< STAT-5361 > HW#6

Hee-Seung, Kim

Oct 19, 2018

Contents

1 Exercises 5.2.1

1.1 (a)

As you can see following result, the g(x) can be expressed as a mixture of two Gamma distributions with different weights. At first, we use that the integral result should be 1 for whole range;

$$g(x) = c(2x^{\theta-1} + x^{\theta-\frac{1}{2}})e^{-x}$$

$$c\int_0^\infty (2x^{\theta-1} + x^{\theta-\frac{1}{2}})e^{-x}dx = c[2\int_0^\infty x^{\theta-1}e^{-x}dx + \int_0^\infty x^{\theta-\frac{1}{2}}e^{-x}dx] = 1$$

Since $\Gamma(z) = \int_0^\infty x^{z-1}e^{-x}dx$ <- definition of Gamma function

$$1)2\int_0^\infty x^{\theta-1}e^{-x}dx = 2\Gamma(\theta)$$
$$2)\int_0^\infty x^{\theta-\frac{1}{2}}e^{-x}dx = \int_0^\infty x^{(\theta+\frac{1}{2})-1}e^{-x}dx = \Gamma(\theta+\frac{1}{2})$$
$$\therefore 2\int_0^\infty x^{\theta-1}e^{-x}dx + \int_0^\infty x^{\theta-\frac{1}{2}}e^{-x}dx = 2\Gamma(\theta) + \Gamma(\theta+\frac{1}{2}) = \frac{1}{c}$$
$$c = \frac{1}{2\Gamma(\theta)+\Gamma(\theta+\frac{1}{2})}$$

in conclusion,

$$g(x) = c(2x^{\theta-1} + x^{\theta-\frac{1}{2}})e^{-x} = \frac{1}{2\Gamma(\theta)+\Gamma(\theta+\frac{1}{2})}(2x^{\theta-1} + x^{\theta-\frac{1}{2}})e^{-x}$$
$$= \frac{2\Gamma(\theta)}{2\Gamma(\theta)+\Gamma(\theta+\frac{1}{2})}(\frac{x^{\theta-1}e^{-x}}{\Gamma(\theta)}) + \frac{\Gamma(\theta+\frac{1}{2})}{2\Gamma(\theta)+\Gamma(\theta+\frac{1}{2})}(\frac{x^{(\theta+\frac{1}{2})-1}e^{-x}}{\Gamma(\theta+\frac{1}{2})})$$

1.2 (b)

this function requires two parameters (N: desired number of sample, θ: Gamma scale)

STEP 0. Prepare slots for the sample X <- rep(0,N);

STEP 1. Calculate the weight of each Gamma distribution (π 1,π 2) π 1 <- 2 $\Gamma(\theta)$/(2 $\Gamma(\theta)$+$\Gamma(\theta$+0.5))

STEP 2. For given N times,draw a Gamma observation according to the chosen index for (i in 1:N) { if (runif(1) > π 1) { X[i] <- rΓ(1,shape=θ,scale=1)} else { X[i] <- rΓ(1,shape=θ+0.5,scale=1) } }

