ENSAE - Computational Statistics

Professor Christian P. Robert

Zakarya Ali

December 15, 2017

Problem 8.6. Reproduce the comparison of Problem 8.5 in the case of (a) the gamma distribution and (b) the Poisson distribution.

(a) Gamma distribution

For any $Gamma(\alpha, \beta)$, the density is proportional to :

$$f(x; \alpha, \beta) = x^{\alpha-1} e^{-\frac{x}{\beta}}$$

f can be decomposed into two terms : $f_1(x) = x^{\alpha-1}$ and $f_2(x) = e^{-\frac{x}{\beta}}$
$(f(x; \alpha, \beta) = f_1(x; \alpha, \beta) f_2(x; \alpha, \beta))$

We can then suggest the following Slice Sampler :

Algorithm 1 Slice Sampler for the Gamma Distribution

$\forall t$,

- Draw $U_1^{(t+1)} | x^{(t)} \sim U_{[0,(x^t)^{\alpha-1}]}$
- Draw $U_2^{(t+1)} | x^{(t)} \sim U_{[0,e^{-\frac{x^t}{\beta}}]}$
- Draw $X^{(t+1)} | \omega_1^{(t+1)}, \omega_1^{(t+1)} \sim U_{[\omega_1^{\frac{1}{\alpha-1}}, -\beta log(\omega_2)]}$

Figure 1 show the Gamma distribution cdf for 10^4 iterations from the Slice Sampler and R function rgamma for $\alpha = 2$ and $\beta = 2$. They are very close.

