Homework 5

Hari Krishna Reddy

2024-03-26

## Problem 1. The Cauchy distribution with scale 1 has following density function f (x) = 1/π [1 + (x − η)^2] , −∞ < x < ∞. Here η is the location parameter.

### 1. Use rcauchy() function to generate 1000 random samples from a Cauchy distribution with η = 1 and scale 1.

set.seed(318)  
n=1000  
m=1   
random\_samples=rcauchy(n,m,1)

### 2. Use the bisection method to find the maximum likelihood estimator of η. Use the random samples you obtained from 1. as sample observations.

mle <- function(m)  
{  
 sum((2\*(random\_samples-m))/(1+(random\_samples-m)^2))  
}  
uniroot(mle, c(-3, 3))$root

## [1] 0.9900273

### 3. Use the Newton’s method to find the maximum likelihood estimator of η. Use the random samples you obtained from 1. as sample observations.

library(numDeriv)  
  
mle\_n <- function(m, n = 1000, rs = random\_samples)  
{  
 -n\*log(pi) - sum(log(1+(rs-m)^2))  
}  
  
m <- 2  
  
for(i in 1:20)  
{  
 z <- genD(func = mle\_n, x = m)$D[1]  
 z\_1 <- genD(func = mle\_n, x = m)$D[2]  
 result <- m - z/z\_1  
 m <- result  
 print(result, digits = 10)  
}

## [1] 0.3638831125  
## [1] 1.10660716  
## [1] 0.9894755088  
## [1] 0.9900314852  
## [1] 0.9900315104  
## [1] 0.9900315102  
## [1] 0.9900315107  
## [1] 0.990031511  
## [1] 0.9900315106  
## [1] 0.9900315105  
## [1] 0.9900315106  
## [1] 0.9900315106  
## [1] 0.9900315109  
## [1] 0.9900315105  
## [1] 0.9900315103  
## [1] 0.9900315104  
## [1] 0.9900315103  
## [1] 0.9900315107  
## [1] 0.9900315103  
## [1] 0.9900315107

## Problem 2

### 1. Load the data set uploaded in course Elearning. (Hint: to import .txt file into R, use read.table() function and you may also need header = T argument inside the function.)

data <- read.table("poisson.txt",header=T)  
head(data)

## x y  
## 1 2 0  
## 2 15 6  
## 3 19 4  
## 4 14 1  
## 5 16 5  
## 6 15 2

### 2. Use the Newton’s method with initial value β0 = 0 and β1 = 0 to find the maximum likelihood estimator of β0 and β1.

attach(data)  
  
g <- function(x, y, beta\_0, beta\_1)  
{  
 g <- rep(0, 2)  
 g[1] <- sum(- exp(beta\_0 + beta\_1\*x)+y)  
 g[2] <- sum((- exp(beta\_0 + beta\_1\*x)+y)\*x)  
 return(g)  
}  
  
new <- function(x, beta\_0, beta\_1)  
{  
 new <- matrix(0, nrow = 2, ncol = 2)  
 new[1,1] <- -sum(exp(beta\_0+beta\_1\*x))  
 new[1,2] <- -sum((exp(beta\_0+beta\_1\*x))\*x)  
 new[2,1] <- new[1,2]  
 new[2,2] <- -sum((exp(beta\_0+beta\_1\*x))\*(x^2))  
 return(new)  
}  
  
beta <- c(0,0)  
  
for(i in 1:10)  
{  
 beta\_2 <- beta - solve(new(x = x, beta\_0 = beta[1], beta\_1 = beta[2])) %\*% g(x = x, y = y, beta\_0 = beta[1], beta\_1 = beta[2])  
 beta <- beta\_2  
 print(beta\_2, digits = 10)  
}

## [,1]  
## [1,] -0.9927826185  
## [2,] 0.3069817883  
## [,1]  
## [1,] -1.5667955969  
## [2,] 0.2886303191  
## [,1]  
## [1,] -1.5912315367  
## [2,] 0.2468418789  
## [,1]  
## [1,] -0.8671073309  
## [2,] 0.1746897095  
## [,1]  
## [1,] -0.01703712108  
## [2,] 0.10610803790  
## [,1]  
## [1,] 0.27383427001  
## [2,] 0.07979304978  
## [,1]  
## [1,] 0.30731352929  
## [2,] 0.07641494587  
## [,1]  
## [1,] 0.30786623056  
## [2,] 0.07635734211  
## [,1]  
## [1,] 0.30786639562  
## [2,] 0.07635732483  
## [,1]  
## [1,] 0.30786639562  
## [2,] 0.07635732483

f <- function(beta, y=data$y , x=data$x)  
{  
 (-1)\*sum(-exp(beta[1]+beta[2]\*x)+(beta[1]+beta[2]\*x)\*y-log(factorial(y)))  
}  
  
optim(c(0,0), f)$par

## [1] 0.30697324 0.07640861

### 3. When we want to run Poisson regression in R, we use glm() function with family = “poisson”. Compare your parameter estimation results in 2. to glm() output.

pr <- glm(y ~ x, family = poisson(), data=data)  
summary(pr)

##   
## Call:  
## glm(formula = y ~ x, family = poisson(), data = data)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.30787 0.28943 1.064 0.287   
## x 0.07636 0.01730 4.413 1.02e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 48.310 on 29 degrees of freedom  
## Residual deviance: 27.842 on 28 degrees of freedom  
## AIC: 124.5  
##   
## Number of Fisher Scoring iterations: 4

pr$coefficients

## (Intercept) x   
## 0.30786640 0.07635732