# Parcial - Estadística Computacional

Justo Andrés Manrique Urbina

Fecha: 28 de Mayo de 2018

## 1 Pregunta 3 (8 puntos)

Una variable aleatoria $X$, definida en toda la recta, tiene distribución normal asimétrica flexible generalizada, propuesta por Ma y Genton (2004), si su función de densidad es dada por la siguiente expresión:

$$2\phi(x|\xi,\sigma^2)\Phi(\alpha(x-\xi)/\sigma+\beta(x-\xi)^3/\sigma^3)$$

en dónde $\xi\in\mathbb{R},\sigma^2>0,\alpha\in\mathbb{R},\beta\in\mathbb{R},\phi(a|b,c^2)$ es la función de densidad de una distribución normal univariada con media $b$, varianza $c^2$, y evaluada en $a$, y $\Phi(a)$ es la función de distribución acumulada de una normal estándar evaluada en $a$. Una variable continua $X$que tenga función de densidad dada por la ecuación anterior se denota por $X\sim FSGN(\xi,\sigma,\alpha,\beta)$ (dónde FSGN son las siglas de *flexible generalized skew-normal*.

### 1.1 Resolución de la pregunta a)

**Resolución:**

Se implementa el algoritmo Metrópolis-Hastings (MH) para generar valores de dicha distribución. Para ello, dado que la función de densidad $FSGN$ tiene como base una distribución normal univariada, se tomará como función generadora de candidatos una distribución normal univariada, representada como $q(\star|x)\sim N(70,400)$, en dónde la media se encuentra centrada en la función a estimar y cuya varianza está basada en la varianza de la distribución a estimar.

Sea Y un valor candidado generado de $q(\star|x)\sim N(70,400)$, entonces el algoritmo MH es de la forma:

$$\alpha(X,Y)=min(1,\frac{f(Y)q(X|Y)}{f(X)q(Y|X)}$$

Dado que $q(X|Y)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-y)^2}{2\sigma^2}}$, y $q(Y|X)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(y-x)^2}{2\sigma^2}}$ ambos eventos se pueden cancelar quedando el algoritmo de la forma:

$$\alpha(X,Y)=min(1,\frac{f(Y)}{f(X)})$$

Ver a continuación la aplicación del algoritmo en el cuadro Algoritmo 1.

**Algorithm 1** Método de Metrópolis - Hastings

```
## Definimos los parámetros ##
rm(list=ls())
set.seed(50000)
eps = 70
sig = 20
alpha = 2
beta = -2
## Definimos la función de interés ##
FSGN = function(x,eps, sig, alpha, beta){
2 * dnorm(x=x,mean=eps, sd=sig) * pnorm(alpha*(x-eps)/sig+beta*(x-
eps)^3/sig^3,mean = 0,sd = 1)
}
## Definimos el algoritmo ##
# Valor Inicial
M=10000
z=1
# Metropolis Hastings
for(h in 2:M){
x = z[h-1]
y = rnorm(1, mean = eps,sd = sig^2)
u = runif(1)
abc= min(1,(FSGN(y,eps,sig,alpha,beta)/FSGN(x,eps,sig,alpha,beta)))
if(u<=abc){z[h]<-y} #acepta al candidato
if(u>abc){z[h]<-x} #rechaza al candidato
}
```

Posteriormente, graficamos los valores generados, así como un gráfico temporal de los valoers generados para observar si es estacionario. Ver gráficos a continuación:

