Homework 8: Solutions

36-350, Fall 2011

Note: Unfortunately, there was a typo in the homework instructions: λ was supposed to follow a gamma distribution with shape a and scale 1. The definition of the gamma density given in the homework instructions reflected the intended distribution, therefore while all students generated their λ values from the $gamma(shape = 1, scale = a)$ distribution but calculated $p(x)$ and $p(\lambda|x)$ as if $\lambda \sim gamma(shape = a, scale = 1)$. This means that the histograms will not match the calculated density in Problem 3, and possibly not match in Problem 6 depending on how students did Problem 5. In order to accommodate this in the grading, we gave everyone full credit for generating their λ's or calculating densities using either case, however we docked points if they failed to recognize the mismatch between the histograms and densities.

1. Solution

   For each value of X, first we must draw a value of λ from the gamma distribution, and then use that to define the exponential distribution from which to draw x.

   ```
   rexpgamma <- function(n, a) rexp(n, rate = rgamma(n, shape = a, scale = 1))
   ```

2. Solution

   ```
   mcmc <- function(x, prior.a, n){

     ## p(x|lambda)*p(lambda)
     ## Since some of our lambda proposals might be non-positive, for these we
     ## force dexpgamma to return 0.
     dexpgamma <- function(x,lam){
       ifelse(lam <= 0,0,dexp(x, rate = lam)*dgamma(lam, scale = 1, shape = prior.a))
     }

     ## Prepare a vector to store sampled lambdas
     lam.vec <- rep(NA,n)

     ## Draw initial value of lambda from prior distribution
     lam.vec[1] <- rgamma(1, shape = 1, scale = prior.a)

     for (ii in 1:(n-1)){

       ## Draw a proposed value of lambda from uniform distribution
       prop.lam <- runif(1, min = lam.vec[ii]-.5, max = lam.vec[ii]+.5)

       ## Draw an acceptance probability
       acc.prob <- runif(1,0,1)

       ## Part 4 of algorithm in class notes
       if (acc.prob < dexpgamma(x,prop.lam)/dexpgamma(x,lam.vec[ii])){
         lam.vec[ii+1] <- prop.lam
       } else {
         lam.vec[ii+1] <- lam.vec[ii]
   ```

```
    }
  }
  return(lam.vec)
}
```

3. Solution

$$\begin{aligned} p(x) &= \int_0^\infty \frac{\lambda^{a-1}e^{-\lambda}}{\Gamma(a)}\lambda e^{-\lambda x} \quad d\lambda \\ &= \int_0^\infty \frac{\lambda^{a}e^{-\lambda(x+1)}}{\Gamma(a)} \quad d\lambda \end{aligned}$$

Letting $\lambda = \frac{t}{1+x}$, then $d\lambda = \frac{dt}{1+x}$ and:

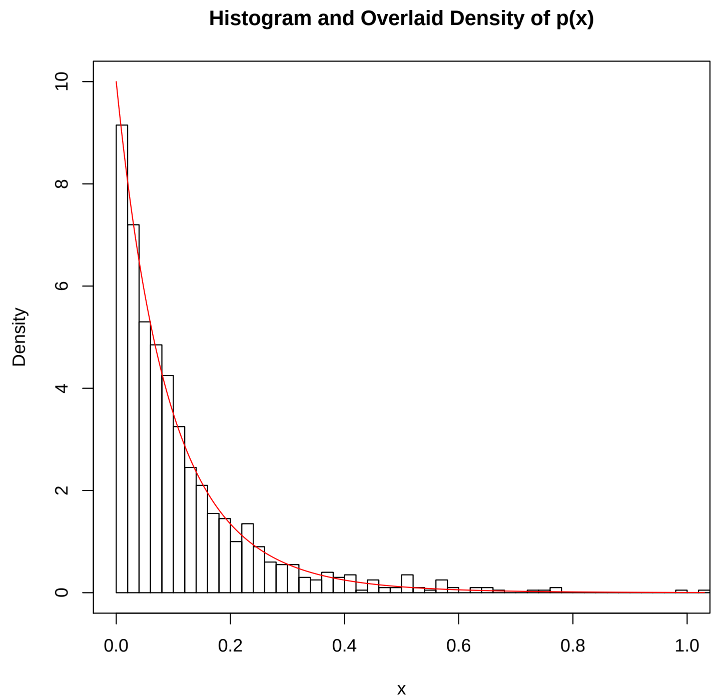
$$\begin{aligned} \int_0^\infty \frac{\lambda^{a}e^{(-\lambda(x+1)}}{\Gamma(a)} \quad d\lambda &= \int_0^\infty \left(\frac{t}{1+x}\right)^a e^{-t} \quad \frac{dt}{1+x} \\ &= \frac{(1+x)^{-(a+1)}}{\Gamma(a)}\int_0^\infty t^a e^{-t} \quad dt \\ &= \frac{\Gamma(a+1)}{\Gamma(a)}(1+x)^{-(a+1)} \\ &= a(1+x)^{-(a+1)} \end{aligned}$$

