Introducción a la Optimización

Módulo 4 : Técnicas computacionales avanzadas para modelar fenómenos sociales
Concentración en Economía Aplicada y Ciencia de Datos
ITESM

20 de septiembre de 2022

Todxs optimizamos

Introducción a la Optimización

- La optimización es una herramienta importante en las ciencias de la decisión y en el análisis de sistemas físicos.
- Primero debemos identificar un **objetivo**, esto es una medida cuantitativa del desempeño del sistema bajo estudio.
- El objetivo depende de ciertas características del sistema, llamadas **variables**.
- Con frecuencia, las variables están **restringidas** (*constrained*) en alguna manera.

Introducción a la Optimización

- El proceso de identificar el objetivo, las variables y las restricciones para un problema dado, se le conoce como **modelación**.
- Una vez que se ha formulado el modelo, podemos utilizar un *algoritmo optimización* para encontrar su solución.
- **No hay** un algoritmo de optimización universal: hay una colección de algoritmos, cada uno de los cuales se adapta a un tipo particular de problema de optimización.

Introducción a la Optimización

- Después de aplicar un algoritmo de optimización al modelo, debemos ser capaces de reconocer si ha sido exitoso en su tarea de encontrar una solución.
- En algunos casos hay expresiones matemáticas conocidas como *condiciones de optimalidad* para cerciorarse que el conjunto actual de variables es la solución del problema.
- El modelo puede ser mejorado al aplicar técnicas como el *análisis de sensibilidad* que revela la sensibilidad de la solución a los cambios en el modelo y los datos.

Formulación matemática

En términos formales, la optimización es la minimización o maximización de una función sujeta a restricciones en sus variables. En notación :

- x es el vector de variables, también conocido como *parámetros*.
- f es la *función objetivo*, una función (escalar[1]) de x que queremos maximizar o minimizar.
- c_i son funciones de restricción, que son funciones escalares de x que definen ciertas ecuaciones y desigualdades que el vector x debe satisfacer.

Usando esta notación, el problema de optimización puede ser escrito como:

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{sujeto a} \quad \begin{array}{ll} c_i(x) = 0, & i \in \mathcal{E} \\ c_i(x) \geq 0, & i \in \mathcal{I} \end{array} \tag{1}$$

donde $\mathcal{E}$ y $\mathcal{I}$ son conjuntos de índices para restricciones de igualdad y desigualdad.

[1] Es decir, una función que va de $\mathbb{R}^n$ a $\mathbb{R}$

Ejemplo

Considere el siguiente problema ,

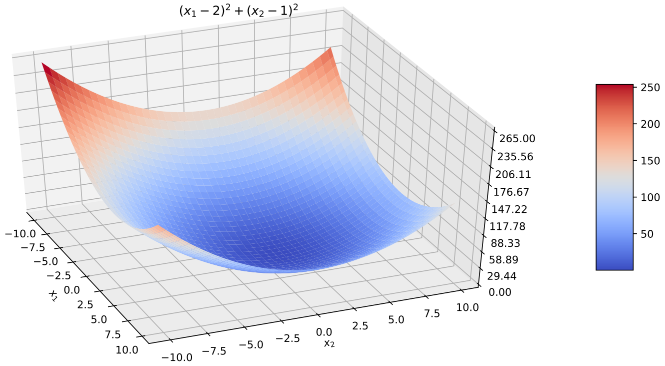
$$\min \ (x_1-2)^2+(x_2-1)^2 \quad \text{sujeto a} \quad \begin{array}{c} x_1^2-x_2 \geq 0, \\ x_1+x_2 \geq 2. \end{array} \tag{2}$$

