1 Task Statement

We received n data points, $x_1, x_2, \ldots, x_n$, and were told that some of them were iid draws from an exponential distribution $\mathrm{Exp}(\lambda)$ and some were generated from a normal distribution $\mathcal{N}(\mu, \sigma^2)$. We further learnt that smaller values were more likely to come from the exponential distribution and larger ones were more likely to come from the normal distribution. We need to decide how many (say k) of the n data points were from the exponential (hence $n-k$ from the normal) and estimate the parameters λ, μ, and σ^2.

2 Derivation of Parameter Estimates

We may firstly sort the data in ascending order. Next, we examine the likelihood function for this size-n dataset:

$$\mathcal{L}(\lambda, \mu, \sigma^2) = \prod_{i=1}^{k} \lambda \mathrm{e}^{-\lambda x_i} \prod_{j=k+1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{-\frac{(x_j-\mu)^2}{2\sigma^2}}, \tag{1}$$

where k (more precisely the mid-way point $[k + (k+1)]/2$) is the cut-off point between the exponential and normal distributions.

The log-likelihood therefore is

$$\begin{aligned}
\ell(\lambda, \mu, \sigma^2) &= \log \left\{ \prod_{i=1}^{k} \lambda \mathrm{e}^{-\lambda x_i} \prod_{j=k+1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{-\frac{(x_j-\mu)^2}{2\sigma^2}} \right\} \\
&= \left\{ \log \left[\lambda^k \right] + \log \left[\prod_{i=1}^{n} \mathrm{e}^{-\lambda x_i} \right] \right\} + \left\{ \log \left[\left(2\pi\sigma^2 \right)^{-\frac{1}{2}(n-k)} \right] + \log \left[\prod_{j=k+1}^{n} \mathrm{e}^{-\frac{(x_j-\mu)^2}{2\sigma^2}} \right] \right\} \\
&= k \log \lambda - \lambda \sum_{i=1}^{k} x_i - \frac{n-k}{2} \log \left(2\pi\sigma^2 \right) - \frac{1}{2\sigma^2} \sum_{j=k+1}^{n} \left(x_j - \mu \right)^2.
\end{aligned} \tag{2}$$

We may use maximum likelihood to estimates λ, μ, and σ^2.

Differentiate ℓ with respect to λ:

$$\frac{\partial \ell}{\partial \lambda} = \frac{k}{\lambda} - \sum_{i=1}^{k} x_i.$$

Apply first order condition:

$$\frac{k}{\widehat{\lambda}_{\mathrm{MLE}}} = \sum_{i=1}^{k} x_i \implies \widehat{\lambda}_{\mathrm{MLE}} = \frac{k}{\sum_{i=1}^{k} x_i}. \tag{3}$$

Differentiate ℓ with respect to μ:

$$\begin{aligned}
\frac{\partial \ell}{\partial \mu} &= -\frac{1}{2\sigma^2} \sum_{j=k+1}^{n} \frac{\partial}{\partial \mu} \left[\left(x_j - \mu \right)^2 \right] = -\frac{1}{2\sigma^2} \sum_{j=k+1}^{n} \left[2 \left(x_j - \mu \right) (-1) \right] \\
&= \frac{1}{\sigma^2} \sum_{j=k+1}^{n} \left(x_j - \mu \right) = \frac{1}{\sigma^2} \sum_{j=k+1}^{n} x_j - \frac{n-k}{\sigma^2} \mu.
\end{aligned}$$

Apply first order condition:

$$\frac{n-k}{\sigma^2} \widehat{\mu}_{\mathrm{MLE}} = \frac{1}{\sigma^2} \sum_{j=k+1}^{n} x_j \implies \widehat{\mu}_{\mathrm{MLE}} = \frac{1}{n-k} \sum_{j=k+1}^{n} x_j. \tag{4}$$

Differentiate ℓ with respect to σ^2:

$$\begin{aligned}\frac{\partial \ell}{\partial \sigma^2} &= -\frac{n-k}{2}\frac{1}{2\pi\sigma^2}(2\pi) - \frac{\partial}{\partial \sigma^2}\left[\frac{1}{2}\left(\sigma^2\right)^{-1}\sum_{j=k+1}^{n}\left(x_j-\mu\right)^2\right] \\ &= -\frac{n-k}{2}\left(\sigma^2\right)^{-1} - \left(-\frac{1}{2}\right)\left(\sigma^2\right)^{-2}\sum_{j=k+1}^{n}\left(x_j-\mu\right)^2 \\ &= -\frac{n-k}{2\sigma^2} + \frac{1}{2\left(\sigma^2\right)^2}\sum_{j=k+1}^{n}\left(x_j-\mu\right)^2.\end{aligned}$$