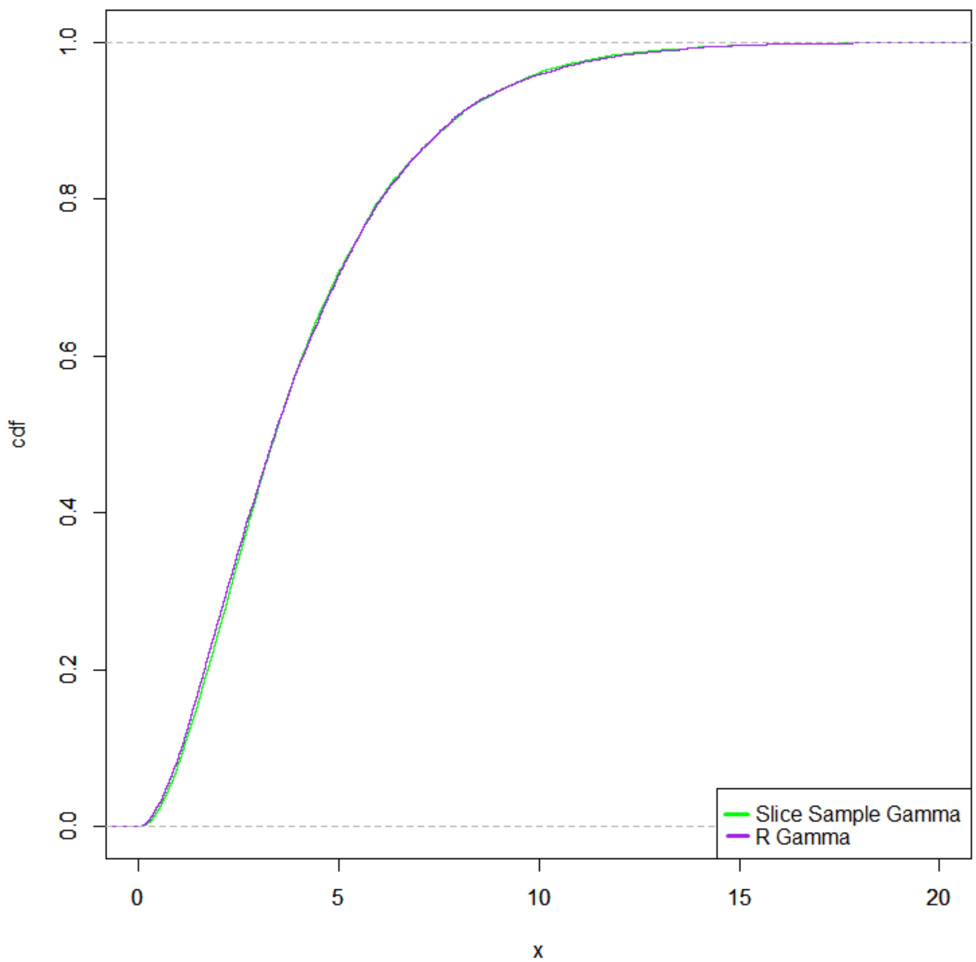

Figure 1: Slice sampler and R Gamma cdf for $Gamma(2, 2)$

```r
# Initialization
#set.seed(1)
iterations = 10**4
#Gamma(shape=alpha, scale=beta)
alpha=2
beta=2

#gamma function
slice_sampler_gamma <- function(n, x0, alpha, beta){
  X = NULL
  Xi = x0
  for(i in seq(n)){
    U1 = runif(1,0,(Xi**(alpha-1)))
    U2 = runif(1,0,exp((-Xi)/beta))
    lim_U1=U1**(1/(alpha-1))
    lim_U2=-log(U2)*beta
    if(lim_U2>lim_U1){
      Xi = runif(1,lim_U1,lim_U2)
      X = c(X,Xi)
    }
  }
  return (X)
}

#generate slice sampler and gamma distribution
sample_gamma <- slice_sampler_gamma(iterations,0.033, alpha, beta)
gamma_distribution <- dgamma(seq(0, 3.5, length=100), shape=alpha,
    scale=beta)

head(sample_gamma)
length(sample_gamma)

#plot
plot(ecdf(sample_gamma), xlim=c(0,20), col="green", ylim=c(0,1), main
    ="", ylab="cdf")
par(new=TRUE)
plot(ecdf(rgamma(iterations,shape=alpha,scale=beta)), xlim=c(0,20),
    col="purple", ylim=c(0,1), main="", ylab="")
legend("bottomright", col=c("green", "purple"), legend=c("Slice
    Sample Gamma","R Gamma"))
```

Listing 1: Code for Gamma Slice sampler generation, histogram and cdf

(b) the Poisson distribution

We use a similar strategy for the Poisson distribution and we take into account the fact that it's a discrete distribution.

The $Poisson(\lambda)$ density is proportional to : $f_1(x)f_2(x)$, with $f_1(x) = \lambda^x$ and $f_2(x) = \frac{1}{x!}$

Algorithm 2 Slice Sampler for the Poisson Distribution

$\forall t$,

- Draw $U_1^{(t+1)}|x^{(t)} \sim U_{[0,f_1(x)]}$
- Draw $U_2^{(t+1)}|x^{(t)} \sim U_{[0,f_2(x)]}$
- Draw $X^{(t+1)}|\omega_1^{(t+1)}, \omega_1^{(t+1)} \sim U_{\{y, y \geq \frac{log(\omega_1)}{log(\lambda)}, \frac{1}{\omega_2} \geq y!\}}$

We get Figure 2 comparison. The cdfs obtained are less conclusive. This may be due to the bounds I had to impose on my computer regarding the upper and lower bounds (see R code in appendix).

