Bayesian Econometrics: Homework 3

Jerin Varghese (Group: Rebecca Rockey, Roman Maranets, Dariene Castro)

February 18, 2019

Contents

Problem 1

a) Show that when U is uniformly distributed on (0,1), 1-U is also uniformly distributed on (0,1). A random variable, U, is uniformly distributed if $P(U \leq x) = x$.

Answer

$$P(1-U \leq x) = P(U \geq 1-x) = 1 - P(U \leq 1-x) = 1-(1-x) = x$$

Thus, 1-U is also uniformly distributed.

b) Using inverse distribution function method, show that $Y = -\frac{1}{\theta}\log U$, where U is uniform random variable on (0,1), is exponentially distributed and use the fact to draw a sample from it given a uniform sample.

Answer

$$\begin{aligned} P(Y \leq y) =& P(-\frac{1}{\theta}\log U \leq y) \\ =& P(U \geq e^{-\theta y}) \\ =& 1 - P(U \leq e^{-\theta y}) \\ =& 1 - e^{-\theta y} \end{aligned}$$

The last line is the CDF of the exponential distribution.

Given the above result, we can indirectly sample from the exponential distribution with the following code:

```r
set.seed(42)
n <- 100
theta <- 0.5
y <- (-1/theta)*log(runif(n))
```

Problem 2

It is sometimes possible to write a probability density function in the form of a finite mixture distribution, which is defined as follows:

$$f(x) = \sum_{i=1}^{k} P_i f_i(x)$$

where $\sum_{i=1}^{k} P_i = 1$. For example, finite mixtures of normal distributions with different means and variances can display a wide variety of shapes. Generate samples from the following distribution:

$$f(x) = 1.8e^{-3x} + 0.4e^{-x}$$

a) Show first that the above density is a finite mixture of exponential distributions.

Answer

$$f(x) = P_1\lambda_1 e^{-\lambda_1 x} + P_2\lambda_2 e^{-\lambda_2 x} + \dots P_k\lambda_k e^{-\lambda_k x}$$

If k=2, then we can express the mixture as follows:

$$f(x) = P_1\lambda_1 e^{-\lambda_1 x} + (1 - P_1)\lambda_2 e^{-\lambda_2 x}$$

Let $P_1 = 0.6, \lambda_1 = 3, \text{and} \lambda_2 = 1$. Plug these into the above function and you get:

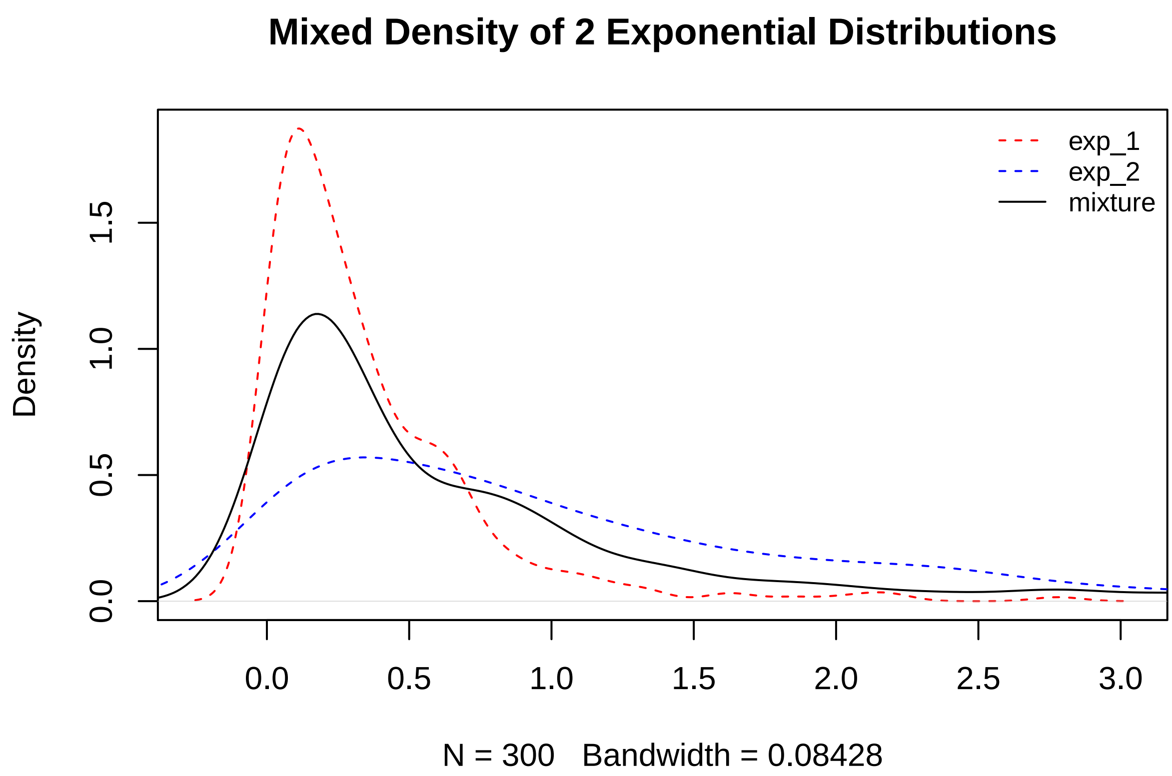
$$f(x) = (0.6)(3)e^{-3x} + (1 - 0.6)(1)e^{-1x} = 1.8e^{-3x} + 0.4e^{-x}$$

