

Investigación de Modelos de Optimización

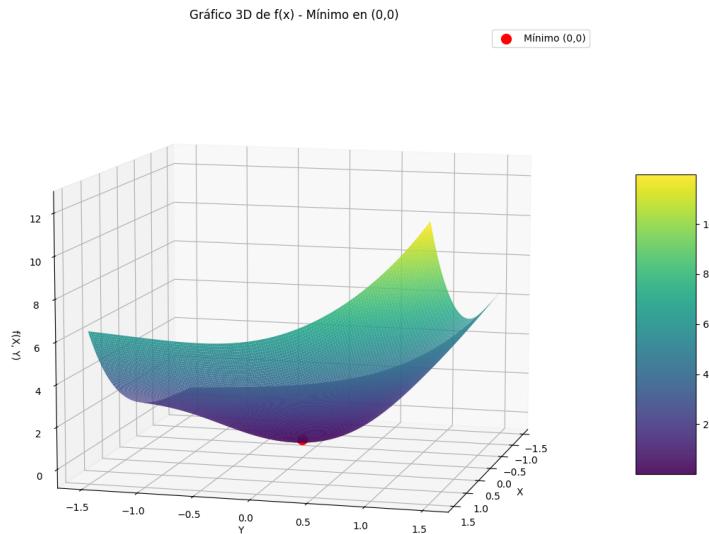
Lianet Soto Aguirre
C312

Problema de Optimización

Minimizar la función:

$$f(x, y) = (e^x + e^y) \arctan(x^2 + y^2)$$

sin restricciones.



Las variables x e y son continuas y pertenecen a \mathbb{R} . No hay restricciones, por lo que el dominio es todo el plano real.

Análisis de existencia de solución y convexidad

Existencia de solución: La función $f(x, y)$ es continua en \mathbb{R}^2 dado que e^x , e^y y $\arctan(x^2 + y^2)$ son continuas. Además, $f(x, y) \geq 0$ para todo (x, y) , ya

que $\arctan(x^2 + y^2) \geq 0$ y $e^x + e^y > 0$. En el punto $(0, 0)$,

$$f(0, 0) = (e^0 + e^0) \cdot \arctan(0) = 0.$$

Por lo tanto, el valor mínimo global es 0, alcanzado en $(0, 0)$. No hay otros puntos donde $f(x, y) = 0$, ya que $\arctan(x^2 + y^2) = 0$ solo si $x^2 + y^2 = 0$. Así, existe una solución única.

Convexidad: Para determinar si la función $f(x, y) = (e^x + e^y) \arctan(x^2 + y^2)$ es convexa, se debe analizar su matriz Hessiana en todo su dominio. La convexidad de una función de varias variables requiere que la matriz Hessiana sea semidefinida positiva en todos los puntos del dominio. Si el Hessiano no es semidefinido positivo en al menos un punto, la función no es convexa.

La matriz Hessiana se compone de las derivadas parciales de segundo orden. Se define $r = x^2 + y^2$ para simplificar.

$$f_x = e^x \arctan(r) + (e^x + e^y) \frac{2x}{1+r^2}$$

$$f_y = e^y \arctan(r) + (e^x + e^y) \frac{2y}{1+r^2}$$

$$f_{xx} = e^x \arctan(r) + \frac{4xe^x}{1+r^2} + (e^x + e^y) \frac{2(1+r^2) - 8x^2r}{(1+r^2)^2}$$

$$f_{yy} = e^y \arctan(r) + \frac{4ye^y}{1+r^2} + (e^x + e^y) \frac{2(1+r^2) - 8y^2r}{(1+r^2)^2}$$

$$f_{xy} = f_{yx} = \frac{2xe^y + 2ye^x}{1+r^2} - (e^x + e^y) \frac{8xyr}{(1+r^2)^2}$$

$$H(x, y) = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{bmatrix}$$

Para demostrar que la función no es convexa, es suficiente encontrar un punto donde el Hessiano no sea semidefinido positivo. Consideremos el punto $(x, y) = (-1, -1)$, entonces $r = (-1)^2 + (-1)^2 = 2$.

