MLE Project

Yiyi Xu

9/21/2018

3.3.2 Many Local Maxima

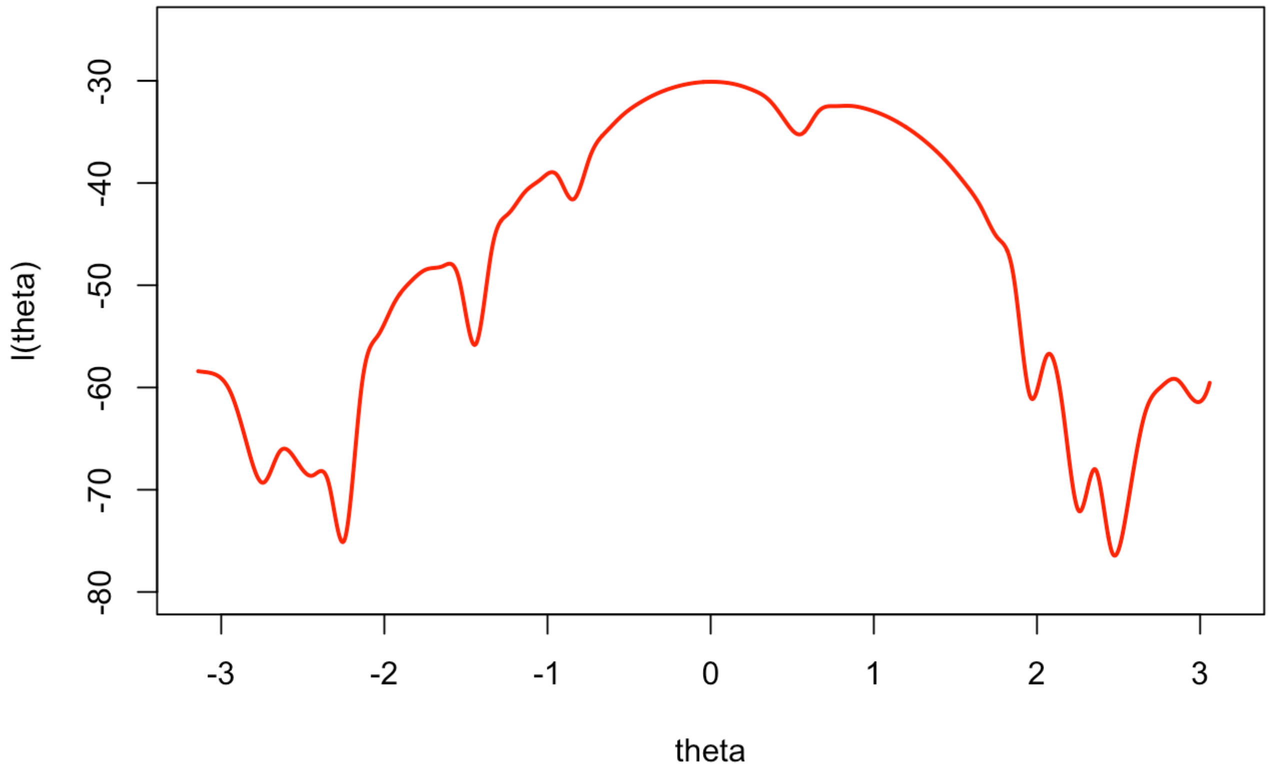
Find the log-likelihood and plot

Equation:

$$L(\theta; x) = \prod_{i=1}^{n} f(x_i; \theta) = (2\pi)^{-n} \prod_{i=1}^{n} [1 - cos(x_i - \theta)]$$

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

```
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)
l <- function(x, y){
  -19*log(2*pi)+sum(log(1-(cos(x-y))))
}
theta <- seq(-pi, pi, 0.1)
L <- numeric(0)
for (i in 1: length(theta)){
   L[i] <- l(sample,theta[i])
}
 sp=spline(theta,L,n=1000)
 plot(sp,col="red",type="l",xlim=c(-pi,pi),ylim=c(-80,-25),lwd=2,xlab="theta",ylab="l(th
eta)",sub="loglikehood function against theta",col.main="green",font.main=2)
```



loglikehood function against theta

Find Method of moments estimator of theta

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

$$\tilde{\theta}_n = \arcsin(\bar{X}_n - \pi)$$

Find the MLE by Newton-Raphson method

$$l'(\theta) = \sum_{i=1}^{n} \frac{-sin(x_i - \theta)}{1 - cos(x_i - \theta)}$$

$$l''(\theta) = \sum_{i=1}^{n} \frac{1}{cos(x_i - \theta) - 1}$$

```
l1 <- function(sample, theta){
  sum((-sin(sample-theta))/(1 -cos(sample-theta)))
}
l2 <- function(sample,theta){
  sum((1)/(cos(sample-theta)-1))
}
Newton.Method <- function(y,f,f1){
   y0 <- y
   for(i in 1:100){
    y1 <- y0 - f(sample,y0)/f1(sample,y0)
    if(abs(y1-y0)<0.0001)
      break
    y0 <- y1
   }
   return(data.frame(init=y,root=y0,iter=i))
}

Newton.Method(pi - asin(mean(sample) - pi),l1,l2)
```

```
##       init     root iter
## 1 3.046199 3.170713    5
```

```
Newton.Method(asin(mean(sample) - pi),l1,l2)
```

```
##         init        root iter
## 1 0.09539407 0.003136419    3
```

