STAT 391 Homework 5

Chongyi Xu
University of Washington
STAT 391 Spring 2018
chongyix@uw.edu

---

1. Problem 1 - Bias and variance for the Poisson distribution

a Calculate $E[\lambda^{ML}]$ as a function of λ.

$$E[\lambda^{ML}] = \lambda$$

b Calculate $Var(\lambda^{ML})$ as a function of λ.

$$Var(\lambda^{ML}) = \frac{\lambda}{n}$$

c Assume that n is large enough for CLT to apply. Express the probability that $\lambda^{ML} \geq \lambda + 1$ as a function of n, λ and ϕ the CDF of the standard normal. Let $Y = (x_1 + \cdots + x_n)$. Then $P(\lambda^{ML} \geq \lambda + 1) = P(Y \geq n\lambda + n)$.

$$\begin{aligned} P(\lambda^{ML} \geq \lambda + 1) &= P(Y \geq n\lambda + n) \\ &= 1 - P(Z \leq \frac{n\lambda + n - n\lambda}{\sqrt{n\lambda}}) \\ &= 1 - \phi(\sqrt{\frac{n}{\lambda}}) \end{aligned}$$

d Numerical answer for $\lambda = 10$, $n = 100$.

$$\begin{aligned} P(\lambda^{ML} \geq \lambda + 1) &= 1 - \phi(\sqrt{\frac{100}{10}}) \\ &= 1 - \phi(3.1622776601683795) \\ &= 1 - 0.99921 \approx 0.000790 \end{aligned}$$

2. Confidence Intervals and Boostrap

a Esimate μ^{ML} the mean of f.

```
import statistics as stat

dir =
    r'C:\Users\johnn\Documents\UW\SchoolWorks\2018Spring\STAT391\HW5'
f = open(dir+r'\hw5-data1.dat')
dat = [float(xx) for xx in f]

mu = stat.mean(dat)
print('mu = ', mu)
```

```
mu =  0.013597092102287756
```

b Estimate $\sigma^{2,ML}$ the Maximum Likelihood variance of f. Then calculate $\sigma^{2,C}$ the unbiased estimator of $Var(f)$.

Remark, $\sigma^{2,C} = \frac{n}{n-1}\sigma^{2,ML}$.

```
n = len(dat)
sigmaML = stat.variance(dat)/n
stdML = stat.stdev(dat)/n
sigmaC = n*sigmaML/(n-1)
print('s^2ML = ', sigmaML)
print('stdML = ', stdML)
print('s^2C = ', sigmaC)
ciC = (mu-2.576*math.sqrt(sigmaC/n),
    mu+2.576*math.sqrt(sigmaC/n))
print('99 confidence interval ', ciC)
```

```
s^2ML =  0.9962067496295647
stdML =  0.9981015728018691
s^2C =  0.9972039535831478
```

c Estimate the variance and standard deviation of μ^{ML}, pretending that $\sigma^{2,C}$ is the true variance $Var(f)$.

$$
\begin{aligned}
Var(\mu^{ML}) &= Var(\frac{\sum_i X_i}{n}) \\
&= \frac{1}{n^2} Var(\sum_i X_i) \quad = \frac{1}{n^2}\sum_{i=1}^{n} Var(X_i) \\
&= \frac{n\sigma^{2,C}}{n^2} \quad = \frac{\sigma^{2,C}}{n} = 0.0009972039535831479
\end{aligned}
$$

d Use the CLT approximation to obtain the Confidence Interval for confidence level $1-\delta$, for $delta = 0.01$.

$$\mu^{ML} \pm 2.576\sqrt{\frac{\sigma^{2,C}}{n}}$$

```
ciC = (mu-2.576*math.sqrt(sigmaC/n),
    mu+2.576*math.sqrt(sigmaC/n))
print('99 confidence interval ', ci)
```

```
99 confidence interval  (-0.06774921735482387, 0.09494340155939937)
```

e Now estimate the variance of μ^{ML} by Bootstrap; denote this by $\sigma^{2,B}$. Take $B = 1000$ bootstrap samples, and calculate from them the numberical value of $\sigma^{2,B}$.