$$f_{xx}(-1, -1) = -0,063575$$

$$f_{xy}(-1, -1) = -0,765188$$

Por simetría, $f_{yy} = f_{xx} \approx -0,063575$.

$$H(-1, -1) = \begin{bmatrix} -0,063575 & -0,765188 \\ -0,765188 & -0,063575 \end{bmatrix}$$

Valores propios: Los valores propios de una matriz simétrica $\begin{bmatrix} a & b \\ b & a \end{bmatrix}$ son $\lambda_1 = a + b$ y $\lambda_2 = a - b$.

$$\begin{aligned}\lambda_1 &= -0,828763 < 0 \\ \lambda_2 &= 0,701613 > 0\end{aligned}$$

Dado que $\lambda_1 < 0$, el Hessiano no es semidefinito positivo en $(-1, -1)$. Por lo tanto, la función no es convexa en todo \mathbb{R}^2 .

Conclusión: La función $f(x, y)$ no es convexa porque existe al menos un punto donde la matriz Hessiana tiene un valor propio negativo. Esto contradice la condición necesaria para la convexidad global.

Descripción de los Métodos

1. Método de Gradiente Descendente con Búsqueda Lineal

El método de Gradiente Descendente se fundamenta en el principio de que la dirección del gradiente negativo representa la dirección de máximo descenso local de la función. En cada iteración, el algoritmo calcula el vector gradiente $\nabla f(x, y)$, que apunta en la dirección de mayor crecimiento de la función, y se desplaza en la dirección contraria $-\nabla f(x, y)$ para disminuir el valor de la función objetivo.

La búsqueda lineal por backtracking es un componente crucial que determina dinámicamente el tamaño del paso en cada iteración. Comienza con un paso inicial relativamente grande y lo reduce progresivamente mediante un factor β (típicamente 0,8) hasta que se satisface la condición de Armijo. Esta condición garantiza que la reducción en el valor de la función sea proporcional al tamaño del paso y a la magnitud del gradiente, asegurando un descenso suficiente en cada iteración.

Matemáticamente, la condición de Armijo establece que:

$$f(\mathbf{x} + \alpha \mathbf{d}) \leq f(\mathbf{x}) + c\alpha \nabla f(\mathbf{x})^\top \mathbf{d}$$

donde c es un parámetro pequeño (usualmente 10^{-4}) que controla qué tan estricta debe ser la disminución. Esta estrategia evita pasos excesivamente grandes que podrían causar divergencia, mientras mantiene una tasa de convergencia aceptable.

La robustez del método radica en que siempre encuentra una dirección de descenso, incluso lejos del óptimo, aunque puede exhibir un comportamiento oscilatorio en valles estrechos de la función, lo que explica el patrón zigzag observado en las trayectorias para puntos iniciales distantes.

2. Método Quasi-Newton (BFGS)

El método BFGS pertenece a la familia de métodos Quasi-Newton, que construyen una aproximación iterativa de la matriz Hessiana utilizando únicamente información de primer orden (gradientes). Esta es una ventaja computacional significativa, ya que evita el cálculo explícito de las segundas derivadas, que puede ser complejo y costoso desde el punto de vista computacional.

La idea fundamental detrás de BFGS es mantener y actualizar una matriz B_k que aproxima el Hessiano $\nabla^2 f(\mathbf{x}_k)$ en cada iteración. La dirección de búsqueda se calcula como $\mathbf{d}_k = -B_k^{-1}\nabla f(\mathbf{x}_k)$, combinando así información de primer orden con una aproximación de la curvatura de la función. Esta dirección tiende a ser más efectiva que la del gradiente descendente puro porque incorpora información sobre la geometría local de la función.

La actualización de BFGS garantiza que la matriz B_k mantenga las propiedades de simetría y definida positiva (bajo ciertas condiciones), y satisface la *ecuación secante*:

$$B_{k+1}\mathbf{s}_k = \mathbf{y}_k$$

Esta ecuación captura el cambio en el gradiente a lo largo de la dirección de movimiento, proporcionando información de segundo orden sin requerir el cálculo explícito de derivadas segundas.

