

Práctica 1: movimiento Browniano

Meta

Con base en el código visto en clase [1] contabiliza el número de ocasiones en que la posición final de la partícula al terminar una caminata se encuentra en el origen en función al número de pasos de la caminata o el número de repeticiones del experimento.

Desarrollo del código

La función `npos` es la función que registra la trayectoria de una partícula en función al cambio de posición en una dirección, al terminar esta caminata se evalúa si el vector de posición final `pos` es igual al vector `origen`, en caso de ser correcto el vector lógico regreso guarda un valor verdadero, en el caso contrario guarda el valor falso. La paralelización en esta práctica se realizó usando el paquete `doParallel` y para la creación de gráficas se utilizó el paquete `ggplot2`.

```
> npos<- function(r) {
+   pos <- rep(0, dimension)
+   for (t in 1:duracion) {
+     cambiar <- sample(1:dimension, 1)
+     cambio<-1
+     if (runif(1) < 0.5) {
+       cambio <- -1
+     }
+
+     pos[cambiar] <- pos[cambiar] + cambio
+   }
+
+   if(all(pos==origen)){
+     return(TRUE)}else{return(FALSE)}
+ }
> suppressMessages(library(doParallel))
> registerDoParallel(makeCluster(detectCores() - 1))
>
> datos <- data.frame()
> for (replica in 1:7){
+ for (duracion in seq(100,400,100)){
+ for(repetir in seq(100,400,100)){
+ for (dimension in 1:8) {
+   origen<-rep(0,dimension)
+
+   regreso <- foreach(r=1:repetir,.combine=c)%dopar% npos(r)
+   res<-cbind(replica,duracion,repetir,dimension,sum(regreso))
+   datos <- rbind(datos, res)
```

```
+ }  
+ }  
+ }  
+ }
```

Resultados

En la Figura 1 se muestra el número de ocasiones en que la partícula regresa al punto de partida o el origen en función al número de pasos y al número de dimensiones en el espacio en el que se encuentra la partícula. Es evidente la disminución del número de regresos al origen al aumentar el número de dimensiones, inclusive este número de regresos es cero en la mayoría de las ocasiones. Esto tiene una explicación simple, en espacios unidimensionales sólo se tienen dos posibles movimientos: hacia números negativos o hacia números positivos, por lo que es muy probable que la partícula regrese al origen más de una vez; sin embargo, conforme el número de dimensiones aumenta el número de movimientos posibles y de combinaciones de movimientos también lo hace disminuyendo la probabilidad de que regrese al origen. Si el número de pasos aumenta, la partícula tiene una mayor probabilidad de alejarse más del origen, por lo que la cantidad de regresos al origen disminuye. Se observa que sólo aumenta dicho valor, cuando se realiza un experimento con más repeticiones.

