Examen Parcial 2

Bernardo Mondragón Brozon

November 7, 2018

Problema 1

Aplicar el algoritmo de Metrópolis-Hastings para simular 500 observaciones de la distribución doble exponencial con densidad

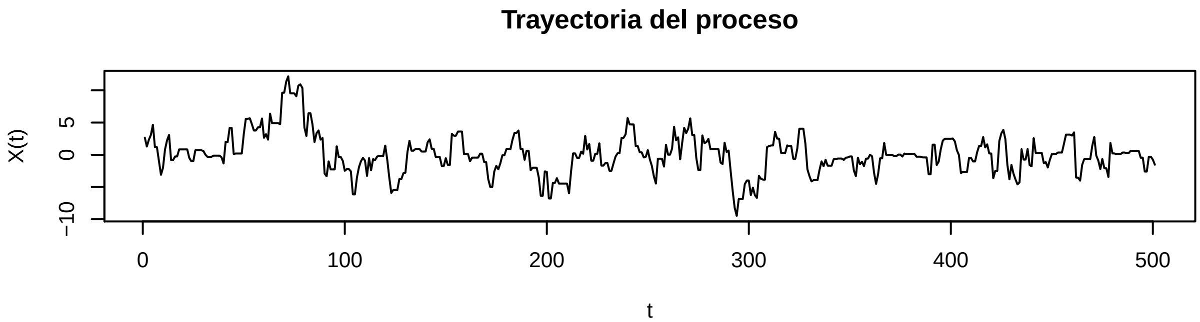
$$f(x) = \frac{\lambda}{2} e^{-\lambda|x|}, \quad x \in \mathbb{R}.$$

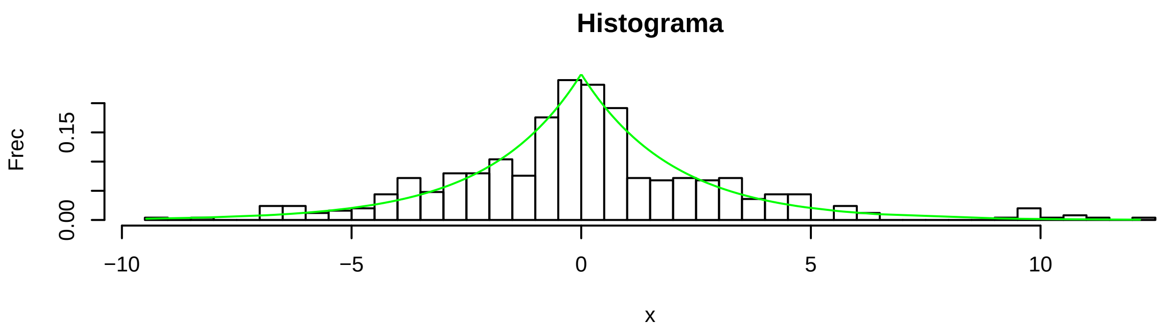
Usar la distribución normal como distribución propuesta. Comprobar estadisticamente con un nivel de confianza del 95% que la muestra obtenida proviene de la distribución indicada.

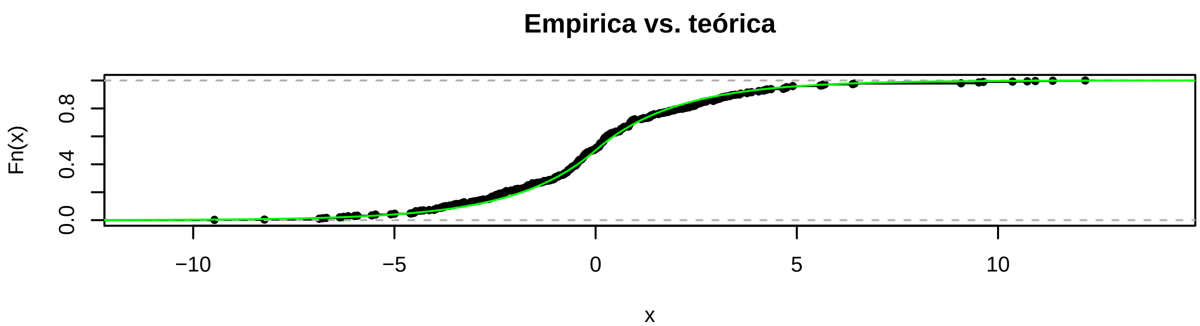
Solución:

