Optimización 1-dimensional

Alan Reyes-Figueroa

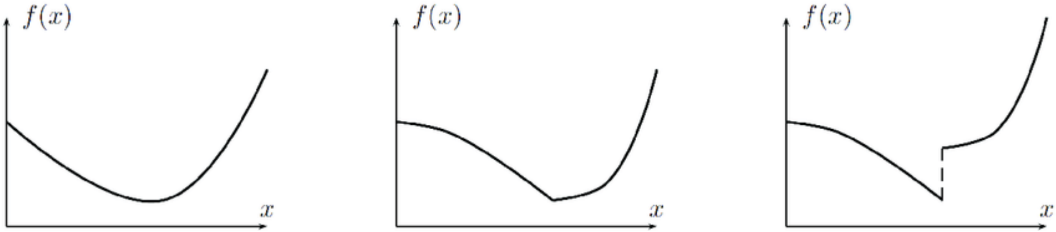
Modelación y Simulación (Aula 21) 03.octubre.2024

Algoritmos para Optimización 1D

Revisamos algunos métodos para optimización de funciones 1-dimensionales $f : [a, b] \to \mathbb{R}$.

Definición

Una función $f : [a, b] \to \mathbb{R}$ es ***unimodal*** *si sólo posee un mínimo local en el intervalo $[a, b]$. En este caso, tal mínimo es global en ese intervalo.*



Ejemplos de funciones unimodales.

Algoritmos para Optimización 1D

Existen algoritmos relativamente simples para optimizar funciones unimodales en cierto intervalo. Estos algoritmos iterativos típicamente requieren un intervalo inicial $[a, b]$ o un punto inicial $\mathbf{x}_0 \in (a, b)$ dentro de una región unimodal de f, y convergen a una solución aproximada del mínimo local.

Algunos de estos esquemas no requieren que f sea diferenciable; otros demandan que f sea de clase C^2 (*e.g.* método de Newton).

- Método de la razon aúrea (*golden ration search*),
- Interpolación parabólica (*quadratic interpolation*),
- Método de Newton,
- $\ldots$

Framework general:

- Elegir $[a, b]$ intervalo inicial ó $\mathbf{x}_0$ punto inicial,
- Definir cómo actualizar $\mathbf{x}_k$,
- Establecer un criterio de paro.

Métodos I (para hallar raíces)

Cuando la función $f : \Omega \subseteq \mathbb{R} \to \mathbb{R}$ es diferenciable, podemos adaptar los métodos para encontrar raíces $g(x) = 0$, a la derivada de f:

$$f'(x) = 0.$$

En este caso, si f es unimodal en Ω, entonces cualquiera de estos métodos converge al único mínimo de f en Ω.

- Método de bisección,
- Método *regula falsi*,
- Método de la secante,
- Método de Newton,
- Steffensen, Broyden, Halley, ...

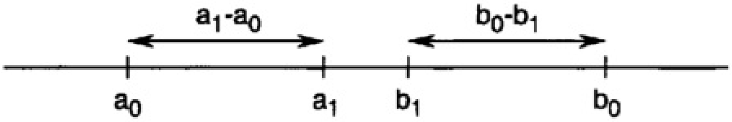
Golden Search

Método de búsqueda de la sección dorada:

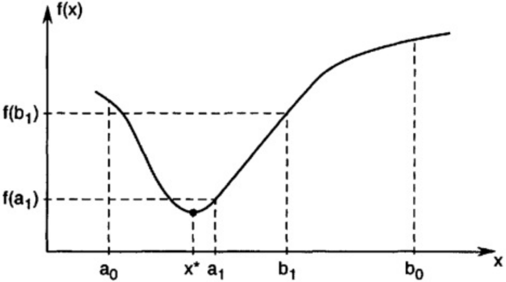
Supongamos que la función $f : [a_0, b_0] \to \mathbb{R}$ es unimodal.

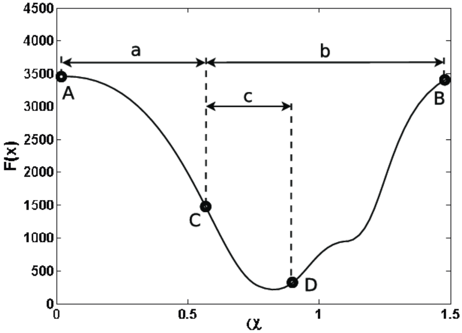
Evaluamos f en dos puntos intermedios $a_0 < a_1 < b_1 < b_0$.
Elegimos a_1, b_1 de modo que la reducción en el rango sea simétrica:

$$a_1 - a_0 = b_0 - b_1 = \rho(b_0 - a_0), \qquad \rho < \tfrac{1}{2}.$$



Golden Search





Comparamos el valor de f en los puntos internos: $f(a_1)$ y $f(b_1)$.