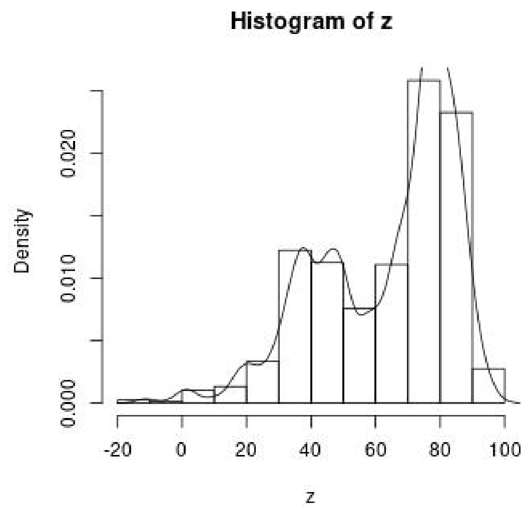


Figure 1: Resultados de la simulación

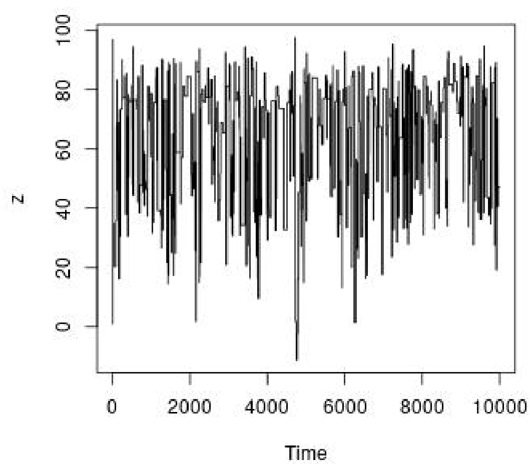


Figure 2: Linea de tiempo

Finalmente, se observa que la línea de tiempo se muestra estacionaria por lo que se toma dicha simulación como válida.

### 1.2 Resolución de la pregunta b)

**Resolución:**

Observamos que la función de verosimilitud de la función $FSGN$ se tiene de la forma[1]:

$$2\phi(x|\xi,\sigma^2)\Phi(\alpha(x-\xi)/\sigma+\beta(x-\xi)^3/\sigma^3)$$

$$\frac{2}{\sqrt{2\pi\sigma^2}}e^{\frac{-(x-\xi)^2}{\sigma^2}}*\frac{1}{2}\left[1+erf(\frac{\alpha(x-\xi)/\sigma+\beta(x-\xi)^3/\sigma^3}{\sigma\sqrt{2}})\right]$$

$$\prod_{j=1}^{n}\frac{2}{\sqrt{2\pi\sigma^2}}e^{\frac{-(x_j-\xi)^2}{\sigma^2}}*\frac{1}{2}\left[1+erf(\frac{\alpha(x_j-\xi)/\sigma+\beta(x_j-\xi)^3/\sigma^3}{\sigma\sqrt{2}})\right]$$

$$\frac{1}{\sigma^n(\sqrt{2\pi})^n}e^{\frac{-1}{\sigma^2}\sum_{j=1}^{n}(x_j-\xi)^2}\prod_{j=1}^{n}\left[1+erf(\frac{\alpha(x_j-\xi)/\sigma+\beta(x_j-\xi)^3/\sigma^3}{\sigma\sqrt{2}})\right]$$

$$\sigma^{-n}(\sqrt{2\pi})^{-n}e^{\frac{-1}{\sigma^2}\sum_{j=1}^{n}(x_j-\xi)^2}\prod_{j=1}^{n}\left[1+erf(\frac{\alpha(x_j-\xi)/\sigma+\beta(x_j-\xi)^3/\sigma^3}{\sigma\sqrt{2}})\right]$$

Posteriormente, la log-verosimilitud quedaría de la siguiente forma:

$$-n*log(\sigma)-n*log(\sqrt{2\pi})+\left(\frac{-1}{2\sigma^2}\sum_{j=1}^{n}(x_j-\xi)^2\right)+\sum_{j=1}^{n}log\left(1+erf(\frac{\alpha(x_j-\xi)/\sigma+\beta(x_j-\xi)^3/\sigma^3}{\sigma\sqrt{2}})\right)$$

Con el propósito de hallar los estimados de máxima verosimilitud se hizo uso del set de datos **faithful** de R. Ver código a continuación:

[1] Siempre y cuando la muestra sea independiente.

