MSMS 206 : Practical 04

Ananda Biswas

May 8, 2025

Question

Consider a lifetime variable that follows a **Weibull distribution** with shape parameter α and scale parameter λ. The probability density function of this distribution is given by:

$$f(t; \alpha, \lambda) = \frac{\alpha}{\lambda}\left(\frac{t}{\lambda}\right)^{\alpha-1} e^{-(t/\lambda)^{\alpha}}, \quad t > 0.$$

The objective is to evaluate the performance of **maximum likelihood estimation (MLE)** for different sample sizes. First, generate random samples of sizes $n = 60, 80, 100, 120$ and 140 from this Weibull distribution. For each sample, estimate the parameters α and λ using MLE.

Next, compute the **standard errors** of the ML estimates and evaluate their accuracy by estimating the **bias** and the **mean squared error (MSE)**.

Build-up for obtaining MLE

For a sample of size n, the likelihood function is given by

$$\begin{aligned} L(\alpha, \lambda) &= \prod_{i=1}^{n} \frac{\alpha}{\lambda}\left(\frac{x_i}{\lambda}\right)^{\alpha-1} e^{-(x_i/\lambda)^{\alpha}} \\ &= \left(\frac{\alpha}{\lambda}\right)^{n} \left(\prod_{i=1}^{n} \frac{x_i}{\lambda}\right)^{\alpha-1} \exp\left\{-\sum_{i=1}^{n}\left(\frac{x_i}{\lambda}\right)^{\alpha}\right\}. \end{aligned}$$

The log-likelihood function is given by

$$l(\alpha, \lambda) = n\log\alpha - n\log\lambda + (\alpha - 1)\sum_{i=1}^{n}\log\left(\frac{x_i}{\lambda}\right) - \sum_{i=1}^{n}\left(\frac{x_i}{\lambda}\right)^{\alpha}.$$

Now,

$$\frac{\partial}{\partial\alpha} l(\alpha, \lambda) = \frac{n}{\alpha} + \sum_{i=1}^{n}\log\left(\frac{x_i}{\lambda}\right) - \sum_{i=1}^{n}\left(\frac{x_i}{\lambda}\right)^{\alpha}\log\left(\frac{x_i}{\lambda}\right) = u(\alpha, \lambda), \text{ say} \tag{1}$$

and

$$\frac{\partial}{\partial \lambda} l(\alpha, \lambda) = -\frac{n}{\lambda} + (\alpha - 1) \sum_{i=1}^{n} \frac{\lambda}{x_i} \cdot \left(-\frac{x_i}{\lambda^2}\right) + \sum_{i=1}^{n} \frac{\alpha \cdot x_i^{\alpha}}{\lambda^{\alpha+1}}$$

$$= -\frac{n}{\lambda} - (\alpha - 1)\frac{n}{\lambda} + \sum_{i=1}^{n} \frac{\alpha \cdot x_i^{\alpha}}{\lambda^{\alpha+1}}$$

$$= -\frac{n\alpha}{\lambda} + \sum_{i=1}^{n} \frac{\alpha \cdot x_i^{\alpha}}{\lambda^{\alpha+1}} = v(\alpha, \lambda), \text{ say.}$$

Setting $v(\alpha, \lambda) = 0$ we get,

$$-\frac{n\alpha}{\lambda} + \sum_{i=1}^{n} \frac{\alpha \cdot x_i^{\alpha}}{\lambda^{\alpha+1}} = 0$$

$$\Rightarrow \frac{n}{\lambda} = \sum_{i=1}^{n} \frac{x_i^{\alpha}}{\lambda^{\alpha+1}}$$

$$\Rightarrow \frac{n}{\lambda} = \frac{1}{\lambda^{\alpha+1}} \sum_{i=1}^{n} x_i^{\alpha}$$

$$\Rightarrow \lambda^{\alpha} = \frac{1}{n} \sum_{i=1}^{n} x_i^{\alpha}$$

$$\therefore \lambda = \left(\frac{1}{n} \sum_{i=1}^{n} x_i^{\alpha}\right)^{\frac{1}{\alpha}} \tag{2}$$

Setting $u(\alpha, \lambda) = 0$ does not yield any closed form solution. So for getting the ML estimate of α, we resort to numerical methods (here Newton-Raphson method).

