

## Inference Tutorial 5

This tutorial is mostly about practical maximization of likelihoods in R. The second question leads into the third question.

1. A random variable  $X$  has probability density function:

$$f(x) = \begin{cases} (b-a)^{-1} & a \leq x \leq b \\ 0 & \text{otherwise} \end{cases}$$

Given observations of  $X$ ,  $x_1, x_2, \dots, x_n$ , find the maximum likelihood estimates of  $a$  and  $b$ . Are the corresponding estimators unbiased? Why can large sample results not be used to find the approximate distribution of the MLEs in this case (you may need to look ahead in the notes for this)?

### Solution

$$L(a, b) = \begin{cases} (b-a)^{-n} & a \leq x_i \leq b \forall i \\ 0 & \text{otherwise} \end{cases}$$

$L(a, b)$  is maximized when  $b - a$  is as small as possible subject to the restriction that  $a \leq x_i \leq b \forall i$ . i.e.  $\hat{a} = \min(x_i)$ ,  $\hat{b} = \max(x_i)$ .

Clearly  $\hat{a} > a$ , almost surely and  $\hat{b} < b$  almost surely, so both estimators are biased.

The derivatives of  $L$  and  $\log L$  are not properly defined at the MLEs (as the likelihood is not continuous at the MLE), so the Taylor expansion used to justify the large sample results is not applicable.

2. R contains functions for optimization which we can use to maximize log likelihoods (equivalently minimize negative log likelihoods). Two such are `nlm` and `optim`, the former offering Newton's method and the latter several alternatives, including quasi-Newton. To use either, you have to write an R function to evaluate the model log likelihood, taking the model parameter vector as first argument, and anything else needed to evaluate the likelihood as further arguments (you pass these extra arguments to `optim` and `nlm` as named arguments which the optimizers simply pass on to the log likelihood function). Here is a simple example, using the example from section 6.1 of the notes. Run the code, and work through it carefully to make sure you understand how it works, so that you would be able to do the same thing with a different model.

```
## simulate the section 6.1 data...
set.seed(10)
x <- sort(runif(50))
y <- rpois(50, exp(2.5*x))

## Produce function to evaluate (negative) log likelihood of model
## y ~ Poi(exp(beta*x))...

ll <- function(beta, x, y) {
  mu <- exp(beta*x)
  loglik <- sum(dpois(y, mu, log=TRUE))
  -loglik
} ## ll

## fit using Newton's method (approximates derivatives numerically)
beta0 <- 0 ## initial parameter values
nfit <- nlm(ll, beta0, x=x, y=y)
nfit
plot(x, y); lines(x, exp(x*nfit$estimate))

## fit using Quasi-Newton method (numerical derivatives)
qfit <- optim(beta0, ll, method="BFGS", x=x, y=y)
qfit
plot(x, y); lines(x, exp(x*qfit$par))
```

### Solution

The plots should both be the same and like the left panel in figure 2 of section 6.1.

3. *Echinus affinis* is a species of deep sea urchin. In a 1985 paper Gage and Tyler reported data on the growth of *E. affinis* collected from the Rockall Trough. The data are in file `urchin.vol.dat` on the course website can be read into R using `uv <- read.table("https://people.maths.bris.ac.uk/~sw15190/TOI/urchin.vol.dat")`. The data consist of volume measurements, `vol`, and age for 142 individuals (the urchins have annual growth rings).

- (a) Produce a sensible plot of the data, in order to get a feel for what they look like. (and google ‘sea urchin pictures’ is you don’t know what sea urchins look like).
- (b) Nisbet and Gurney (1998) suggested simple energy budget based arguments to arrive at a model for average volume as a function of age which is

$$\frac{dV}{da} = \begin{cases} \gamma V & V < \phi/\gamma \\ \phi & \text{otherwise} \end{cases}$$

