Statistical Methods for Data Science
CS 6313.001: Mini Project #3

Due on Thursday March 7, 2019 at 10am

Instructor: Prof. Min Chen

Shyam Patharla **(sxp178231)**

Contents

Section 1 Answers

Problem 1.1

(a) Compute the mean squared error of an estimator using Monte Carlo simulation

Given: n, θ, nsim

1. First, we generate a sample of size n from the uniform distribution in the interval **(0,** θ).
2. We replicate Step 1 for **nsim** times and store the values of the nsim samples in a vector called **data**.
3. To get the maximum likelihood estimators for each of these nsim samples, we take the mean of each column in the vector **data**. We store this in a vector **mle.est**.

```
>  mle.est <- apply(data,2,max)
```

4. We get the estimator mean squared error using the formula:

```
> result1 <- sum((mle.est-theta)^2)/rep
```

5. To get the method of moments estimators for each of these nsim samples, we take the mean of each row in the vector **data** and multiply it by 2. We store this in a vector **mom.est**.

```
>  mom.est <- apply(data,2,max)
```

6. We get the estimator mean squared error using the formula:

```
> result1 <- sum((mom.est-theta)^2)/rep
```

(b) For a given combination(n, theta), compute the mean squared errors of estimators obtained using maximum likelihood estimation and method of moments

```
# Let n=5, theta=50,rep=1000
> compare.est(5,50,1000)
# [1] 120.4905 674.0310
```

The first element of the result array is the mean squared error for the estimator obtained using maximum likelihood estimation. The second is for the one obtained using the method of moments.

The error for the estimator obtained using method of moments is significantly higher compared to its MLE counterpart.

(c) Repeat part (b) for the remaining combinations of (n, theta) and summarize your results graphically

	$\theta = 1$	$\theta = 5$	$\theta = 50$	$\theta = 100$
n=1	0.339195	7.703114	811.1820	3311.070
n=2	0.1621363	4.434155	419.0197	1784.770
n=3	0.09695403	2.431108	262.1221	921.4235
n=5	0.05210957	1.160537	120.4905	462.295
n=10	0.01460649	0.4167772	35.15594	155.2607
n=30	0.001593959	0.04953917	4.541694	20.21065

Table 1: Mean Squared Error for Maximum Likelihood Estimator

	$\theta = 1$	$\theta = 5$	$\theta = 50$	$\theta = 100$
n=1	0.320539	7.841863	816.5786	3265.399
n=2	0.2893060	7.506875	734.2253	3041.526
n=3	0.27582063	6.867733	689.8118	2687.4609
n=5	0.27065755	6.642441	674.0310	2630.757
n=10	0.25359251	6.5436014	641.09682	2572.9569
n=30	0.252542160	6.23571907	634.189762	2516.58880

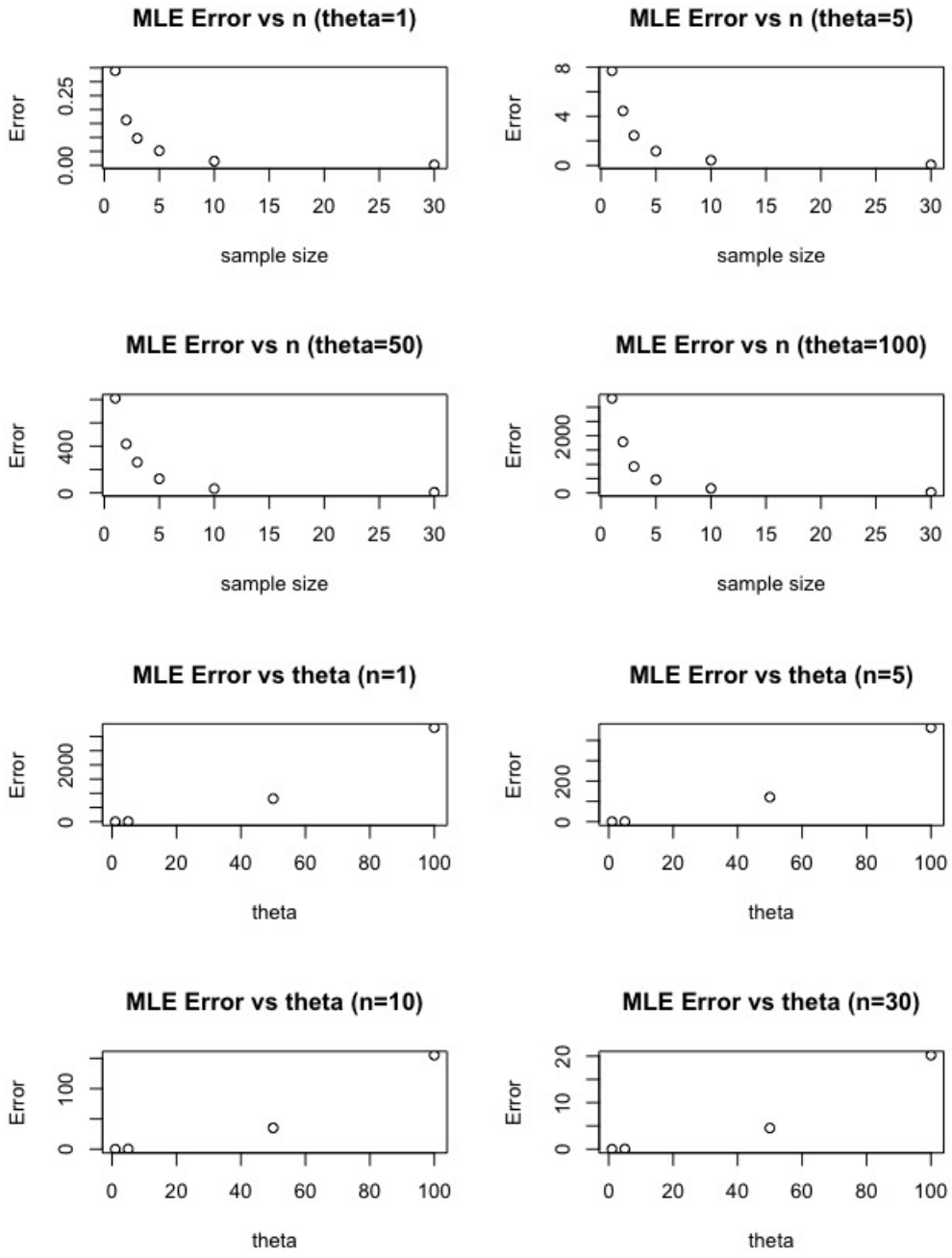
Table 2: Mean Squared Error for Method of Moments Estimator

