Optimization in R Markdown

Cosmin Borsa[*]

28 September 2018

Abstract

This document is a homework assignment for the course Statistical Computing at the University of Connecticut.

Keywords: Template; R Markdown; **bookdown**; **knitr**; **Pandoc**

1 Local Maxima

In the first problem we were given a probability density function with parameter θ and a sample of size n from that distribution. Using the density function $f(x;\theta)$ defined on the interval $(0, 2\pi)$, and the sample, we have to compute the log-likelihood function $l(\theta)$. The probability density function is given in (1).

$$f(x;\theta) = \frac{1 - cos(x-\theta)}{2\pi}, \qquad x \in (0, 2\pi),\, \theta \in (-\pi, \pi) \tag{1}$$

Before we compute the log-likelihood function $l(\theta)$ we are going to compute the likelihood function $L(\theta)$.

$$\begin{align} L(\theta) = f(x_1;\theta) \cdot f(x_2;\theta) \cdot \ldots \cdot f(x_n;\theta) = \prod_{i=1}^{n} \frac{1 - cos(x_i - \theta)}{2\pi} = \frac{1}{(2\pi)^n} \prod_{i=1}^{n} 1 - cos(x_i - \theta) \tag{2} \\ \tag{3} \\ \tag{4} \end{align}$$

We are now going to compute the log-likelihood function $l(\theta)$.

$$\begin{align} l(\theta) = \log(L(\theta)) = -n\log(2\pi) + \sum_{i=1}^{n} \log(1 - cos(x_i - \theta)), \qquad \theta \in (-\pi, \pi) \tag{5} \\ \tag{6} \\ \tag{7} \end{align}$$

Next, we will code the log-likelihood function $l(\theta)$ in R using the given sample.

[*] cosmin.borsa@uconn.edu; M.S. in Applied Financial Mathematics, Department of Mathematics, University of Connecticut.

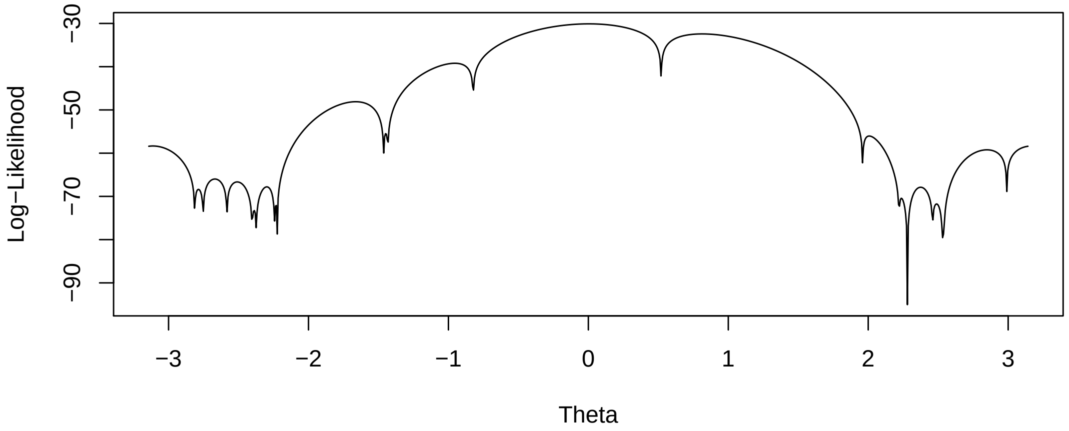

Figure 1: Log-Likelihood function of theta using the sample data

```r
sample <- 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)

loglikelihood <- function(x){
  loglikelihood <- 0
  if (-pi <= x && x <= pi) {
    loglikelihood <- -length(sample)*log(2*pi)
    for (y in sample) {
      loglikelihood <- loglikelihood + log(1 - cos(y - x))
    }
  }
  return(loglikelihood)
}
```

Next, we are going to graph the log-likelihood function $l(\theta)$ in the interval $(-\pi, \pi)$.

