

Programación Estadística: Búsqueda en malla o cuadrícula

Adrián Sosa

Universidad Veracruzana

Búsqueda en malla o cuadrícula

Este modelo explora el espacio de búsqueda a partir de una solución de poblaciones llamada malla que se expande y contrae con la finalidad de encontrar una solución.

Existe una amplia variedad de problemas y aplicaciones que tienen las siguientes finalidades: encuentra el área mínima, el menor coste, la forma óptima, la menor resistencia, el máximo beneficio, el mayor alcance... Todos estos problemas, se engloban dentro de la categoría de Optimización de funciones y pueden ser resueltos aplicando el cálculo diferencial.

Codificación

La función de *Búsqueda en malla* es sencilla de codificar, podemos verlo a continuación:

```
## function (step, lower, upper, FUN, type = "min", ...)  
## {  
##   D <- length(step)  
##   domain <- vector("list", D)  
##   L <- vector(length = D)  
##   for (i in 1:D) {  
##     domain[[i]] = seq(lower[i], upper[i], by = step[i])  
##     L[i] = length(domain[[i]])  
##   }  
##   LS <- prod(L)  
##   s <- matrix(ncol = D, nrow = LS)  
##   for (i in 1:D) {  
##     if (i == 1)  
##       E <- 1  
##     else E <- E + L[i + 1]  
##     s[, i] <- rep(domain[[i]], length.out = LS, each = E)  
##   }  
##   fsearch(sol = s, Fx = FUN, type = type, ...)  
## }
```

Implementación

Su implementación podemos verla a continuación para la función $Fx1$ en un intervalo de -2 a 3.

$$Fx1 = x^3 + 6x - 1$$

Donde primero se construye un espacio de búsqueda con todas las posibles soluciones y después se utiliza la *Búsqueda Ciega en amplitud* para encontrar el resultado deseado.

Se definen las variables que nos ayudaran a crear el espacio de búsqueda:

```

limI <- -2
limS <- 3
steps <- 0.01
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
step  <- rep (steps,var)                # vector con el tamaño del paso para cada dimensión D

```

Ahora la función se implementa de la siguiente manera dándole los argumentos previamente definidos:

```

min <- gsearch(step,lower,upper,fx1,type="min")
max <- gsearch(step,lower,upper,fx1,type="max")

```

Para conocer los resultados podemos imprimir los valores sol y eval de las variables min y max:

```

## Solución mínima:  1.41
## Evaluación:  -6.656779

## Solución máxima:  3
## Evaluación:  8

```

Si deseamos ver los resultados gráficamente basta con graficar la función y mostrar los resultados mínimos y máximos:

