Optimization HW4

Yichu Li.

9/25/2018

Abstract

This project is about using various optimization techniques, such as Newton-Raphson, Fisher's Scoring, Fixed point method in trying to maximize likelihood of Cauchy distribution functions. Also needs to compare speed and stability of these techniques. After this, try using the technique above to solve some more practical problems like finding the local maxima and population modeling application.

3.3.2

Many local maxima

From the given density function, we have the log-likelihood function as:

$$l(\theta) = \sum_{i=1}^{19} ln(p(x;\theta)) = l(\theta) = \sum_{i=1}^{19} ln(\frac{1-cos(x-\theta)}{2\pi})$$

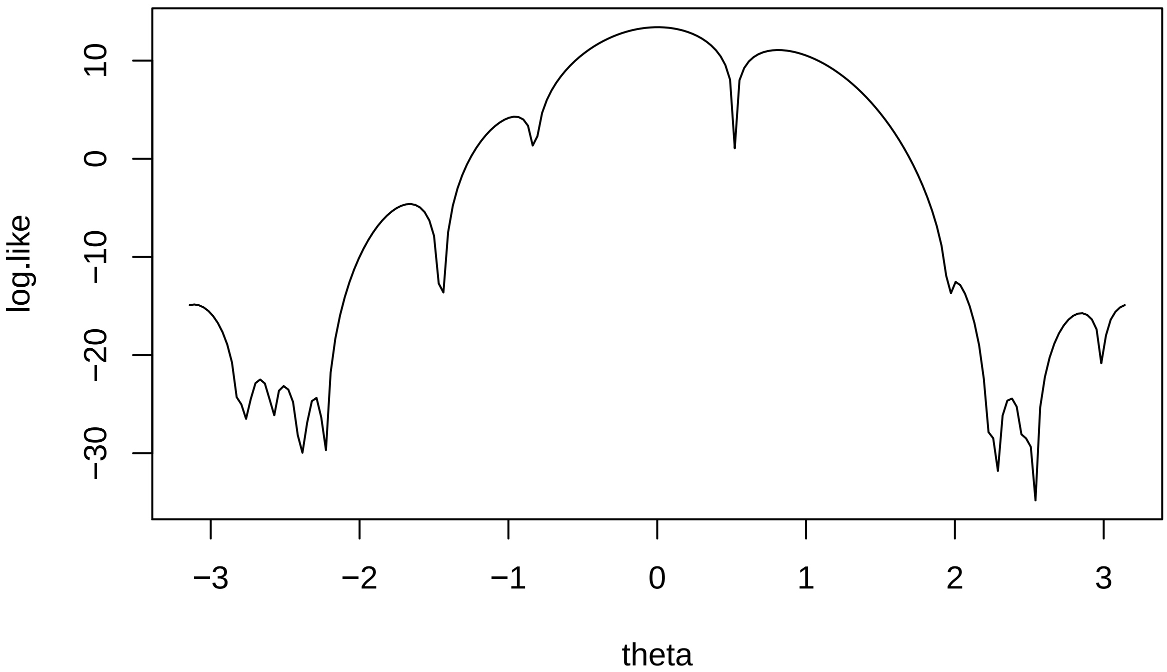
Loglikelihood function and plot

Log-likelihood function are given as follows:

$$l(\theta) = -nln\pi - \sum_{i=1}^{n} ln[1+(x_i-\theta)^2]$$

```
x <- c(3.91, 4.85, 2.28, 4.06, 3.70, 4.04, 5.46, 3.53, 2.28, 1.96,
       2.53, 3.88, 2.22, 3.47, 4.82, 2.46, 2.99, 2.54, 0.52)

theta <- seq(from = -pi, to = pi, by = 2*pi/199)
i <- 1
log.like <- rep(0,200)
for (i in 1:200) {
  log.like[i] <- sum(log((1-cos(x-theta[i]))/2*pi))
}
plot(theta,log.like,type = "l")
```



Method-of-moments estimater of theta

$$E[x|\theta] = \int_0^{2\pi} x\frac{1-cos(x-\theta)}{2\pi} = E[x|\theta] = \frac{1}{2\pi}[\int_0^{2\pi} xdx - \int_0^{2\pi} xcos(x-\theta)dx)]$$

$$E[x|\theta] = \pi - \frac{1}{2\pi}(xsin(x-\theta)+cos(x-\theta))|_0^{2\pi} = E[x|\theta_{moment}] = \pi + sin(\theta_{moment})$$

$$\theta_{moment} = arcsin(E[x|\theta]-\pi)$$

```
mean(x)
```

```
## [1] 3.236842
```

```
f1<-function(theta){
  pi+sin(theta)-mean(x)
}

uniroot(f1,lower = -pi,upper = -pi/2,extendInt = "yes")$root[1]
```

```
## [1] -3.236988
```

```
uniroot(f1,lower = -pi/2,upper = 0,extendInt = "yes")$root[1]
```

```
## [1] -3.236988
```

```
uniroot(f1,lower = 0,upper = pi/2,extendInt = "yes")$root[1]
```

```
## [1] 0.09539388
```

```
uniroot(f1,lower = pi/2,upper = pi,extendInt = "yes")$root[1]
```

```
## [1] 3.046199
```

```
theta.mom <- c(-3.236988, 0.09539388, 3.046199)
```