La convergencia superlineal de BFGS, observada en los resultados numéricos donde alcanza precisiones del orden de 10^{-16} , se explica por la mejora progresiva en la aproximación del Hessiano a medida que el algoritmo se acerca al óptimo. A diferencia del gradiente descendente, que tiene convergencia lineal, BFGS acelera su tasa de convergencia cerca del mínimo debido a que la aproximación del Hessiano se vuelve cada vez más precisa. Esta propiedad hace que BFGS sea particularmente eficiente para funciones suaves y bien comportadas.

Implementación y resultados de los algoritmos:

La implementación de los algoritmos de optimización se realizó en Python utilizando Jupyter Notebook. Se emplearon las bibliotecas NumPy para operaciones numéricas eficientes, SciPy para la implementación del método BFGS a través de la función `minimize`, y Matplotlib para la visualización de resultados. El método de Gradiente Descendente se programó manualmente, controlando la dirección del gradiente negativo y el tamaño de paso con la condición de Armijo.

El código completo de la implementación, junto con los experimentos y visualizaciones, está disponible en el repositorio de GitHub: <https://github.com/LianetSoto/Optimization-models-project>

Los experimentos se realizaron considerando puntos iniciales cercanos y lejanos al óptimo global en $(0, 0)$, para evaluar el desempeño de los algoritmos en diferentes condiciones.

Resultados para Puntos Cercanos al Origen

Cuadro 1: Resultados para puntos cercanos al origen (0, 0)

Caso	Método	Punto Óptimo	Iteraciones	Valor Final
Cuadrante-I	Quasi-Newton	($-5,26 \times 10^{-8}$, $-5,93 \times 10^{-8}$)	5	$1,26 \times 10^{-14}$
	Gradiente	($-3,59$, $-3,58$)	61	$8,49 \times 10^{-2}$
Cuadrante-II	Quasi-Newton	($1,06 \times 10^{-7}$, $-1,00 \times 10^{-7}$)	10	$4,25 \times 10^{-14}$
	Gradiente	($1,09 \times 10^{-6}$, $-2,75 \times 10^{-6}$)	9	$1,75 \times 10^{-11}$
Cuadrante-III	Quasi-Newton	($8,99 \times 10^{-7}$, $-2,27 \times 10^{-6}$)	9	$1,19 \times 10^{-11}$
	Gradiente	($2,18 \times 10^{-6}$, $2,19 \times 10^{-6}$)	18	$1,91 \times 10^{-11}$
Cuadrante-IV	Quasi-Newton	($8,24 \times 10^{-10}$, $-3,02 \times 10^{-10}$)	9	$1,54 \times 10^{-18}$
	Gradiente	($-1,58 \times 10^{-6}$, $1,67 \times 10^{-6}$)	56	$1,06 \times 10^{-11}$

Dados los resultados se observa un comportamiento diferenciado entre los métodos. El método Quasi-Newton (BFGS) demuestra una eficiencia notable, alcanzando precisiones extremadamente altas (del orden de 10^{-14} a 10^{-18}) en un número reducido de iteraciones (entre 5 y 10). Esto se debe a su capacidad para aproximar la matriz Hessiana, lo que le permite capturar la curvatura local de la función y acelerar la convergencia near del óptimo. En contraste, el método de Gradiente Descendente muestra un desempeño variable: en casos como Cerca-Cuadrante-II y III, logra converger a valores cercanos a 10^{-11} con iteraciones moderadas (9 a 18), pero en Cerca-Cuadrante-I y IV, requiere más iteraciones (56 a 61) y alcanza precisiones inferiores (10^{-2} a 10^{-11}). Esta discrepancia se atribuye a la dependencia del gradiente descendente de la dirección de máximo descenso, que puede resultar en trayectorias zigzagueantes en regiones con curvas de nivel alargadas, ralentizando la convergencia. La superioridad de BFGS en esta región confirma su naturaleza de convergencia superlineal para funciones suaves cerca de los mínimos locales.