1.1 Method-of-Moments Estimator

In this section we will find the method-of-moments estimator of θ. However, before we do that, we need to compute the expectation of a random variable X with a given θ under the probability density function $f(x; \theta)$. Thus,

$$E(X|\theta) = \int_0^{2\pi} x \cdot \Big(\frac{1-\cos(x-\theta)}{2\pi}\Big)dx \tag{8}$$

$$\tag{9}$$

$$E(X|\theta) = \frac{1}{2\pi} \cdot \int_0^{2\pi} xdx - \frac{1}{2\pi} \cdot \int_0^{2\pi} x\cos(x-\theta)dx \tag{10}$$

$$\tag{11}$$

$$E(X|\theta) = \frac{1}{2\pi} \cdot \Big(\frac{x^2}{2}\Big|_0^{2\pi}\Big) - \frac{1}{2\pi} \cdot \Big[x\sin(x-\theta)\Big|_0^{2\pi} - \int_0^{2\pi} \sin(x-\theta)dx\Big] \tag{12}$$

$$\tag{13}$$

$$E(X|\theta) = \frac{1}{2\pi} \cdot \frac{4\pi^2}{2} - \frac{1}{2\pi} \cdot \Big[2\pi \cdot \sin(2\pi-\theta) + \cos(2\pi-\theta) - \cos(-\theta)\Big] \tag{14}$$

$$\tag{15}$$

$$E(X|\theta) = \pi + \sin(\theta) \tag{16}$$

$$\tag{17}$$

$$\tag{18}$$

To compute the method-of-moments estimator $\hat{\theta}_{MM}$, we need to set the expectation of a random variable X with the the probability density function $f(x;\theta)$ to be equal to the sample mean $\bar{X}_n$. (19)

$$E(X|\hat{\theta}_{MM}) = \bar{X}_n \tag{19}$$

$$\tag{20}$$

$$\tag{21}$$

By solving the equation (19) for $\hat{\theta}_{MM}$ we will obtain the method-of-moments estimator in (31).

$$E(X|\hat{\theta}_{MM}) = \pi + \sin(\hat{\theta}_{MM}) \tag{22}$$

$$\tag{23}$$

$$\tag{24}$$

$$\pi + \sin(\hat{\theta}_{MM}) = \bar{X}_n \tag{25}$$

$$\tag{26}$$

$$\tag{27}$$

$$\sin(\hat{\theta}_{MM}) = \bar{X}_n - \pi \tag{28}$$

$$\tag{29}$$

$$\tag{30}$$

$$
\begin{align}
\hat{\theta}_{MM} &= \arcsin(\bar{X_n} - \pi) \tag{31}\\
 \tag{32}\\
 \tag{33}
\end{align}
$$

There are two solutions for the method-of-moments estimator $\hat{\theta}_{MM}$. The first solution is $\hat{\theta}_{MM_1} \approx 0.09539407$ and the second solution is $\hat{\theta}_{MM_2} = \pi - \hat{\theta}_{MM_1} \approx 3.046199$.

1.2 Maximum likelihood Estimation

In this section we will find the maximum likelihood estimator of θ using the Newton Raphson method. The initial points of the estimation will be given by the method-of-moments estimator $\hat{\theta}_{MM}$ from the previous section. However, before we proceed with the Newton Raphson method, we have to compute the first and the second derivative of the log-likelihood function $l(\theta)$ with respect to θ in (34) and (36)

$$
\begin{align}
\frac{\partial l(\theta)}{\partial \theta} &= \sum_{i=1}^{n} \frac{\sin(\theta - x_i)}{1 - \cos(x_i - \theta)} \tag{34}\\
 \tag{35}\\
\frac{\partial^2 l(\theta)}{\partial \theta^2} &= -\sum_{i=1}^{n} \frac{1}{1 - \cos(x_i - \theta)} \tag{36}
\end{align}
$$

Next, we will implement the score function which is the first derivative of the log-likelihood function.

```
sample <- 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)
score  <- function(x){
  score <- 0
  for (y in sample) {
    score <- score + sin(x - y)/(1 - cos(y - x))
  }
  return(score)
}
```

We will also implement the derivative of the score function with respect to the parameter θ.

```
sample <- 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)

dscore <- function(x) {
  dscore <- 0
  for (y in sample) {
    dscore <- dscore - 1/(1 - cos(y - x))
  }
  return(dscore)
}
```