STEP 3. Return generated values return(X);

```
rs1b <- function(N,theta) {

  X <- rep(0,N);
  pi1 <- 2*gamma(theta)/(2*gamma(theta)+gamma(theta+0.5))
  pi2 <- 1-pi1
```

```r
  for (i in 1:N) {
    if (runif(1) > pi1) { X[i] <- rgamma(1,shape=theta,scale=1)}
    else { X[i] <- rgamma(1,shape=theta+0.5,scale=1)  }
  }
  return(X);
}
rs1b(100,3)
```

```
##   [1] 1.7418791 2.6113442 1.9791542 4.8028047 3.4618053 3.0350260 2.2892210
##   [8] 3.1713735 2.0323238 0.6804200 2.9914424 2.6402017 0.5120304 4.7070255
##  [15] 0.9155099 4.1764744 3.3241502 3.0225825 4.1432065 1.1850393 3.5684914
##  [22] 5.9716897 3.5886659 4.1290651 1.2998803 2.5019150 2.1877165 2.6158671
##  [29] 5.5653241 3.7540501 6.3399115 2.6989973 5.6160615 6.9225070 4.8647924
##  [36] 1.1247687 2.5687902 1.9531727 1.5495930 3.2249099 1.9442201 1.0158576
##  [43] 2.0953008 0.7487865 3.2381820 2.3504122 6.9918843 1.9848389 5.3835203
##  [50] 2.6044068 1.0935249 3.6789729 7.1732656 8.0518496 2.4190679 0.3135262
##  [57] 0.7040286 1.9220885 3.1590284 2.5113989 6.6468957 1.2299248 2.6721625
##  [64] 4.2283976 5.4828750 7.8137333 3.1710143 7.2915583 0.8442875 2.6770463
##  [71] 1.7999961 2.0087720 7.1078180 3.9068681 3.6683176 2.9480998 2.9411289
##  [78] 5.0202582 4.4923198 2.7845972 3.3455776 3.9213880 2.5691217 6.9672954
##  [85] 4.0607839 2.2516038 1.9557513 1.4909107 3.5826493 1.9376051 4.2310862
##  [92] 4.2434095 3.7432813 5.1867233 3.8485351 4.0322448 3.2634425 1.8800554
##  [99] 1.5254833 2.5805980
```

1.3 (c)

this function requires two parameters (N: desired number of sample, θ: Gamma scale)

STEP 0. Prepare slots for the sample X <- rep(0,N);

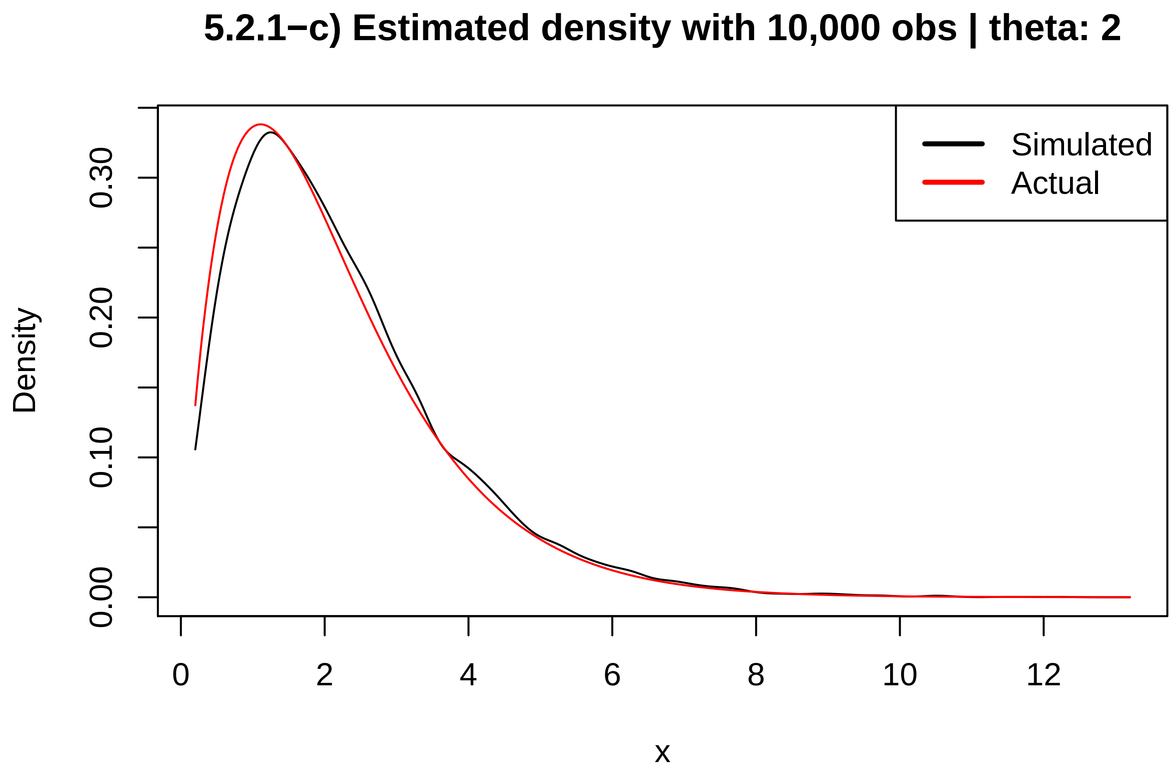
STEP 1. Draw a sample from g,using the rs1b function defined above. while (1) { Y = rs1b(1,θ);