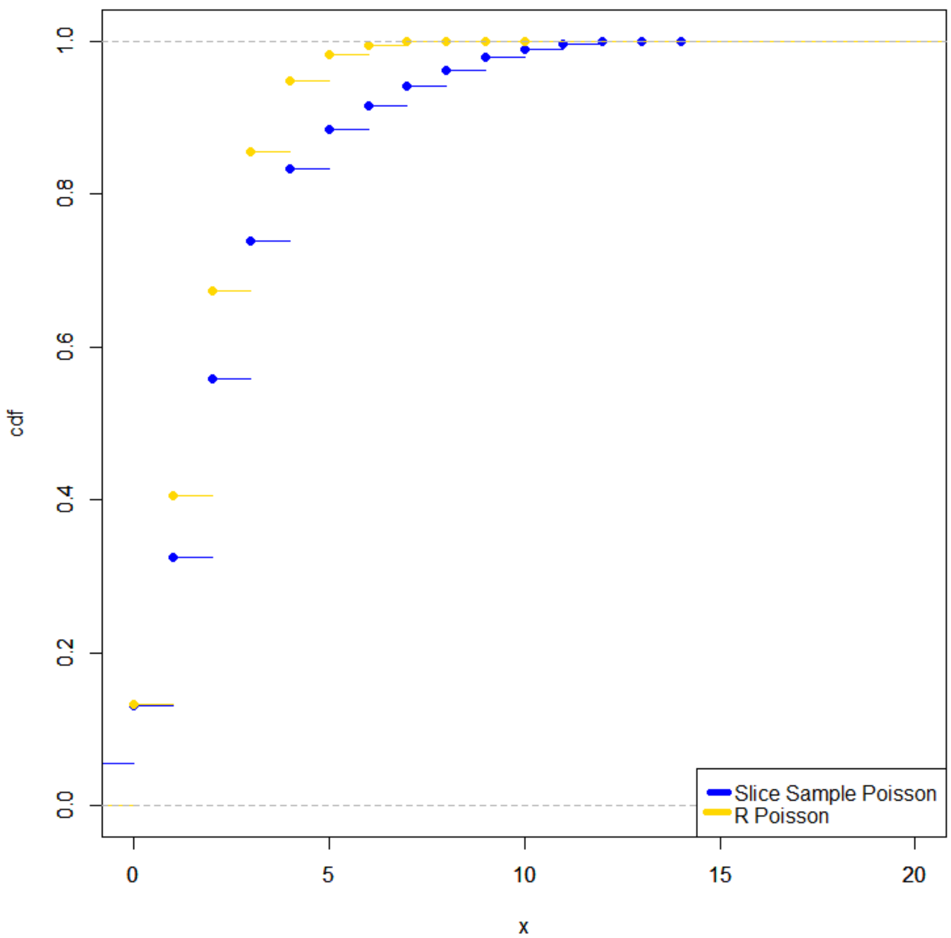

Figure 2: Slice sampler and R Poisson cdf for $Poisson(2)$

```r

#Poisson
lambda=2

max_factorial = function(max){
  x = 1
  factorial_x = x
  while(factorial_x < max){
    x=x+1
    factorial_x = factorial_x * x
  }
  return(x-1)
}

#Poisson function
slice_sampler_poisson <- function(n, x0, lambda){
  X = NULL
  Xi = x0
  for(i in seq(n)){
    U1 = runif(1,0,min(10**12,lambda**(Xi)))
    U2 = runif(1,0,max(1/factorial(Xi),10**(-9)))
    lim_U1=ceiling(log(U1)/log(lambda))
    lim_U2=max_factorial(1/U2)
    if(lim_U2>lim_U1){
      Xi = runif(1,lim_U1,lim_U2)
      X = c(X,ceiling(Xi))
    }
  }
  return (X)
}

#generate slice sampler and Poisson distribution
sample_poisson <- slice_sampler_poisson(iterations,1, lambda)

head(sample_poisson)
length(sample_poisson)

#plot
plot(ecdf(sample_poisson), xlim=c(0,20), col="blue", ylim=c(0,1), main="", ylab="cdf")
par(new=TRUE)
plot(ecdf(rpois(iterations,lambda)), xlim=c(0,20), col="gold", ylim=c(0,1), main="", ylab="")
legend("bottomright", col=c("blue", "gold"), legend=c("Slice Sample Poisson","R Poisson"))
```

Listing 2: Code for Poisson Slice sampler generation, histogram and cdf