We will now implement the Newton Raphson method.

```
NewtonRaphson <- function (x, eps = 1e-6, itmax = 1000, verbose = TRUE){
  i <- 1
  oldguess <- x
  while(i <= itmax){
    newguess <- oldguess - (score(oldguess)/dscore(oldguess))
    if (abs(newguess - oldguess) < eps) break
    i <- i + 1
    oldguess <-  newguess
  }
  if (i < itmax + 1){
    return(newguess)
  }
  else{
    return("No Convergence")
  }
}
```

Lastly, we will find the maximum likelihood estimator for θ using the initial points $\hat{\theta}_{MM_1}$ and $\hat{\theta}_{MM_2}$.

```
sample <- 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)
MM1 <- asin(mean(sample) - pi)
MM2 <- pi - asin(mean(sample) - pi)
Newton1 <- NewtonRaphson(MM1, eps = 1e-6, itmax = 1000, verbose = TRUE)
Newton2 <- NewtonRaphson(MM2, eps = 1e-6, itmax = 1000, verbose = TRUE)
Newton1
```

```
## [1] 0.003118157
```

```
Newton2
```

```
## [1] 3.170715
```

After using the initial points $\hat{\theta}_{MM_1}$ and $\hat{\theta}_{MM_2}$ we obtain two local maximum likelihood estimates: 0.003118157 and 3.170715.

We will now run the the newton raphson algorithm with the initial points -2.7 and 2.7.

```
sample <- 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)
Newton3 <- NewtonRaphson(-2.7, eps = 1e-6, itmax = 1000, verbose = TRUE)
Newton4 <- NewtonRaphson(2.7, eps = 1e-6, itmax = 1000, verbose = TRUE)
Newton3
```

```
## [1] -2.668857
```

```
Newton4
```

```
## [1] 2.848415
```

The algorithm finds two local maxima. When we start with the initial value -2.7, the algorithm converges to -2.668857. However, when the inital value is 2.7, the newton raphson algorithm goes to 2.848415. These resiults come in line with our mental expectations, since the log-likelihood function has many local maxima. Thus, the initial point is crucial in obtaining a local maximum.

1.3 Clustering

In this section we will repeat the maximum likelihood estimation for θ using 200 equally spaced starting values in the interval $(-\pi, \pi)$. The distribution has many maxima; therefore, we will partition the values into clusters so that each cluster corresponds to a unique outcome of the optimization. We observe that there are 18 outcomes.

```r
clustering <- data.frame(initialpoints = seq(-pi, pi, length.out = 200))
for (i in 1:nrow(clustering)) {
  NewtonCluster = NewtonRaphson(clustering[i,1], eps = 1e-6, itmax = 1000, verbose = TRUE)
  if (NewtonCluster == "No Convergence") {
    clustering[i, 2] = "No Convergence"
  }
  else {
    clustering[i, 2] = NewtonCluster
  }
}
```

In the table 1 we display the 18 outcomes for θ. Morover, we show the range of values of the initial points that return those outcomes. The numbers that lie between the values of the columns *Minimum Initial Point* and *Maximum Initial Point* provide the bounds for the interval of initial points that returns the specified outcome.

```r
uniqueoutputs <- unique(strtrim(clustering[,2],9))
finalcluster <- data.frame(uniqueoutputs)

for ( i in 1:length(uniqueoutputs)) {
    indices <- which(strtrim(clustering[,2], 9) == uniqueoutputs[i])
    inputsgroup <- clustering[indices,1]
    finalcluster[i,2] = min(inputsgroup)
    finalcluster[i,3] = max(inputsgroup)
  eval(parse(text = paste0("group",i," <- cbind(inputsgroup, rep(uniqueoutputs[",i,"], length(
}

colnames(finalcluster) <- c("Outcome", "Minimum Initial Point", "Maximum Initial  Point")

knitr::kable(finalcluster, booktabs = TRUE, caption = "Clustering of Initial Points using 18 u
```

From the table 1 we can see that there are 18 groups with different local maxima. These results come in line with the likelihood function graphed in Figure 1. To get the global maximum, which is close to 0, we should use initial points which are roughly inside the interval $(-0.8, 0.5)$.

