CS536 HW1: Estimation Problems

Student Name: Xuenan Wang

NetID: xw336

Uniform Estimators

Let $X_1, X_2, \ldots, X_n$ be i.i.d. random variables, uniformly distributed on $[0, L]$ (i.e., with density $1/L$ on this interval). In the posted notes on estimation, it is shown that the method of moments and maximum likelihood estimators for L are given by

$$\begin{aligned} \hat{L}_{\text{MOM}} &= 2\overline{X}_n \\ \hat{L}_{\text{MLE}} &= \max_{i=1,\ldots,n} X_i. \end{aligned} \tag{1}$$

We want to consider the question of which estimator is better. If $\hat{L}$ is meant to be an estimator for L, we define the **mean squared error** to be

$$\text{MSE}(\hat{L}) = \mathbb{E}\left[\left(\hat{L} - L\right)^2\right], \tag{2}$$

the expected square discrepancy between the estimator and the thing it is supposed to be estimating.

1) Show that in general, $\text{MSE}(\hat{\theta}) = \text{bias}(\hat{\theta})^2 + \text{var}(\hat{\theta})$, where var is the variance, and bias is given by

$$\text{bias}(\hat{\theta}) = \theta - \mathbb{E}\left[\hat{\theta}\right]. \tag{3}$$

Note that an estimator might have no bias, but huge variance, or no variance (constant), but significant bias - the MSE summarizes these two sources of 'error' in an estimator.

$MSE(\theta^*) = E[(\theta^* - \theta)^2] = E[(\theta^* - \mu + \mu - \theta)^2], where\ \mu = E[\theta]$

$$\begin{aligned} MSE(\theta^*) &= \int_0^L (\theta^* - \mu + \mu - \theta)^2 f(x)dx \\ &= \int_0^L (\theta^* - \mu)^2 f(x)dx + \int_0^L (\mu - \theta)^2 f(x)dx + \int_0^L 2(\theta^* - \mu)(\mu - \theta) f(x)dx \\ &= E[(\theta^* - \mu)^2] + E[(\mu - \theta)^2] + 2E[(\theta^* - \mu)(\mu - \theta)] \end{aligned}$$

$E[(\theta^* - \mu)^2] = Var(\theta^*)$

$E[(\mu - \theta)^2] = (\mu - \theta)^2 = (\theta - \mu)^2 = (\theta - E[\theta^*])^2 = bias(\theta^*)^2$

$E[(\theta^* - \mu)(\mu - \theta)] = (\mu - \theta)E[\theta^* - \mu] = 0$

Therefore,

$$MSE(\theta^*) = bias(\theta^*)^2 + Var(\theta^*)$$

> 2) Compute the bias of $\hat{L}_{\text{MOM}}$ and $\hat{L}_{\text{MLE}}$. In general, $\hat{L}_{\text{MLE}}$ consistently underestimates L - why? *Hint: What is the pdf for $\hat{L}_{MLE}$?*

We know that $L^*_{MOM} = 2\overline{X}_n = L$, so

$$bias(L^*_{MOM}) = L - E[L^*_{MOM}] = L - 2E[\overline{X}_n] = L - \frac{2}{n}\sum_{i=1}^{n} E[X_i] = L - \frac{2}{n} \times n \times \frac{L}{2} = 0$$

This indicates that L^*_{MOM} is unbiased.

$$f_L(x) = \frac{d}{dx}P(X_i \leq x) = \frac{d}{dx}\left[\left(\frac{x}{L}\right)^n\right] = \frac{nx^{n-1}}{L^n}$$

$$E[L^*_{MLE}] = \int_0^L x f_L(x)dx = \frac{n}{L^n}\left[\frac{x^{n+1}}{n+1}\right]_0^L = \frac{nL}{n+1}$$

Therefore,

$$bias(L^*_{MLE}) = L - E[L^*_{MLE}] = \frac{L}{n+1} > 0$$

This indicates that L^*_{MLE} is biased.

Generally, we have $bias(L^*_{MLE})$ always larger than 0, which leads $L > E[L^*_{MLE}]$. And this is the reason why L^*_{MLE} constantly underestimates L.