Suponiendo $\lambda = 0.5$ se obtiene lo siguiente:

```
breaks <- 50 # Prticion del histograma
simulaDobleExponencial <- function(n, lambda) {
  f <- function(x) {(lambda/2)*exp(-lambda*abs(x))}
  x <- NULL
  x0 <- 3
  for(i in 0:n) {
    w <- ifelse(i==0,x0,x[i])
    y <- rnorm(1, mean = w, sd = 2.5)
    alfa <- min(1,(f(y)*dnorm(w,mean=y,sd=1))/(f(w)*dnorm(y,mean=w,sd=1)))
    x <- append(x,ifelse(runif(1)<alfa,y,w))
  }
  return(list(x=x,f=f(sort(x))))
}
data <- simulaDobleExponencial(n, lambda)

par(mfrow = c(3,1))
plot(data$x, type="l", main="Trayectoria del proceso", xlab="t", ylab="X(t)")
hist(data$x, probability=T, breaks = breaks, main="Histograma", xlab="x", ylab="Frec")
lines(sort(data$x), data$f, col="green")

li <- 1
ls <- n

# CDF
plot(ecdf(data$x[li:ls]), main="Empirica vs. teórica")
x <- seq(-15,15,by=0.01)
library(rmutil)
lines(x, plaplace(x, m=0, s=1/lambda), col="green")
```







A continuación se muestra la salida de la prueba de bondad de ajusta de Kolmogorov-Smirnov:

```
ks.test(data$x, "plaplace", 0, 1/lambda)
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  data$x
## D = 0.042779, p-value = 0.3184
## alternative hypothesis: two-sided
```

Si el valor p (p-value) obtenido es mayor que 0.05, entonces podemos concluir que la muestra efectivamente proviene de la distribución de la cual se pide la muestra.

■

Problema 2

Supongan que $V \sim \exp(1)$ y consideren que dado $V = v$, $W \sim \exp(1/v)$, de manera que $E(W|V = v) = v$. Describir un algoritmo para estimar $P(VW \leq 3)$, que solo requiera generar una variables aleatoria por muertra. Programar el algoritmo y mostrar que funciona, generando 100 muestras.

Solución:

Sea $n = 100$ el tamaño de la muestra. La probabilidad que se pide se puede determinar de dos maneras. El primer método es más intiutivo y el segundo es la solución que se pide.

Método 1

Para el i-ésimo valor x_i de la muestra hacer lo siguiente:

- Generar una variable aleatoria v_i distribuida exponencialmente con media 1.
- Generar una variable aleatoria w_i distribuida exponencialmente con media v_i.
- Hacer $x_i = v_i w_i$.

Entonces un estimador de $P = Pr\{VW \leq z\}$ puede ser el siguiente:

$$\widehat{P} = \frac{\#\text{ de } x_i\text{'s} \leq z}{n}$$

```r
n <- 100 # tamaño de la muestra
z <- 3

# Metodo 1
total <- 0
for (i in 1:n) {
  v <- rexp(1, rate = 1)
  w <- rexp(1, rate = 1/v)
  vw <- v*w
  if (vw <= z) {
    total <- total + 1
  }
}
p.est.1 <- total/n
```

De esta manera se tiene que

$$\widehat{P} = \frac{\#\text{ de } x_i\text{'s} \leq 3}{100} = 0.91$$

Para estimar la probabilidad anterior se generaron un total de 200 variables aleatorias. El segundo método (que es el que pide el problema) es más eficiente y solo requiere la generación de 100 variables aleatorias.

Método 2

Sea $z = 3$, entonces la probabilidad P que se pide está dada por

$$\begin{aligned} P = Pr\{VW \leq z\} &= \int_V Pr\{W \leq \frac{z}{v}|V = v\}Pr\{V = v\}\ \mathrm{d}v \\ &= \int_0^\infty \left[1 - e^{-\frac{z}{v^2}}\right] e^{-v}\ \mathrm{d}v \\ &= E_V\left[1 - e^{-\frac{z}{V^2}}\right] \end{aligned}$$

Entonces un estimador de esta probabilidad está dado por

$$\widehat{P} = \widehat{E_V}\left[1 - e^{-\frac{z}{V^2}}\right] = \frac{1}{n}\sum_{i=1}^{n}\left[1 - e^{-\frac{z}{v_i^2}}\right].$$

Entonces para hallar este valor estimado se describe el siguiente algoritmo. Para el i-ésimo valor x_i de la muestra hacer lo siguiente:

- Generar una variable aleatoria v_i distribuida exponencialmente con media 1.
- Hacer $x_i = \left[1 - e^{-\frac{z}{v_i^2}}\right]$.
- Calcular $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$.

```
# Metodo 2
vs <- rexp(n, rate = 1)
xs <- 1-exp((-(z/(vs^2))))
p.est.2 <- mean(xs)
```

Por lo tanto se tiene que

$$\widehat{P} = \widehat{E_V}\left[1 - e^{-\frac{3}{V^2}}\right] = \frac{1}{100}\sum_{i=1}^{100}\left[1 - e^{-\frac{3}{v_i^2}}\right] = 0.8763643.$$

■

Problema 3

Probar que si se elige a la distribucion candidata como una caminata aleatoria en el algoritmo de Metropolis-Hastings, entonces $\frac{q(y|x)}{q(x|y)}$ es de la forma $h(|y-x|)$ para alguna función h.