2 Population Growth Model

In this section we are going to model the population growth of beetles. The population growth is modeled using the differential equation (37)

Table 1: Clustering of Initial Points using 18 unique outcomes

Outcome	Minimum Initial Point	Maximum Initial Point
-3.112470	-3.1415927	-2.8258547
-2.786556	-2.7942809	-2.7627071
-2.668857	-2.7311333	-2.6048381
-2.509356	-2.5732643	-2.4153954
-2.388266	-2.3838216	-2.3838216
-2.297925	-2.3522478	-2.2575264
-2.232191	-2.2259526	-2.2259526
-1.662712	-2.1943788	-1.4681815
-1.447502	-1.4366077	-1.4366077
-0.954405	-1.4050339	-0.8367056
0.0031181	-0.8051318	0.4893938
0.8126374	0.5209676	1.9417884
2.0072232	1.9733622	2.1943788
2.2370129	2.2259526	2.2575264
2.3747116	2.2891002	2.4469692
2.4884496	2.4785429	2.5101167
2.8484153	2.5416905	2.9837237
3.1707148	3.0152975	3.1415927

$$\frac{dN}{dt} = rN\Big(1 - \frac{N}{K}\Big) \tag{37}$$

$$\tag{38}$$

$$\tag{39}$$

where N is the population size, t is the time, r is the growth rate parameter, and K is the parameter that represents the population carrying capacity of the environment. The solution of the differential equation is given in (40).

$$N_t = f(t) = \frac{KN_0}{N_0 + (K - N_0)\exp(-rt)} \tag{40}$$

$$\tag{41}$$

N_t denotes the population size at time t.

2.1 Fitting the Population Growth Model

We will now fit the population growth model to the beetles data. To minimize the sum of squared errors between the model predictions and the observed counts, we will use the Gauss-Newton approach. However, before we do that, we would like to obtain a forumla for the growth rate parameter r. (57)

$$N_t = \frac{KN_0}{N_0 + (K - N_0)\exp(-rt)} \tag{42}$$

$$\tag{43}$$

$$\tag{44}$$

$$N_0 + (K - N_0)\exp(-rt) = \frac{KN_0}{N_t} \tag{45}$$

$$\tag{46}$$

$$\tag{47}$$

$$(K - N_0)\exp(-rt) = \frac{N_0(K - N_t)}{N_t} \tag{48}$$

$$\tag{49}$$

$$\tag{50}$$

$$\exp(-rt) = \frac{N_0(K - N_t)}{N_t(K - N_0)} \tag{51}$$

$$\tag{52}$$

$$\tag{53}$$

$$rt = \log\left(\frac{N_t(K - N_0)}{N_0(K - N_t)}\right) \tag{54}$$

$$\tag{55}$$

$$\tag{56}$$

$$r = \frac{1}{t}\log\left(\frac{N_t(K - N_0)}{N_0(K - N_t)}\right) \tag{57}$$

$$\tag{58}$$

$$\tag{59}$$

To apply the Gauss Newton approach, we will use the *nls* function to determine the nonlinear least-squares estimates of the parameters of the model. For the estimation, the initial point $K_0 = 1,500$ was choosen for K, and r_0 was computed from the mean of the observed growth rates. The values of the observed growth rates r were obtained using the equation (57) and the given beetle data.

```
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))

K0 = 1500
r0_v <- log(beetles$beetles*(K0 -2)/(K0 - beetles$beetles)*2)/beetles$days
r0 = mean(r0_v[2:10])

pmodel <- nls(beetles ~ K*2/(2 + (K - 2)*exp(-r*days)), start = list(K = K0, r = r0), data = be
```

```
## 1980050 :  1500.0000000    0.1652033
## 117991 :  997.5137694   0.1394446
## 78622.18 :  1019.990949    0.123586
## 73557.02 :  1044.6290612    0.1190586
## 73423.02 :  1048.865917    0.118403
## 73419.79 :  1049.3219772    0.1182912
## 73419.7 :  1049.3929896    0.1182723
## 73419.7 :  1049.4048387    0.1182691
## 73419.7 :  1049.4068372    0.1182685
```

We can see that the population carrying capacity K converges to 1049.4068372 while the growth rate parameter r goes to 0.1182685.