> 3) Compute the variance of $\hat{L}_{\text{MOM}}$ and $\hat{L}_{\text{MLE}}$.

$$Var(L^*_{MOM}) = Var(2\overline{X}_n) = 4Var(\overline{X}_n) = 4 \times \frac{1}{n^2}\sum_{i=1}^{n} Var(X_i) = 4 \times \frac{L^2}{12n} = \frac{L^2}{3n}$$

$$Var(L^*_{MLE}) = \int_0^L x^2 f_{L^*}(x)dx - (E[L^*])^2 = \frac{n}{L^n}\left[\frac{x^{n+2}}{n+2}\right]_0^L - \left(\frac{nL}{n+1}\right)^2 = \frac{nL^2}{(n+1)^2(n+2)}$$

4) Which one is the better estimator, i.e., which one has the smaller mean squared error?

$$MSE(L^*_{MOM}) = Var(L^*_{MOM}) = \frac{L^2}{3n}$$

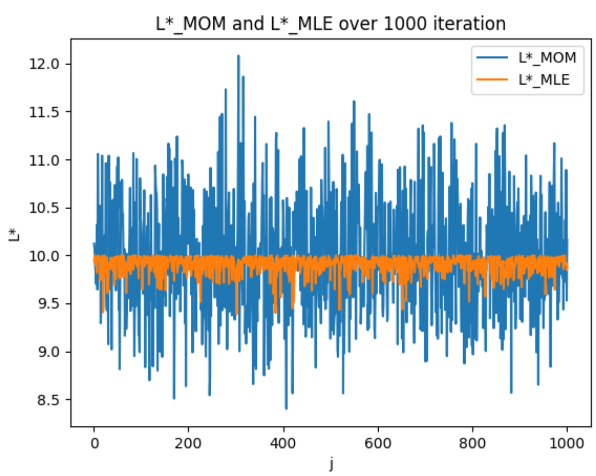
$$MSE(L^*_{MLE}) = bias(L^*_{MLE})^2 + Var(L^*_{MLE}) = \left(\frac{L}{n+1}\right)^2 + \frac{nL^2}{(n+1)^2(n+2)} = \frac{2L^2}{(n+1)(n+2)}$$

$$MSE(L^*_{MOM}) - MSE(L^*_{MLE}) = \frac{L^2}{3n} - \frac{2L^2}{(n+1)(n+2)} = \frac{L^2(n-1)(n-2)}{3n(n+1)(n+2)}$$

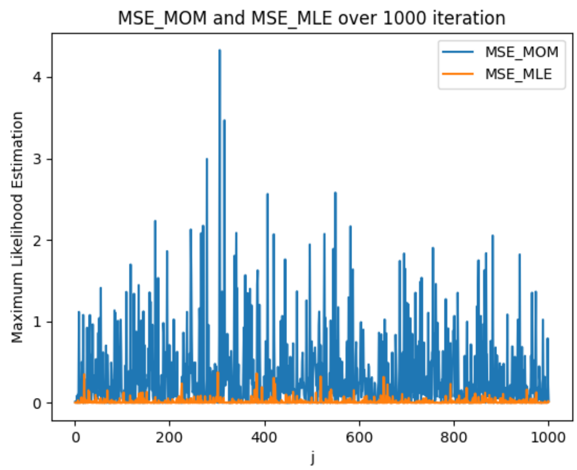
As $n \to \infty$, $(n-1)(n-2) > 0$ and $(n+1)(n+2) > 0$, which gave us $MSE(L^*_{MOM}) - MSE(L^*_{MLE}) > 0$. Therefore, $MSE(L^*_{MOM}) > MSE(L^*_{MLE})$, which makes MLE a better estimator.

5) Experimentally verify your computations in the following way: Taking $n = 100$ and $L = 10$,

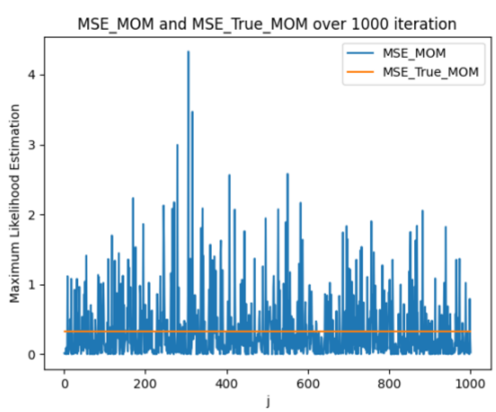
- For $j = 1, \ldots, 1000$:
  - Simulate $X_1^j, \ldots, X_n^j$ and compute values for $\hat{L}^j_{\text{MOM}}$ and $\hat{L}^j_{\text{MLE}}$
- Estimate the mean squared error for each population of estimator values.
- How do these estimated MSEs compare to your theoretical MSEs?

