HW3

Qi Qi

9/20/2018

1.

$$f(x_i;\theta) = \frac{1}{\pi[1+(x_i-\theta)^2]}$$

$$\ell = \sum_{i=1}^{n} logf(X_i;\theta) = \sum_{i=1}^{n} -log(\pi[1+(X_i-\theta)^2]) = -nlog\pi - \sum_{i=1}^{n} log[1+(X_i-\theta)^2]$$

$$\ell' = \frac{\partial \ell}{\partial \theta} = -\sum_{i=1}^{n} \frac{2\theta - 2X_i}{1+(X_i-\theta)^2} = -2\sum_{i=1}^{n} \frac{\theta - X_i}{1+(X_i-\theta)^2}$$

$$\ell'' = \frac{\partial \ell'}{\partial \theta} = -2\sum_{i=1}^{n} \frac{1+(X_i-\theta)^2 - 2(\theta-X_i)^2}{[1+(X_i-\theta)^2]^2} = -2\sum_{i=1}^{n} \frac{1-(X_i-\theta)^2}{[1+(X_i-\theta)^2]^2}$$

$$I_n(\theta) = nE[(\frac{\partial logf(x;\theta)}{\partial \theta})^2] = nE[\frac{4(X-\theta)^2}{[1+(X-\theta)^2]^2}] = 4n\int_R \frac{(x-\theta)^2}{[1+(x-\theta)^2]^2}\frac{1}{\pi[1+(x-\theta)^2]}dx$$

$$= 4n\int_R \frac{x^2}{[1+x^2]^2}\frac{1}{\pi[1+x^2]}dx = \frac{4n}{\pi}\int_R \frac{x^2}{(1+x^2)^3}dx = \frac{8n}{\pi}\int_0^\infty \frac{x^2}{(1+x^2)^3}dx$$

Let $y = \frac{1}{1+x^2}$, then $x = (\frac{1}{y}-1)^{1/2}$, $dx = -\frac{1}{2}(\frac{1}{y}-1)^{-1/2}y^{-2}dy$.

$$I_n(\theta) = \frac{8n}{\pi}\int_1^0 (\frac{1}{y}-1)y^3(-\frac{1}{2})(\frac{1}{y}-1)^{-1/2}y^{-2}dy$$

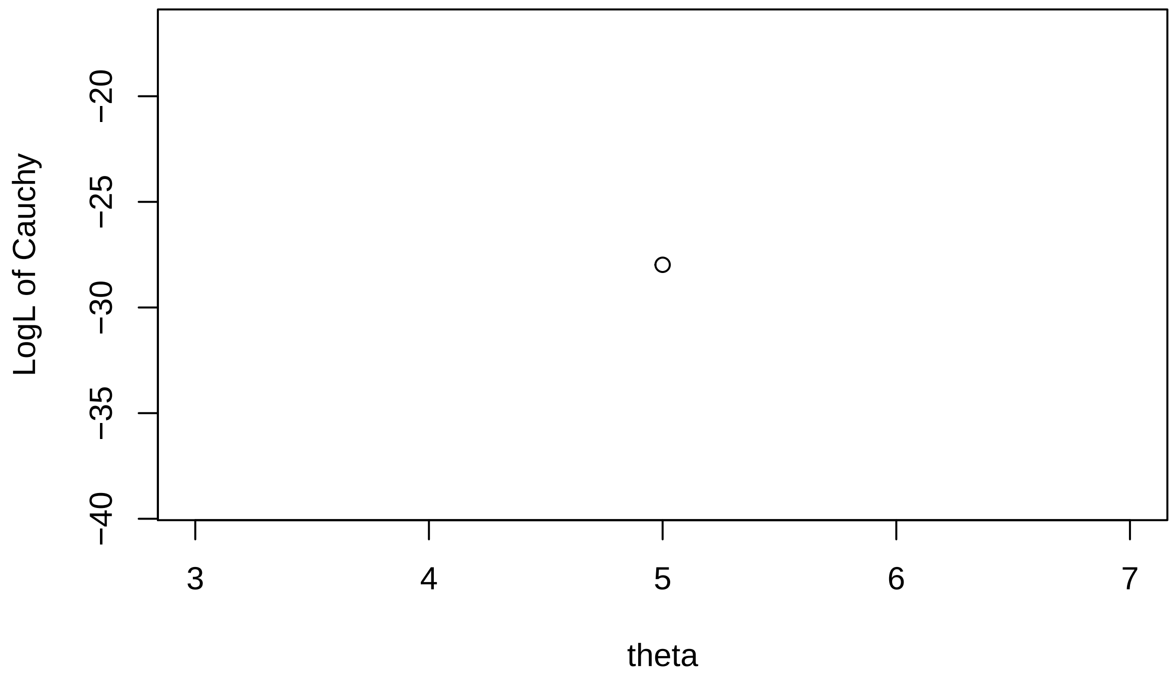
$$= \frac{4n}{\pi}\int_0^1 y^{1/2}(1-y)^{1/2}dy = \frac{4n}{\pi}\frac{\Gamma(3/2)\Gamma(3/2)}{\Gamma(3)} = \frac{4n}{\pi}\frac{(\frac{1}{2}\pi)^2}{2!} = \frac{n}{2}$$