Conclusions:

1. For a fixed value of n, the mean squared error increases with θ for both MLE and MOM estimators.

2. For a fixed value of θ, the mean squared error decreases with n for both MLE and MOM estimators.

3. For a given (n,θ),the error for the estimator obtained using method of moments is significantly higher compared to its MLE counterpart.



Similar characteristics exist for the method of moments estimator.

(d) Which estimator is better?

The **maximum likelihood estimator** is clearly better for our problem irrespective of n or θ because the errors are less than thos of there Method of Moments counterparts for all (n, θ).

Problem 1.2

(a) Derive an expression for the maximum likelihood estimator of theta

We have:

$$f(x) = \begin{cases} \frac{\theta}{x^{\theta+1}}, x \geq 1 \\ 0, x < 1 \end{cases} \tag{1}$$

$$lnf(X) = lnf(X_1, X_2, ...X_n) = ln \prod_{i=1}^{n} f(X_i) \tag{2}$$

$$lnf(X) = \sum_{i=1}^{n} lnf(X_i) \tag{3}$$

$$lnf(X) = \sum_{i=1}^{n} ln(\frac{\theta}{x^{\theta+1}}) \tag{4}$$

$$lnf(X) = \sum_{i=1}^{n} (ln(\theta) - (\theta + 1)ln(X_i)) \tag{5}$$

$$lnf(X) = nln(\theta) - \theta \sum_{i=1}^{n} ln(X_i) - \sum_{i=1}^{n} ln(X_i) \tag{6}$$

Taking the derivative and equating it to 0, we get:

$$\frac{\partial lnf(X)}{\partial \theta} = 0 \tag{7}$$

$$\frac{n}{\hat{\theta}} - \sum_{i=1}^{n} ln(X_i) = 0 \tag{8}$$

$$\hat{\theta} = \frac{n}{\sum_{i=1}^{n} lnf(X_i)} \tag{9}$$

(b) Calculate an estimate for theta for the given data

We have:
$X_1 = 21.72, X_2 = 14.65, X_3 = 50.42$
$X_4 = 28.78, X_5 = 11.23$

$n = 5$
Substituting X_1, X_2, X_3, X_4, X_5 and n, we get:

$$\hat{\theta} = 0.323394 \tag{10}$$

(c) Obtain the estimate by numerically maximizing log-likelihood function in R

We first write the negative log likelihood function:

```
neg.loglik.fun <- function(par,dat)
{
  result <- NROW(dat)*log(par) - (par+1)*sum(log(dat))
  return(-1*result)
}
```

We then run the **optim** function to find the value of θ where the negative log likelihood function has the minimum value:

```
> ml.est <- optim(par=c(0.1), fn=neg.loglik.fun,
method = "L-BFGS-B",lower=rep(0.005), hessian=TRUE, dat=data)

> ml.est
$par
[1] 0.3233885

$value
[1] 26.10585

$counts
function gradient
       9        9

$convergence
[1] 0

$message
[1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"

$hessian
         [,1]
[1,] 47.81116
```

The value of 0.3233885 obtained is very close to the manually calculated value of $\hat{\theta} = 0.3234$.

