STA 6351, Report.1.6

Carson Slater *Baylor University*

1.6

Truncated exponential sampling. Consider the truncated exponential density

$$g(x, \lambda) = \frac{1}{\lambda} \exp\left(-\frac{(x-1)}{\lambda}\right),$$

for $x > 1$ and $\lambda > 0$.

- Derive the MLE for λ in the "usual" way, by solving the score equation.
- Obtain the MLE by using Fisher scoring and the convergence criteria. The data are given below. Use an initial value of 2.8 (Of course, this is unnecessary given the closed-form MLE, but it will further illustrate the method.) 1.884484 5.306045 3.501055 3.494543 4.067554 1.920979 4.425221 3.189805 1.964026 2.612595 1.011752 4.084178 4.114090 10.168193 2.734757 1.932564 3.194626 2.973989 2.975432 1.783302

This problem asks to find the Maximum Likelihood Estimator (MLE) for λ for a truncated exponential distribution. The probability density function (PDF) is given by:

$$g(x, \lambda) = \frac{1}{\lambda} \exp\left(-\frac{(x-1)}{\lambda}\right), \quad \text{for } x > 1, \lambda > 0.$$

Assume we have an independent and identically distributed (i.i.d.) sample $X_1, X_2, \ldots, X_n$ from this distribution.

1. The Likelihood Function

The likelihood function $L(\lambda)$ for the sample $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is the product of the individual PDFs:

$$L(\lambda) = \prod_{i=1}^{n} g(x_i, \lambda) = \prod_{i=1}^{n} \left[\frac{1}{\lambda} \exp\left(-\frac{(x_i-1)}{\lambda}\right)\right]$$

$$L(\lambda) = \left(\frac{1}{\lambda}\right)^n \exp\left(-\frac{1}{\lambda}\sum_{i=1}^{n}(x_i-1)\right)$$

It is easier to maximize the log-likelihood function, $l(\lambda) = \ln L(\lambda)$:

$$\ell(\lambda) = \ln\left[\left(\frac{1}{\lambda}\right)^n \exp\left(-\frac{1}{\lambda}\sum_{i=1}^{n}(x_i-1)\right)\right]$$

$$\ell(\lambda) = -n\ln(\lambda) - \frac{1}{\lambda}\sum_{i=1}^{n}(x_i - 1)$$

Let $T = \sum_{i=1}^{n}(x_i - 1)$. Note that since $x_i > 1$ for all i, $T > 0$. The log-likelihood is:

$$\ell(\lambda) = -n\ln(\lambda) - \frac{T}{\lambda}$$

3. The Score Equation

The score function $S(\lambda)$ is the first derivative of the log-likelihood with respect to λ:

$$\begin{aligned} U(\lambda) &= \frac{d}{d\lambda}l(\lambda) = \frac{d}{d\lambda}\left(-n\ln(\lambda) - T\lambda^{-1}\right) \\ &= -n\left(\frac{1}{\lambda}\right) - T(-1\lambda^{-2}) \\ &= -\frac{n}{\lambda} + \frac{T}{\lambda^2}. \end{aligned}$$

The MLE, $\hat{\lambda}$, is found by solving the score equation $U(\lambda) = 0$:

$$-\frac{n}{\hat{\lambda}} + \frac{T}{\hat{\lambda}^2} = 0$$

which, after some algebra, yields $\hat{\lambda} = \bar{X} - 1$, where $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ is the sample mean.

To confirm this is a maximum, we check the second derivative (the negative of the observed information):

$$\begin{aligned} \frac{d^2}{d\lambda^2}l(\lambda) = \frac{d}{d\lambda}\left(-n\lambda^{-1} + T\lambda^{-2}\right) \\ &= -n(-1\lambda^{-2}) + T(-2\lambda^{-3}) \\ &= \frac{n}{\lambda^2} - \frac{2T}{\lambda^3}. \end{aligned}$$

Evaluate this at the solution $\hat{\lambda} = T/n$:

$$\left.\frac{d^2}{d\lambda^2}l(\lambda)\right|_{\hat{\lambda}} = \frac{n}{(T/n)^2} - \frac{2T}{(T/n)^3} = \frac{n^3}{T^2} - \frac{2Tn^3}{T^3} = \frac{n^3}{T^2} - \frac{2n^3}{T^2} = -\frac{n^3}{T^2}$$

Since n is a positive integer and $T > 0$, the second derivative is strictly negative. This confirms that $\hat{\lambda}$ is indeed the MLE.

Computing the MLE with Data

The data are given as follows:

$1.884484\ 5.306045\ 3.501055\ 3.494543\ 4.067554\ 1.920979\ 4.425221\ 3.189805\ 1.964026\ 2.612595\ 1.011752$
$4.084178\ 4.114090\ 10.168193\ 2.734757\ 1.932564\ 3.194626\ 2.973989\ 2.975432\ 1.783302$

On the second iteration, using $|\hat{\lambda}^{(t+1)} - \hat{\lambda}^{(t)}| < 1e^{-6}$ as convergence criterion, we obtained an MLE of 2.36696.

Appendix

```r
knitr::opts_chunk$set(
  dev = "cairo_pdf",
  fig.width = 5,
  fig.height = 5,
  fig.align = 'center',
  echo = FALSE,
  message = FALSE,
  warning = FALSE,
  error = FALSE,
  results = 'markup'
)

# Load required libraries
library("tidyverse")
library("patchwork")
library("glue")
library("scales", warn.conflicts = FALSE)
library("extrafont")
library("tinytex")
library("knitr")
library("tidyr")
library("latex2exp")
library("MASS")
library("kableExtra")

theme_set(theme_minimal(base_family = "Roboto Condensed"))

conflicted::conflicts_prefer(
  readr::col_factor(),
  purrr::discard(),
  dplyr::lag(),
  readr::parse_date(),
  kableExtra::group_rows(),
  dplyr::select
)
# Fisher Scoring for the MLE of a truncated exponential distribution
# The update rule is: lambda_new = lambda_old + S(lambda_old) / I(lambda_old)

# 1. Data and Constants
x <- c(
  1.884484,
  5.306045,
  3.501055,
  3.494543,
  4.067554,
```

```r
  1.920979,
  4.425221,
  3.189805,
  1.964026,
  2.612595,
  1.011752,
  4.084178,
  4.114090,
  10.168193,
  2.734757,
  1.932564,
  3.194626,
  2.973989,
  2.975432,
  1.783302
)
n <- length(x)
T_sum <- sum(x - 1)

# 2. Functions for the Score and Fisher Information
# The score function S(lambda)
score <- function(lambda) {
  -n / lambda + T_sum / lambda2
}

# The Fisher Information I(lambda)
fisher_info <- function(lambda) {
  n / lambda2
}

# 3. Fisher Scoring Iterative Algorithm
# Initial value and parameters
lambda_old <- 2.8
tolerance <- 1e-6
iteration <- 0

# Loop until convergence
while (TRUE) {
  iteration <- iteration + 1

  # Calculate score and Fisher information at current lambda
  S_val <- score(lambda_old)
  I_val <- fisher_info(lambda_old)

  # Update lambda using the Fisher Scoring rule
  lambda_new <- lambda_old + S_val / I_val

  # Check for convergence
```

```r
  difference <- abs(lambda_new - lambda_old)

  if (difference < tolerance) {
    break
  }

  lambda_old <- lambda_new
}

# 4. Final Result
# The final result is stored in the 'lambda_new' variable.
# The number of iterations is stored in the 'iteration' variable.

# For comparison, the closed-form MLE is calculated.
mle_closed_form <- T_sum / n
```