**Algorithm 2** Código R para estimación de máxima verosimilitud

```
data("faithful")
library(pracma)
log.like = function(theta){
eps = theta[1]
sig = theta[2]
alpha = theta[3]
beta = theta[4]
ftwt = faithful$waiting
-((-length(ftwt))*log(sig)-length(ftwt)*log(sqrt(2*pi))+((-
1/(2*sig^2))*sum((ftwt-eps)^2))+sum(log(pnorm(alpha*(ftwt-eps)/sig +
(beta*(ftwt-eps)^3/sig^3)))))
}
res.L = optim(c(70,20,0,0),log.like,method = "L-BFGS-B",hessian = T)
EMV = round(res.L$par,3)
```

Esto da como resultado, para $\beta$, el valor de -1.428.

## 1.3 Resolución de la pregunta c)

**Resolución:**

El estimador de máxima verosimilitud, en tanto la muestra sea grande, tiene la distribución:$\widehat{\theta} \overset{aprox}{\sim} N(\theta, V)$. En dónde $V$ se obtiene de la forma $\hat{V} = I_F(\hat{\theta})^{-1}$. Posteriormente, el intervalo de confianza se obtiene de la forma:

$$\left[\theta \pm Z_{1-\alpha} * (\sqrt{I_{F_{[4,4]}}(\hat{\theta})^{-1}})\right]$$

En dónde $I_{F_{[4,4]}}(\hat{\theta})^{-1}$es el valor asociado a $\beta$.

Ver código R a continuación:

**Algorithm 3** Código R para hallar el intervalo de confianza

```
fisher_info = solve(res.L$hessian)
fisher_info
sigma = sqrt(diag(fisher_info))
sigma = sigma[4]
ftwt = faithful$waiting
up = res.L$par[4] +1.96*(sigma)
low = res.L$par[4] -1.96*(sigma)
ci = c(low, res.L$par[4], up)
ci
```

Se obtiene un intervalo de confianza de [-2.018, -0.837], para un estimador puntual -1.428.

## 1.4 Resolución de la pregunta d)

**Resolución:**

Para resolver la pregunta, tomaremos $\beta = 0$ en la función de log-verosimilitud. Por lo tanto, la nueva ecuación sería:

$$-n*log(\sigma) - n*log(\sqrt{2\pi}) + \left(\frac{-1}{2\sigma^2}\sum_{j=1}^{n}(x_j - \xi)^2\right) + \sum_{j=1}^{n} log\left(1 + erf(\frac{\alpha(x_j - \xi)/\sigma}{\sigma\sqrt{2}})\right)$$

Se obtendrán los valores óptimos de esta ecuación a través de la función *optim* en R. Posteriormente, se evaluará la ecuación en sus valores óptimos frente a la ecuación inicial (con sus respectivos valores óptimos). Finalmente se realizará una prueba *ji-cuadrado* sobre esta evaluación para así probar la hipótesis.

Ver el código R a continuación:

**Algorithm 4** Código R para la prueba de hipótesis

```
log.like.0 = function(theta){
eps = theta[1]
sig = theta[2]
alpha = theta[3]
ftwt = faithful$waiting
-((-length(ftwt)*log(sig)-length(ftwt)*log(sqrt(2*pi))+((-
1/(2*sig^2))*sum((ftwt-eps)^2))+sum(log(pnorm(alpha*(ftwt-eps)/sig)))))
}
res.L.0 = optim(c(70,20,0),log.like.0,method = "L-BFGS-B")
EMV.0 = round(res.L.0$par,3)
l = 2*(log.like.0(EMV.0)-log.like(EMV))
pval = 1-pchisq(l,1)
pval
```

El p-valor resulta menor a 0.05, por lo que se rechaza la hipótesis nula. Por lo tanto, $\beta$ es distinto de 0.