Where  $\gamma$  and  $\phi$  are parameters. The initial volume is  $\omega$ , also a model parameter. Growth is in two phases — in the first the animal grows as fast as it can, given the food it can obtain, and in the second it grows less quickly, putting the surplus food energy into reproduction. The age at onset of reproduction can be solved for:  $a_m = \log\{\phi/(\gamma\omega)\}/\gamma$ , and the model can then be solved analytically:

$$V(a) = \begin{cases} \omega \exp(\gamma a) & a < a_m \\ \phi/\gamma + \phi(a - a_m) & \text{otherwise} \end{cases}$$

A reasonable model for the observed volumes in the data is

$$\sqrt{\text{vobs}_i} = \sqrt{V(\text{age}_i)} + \epsilon_i \text{ where } \epsilon_i \sim N(0, \sigma^2)$$

Suppose the model parameters are combined in one vector  $\theta^T = (\omega, \phi, \gamma, \sigma)$ . Write an R function to evaluate the log likelihood of  $\theta$  given the age and volume data. That is complete the function

```
urchin.ll <- function(theta, age, vol) {
  omega <- theta[1]
  phi <- theta[2]
  gamma <- theta[3]
  am <- log(phi/(gamma*omega))/gamma
  ...
  ll ## return computed log likelihood
} ## urchin.ll
```

You should not need a loop in this — Chapter 3 of *Core Statistics* contains more on writing functions in R.

- (c) If the likelihood is to be maximized numerically, it helps if things are set up so that there is no chance of the parameters taking impossible values. In the current model it makes no sense for any of the parameters to be negative, but a simple optimization method can’t know that, and may try out a negative value, causing a problem (where?). A simple way to avoid this is to define  $\omega = \exp(\theta_1)$ ,  $\phi = \exp(\theta_2)$  etc, so that the elements of  $\theta$  can take any values, but  $\omega, \phi$  etc. will always be positive. Modify your log likelihood function to work this way. Also, change it so that it returns the value of the negative log likelihood, as the R optimizers are set up to minimize, by default.
- (d) By looking at the plotted data try to guess reasonable initial values for the parameters (they just need to give volumes somewhere in the right ballpark). Then try to maximize the likelihood using something like

```
optim(theta0, urchin.ll, method="BFGS", vol=uv$vol, age=uv$age)
```

With reference to the `?optim` helpfile and output, check whether the optimization appears to have converged.

- (e) Produce a version of your log likelihood function that returns the volumes for each age, given the model parameters, and use it to obtain the maximum likelihood estimates of the expected volumes. Produce a plot in which these expected volumes are overlaid on the plot of measured volumes against age. Keep your code from this question for future use.

### Solution

```
## a)
uv <- read.table("https://people.maths.bris.ac.uk/~sw15190/TOI/urchin.vol.dat")
```

```

with(uv,plot(age,vol))

## b) and c)
urchin.ll <- function(theta,age,vol) {
  theta <- exp(theta)
  omega <- theta[1]
  phi <- theta[2]
  gamma <- theta[3]
  sigma <- theta[4]
  am <- log(phi/(gamma*omega))/gamma ## critical age
  ind <- age < am ## before critical age?
  V <- age*0 ## create expected volume vector
  V[ind] <- omega*exp(gamma*age[ind])
  V[!ind] <- phi/gamma + phi*(age[!ind]-am)
  ll <- sum(dnorm(sqrt(vol),sqrt(V),sigma,log=TRUE)) ## log lik.
  attr(ll,"V") <- V ## return V as an attribute of ll for part e)
  -ll ## return -ve computed log likelihood
} ## urchin.ll

## d)
min(uv$vol) ## 0.1 - use as starting omega
with(uv,plot(age,vol))
## ... gamma = phi = 1 seems a reasonable starting value

theta0 <- c(log(.1),0,0,0)
fit <- optim(theta0,urchin.ll,method="BFGS",vol=uv$vol,age=uv$age)

> fit
$par
[1] -4.0021570  0.1715777 -0.2136567 -0.7521012

$value
[1] 94.69095

$count
function gradient
      58      18

$convergence
[1] 0 ## ... apparently converged

## part e)
V <- attr(urchin.ll(fit$par,uv$age,uv$vol),"V")
with(uv,plot(age,vol))
lines(uv$age,V)

```