Podemos escribir este problema en la forma (1) al definir

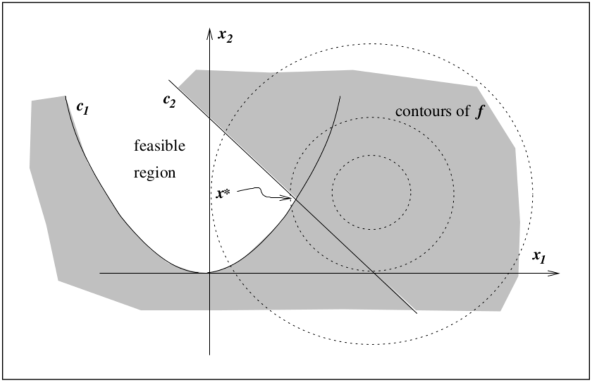
$$f(x)=(x_1-2)^2+(x_2-1)^2, \qquad x=\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$c(x)=\begin{bmatrix} c_1(x) \\ c_2(x) \end{bmatrix}=\begin{bmatrix} -x_1^2+x_2 \\ -x_1-x_2+2 \end{bmatrix}, \quad \mathcal{I}=\{1,2\}, \mathcal{E}=\emptyset$$

Conozcamos la función objetivo



Contorno de la función objetivo



- La *región factible* es el conjunto de puntos que satisfacen todas las restricciones.
- El punto x^* es la solución al problema.

Maximización

Un problema de maximización de una función $f(x)$ puede ser rescrito de acuerdo al modelo (1) como :

$$\min_{x \in \mathbb{R}^n} -f(x) \quad \text{sujeto a} \quad \begin{array}{ll} c_i(x) = 0, & i \in \mathcal{E} \\ c_i(x) \geq 0, & i \in \mathcal{I} \end{array} \tag{3}$$

Optimización discreta y continua

- Los problemas de *optimización discreta* pueden contener variables enteras, binarias y objetos variables más abstractos, como permutaciones de un conjunto ordenado. Ejemplos:
  - TSP (Travelling salesman problem).
- La característica que define a los problemas de optimización discreta es que x pertenece a un conjunto finito y numerable.
- Por su parte, el conjunto factible para los problemas de *optimización continua* es infinito no numerable, como cuando los componentes de x pueden ser números reales.

Optimización restringida y no restringida

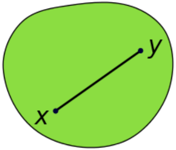
- En los problemas de *optimización no restringida* tenemos que en (1), $\mathcal{E} = \mathcal{I} = \emptyset$.
- Este tipo de problemas surgen también como problemas de optimización restringida, en los que las restricciones son reemplazadas por términos de penalización agregados a la función objetivo.
- Los problemas de *optimización restringida* surgen de modelos en los que las restricciones tienen un rol importante, como por ejemplo la restricción presupuestaria en el problema de consumidor.

Programación lineal y no lineal

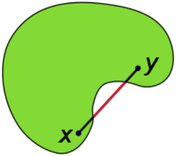
- Cuando la función objetivo y todas las restricciones son funciones lineales de x, el problema se le conoce como de *programación lineal*.
- Los problemas de *programación no lineal* son aquellos en los que al menos una restricción o la función objetivo son no lineales.

Convexidad

- El término *convexo* puede aplicarse tanto a conjuntos como a funciones.
- Un conjunto $S \in \mathbb{R}^n$ es un *conjunto convexo* si podemos trazar una línea que conecte a dos puntos en S y que se encuentre por completo en dentro de S.



(a) Convexo



(b) No Convexo

- Formalmente, para cualquier dos puntos $x \in S$ y $y \in S$, tenemos $\alpha x + (1 - \alpha)y \in S$ para todo $\alpha \in [0, 1]$

Convexidad

- La función f es una *función convexa* si su dominio S es un conjunto convexo y si para cualquier dos puntos x y y en S, se satisface la siguiente propiedad:

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y), \text{ para todo } \alpha \in [0,1] \quad (4)$$

- Decimos que f es *estrictamente convexa* si las desigualdades en (4) es estricta si $x \neq y$ y α está en el intervalo abierto $(0,1)$.
- Una función f se dice que es *concava* si $-f$ es convexa.

