Métodos de muestreo: Monte Carlo Markov Chain

David Montaño Castro

Metropolis-Hastings

1.- Sea Y ~ Binomial(n,p) con n fijo. Considere el logaritmo de los momios $\theta = log(\frac{p}{1-p})$ tal que su distrubión inicial (prior) es θ ~ Normal(μ,σ^2), con $\mu = 1$ y $\sigma^2 = 0.16$. Suponga que se observan n = 10 y y = 7.

Use el algoritmo de Metropolis-Hastings para simular la distribución final (posterior) de p, y de la θ.

Respuesta

Por conveniencia iniciaré por poner la verosimilitud en términos de θ, ya que de esta se conocen sus hiperparámetros por ser la distribución a priori.

No sin antes despejar de la igualdad dada la p.

$$
\begin{aligned}
\theta &= ln(\frac{p}{1-p}) \\
e^\theta &= \frac{p}{1-p} \\
e^\theta(1-p) &= p \\
e^\theta - pe^\theta &= p \\
e^\theta &= p + pe^\theta \\
e^\theta &= p(1+e^\theta) \\
p &= \frac{e^\theta}{1+e^\theta}
\end{aligned}
$$

Verosimilitud

Dado que Y ~ Binomial(n,p) :

$$
\begin{aligned}
f(y|p) &= \binom{n}{y} p^y (1-p)^{n-y} \\
&= \binom{n}{y} (\frac{p}{1-p})^y (1-p)^n \\
&= \binom{n}{y} e^{yln(\frac{p}{1-p})} (1-p)^n \\
&= \binom{n}{y} e^{y\theta} (1 - \frac{e^\theta}{1+e^\theta})^n \\
&= \binom{n}{y} e^{y\theta} (\frac{1+e^\theta}{1+e^\theta} - \frac{e^\theta}{1+e^\theta})^n \\
&= \binom{n}{y} e^{y\theta} (\frac{1}{1+e^\theta})^n
\end{aligned}
$$

Prior

Por hipótesis, se sabe que $\theta \sim Normal(\mu = 1, \sigma^2 = 0.16)$

Posterior θ

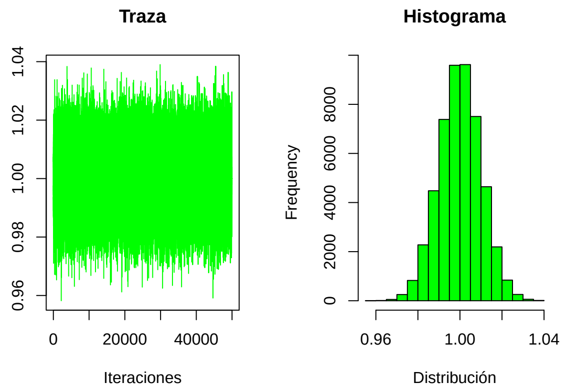
También por hipótesis, se sabe que y = 7 y n = 10

$$\begin{aligned} f(\theta|y=7,n=10) &= f(\theta)f(y|\theta) \\ &= Normal(\mu=1,\sigma^2=0.16)\binom{10}{7}e^{7\theta}(\frac{1}{1+e^{\theta}})^{10} \\ &\propto e^{-\frac{(\frac{\theta-1}{\sqrt{0.16}})^2}{2}}e^{7\theta}(\frac{1}{1+e^{\theta}})^{10} \end{aligned}$$

La distribución de la posterior luce complicada de calcular, incluso de ponerla de alguna forma conocida. Es por eso que se necesita utilizar una técnica de muestreo estocástico.

Metropolis-Hastings

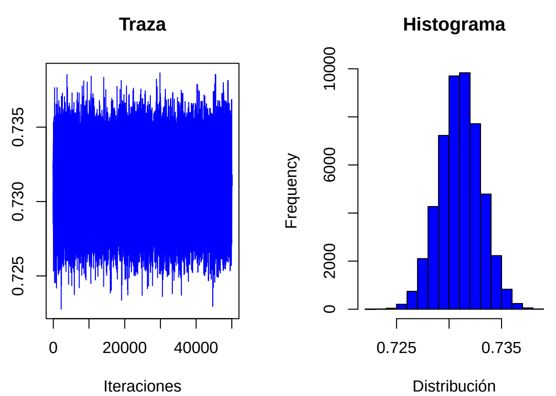
```
N = 50000

mu = 1
sigma = sqrt(.16)

X = array(0, dim=c(1,N))

X[1,] = rep(rnorm(n = 1, mean = mu, sd = 0.001 ), N)

for (j in 2:N) {
  y = rnorm(n = 1, mean = mu, sd = 0.01)
  alpha = min(dnorm(X[1, j-1],mu,sigma)*exp(7*y)*(1/(1+exp(y)))^10/(dnorm(y,mu,sigma)*exp(7*X[j-1])*(1/
  X[1,j] = X[1, j-1] + (y[1] - X[1,j-1]) * (runif(1)<alpha)
}


par(mfrow = c(1,2))
plot(X[1,], type = "l", main = "Traza", xlab = "Iteraciones", ylab = "", col = "green")
hist(X[1,], main = "Histograma", col = "green", xlab = "Distribución" )
```