Apply first order condition:

$$\frac{\sum_{j=k+1}^{n}\left(x_j-\mu\right)^2}{\left(\widehat{\sigma}^2_{\mathrm{MLE}}\right)^2} = \frac{n-k}{\widehat{\sigma}^2_{\mathrm{MLE}}} \implies \widehat{\sigma}^2_{\mathrm{MLE}} = \frac{1}{n-k}\sum_{j=k+1}^{n}\left(x_j-\mu\right)^2. \tag{5}$$

To summarise, the MLEs of λ, μ, and σ^2 are

$$\begin{cases}\widehat{\lambda}_{\mathrm{MLE}} = \dfrac{k}{\sum_{i=1}^{k} x_i}, \\ \widehat{\mu}_{\mathrm{MLE}} = \dfrac{1}{n-k}\displaystyle\sum_{j=k+1}^{n} x_j, \\ \widehat{\sigma}^2_{\mathrm{MLE}} = \dfrac{1}{n-k}\displaystyle\sum_{j=k+1}^{n}\left(x_j-\mu\right)^2.\end{cases} \tag{6}$$

3 Determine Cut-off k

Substitute (6) to (2):

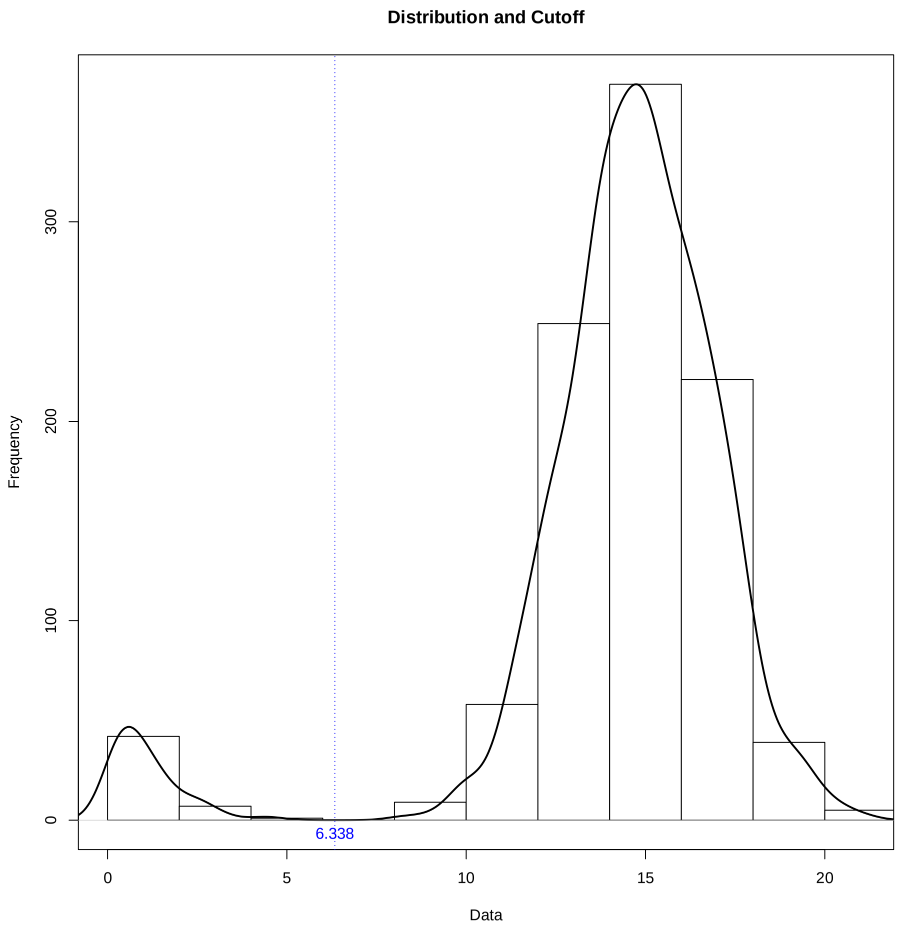
$$\begin{aligned}\widehat{\ell}_{\mathrm{MLE}}(k) = k\log\left[\frac{k}{\sum_{i=1}^{k} x_i}\right] - \frac{k}{\sum_{i=1}^{k} x_i}\sum_{i=1}^{k} x_i - \frac{n-k}{2}\log\left[\frac{2\pi}{n-k}\sum_{j=k+1}^{n}\left(x_j-\mu\right)^2\right] - \frac{1}{2}\frac{n-k}{\sum_{j=k+1}^{n}\left(x_j-\mu\right)^2}\sum_{j=k+1}^{n}\left(x_j-\mu\right)^2 \\ = -k\log\left[\text{mean(exp data)}\right] - k - \frac{n-k}{2}\log\left[2\pi\times\text{var(norm data)}\right] - \frac{n-k}{2}.\end{aligned} \tag{7}$$