Algoritmos de Optimización

- Los algoritmos de optimización son iterativos.
- Estos inician con un valor inicial de la variable x y generan una secuencia de estimaciones mejoradas (llamadas *iteraciones*) hasta que terminen, con suerte, en una solución.
- La estrategia utilizada para pasar de una iteración a la siguiente es lo que distingue a los algoritmos entre sí.
- Algunos algoritmos acumulan información recopilada en iteraciones anteriores, mientras que otros usan solo información local obtenida en el punto actual.

Algoritmos de Optimización

Los algoritmos de optimización deben poseer las siguientes propiedades.

- **Robustez**: Deben tener un buen desempeño en una amplia variedad de problemas en su tipo, para todos los valores razonables del punto de inicio.
- **Eficiencia**: No deben requerir un excesivo uso de tiempo de cómputo o almacenamiento.
- **Precisión**: Deben ser capaces de encontrar una solución con precisión, sin ser demasiado sensible a los errores en los datos o a los errores de redondeo aritmético.

¿Qué es una solución?

Cuando minimizamos f, deseamos encontrar un minimizador global , un punto donde la función alcanza su valor mínimo. Una definición formal es la siguiente:

Minimizador global

Un punto x^* es un *minimizador global* si $f(x^*) \leq f(x)$ para todo x, donde x ranges over all $\mathbb{R}^n$.

El minizador global puede ser dificil de encontrar porque nuestro conocimiento de f es usualmente sólo local.
En general, encontrar un punto (maximizador o minimizador) global es computacionalmente intratable(Neumaier, 2004).
En tal caso, nos conformaremos con encontrar **óptimos locales**.

¿Qué es una solución?

La mayoría de los algoritmos son capaces de encontrar un minimizador local, el cual es un punto que alcanza el valor más pequeño de f en una vecindad. Formalmente:

Minimizador local (débil)

Un punto x^* es un *minimizador local* si hay una vecindad $\mathcal{N}$ de x^* tal que $f(x^*) \leq f(x)$ para todo $x \in \mathcal{N}$.

(Una vecindad de x^* es un conjunto abierto que contiene a x^*)
Un punto que satisface esta definición se le denomina un minimizador local débil.

¿Qué es una solución?

Un minimizador local estricto es aquel que

Minimizador local (estricto)

Un punto x^* es un *minimizador local estricto* si hay una vecindad $\mathcal{N}$ de x^* tal que $f(x^*) < f(x)$ para todo $x \in \mathcal{N}$ con $x \neq x^*$.

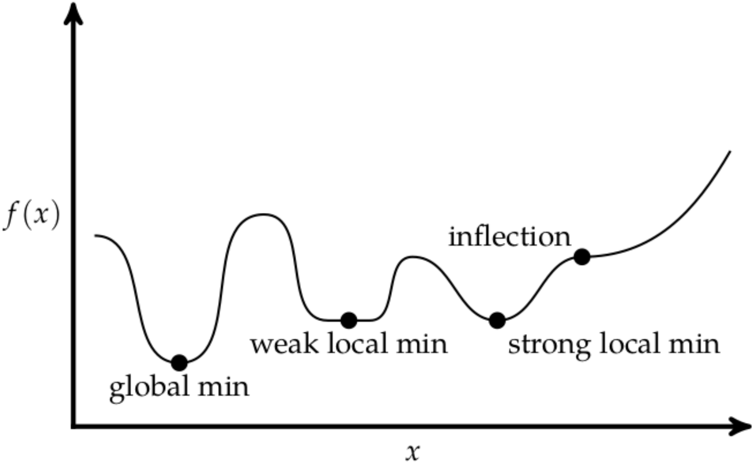


Figura: Tomado de Kochenderfer and Wheeler (2019)

Condiciones para mínimos locales (en funciones univariadas)

Asumimos que:

- La función objetivo es univariada y diferenciable.
- No hay restricciones.