2.

Loglikelihood function and plot:

```r
set.seed(20180909)

loglik <- function(theta,x) sum(-log(pi) - log(1 + (x - theta)^2))
theta<- 5
x<- rcauchy(10, 5, 1)
plot(theta,sapply(theta,loglik,x),ylab="LogL of Cauchy")
```



3.

Iteration:

```
g <- function(theta) sum(2 * (x - theta)/(1 + (x - theta)^2))
dg <- function(theta) sum((2 * (x - theta)^2 - 2)/(1 + (x - theta)^2)^2)
newton<- function(fun, dfun, x0, eps, maxit){
  for (i in 1:maxit) {
    x1 <- x0 - fun(x0) / dfun(x0)
    if (abs(x1 - x0) < eps || abs(fun(x1)) < eps)
      return(x1)
    x0 <- x1
  }
  return(NA)
}

init <- seq(-10, 20, by = 0.5)
sapply(init, function(x) newton(g, dg, x, 1e-6, 1000))
```

```
##  [1] -3.577341e+07 -3.468845e+07 -3.360213e+07 -3.251438e+07 -3.142511e+07
##  [6] -3.033423e+07 -2.924165e+07 -2.814726e+07 -2.705095e+07 -2.595261e+07
## [11] -2.485211e+07 -2.374933e+07 -2.264414e+07 -2.153645e+07 -2.042617e+07
## [16] -3.862655e+07 -3.639562e+07 -3.416006e+07 -3.192102e+07 -2.968088e+07
## [21] -2.744455e+07 -2.522209e+07 -2.303525e+07 -2.093492e+07 -3.811394e+07
## [26] -3.571294e+07 -3.969638e+07  3.108044e+07            NA            NA
```

```
## [31]  5.685422e+00  5.685422e+00  5.685422e+00  5.685422e+00  2.660193e+07
## [36] -3.162770e+07            NA            NA  5.685422e+00 -2.898271e+07
## [41]            NA  5.685422e+00  2.461545e+07  3.663648e+07            NA
## [46]            NA            NA            NA            NA            NA
## [51]            NA            NA            NA            NA            NA
## [56]            NA -3.739490e+07            NA            NA            NA
## [61]            NA
```

We can see the iteration returns different results with different initial values.

