HOJA DE FÓRMULAS OPTIMIZACIÓN.

Sofia Robayo Bonilla.

- Si $D_1 > 0$ y $D_2 > 0$ → F es estrictamente convexa.
- Si $D_1 \geq 0$ y $D_2 \geq 0$ → F es convexa.
- Si $D_1 < 0$ y $D_2 > 0$ → F es estrictamente cóncava.
- Si $D_1 \leq 0$ y $D_2 \geq 0$ → F es cóncava.
- Si $D_1 \leq 0$ y $D_2 < 0$ → No se puede determinar.

$D_1 = F_{xx}$

$D_2 = H(X) = \begin{vmatrix} F_{xx} & F_{xy} \\ F_{yx} & F_{yy} \end{vmatrix}$

Condición necesaria de 1er orden: $\nabla f(\bar{x}) = 0$

Condiciones necesarias de 2do orden: $\nabla f(\bar{x}) = 0$ y $H(x)$ es semidefinida +.

Condición suficiente 2do orden: $\nabla f(\bar{x}) = 0$ y $H(x)$ es definida +. Entonces es un mínimo local estricto de f en Ω.

- Toda función convexa es cuasi-convexa.

Direcciones factibles: d: $f(x + \alpha d) \Rightarrow \dfrac{\partial f(x)}{\partial d} = Df(x)\,d = d'\nabla f(x)$.

BÚSQUEDA DE LINEA.

PNL - Métodos optimización no restringida

Sección áurea - Algoritmo

Paso 0: Long. final de intervalo l, intervalo inicial $[a_0, b_0]$, $k = 0$
$\lambda_0 = a_0 + (1-\alpha)(b_0 - a_0)$, $f(\lambda_0)$
$\mu_0 = a_0 + \alpha(b_0 - a_0)$, $f(\mu_0)$
Paso 1:
if $b_k - a_k < l$ **then**
  Stop, $\lambda^* \in [a_k, b_k]$
else
  if $f(\lambda_k) > f(\mu_k)$ **then**
    Go to Paso 2
  else
    Go to Paso 3
  end if
end if

Paso 2 ($f(\lambda_k) > f(\mu_k)$):

$a_{k+1} = \lambda_k$, $b_{k+1} = b_k$, $\lambda_{k+1} = \mu_k$

$\mu_{k+1} = a_{k+1} + \alpha(b_{k+1} - a_{k+1})$

$f(\mu_{k+1})$, $k = k + 1$, Go to Paso 1

Paso 3 ($f(\lambda_k) < f(\mu_k)$):

$a_{k+1} = a_k$, $b_{k+1} = \mu_k$, $\mu_{k+1} = \lambda_k$

$\lambda_{k+1} = a_{k+1} + (1-\alpha)(b_{k+1} - a_{k+1})$

$f(\lambda_{k+1})$, $k = k + 1$, Go to Paso 1

$\alpha \approx 0{,}618$

Ajuste cuadrático - Algoritmo

Paso 0: Long. final de intervalo l, intervalo inicial $[x_1, x_2, x_3]$, $k = 0$
Paso 1: $x_1, f(x_1)$; $x_2, f(x_2)$; $x_3, f(x_3)$
$\hat{x}_k, f(\hat{x}_k)$
if $x_3 - x_1 < \varepsilon$ **then**
  Stop, $\hat{x}_k \in [x_1, x_3]$
else
  if $\hat{x}_k > x_2$ **then**
    Go to Paso 2
  else if $\hat{x}_k < x_2$ **then**
    Go to Paso 3
  else if $\hat{x}_k = x_2$ **then**
    Go to Paso 4
  end if
end if

Paso 2 ($\hat{x}_k > x_2$):

if $f(\hat{x}_k) \geq f(x_2)$ **then** intervalo $[x_1, x_2, \hat{x}_k]$, Go to Paso 1

if $f(\hat{x}_k) \leq f(x_2)$ **then** intervalo $[x_2, \hat{x}_k, x_3]$, Go to Paso 1

Paso 3 ($\hat{x}_k < x_2$):

if $f(\hat{x}_k) \geq f(x_2)$ **then** intervalo $[\hat{x}_k, x_2, x_3]$, Go to Paso 1

if $f(\hat{x}_k) \leq f(x_2)$ **then** intervalo $[x_1, \hat{x}_k, x_2]$, Go to Paso 1

Paso 4 ($\hat{x}_k = x_2$):

$$\hat{x}_k = \begin{cases} x_2 + \varepsilon/2 & \text{if } x_2 - x_1 < x_3 - x_2 \\ x_2 - \varepsilon/2 & \text{if } x_2 - x_1 > x_3 - x_2 \end{cases}, \text{ Go to Paso 1}$$

$$\hat{x}_k = \frac{b_{23} f_1 + b_{31} f_2 + b_{12} f_3}{2(a_{23} f_1 + a_{31} f_2 + a_{12} f_3)}$$

donde $b_{ij} = x_i^2 - x_j^2$

$a_{ij} = x_i - x_j$

Cabe resaltar que dichos algoritmos son sacados de las diapositivas de clase, se hace la aclaración para evitar problemas de plagio.

Método de Newton - Algoritmo

Paso 0: Punto inicial x_0, $k = 0$, error ε
Paso 1:
$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$
if $x_{k+1} - x_k < \varepsilon$ **then**
Stop, $x^* \in [x_{k+1}, x_k]$
else
Repetir
end if

$\to$ $X_{i+1} = X_i - \frac{f(X_i)}{f'(X_i)}$

$\to$ valor óptimo satisface $g(\hat{X}) = f'(\hat{X}) = 0$

$\to$ mínimo de una función

$$X_{i+1} = X_i - \frac{g(X_i)}{g'(X_i)} = X_i - \frac{f'(X_i)}{f''(X_i)}$$

MULTIVARIABLES

Optimización multidimensional

$f'(x, d) = \nabla f(x)'d$, d es una dirección de descenso.