Se garantiza que un punto es un mínimo local estricto si su primer derivada es igual a cero y su segunda derivada es positiva[2]

- $f'(x^*) = 0$
- $f''(x^*) > 0$

[2] Si $f'(x^*) = 0$ y $f''(x^*) < 0$ entonces x^* es un máximo local.

Condiciones **necesarias** de mínimos locales

Un punto es un mínimo local si su primera derivada es igual a cero y la segunda derivada es no negativa:

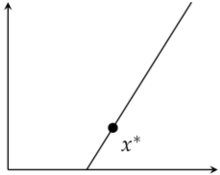
- $f'(x^*) = 0$, es la *condición necesaria de primer orden* (FONC)[3].
- $f''(x^*) \geq 0$, es la *condición necesaria de segundo orden* (SONC)

Estas condiciones se denominan *necesarias* porque todos los mínimos locales las cumplen.

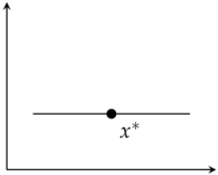
[3] Un punto que satisface la condición necesaria de primer orden se le denomina *punto estacionario*

Condiciones **necesarias** de mínimos locales

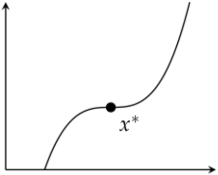
Desafortunadamente, no todos los puntos con primera derivada y segunda derivada igual a cero son mínimos locales, como se muestra a continuación:



SONC but not FONC



FONC and SONC



FONC and SONC

Caso Multivariado. Repaso de Álgebra Lineal

Dado un vector $x \in \mathbb{R}^n$, usamos x_i para denotar a la i-ésima entrada. Se asume que x es un vector *columna*:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

La transpuesta de x, denotada como x^T es un vector *fila*:

$$x = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}$$

Usamos $x \geq 0$ para indicar que todas las entradas son no negativas, esto es, $x_i \geq 0$ para todo $i = 1, 2, \dots, n$, mientras que $x > 0$ indica $x_i > 0$ para todo $i = 1, 2, \dots, n$.

Dados $x \in \mathbb{R}^n$ y $y \in \mathbb{R}^n$, el producto punto es $x^T y = \sum_{i=1}^{n} x_i y_i$.

Caso Multivariado. Repaso de Álgebra Lineal

- Dada una matriz $A \in \mathbb{R}^{m \times n}$, sus entradas las especificamos con los subíndices i,j, es decir, A_{ij}, para $i = 1, 2, \ldots, m$ y $j = 1, 2, \ldots, n$.
- La transpuesta de A, denotada como A^T, es una matriz $n \times m$ cuyos componentes son A_{ji}.
- Se dice que la matriz A es *cuadrada* si $m = n$.
- Una matriz es simétrica si $A = A^T$.

Caso Multivariado. Repaso de Álgebra Lineal

Una matriz cuadrada A es *positiva definida* si hay un escalar positivo α tal que

$$x^T Ax \geq \alpha x^T x, \text{ para todo } x \in \mathbb{R}^n$$

Es *positiva semidefinida* si

$$x^T Ax \geq 0, \text{ para todo } x \in \mathbb{R}^n$$

Podemos reconocer que una matriz simétrica es positiva definida si todos sus eigenvalores son positivos.

Eigenvectores y eigenvalores

Un vector columna $\bar{x} \in \mathbb{R}^n$ es un eigenvector de una matriz $A \in \mathbb{R}^{n \times n}$ si se cumple la siguiente expresión para un escalar λ:

$$A\bar{x} = \lambda\bar{x}$$

el escalar λ es llamado eigenvalor.

Polinomio característico

El polinomio característico de una matriz $A \in \mathbb{R}^{n \times n}$ es el polinomio de grado n en λ obtenido de expandir $\det(A - \lambda I)$

Note que este es un polinomio de grado n siempre tiene n raíces (ya sea repetidas o complejas).