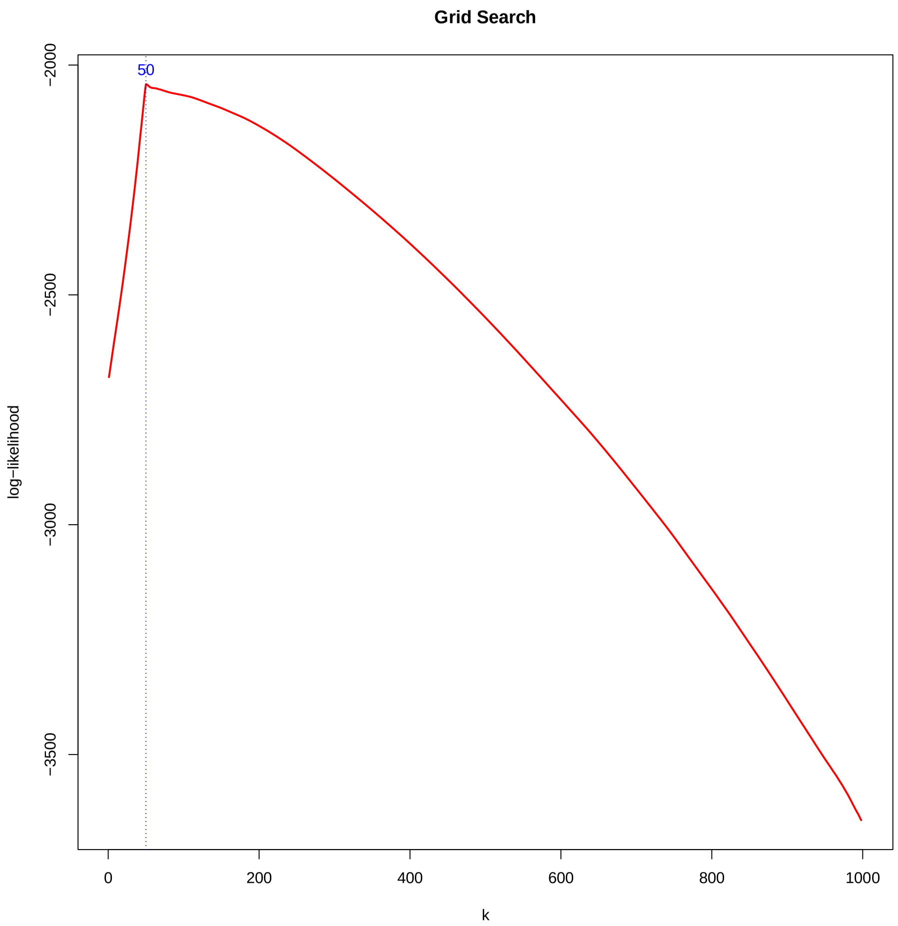
Notice that (7) is now a univariate function of k. Since $\{k \mid k \in \mathbb{Z}, 1 \le k \le n-2\}$ (we need at least the first data point to come from the exponential in order to calculate mean(exp data), and the last two data points to calculate var(norm data)), computers can perform a grid search for $\widehat{k}$ that maximises $\widehat{\ell}_{\mathrm{MLE}}(k)$.