STEP 2. If randomly generated uniform number exceed given thresh-hold, then sample Y again,o.w,retain Y. if (runif(1) $<$ sqrt(4+Y)/(sqrt(Y)+2)) { X[i] = Y; break; }

STEP 3. Return generated values return(X);

```r
rs1c <- function(N,theta) {

  X <- rep(0,N);

  for (i in 1:N) {
    while (1) {
      Y = rs1b(1, theta);
      if (runif(1) < sqrt(4+Y)/(sqrt(Y)+2)) {
        X[i] = Y; break;
      }
    }
```

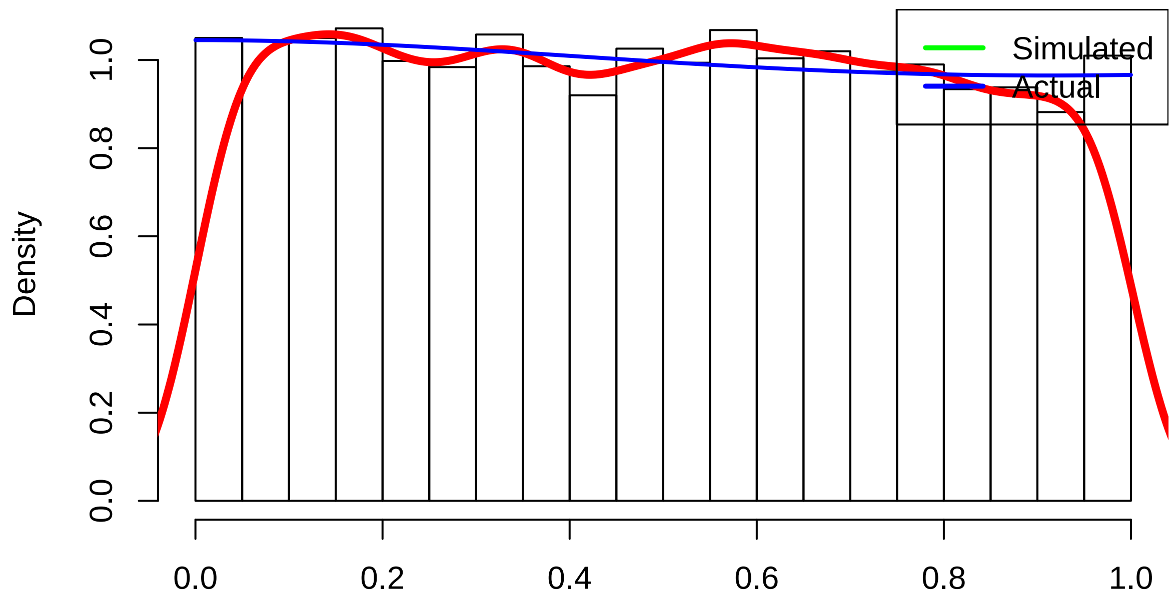
```
  }
  return(X);
}

theta <- 2

sample1c <-rs1c(10000,theta)

## Estimate the density and compare it to q(x) to see if they are
## proportional to each other
## Specify on which interval the density function is to be estimated
intvl=c(.2, floor(max(sample1c)+1)+.2);

## Specify the number of points in the interval on which density is
## to be estimated.  The points are equally spaced in the interval,
## with the first one and last one being the end points of the interval
n.p=1+(intvl[2]-intvl[1])/0.02;

## Estimate the density.  The returned is a
df.obj = density(sample1c, from=intvl[1], to=intvl[2], n=n.p)
x = df.obj$x;
df = df.obj$y;

## Calculate the value of q(x)
q.val = (4+x)^0.5 * x^(theta-1)*exp(-x);

## Make the two on the same scale
q.val = q.val*mean(df)/mean(q.val);

plot(ts(cbind(df, q.val), start=intvl[1], deltat=diff(intvl)/(n.p-1)),
     plot.type="single", col=c("black", "red"),
     ylab="Density", xlab="x",
     main = paste("5.2.1-c) Estimated density with 10,000 obs | theta:", theta));

legend('topright', legend=c("Simulated","Actual"),lty=c(1,1),
       lwd=c(2.5,2.5),col=c('black','red'))
```


