MAT7035: Computational Statistics

Suggested Solutions to Assignment 5

5.1 Solution. (a) Let $x_1, \ldots, x_n \stackrel{\text{iid}}{\sim} N(\mu, \sigma^2)$, then the MLEs of μ and σ^2 are given by

$$\hat{\mu} = \bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad \text{and} \quad \hat{\sigma}^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n},$$

respectively. Now $n = 24$, $\hat{\mu} = \bar{x} = 43.729$ and $\hat{\sigma}^2 = 79.99$.

(b) The MLE of CV is $\widehat{\text{CV}} = \hat{\sigma}/\hat{\mu} = 0.20453$. We first generated $G = 20,000$ bootstrap samples from $N(\hat{\mu}, \hat{\sigma}^2)$, computed $20,000$ bootstrap replications $\{\widehat{\text{CV}}^*(g)\}_{g=1}^G$, and obtained

$$\begin{aligned} \bar{\text{CV}}^* &= 0.1984, \\ \widehat{\text{Se}}^*(\widehat{\text{CV}}) &= 0.030644, \\ [\widehat{\text{CV}}_L^*,\ \widehat{\text{CV}}_U^*] &= [0.14164,\ 0.26093]. \end{aligned}$$

The R code is as follows.

```
function(G)
{
     # function name: CS.Exercise5.1(G=20000)
     # Call: c(thmean, thstd, thl, thu, thL, thU)
     #       <- mean.std.CI(thsample)
     # G is the bootstrap sample size
     x <- c(32., 46.4, 48.1, 27.7, 35.5, 52.6, 66., 41.3,
           49.9, 36.1, 50., 44.7, 48.2, 36.9, 40.8, 35.1,
```

```
          63.3, 42.5, 52.4, 40.9, 38.6, 43.2, 41.7, 35.6)
     n <- length(x)
     muMLE <- mean(x)
     si2MLE <- sum((x - muMLE) * (x - muMLE))/n
     siMLE <- sqrt(si2MLE)
     CVMLE <- siMLE/muMLE
     CV.star.sample <- matrix(0, G, 1)
     for(g in 1:G) {
          xstar <- rnorm(n, mean = muMLE, sd = siMLE)
          mustar <- mean(xstar)
          CVstar <- sqrt(sum((xstar - mustar) * (xstar -
                    mustar))/n)/mustar
          CV.star.sample[g, 1] <- CVstar
     }
     M <- mean.std.CI(CV.star.sample)
     CVmean <- M[[1]]
     CVstd <- M[[2]]
     CVL <- M[[5]]
     CVU <- M[[6]]
     return(CVMLE, CVmean, CVstd, CVL, CVU)
}
```

(c1) The parametric bootstrap method. The MLE of the population median θ is $\hat{\theta} = (x_{(12)} + x_{(13)})/2 = 42.1$. We first generated $G = 20,000$ bootstrap samples from $N(\hat{\mu}, \hat{\sigma}^2)$, computed $20,000$ bootstrap replications $\{\hat{\theta}^*(g)\}_{g=1}^G$, and obtained

$$
\begin{aligned}
\bar{\theta}^* &= 43.76, \\
\widehat{\mathrm{Se}}^*(\hat{\theta}) &= 2.2594, \\
[\hat{\theta}_L^*,\ \hat{\theta}_U^*] &= [39.346,\ 48.14].
\end{aligned}
$$

The R code is as follows.

```
function(G)
{
     # Name: CS.Exercise5.1.parametric.median(G=20000)
     # Call: c(thmean, thstd, thl, thu, thL, thU)
     #       <- mean.std.CI(thsample)
     # G is the bootstrap sample size
     x <- c(32., 46.4, 48.1, 27.7, 35.5, 52.6, 66., 41.3,
          49.9, 36.1, 50., 44.7, 48.2, 36.9, 40.8, 35.1,
          63.3, 42.5, 52.4, 40.9, 38.6, 43.2, 41.7, 35.6)
     n <- length(x)
     muMLE <- mean(x)
     si2MLE <- sum((x - muMLE) * (x - muMLE))/n
     siMLE <- sqrt(si2MLE)
     thMLE <- median(x)
     th.star.sample <- matrix(0, G, 1)
     for(g in 1:G) {
     xstar <- rnorm(n, mean = muMLE, sd = siMLE)
          thstar <- median(xstar)
          th.star.sample[g, 1] <- thstar
     }
     M <- mean.std.CI(th.star.sample)
     thmean <- M[[1]]
     thstd <- M[[2]]
     thL <- M[[5]]
     thU <- M[[6]]
     return(thMLE, thmean, thstd, thL, thU)
}
```