La sol. óptima al problema $\min f'(x,d)$ s.a. $\|d\| = 1$ es

$$d = -\frac{\nabla f(x)}{\|\nabla f(x)\|}$$

Escoge dirección d y se hace búsqueda de línea en dirección d.

Método de descenso del gradiente - Algoritmo

Paso 0: Determine $\epsilon > 0$, punto inicial de búsqueda x_1, $k = 1$
Paso 1:
while $\|\nabla f(x_k)\| > \epsilon$ **do**
$d_k = -\nabla f(x_k)$
$\lambda_k = \text{argmin}_{\lambda \geq 0} f(x_k + \lambda d_k)$
$x_{k+1} = x_k + \lambda_k d_k$
$k = k + 1$
end while

Rest. de ig.

Condición 1er orden. $\nabla f(x) + \lambda_n h_n(x) = 0$

$h_n(x) = 0$.

Condición 2do orden.

$l(x, \lambda) = f(x) + \lambda h(x) = F(x) + \ldots$

$\ldots + \sum_{j=1}^{n} \lambda_j h_j(x)$

$T(X_0)\ \{y : D_h(x^*)\, y = 0\}$

Método de descenso del gradiente - Convergencia

- $\Omega = \{x^* : \nabla f(x^*) = 0\}$
- $A = MD$: mapa algorítmico del método de descenso del gradiente
- Si $f \in C^1$, D es un mapa cerrado.
- Si D y M son cerrados, A es cerrado
- Función de descenso: f
- Suponiendo que $\{x_k\}$ permanece en un conjunto compacto, el método de descenso del gradiente definido por el mapa A converge a un punto en Ω

Método de Newton

Usar aproximación cuadrática de la función en x_k:

$$q(x) = f(x_k) + \nabla f(x_k)'(x - x_k) + \frac{1}{2}(x - x_k)'H(x_k)(x - x_k)$$

Condición necesaria para un mínimo local de $q(x)$: $\nabla q(x) = 0$

$$\begin{aligned}\nabla q(x) &= \nabla f(x_k) + \frac{1}{2}\left(H(x_k)(x - x_k) + H(x_k)'(x - x_k)\right)\\ &= \nabla f(x_k) + H(x_k)(x - x_k) = 0\\ H(x_k)x &= H(x_k)x_k - \nabla f(x_k)\\ x_{k+1} &= x_k - H(x_k)^{-1}\nabla f(x_k)\end{aligned}$$

$$x_{k+1} = x_k - H(x_k)^{-1}\nabla f(x_k)$$

Suponiendo $\nabla f(\bar{x}) = 0$ y $H(\bar{x}) \succ 0$ en un mínimo local $\bar{x}$, y $f \in C^2$:

- $H(x_k) \succ 0$ para todo punto x_k *cerca de* $\bar{x}$
- $H(x_k)^{-1}$ bien definida

Equivalente a método de Newton-Raphson para resolver $\nabla f(x) = 0$ (sist. de ecuaciones no lineales):

- En cada iteración se resuelve un sistema lineal
- Sistema lineal: aproximación de primer orden
- Aproximación de primer orden de $\nabla f(x_k) = 0$:

$$\nabla f(x_k) + H(x_k)(x - x_k) = 0$$

- Se encuentra x, que es el nuevo iterado x_{k+1}

Convergencia.

$\to$ No necesariamente converge

$\to$ $H(X_k)$ puede ser singular.

$\to$ $f(X_{k+1})$ puede ser mayor que $f(X_k)$

$\to$ Convergencia: $\|X_{k+1} - \bar{X}\| \leq k_1 k_2 \|X_k - \bar{X}\|^2$

Métodos de direcciones conjugadas - Caso cuadrático

- $f(x) = c'x + \frac{1}{2}x'Hx$ con H simétrica definida positiva
- $\{d_1, \ldots, d_n\}$: H-conjugadas
- $x = x_1 + \sum_{j=1}^{n} \lambda_j d_j$
- $\phi(\lambda) = f(x) = f(x_1 + \sum_{j=1}^{n} \lambda_j d_j)$

- $x = x_1 + \sum_{j=1}^{n} \lambda_j d_j$
- $\phi(\lambda) = f(x) =$
$c'x_1 + \sum_{j=1}^{n} \lambda_j c'd_j + \frac{1}{2}\left(x_1 + \sum_{j=1}^{n} \lambda_j d_j\right)' H \left(x_1 + \sum_{j=1}^{n} \lambda_j d_j\right)$
- Por ser H-conjugadas:

$$\phi(\lambda) = c'x_1 + \frac{1}{2}x_1'Hx_1 + \sum_{j=1}^{n}\left(\lambda_j c'd_j + \lambda_j d_j'Hx_1 + \frac{1}{2}\lambda_j^2 d_j'Hd_j\right)$$

- Valor óptimo de λ_j:

$$\frac{\partial\phi(\lambda)}{\partial\lambda_j} = c'd_j + d_j'Hx_1 + \lambda_j d_j'Hd_j = 0$$

$$\lambda_j^* = -\frac{c'd_j + d_j'Hx_1}{d_j'Hd_j}$$

- Tamaño de paso óptimo en cada dirección d_j