MTH210: Lab 7 Solutions

Likelihood functions

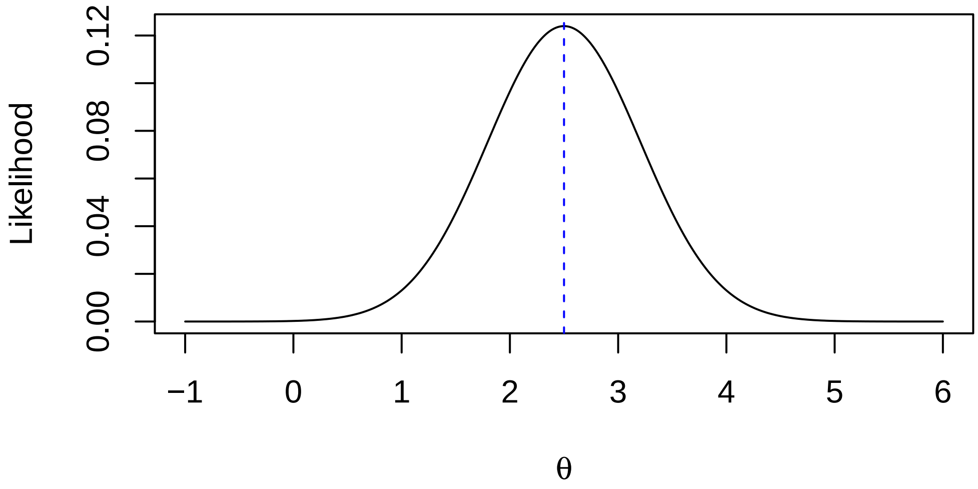
1. **Suppose $X_1, X_2 \overset{iid}{\sim} N(\theta, 1)$, and you observed $X_1 = x_1 = 2$ and $X_2 = x_2 = 3$. For various values of θ, draw the likelihood function of θ.**

   Recall that

   $$L(\theta|\mathbf{X}) = f(x_1|\theta)f(x_2|\theta)\,,$$

   where $f(x|\theta)$ is the density of a $N(\theta, 1)$ distribution evaluated at the value x. So you have to find this product and then choose a grid of values of θ on the x-axis and plot the corresponding value of $L(\theta|x_1 = 2, x_2 = 3)$.

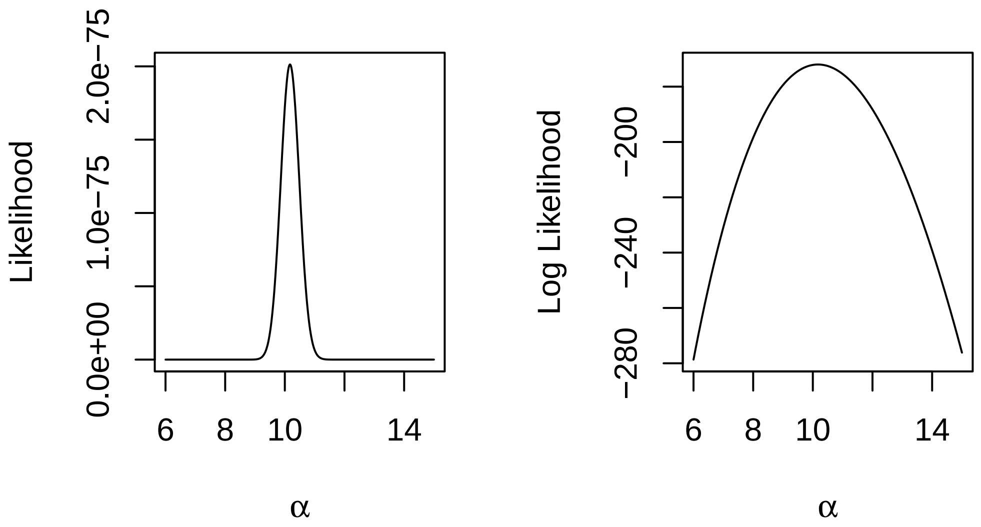
```r
x1 <- 2
x2 <- 3

like <- function(theta)
{
  dnorm(x1, mean = theta)*dnorm(x2, mean = theta)
}

theta <- seq(-1, 6, length = 5e2)
plot(theta, like(theta), type = 'l',
     xlab = expression(theta), ylab = "Likelihood")
abline(v = (x1 + x2)/2, col = "blue", lty = 2)
```



2. **Suppose you obtain $X_1, X_2, \ldots, X_{100} \overset{iid}{\sim}$ Gamma$(\alpha, 2)$, with $\alpha = 10$. For various values of α, draw the likelihood function $L(\alpha|\mathbf{X})$ and draw log of the likelihood function: $\log L(\alpha|\mathbf{X})$.**

```
alpha <- 10
dat <- rgamma(100, shape = alpha, rate = 2)
alpha.grid <- seq(6, 15, length = 5e2)

like <- function(alpha)
{
  prod(dgamma(dat, shape = alpha, rate = 2))
}
loglike <- function(alpha)
{
  sum(dgamma(dat, shape = alpha, rate = 2, log = TRUE))
}

cal.like <- numeric(length = length(alpha.grid))
cal.loglike <- numeric(length = length(alpha.grid))

## calculate log likelihood for each value of alpha
for(i in 1:length(alpha.grid))
{
  cal.like[i] <- like(alpha.grid[i])
  cal.loglike[i] <- loglike(alpha.grid[i])
}

par(mfrow = c(1,2))
plot(alpha.grid, cal.like, type = 'l',
     xlab = expression(alpha), ylab = "Likelihood")
```

```
plot(alpha.grid, cal.loglike, type = 'l',
     xlab = expression(alpha), ylab = "Log Likelihood")
```



You can see that the log-likelihood is much better behaved and on a scale that will allow for less numerical stability. This is why we will often deal with the log-likelihood function, and not the likelihood.

To check if the maximas are happening at the same point, we can do the following

```
# same value
which.max(cal.like)
```

```
[1] 232
```

```
which.max(cal.loglike)
```

```
[1] 232
```

```
# actual MLE
alpha.grid[which.max(cal.like)]
```

```
[1] 10.16633
```