- Si $f(a_1) < f(b_1)$, el mínimo de f está en $[a_0, b_1]$.
- Si $f(a_1) > f(b_1)$, el mínimo de f está en $[a_1, b_0]$.

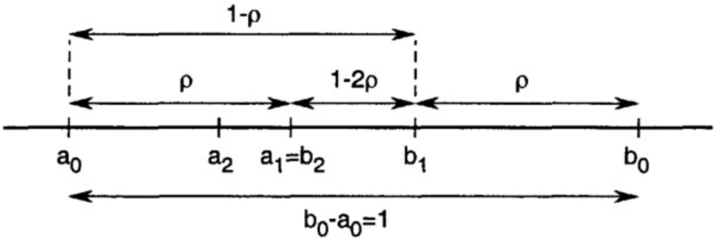
UNIVERSIDAD DEL VALLE DE GUATEMALA

Golden Search

Queremos minimizar la cantidad de evaluaciones en la función objetivo f.

En el caso $f(a_1) < f(b_1)$, sabemos que el mínimo $x^* \in [a_0, b_1]$. Como a_1 está en este intervalo y ya conocemos $f(a_1)$, hacemos que a_1 coincida con b_2.

Así, sólo necesitamos evaluar f una vez más: a saber, para calcular $f(a_2)$.



Golden Search

Sin pérdida de generalidad, asumimos que el intervalo original $[a_0, b_0]$ es de longitud unitaria. Luego, $\rho(b_1 - a_0) = b_1 - b_2$.

Como $b_1 - a_0 = 1 - \rho$ y $b_1 - b_2 = 1 - 2\rho$, entonces $\rho(1 - \rho) = 1 - 2\rho$, de donde $\rho^2 - 3\rho + 1 = 0$, y obtenemos

$$\rho = \frac{3 \pm \sqrt{5}}{2} \qquad \Longrightarrow \qquad \frac{3 - \sqrt{5}}{2} \approx 0.382 \ \ (\text{ya que } \rho < \tfrac{1}{2}).$$

Observe que $1 - \rho = \dfrac{\sqrt{5} - 1}{2} \approx 0.618$, la razón dorada.

Dividir un rango en la razón de ρ tiene el efecto de que el la razón del segmento más corto al más largo es igual a la razón de el mayor a la suma de los dos. Esta regla se llama la sección dorada.

Método de búsqueda de la sección dorada:

Algoritmo: (*Golden search*)
Inputs: $[a_0, b_0]$: intervalo de búsqueda.
Outputs: $\mathbf{x}$ mínimo global de f en $[a_0, b_0]$.

```
For k = 0,1,2,... hasta que se cumpla un criterio de paro:
    Compute
```

$$c_k = b_k - (b_k - a_k)\tfrac{\sqrt{5}-1}{2}, \quad d_k = a_k + (b_k - a_k)\tfrac{\sqrt{5}-1}{2}.$$

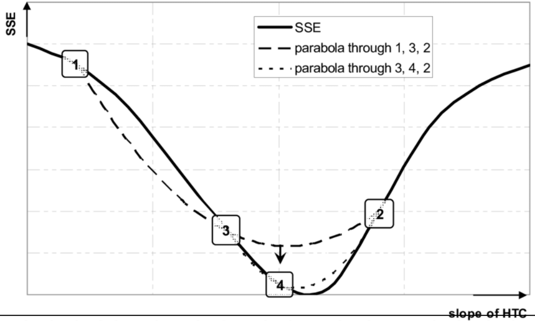
Compute $x_{k+1} = \frac{1}{2}(c_k + d_k)$.
If $(f(c_k) < f(d_k))$: Set
$a_{k+1} = a_k, b_{k+1} = d_k$.
Else: Set
$a_{k+1} = c_k, b_{k+1} = b_k$.

Return x_{k+1}.

Interpolación Parabólica

Otro método de especial simplicidad e interés es el llamado **método de interpolación parabólica** (de Newton).

Idea: Aproximar la función objetivo f por una secuencia de funciones cuadráticas $\widehat{f}_k(x) = a_k x^2 + b_k x + c_k$. Cada una de las $\widehat{f}_k$ aproxima mejor la función f en las cercanías del mínimo global x^*.