5.2.1−c) Estimated density with 10,000 obs | theta: 2
Simulated
Actual
Density
0.00
0.10
0.20
0.30
0
2
4
6
8
10
12
x

2 Exercises 5.2.2

2.1 (a)

```
rsa = function(n, theta, beta)
{
   ho = runif(n)
   number = sum( ho < beta(theta, 1)/( 2*beta(1, beta) + beta(theta, 1) ) )
   data = c(rbeta(number, theta, 1), rbeta(n-number, 1, beta))
   return(data)
}
algorithm3 = function(n, theta, beta)
{
   data = c()
   while( length(data) <= n )
   {
      ho = runif(1); x = rsa(1, theta, beta)
      q = (x^(theta - 1)/(1+x^2)) + sqrt(2+x^2)*(1-x)^(beta-1)
      g = x^(theta -1) + 2*(1-x)^(beta-1)
      if(ho <= q/g)
         data= c(data, x)
   }
   return(data)
}
theta = 1
beta = 1
integrand = function(x)
{
   (x^(theta - 1)/(1+x^2)) + sqrt(2+x^2)*(1-x)^(beta-1)
}
integral = integrate(integrand, 0, 1)
aa = algorithm3(10000, theta, beta)
hist(aa,freq=F,xlab='',main='')
lines(density(aa),col="red",lwd=4)
curve( ((x^(theta - 1)/(1+x^2)) + sqrt(2+x^2)*(1-x)^(beta-1))/as.numeric(integral[1]),
       add = T,col="blue",lwd=2)
legend('topright', legend=c("Simulated","Actual"),lty=c(1,1),
       lwd=c(2.5,2.5),col=c('green','blue'))
```



2.2 (b)

```
rsb = function(N, theta, beta)
{
  data = c()
  while(length(data) <= N )
  {
    d =sample(c(0,1),1)
    u = runif(1);
    if(d == 0)
    {
      ho = rbeta(1,theta, 1)
      q = (ho^(theta - 1)/(1+ho^2))
      g = ho^(theta -1)
      if(u <= q/g)
        data= c(data, ho)
    }
    else {
      x = rbeta(1, 1, beta)
      q = sqrt(2+ho^2)*(1-ho)^(beta-1)
      g = 2*(1-ho)^(beta-1)
```

```
      if(u <= q/g)
        data= c(data, ho)
    }
  }
  return(data)
}

theta = 1
beta = 1

Func_ = function(x)
{
  (x^(theta - 1)/(1+x^2))+sqrt(2+x^2)*(1-x)^(beta-1)
}
integral = integrate(Func_, 0, 1)
tem = rsb(10000, theta, beta)
hist(tem,freq=F,xlab='',main='')
lines(density(tem),col="red",lwd=4)
curve( ((x^(theta - 1)/(1+x^2)) + sqrt(2+x^2)*(1-x)^(beta-1))/as.numeric(integral[1]),
       add = T,col="blue",lwd=2)
legend('topright', legend=c("Simulated","Actual"),lty=c(1,1),
       lwd=c(2.5,2.5),col=c('green','blue'))
```