Solución:

Si la distribución objetivo de la que se quiere muestrear es la distribución límite de una caminata aleatoria, entonces se requiere que las probabilidades de transición sean simetricas y dependan unicamente de la distancia entre los posibles estados, de manera que

$$q(y|x) = h(|y-x|) = q(x|y).$$

Entonces la probabilidad de aceptación estrá dada por

$$\alpha = \min\left(1, \frac{\pi(y)q(y|x)}{\pi(x)q(x|y)}\right) = \min\left(1, \frac{\pi(y)}{\pi(x)}\right)$$

Con esta probabilidad de aceptación, la distribución límite será la distribución objetivo.

■

Problema 4

Si $\hat{\theta}_1$ y $\hat{\theta}_2$ son cualesquiera dos estimadores insesgados de θ, encontrar el valor de c que minimixa la varianza del estimador $\hat{\theta}_c = c\hat{\theta}_1 + (1-c)\hat{\theta}_2$.

Solución:

$$
\begin{aligned}
Var(\hat{\theta}_c) &= Var(c\hat{\theta}_1 + (1-c)\hat{\theta}_2) \\
&= c^2 Var(\hat{\theta}_1) + (1-c)^2 Var(\hat{\theta}_2) + 2c(1-c)Cov(\hat{\theta}_1, \hat{\theta}_2) \\
&= c^2 \left[Var(\hat{\theta}_1) + Var(\hat{\theta}_2) - 2Cov(\hat{\theta}_1, \hat{\theta}_2) \right] + 2c \left[Cov(\hat{\theta}_1, \hat{\theta}_2) - Var(\hat{\theta}_2) \right] + Var(\hat{\theta}_2)
\end{aligned}
$$

Si c^* es el valor que minimiza $Var(\hat{\theta}_c)$ entonces

$$
\frac{dVar(\hat{\theta}_c)}{dc}\Big|_{c=c^*} = 2c^* \left[Var(\hat{\theta}_1) + Var(\hat{\theta}_2) - 2Cov(\hat{\theta}_1, \hat{\theta}_2) \right] + 2 \left[Cov(\hat{\theta}_1, \hat{\theta}_2) - Var(\hat{\theta}_2) \right] = 0
$$

$$
\Rightarrow c^* = \frac{Var(\hat{\theta}_2) - Cov(\hat{\theta}_1, \hat{\theta}_2)}{Var(\hat{\theta}_1) + Var(\hat{\theta}_2) - 2Cov(\hat{\theta}_1, \hat{\theta}_2)}.
$$

■

Problema 5

Encontrar dos funciones de importancia f_1 y f_2 que tengan soporte en $(1, \infty)$ y estén 'cerca' de

$$
g(x) = \frac{x^2}{\sqrt{2\pi}} \exp(-x^2/2)dx, \quad x > 1
$$

Solución:

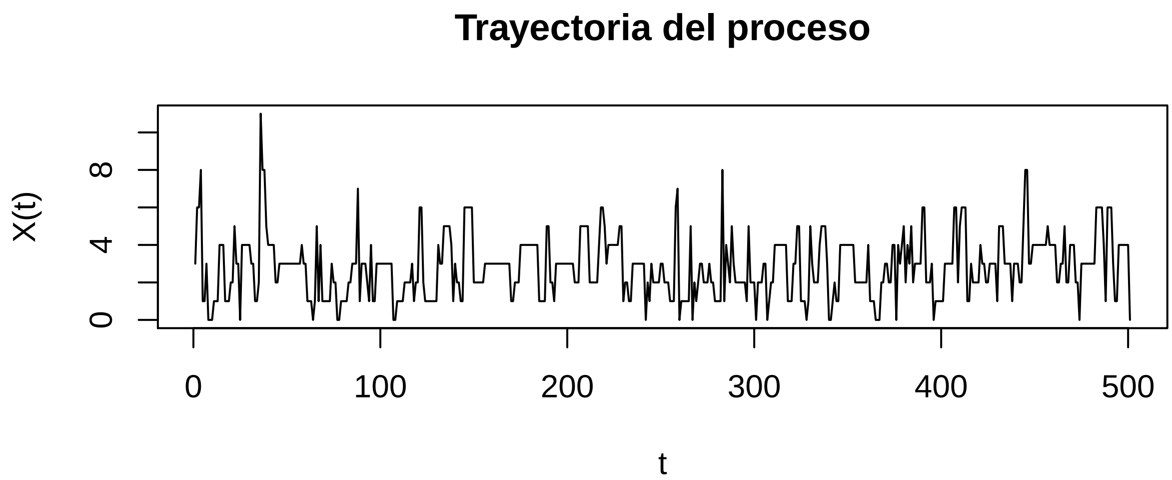
■

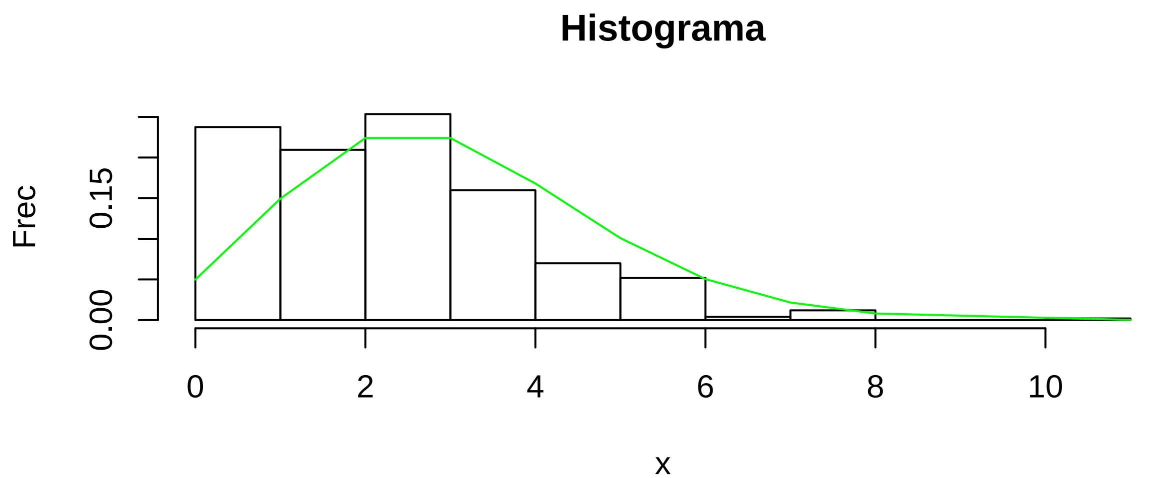
Problema 6

Consideren una distribución Poisson con parámetro $\lambda = 3$ condicionada a que no sea 0. Implementar un algoritmo MCMC para simular de esta distribución, usando una distribución propuesta que sea geométrica con parámetro $p = 1/3$. Usar la simulación para estimar la media y la varianza.

Solución:

```r
simulaPoisson <- function(n, lambda = 3) {
  x <- NULL
  x0 <- 3
  for(i in 0:n) {
    w <- ifelse(i==0,x0,x[i])
    y <- rgeom(1, prob=1/3)
    alfa <- min(1,(dpois(y, lambda)*dgeom(w, prob=1/3))/(dpois(w, lambda)*dgeom(y, prob=1/3)))
    x <- append(x,ifelse(runif(1)<alfa,y,w))
  }
  return(list(x=x,f=dpois(sort(x), lambda)))
}
pois <- simulaPoisson(500)
par(mfrow = c(2,1))
plot(pois$x,type="l", main="Trayectoria del proceso", xlab="t", ylab="X(t)")
p1_pois <- hist(pois$x,probability=T, breaks=10, main="Histograma", xlab="x", ylab="Frec")
lines(sort(pois$x),pois$f,col="green")
```





En efecto, los valores simulados provienen de la distribución de la cual se pide la muestra. A continuación se muestra la salida de la prueba de bondad de ajuste de Kolmogorov-Smirnov:

```
y <- rpois(500, 3)
library(stats)
ks_pois <- ks.test(pois$x, y)
est_lambda <- mean(pois$x)
ks_pois
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  pois$x and y
## D = 0.058599, p-value = 0.3566
## alternative hypothesis: two-sided
```

La media y la varianza de la distribución Poisson con parámetro λ es λ. En este caso, el estimador de máxima verocimilitud de λ está dado por

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{1}{500}\sum_{i=1}^{500} x_i = 2.8143713$$

$\blacksquare$