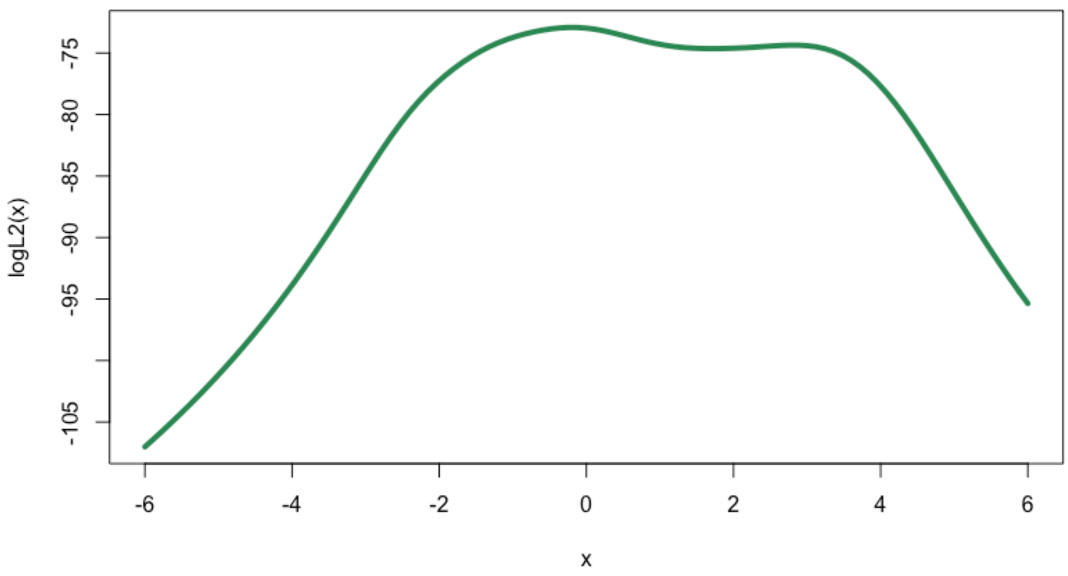
刘程华 2018011687 计91

Prob 1. **The following data are an i.i.d. sample from a Cauchy** $(\theta, 1)$ **distribution:1.77,-0.23,2.76,3.80,3.47,56.75,-1.34,4.24,-2.44,3.29,3.71,-2.40,4.53,-0.07,-1.05,-13.87,-2.53,-1.75,0.27,43.21**

(a) Graph the log likelihood function. Apply the bisection method with starting points -1 and 1. **Use additional runs to illustrate manners in which the bisection method may fail to find the global maximum.**

Graph the log likelihood function：




```
x<-c(1.77,-0.23, 2.76, 3.80, 3.47, 56.75, -1.34, 4.24, -2.44, 3.29, 3.71, -2.40,
4.53, -0.07, -1.05, -13.87, -2.53, -1.75, 0.27, 43.21)
logL<-function(theta)sum(dcauchy(x,theta,log = TRUE))
logL2<-function(theta)sapply(theta, logL)
curve(logL2(x),-6,6,lwd=4,col="seagreen")
```

the bisection method :

```
g<-function(theta){
  1/((pi)*(1+(x-theta)^2))
}
dg<-function(theta){
  2*sum((x-theta)/(1+(x-theta)^2))
}
a<--1
b<-1
theta<-(a+b)/2
inter<-50
```

```
for(i in 1:inter){
  if(dg(theta)!=0&&dg(a)*dg(theta)<0){
    b=theta
  }else{
    a=theta
  }
  theta<-(a+b)/2
}
```

$\hat{\theta} = -0.1922866$

Changed a from -1 to 0 and b from 1 to 3,we get $\hat{\theta} = 1.713587$, which is not the global maximum!