Using uniroot function to solve the root for function $\pi + sin(\theta_{moment}) - E[x|\theta] = 0$, and there are three solutions been found. 3.046199, -3.236988 and 0.09539388

Newton-Raphson method

```
library(elliptic)
x <- c(3.91, 4.85, 2.28, 4.06, 3.70, 4.04, 5.46, 3.53, 2.28, 1.96,
       2.53, 3.88, 2.22, 3.47, 4.82, 2.46, 2.99, 2.54, 0.52)

f <- function(theta){
  f <- 0
  i <- 1
  for (i in 1:19){
  f <- f + sin(theta-x[i])/(1-cos(theta-x[i]))
  }
  return(f)
}


fdash <- function(theta){
  f.d <- sum( -1 / (1-cos(theta-x)))
  return(f.d)
}

theta.mom <- c(-3.236988, 0.09539388, 3.046199)

newton_raphson( -3.236988, f, fdash, maxiter = 1000)
```

```
## $root
## [1] -3.112471
##
## $f.root
## [1] 1.904032e-14
##
## $iter
## [1] 7
```

```
newton_raphson( 0.09539388, f, fdash, maxiter = 1000)
```

```
## $root
## [1] 0.003118157
##
## $f.root
## [1] 0
##
## $iter
## [1] 6
```

```
newton_raphson( 3.046199, f, fdash, maxiter = 1000)
```

```
## $root
## [1] 3.170715
##
## $f.root
```

```
## [1] -1.720846e-15
##
## $iter
## [1] 7
```

Staring value is 2.7 and -2.7

```r
newton_raphson(-2.7, f, fdash, maxiter = 100)
```

```
## $root
## [1] -2.668857
##
## $f.root
## [1] -2.88259e-13
##
## $iter
## [1] 5
```

```r
newton_raphson(2.7, f, fdash, maxiter = 100)
```

```
## $root
## [1] 2.848415
##
## $f.root
## [1] -3.069767e-14
##
## $iter
## [1] 6
```

The result is -2.668857 and 2.848415.

200 equally spaced starting value between {-pi} and {pi}

```r
N_R <- function (initial, f, fdash, maxiter, give = TRUE, tol = .Machine$double.eps)
{
  old.guess <- initial
  for (i in seq_len(maxiter)) {
    new.guess <- old.guess - f(old.guess)/fdash(old.guess)
    jj <- f(new.guess)
    if (is.na(jj) | is.infinite(jj)) {
      break
    }
    if (near.match(new.guess, old.guess) | abs(jj) < tol) {
      if (give) {
        return(list(root = new.guess, f.root = jj, iter = i))
      }
      else {
        return(new.guess)
      }
    }
    old.guess <- new.guess
  }
  return(list(root = "Failed to Converge", f.root = jj, iter = i))
```

```r
}

s.p <- seq(from = -pi, to = pi, length.out = 200)


out <- data.frame(
  start.point <- s.p[1:200],
  root <- rep(0,200)
)
names(out) <- c("start.point","root")

for(i in 1:200) {
  result <- N_R(s.p[i],f ,fdash, maxiter = 1000)
  out[i,2] <- result$root
}
target <- which(out$root == "Failed to Converge")
out$root <- round(as.numeric(out$root), digits = 10)
out$root[target] <- c("Failed to Converge")
out$root <- as.factor(out$root)

i <- 1
for (i in 1:length(levels(out$root))){
  subgrp <- data.frame(
    start.point <- rep(0,length(which(out$root == levels(out$root)[i]))),
    root <- rep(0,length(which(out$root == levels(out$root)[i])))
  )
  names(subgrp) <- c("start.point","root")
  subgrp$start.point <- out[which(out$root == levels(out$root)[i]),1]
  subgrp$root <- out[which(out$root == levels(out$root)[i]),2]
  assign(paste0("root.",i), subgrp)
}
```

```r
kable(root.9, caption = " Result type 1")
kable(root.8, caption = " Result type 2")
kable(root.7, caption = " Result type 3")
kable(root.6, caption = " Result type 4")
kable(root.5, caption = " Result type 5")
kable(root.4, caption = " Result type 6")
kable(root.3, caption = " Result type 7")
kable(root.2, caption = " Result type 8")
kable(root.1, caption = " Result type 9")
kable(root.10, caption = " Result type 10")
kable(root.11, caption = " Result type 11")
kable(root.12, caption = " Result type 12")
kable(root.13, caption = " Result type 13")
kable(root.14, caption = " Result type 14")
kable(root.15, caption = " Result type 15")
kable(root.16, caption = " Result type 16")
kable(root.17, caption = " Result type 17")
kable(root.18, caption = " Result type 18")
```