Los eigenvalores son las n raíces del polinomio característico de cualquier matriz $n \times n$.

Observación

El polinimio característico $f(\lambda)$ de una matriz $A \in \mathbb{R}^{n \times n}$ es un polinomio en λ que toma la siguiente forma, donde $\lambda_1, \ldots, \lambda_n$ son eigenvalores de A:

$$\det(A - \lambda I) = (\lambda_1 - \lambda)(\lambda_2 - \lambda) \ldots (\lambda_n - \lambda)$$

Eigenvectores y eigenvalores: ejemplo

De esta forma, los eigenvalores y eigenvectores de una matriz A pueden calcularse al expandir $\det(A - \lambda I)$, igualar a cero y resolver para λ.

Considera la siguiente matriz :

$$B = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$$

La matriz $B - \lambda I$ puede ser escrita como:

$$B - \lambda I = \begin{bmatrix} 1 - \lambda & 2 \\ 2 & 1 - \lambda \end{bmatrix}$$

El determinante $\det(B - \lambda I)$ es igual a :

$$\begin{aligned} \det(B - \lambda I) &= (1 - \lambda)(1 - \lambda) - 2 * 2 \\ &= (1 - \lambda)^2 - 4 \\ &= \lambda^2 - 2\lambda - 3 \end{aligned}$$

Eigenvectores y eigenvalores: ejemplo

La expresión $\lambda^2 - 2\lambda - 3$ puede ser reescrita como $(3-\lambda)(-1-\lambda)$. Al igualar esta expresión a cero,

$$(3-\lambda)(-1-\lambda) = 0$$

tenemos que los eigenvalores (las raices del polinomio) son 3 y -1.

Derivadas en múltiples dimensiones: Gradiente

- El *gradiente* es la generalización de la derivada en funciones multivariadas.
- Captura la pendiente local de la función, permitiéndonos predecir el efecto de dar un pequeño paso desde un punto en cualquier dirección.
- El gradiente apunta en la dirección del ascenso más pronunciado del *hiperplano* tangente.

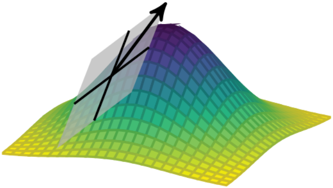


Figura: Tomado de Kochenderfer and Wheeler (2019)

Derivadas en múltiples dimensiones: Gradiente

El *gradiente* de f en $\mathbf{x}$ es denotado como $\nabla f(\mathbf{x})$ y es un vector.
Cada entrada del vector es la *derivada parcial*[4] de f con respecto a cada variable:

$$\nabla f(\mathbf{x}) = \left[\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \ldots, \frac{\partial f(\mathbf{x})}{\partial x_n}\right] \tag{5}$$

Cálculo de gradiente en un punto particular

Calcula el gradiente de $f(\mathbf{x}) = x_1 x_2$ en $\mathbf{c} = [2, 0]$.

$$f(\mathbf{x}) = x_1 x_2$$

$$\nabla f(\mathbf{x}) = \left[\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}\right] = [x_2, x_1]$$

$$f(\mathbf{c}) = [0, 2]$$

[4] La derivada parcial de una función con respecto a una variable es la derivada asumiendo que el resto de las variables se mantienen constantes. Se denota como $\frac{\partial f}{\partial x}$

Derivadas en múltiples dimensiones: Hessiano

El *Hessiano* de una función multivariada es una matriz que contiene todas las segundas derivadas con respecto a la entrada.