solutions

```
Newton.Method(-2.7,l1,l2)
```

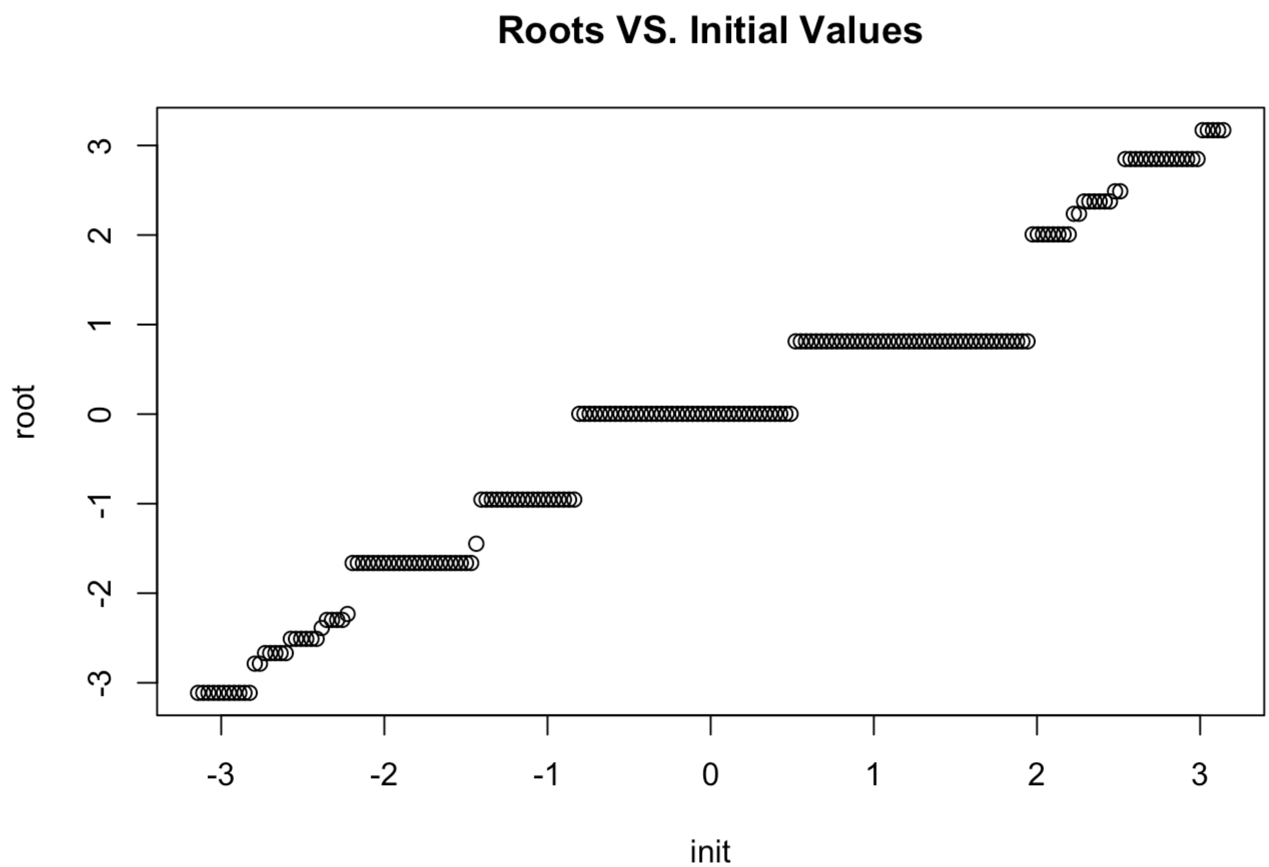
```
##   init      root iter
## 1 -2.7 -2.668857    4
```

```
Newton.Method(2.7,l1,l2)
```

```
##   init     root iter
## 1  2.7 2.848423    4
```

repoeat 200 times

```
init <- seq(-pi,pi,length=200)
root <- numeric(length(init))
for(i in 1:length(init)){
  root[i] <- Newton.Method(init[i],l1,l2)[1,2]
}
plot(init,root,main = "Roots VS. Initial Values")
```



3.3.3 Modeling beetle data

Fit the population growth model to the beetles data using the Gauss-Newton approach, to minimize the sum of squared errors between model predictions and observed counts. From data frame, assues initial N is 2.

$$F = \sum_{i=1}^{n}[N_i - \frac{2K}{2 + (K-2)e^{-rt_i}}]^2$$

```
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))
N0 <- 2   ## initial value by beetles.

func <- function(x){
   beetles$beetles-(2*x[1])/(2+(x[1]-2)*exp(-x[2]*beetles$days))
}

library(pracma)
gaussNewton(c(100,1),func)
```

```
## $xs
## [1] 1049.4072453    0.1182684
##
## $fs
## [1] 73419.7
##
## $niter
## [1] 10
##
## $relerr
## [1] 1.455192e-11
```

