HW#6

- **Example 10.1.21**

Let $X_1, \cdots, X_n$ be independent and identical distributed random variables from $\text{Ber}(p)$. Then the maximum likelihood estimator (MLE) of p is $\hat{p} = \Sigma_{i=1}^n X_i / n$. By the invariant property of the MLE, the MLE of $p(1-p)$ is $\hat{p}(1-\hat{p}) = \Sigma_{i=1}^n X_i(1-\Sigma_{i=1}^n X_i/n)/n$.

We first derive the true variance of $\hat{p}(1-\hat{p})$. Since $Y \equiv \Sigma_{i=1}^n X_i \sim \text{Bin}(n, p)$, we have

$$\begin{aligned}\text{var}\{\hat{p}(1-\hat{p})\} &= \text{var}\left\{\frac{1}{n}\sum_{i=1}^n X_i - \frac{1}{n^2}\left(\sum_{i=1}^n X_i\right)^2\right\} = \text{var}\left\{\frac{1}{n}Y - \frac{1}{n^2}Y^2\right\} \\ &= \frac{1}{n^2}\text{var}(Y) + \frac{1}{n^4}\text{var}(Y^2) - \frac{2}{n^3}\text{cov}(Y, Y^2) \\ &= \frac{E(Y^2) - \{E(Y)\}^2}{n^2} + \frac{E(Y^4) - \{E(Y^2)\}^2}{n^4} - \frac{2\{E(Y^3) - E(Y)E(Y^2)\}}{n^3},\end{aligned}$$

where the binominal moments are

$$E(Y) = np, \qquad E(Y^2) = np(1-p+np),$$

$$E(Y^3) = np(1-3p+3np-2p^2-3np^2+n^2p^2),$$

$$E(Y^4) = np(1-7p+7np-12p^2-18np^2+6n^2p^2-6p^3+11np^3-6n^2p^3+n^3p^3).$$

Therefore, we have obtained the true variance of $\hat{p}(1-\hat{p})$.

However, the true variance of $\hat{p}(1-\hat{p})$ depends on p. In real applications, we can never know the true value of p. Thus, we have to apply some methods to estimate its variance. One possible approach is applying the delta method. According to Example 10.1.14, we have

$$\sqrt{n}(\hat{p}-p) \xrightarrow{d} N(0, p(1-p)), \qquad \text{as } n \to \infty,$$

where "$\xrightarrow{d}$" denotes convergence in distribution. Let $g(p) = p(1-p)$ then we have $g'(p) = dg(p)/dp = 1-2p$ and $g''(p) = dg'(p)/dp = -2$. For $p \neq 1/2$ (i.e., $g'(p) \neq 0$), by applying the first-order delta method, we obtain

$$\sqrt{n}\{\hat{p}(1-\hat{p}) - p(1-p)\} \xrightarrow{d} N(0, p(1-p)(1-2p)^2), \qquad \text{as } n \to \infty.$$

For $p = 1/2$, by applying the second-order delta method, we obtain

$$n\{\hat{p}(1-\hat{p}) - p(1-p)\} \xrightarrow{d} p(p-1)\chi^2_{\mathrm{df}=1}, \quad \text{as} \quad n \to \infty.$$

With replacing p by $\hat{p}$, we can estimate the variance of $\hat{p}(1-\hat{p})$ by

$$\frac{\hat{p}(1-\hat{p})(1-2\hat{p})^2}{n}, \quad \text{if} \quad p \neq 1/2; \qquad \frac{2\hat{p}^2(1-\hat{p})^2}{n^2}, \quad \text{if} \quad p = 1/2.$$

Another possible approach is applying the non-parametric bootstrap method to approximate the distribution of $\hat{p}(1-\hat{p})$ then estimates its variance. Concretely, we perform:

Algorithm 1   Non-parametric bootstrap variance

Let B be a large integer.

Step 1.   Resample $X_1^{(b)}, \cdots, X_n^{(b)}$ from data $X_1, \cdots, X_n$ with replacement for $b = 1, \cdots, B$.

