

Búsqueda local

Edson Raúl Cepeda Márquez

18 de marzo de 2019

1. Objetivo

El objetivo principal de esta práctica es buscar los máximos locales de una función mediante un modelo de comparación de vecinos, también se visualiza como proceden 15 replicas del experimento. En esta práctica se hace uso del material de apoyo y del código escrito en el lenguaje de programación R [1] disponible en la página [2] de la Dra. Elisa Schaeffer.

2. Desarrollo

La búsqueda local del máximo de la función consiste en generar distintos vecinos alrededor de un punto inicial aleatorio sobre la función, se compara cual de los vecinos del punto es el más cercano a un valor máximo de la función y se cambia de posición. Este se sigue reemplazando con valores cada vez más cercanos al máximo lo que garantiza que en un determinado paso se alcance el máximo verdadero de la función. Para conseguir esto primero se modifica la búsqueda local en el código para encontrar máximos. Para esto se invierten los valores limite superiores e inferiores de la función.

```
1 g <- function(x, y) {  
2   return((((x + 0.5)^4 - 30 * x^2 - 20 * x + (y + 0.5)^4 - 30 * y^2 - 20 * y)/100))  
3 }  
4 low <- -3 #Limite inferior invertido  
5 high <- -low #Limite superior invertido
```

Se establece un número máximo de pasos y se establece la magnitud con la que estos avanzan.

```
1 tmax = 30 #Numero de pasos  
2 step <- 0.3 #Magnitud de los pasos
```

Se especifica el número de réplicas del experimento y se genera una posición aleatoria entre los valores limites de la función, así se empieza la búsqueda siempre dentro de la función.

```
1 for(rep in 1:15) {  
2   curr <- data.frame(x= runif(1, low, high), y = runif(1, low, high)) #Nueva posicion sobre la funcion  
3   best <- curr
```

La variable *best* sirve para reemplazar los mejores valores en cada iteración y así avanzar hacia el máximo de la función. Seguido se itera el número de pasos antes establecido y se genera un cambio en la primera posición.

```
1   for (tiempo in 1:tmax) {  
2     delta <- runif(2,0,step) #Delta x, Delta Y  
3     vecinos <- data.frame(x = numeric(), y = numeric())
```

La variable *vecinos* sirve para guardar cada uno de los vecinos a comparar en próximas iteraciones. Se procede a hacer el cambio de posición sobre cada uno de sus vecinos y se establecen como condición que no rebasen ni estén por debajo de los valores 3 y -3.

```

1   for(cx in c(-delta[1], 0 , delta[1])) { #Cambio en X
2     for(cy in c(-delta[2], 0 , delta[2])) { #Cambio en Y
3       nx = curr[1] + cx #Nuevo x
4       if(nx <= 3 && nx >= -3) { #Condiciones X
5         ny = curr[2] + cy #Nuevo y
6         if(ny <= 3 && ny >= -3) { #Condiciones Y
7           vecinos <- rbind(vecinos, data.frame(x = nx, y = ny))
8         }
9       }
10    }
11  }

```

Ahora se compara los vecinos para encontrar el mejor y cambiar la posición sobre la función.

```

1   vecinos$g = g(vecinos$x, vecinos$y)
2   curr = vecinos[which.max(vecinos$g),1:2] #Maximo
3   if (g(curr[1], curr[2]) > g(best[1], best[2])) { # Reemplazar mejor maximo
4     best <- curr
5   }

```

Por último se realizan las visualizaciones como gráficas de proyección plana. Se seleccionan tres réplicas en las que la posición inicial es favorable para su comparación y se generan las gráficas cada cinco pasos hasta llegar al máximo. El resto de las replicas pueden ser encontradas en forma de GIF en el repositorio [3] de este mismo documento.

3. Resultados y conclusiones

En estas proyecciones las zonas rojas representan los valores mínimos de la función y las zonas amarillas representan los valores máximos de la función. El punto de color azul con blanco es una representación de como avanza la búsqueda hacia el máximo de la función.