(d) Provide a standard error and a 95 percent confidence interval for theta

The standard error is given by:

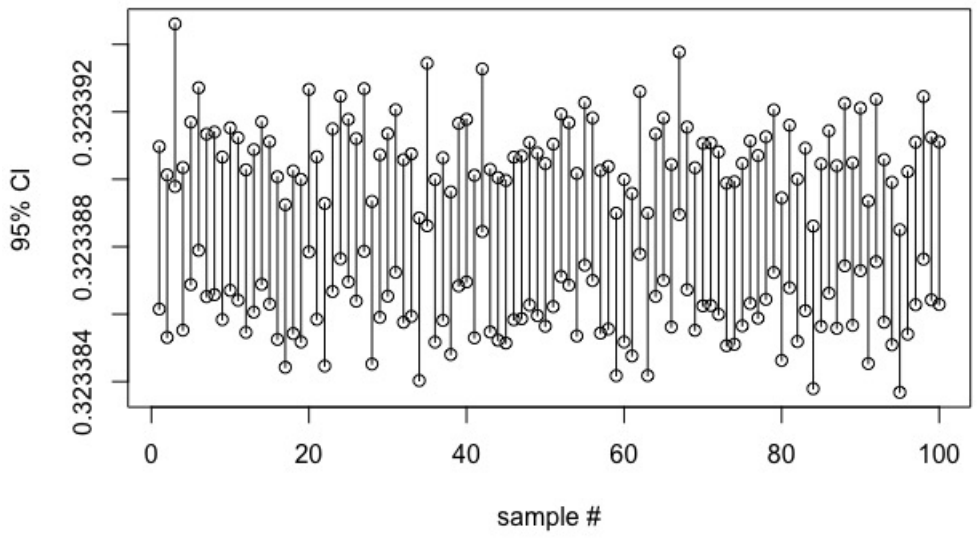
```
# Calculating the standard error
> (0.3233885-0.323394)^2
# [1] 3.025e-11

> sqrt(3.025e-11)
# [1] 5.5e-06
```

```
> conf.int(0.3233885,5.5e-06,20,0.05)
# [1] 0.3233860 0.3233908
```

We repeat the process to get many confidence intervals.

```
> nsim<-1000
> ci.mat<-replicate(1000,conf.int(0.3233885,5.5e-06,20,0.05))
```



The confidence intervals seem to give us a good idea of where θ lies.