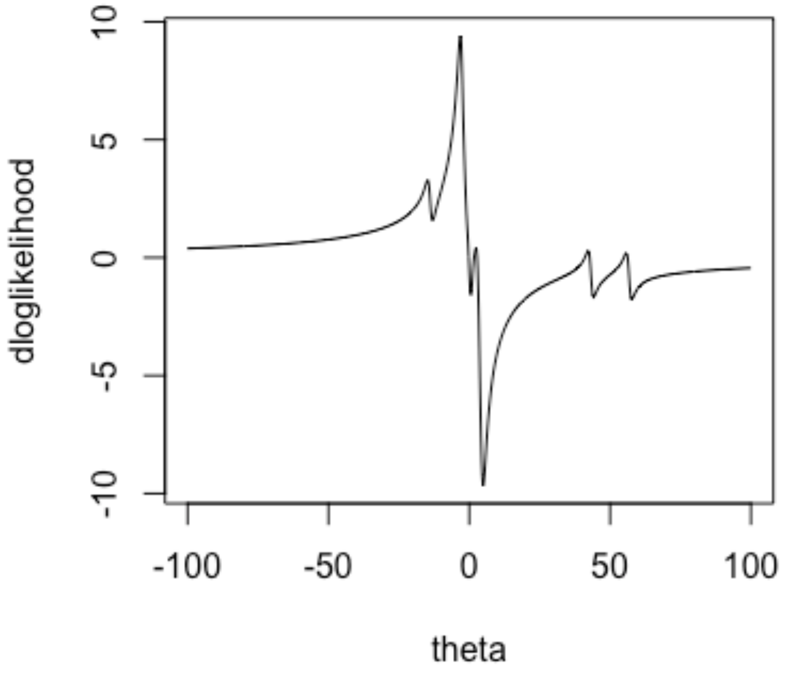
(b) Finish the R **code of Newton's method on** p.32. **Find the** MLE **for** θ **using the Newton-Raphson method. Try all of the following starting points:** $-11, -1, 0, 1.5, 4, 4.7, 7, 8,$ **and** 38. **Discuss your results. Is the mean of the data a good starting point?**

```
f <- function(x) {
  (1+1/x-log(x))/(1+x)^2
}
Newton <- function(f, x0)
{
  tol = 1E-12
  epsilon = .Machine$double.eps*4
  df = (f(x0+tol)-f(x0-tol))/2/tol
  x = x0-f(x0)/df
  p = x
  i = 1
  while(1){
    df = (f(x+tol)-f(x-tol))/2/tol
    x = x-f(x)/df
    if(is.nan(x) || abs(x-p[i])<epsilon){
      p = cbind(p,x)
      i = i+1
      break
    }
    i = i+1
    p = cbind(p,x)
  }
  return(list(root = p[i], process = p[1:i]))
}
x0 =  c(-11, -1, 0, 1.5, 4, 4.7, 7, 8, 38)
f = dg
for(i in 1:9){
  print(c(x0[i],Newton(dg,x0[i])$root))
}
```

Output:

```
[1] -11 NaN
[1] -1.0000000 -0.1922866
[1]  0.0000000 -0.1922866
[1] 1.500000 1.713587
[1] 4.000000 2.817472
[1]  4.7000000 -0.1922866
[1]  7.00000 41.04085
[1]   8 NaN
[1] 38.00000 42.79538
```

the image of the derivative of $\ell(x;\theta)$ is as follows:



The initial value has a great influence on it. If the initial value deviates far, it will easily fall into the local optimal or not converge.The mean of the data is around 5, which is not suitable as an initial value!

(c) Apply the Fisher Scoring method. Compare it with the NewtonRaphson method, comment on the similarities and differences.

```
fisher_scoring <- function(f, x0 = 1)
{
  tol=1E-12
  epsilon = 1e-10
  df = (f(x0+tol)-f(x0-tol))/2/tol
  I = 1.5625
  x = x0+f(x0)/I
  p = x
  i = 1
```

```
  while(1){
    df = (f(x+tol)-f(x-tol))/2/tol
    x = x+f(x)/I
    #print(x)
    if(is.nan(x) || abs(x-p[i])<epsilon){
      p = cbind(p,x)
      i = i+1
      break
    }
    i = i+1
    p = cbind(p,x)
  }
  return(list(root = p[i], process = p[1:i]))
}
x0 =  c(-11, -1, 0, 1.5, 4, 4.7, 7, 8, 38)
f = dg
for(i in 1:9){
  print(c(x0[i],fisher_scoring(dg,x0[i])$root))
}
```