Now,

$$u_{\alpha}(\alpha, \lambda) = \frac{\partial}{\partial \alpha} u(\alpha, \lambda) = -\frac{n}{\alpha^2} - \sum_{i=1}^{n} \left(\frac{x_i}{\lambda}\right)^{\alpha} \left[\log\left(\frac{x_i}{\lambda}\right)\right]^2; \tag{3}$$

At each iteration, with the present value of α we calculate λ by using (2); then we use the obtained value of λ in (1) and (3) to improve the estimate of α by Newton-Raphson method.

➜ R Program

```
estimate_lambda <- function(s, alpha) mean(s^alpha)^(1/alpha)
```

```
u <- function(alpha, lambda, s){

  a <- length(s) / alpha
```

```r
  b <- sum(log(s / lambda))

  c <- sum((s / lambda)^alpha * log(s / lambda))

  return(a + b - c)
}
```

```r
u_alpha <- function(alpha, lambda, s){

  a <- - length(s) / alpha^2

  b <- sum((s / lambda)^alpha * log(s / lambda)^2)

  return(a - b)
}
```

```r
estimate_alpha <- function(s, initial, epsilon = 0.0001, iterations = 100){

  alphas <- c(initial)

  for (i in 2:iterations) {
    l <- estimate_lambda(s, alphas[i-1])

    alphas[i] <- alphas[i-1] - u(alphas[i-1], l, s) / u_alpha(alphas[i-1], l, s)

    if(abs((alphas[i] - alphas[i-1])) < epsilon) break
  }

  return(alphas[length(alphas)])
}
```

```r
true_alpha <- 3; true_lambda <- 2
```

```r
sample_sizes <- c(60, 80, 100, 120, 140)
```

```r
alpha_hat = lambda_hat = c()
```

```r
for (n in sample_sizes) {

  x <- rweibull(n, shape = true_alpha, scale = true_lambda)

  a <- estimate_alpha(x, 1)

  alpha_hat <- append(alpha_hat, a)

  lambda_hat <- append(lambda_hat, estimate_lambda(x, a))
}
```

```
df1 <- data.frame(Sample_Size = sample_sizes,
                  alpha_hat = alpha_hat,
                  lambda_hat = lambda_hat)
```

✎ Estimates of the parameters for different sample sizes are as follows :

```
df1

##   Sample_Size alpha_hat lambda_hat
## 1          60  2.859356   2.068752
## 2          80  3.227570   2.024756
## 3         100  3.100148   2.085612
## 4         120  3.113117   2.135652
## 5         140  3.123736   2.045556
```

Now we shall evaluate the accuracy of the estimates.

```
alpha_bias = lambda_bias = alpha_SE = lambda_SE = alpha_MSE = lambda_MSE = c()
```

```
for (k in 1:length(sample_sizes)) {

  alpha_estimates = lambda_estimates = c()

  for (i in 1:100){

    x <- rweibull(sample_sizes[k], shape = true_alpha, scale = true_lambda)

    alpha_estimates[i] <- estimate_alpha(x, 1)

    lambda_estimates[i] <- estimate_lambda(x, alpha_estimates[i])
  }

  alpha_bias[k] <- mean(alpha_estimates) - true_alpha

  lambda_bias[k] <- mean(lambda_estimates) - true_lambda

  alpha_SE[k] <- sd(alpha_estimates)

  lambda_SE[k] <- sd(lambda_estimates)

  alpha_MSE[k] <- mean( (alpha_estimates - true_alpha)^2 )

  lambda_MSE[k] <- mean( (lambda_estimates - true_lambda)^2 )
}
```

```
df2 <- data.frame(sample_sizes,
                  alpha_bias, alpha_SE, alpha_MSE,
                  lambda_bias, lambda_SE, lambda_MSE)
```

✎ Bias, Standard error and MSE of the estimates for different sample sizes are as follows :

```
df2

##   sample_sizes alpha_bias  alpha_SE  alpha_MSE  lambda_bias  lambda_SE
## 1           60 0.08410573 0.2974252 0.09465089 -0.004768764 0.08186733
## 2           80 0.04029086 0.2849094 0.08198498 -0.004317026 0.08148510
## 3          100 0.07690680 0.2463622 0.06600206  0.005209200 0.08189805
## 4          120 0.01963392 0.2220456 0.04919668 -0.009140331 0.06363508
## 5          140 0.03633982 0.1684986 0.02942845  0.009296586 0.05662430
##    lambda_MSE
## 1 0.006657978
## 2 0.006592059
## 3 0.006667354
## 4 0.004092475
## 5 0.003260675
```

➔ Conclusion

✎ Accuracy of the estimates increase as sample size increases.