(c2) The non-parametric bootstrap method. We first generated $G =$

20,000 bootstrap samples from the empirical distribution based on $x_1, \ldots, x_n$, computed 20,000 bootstrap replications $\{\hat{\theta}^*(g)\}_{g=1}^{G}$, and obtained

$$
\begin{aligned}
\bar{\theta}^* &= 42.552, \\
\widehat{\mathrm{Se}}^*(\hat{\theta}) &= 2.1089, \\
[\hat{\theta}_L^*,\ \hat{\theta}_U^*] &= [338.6,\ 48.1].
\end{aligned}
$$

The R code is as follows.

```
function(G)
{
     # Name: CS.Exercise5.1.nonparametric.median(G=20000)
     # Call: c(thmean, thstd, thl, thu, thL, thU)
     #       <- mean.std.CI(thsample)
     # G is the bootstrap sample size
     x <- c(32., 46.4, 48.1, 27.7, 35.5, 52.6, 66., 41.3,
          49.9, 36.1, 50., 44.7, 48.2, 36.9, 40.8, 35.1,
          63.3, 42.5, 52.4, 40.9, 38.6, 43.2, 41.7, 35.6)
     n <- length(x)
     p <- rep(1/n, n)
     th.star.sample <- matrix(0, G, 1)
     for(g in 1:G) {
          xstar <- sample(x, n, prob = p, replace = T)
          thstar <- median(xstar)
          th.star.sample[g, 1] <- thstar
     }
     M <- mean.std.CI(th.star.sample)
     thmean <- M[[1]]
     thstd <- M[[2]]
     thL <- M[[5]]
```

```
        thU <- M[[6]]
        return(thmean, thstd, thL, thU)
    }
```

5.2 Solution. (a) The observed likelihood function for (ϕ, λ) is

$$L(\phi, \lambda | Y_{\text{obs}}) = [\phi + (1-\phi)\,\mathrm{e}^{-\lambda}]^m \times (1-\phi)^{n-m} \prod_{y_i \notin \mathbb{O}} \frac{\mathrm{e}^{-\lambda}\lambda^{y_i}}{y_i!}.$$

(b) We augment Y_{obs} with a latent r.v. Z by splitting the observed m into Z and $(m - Z)$ so that the conditional predictive distribution is

$$f(z|Y_{\text{obs}}, \phi, \lambda) = \text{Binomial}\Big(z \big| m,\ \phi/[\phi + (1-\phi)\,\mathrm{e}^{-\lambda}]\Big). \tag{6.1}$$

Note that the complete-data likelihood for (ϕ, λ) is given by

$$\begin{aligned} L(\phi, \lambda | Y_{\text{obs}}, z) &\propto \phi^z [(1-\phi)\,\mathrm{e}^{-\lambda}]^{m-z} \times (1-\phi)^{n-m} \prod_{y_i \notin \mathbb{O}} \frac{\mathrm{e}^{-\lambda}\lambda^{y_i}}{y_i!} \\ &\propto \phi^z (1-\phi)^{n-z}\,\mathrm{e}^{-(n-z)\lambda} \lambda^{\sum_{y_i \notin \mathbb{O}} y_i}. \end{aligned}$$

Thus, the complete-data MLEs are given by

$$\hat{\phi} = \frac{z}{n} \quad \text{and} \quad \hat{\lambda} = \frac{\sum_{y_i \notin \mathbb{O}} y_i}{n - z}. \tag{6.2}$$

Hence, the E-step is to compute the conditional expectation

$$E(Z|Y_{\text{obs}}, \phi, \lambda) = \frac{m\phi}{\phi + (1-\phi)\,\mathrm{e}^{-\lambda}}, \tag{6.3}$$

and the M-step is to update (6.2) by replacing z with $E(Z|Y_{\text{obs}}, \phi, \lambda)$.