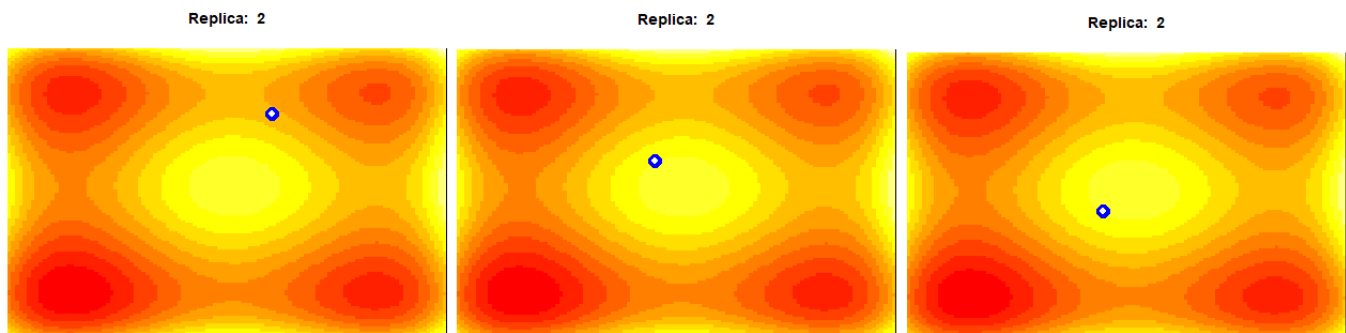


Figura 1: Búsqueda local de máximos en replica dos

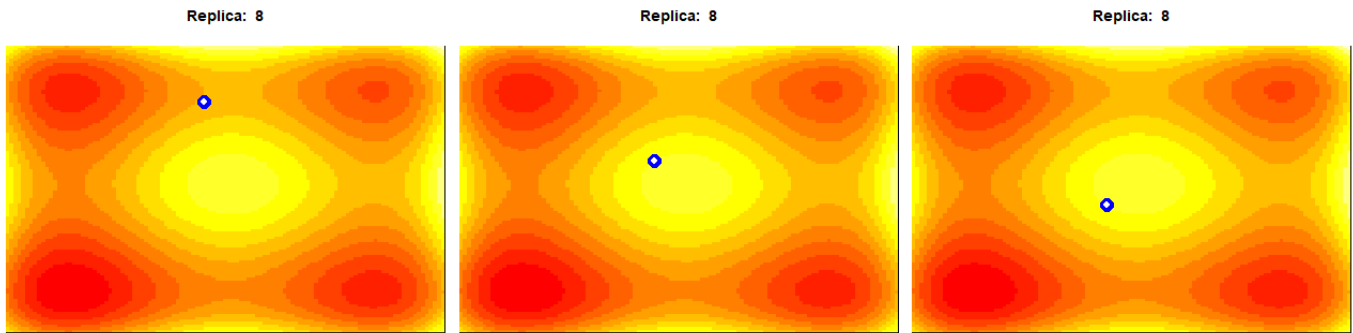


Figura 2: Búsqueda local de máximos en replica ocho

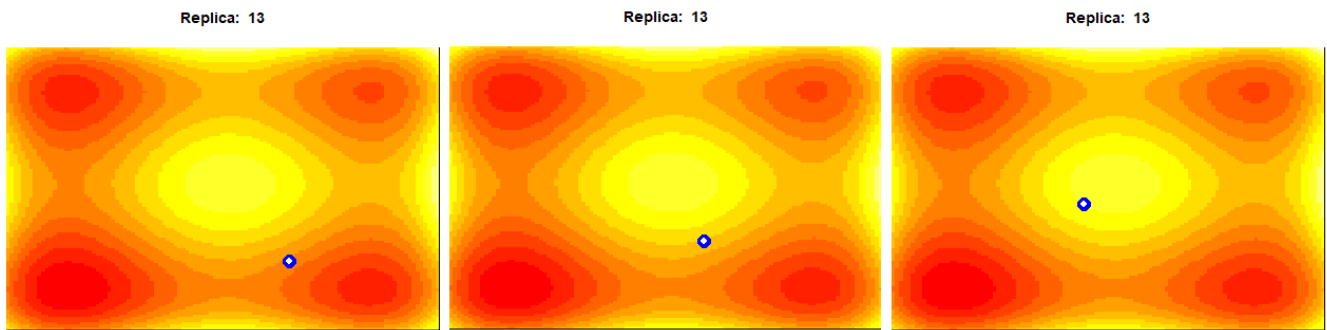


Figura 3: Búsqueda local de máximos en replica 13

Como se puede observar cada una de las réplicas llegan al mismo punto máximo. Lo que es importante destacar es que las réplicas llegan al punto máximo en un distinto número de pasos, esto se debe a que ninguna tiene el mismo punto inicial de partida, es decir, puede tomar más o menos pasos en llegar al máximo dependiendo de donde sea la posición inicial sobre la función. También es importante mencionar que muchas de las veces las réplicas alcanzan el punto máximo antes de completar el número de pasos establecido.

Referencias

- [1] R: R Project, 2019
<https://www.r-project.org/>
- [2] Satu Elisa Schaeffer: Práctica 7: búsqueda local, 2019
<https://elisa.dyndns-web.com/teaching/comp/par/p7.html>
- [3] Edson Raúl Cepeda Márquez, P7, 2019
<https://sourceforge.net/projects/systemssimulation/>