Tareita 18

Simulación estocástica

Marco Antonio Andrade Barrera

17 de mayo de 2018

Tenemos $\boldsymbol{x} = \{x1 = z_1 + z_2 = 125, x_2 = 18, x_3 = 20, x_4 = 34\} \sim Mult_4(\boldsymbol{x}|1/2, \theta/4, 0.25(1-\theta), 0.25(1-\theta), \theta/4)$, con $\theta \in (0,1)$ desconocido, donde z_1 y z_2 no son observados y queremos encontrar un estimador $\hat{\theta}$. De acuerdo al desarrollo visto en clase, podemos obtener el estimador usando el siguiente algoritmo. Dado un $\theta^{(0)}$ la transición está dada por

$$\theta^{(j+1)} = \frac{125 \cdot \frac{\theta^{(j)}}{\theta^{(j)}+2} + 34}{125 \cdot \frac{\theta^{(j)}}{\theta^{(j)}+2} + 72}$$

Debería converger en 7 pasos.

En el siguiente bloque de código se implementa el algoritmo, dado un θ_0 y un criterio de convergencia, donde el algoritmo se detendrá cuando la diferencia entre dos estimaciones consecutivas sea menos a un umbral u.

```
algo <- function(t0=0.01,u=0.00001){
  theta <- list()
  theta[[1]] <- t0
  i <- 2
  repeat{
    theta[[i]] <- (125*theta[[i-1]]/(theta[[i-1]] + 2) + 34) / (125*theta[[i-1]]/(theta[[i-1]] + 2) + 72
    if(abs(theta[[i]] - theta[[i-1]]) < u) break
    i <- i +1
  }
  unlist(theta)
}
```

```
#Ejecutar el algoritmo con theta inicial igual a 0.01 y umbral igual a 0.00001
algo()
```

```
## [1] 0.0100000 0.4767418 0.6044178 0.6237969 0.6264190 0.6267680 0.6268144
## [8] 0.6268206
```

Note que efectivamente, en la iteración 7 se obtiene la convergencia fijada bajo el criterio antes mencionado.

Como un segundo ensayo, se implementará el siguiente algoritmo (remplazando la expresión de la esperanza verdadera de z_2 por una estimación mediante la generación de muestras de una binomial)

1. Generar $x_1, \ldots, x_{1000} \sim bin(x|\frac{\theta^{(j)}}{\theta^{(j)}+2})$
2. $\theta^{(j+1)} = \frac{\bar{x}^{(j)}+34}{\bar{x}^{(j)}+72}$

```
algo <- function(t0=0.01,u=0.00001){
  theta <- list()
  theta[[1]] <- t0
  i <- 2
  repeat{
    x <- rbinom(n = 1000,size = 125,theta[[i-1]]/(theta[[i-1]] + 2))
    xbar <- mean(x)
    theta[[i]] <- (xbar + 34) / (xbar + 72)
    if(abs(theta[[i]] - theta[[i-1]]) < u) break
```

```
    i <- i +1
  }
  unlist(theta)
}
```

```
set.seed(62)
#Ejecutar el NUEVO algoritmo con theta inicial igual a 0.01 y umbral igual a 0.00001
algo()
```

```
##  [1] 0.0100000 0.4767138 0.6056660 0.6241977 0.6269792 0.6268327 0.6277575
##  [8] 0.6270598 0.6266310 0.6259548 0.6272207 0.6269023 0.6269939 0.6268657
## [15] 0.6261609 0.6270488 0.6265209 0.6265540 0.6267227 0.6271878 0.6258222
## [22] 0.6259732 0.6261682 0.6262160 0.6267924 0.6262675 0.6273341 0.6266714
## [29] 0.6270524 0.6275459 0.6272317 0.6266237 0.6270488 0.6277575 0.6265246
## [36] 0.6267997 0.6262087 0.6265723 0.6267447 0.6264989 0.6273596 0.6268730
## [43] 0.6267007 0.6275313 0.6273998 0.6268950 0.6272280 0.6265356 0.6260947
## [50] 0.6275386 0.6264292 0.6261535 0.6267960 0.6264952 0.6273925 0.6255863
## [57] 0.6264659 0.6272975 0.6261829 0.6265026 0.6272280 0.6274583 0.6269865
## [64] 0.6274802 0.6270122 0.6268693 0.6262785 0.6272134 0.6271622 0.6270195
## [71] 0.6268327 0.6277903 0.6273085 0.6268913 0.6272646 0.6267484 0.6264365
## [78] 0.6264769 0.6267557 0.6268254 0.6269865 0.6266457 0.6259622 0.6267850
## [85] 0.6267667 0.6274364 0.6269463 0.6269463
```

En este caso, se requieren muchas más iteraciones para satisfacer el criterio de convergencia dado. No obstante, la estimación es muy cercana al valor real (0.626821) con una diferencia del orden de diez milésimas.