b) Draw 300 sample from the above mixture density by a) drawing each exponential distribution (e.g., see problem 1 for sampling from exponential distribution) and b) using the probability of drawing each exponential distribution (i.e., $P_i^{\prime}s$ above).

```r
set.seed(42)
n <- 300
p_1 <- 0.6
p_2 <- 1- p_1
lambda_1 <- 3
lambda_2 <- 1

# a)
exp_1 <- (-1/lambda_1)*log(runif(n))
exp_2 <- (-1/lambda_2)*log(runif(n))

# b)
components <- sample(1:2,prob=c(p_1,p_2),size=n,replace=TRUE)
lambdas <- c(lambda_1,lambda_2)

mix_exp <- (-1/lambdas[components])*log(runif(n))
```

c) Compare densities (or histograms) of all three distributions (two exponential distributions and the mixture distribution).

```
invisible(plot(density(exp_1), col="red",lty=2, main="Mixed Density of 2 Exponential Dis
lines(density(exp_2), col="blue",lty=2)+
lines(density(mix_exp), col="black"))
legend("topright",legend=c('exp_1','exp_2','mixture'),cex=0.8,bty="n",lty=c(2,2,1),
       col=c("red","blue","black"))
```



Problem 3

Consider the problem of sampling from Beta(3,3) using Accept-Reject sampling method using the following information:

1) Use uniform distribution on (0,1) as the proposal density.
2) Note that the probability density of Beta(3,3) is symmetric around 0.5 with the mode of 1.875.

a) Provide the accept-reject algorithm for the problem.

Answer

1. Draw 2 uniform random variables u1, u2.
2. Compute h(u1) = f(u1)/g(u1)K where K = 1.875. The maximum of f(x), a Beta(3,3), is 1.875. The minimum of g(x), a U(0,1), is 1. Thus, in order to ensure $f(x) \leq g(x)K$, K must be 1.875.
3. If u2 <= h(u1), accept u1; otherwise return to step 1

b) Draw 600 observations using the algorithms. Compare the mean and variance of the sample to the true values.

```r
set.seed(42)

n <- 600
alpha <- 3;
beta <- 3;
K <- 1.875;
draws <- 0;
tdraws <- 0;

while (length(draws) < n + 1){
  u <- runif(2)
  h <- dbeta(u[1], alpha, beta)/(dunif(u[1]) * K)
  if (u[2] <= h)
      draws <- c(draws, u[1])
  tdraws <- tdraws +1
}

target <- draws[2:(n+1)]
arate <- n/tdraws
```

Acceptance rate of the algorithm is 53.96%. Below is a table that compares the mean and variance of the sample to the true values. The results are close to the true values.

Statistic	Sample	True
Mean	0.5047679	0.5
Variance	0.0367806	0.0357143

Problem 4

Consider the following expectation of a function of truncated exponential distribution:

$$E(\frac{1}{1+x^2}) \tag{1}$$

where x ~ $Exp(1)$ truncated to [0,1]. I.e., x is truncated exponential distribution from 0 and 1 with parameter=1 (i.e. $\theta = 1$ from problem 1.)

a) Show that the above problem is equivalent to the following problem:

$$\frac{1}{1-e^{-1}} \int_0^1 \frac{1}{1+x^2} e^{-x} dx \tag{2}$$

Answer

The expectation of a distribution truncated to [0,1] is as follows:

$$E(X|0 \leq x \leq 1) = \frac{\int_0^1 xg(x)dx}{F(1) - F(0)}$$

where g(x) is the probability density function and F(x) is the cumulative distribution function. Thus, the above can be simplified to:

$$\frac{\int_0^1 xg(x)dx}{F(b) - F(a)} = \frac{\int_0^1 (\frac{1}{1+x^2})(\theta e^{-\theta x})dx}{(1 - e^{-\theta(1)}) - (1 - e^{-\theta(0)})}$$