Posterior P

Dada la simulación de los datos, se pueden obtener la distribución posterior de P aplicando la transformación inicial.

```
X_ = as.vector(X)

Y = exp(X_)/(1 + exp(X_) )#log(X_/(1-X_))

par(mfrow = c(1,2))
plot(Y, type = "l", main = "Traza", xlab = "Iteraciones", ylab = "", col = "blue")
hist(Y, main = "Histograma", col = "blue", xlab = "Distribución" )
```



Simulación con JAGS

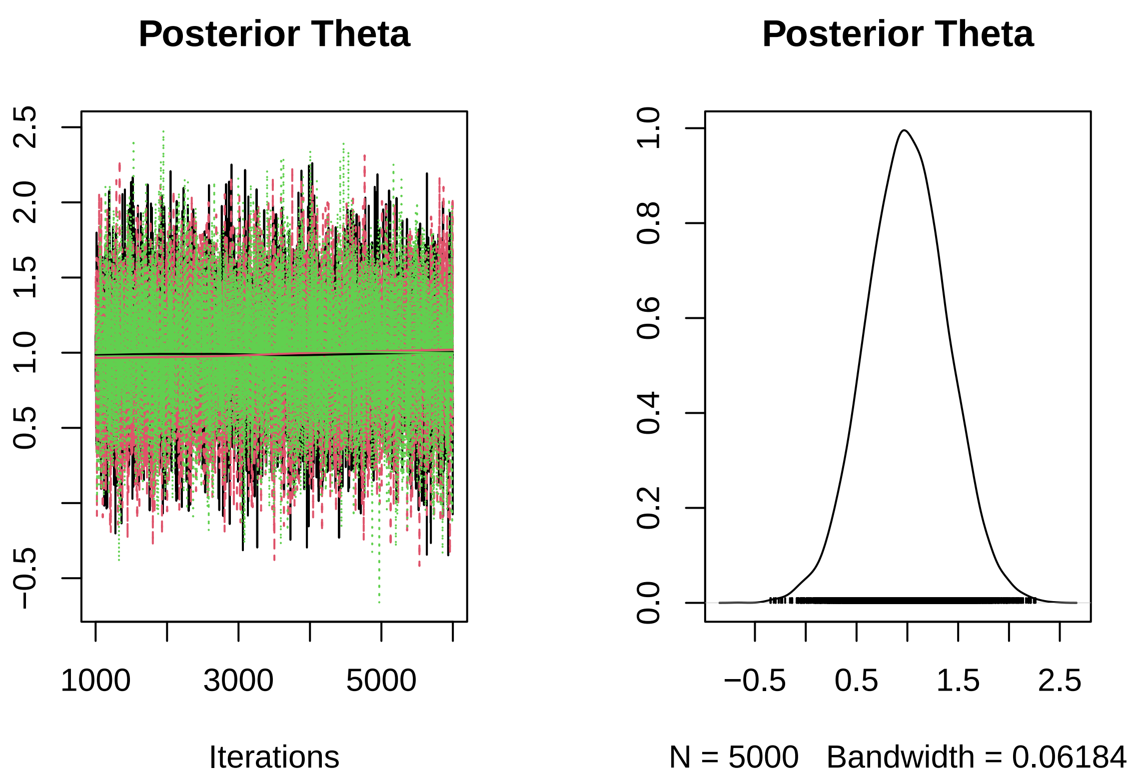
Se puede simular con ayuda de JAGS la distribución. No supe como indicarle que tomara un valor en particular (y = 7) pero sí se puede generar una muestra de la distribución.

