n} (m(x_i) - \hat{m}(x_i; k^*))^2 \tag{4}$$

- 단, $\hat{m}(x_i; k^*)$는 $k = k^*$일 때, k-nn방법으로 얻어진 회귀함수 $m(x)$의 x_i에서의 추정지다. 적절한 시뮬 모형을 설정한 다음, 설정한 모형으로부터 $n = 300$자료를 생성해서 아래의 과정을 실행하라.

1. 적절한 코드를 통해 $k^* = 1, \cdots, 20$에 대해 $RSS(k^*), TE(k^*)$를 계산하고, $RSS(k^*), TE(k^*)$를 최소화하는 k^*를 구하라.

install.packages("scales") library(scales)

install.packages("class") library(class)

```r
library(class)
```

```
## Warning: package 'class' was built under R version 3.6.1
```

```r
set.seed(555)

y <- runif(300,-1,1)
y1 <- sort(y, decreasing = F)
x <- seq(1:300)
x1 <- sort(x, decreasing = F)
cl <- factor(c(rep("0",150), rep("1",150)))

data = data.frame(x=c(x1),y=c(y1),cl)

data1 <- as.data.frame((data[1:3]))

idx = sample(1:nrow(data1), 0.7*nrow(data1))
train = data1[idx, ] #
test = data1[-idx, ] #
dim(train) #210 2
```

```
## [1] 210   3
```

```r
dim(test) #90 2
```

```
## [1] 90  3
```

```r
#      y          -
train_y <- data1[idx, 3] #      diagnosis
test_y <- data1[-idx, 3] #      diagnosis

# k    -
a <- sqrt(dim(train))
opti_k <- a[1]
```

```r
pred1 <- knn(train, test, train_y, k=opti_k)
pred1 #
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [36] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
## [71] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## Levels: 0 1
```

```r
table(pred1, test_y)
```

```
##      test_y
## pred1  0  1
##     0 58  0
##     1  0 32
```

```r
result <- numeric()
k = 1:20
for(i in k ){
    all_pred <- knn(train, test, train_y, k=i)
    t <- table(all_pred, test_y)
    result[i] <- (t[1,1]+t[2,2])/sum(t)
}

result
```

```
##  [1] 1.0000000 0.9888889 0.9888889 0.9888889 0.9888889 0.9888889 0.9888889
##  [8] 0.9888889 0.9888889 0.9888889 0.9888889 0.9888889 0.9888889 0.9888889
## [15] 0.9888889 1.0000000 0.9888889 1.0000000 0.9888889 0.9777778
```

```r
which(result==max(result))
```

```
## [1]  1 16 18
```

2. 1. 에서 두가지 방법에 의해 구한 두 개의 k^*를 이용하여 적절한 x값 grid위에서 $\hat{m}(x)$를 그려보아라. 두 개의 $\hat{m}(x)$값의 추정치를 비교해보고, $RSS(k^*), TE(k^*)$ 기준방법을 비교하라.