HW4_simulation

Cheng Zhong

16307110259

2019.4.8

实验题目： 使用 Newton-Raphson 算法模拟逻辑斯蒂回归的过程

实验过程：

一、参数初始化

我们先生成一组均值为 0，协方差矩阵为单位矩阵的 X1，X2 (样本量为 200)，随后根据真实β通过逻辑斯蒂函数计算出 y=1 的概率值 p，最后按此概率 p 根据二项分布生成模拟值 y'。

二、Newton-Raphson 算法估算β

根据逻辑斯蒂函数，有

$$P(x_i,\beta)=P(Y=1|x_i)=\frac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}}$$

估算β的过程即为计算出一个最优解β'，使得最大似然函数

$$L(\beta)=\prod_{i=1}^{N}[P(x_i,\beta)^{y_i}(1-P(x_i,\beta)^{1-y_i})]$$

取得最大值的过程，在 β' = argmax L(β) 时，模拟程度最好。

根据 Newton-Raphson 算法，有

$$\beta^{new}=\beta^{old}-(\frac{\nabla^2 L(\beta)}{\nabla\beta\nabla\beta^T})^{-1}\frac{\nabla L(\beta)}{\nabla\beta}$$

对最大似然函数两端取对数后求导，可得

$$\frac{\nabla L(\beta)}{\nabla \beta}=x^T(y-p)$$

$$\frac{\nabla^2 L(\beta)}{\nabla \beta \nabla \beta^T}=x^T wx, w=diag(P(x_i,\beta)*(1-P(x_i,\beta)))$$

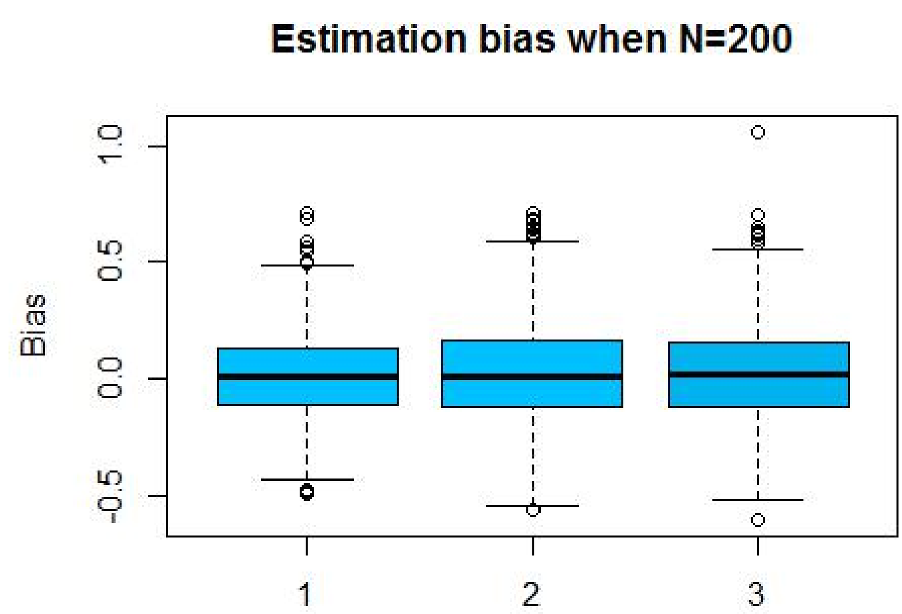
三、比较样本量对模拟效果的影响

设置样本量分别为 200，500，800，1000，观察模拟值与真实值的差别

实验代码与运行结果：

```
rm(list = ls())
library(MASS)

## Warnin : package 'MASS' was built under R version 3.5.3

         g

beta_true = c(0.5,1.2,1)
Sigma <- matrix(c(1,0,0,1),2,2)#Set the co-variance matrix
length <- 200                    #Sample size = 200
e200 <- matrix(1,nrow = 1000,ncol = 3)
for (j in 1:1000)
{
  r <- mvrnorm(n=length, rep(0,2), Sigma)
  x <- cbind(1,r)

                             # Initialize x by normal distribution
  p_true <- as.vector(exp(x %*% beta_true)/(1 + exp(x %*% beta_true)))


  y <- rep(1,length)
  for (i in 1:length)
  {
    y[i] <- rbinom(1,1,p_true[i])
  }                                # Initialize y by binomial distribution


  beta <- c(1,1,1)                # Initialize β
  trans <- matrix(1,1,1)
  while (norm(trans) > 1e-10)
  {
    p <- as.vector(exp(x %*% beta)/(1 + exp(x %*% beta)))
    prime1 <- t(x) %*% (y - p)  #Calculate the first-order derivative
```