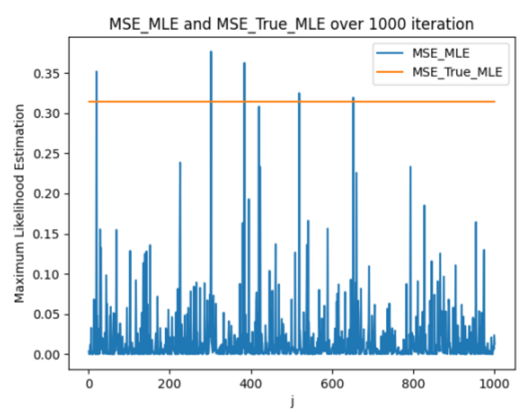


From above figure, it's clear that MLE is closer to true value (10.0) than MOM.



From above figure, it's clear that MLE got smaller MSE than MOM, which MLE is a better estimator and has a smaller error.





From above two figures, we can see that estimated value of $MSE(L^*_{MOM})$ is larger than theoretical value while estimated value of $MSE(L^*_{MLE})$ is smaller than theoretical value.

6) You should have shown that $\hat{L}_{\mathrm{MLE}}$, while biased, has a smaller error over all. Why? The mathematical justification for it is above, but is there an explanation for this?

"Unbiased" is often misunderstood as "superior." This is correct only if the unbiased estimator also has a higher accuracy. But the overall error of the biased estimator is usually smaller than that of the unbiased estimator.

MOM only considers the characteristics of the data and ignores the distribution of the data, which means that it considers the mean and variance of the data but ignores the overall distribution of the data. Therefore, MOM will have less accuracy, that is, if there are some extreme values, the estimated variance of MOM will be greater, and it will not be "superior". Only when the data set is large enough can we make a "superior" estimate.

Although MLE is used to find the parameter with the greatest probability of occurrence, it will place the sample in a tightly distributed area. Therefore, the estimated variance of MLE will be smaller than that of MOM. Even if MLE is biased, the overall MSE of MLE will be smaller than the MSE of MOM.

7) Find $\mathbb{P}\left(\hat{L}_{\mathrm{MLE}} < L - \epsilon\right)$ as a function of L, ϵ, n. Estimate how many samples I would need to be sure that my estimate was within ϵ with probability at least δ.

Based on Markov Inequality, we have:

$$\mathrm{P}(L^*_{MLE} > L - \epsilon) \leq \frac{E[L^*_{MLE}]}{L - \epsilon}$$

Since we need:

$$\mathrm{P}(L^*_{MLE} > L - \epsilon) \geq \delta$$

$$\mathrm{P}(L^*_{MLE} > L - \epsilon) \leq \frac{E[L^*_{MLE}]}{L - \epsilon} = 1 - \delta$$

$$\frac{nL}{n+1} \times \frac{1}{L - \epsilon} = 1 - \delta, where\ 0 < \epsilon < L, 0 < \delta < 1$$

Therefore,

$$\frac{nL}{L - \epsilon} = (1 - \delta)n + (1 - \delta)$$

$$\left(\frac{L}{L - \epsilon} - 1 + \delta\right)n = 1 - \delta$$

$$n = \frac{(1 - \delta)(L - \epsilon)}{\delta L + \epsilon - \delta\epsilon}$$

We need at least $O\left(\frac{(1-\delta)(L-\epsilon)}{\delta L+\epsilon-\delta\epsilon}\right)$ samples to be sure that the estimate was within ϵ with probability at least δ.

8) Show that

$$\hat{L} = \left(\frac{n+1}{n}\right) \max_{i=1,\dots,n} X_i, \tag{4}$$

is an unbiased estimator, and has a smaller MSE still.

Since we have $E[L^*_{MLE}] = \int_0^L x f_L(x) dx = \frac{n}{L^n}\left[\frac{x^{n+1}}{n+1}\right]_0^L = \frac{nL}{n+1}$

$$E[L^*] = \frac{n+1}{n}\int_0^L x f_{L^*}(x) dx = \frac{n+1}{n} \times \frac{n}{L^n} \times \left[\frac{x^{n+1}}{n+1}\right]_0^L = L$$

$$bias(L^*) = L - E[L^*] = 0$$

Therefore, we can see that it's an unbiased estimator.