(c) We first present the algorithm for generating a sample from $Y \sim \text{ZIP}(\phi, \lambda)$. Let $X \sim \text{Poisson}(\lambda)$, then we have

$$Y = \begin{cases} 0, & \text{with probability } \phi, \\ X, & \text{with probability } 1 - \phi. \end{cases}$$

Step A: Draw $U \sim U(0,1)$ and independently draw $X \sim \text{Poisson}(\lambda)$.
Step B: If $U \leqslant \phi$, then $Y = 0$; otherwise $Y = X$.

Let $\theta = \phi$ or $\theta = \lambda$. The parametric bootstrap method for obtaining $100(1-\alpha)\%$ bootstrap CIs for ϕ and λ is as follows:

Step 1. Calculate the MLEs $\hat{\phi}$ and $\hat{\lambda}$ for ϕ and λ via the EM algorithm (6.2) and (6.3).

Step 2. Generate a bootstrap sample $\mathbf{y}^* = (y_1^*, \dots, y_n^*) \overset{\text{iid}}{\sim} \text{ZIP}(\hat{\phi}, \hat{\lambda})$ and compute the corresponding bootstrap replication $\hat{\phi}^*$ and $\hat{\lambda}^*$, or $\hat{\theta}^*$.

Step 3. Independently repeating this process (i.e., Step 2) G times, we obtain G bootstrap replications $\{\hat{\theta}^*(g)\}_{g=1}^G$.

Step 4. Consequently, the standard error, $\text{Se}(\hat{\theta})$, of $\hat{\theta}$ can be estimated by the sample standard deviation of the G replications, i.e.,

$$\widehat{\text{Se}}^*(\hat{\theta}) = \sqrt{\frac{1}{G-1}\sum_{g=1}^{G}[\hat{\theta}^*(g) - \bar{\theta}^*]^2}\,, \tag{6.4}$$

where

$$\bar{\theta}^* = [\hat{\theta}^*(1) + \cdots + \hat{\theta}^*(G)]/G. \tag{6.5}$$

Step 5. If $\{\hat{\theta}^*(g)\}_{g=1}^G$ are approximately normally distributed, a $100(1-\alpha)\%$ bootstrap CI for θ is

$$[\hat{\theta}_l^*, \;\; \hat{\theta}_u^*] = [\bar{\theta}^* - z_{\alpha/2} \cdot \widehat{\text{Se}}^*(\hat{\theta}), \;\; \bar{\theta}^* + z_{\alpha/2} \cdot \widehat{\text{Se}}^*(\hat{\theta})]. \tag{6.6}$$

Step 6. If the bootstrap CI (6.13) is beyond the unit interval $[0,1]$ or the bootstrap replications $\{\hat{\theta}^*(g)\}_{g=1}^G$ are non-normally distributed, a $100(1-\alpha)\%$ bootstrap CI for θ is

$$[\hat{\theta}_L^*, \;\; \hat{\theta}_U^*], \tag{6.7}$$

where $\hat{\theta}_L^*$ and $\hat{\theta}_U^*$ are the $(\alpha/2)G$-th and the $(1-\alpha/2)G$-th order statistics of $\{\hat{\theta}^*(g)\}_{g=1}^G$.

5.3 Solution. (a) The observed likelihood function for π is

$$
\begin{aligned}
L(\pi|Y_{\rm obs}) &= \prod_{i=1}^{n}\left[\frac{1}{1-(1-\pi)^m}\cdot\binom{m}{x_i}\pi^{x_i}(1-\pi)^{m-x_i}\right] \\
&\propto \pi^{n\bar{x}}(1-\pi)^{n(m-\bar{x})}\cdot\left[\frac{1}{1-(1-\pi)^m}\right]^n,
\end{aligned}
$$

where $\bar{x}=(1/n)\sum_{i=1}^n x_i$. So the log-likelihood function is

$$
\begin{aligned}
& \ell(\pi|Y_{\rm obs}) \\
= \;& n\bar{x}\log(\pi)+n(m-\bar{x})\log(1-\pi)-n\log[1-(1-\pi)^m] \\
= \;& n[\bar{x}\log(\pi)+(m-\bar{x})\log(1-\pi)+h(\pi)] \qquad (6.8)
\end{aligned}
$$