4.

```
f1<- function(theta) sum(2 * (x - theta)/(1 + (x - theta)^2))+ theta
f2<- function(theta) 0.64 * sum(2 * (x - theta)/(1 + (x - theta)^2))+ theta
f3<- function(theta) 0.25 * sum(2 * (x - theta)/(1 + (x - theta)^2))+ theta

fixed<- function(fun, x0, eps, maxit){
  for (i in 1 : maxit){
    x1<- fun(x0)
    if (abs(x1-x0)< eps)
      return(x1)
    x0 <- x1
  }
  return(NA)
}

init <- seq(-10, 20, by = 0.5)
sapply(init, function(x) fixed(f1, x, 1e-6, 1000))
```

```
##  [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
## [24] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
## [47] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
```

```
init <- seq(-10, 20, by = 0.5)
sapply(init, function(x) fixed(f2, x, 1e-6, 1000))
```

```
##  [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
## [24] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
## [47] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
```

```
init <- seq(-10, 20, by = 0.5)
sapply(init, function(x) fixed(f3, x, 1e-6, 1000))
```

```
##  [1] 5.685421 5.685422 5.685422 5.685421 5.685421 5.685422 5.685421
##  [8] 5.685421 5.685422 5.685421 5.685422 5.685421 5.685422 5.685422
## [15] 5.685422 5.685422 5.685422 5.685422 5.685421 5.685422 5.685421
## [22] 5.685422 5.685422 5.685422 5.685422 5.685422 5.685421 5.685422
## [29] 5.685421 5.685422 5.685422 5.685422 5.685422 5.685421 5.685422
## [36] 5.685422 5.685421 5.685422 5.685422 5.685421 5.685422 5.685422
## [43] 5.685422 5.685422 5.685421 5.685422 5.685422 5.685421 5.685421
## [50] 5.685421 5.685421 5.685421 5.685422 5.685422 5.685421 5.685422
## [57] 5.685422 5.685422 5.685421 5.685422 5.685422
```

When α is set as 1 and 0.64, the iteration cannot converge. When $\alpha = 0.25$, the iteration can return two very close results (5.685421 and 5.68542) with different initial values.

5.

```r
g <- function(theta) sum(2 * (x - theta)/(1 + (x - theta)^2))

newtons<- function(fun, x0, eps, maxit){
  for (i in 1 : maxit){
    x1<- x0 + 2 * fun(x0)/ length(x)
    if (abs(x1 - x0) < eps || abs(fun(x1)) < eps)
      return(x1)
    x0 <- x1
  }
  return(NA)
}

init <- seq(-10, 20, by = 0.5)
sapply(init, function(x) newtons(g, x, 1e-6, 1000))
```

```
##  [1] 5.685422 5.685422 5.685422 5.685422 5.685422 5.685422 5.685422
##  [8] 5.685422 5.685422 5.685422 5.685422 5.685422 5.685422 5.685422
## [15] 5.685422 5.685422 5.685422 5.685422 5.685422 5.685422 5.685422
## [22] 5.685422 5.685422 5.685421 5.685422 5.685422 5.685422 5.685422
## [29] 5.685422 5.685422 5.685422 5.685422 5.685422 5.685422 5.685422
## [36] 5.685422 5.685422 5.685422 5.685422 5.685422 5.685422 5.685422
## [43] 5.685422 5.685422 5.685422 5.685422 5.685422 5.685422 5.685422
## [50] 5.685422 5.685422 5.685422 5.685422 5.685422 5.685422 5.685422
## [57] 5.685422 5.685422 5.685421 5.685422 5.685422
```

Iteration returns same result (5.68542) with different initial values except only one 5.68541 which is very close to the others.

6.

For such special distribution as Cauchy, Newton–Raphson method shows variety of estimate of MLE. With some initial values, this method does not converge. With some initial values, this method returns very different estimate of MLE.

When using fixed point mehtod, scaling choices of α is important. Some α cannot return converged result. For the α's which can return convergence, they give us very similar estimate of MLE with different initial values.

When using Fisher scoring and refining the estimate by running Newton-Raphson method, this method returns the same result with different initial values.

The speed of these three iterations are similar. I think Newton-Raphson with Fisher scoring is the most stable one among these three method.