min	max	root
-3.1415927	-2.8258547	-3.112470

min	max	root
-2.7942809	-2.7627071	-2.786556
-2.7311333	-2.6048381	-2.668857
-2.5732643	-2.4153954	-2.509356
-2.2891002	-2.2575264	-2.297925
-2.2259526	-2.2259526	-2.232191
-2.1943788	-1.4681815	-1.662712
-1.4366077	-1.4366077	-1.447502
-1.4050339	-0.8367056	-0.954405
-0.8051318	0.4893938	0.0031181
0.5209676	1.9417884	0.8126374
1.9733622	2.1943788	2.0072232
2.2259526	2.2575264	2.2370129
2.2891002	2.4469692	2.3747116
2.5101167	2.5101167	2.4884496
2.5416905	2.9837237	2.8484153
3.0152975	3.1415927	3.1707148

Also, there is one outcomes said Failed to Converge. From the data, if we want to find the optimum, the intial point we should use is -0.805 to 0.489.

3.3.3

Modeling beetle data

$$N_t = \frac{KN_0}{N_0 + (K - N_0)exp(-rt)}$$

$$N_0 + (K - N_0)exp(-rt) = \frac{KN_0}{N_t}$$

$$exp(-rt) = \frac{N_0(k - N_t)}{N_t(k - N_0)}$$

$$r_{es} = \frac{1}{t}log(\frac{N_t(k - N_0)}{N_0(k - N_t)}$$

```r
beetles <- data.frame(
  days = c(0, 8, 28, 41, 63, 69, 97, 117, 135, 154),
  beetles = c(2, 47, 192, 256, 768, 896, 1120, 896, 1184, 1024))

K <- 1200
r.t <- log((beetles$beetles*(K-2))/(K - beetles$beetles)*2)
r.series <- r.t / beetles$days
mean(r.series[2:10])
```

```
## [1] 0.1716788
```

```r
#r <- 0.1716788

#theta1 <- N
#theta2 <- r
```

```r
#beetles ~ theta1*2/(2+(theta1-2)exp(-1*theta2*days))
pop.mod <- nls(beetles ~ N*2/(2+(N-2)*exp((-r)*days)),start = list(N = 1200, r = 0.1716788),data = beet
```

```
## 677146.3 :  1200.0000000    0.1716788
## 110309.7 :  988.5766717   0.1367234
## 76986.61 :  1024.1437803    0.1224583
## 73503.58 :  1045.8468846    0.1189023
## 73421.83 :  1048.9806817    0.1183764
## 73419.76 :  1049.3390375    0.1182867
## 73419.7 :  1049.3958171    0.1182715
## 73419.7 :  1049.4053152    0.1182689
## 73419.7 :  1049.4069185    0.1182685
```

