

Programación Estadística: Ascenso a la cordillera

Adrián Sosa

Universidad Veracruzana

Ascenso a la cordillera

El ascenso a la colina es un método simple de optimización local que “sube” la colina hasta se encuentra un óptimo local (asumiendo una meta de maximización). El método funciona mediante la búsqueda iterativa de nuevas soluciones dentro del entorno de las soluciones, adoptando nuevas soluciones si son mejores, como se muestra en el pseudocódigo del algoritmo. El propósito del cambio de función es producir una solución, realizando una búsqueda completa en todo el vecindario o aplicando un pequeño cambio aleatorio en los valores de la solución actual. Cabe señalar que si bien el algoritmo estándar de escalada es determinista, cuando se utilizan cambios aleatorios paros perturbando una solución, se logra un comportamiento estocástico. Por esta razón en ascenso a la colina es situado en el medio de la dimensión determinista / estocástica. Hay varias variantes de escalada, como la escalada más empinada

Ventajas

- Reduce el número de nodos a analizar

Desventajas

- Puede ser que encuentre una solución, pero no sea la más óptima

PSEUDOCÓDIGO

Selección: empezar desde la raíz R y seleccionar nodos hijos sucesivos hasta alcanzar un nodo hoja L . La selección describe una manera de elegir los nodos hijos, que permitan que el árbol se expanda hacia movimientos mas prometedores, que es la esencia del árbol de búsqueda Monte Carlo.

Algorithm 2 Pure hill climbing optimization method

```
1: Inputs:  $S, f, C$     ▷  $S$  is the initial solution,  $f$  is the evaluation function,  $C$  includes control
   parameters
2:  $i \leftarrow 0$                                 ▷  $i$  is the number of iterations of the method
3: while not termination_criteria( $S, f, C, i$ ) do
4:    $S' \leftarrow \text{change}(S, C)$                                 ▷ new solution
5:    $B \leftarrow \text{best}(S, S', f)$                                 ▷ best solution for next iteration
6:    $S \leftarrow B$                                               ▷ deterministic select function
7:    $i \leftarrow i + 1$ 
8: end while
9: Output:  $B$                                               ▷ the best solution
```

Figure 1: Pseudocódigo

Codificación

La codificación de Búsqueda Monte Carlo es relativamente sencilla, al hacer uso de el método de búsqueda ciega:

```
hclimbing

## function (x, Fx, lower, upper, type = "min", N = 1000, ...)
## {
##   D <- length(lower)
##   s <- matrix(nrow = N, ncol = D)
##   for (i in 1:N) s[i, ] = runif(D, lower, upper)
##   best <- fsearch(sol = s, Fx = Fx, type = type, ...)
##   best_1 <- Fx(x, ...)
##   if (type == "min") {
##     if (best_1 < best$eval)
##       return(list(sol = x, eval = best_1))
##     else hclimbing(best$sol, Fx, lower, upper, type, N, ...)
##   }
##   else {
##     if (best_1 > best$eval)
##       return(list(sol = x, eval = best_1))
##     else hclimbing(best$sol, Fx, lower, upper, type, N, ...)
##   }
## }
```

Su implementación de igual manera es sencilla:

```
N=10000                                # se define el número de muestras
limI <- -5
limS <- 5
var <- 1                                # número de variables
upper <- rep (limS,var)                 # vector con el valor mas alto de cada dimensión
lower <- rep (limI,var)                 # vector con el valor mas bajo de cada dimensión
x <- 1                                  # punto inicial

fx <- rastrigin
label <- "Rastrigin: "
min <- hclimbing(x,fx,lower,upper,type="min")
max <- hclimbing(x,fx,lower,upper,type="max")

cat(c(label, "Solución mínima: ", min$sol, " Evaluación: ", min$eval,"\n"))

## Rastrigin:  Solución mínima:  -0.00150685897096992  Evaluación:  0.000450470480126697

cat(c(label, "Solución máxima: ", max$sol, " Evaluación: ", max$eval,"\n"))

## Rastrigin:  Solución máxima:  -4.52048848383129  Evaluación:  40.3520696513191
```

Para la función de esfera seria de la siguiente manera:

```
x <- 3
fx <- sphere
label <- "Esfera: "
min <- hclimbing(x,fx,lower,upper,type="min")
max <- hclimbing(x,fx,lower,upper,type="max")

cat(c(label, "Solución mínima: ", min$sol, " Evaluación: ", min$eval,"\n"))

## Esfera: Solución mínima: -0.000197973567992449 Evaluación: 3.91935336236608e-08

cat(c(label, "Solución máxima: ", max$sol, " Evaluación: ", max$eval,"\n"))

## Esfera: Solución máxima: 4.99756193486974 Evaluación: 24.9756252928589
```