# F79MA 2018-19: Assessed Project 1 Report

## Introduction and Summary

This project is aim to look at parameter estimation for an inverse gamma model, which can be considered as a survival distribution. Also, we will show that computer simulation can be a very useful tool to characterize the performance of statistical estimators.

## Analysis

The probability density function of inverse distribution is

where is a shape parameter, is a scale parameter, and is the gamma function.

Let be a random sample from an inverse gamma distribution with shape parameter and unknown scale parameter .

### Task 1

*As*  is i.i.d, the joint density function is

 (1.1)

Then

Differentiating with respect to , we get the likelihood equation

Differentiating again, we get

Which is negative for all , and any n.

Solving equation (1.3) for , we get

So is the maximum likelihood estimator for .

The inverse gamma density’s three properties:

1) for, , , , for, they all exist.

2) for , Where are both integrable, and satisfies M is independent from

3) for , .

### Task 2

Let denote the true value of , then the bias of is

.

Let , take as a function of , and conduct Taylor expansions at , then

 (2.1)

Where , summation (2.1) for

Then we get

Where,

As all are arithmetic mean of n iid variables. as , differentiating with twice, then we have

Then,

For , according to Big Data Theorem, as 0<, and , we can find that only relates with , when , the following inequalities hold.

Now, define the event

When , we have . As the unique MLE, so when event B occurs, so when , we have for any small . Here we have proved that the bias vanishes as n .

### Task 3

If , then .

Let

 So, we can use Gamma distribution to produce random sample of inverse Gamma distribution. Then we plot the mean square error of with sample size n as shown in fig 1. We can see MSE deceases with the increase of n.



Fig-1 Mean square error and sample size

### Task 4

Fisher information

As shown in Task 2, we can know that

Cramer-Rao lower bound on is .

### Task 5

Set , and we can calculate the relative Cramer-Rao lower bound on

using R, we can find that MSE are always bigger than Cramer-Rao lower bound (shown in Fig2).



Fig 2 the difference between MSE and C-R bound

### Task 6

Let denote the as moment estimator.

As

 , given

So, , where

Both maximum likelihood estimator (MLE) and Moment estimator (MME) are asymptotic unbiased and consistent. Compute MSE of these two estimators in R (shown in table1), we can find that moment estimator doesn’t produce better MSE result than maximum likelihood estimator (MLE).

|  |  |  |
| --- | --- | --- |
| 　 | MLE | MME |
| beta | n=20 | n=40 | n=60 | n=20 | n=40 | n=60 |
| 0.3 | 0.0006 | 0.0003 | 0.0001 | 0.0178 | 0.0114 | 0.0057 |
| 0.5 | 0.0012 | 0.0006 | 0.0005 | 0.0869 | 0.0242 | 0.0198 |
| 0.8 | 0.0029 | 0.002 | 0.0009 | 0.129 | 0.0633 | 0.0434 |
| 1 | 0.0058 | 0.0022 | 0.0014 | 0.1472 | 0.1014 | 0.0732 |
| 1.2 | 0.0071 | 0.0042 | 0.0019 | 0.3258 | 0.1737 | 0.0938 |

Table 1 MSE of MME and MLE

### Task 7

As inverse gamma distribution satisfies properties shown in task1, and MLE is the unique root, then we have

So, the approximate 95% confidence interval for is

The approximate error is shown in table 2.

|  |  |
| --- | --- |
| 　 | MLE |
| beta | n=20 | n=40 | n=60 |
| 0.3 | 0.0203 | 0.0151 | 0.0108 |
| 0.5 | 0.0354 | 0.0268 | 0.0205 |
| 0.8 | 0.0645 | 0.039 | 0.0294 |
| 1 | 0.0675 | 0.0582 | 0.0376 |
| 1.2 | 0.0928 | 0.0555 | 0.0522 |

Table 2 approximate error: standard deviation of MLE

## Appendix

### Task 3’s R code

rm(list=ls(all=TRUE))

library('invgamma')

alpha<-10

beta<-1 #true value

ntime<-1000 #simulation runs

nmax<-50 #sample sizes

betahat<-rep(0,ntime)

vac<-rep(0,nmax)

MSE<-rep(0,nmax)

for (n in 1:nmax){

 # repeat times

 for (t in 1:ntime){

 Y<-rinvgamma(n,alpha,rate=1)

 betahat[t]<-alpha/mean(1/Y)

 }

 vac[n]<-var(betahat)

 MSE[n]<-1/ntime\*sum(betahat-beta)^2

}

n<-c(1:50)

plot(n,MSE,main='MSE vs n')

### Task 5’s R code

crbound<-beta^2/alpha/n^2

plot(MSE-crbound)

abline(h=0,col='red')

### Task 6’s R code

betas<-c(0.3,0.5,0.8,1,1.2)

MSEMME<-matrix(0,nrow=5,ncol=3)

MSEMLE<-matrix(0,nrow=5,ncol=3)

ns<-c(20,40,60)

for (i in 1:5){

 beta<-betas[i]

 for (j in 1:3){

 n<-ns[j]

 mme<-rep(0,100)

 mle<-rep(0,100)

 for (t in 1:100){

 sp<- rinvgamma(n,alpha,beta)

 mme[t]<-mean(sp)\*((mean(sp))^2/var(sp)+1)

 mle[t]<-alpha/mean(1/sp)

 }

 MSEMME[i,j]<-1/100\*sum((mme-beta)^2)

 MSEMLE[i,j]<-1/100\*sum((mle-beta)^2)

 }

}

round(MSEMME,4)

round(MSEMLE,4)

### Task 7’s R code

betas<-c(0.3,0.5,0.8,1,1.2)

AVsMLE<-matrix(0,nrow=5,ncol=3)

ns<-c(20,40,60)

for (i in 1:5){

 beta<-betas[i]

 for (j in 1:3){

 n<-ns[j]

 mle<-rep(0,100)

 for (t in 1:100){

 sp<- rinvgamma(n,alpha,beta)

 mle[t]<-alpha/mean(1/sp)

 }

 AVsMLE[i,j]<-sd(mle)

 }

}

round(AVsMLE,4)