homework 3

Exercise 3.2

Given

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

The Likelihood function becomes

$$L(\theta) = \Pi_{i=1}^{n} f(x;\theta)$$

and the log-likelihood function becomes

$$\ell(\theta) = logL(\theta)$$

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

Using properties of logarithms to separate $\sum \pi$ which results in $-n \ln \pi$
and get $-\sum_{i=1}^{n} \ln[1+(\theta-X_i)^2]$ for the second term

Using derrivative properties of logarithms, we find $\ell'(\theta)$ by taking the derrivative of the each logarithm and dividing it by the dividend ($\frac{u'}{u}$)

$$\Rightarrow \ell'(\theta) = -n*(0/\pi) - 2\sum_{i=1}^{n} \frac{\theta-X_i}{1+(\theta-X_i)^2} = -2\sum_{i=1}^{n} \frac{\theta-X_i}{1+(\theta-X_i)^2}$$

We solve for $\ell''(\theta)$ by using quotient rule

$$\frac{(1)*(1+(\theta-X_i)^2)-(\theta-X_i)*2(\theta-X_i)}{(1+(\theta-X_i)^2)^2}$$

Reduce the numerator:

$$= 1+(\theta-X_i)^2 - 2(\theta-X_i)^2 = 1-(\theta-X_i)^2$$

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

$$\Rightarrow I_n(\theta) = -E_X[\ell''(\theta)]$$

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

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

$$= 2n*E_X[\frac{1-(\theta-X_i)^2}{(1+(\theta-X_i)^2)^2}]$$

$$= 2n\int_{-\infty}^{\infty} \frac{1-(x-\theta)^2}{(1+(x-\theta)^2)^2} * \frac{1}{\pi(1+(x-\theta)^2)} dx$$

$$= \frac{2n}{\pi}\int_{-\infty}^{\infty} \frac{1-x^2}{(1+x^2)^3} dx$$

```
Fisher.intv <- integrate(Fisher.int, -Inf, Inf)

Fisher.int <- function(x) {
   (1-x^2)/ ((1+x^2)^3)
}

pi/Fisher.intv$value
[1] 4
```

$$\Rightarrow I_n(\theta) = \frac{2n}{\pi} * \frac{\pi}{4} = n/2$$

```
set.seed(20180909)
rand.sample <- rcauchy(n=10, 5)

llk <- function(x, rand.sample){
  llk <- 0
  for (i in 1:length(rand.sample)) {
    llk <- llk - log(pi) - log(1 + (x - rand.sample[i])^2)
  }
  llk
}

ggplot(data.frame(x = c(0, 10)), aes(x=x)) +
stat_function(fun = function(x) llk(x, rand.sample)) +
xlab("theta") + ylab("llk")
```

Find MLE using Newton-Raphson

```
llk.prime <- function(x){
  llk.prime <- 0
  for (i in 1:length(rand.sample)) {
  llk.prime <- llk.prime - 2 * (x - rand.sample[i]) / (1 + (x - rand.sample[i])^2)
  }
  llk.prime
}

llk.prime2 <- function(x){
  llk.prime2 <- 0
  for (i in 1:length(rand.sample)) {
    llk.prime2 <- llk.prime2 - 2* (1 - (x - rand.sample[i])^2) / (1 + (x - rand.sample[i])^2)^2
  }
  llk.prime2
}



newton <- function(der.llk, der2.llk, tol = 1e-7, x0, n = 100){
  x <- x0
  for (i in 1: n) {
  x1 <- x - (der.llk(x) / der2.llk(x))
  if (abs(x1 - x) < tol) break
  x <- x1
```

```
  }
  if (i == n)
  return(c(x0 = i, root = x1))
  plot(x0 = i, root = x1)
}

x0 <- seq(-10, 20, 0.5)
newton(llk.prime, llk.prime2,x0 = seq(-10, 20, 0.5) )
plot(newton(llk.prime, llk.prime2,x0 = seq(-10, 20, 0.5) ), xlab = "x_i", ylab = "newton-Raphson Root")
```

As we can see, the Newton-Raphson roots increase in positive correlation with the starting values, then level-out at approx. x[i] = 5.

Fixed Point Iterations

```
set.seed(20180909)
rand.sample <- rcauchy(10, 5)

llk.prime <- function(x){
  llk.prime <- 0
  for (i in 1:length(rand.sample)) {
  llk.prime <- llk.prime - 2 * (x - rand.sample[i]) / (1 + (x - rand.sample[i])^2)
  }
  llk.prime
}

fptiter <- function(der.llk, x0, alpha, n = 100, tol = 1e-7){
  x <- x0
  for (i in 1:100){
  x1 <- alpha * llk.prime(x) + x
  if (abs(x1 - x) < tol )  break
  x <- x1
  }
  return(data.frame(root = x1, x0 = i))
}
```

```
fptiter(llk.prime, x0 = seq(-10, 20, 0.5), alpha = 1)     # alpha = 1
fptiter(llk.prime, x0 = seq(-10, 20, 0.5), alpha =.64)    # alpha = 0.64
fptiter(llk.prime, x0 = seq(-10, 20, 0.5), alpha =.25)    # alpha = 0.25
```

In <- 5 Fishersc <- function(x, fun, In){ x0 <- x for (i in 1:100){ x1 <- x0 + llk.prime / In if(abs(x1-x0)<1e-7) break x0 <- x1 } return(x1) }

Newton <- function(x, llk.p, llk.p2) { x0 <- x for (i in 1:100){ x1 <- x0 - llk.prime(x) / llk.prime2(x) if(abs(x1-x0) < 1e-7) break x0 <- x1 } return(root = x1) }

Comments

Newton-Raphson is a more accurate method than Fisher, but Fisher refined by Newton-Raphson is shown to be the most stable and effecive method.