La segunda derivada captura información acerca de la curvatura local de la función.

$$\nabla^2 f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} \\ & \vdots & & \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_n} \end{bmatrix} \tag{6}$$

Condiciones para mínimos locales (en funciones multivariadas)

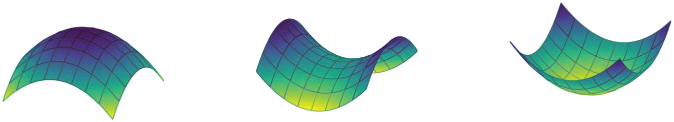
La siguientes condiciones son necesarias para que $\mathbf{x}$ sea un mínimo local de f:

- $\nabla f(\mathbf{x}) = 0$, es la *condición necesaria de primer orden* (FONC).
- $\nabla^2 f(\mathbf{x})$ sea positiva semi definida, es la *condición necesaria de segundo orden* (SONC).

FONC y SONC son generalizaciones del caso univariado. FONC no dice que la función no está cambiando en $\mathbf{x}$.

Condiciones para mínimos locales (en funciones multivariadas)

Las siguientes figuras son ejemplos de funciones multivariadas donde se satisface la FONC.



A *local maximum*. The gradient at the center is zero, but the Hessian is negative definite.

A *saddle*. The gradient at the center is zero, but it is not a local minimum.

A *bowl*. The gradient at the center is zero and the Hessian is positive definite. It is a local minimum.

Figura: Tomado de Kochenderfer and Wheeler (2019)

Condiciones para mínimos locales: ejemplo

Considera la función de Rosenbrock:

$$f(\mathrm{x}) = (1 - x_1)^2 + 5(x_2 - x_1^2)^2$$

El punto $(1, 1)$ satisface las condiciones de primer y segundo orden?

El gradiente es:

$$\nabla f(\mathrm{x}) = \begin{bmatrix} \dfrac{\partial f}{\partial x_1} \\ \dfrac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} -2(1 - x_1) + 20(x_2 - x_1^2)x_1 \\ 10(x_2 - x_1^2) \end{bmatrix}$$

$$= \begin{bmatrix} 2(10x_1^3 - 10x_1x_2 + x_1 - 1) \\ 10(x_2 - x_1^2) \end{bmatrix}$$

Condiciones para mínimos locales: ejemplo

Sustituyendo el punto $(1, 1)$ en el gradiente, tenemos $\nabla f(\begin{bmatrix}1 & 1\end{bmatrix} = \begin{bmatrix}0 & 0\end{bmatrix}$ de manera que satisface las condiciones de primer orden.
El hessiano de la función es

$$\nabla^2 f(\mathrm{x}) = \begin{bmatrix} \frac{\partial^f}{\partial x_1 \partial x_1} & \frac{\partial^f}{\partial x_1 \partial x_2} \\ \frac{\partial^f}{\partial x_2 \partial x_1} & \frac{\partial^f}{\partial x_2 \partial x_2} \end{bmatrix} = \begin{bmatrix} 2(30x_1^2 - 10x_2 + 1) & 2(-10x_1) \\ 10(-2x_1) & 10 \end{bmatrix}$$

Sustituyendo $(1, 1)$ en el hessiano,

$$\nabla^2 f(\mathrm{x}) = \begin{bmatrix} 42 & -20 \\ -20 & 10 \end{bmatrix}$$

Condiciones para mínimos locales: ejemplo

Que es positiva definida, de forma que cumple con la condición de segundo orden.

$$\nabla^2 f(x) - \lambda I = \begin{bmatrix} 42-\lambda & -20 \\ -20 & 10-\lambda \end{bmatrix}$$

El determinante $\det(\nabla^2 f(x) - \lambda I)$ es igual a :

$$\begin{aligned} \det(\nabla^2 f(x) - \lambda I) &= (42-\lambda)(10-\lambda) - (-20) * (-20) \\ &= 420 - 42\lambda - 10\lambda + \lambda^2 - 400 \\ &= \lambda^2 - 52\lambda + 20 \end{aligned}$$

Con raices $\lambda_1 = 51{,}61249695$ y $\lambda_2 = 0{,}38750305$

Métodos de búsqueda directa

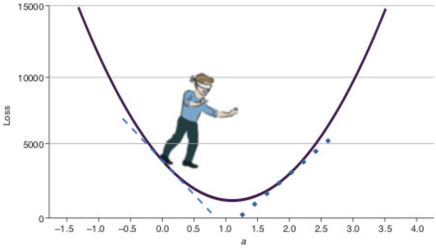


Figura: Tomado de Durr y Sick (2020). *Probabilistic deep learning*

Métodos de búsqueda directa

Elige una dirección p_k y busca en esa dirección desde algún valor x_k algún punto x_{k+1} tal que $f(x_{k+1}) < f(x_k)$.