Next, we will plot the contour of the sum of squared errors (SSE). However, before we do that, we would like to code the sum of squared errors.

```
SSE <- function(K, r) {
  estbeetles <-  K*2/(2 + (K - 2)*exp(-r*beetles$days))
  diffbeetles <- estbeetles - beetles$beetles
  SSE <- t(diffbeetles)%*%diffbeetles
  return(SSE)
}

K_v = seq(500, 1500, length.out = 200)
r_v = seq(0.1, 0.2, length.out = 200)
SSEmatr <- matrix(nrow = length(K_v), ncol = length(r_v))

for (i in 1:length(K_v)) {
  for (j in 1:length(r_v)) {
    SSEmatr[i,j] = SSE(K_v[i],r_v[j])
  }
}
```

The contour plots are displayed in Figure 2

The minimum of the sum of squared errors of the logistic model is approximately $200,000$. This value is attained when the population carrying capacity K is roughly between 850 and 1250, and the growth rate r takes values in the interval $(0.10, 0.15)$.

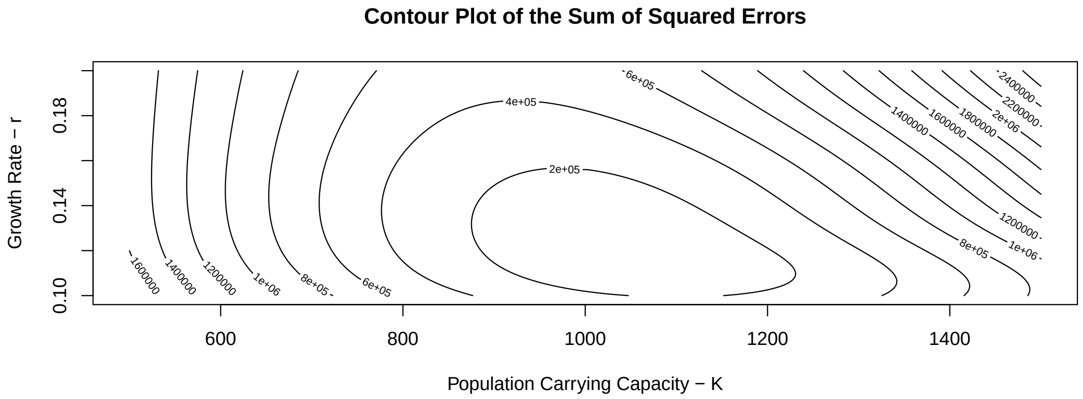

Figure 2: Contour Plot of the Sum of Squared Errors of the Logistic Model.

2.2 Log-Normal Distribution

In this section we will assume that $\log N_t$ is independent and normally distributed with mean $\log f(t)$ and variance σ^2. We will find the maximum likelihood estimator of $\theta = (r, K, \sigma^2)$ using $K_0 = 1,500$, and we will compute r_0 from the mean of the observed growth rates.

```
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))

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

thetainitial = numeric(3)
thetainitial[1] <- 1500
r0_v <- log(beetles$beetles*(K0 - 2)/(K0 - beetles$beetles)*2)/beetles$days
r0 = mean(r0_v[2:10])
thetainitial[2] <- r0
thetainitial[3] <- sqrt(var(log(beetles$beetles)))

mle <- nlm(aloglikelihood, thetainitial, hessian = TRUE)
mle$estimate
```

```
## [1] 820.3814175   0.1926394   0.6440835
```

```
hessianmatr <- mle$hessian
hessianmatr
```

```
##               [,1]         [,2]          [,3]
## [1,]  2.389708e-05   0.05442391 -3.029013e-06
## [2,]  5.442391e-02 373.52687023 -5.759091e-02
## [3,] -3.029013e-06  -0.05759091  4.817369e+01
```

```
estvariance <- solve(hessianmatr)
estvariance
```

```
##               [,1]          [,2]          [,3]
## [1,]  6.262775e+04 -9.125039e+00 -6.971005e-03
## [2,] -9.125039e+00  4.006728e-03  4.216227e-06
## [3,] -6.971005e-03  4.216227e-06  2.075822e-02
```

The maximum likelihood estimates for the population carrying capacity K, the growth rate r, and the standard deviation σ are 820.3811453, 0.1926394, and 0.6440837. The values that we have obtained for K and r differ from the values that we got in our previous estimates since we have taken the variance into consideration. The variance of the estimators can be obtained from the hessian matrix. Thus, we get $6.262775e + 04$ for variance of the population carrying capacity K, $4.006728e - 03$ for the variance of the growth rate r, and lastly, $2.075825e - 02$ for the variance of the standard deviation σ.