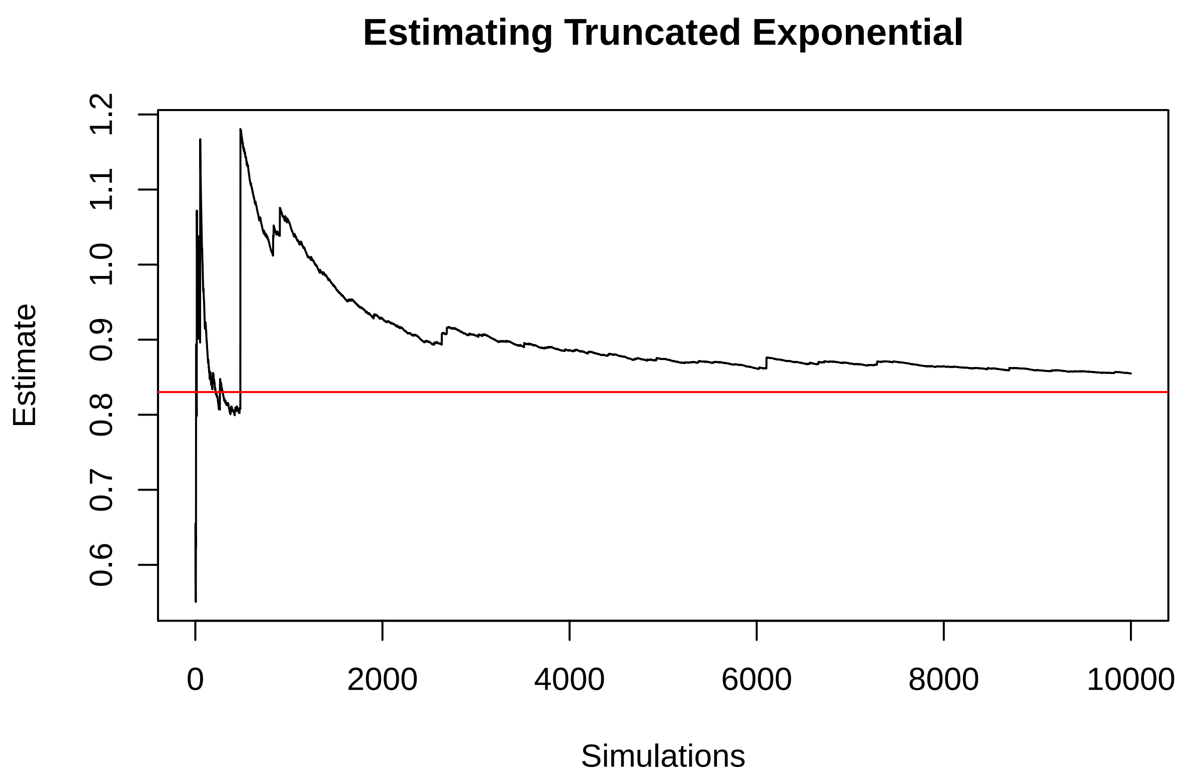
With $\theta = 1$, it can be further simplified to:

$$\begin{aligned} \frac{\int_0^1 (\frac{1}{1+x^2})(\theta e^{-\theta x})dx}{(1 - e^{-\theta(1)}) - (1 - e^{-\theta(0)})} &= \frac{\int_0^1 (\frac{1}{1+x^2})(e^{-x})dx}{(1 - e^{-(1)}) - 0} \\ &= \frac{1}{1-e^{-1}} \int_0^1 \frac{1}{1+x^2} e^{-x} dx \end{aligned}$$

b) Calculate equation (2) using importance sampling method. Use Beta(2,3) density function as the importance function (i.e., h in the lecture note). First, describe the algorithm of the method for this problem and Second, implement the algorithm in R with the number of simulations=10,000.

```r
set.seed(42)

integrand <- function(x) {exp(-x)/(1+x^2)}
scale <- (1/(1-exp(-1)))
true_val <- scale*integrate(integrand, 0, 1)$value

Nsim = 10000
x = rbeta(Nsim, 2, 3)
```

```r
f<-function(x) (exp(-x)/(1+x^2))
#weight = dexp(1/(1+x^2))/dbeta(x, 2, 3)
weight = f(x)/dbeta(x,2,3)
plot(scale*cumsum(weight)/1:Nsim, type="l", xlab="Simulations", ylab="Estimate",
     main="Estimating Truncated Exponential")
abline(a=true_val,b=0,col="red")
```



Using 10000 simulations, the estimate of equation (2) is 0.8549389. The true value is 0.8302169 so the estimate is pretty close.