Output:

```
[1] -11.0000000  -0.1922866
[1] -1.0000000 -0.1922866
[1]  0.0000000 -0.1922866
[1]  1.5000000 -0.1922866
[1]  4.0000000 -0.1922866
[1]  4.7000000 -0.1922866
[1] 7.000000 2.817472
[1]  8.0000000 -0.1922866
[1] 38.0000000 -0.1922866
```

In the Newton Raphson algorithm, the second derivative (matrix) of the loss function needs to be obtained during parameter estimation. In Fisher scoring, we use the expectation of the second derivative matrix instead. This is the difference between the two. In GLM, when the link function is canonical link, the two algorithms are equivalent, such as logistic regression.

In conclusion, Fisher scoring is a hill-climbing algorithm for getting results - it maximizes the likelihood by getting successively closer and closer to the maximum by taking another step (an iteration). It knows when it has reached the top of the hill in that taking another step does not increase the likelihood.

Prob 2. What convergence order does the Bisection method have and why?

Let $a_0 = a$ and $b_0 = b$ and $[a_n, b_n]$ $(n \geq 0)$ are the successive intervals in the bisection process. Clearly

$$a_0 \leq a_1 \leq a_2 \leq \cdots \leq b_0 = b$$

and

$$b_0 \geq b_1 \geq b_2 \geq \cdots \geq a_0 = a$$

Now the sequence $\{a_n\}$ is monotonic increasing and bounded above and the sequence $\{b_n\}$ is monotonic decreasing and bounded below. Hence both the sequences converge. Further,

$$b_n - a_n = \frac{b_{n-1} - a_{n-1}}{2} = \cdots = \frac{b-a}{2^n}$$

Hence $\lim_{n\to\infty} a_n = \lim_{n\to\infty} b_n = \alpha$. Further, taking limit in $f(a_n) f(b_n) \leq 0$, we get $[f(r)]^2 \leq 0$ and that implies $f(r) = 0$. Hence, both a_n and b_n converges to a root of $f(x) = 0$. Let us apply the bisection method to the interval $[a_n, b_n]$ and calculate midpoint $c_n = (a_n + b_n)/2$. Then the root lies either in $[a_n, c_n]$ or $[c_n, b_n]$. In either case

$$|\alpha - c_n| \leq \frac{b_n - a_n}{2} = \frac{b-a}{2^{n+1}}$$

Hence, $c_n \to \alpha$ as $n \to \infty$ and inequality is tight. In this method, we can calculate the number of iteration n that need to be done to achieve a specified accuracy. Suppose we want relative accuracy ϵ of the root. Hence we want

$$\frac{|\alpha - c_n|}{|\alpha|} \leq \epsilon$$

Suppose that the root lies in $[a, b]$ where $b > a > 0$. Clearly $|\alpha| > a$ and hence the above relation is true if

$$\frac{|\alpha - c_n|}{a} \leq \epsilon$$

which is true if

$$\frac{b-a}{2^{n+1}a} \leq \epsilon$$

Solving this we can find minimum number of iteration needed to obtain the desired accuracy. Now

$$|e_{n+1}| = |\alpha - c_{n+1}| \sim \frac{1}{2}(b_{n+1} - a_{n+1}) = \frac{1}{2}\frac{b_n - a_n}{2}$$

and

$$|e_n| = |\alpha - c_n| \sim \frac{1}{2}(b_n - a_n)$$

Thus we find

$$|e_{n+1}| \sim \frac{1}{2}|e_n|$$

Hence the bisection method converges linearly.

Convergence order p=1.