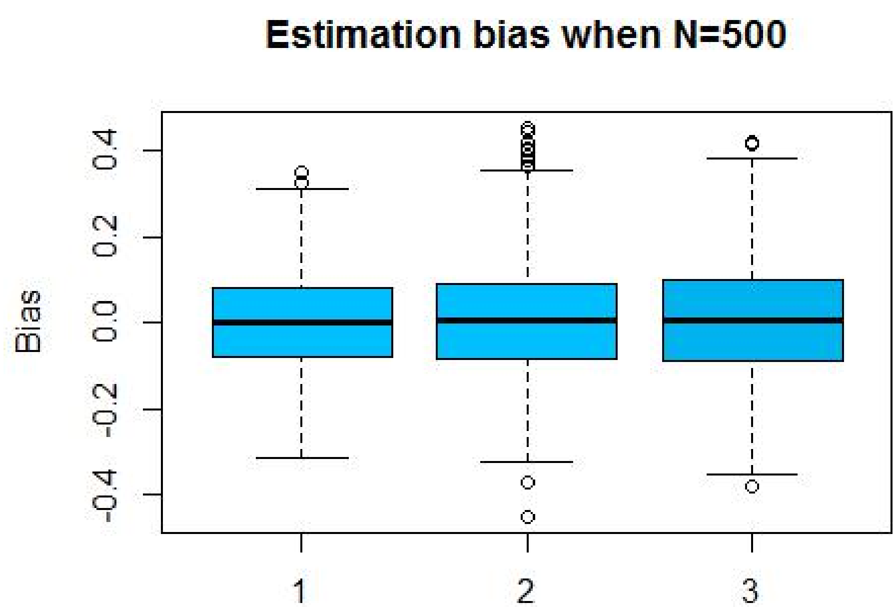
```
    W = diag(p * (1-p))
    prime2_inv <- solve((t(x) %*% W %*% x))
                          #Calculate the inverse of second-order derivative
    trans <- prime2_inv %*% prime1
    beta = beta + trans
  }
  e200[j,] <- t(beta - beta_true)
}
boxplot(e200, group = c("w","B1","B2"), ylab = 'Bias',main = 'Estimation
 bias when N=200',col = c("deepskyblue","deepskyblue1","deepskyblue2"))
# Draw box-plot of the bias between the true value and the estimated val
ue
```



图一 样本量为 200 时真实值与模拟值的误差箱线图（横轴为 w，β1，β2）

```
length <- 500 #change the sample size
e500 <- matrix(1,nrow = 1000,ncol = 3)
for (j in 1:1000)
{
  r <- mvrnorm(n=length, rep(0,2), Sigma)
  x <- cbind(1,r)
  p_true <- as.vector(exp(x %*% beta_true)/(1 + exp(x %*% beta_true)))
  y <- rep(1,length)
  for (i in 1:length)
  {
```

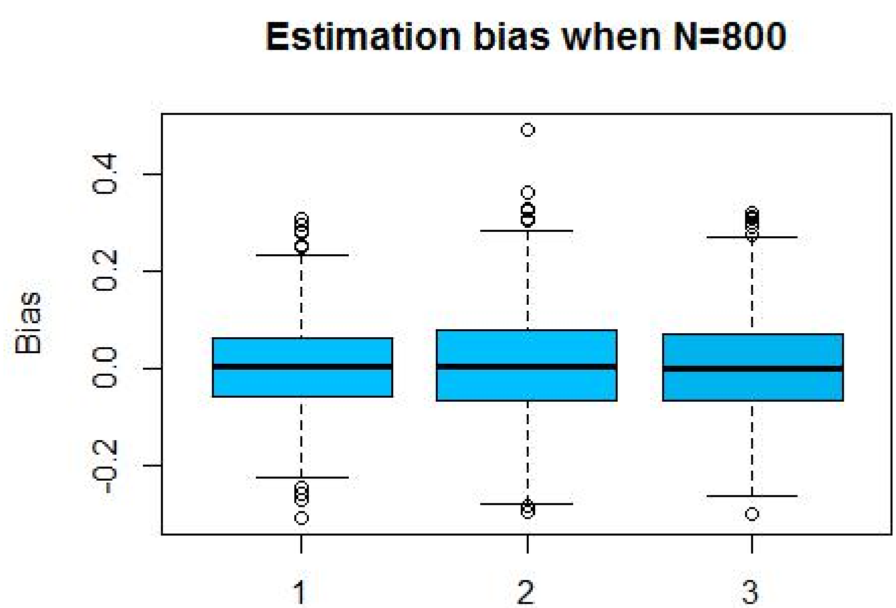
```
    y[i] <- rbinom(1,1,p_true[i])
  }

  beta <- c(1,1,1)
  trans <- matrix(1,1,1)
  while (norm(trans) > 1e-10)
  {
    p <- as.vector(exp(x %*% beta)/(1 + exp(x %*% beta)))
    prime1 <- t(x) %*% (y - p)
    W = diag(p * (1-p))
    prime2_inv <- solve((t(x) %*% W %*% x))
    trans <- prime2_inv %*% prime1
    beta = beta + trans
  }
  e500[j,] <- t(beta - beta_true)
}
boxplot(e500,ylab = 'Bias',main = 'Estimation bias when N=500',col = c("
deepskyblue","deepskyblue1","deepskyblue2"))
```