Show the contour plot of the sum of squared errors

```
func1 <- function(a,b){
   sum((beetles$beetles-(2*a)/(2+(a-2)*exp(-b*beetles$days)))^2)
}

library(plotly)
```

```
## Loading required package: ggplot2
```

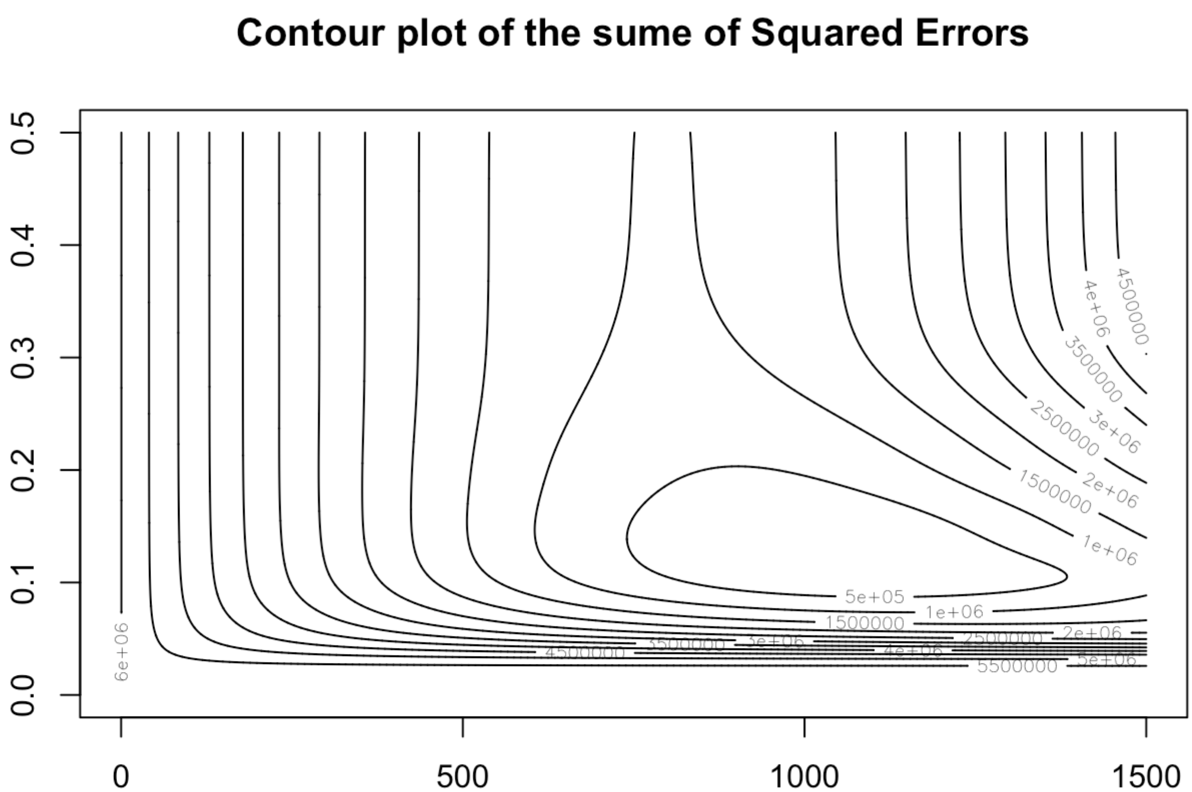
```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
k <- seq(0,1500,0.1)
r <- seq(0,0.5,0.001)
y <-outer(k,r,Vectorize(func1))
contour(k,r,y,main="Contour plot of the sume of Squared Errors")
```



Find the maximum likelihood estimators & Estimate the variance your parameter estimates.

```r
lnday <- log(beetles$beetles)
# by previous result, set:
k1 <- 1049.4072453
r1 <- 0.1182684

lnfunc <- function(x){
  k <- x[1]
  r <- x[2]
  sigma <- x[3]
  -sum(-(log(2 * pi * (sigma)) / 2 )- (log(beetles$beetles) - log((2*k)/(2 + (k - 2) * e
xp(-r * beetles$days)))) ^ 2 / (2 * (sigma)))
}

x <- c(k1,r1,1)
lnfunc(x)
```

```
## [1] 12.53528
```

```r
MLEfunc <- optim(x, lnfunc, method = "BFGS",hessian = TRUE)
```

```
## Warning in log(2 * pi * (sigma)): NaNs produced

## Warning in log(2 * pi * (sigma)): NaNs produced
```

```r
MLEfunc
```

```
## $par
## [1] 1038.8791759    0.1719385    0.4386654
##
## $value
## [1] 10.09091
##
## $counts
## function gradient
##       80       37
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##               [,1]         [,2]         [,3]
## [1,]  1.091616e-05   0.05120572 -0.005480509
## [2,]  5.120572e-02 795.90868517  0.012541935
## [3,] -5.480509e-03   0.01254193 26.209272440
```