Posterior θ

```r
data <- list(
  n = 10,
  theta0 = 1,
  tau2inv = 1/.16
)

param <- c("theta")

fit <- jags.model("E1_T4.bug", data,n.chains=3)
```

```
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
Graph information:
   Observed stochastic nodes: 0
   Unobserved stochastic nodes: 2
   Total graph size: 9

Initializing model
```

```r
update(fit,1000)

sample <- coda.samples(fit, param, n.iter=5000, thin=1)

plot(sample,main = "Posterior Theta")
```



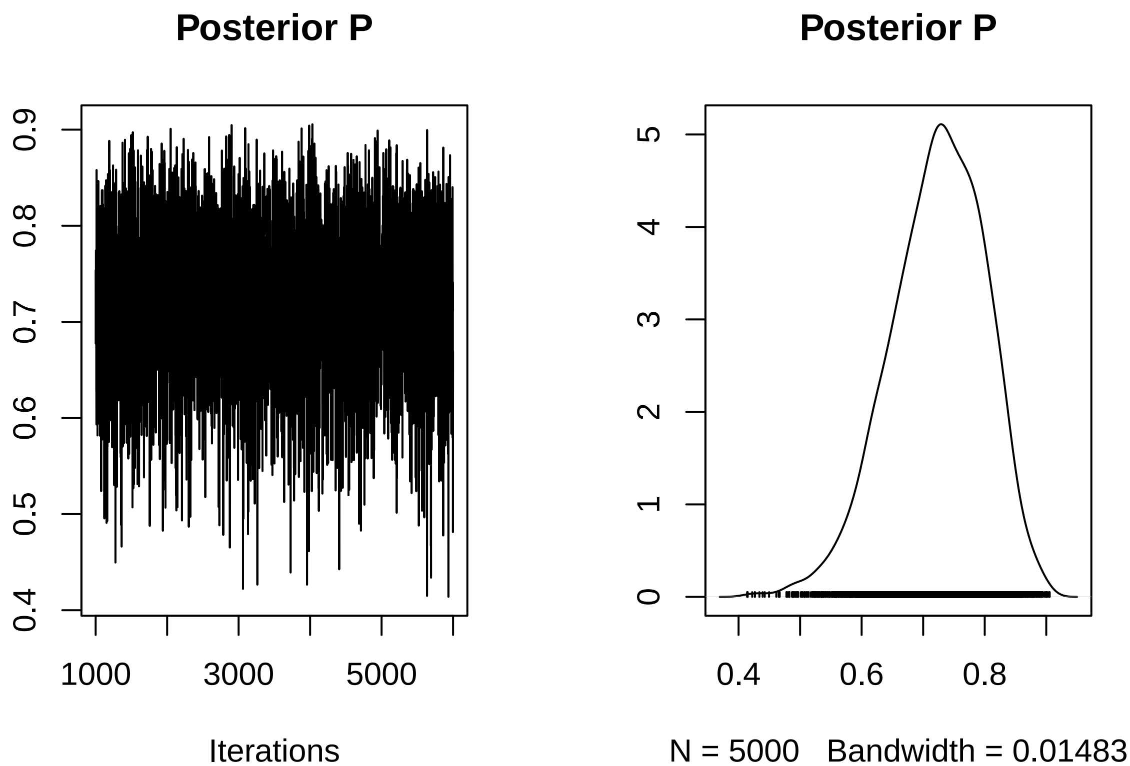
```r
summary(sample)
```

```

Iterations = 1001:6000
Thinning interval = 1
Number of chains = 3
Sample size per chain = 5000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

          Mean             SD       Naive SE Time-series SE
      0.996597       0.399223       0.003260       0.003227

2. Quantiles for each variable:

  2.5%    25%    50%    75%  97.5%
0.2164 0.7291 0.9950 1.2659 1.7824
```

Posterior P

De igual manera, con las muestras ya obtenidas de la simulación anterior puedo aplicar la transformación para obtener la distribución posterior de P.

```r
Y = exp(sample[[1]])/(1 + exp(sample[[1]]))

plot(Y, main = "Posterior P")
```



Posterior P
0.4
0.5
0.6
0.7
0.8
0.9
1000
3000
5000
Iterations
Posterior P
0
1
2
3
4
5
0.4
0.6
0.8
N = 5000 Bandwidth = 0.01483

Muestreo de Gibbs

2. Sea $X_1, ..., X_n$ una muestra aleatoria de una distribución Normal truncada, definida en X >0, $X \sim NormalTrunc(\mu, \sigma)I(0, \infty)$. Considere las distribuciones prior $\mu \sim Normal(\mu_0, \tau_0^2)$ y $\sigma^2 \sim InvGamma(a, b)$.

Use el algoritmo de muestreo de Gibbs para estimar la distribución posterior de μ y σ^2. Simule un conjunto de datos para ejemplificar.

Adicionalmente, incluye la estimación usando JAGS.

Respuesta