图二 样本量为 500 时真实值与模拟值的误差箱线图（横轴为 w，β1，β2）

```
length <- 800 #change the sample size
e800 <- matrix(1,nrow = 1000,ncol = 3)
for (j in 1:1000)
{
  r <- mvrnorm(n=length, rep(0,2), Sigma)
  x <- cbind(1,r)
  p_true <- as.vector(exp(x %*% beta_true)/(1 + exp(x %*% beta_true)))
  y <- rep(1,length)
```

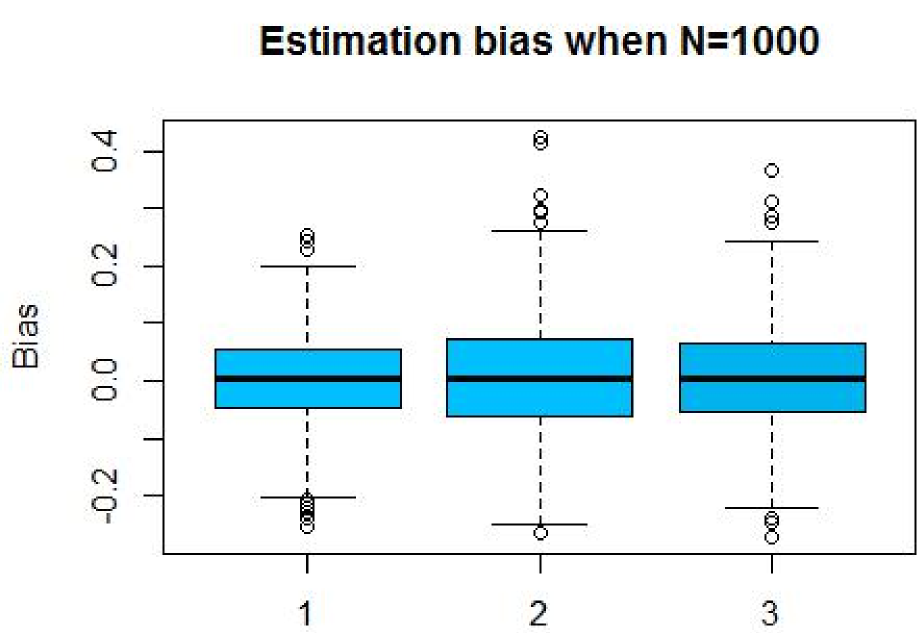
```
  for (i in 1:length)
  {
    y[i] <- rbinom(1,1,p_true[i])
  }

  beta <- c(1,1,1)
  trans <- matrix(1,1,1)
  while (norm(trans) > 1e-10)
  {
    p <- as.vector(exp(x %*% beta)/(1 + exp(x %*% beta)))
    prime1 <- t(x) %*% (y - p)
    W = diag(p * (1-p))
    prime2_inv <- solve((t(x) %*% W %*% x))
    trans <- prime2_inv %*% prime1
    beta = beta + trans
  }
  e800[j,] <- t(beta - beta_true)
}
boxplot(e800,ylab = 'Bias',main = 'Estimation bias when N=800',col = c("
deepskyblue","deepskyblue1","deepskyblue2"))
```