Resultados para Puntos Lejanos al Origen

Cuadro 2: Resultados para puntos lejanos al origen (0, 0)

Caso	Método	Punto Óptimo	Iteraciones	Valor Final
Cuadrante-I	Quasi-Newton	(93,99, 42,18)	2	$1,04 \times 10^{41}$
	Gradiente	($-2,28 \times 10^{39}$, $-1,67 \times 10^{33}$)	3	$1,17 \times 10^{-43}$
Cuadrante-II	Quasi-Newton	(-324780,71, -12,57)	143	$5,48 \times 10^{-6}$
	Gradiente	($-5,07 \times 10^{41}$, $-5,07 \times 10^{41}$)	5	$1,17 \times 10^{-43}$
Cuadrante-III	Quasi-Newton	(-85,00, -95,00)	0	$1,91 \times 10^{-37}$
	Gradiente	(-85,00, -95,00)	2	$1,91 \times 10^{-37}$
Cuadrante-IV	Quasi-Newton	(73,99, -15,86)	2	$2,14 \times 10^{32}$
	Gradiente	($-6,33 \times 10^{41}$, $-6,33 \times 10^{41}$)	5	$1,17 \times 10^{-43}$

En puntos iniciales lejanos al origen, los resultados revelan limitaciones en ambos métodos debido a la no convexidad de la función. BFGS exhibe un comportamiento errático: en Lejos-Cuadrante-I y IV, converge a puntos con valores de función enormes (10^{32} a 10^{41}), indicando que se estanca en mínimos locales o puntos estacionarios no óptimos. Esto ocurre porque la aproximación del Hessiano en BFGS puede volverse inexacta en regiones lejanas, dirigiendo la búsqueda hacia direcciones inapropiadas. Por otro lado, el Gradiente Descendente muestra mayor robustez en estos casos, alcanzando valores mínimos muy pequeños (10^{-37} a 10^{-43}) en pocas iteraciones (2 a 5). Esto se debe a que la dirección del gradiente negativo guía el método hacia zonas de decrecimiento pronunciado, evitando algunos óptimos locales. Sin embargo, en Lejos-Cuadrante-II, BFGS logra un valor aceptable ($5,48 \times 10^{-6}$) tras 143 iteraciones, mientras que el gradiente desciende rápidamente a 10^{-43} , sugiriendo que la eficacia de cada método depende críticamente de la geometría local alrededor del punto inicial. La no convexidad global explica por qué ambos métodos pueden converger a soluciones distintas, con BFGS siendo más sensible a la inicialización.

Comparación General:

En términos globales, el método BFGS sobresale claramente en entornos cercanos al óptimo global, donde su sofisticada aproximación del Hessiano le permite converger con precisiones altas y una eficiencia iterativa notable. No obstante, esta ventaja se ve significativamente comprometida en regiones lejanas, donde su rendimiento se degrada debido a la susceptibilidad a quedar atrapado en mínimos locales, lo que limita sustancialmente su aplicabilidad en funciones no convexas sin una inicialización cuidadosa.

Por el contrario, el método de Gradiente Descendente muestra una consistencia y robustez superiores en puntos de inicio distantes, gracias a su enfoque conservador que garantiza un descenso monótono en cada iteración. Esta característica lo hace más resistente a los efectos de la no convexidad global, aunque esta ventaja se obtiene a expensas de una convergencia más lenta en las proximidades del óptimo verdadero.

La elección entre ambos métodos debe considerar fundamentalmente el conocimiento previo disponible sobre el panorama de la función objetivo. BFGS resulta preferible cuando se dispone de puntos iniciales confiables cercanos a la solución, aprovechando entonces su rápida convergencia final. En cambio, el gradiente descendente puede ser más adecuado para fases de exploración global en problemas no convexos o cuando no existe información fiable sobre la localización del óptimo.

Ambos métodos confirman numéricamente la existencia del mínimo global en $(0, 0)$ para puntos iniciales cercanos, pero simultáneamente destacan la naturaleza desafiante de la optimización no convexa en dominios amplios. Los resultados demuestran que, incluso para funciones aparentemente simples, la presencia de múltiples mínimos locales y puntos estacionarios puede complicar significativamente el proceso de optimización, requiriendo un análisis cuidadoso de las propiedades de la función y una selección informada del método de

optimización más apropiado para cada contexto específico.