Section 2 R Code

```
################################################
# R Code for Question 1
################################################


compare.est<-function(n,theta,nsim)
{
  # Getting the data
  data <- replicate(nsim,runif(n,0,theta))

  # Maximum likelihood estimation
  # Generate a vector of length n from the uniform distribution in the
     interval (0,theta). Take
  # the mean. Replicate for rep times
  if(NCOL(data)!=1)
    {
    mle.est <- apply(data,2,max)
  }
  else
  {
    mle.est<-data
  }

  # Find the mean squared error
  result1 <- sum((mle.est-theta)^2)/nsim



  # Method of moments
  # Generate a vector from the uniform distribution in the interval (0,theta).
      Take
  # the mean times 2. Replicate for rep times
  if(NCOL(data)!=1)
  {
    mom.est <- apply(data,2,mean)
  }
  else
  {
    mom.est<-2*data
  }
  # Find the mean squared error
  result2 <- sum((mom.est-theta)^2)/nsim


  # Return both the result
  return(c(result1,result2))
```

```
}

# Part (b):
# Simulate for a given combination (n,theta)
# Let n=5, theta=50,rep=1000

# > compare.est(5,50,1000)
# [1] 117.6610 663.7483

# Part(c)
# Repeat the above step for all remaining combinations of (n,theta)

# > compare.est(1,1,1000)
# [1] 0.339195 0.320539

# > compare.est(1,5,1000)
# [1] 7.703114 7.841863

# > compare.est(1,50,1000)
# [1] 811.1820 816.5786

# > compare.est(1,100,1000)
# [1] 3311.070 3265.399

# > compare.est(2,1,1000)
# [1] 0.1621363 0.2893060

# > compare.est(2,5,1000)
# [1] 4.434155 7.506875

# > compare.est(2,50,1000)
# [1] 419.0197 734.2253

# > compare.est(2,100,1000)
# [1] 1784.770 3041.526

# > compare.est(3,1,1000)
# [1] 0.09695403 0.27582063

# > compare.est(3,5,1000)
# [1] 2.431108 6.867733

# > compare.est(3,50,1000)
# [1] 262.1221 689.8118

# > compare.est(3,100,1000)
# [1] 921.4235 2687.4609

# > compare.est(5,1,1000)
```

```
# [1] 0.05210957 0.27065755

# > compare.est(5,5,1000)
# [1] 1.160537 6.642441

# > compare.est(5,50,1000)
# [1] 120.4905 674.0310

# > compare.est(5,100,1000)
# [1] 462.295 2630.757

# > compare.est(10,1,1000)
# [1] 0.01460649 0.25359251

# > compare.est(10,5,1000)
# [1] 0.4167772 6.5436014

# > compare.est(10,50,1000)
# [1]  35.15594 641.09682

# > compare.est(10,100,1000)
# [1] 155.2607 2572.9569

# > compare.est(30,1,1000)
# [1] 0.001593959 0.252542160

# > compare.est(30,5,1000)
# [1] 0.04953917 6.23571907

# > compare.est(30,50,1000)
# [1] 4.541694 634.189762

# > compare.est(30,100,1000)
# [1] 20.21065 2516.58880


# Plotting graphs for error of MLE estimator vs sample size
par(mfrow=c(2,2))

n<-c(1,2,3,5,10,30)

t1<-c(0.339195,0.1621363,0.09695403,0.05210957,0.01460649,0.001593959)
t5<-c(7.703114,4.434155,2.431108,1.160537,0.4167772,0.04953917)
t50<-c(811.1820,419.0197,262.1221,120.4905,35.15594,4.5416940)
t100<-c(3311.070,1784.770,921.4235,462.295,155.2607,20.21065)

plot(n,t1,xlab="sample size", ylab="Error", main="MLE Error vs n (theta=1)" )
plot(n,t5 ,xlab="sample size",ylab="Error", main="MLE Error vs n (theta=5)")
plot(n,t50 ,xlab="sample size",ylab="Error", main="MLE Error vs n (theta=50)")
```

```
plot(n,t100 ,xlab="sample size",ylab="Error", main="MLE Error vs n (theta=100)
    ")



# Plotting graphs for error of MLE estimator vs theta
par(mfrow=c(2,2))
t<-c(1,5,50,100)

n1<-c(0.339195,7.703114,811.1820,3311.070)
n5<-c(0.05210957,1.160537,120.4905,462.295)
n10<-c(0.01460649,0.4167772,35.15594,155.2607)
n30<-c(0.001593959,0.04953917,4.541694,20.21065)

plot(t,n1 ,xlab="theta",ylab="Error", main="MLE Error vs theta (n=1)")
plot(t,n5 ,xlab="theta",ylab="Error", main="MLE Error vs theta (n=5)")
plot(t,n10 ,xlab="theta",ylab="Error", main="MLE Error vs theta (n=10)")
plot(t,n30 ,xlab="theta",ylab="Error", main="MLE Error vs theta (n=30)")
```

```
################################################
# R Code for Question 2
################################################

# Getting the data
data <- c(21.72,14.65,50.42,28.78,11.23)

# Negative of log-likelihood function
neg.loglik.fun <- function(par,dat)
{ c
  result <- NROW(dat)*log(par) - (par+1)*sum(log(dat))
  return(-1 * result)
}



# Minimize -log (L), i.e., maximize log (L)
ml.est <- optim(par = c(0.1), fn = neg.loglik.fun, method = "L-BFGS-B",lower =
    rep(0.005),
               hessian = TRUE, dat = data)

# Estimating the mean squared error
calc = 0.323394
(ml.est$par-0.323394)^2

# Estimating the standard error
(0.3233885-0.323394)^2
# [1] 3.025e-11

sqrt(3.025e-11)
# [1] 5.5e-06
```

```
# Finding the confidence interval
conf.int<-function(mu,sigma,n,alpha)
{
  x <- rnorm(n,mu,sigma)
  ci <- mean(x) + c(-1,1) * qnorm(1-alpha/2) * sigma/sqrt(n)
  return(ci)
}


# Get one confidence interval
conf.int(0.3233885,5.5e-06,20,0.05)


# Repeat the simulation for nsim times
nsim <- 1000
ci.mat <- replicate(1000,
                  conf.int(0.3233885,5.5e-06,20,0.05))


# Plot the confidence intervals
plot(1:100, ci.mat[1, 1:100],
    ylim=c(min(ci.mat[,1:100]), max(ci.mat[,1:100])),
     xlab="sample #", ylab="95% CI", type="p")
points(1:100, ci.mat[2, 1:100])
for (i in 1:100) {
  segments(i, ci.mat[1, i], i, ci.mat[2,i], lty=1)
}
abline(h=5, lty=2)
```