图三 样本量为 800 时真实值与模拟值的误差箱线图（横轴为 w，β1，β2）

```
length <- 1000 #change the sample size
e1000 <- matrix(1,nrow = 1000,ncol = 3)
for (j in 1:1000)
{
  r <- mvrnorm(n=length, rep(0,2), Sigma)
```

```
  x <- cbind(1,r)
  p_true <- as.vector(exp(x %*% beta_true)/(1 + exp(x %*% beta_true)))
  y <- rep(1,length)
  for (i in 1:length)
  {
    y[i] <- rbinom(1,1,p_true[i])
  }

  beta <- c(1,1,1)
  trans <- matrix(1,1,1)
  while (norm(trans) > 1e-10)
  {
    p <- as.vector(exp(x %*% beta)/(1 + exp(x %*% beta)))
    prime1 <- t(x) %*% (y - p)
    W = diag(p * (1-p))
    prime2_inv <- solve((t(x) %*% W %*% x))
    trans <- prime2_inv %*% prime1
    beta = beta + trans
  }
  e1000[j,] <- t(beta - beta_true)
}
boxplot(e1000,ylab = 'Bias',main = 'Estimation bias when N=1000',col = c
("deepskyblue","deepskyblue1","deepskyblue2"))
```

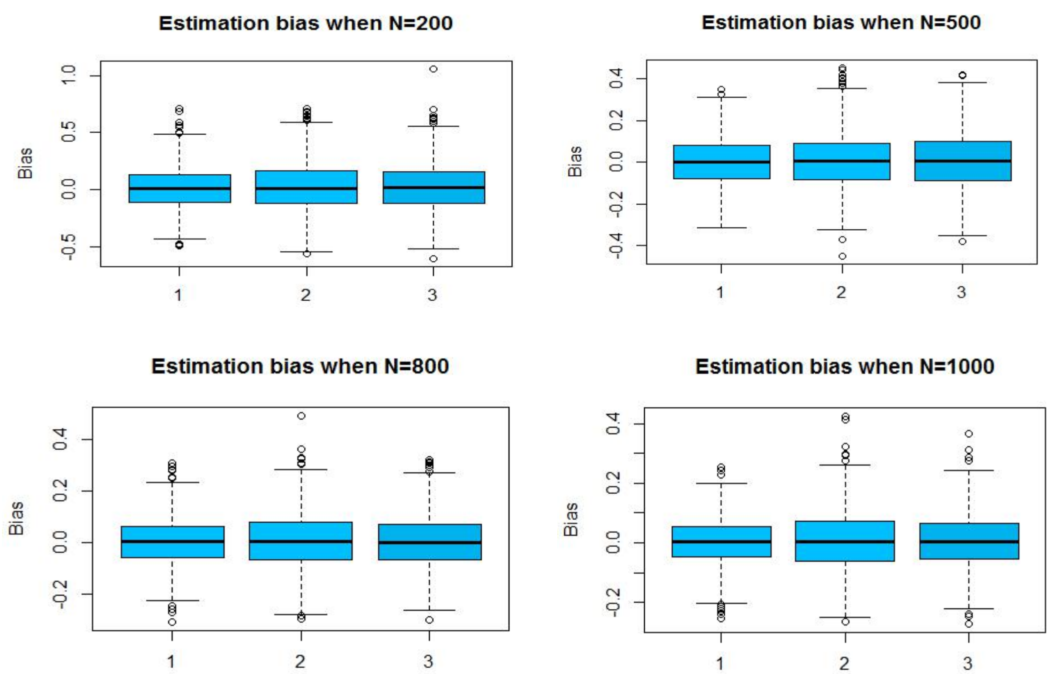


图四 样本量为 1000 时真实值与模拟值的误差箱线图（横轴为 w，β1，β2）

实验结果：

一、 样本量为 200，500，800，1000 时，模拟得到的β均能收敛到真实值附近，故可以认为 Newton-Raphson 算法是一种计算逻辑斯蒂回归中β值的有效算法。

二、 将四种样本量的偏差箱线图横向比较，可以发现当样本量越大时，模拟β值的偏差越小，模拟效果越好。



图五 样本量为 200，500，800，1000 时真实值与模拟值的误差箱线图横向比较

	W		β1		β2	
N 值	均值	方差	均值	方差	均值	方差
200	0.013	0.033	0.033	0.052	0.017	0.046
500	7e-5	0.013	0.016	0.020	0.012	0.018
800	0.002	0.008	0.010	0.012	0.006	0.011
1000	0.005	0.007	0.008	0.010	0.010	0.009

表一 不同样本量模拟结果偏差的均值与方差比较