$$Var(L^*) = \left(\frac{n+1}{n}\right)^2 Var(L^*_{MLE}) = \left(\frac{n+1}{n}\right)^2 \frac{nL^2}{(n+1)^2(n+2)} = \frac{L^2}{n(n+2)}$$

$$MSE(L^*) = Var(L^*) = \frac{L^2}{n(n+2)}$$

We compare this to $MSE(L^*_{MOM})$

$$\lim_{n\to\infty}\left(\frac{MSE(L^*)}{MSE(L^*_{MOM})}\right) = \lim_{n\to\infty}\left(\frac{\frac{L^2}{n(n+2)}}{\frac{L^2}{3n}}\right) = \lim_{n\to\infty}\left(\frac{3}{n+2}\right) \to 0$$

Therefore, $\lim_{n\to\infty}\left(\frac{MSE(L^*)}{MSE(L^*_{MOM})}\right) < 1$, which means $MSE(L^*) < MSE(L^*_{MOM})$. L^* has a smaller MSE.

Source Code (Python 3)

UniformEstimators.py

```python
from random import uniform
from numpy import mean
from sklearn.metrics import mean_squared_error
```

```
import matplotlib.pyplot as plt

class Data:

    def __init__(self, n, L):
        self.n = n
        self.L = L
        self.data = [[uniform(0, self.L) for i in range(self.n)] for j in range(1000)]

class Estimator:

    def __init__(self, data):
        self.data = data.data
        self.n = data.n
        self.L = data.L

    def calcMOM(self, j):
        lStar = mean(self.data[j]) * 2
        lTrue = [self.L]
        lPred = [lStar]
        mseStar = mean_squared_error(lTrue, lPred)
        mseT = self.L * self.L / (3 * self.n)
        return lStar, mseStar, mseT

    def calcMLE(self, j):
        lStar = max(self.data[j])
        lTrue = [self.L]
        lPred = [lStar]
        mseStar = mean_squared_error(lTrue, lPred)
        mseT = self.L * self.L * (self.n - 1) * (self.n - 2) / (3 * self.n * (self.n + 1) * (self.n + 2))
        return lStar, mseStar, mseT


if __name__ == "__main__":
    n = 100
    L = 10
```

```
data = Data(n, L)
estimator = Estimator(data)
lMOM = []
mseMOM = []
mseTMOM = []
lMLE = []
mseMLE = []
mseTMLE = []
for j in range(1000):
    _lMOM, _mseMOM, _mseTMOM = estimator.calcMOM(j)
    _lMLE, _mseMLE, _mseTMLE = estimator.calcMLE(j)
    lMOM.append(_lMOM)
    mseMOM.append(_mseMOM)
    mseTMOM.append(_mseTMOM)
    lMLE.append(_lMLE)
    mseMLE.append(_mseMLE)
    mseTMLE.append(_mseTMLE)
xRange = list(range(1, 1001))
plt.clf()
plt.plot(xRange, lMOM, label="L*_MOM")
plt.plot(xRange, lMLE, label="L*_MLE")
plt.xlabel("j")
plt.ylabel("L*")
plt.title("L*_MOM and L*_MLE over 1000 iteration")
plt.legend()
fileName = "L*MOM_and_L*MLE_over_1000_iteration.png"
plt.savefig(fileName)
plt.clf()
plt.plot(xRange, mseMOM, label="MSE_MOM")
plt.plot(xRange, mseMLE, label="MSE_MLE")
plt.xlabel("j")
plt.ylabel("Maximum Likelihood Estimation")
plt.title("MSE_MOM and MSE_MLE over 1000 iteration")
plt.legend()
fileName = "MSE_MOM_and_MSE_MLE_over_1000_iteration.png"
plt.savefig(fileName)
```

8

```
plt.clf()
plt.plot(xRange, mseMOM, label="MSE_MOM")
plt.plot(xRange, mseTMOM, label="MSE_True_MOM")
plt.xlabel("j")
plt.ylabel("Maximum Likelihood Estimation")
plt.title("MSE_MOM and MSE_True_MOM over 1000 iteration")
plt.legend()
fileName = "MSE_MOM_and_MSE_True_MOM_over_1000_iteration.png"
plt.savefig(fileName)
plt.clf()
plt.plot(xRange, mseMLE, label="MSE_MLE")
plt.plot(xRange, mseTMLE, label="MSE_True_MLE")
plt.xlabel("j")
plt.ylabel("Maximum Likelihood Estimation")
plt.title("MSE_MLE and MSE_True_MLE over 1000 iteration")
plt.legend()
fileName = "MSE_MLE_and_MSE_True_MLE_over_1000_iteration.png"
plt.savefig(fileName)
```