Problem 5

Consider the Gibbs sampling example for a standard bivariate normal distribution discussed in the class. Estimate the mean and standard deviation of y_1 using Monte Carlo with Gibbs Sampling method using last 100 observations.

Do this with:

a) $\rho = 0.1$ and 0.99
b) starting values of y_{20}=-1.5 and 1.5
c) Number of simulations=200, 500, 5000,10,000.

```r
set.seed(42)

gibbs_binormal <- function(rho, init, n_sim) {
  y1 <- rep(0,n_sim)
  y2 <- rep(0,n_sim)
  y0 <- init
  y2[1] <- y0;
  for (i in 2:n_sim) {
    y1[i] <- rnorm(1,rho*y2[i-1],sqrt(1-rho^2))
    y2[i] <- rnorm(1,rho*y1[i],sqrt(1-rho^2))
  }

  last_100 <- tail(y1, 100)
  return(c(mean(last_100),sd(last_100)))
}

df <- data.frame(Simulations = c(rep(c(200,500,5000,10000),4)),
                 Rho = c(rep(0.1,8),rep(0.99,8)),
                 Y2_Initial_Val = c(rep(c(rep(-1.5,4),rep(1.5,4)),2)))

all_results <- apply(df,1,function(params)gibbs_binormal(params[2],params[3],params[1]))
df$Y2_Mean <- all_results[1,]
df$Y2_SD <- all_results[2,]
```

Compare your results with the true marginal distribution of y_1. What does your result tell you about Gibbs Sampling?

Simulations	Rho	Y2 Initial Value	Y1 Gibbs Mean	Y1 Gibbs SD	Y1 True Mean	Y1 True SD
200	0.1	-1.5	0.0375776	0.974377	-0.15	0.99
500	0.1	-1.5	-0.0363149	0.9952976	-0.15	0.99
5000	0.1	-1.5	-0.0916586	0.9747732	-0.15	0.99
10000	0.1	-1.5	-0.0193945	0.9907541	-0.15	0.99
200	0.1	1.5	0.1282435	1.0080023	0.15	0.99
500	0.1	1.5	-0.0407912	1.0603981	0.15	0.99
5000	0.1	1.5	-0.0491358	0.9117949	0.15	0.99
10000	0.1	1.5	0.1086134	0.9571421	0.15	0.99
200	0.99	-1.5	-0.5780039	0.9185936	-1.485	0.0199
500	0.99	-1.5	-0.1399878	0.4668545	-1.485	0.0199
5000	0.99	-1.5	0.1948645	0.5608134	-1.485	0.0199
10000	0.99	-1.5	-0.5282504	0.5384953	-1.485	0.0199

Simulations	Rho	Y2 Initial Value	Y1 Gibbs Mean	Y1 Gibbs SD	Y1 True Mean	Y1 True SD
200	0.99	1.5	-0.2155421	0.4694503	1.485	0.0199
500	0.99	1.5	-0.6524142	0.3647649	1.485	0.0199
5000	0.99	1.5	0.2146238	0.933904	1.485	0.0199
10000	0.99	1.5	-0.624498	0.3297125	1.485	0.0199

Problem 6

Describe the steps for constructing a random walk Metropolis-Hasting sampler to generate a sample of 10,000 from the distribution with the p.d.f $= 0.5e^{-|x|}$, known as Laplace distribution using the standard normal for ϵ to generate proposals: $y = x^{i-1} + \epsilon$. I do **not** need a R-script for this, but I recommend you try this in R as well. Please do **not** submit R-script. I will make a R-script for this available when I post the answer keys. FYI, Laplace distribution is not implemented in the base R, although there are several packages which have it.

Answer

1. Choose an initial value x_0, say 0.
2. Generate proposal y, where $y = x_{i-1} + N(0, 1)$.
3. Derive α', the minimum of the following two values:
   - The ratio P(y)/P(x), where P is the pdf of the Laplace distribution.
   - 1
4. Generate a uniform random variable, u.
5. Generate x_i, where if u $< \alpha'$, $x_i = y$, else $x_i = x_{i-1}$.
6. Repeat steps 2-6 until 10,000 observations (i=10000).

```r
x.start= 0 # initial value
Nsim = 10000;
X=rep(x.start,Nsim) # initialize the chain
 for (i in 2:Nsim){
  Y=X[i-1]+rnorm(1) # generate proposal

  # can indirectly calculate density of laplace using exponential
  ratio= min((dexp(abs(Y))/2)/(dexp(abs(X[i-1]))/2),1)

  X[i]=X[i-1] + (Y-X[i-1])*(runif(1)<ratio)
 }
```