4. Solution

```
X <- rexpgamma(n=1000,a=10)
p <- function(x,a){a*(1+x)^-(a+1)}

hist.density <- function(X, xlim = range(X), breaks = 500){
  plot(xlim, c(0,10), type = "n", main = "Histogram and Overlaid Density of p(x)",
       xlab = "x", ylab = "Density")
  hist(X,freq = FALSE, breaks = breaks, add = T)
  x <- seq(0,max(X),length.out = 1000)
  lines(x,p(x,10), col = "red")
}

hist.density(X,xlim = c(0,3), breaks = 50)
```



Yes, the histogram and the density do match.

5. Solution

$$
\begin{aligned}
p(\lambda|x) &= \frac{p(x|\lambda)p(\lambda)}{p(x)} \\
&= \frac{\lambda e^{-\lambda x}\frac{\lambda^{a-1}e^{-\lambda}}{\Gamma(a)}}{a(1+x)^{-(a+1)}} \\
&= \frac{\lambda^{a}(x+1)^{a+1}e^{-\lambda(x+1)})}{a\Gamma(a)}
\end{aligned}
$$

which is a Gamma($shape = a + 1, rate = (x + 1)$). So our function can be written

```
p.lam.x <- function(lam, x , a) {
  (dgamma(lam, shape = a, scale = 1)*dexp(x, rate = lam))/(a/((x+1)^(a+1)))
}
```

6. Solution

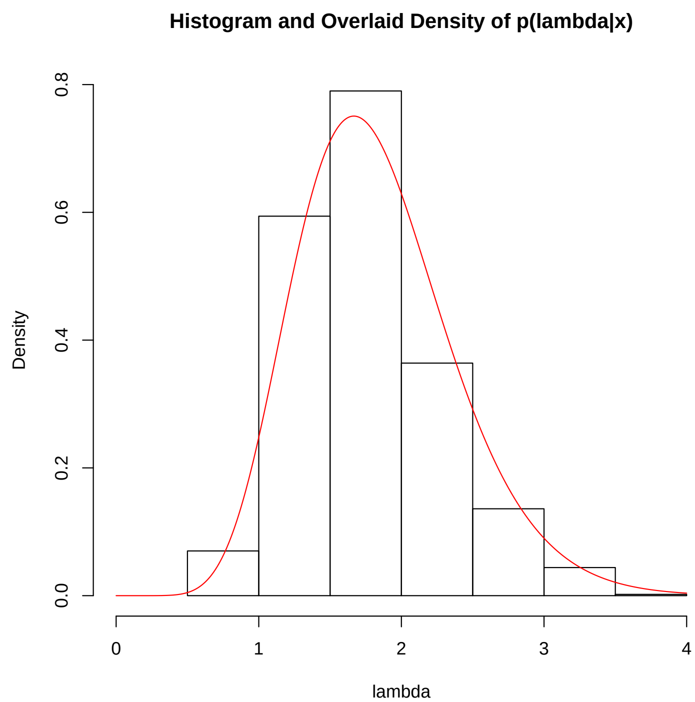
```
lambdas <- mcmc(x = 5, prior.a = 10, n = 2000)

posterior <- function(lambda,x,a){
  (1+x)^(a+1)/(a*gamma(a))*(lambda^a)*exp(-(x+1)*lambda)
}

hist.density <- function(lambdas, xlim = range(lambdas)){
  hist(lambdas,freq = FALSE, main = "Histogram and Overlaid Density of p(lambda|x)",
     xlab = "lambda", ylab = "Density", xlim = xlim)
```

```
  x <- seq(0,max(xlim),length.out = 1000)
  lines(x,posterior(lambda = x, x = 5, a = 10), col = "red")
}

hist.density(lambdas[-(1:1000)], xlim = c(0,4))
```



The purpose of the Metropolis algorithm is to make draws from $p(\lambda|x)$ without having to know $p(x)$, so if our Metopolis sampler is working then the histogram should match the density calculated in Problem 5. Figure 2 shows that it does.

7. Solution

If we want λ to come from a different density, then Problem 1 would require a substitution for the `rgamma` function. Problem 2 only needs a substitution for the `dgamma` function within the `dexpgamma()` function (near the top of the code for the `mcmc()` function). We would also need to change the inputs to both the `rexpgamma()` function from Problem 1 and `mcmc()` function from Problem 2 to accommodate the new parameters for λ's distribution.