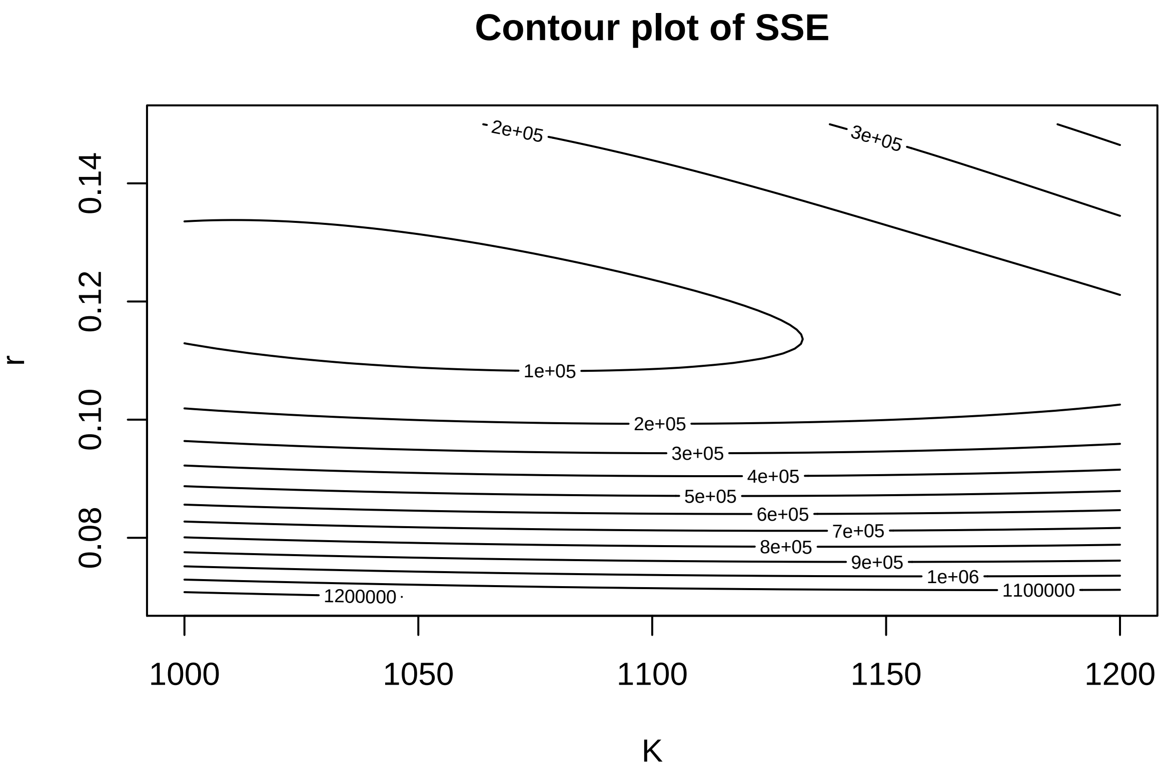
```r
K <- seq(1000,1200, by = (1200-1000)/99)
r <- seq(0.07,0.15, by = (0.15 - 0.07)/99)

n.b <- as.numeric(beetles$beetles)
t.d <- as.numeric(beetles$days)

sse <- function(K,r){
    error.sq <- sum((n.b - (K*2)/(2+(K-2)*exp(-r*t.d)))^2)
    return(error.sq)
}

z <- matrix(rep(0,10000),nrow = 100)
j <- 1 #for K
k <- 1 #for r
for (j in 1:100){
  for(k in 1:100){
    z[j,k] <- sse(K[j],r[k])
  }
}


contour(K, r, z, xlab = 'K', ylab = 'r', plot.title = title ("Contour plot of SSE"))
```



From the plot, when (K, r) approach $(1049.4069185, 0.1182685)$, the trend is decreasing.

$$(K, r, \sigma) = (820.3811422, 0.1926394, 0.6440836)$$

The variances of the related parameters are below.

$$(var(K), var(r), var(\sigma)) = (6.262790 * 10^4, 4.006745 * 10^{-3}, 2.075824 * 10^{-2})$$

```
mlogl3 <- function(theta, N, days) {
  K <- theta[1]
  r <- theta[2]
  sigma <- theta[3]
  t <- days
  mu <- log((K*2)/(2+(K-2)*exp(-r*t)))
  m <- - sum(dnorm(log(N), mu, sigma, log = TRUE))
  return(m)
}

sqrt(var(log(beetles$beetles)))
```

```
## [1] 2.031806
```

```
#2.03
theta.start <- c(1200, 0.17,2.03)
out <- nlm(mlogl3, theta.start, N = beetles$beetles,days = beetles$days,hessian = TRUE)
out
```

```
## $minimum
## [1] 9.790127
##
## $estimate
```

```
## [1] 820.3815694   0.1926394   0.6440836
##
## $gradient
## [1]  1.138072e-08  2.899903e-05 -2.716050e-06
##
## $hessian
##               [,1]         [,2]          [,3]
## [1,]  2.389705e-05   0.05442387 -3.048933e-06
## [2,]  5.442387e-02 373.52578186 -5.767458e-02
## [3,] -3.048933e-06  -0.05767458  4.817365e+01
##
## $code
## [1] 2
##
## $iterations
## [1] 38
```

```
theta.hat <- out$estimate
#K = 820.3811422 , r = 0.1926394, sigma = 0.6440836
theta.hat
```

```
## [1] 820.3815694   0.1926394   0.6440836
```

```
hes <- out$hessian
hes
```

```
##               [,1]         [,2]          [,3]
## [1,]  2.389705e-05   0.05442387 -3.048933e-06
## [2,]  5.442387e-02 373.52578186 -5.767458e-02
## [3,] -3.048933e-06  -0.05767458  4.817365e+01
```

```
var.matrix <- solve(hes)
# 6.262790e+04, 4.006745e-03, 2.075824e-02
diag(var.matrix)
```

```
## [1] 6.262790e+04 4.006745e-03 2.075824e-02
```

These results shows a larger departure from pervious K and r, the explaination could be the size of the observation set is not sufficient.