4 Simulation

Using an R seed 2023, we draw 50 random samples from Exp(0.8), mixing with another 950 from $\mathcal{N}(15, 2^2)$. This dataset is shown on the left panel on the next page. The right panel shows the grid search result using (7), yielding $\widehat{k} = 50$. We take the cutoff as the midway point between the 50th and 51st data point, which is 6.338. This data-driven cutoff sits nicely between the two distributions, agreeing with intuition.

Following (6), the point estimates for $\widehat{\lambda}_{\mathrm{MLE}}$, $\widehat{\mu}_{\mathrm{MLE}}$, and $\widehat{\sigma}^2_{\mathrm{MLE}}$ are 0.942, 14.854, and 3.853 respectively, which are not too far from true values 0.8, 15, and 2^2 given the limited sample size, particularly for the exponential part.


Distribution and Cutoff
Frequency
0
100
200
300
6.338
0
5
10
15
20
Data


Grid Search
log−likelihood
−2000
−2500
−3000
−3500
50
0
200
400
600
800
1000
k

5 R Code

```
        ##############################
        # Step 1: Simulate a dataset #
        ##############################

# Generate 1000 mixed data:
# t of them from Exp(0.8)
# 1000-t from N(15,2^2)
t <- 50
set.seed(2023)
data <- c(
    rexp(t, rate = 0.8),
    rnorm(1000 - t, mean = 15, sd = 2)
)
# Sort data in ascending order
data <- data[order(data)]

# Create a placeholder table
n <- length(data)
temp <- data.frame(matrix(NA, nrow = n - 2, ncol = 2))
names(temp) <- c("k", "l")

        #####################################
        # Step 2: Grid search for optimal k #
        #####################################

# Compute log-likelihood for each k
for (k in 1:(n - 2)) {
    # Partition the data into two parts
    data_exp <- data[1:k] # Exponential part
    data_norm <- data[(k + 1):n] # Normal part
    # Compute the log-likelihood
    temp[k, 1] <- k
    temp[k, 2] <- - k * log(mean(data_exp)) - k - (n - k) / 2 * log(2 * pi * var(data_norm)) - (n - k) / 2 # Equation (7)
}

# Locate which row the maximal log-likelihood appears
maxloglik <- max(temp[, 2])
line_maxloglik <- temp[which(temp == maxloglik, arr.ind = TRUE)[1], ]
print(line_maxloglik)
k_hat <- line_maxloglik[[1]]

# Compute cutoff
cutoff <- (data[k_hat] + data[k_hat + 1]) / 2
print(cutoff)

# Visualisation
# Left panel: Data distribution
par(mfrow = c(1, 2))
hist(data,
    col = "white", border = "black",
    xlim=c(min(data), max(data)),
    xlab="Data", ylab="Frequency", main="Distribution and Cutoff"
)
par(new = TRUE)
plot(density(data),
    col = "black", lwd = 2,
    xlim=c(min(data), max(data)),
```

```r
    xaxt = "n", yaxt = "n", ann = FALSE
)
abline(
    v = cutoff,
    col = "blue", lwd = 1, lty = 3
)
text(cutoff, 0, round(cutoff, 3), # Round to 3 decimal places
    cex = 1, pos = 1,col = "blue") # character expansion = 100%; pos = below

# Right panel: Log-likelihood as a function of k
plot(temp[, 2] ~ temp[, 1],
    type = "l", col = "red", lwd = 2,
    xlim = c(0, n), ylim = c(min(temp[, 2]), max(temp[, 2])),
    xlab = "k", ylab = "log-likelihood", main = "Grid Search"
)
abline(
    v = k_hat,
    col = "blue", lwd = 1, lty = 3
)
text(k_hat, maxloglik, k_hat,
    cex = 1, pos = 3, col = "blue")
par(mfrow = c(1, 1))

        ####################################################
        # Step 3: Compute MLEs for distribution parameters #
        ####################################################

# Re-partition the data using the optimal k
data_exp ←data[1:k_hat] # Exponential data
data_norm ←data[(k_hat + 1):n] # Normal data

# Compute MLEs
lambda_hat ←round(1/mean(data_exp), 3)
mu_hat ←round(mean(data_norm), 3)
sigma_sq_hat ←round(var(data_norm), 3)
print(c(lambda_hat, mu_hat, sigma_sq_hat))

        ##############################
        # Step 4: Interval estimates #
        ##############################
```