Using the method of Bootstrap, resample with replacement from n data points , leading to new observations $X_1^*, \cdots, X_B^*$.

```
B=1000
np.random.seed(391)
xB = np.random.choice(dat, size=B, replace=True)
sigmaB = stat.variance(xB)
stdB = stat.stdev(xB)
print('s^2B = ', sigmaB/n)
```

```
s^2B =  0.0009465929207450022
```

f Use CLT approximation again to obtain the CI.

```
ciB = (mu-2.576*math.sqrt(sigmaB/n),
   mu+2.576*math.sqrt(sigmaB/n))
print('99 confidence interval ', ci)
```

```
99 confidence interval  (-0.06565805653330303, 0.09285224073787854)
```

The difference between CI^B and CI^C is

```
print('The difference is ', tuple(np.subtract(ciB, ciC)))
```

```
The difference is  (0.002091160821520832, -0.002091160821520832)
```

Calculate $e_r = \frac{|SD_C(\mu^{ML}) - \sigma^B|}{SD_C(\mu^{ML})}$.

```
print('e_r = ', abs(stdML-stdB)/stdML)
```

```
e_r =  0.02521938014674153
```

We can see that e_r is much smaller than 1, we can say the CI's are close.

3. **Problem 3- Median of Means(MOM)**

   a Compute μ^{ML} the mean of the data, and μ^{MOM} for $K = 56(n = 2800$ for this data set).

```
import statistics as stat
import numpy as np
import math

dir =
   r'C:\Users\johnn\Documents\UW\SchoolWorks\2018Spring\STAT391\HW5'
f = open(dir+r'\hw5-data2.dat')
dat = [float(xx) for xx in f]

muML = stat.mean(dat)
print('muML = ', stat.mean(dat))
```

```
def MOMhelper(dat, K):
    muk = [0.]*K
    m = int(len(dat)/K)
    for k in range(K):
        muk[k] = stat.mean(dat[k*m:(k+1)*m+1])
    return muk

print('muMOM = ', stat.median(MOMhelper(dat, 56)))
```

```
muML =  5.630414291829327
muMOM =  -0.06647177349707914
```

b Extract $B = 1000$ bootstrap samples, and compute $\mu^{MOM,b}$ for $b = 1 : B$. Then estimate the variance of μ^{MOM}, μ^{ML} by bootstrap.

```
B=1000
np.random.seed(391)
datB = np.random.choice(dat, size=B, replace=True)
mom = MOMhelper(datB)
print('muMLb = ', stat.mean(datB))
print('muMOMb = ', stat.mean(mom))
print('sigma(muML) = ', stat.variance(datB)/B)
print('sigma(muMOM) = ', stat.variance(mom)/B)
```

```
muMLb =  -3.4009646037169663
muMOMb =  18.18605627055922
sigma(muML) =  3126.105596362195
sigma(muMOM) =  251.92774632619899
```

This experiment agrees with the theory that μ^{MOM} is robust since we can see that it has less variance rather than μ^{ML}, which means it is less influenced than μ^{ML}.

c Repeat a,b for the data from the previous problem.
For hw5-dat1.dat, $n = 1000$, use $K = 20$, then there are 20 groups as we had in part(a),(b).

```
print('Considering file hw5-data1.dat')
```

```
f = open(dir+r'\hw5-data1.dat')
dat = [float(xx) for xx in f]

print('muML = ', stat.mean(dat))
print('muMOM = ', stat.median(MOMhelper(dat, 20)))

B=1000
datB = np.random.choice(dat, size=B, replace=True)
mom = MOMhelper(datB, 20)
print('muMLb = ', stat.mean(datB))
print('muMOMb = ', stat.mean(mom))
print('var(muML) = ', stat.variance(datB)/B)
print('var(muMOM) = ', stat.variance(mom)/B)
```

```
Considering file hw5-data1.dat
muML =  0.013597092102287756
muMOM =  0.02477232801920145
muMLb =  0.01116517673353682
muMOMb =  0.006316025344143354
var(muML) =  0.0009170493279787797
var(muMOM) =  4.685839582870582e-05
```

We can see that for hw5-data1.dat, the MOM estimation is really close to ML estimation. The reason might be that for hw5-data2.dat, there are too many outliers that influences μ^{ML}.