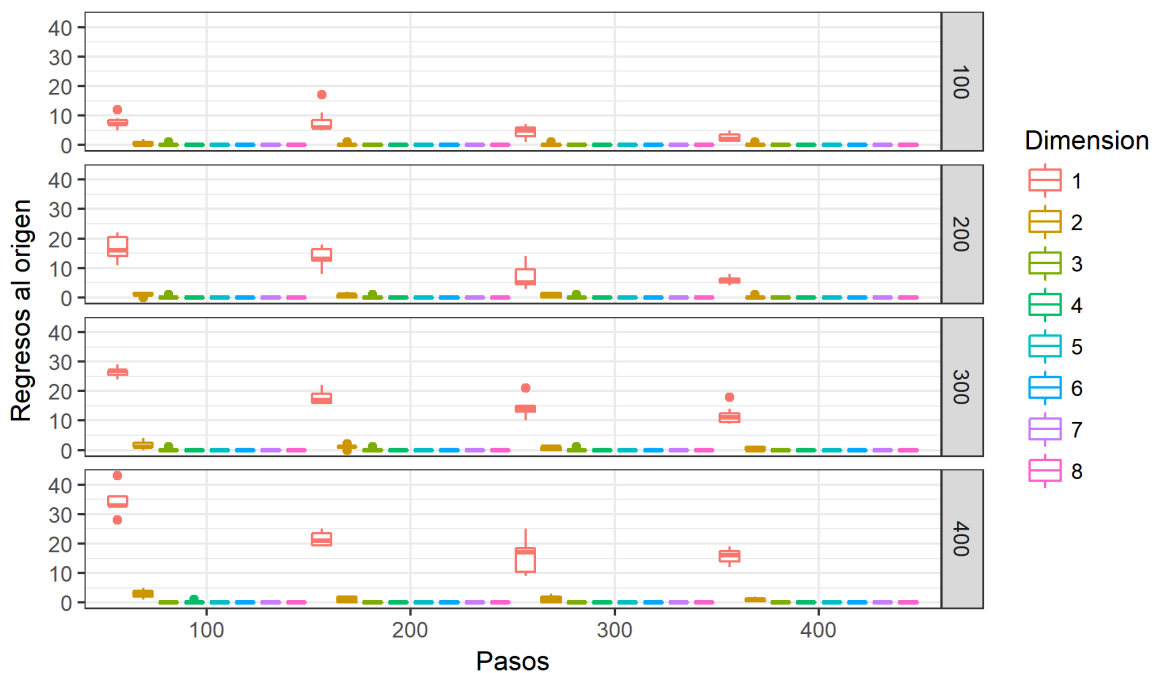


Figura 1 Número de regresos al origen en función al número de pasos y al número de dimensión

Reto 1

Meta

Estudiar el tiempo de ejecución de una caminata en función al largo de la caminata y al número de dimensiones.

Desarrollo del código

Para cumplir con el objetivo se añadieron dos medidores de tiempo `ti` y `tf` al inicio y al final de la caminata, para poder medir el tiempo que tarda una caminata mediante una resta de estos dos tiempos.

```
> repetir<-200
> suppressMessages(library(doParallel))
> registerDoParallel(makeCluster(detectCores() - 1))
>
> datos <- data.frame()
> for(replica in 1:15){
+   for (duracion in seq(100,400,100)){
+     for (dimension in 1:8){
+       origen<-rep(0,dimension)
+       ti<-Sys.time()
+       regreso <- foreach(r=1:repetir,.combine=c)%dopar% npos(r)
+       tf<-Sys.time()
+       t<-difftime(tf,ti,units="secs")
+
+       res<-cbind(replica,duracion,dimension,t)
+       datos <- rbind(datos, res)
+     }
+   }
+ }
```

Resultados

En la Figura 2 se observa que la duración de la caminata es directamente proporcional a número de pasos de la caminata, y que el número de dimensiones no parece tener una influencia en el aumento de este lapso. Esto tiene una explicación sencilla puesto que durante un paso de la caminata se modifica la posición en una sola dimensión, la cual es tomada al azar por la instrucción `cambiar <- sample(1:dimension, 1)`, por lo que no importa la cantidad de dimensiones en el tiempo de ejecución del código paralelizado.

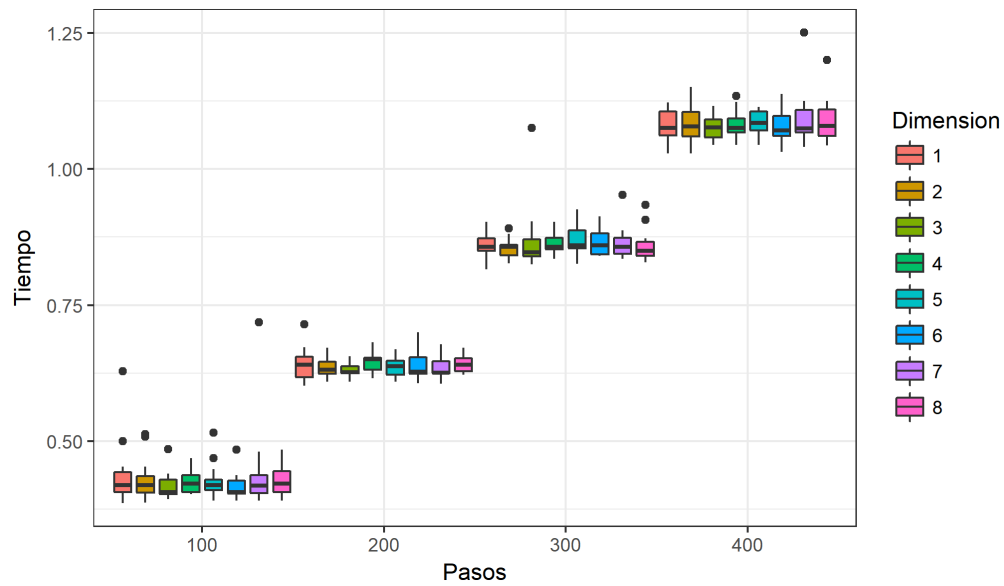


Figura 2 Variación del tiempo en función a la dimensión y al largo de la trayectoria

Referencias

[1] Elisa.dyndns-web.com. (2017). P1 — R paralelo — Schaeffer. [online] Available at: <http://elisa.dyndns-web.com/teaching/comp/par/p1.html> [Accessed 31 Oct. 2017].