Interpolación Parabólica

En la iteración k, $k = 0, 1, 2, \ldots$, tomamos tres puntos distintos x_k, x_{k+1} y $x_{k+2} \in \mathbb{R}$. Construimos la parábola interpolante que pasa por los puntos $(x_k, f(x_k))$, $(x_{k+1}, f(x_{k+1}))$ y $(x_{k+2}, f(x_{k+2}))$, la cual se obtiene al resolver el sistema

$$\begin{pmatrix} 1 & x_k & x_k^2 \\ 1 & x_{k+1} & x_{k+1}^2 \\ 1 & x_{k+2} & x_{k+2}^2 \end{pmatrix} \begin{pmatrix} c_k \\ b_k \\ a_k \end{pmatrix} = \begin{pmatrix} f(x_k) \\ f(x_{k+1}) \\ f(x_{k+2}) \end{pmatrix}. \tag{1}$$

La solución de (1) es

$$\begin{aligned}
c_k &= -\frac{(x_{k+1}-x_{k+2})f(x_k)+(x_{k+2}-x_k)f(x_{k+1})+(x_k-x_{k+1})f(x_{k+2})}{(x_k-x_{k+1})(x_{k+1}-x_{k+2})(x_{k+2}-x_k)}, \\
b_k &= +\frac{(x_{k+1}^2-x_{k+2}^2)f(x_k)+(x_{k+2}^2-x_k^2)f(x_{k+1})+(x_k^2-x_{k+1}^2)f(x_{k+2})}{(x_k-x_{k+1})(x_{k+1}-x_{k+2})(x_{k+2}-x_k)}, \\
a_k &= +\frac{(x_{k+1}x_{k+2}^2-x_{k+2}x_{k+1}^2)f(x_k)+(x_{k+2}x_k^2-x_kx_{k+2}^2)f(x_{k+1})+(x_kx_{k+1}^2-x_{k+1}x_k^2)f(x_{k+2})}{(x_k-x_{k+1})(x_{k+1}-x_{k+2})(x_{k+2}-x_k)}.
\end{aligned}$$

Finalmente, hacemos $x_{k+3} = -\frac{b_k}{2a_k}$, el mínimo de la parábola $\widehat{f}_k$.

Interpolación Parabólica

Algoritmo: (*Parabolic interpolation*)
Inputs: $f : \mathbb{R}^n \to \mathbb{R}$ unimodal, $x_0, x_1, x_2 \in [a, b]$ tres puntos distintos.
Outputs: $\mathbf{x}$ mínimo global de f en $[a, b]$.

```
For k = 0,1,2,... hasta que se cumpla un criterio de paro:
    Solve the parabolic interpolation problem (1):
    Compute
```

$$
\begin{aligned}
c_k &= -\frac{(x_{k+1}-x_{k+2})f(x_k)+(x_{k+2}-x_k)f(x_{k+1})+(x_k-x_{k+1})f(x_{k+2})}{(x_k-x_{k+1})(x_{k+1}-x_{k+2})(x_{k+2}-x_k)},\\
b_k &= +\frac{(x_{k+1}^2-x_{k+2}^2)f(x_k)+(x_{k+2}^2-x_k^2)f(x_{k+1})+(x_k^2-x_{k+1}^2)f(x_{k+2})}{(x_k-x_{k+1})(x_{k+1}-x_{k+2})(x_{k+2}-x_k)},\\
a_k &= +\frac{(x_{k+1}x_{k+2}^2-x_{k+2}x_{k+1}^2)f(x_k)+(x_{k+2}x_k^2-x_kx_{k+2}^2)f(x_{k+1})+(x_kx_{k+1}^2-x_{k+1}x_k^2)f(x_{k+2})}{(x_k-x_{k+1})(x_{k+1}-x_{k+2})(x_{k+2}-x_k)}.
\end{aligned}
$$

    Compute $x_{k+3} = -\frac{b_k}{2a_k}$,
Return $\mathbf{x}_{k+3}$.

Ventajas y Desventajas

Mencionamos algunas ventajas y limitantes de los algoritmos anteriores.

Los algoritmos para hallar ceros (bisección, *regula falsi*, secante, punto fijo, Newton, . . .), cuando aplicados a la función $f'(x)$, requieren que al menos la función f objetivo sea diferenciable.

Newton y secante son los que tienen la mayor tasa de convergencia (bisección tiene convergencia lineal, falsa posición y secante tienen convergencia más que lineal, Newton tiene convergencia cuadrática). Sin embargo, presentan algunas limitantes: se pueden ciclar (no convergen); muestran convergencia lenta en zonas de curvatura plana.

Por otro lado, el algoritmo de bisección es lento, pero tiene la ventaja que (bajo las hipótesis adecuadas de función unimodal diferenciable), siempre converge.

- *Golden search* e interpolación parabólica tiene convergencias cercanas a lineal, y **son más lentos**, pero **no requieren** f **diferenciable**.
- *Golden search* **siempre converge**, toda vez f sea unimodal, aunque no sea continua.