Step 2.   Based on the bootstrap samples, compute

$$\hat{p}^{(b)}(1-\hat{p}^{(b)}) = \frac{1}{n}\sum_{i=1}^{n} X_i^{(b)}\left(1 - \frac{1}{n}\sum_{i=1}^{n} X_i^{(b)}\right), \quad \text{for each} \quad b = 1, \cdots, B.$$

Step 3.   Estimate $\mathrm{var}\{\hat{p}(1-\hat{p})\}$ by

$$\mathrm{var}_B\{\hat{p}(1-\hat{p})\} = \frac{1}{B-1}\sum_{b=1}^{B}\{\hat{p}^{(b)}(1-\hat{p}^{(b)}) - \overline{\hat{p}^{(b)}(1-\hat{p}^{(b)})}\}^2,$$

where $\overline{\hat{p}^{(b)}(1-\hat{p}^{(b)})} = \Sigma_{b=1}^{B}\hat{p}^{(b)}(1-\hat{p}^{(b)})/B$.

Remark 1:   Following the textbook, we set $n = 24$ for simulations. For the case $\hat{p} = 1/4$, we set data $X_1 = \cdots = X_6 = 1$ and $X_7 = \cdots = X_{24} = 0$ to yield the desired MLE. Similarly, we set data $X_1 = \cdots = X_{12} = 1$ and $X_{13} = \cdots = X_{24} = 0$ for the case $\hat{p} = 1/2$; $X_1 = \cdots = X_{16} = 1$ and $X_{17} = \cdots = X_{24} = 0$ for the case $\hat{p} = 2/3$.

Now, we compare the estimates of $\mathrm{var}\{\hat{p}(1-\hat{p})\}$ based on the bootstrap (Algorithm 1) and delta method. Table 1 shows that the results on the delta method agree with the textbook. However, the results on the bootstrap method and true value violate the textbook. Table 1 also reveals that the bootstrap variance gives a slightly better approximation than the delta method variance. This is due to that the delta method relies on asymptotic approximation while our sample size $n = 24$ is small. One should note that the computational cost of the bootstrap method is much higher than the delta method. We provide R codes that reproduce Table 1 in Appendix 1.

Table 1. The non-parametric bootstrap and delta method variances for $\hat{p}(1-\hat{p})$. The true variance is calculated by assuming $\hat{p} = p$.

Variance	$\hat{p} = 1/4$	$\hat{p} = 1/2$	$\hat{p} = 2/3$
Bootstrap ($B = 1{,}000$)	0.002068	0.000214	0.001105
Bootstrap ($B = 10{,}000$)	0.001963	0.000208	0.001087
Bootstrap ($B = 100{,}000$)	0.001904	0.000206	0.001110
Delta Method	0.001953	0.000217	0.001029
True	0.001911	0.000208	0.001109

- **Example 10.1.22**

Suppose that we have a sample

-1.81, 0.63, 2.22, 2.41, 2.95, 4.16, 4.24, 4.53, 5.09.

with

$$\overline{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i = 2.71 \quad \text{and} \quad S_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X}_n)^2 = 4.83.$$

We aim to apply the parametric bootstrap method based on normal distribution to estimate the variance of S_n^2. Here, we first follow the textbook and estimate the normal parameters μ and σ^2 by $\overline{X}_n$ and S_n^2, respectively. Concretely, we perform:

Algorithm 2 Parametric bootstrap variance (textbook)

Let B be a large integer.

Step 1. Generate samples $X_1^{(b)}, \cdots, X_n^{(b)}$ from $N(\overline{X}_n = 2.713, S_n^2 = 4.820)$ for $b = 1, \cdots, B$.

Step 2. Based on the bootstrap samples, compute

$$S_n^{2(b)} = \frac{1}{n-1}\sum_{i=1}^{n}(X_i^{(b)} - \overline{X}_n^{(b)})^2, \quad \text{for each } b = 1, \cdots, B,$$

where $\overline{X}_n^{(b)} = \Sigma_{i=1}^{n} X_i^{(b)} / n$.

Step 3. Estimate $\text{var}(S_n^2)$ by

$$\text{var}_B(S_n^2) = \frac{1}{B-1}\sum_{i=1}^{B}(S_n^{2(b)} - \overline{S}_n^{2(b)})^2,$$

where $\overline{S}_n^{2(b)} = \Sigma_{b=1}^{B} \overline{S}_n^{2(b)} / B$.