La distancia a moverse puede encontrarse resolviendo:

$$\min_{\alpha > 0} \quad f(x_k + \alpha p_k)$$

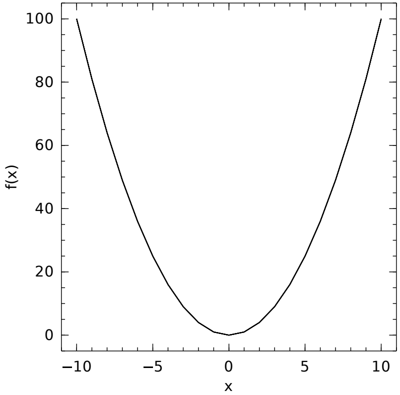


Figura: $f(x) = x^2$

Gradiente descendente

¿Cómo calculamos α?
Mediante un método iterativo: Gradiente descendente.
La iteración está dada por:

$$x_{k+1} = x_k + \alpha_k p_k$$

Donde $\alpha \in \mathbb{R}^+$ es e tamaño de paso.

Normalmente se requiere que p_k sea una dirección descendente, $p_k^t \nabla f_k < 0$, esto es que, como en el perreque, nos lleve hacia abajo.

Si elegimos que $p_k^t = -1$, es decir $-\nabla f_k$, da lugar al método de gradiente descendente.

¿Qué es ∇f_k?

Funciones en múltiples dimensiones

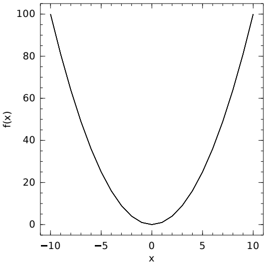


Figura: $f(x) = x^2$

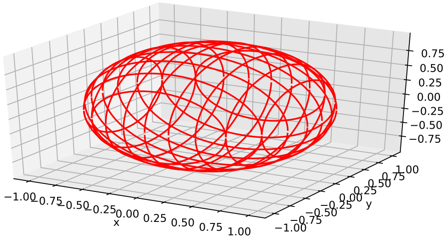


Figura: $f(x) = x^2 + y^2$

Derivadas en múltiples dimensiones

- La *derivada direccional* $\nabla_s f(\mathbf{x})$ de una función multivariada f es la tasa de cambio instantánea de $f(\mathbf{x})$ cuando $\mathbf{x}$ se mueve con dirección $\mathbf{s}$.
- La derivada direccional puede ser calculada usando el gradiente de la función:

$$\nabla_s f(\mathbf{x}) = \nabla f(\mathbf{x})^\intercal \mathbf{s}$$

- Otra forma de calcular la derivada direccional $\nabla_s f(\mathbf{x})$ es definiendo

$$g(\alpha) \equiv f(\mathbf{x} + \alpha \mathbf{s})$$

y posteriormente calcular $g'(0)$.

Derivadas en múltiples dimensiones

Sea la función multivariada $f(x_1, x_2) = x_1 x_2$. La gráfica de superficie es la siguiente:

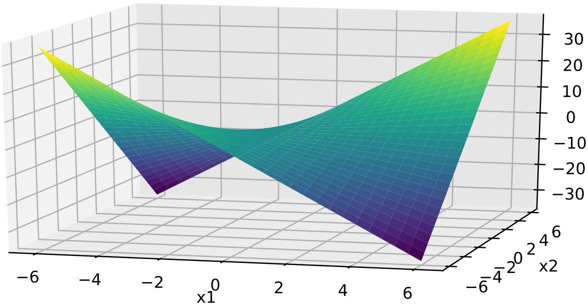


Figura: $f(x_1, x_2) = x_1 x_2$

Derivadas en múltiples dimensiones

Cálculo de derivada direccional

Calcula la derivada direccional de $f(\mathbf{x}) = x_1 x_2$ en $\mathbf{x} = [2, 0]$ en la dirección $\mathbf{s} = [-1, -1]$

$$f(\mathbf{x}) = x_1 x_2$$

$$\nabla f(\mathbf{x}) = \left[\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}\right] = [x_2, x_1]$$

$$\nabla_s f(\mathbf{x}) = \nabla f(\mathbf{x})^\intercal \mathbf{s} = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ -1 \end{bmatrix} = -1$$

Gradiente descendente: el algoritmo

Algorithm 1: Gradiente descendente

Input: f,∇f,$\mathbf{x}_0 \in \mathbb{R}^n$,$\epsilon$

Output: $\mathbf{x}_{k+1}$

$\mathbf{x}_k \leftarrow \mathbf{x}_0$

while $|f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k)| > \epsilon$ **do**

  $p_k \leftarrow -\nabla f(\mathbf{x}_k)$

  Calcular tamaño de paso α_k

  $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k p_k$

end while

return $\mathbf{x}_{k+1}$

Gradiente descendente: el algoritmo

Algorithm 2: Gradiente descendente

Input: f,∇f,$\mathbf{x}_0 \in \mathbb{R}^n$,$\epsilon$

Output: $\mathbf{x}_{k+1}$

$\mathbf{x}_k \leftarrow \mathbf{x}_0$

while $|f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k)| > \epsilon$ **do**

$p_k \leftarrow -\nabla f(\mathbf{x}_k)$

Calcular tamaño de paso α_k ¿cómo lo calculamos?

$\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k p_k$

end while

return $\mathbf{x}_{k+1}$

Algorithm 3: Gradiente descendente

Input: $f,\nabla f,\mathbf{x}_0 \in \mathbb{R}^n,\epsilon,\tilde{\alpha},c,\rho$

Output: $\mathbf{x}_{k+1}$

```
x_k ← x_0

while |f(x_{k+1}) − f(x_k)| > ε do
    p_k ← −∇f(x_k)
    /* ------- Algoritmo backtracking para tamaño de paso -------      */
    α_k ← α̃
    while f(x_k + α_k p_k) ≤ f(x_k) + c α_k ∇f_k^T p_k do
        α_k ← ρ α_k
    end while
    /* ---------------------------------------------------------      */
    x_{k+1} ← x_k + α_k p_k

end while

return  x_{k+1}
```

Funciones de prueba

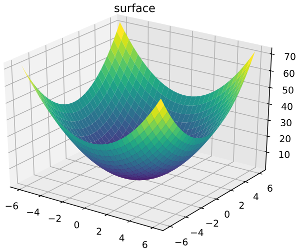


Figura: $f(x_1, x_2) = x_1^2 + x_2^2$

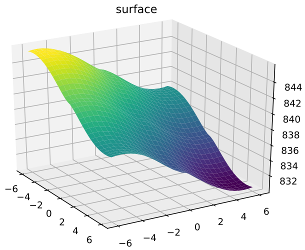


Figura:
$f(x_1, x_2) = 418{,}9829 * 2 - x_1 \sin\sqrt{|x_1|} - x_2 \sin\sqrt{|x_2|}$

References I

Aggarwal, C. C. (2020). *Linear Algebra and Optimization for Machine Learning - A Textbook*. Springer.

Kochenderfer, M. J. and Wheeler, T. A. (2019). *Algorithms for optimization*. MIT Press.

Neumaier, A. (2004). Complete search in continuous global optimization and constraint satisfaction. *Acta numerica*, 13:271–369.

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, New York, NY, USA, 2e edition.