where $h(\pi)=-\log[1-(1-\pi)^m]$. Define $(1-\pi)^m=\mathrm{e}^{-\lambda}$ or

$$\lambda=-m\log(1-\pi), \tag{6.9}$$

we have $h(\pi)=-\log(1-\mathrm{e}^{-\lambda})\,\hat{=}\,g(\lambda)$. Since $g'(\lambda)=-\mathrm{e}^{-\lambda}/(1-\mathrm{e}^{-\lambda})$ and $g''(\lambda)=\mathrm{e}^{-\lambda}/(1-\mathrm{e}^{-\lambda})^2>0$ for all $\lambda>0$. Thus, $g(\lambda)$ is a strictly convex function. Applying the second order Taylor expansion, we have

$$g(\lambda)\geqslant g(\lambda^{(t)})+(\lambda-\lambda^{(t)})g'(\lambda^{(t)}), \quad \forall\lambda>0, \quad \lambda^{(t)}>0.$$

or

$$
\begin{aligned}
h(\pi) &\geqslant h(\pi^{(t)})-\frac{(1-\pi^{(t)})^m}{1-(1-\pi^{(t)})^m}[-m\log(1-\pi)+m\log(1-\pi^{(t)})] \\
&= c_0+\frac{m(1-\pi^{(t)})^m}{1-(1-\pi^{(t)})^m}\log(1-\pi).
\end{aligned}
$$

We have

$$
\begin{aligned}
& \ell(\pi|Y_{\rm obs}) \\
= \;& n[\bar{x}\log(\pi)+(m-\bar{x})\log(1-\pi)+h(\pi)] \\
\geqslant \;& n\left[\bar{x}\log(\pi)+(m-\bar{x})\log(1-\pi)+c_0+\frac{m(1-\pi^{(t)})^m}{1-(1-\pi^{(t)})^m}\log(1-\pi)\right] \\
\hat{=} \;& Q(\pi|\pi^{(t)}).
\end{aligned}
$$

Let $\mathrm{d}Q(\pi|\pi^{(t)})/\mathrm{d}\lambda = 0$, we have the following MM algorithm

$$\pi^{(t+1)} = \frac{\bar{x}[1 - (1 - \pi^{(t)})^m]}{m}. \tag{6.10}$$

(b) The parametric bootstrap method for obtaining $100(1-\alpha)\%$ bootstrap CI for π is as follows:

Step 1. Calculate the MLE $\hat{\pi}$ for π via the MM algorithm (6.10).

Step 2. Generate a bootstrap sample $\mathbf{x}^* = (x_1^*, \dots, x_n^*) \stackrel{\text{iid}}{\sim} \text{ZTB}(m, \hat{\pi})$ and compute the corresponding bootstrap replication $\hat{\pi}^*$.

Step 3. Independently repeating this process (i.e., Step 2) G times, we obtain G bootstrap replications $\{\hat{\pi}^*(g)\}_{g=1}^G$.

Step 4. Consequently, the standard error, $\text{Se}(\hat{\pi})$, of $\hat{\pi}$ can be estimated by the sample standard deviation of the G replications, i.e.,

$$\widehat{\text{Se}}^*(\hat{\pi}) = \sqrt{\frac{1}{G-1}\sum_{g=1}^{G}[\hat{\pi}^*(g) - \bar{\theta}^*]^2}\ , \tag{6.11}$$

where

$$\bar{\theta}^* = [\hat{\pi}^*(1) + \cdots + \hat{\pi}^*(G)]/G. \tag{6.12}$$

Step 5. If $\{\hat{\pi}^*(g)\}_{g=1}^G$ are approximately normally distributed, a $100(1-\alpha)\%$ bootstrap CI for θ is

$$[\hat{\pi}_l^*,\ \ \hat{\pi}_u^*] = [\bar{\theta}^* - z_{\alpha/2}\cdot\widehat{\text{Se}}^*(\hat{\pi}),\ \ \bar{\theta}^* + z_{\alpha/2}\cdot\widehat{\text{Se}}^*(\hat{\pi})]. \tag{6.13}$$

Step 6. If the bootstrap CI (6.13) is beyond the unit interval $[0, 1]$ or the bootstrap replications $\{\hat{\pi}^*(g)\}_{g=1}^G$ are non-normally distributed, a $100(1-\alpha)\%$ bootstrap CI for θ is

$$[\hat{\pi}_L^*,\ \ \hat{\pi}_U^*], \tag{6.14}$$

where $\hat{\pi}_L^*$ and $\hat{\pi}_U^*$ are the $(\alpha/2)G$-th and the $(1-\alpha/2)G$-th order statistics of $\{\hat{\pi}^*(g)\}_{g=1}^G$.