Prob 3. **Calculate the convergence order of the secant method.**

Here we don't insist on bracketing of roots. Given two initial guess. Given two approximation x_{n-1}, x_n, we take the next approximation x_{n+1} as the intersection of line joining $(x_{n-1}, f(x_{n-1}))$ and $(x_n, f(x_n))$ with the x -axis. Thus x_{n+1} need not lie in the interval $[x_{n-1}, x_n]$. If the root is α and α is a simple zero, then it can be proved that the method converges for initial guess in sufficiently small neighbourhood of α. Now x_{n+1} is given by

$$x_{n+1} = \frac{x_{n-1} f(x_n) - x_n f(x_{n-1})}{f(x_n) - f(x_{n-1})}$$

Now

$$\begin{aligned} e_{n+1} = \alpha - x_{n+1} &= \alpha - \frac{x_{n-1} f(x_n) - x_n f(x_{n-1})}{f(x_n) - f(x_{n-1})} \\ &= \alpha - x_n + \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n) \\ &= \alpha - x_i + \frac{f(x_n) - f(\alpha)}{f[x_n, x_{n-1}]} \\ &= -(\alpha - x_n)(\alpha - x_{n-1}) \frac{f[\alpha, x_n, x_{n-1}]}{f[x_n, x_{n-1}]} \\ &= -e_n e_{n-1} \frac{-f''(\eta_1)}{2f'(\eta_2)} \end{aligned}$$

where $\eta_1 \in I(x_n, x_{n-1}, x_n)$ and $\eta_2 \in I(x_n, x_{n-1})$. Here, $I(a, b, c)$ denotes the interior of the interval formed by a, b and c. since α is a simple zero, $f'(\alpha) \neq 0$. Consider the interval $J = \{x : |x - \alpha| \leq \delta\}$ such that

$$\left| \frac{-f''(\eta_1)}{2f'(\eta_2)} \right| \leq M, \quad \eta_1, \eta_2 \in J$$

Now we have

$$|e_{n+1}| \leq M |e_n| |e_{n-1}|$$

Let $\varepsilon_n = M|e_n|$. Then $\varepsilon_{n+1} \leq \varepsilon_n \varepsilon_{n-1}$. Now choose initial guess x_0 and x_1 such that

$$|x_i - \alpha| < \min\{1/M, \delta\}, \quad i = 0, 1$$

This implies $\varepsilon_i = M|x_i - \alpha| < \min\{1, M\delta\}$ for $i = 0, 1$. Now choose $0 < D < \min\{1, M\delta\}$ and thus $0 < D < 1$ and $\varepsilon_0, \varepsilon_1 \leq D < 1$. Now

$$\varepsilon_2 \leq \varepsilon_1 \varepsilon_0 \leq D^2$$

Also, $\varepsilon_3 \leq \varepsilon_2 \varepsilon_1 \leq D^3$ etc. By induction, we can show that $\varepsilon_n \leq D^{\lambda_n}$ where $\lambda_0 = \lambda_1 = 1$ and $\lambda_n = \lambda_{n-1} + \lambda_{n-2}$ for $n \geq 2$. Using $\lambda_n \propto r^n$, we find

$$\lambda_n = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^{n+1} - \left(\frac{1-\sqrt{5}}{2} \right)^{n+1} \right] \sim \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^{n+1}, \quad \text{as} \quad n \to \infty$$

since $D < 1$ and $\lambda_n \to \infty$, we get $\varepsilon_n \to 0$ as $n \to \infty$ and thus $x_n \to \alpha$ Now as $x_n \to \alpha, \eta_1, \eta_2 \to \alpha$. This implies

$$|e_{n+1}| \sim C |e_n| |e_{n-1}|, \quad C = \left| \frac{f''(\alpha)}{2f'(\alpha)} \right|$$

Let

$$|e_{n+1}| \sim C^{\beta}|e_n|^p \Longrightarrow |e_n| \sim C^{\beta}|e_{n-1}|^p \Longrightarrow |e_{n-1}| = \operatorname{sim} C^{-\beta/p}|e_n|^{1/\beta}$$

Now from $|e_{n+1}| \sim C\,|e_n|\,|e_{n-1}|$, we get

$$C|e_n|^p \sim C|e_n||e_n|^{1/p} C^{-\beta/p}$$

which is true provided

$$p = 1 + 1/p, \quad \beta = 1 - \beta/p \Longrightarrow \beta = p/(1+p) = p - 1$$

Taking the positive value of p, we find $p = (1+\sqrt{5})/2 = r(\text{ golden ratio })$ and $\beta = r - 1$. Hence

$$|e_{n+1}| \sim C^{r-1}|e_n|^r$$

Convergence order $p = (1+\sqrt{5})/2$.