By using Algorithm 2 with $B = 1000$, we obtain $\text{var}_B(S_n^2) = 5.764$. On the other hand, based on the normal assumption, we have $(n-1)S_n^2/\sigma^2 \sim \chi^2_{\text{df}=n-1}$. Therefore, we obtain $\text{var}(S_n^2) = 2\sigma^4/(n-1)$. The textbook suggests to estimate it by utilizing the MLE. However, the textbook mistakenly estimates σ^2 by the sample variance. The MLE should be

$2(\hat{\sigma}^2)^2/(9-1)=4.590$, where $\hat{\sigma}^2=\Sigma_{i=1}^n(X_i-\overline{X}_n)^2/n=4.285$. In addition, according to the textbook, the true variance of S_n^2 is 4. Thus, our results violate the textbook. The MLE gives a better approximation to the true variance.

Algorithm 2 can be improved by estimating normal parameters based on MLEs $\hat{\mu}=2.713$ and $\hat{\sigma}^2=4.285$. To be specific, we perform:

Algorithm 3 Parametric bootstrap variance (MLE)

Let B be a large integer.

Step 1. Generate samples $X_1^{(b)},\cdots,X_n^{(b)}$ from $N(\hat{\mu}=2.713,\hat{\sigma}^2=4.285)$ for $b=1,\cdots,B$.

Step 2. Based on the bootstrap samples, compute

$$S_n^{2(b)}=\frac{1}{n-1}\sum_{i=1}^n(X_i^{(b)}-\overline{X}_n^{(b)})^2, \quad \text{for each } b=1,\cdots,B,$$

where $\overline{X}_n^{(b)}=\Sigma_{i=1}^n X_i^{(b)}/n$.

Step 3. Estimate $\text{var}(S_n^2)$ by

$$\text{var}_B(S_n^2)=\frac{1}{B-1}\sum_{i=1}^B(S_n^{2(b)}-\overline{S}_n^{2(b)})^2,$$

where $\overline{S}_n^{2(b)}=\Sigma_{b=1}^B\overline{S}_n^{2(b)}/B$.

By using Algorithm 3 with $B=1000$, we obtain $\text{var}_B(S_n^2)=4.554$ which is more close to the true value and agree with the MLE. We summarize all the results in Table 2. R codes that reproduce Table 2 are available in Appendix 2.

Table 2. The parametric bootstrap variance and the MLE.

Method	Bootstrap (Algorithm 2)	Bootstrap (Algorithm 3)	MLE	True
Variance	5.764	4.554	4.590	4.000

Appendix 1 R codes for Example 10.1.21

```
B = 1000
n = 24
p.hat = 1/4
### bootstrap method ###
var.boot = rep(0,B)
set.seed(816)
data = c(rep(1,n*p.hat),rep(0,n*(1-p.hat))); mean(data)
for (b in 1:B) {

   data.boot = sample(data,n,replace = TRUE)
   p.boot = mean(data.boot)
   var.boot[b] = p.boot*(1-p.boot)

}
round(var(var.boot),6)

### delta method ###
if (p.hat != 1/2) {

   round(p.hat*(1-p.hat)*(1-2*p.hat)^2/n,6)

} else {

   round(p.hat^2*(p.hat-1)^2/n^2*2,6)

}

### true ###
p = p.hat
EY   = n*p
EY2 = n*p*(1-p+n*p)
EY3 = n*p*(1-3*p+3*n*p+2*p^2-3*n*p^2+n^2*p^2)
EY4 = n*p*(1-7*p+7*n*p+12*p^2-18*n*p^2+6*n^2*p^2-6*p^3+11*n*p^3-6*n^2*p^3+n^3*p^3)
round((EY2-EY^2)/n^2+(EY4-EY2^2)/n^4-2*(EY3-EY*EY2)/n^3,6)
```

Appendix 2   R codes for Example 10.1.22

```
x = c(-1.81,0.63,2.22,2.41,2.95,4.16,4.24,4.53,5.09)
B = 1000
n = length(x); n
mu = mean(x); mu
s2.sample = var(x); s2.sample
s2.MLE = sum((x-mean(x))^2)/n; s2.MLE
s2.boot = rep(0,B)
set.seed(816)
for (b in 1:B) {

   x.boot = rnorm(n,mean = mu,sd = sqrt(s2.sample)) # Algorithm 2
   #x.boot = rnorm(n,mean = mu,sd = sqrt(s2.MLE)) # Algorithm 3
   s2.boot[b] = var(x.boot)

}
var(s2.boot)
2*s2.MLE^2/(n-1)
```