

Optimización Numérica sin restricciones

Tema 3: Gradiente Conjugado

Oscar Dalmau

Centro de Investigación en Matemáticas
CIMAT

23 de abril de 2021

Orden del Tema

- 1 Gradiente Conjugado Lineal
- 2 Gradiente Conjugado No Lineal
- 3 Gradiente Conjugado Precondicionado
 - Precondicionamiento
 - Precondicionadores

Gradiente Conjugado

Se quiere resolver el problema de optimización sin restricciones

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} x^T Q x - b^T x \quad (1)$$

donde Q es una matriz simétrica y definida positiva.

Direcciones Conjugadas

- Sea \mathbf{Q} una matriz simétrica y definida positiva. Dos vectores $\mathbf{d}_1, \mathbf{d}_2$ se dicen *conjugados con respecto a \mathbf{Q}* o simplemente \mathbf{Q} -ortogonales si $\mathbf{d}_1^T \mathbf{Q} \mathbf{d}_2 = 0$.
- Un conjunto de vectores $\mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_k$ son mutuamente \mathbf{Q} -ortogonales si $\mathbf{d}_i^T \mathbf{Q} \mathbf{d}_j = 0$ para $i \neq j$.

Algoritmo Gradiente Conjugado

Algorithm 1 GC-Versión Preliminar

Require: x_0

Ensure: x^*

1: Hacer $g_0 = Qx_0 - b, d_0 = -g_0, k = 0$

2: **while** $\|g_k\| \neq 0$ (No conveja) **do**

3: $\alpha_k = -\frac{g_k^T d_k}{d_k^T Q d_k}$

4: $x_{k+1} = x_k + \alpha_k d_k$

5: $g_{k+1} = Qx_{k+1} - b = \nabla f(x_{k+1})$

6: $\beta_{k+1} = \frac{d_k^T Q g_{k+1}}{d_k^T Q d_k}$

7: $d_{k+1} = -g_{k+1} + \beta_{k+1} d_k$

8: $k = k + 1$

9: **end while**

Forma práctica de GC

Algunas relaciones

- $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$
- $\mathbf{g}_{k+1} = \mathbf{Q}\mathbf{x}_{k+1} - \mathbf{b}$
- $\mathbf{g}_{k+1} = \mathbf{g}_k + \alpha_k \mathbf{Q}\mathbf{d}_k$

Forma práctica de GC

Otras relaciones

- $\mathbf{g}_{k+1}^T \mathbf{d}_k = 0$
- $\alpha_k = -\frac{\mathbf{g}_k^T \mathbf{d}_k}{\mathbf{d}_k^T \mathbf{Q} \mathbf{d}_k}$
- $\beta_{k+1} = \frac{\mathbf{g}_{k+1}^T \mathbf{Q} \mathbf{d}_k}{\mathbf{d}_k^T \mathbf{Q} \mathbf{d}_k}$
- $\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_{k+1} \mathbf{d}_k$

Forma práctica de GC

Proposición

Si $\{d_0, d_1, \dots, d_k\}$ son Q -ortogonales entonces,

$$g_{k+1}^T d_i = 0$$

$$g_{k+1}^T g_i = 0$$

para $i = 0, 1, \dots, k$

Algoritmo Gradiente Conjugado

$$\alpha_k = \frac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{d}_k^T \mathbf{Q} \mathbf{d}_k} = - \frac{\mathbf{g}_k^T \mathbf{d}_k}{\mathbf{d}_k^T \mathbf{Q} \mathbf{d}_k}$$
$$\beta_{k+1} = \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{g}_k} = \frac{\mathbf{g}_{k+1}^T \mathbf{Q} \mathbf{d}_k}{\mathbf{d}_k^T \mathbf{Q} \mathbf{d}_k}$$

Nos basamos en:

$$\begin{aligned}\mathbf{g}_k^T \mathbf{g}_k &= -\mathbf{g}_k^T \mathbf{d}_k \\ \alpha_k \mathbf{Q} \mathbf{d}_k &= \mathbf{g}_{k+1} - \mathbf{g}_k \\ \mathbf{g}_{k+1}^T \mathbf{g}_k &= 0 \\ \mathbf{g}_{k+1}^T \mathbf{d}_k &= 0\end{aligned}$$

Algoritmo Gradiente Conjugado

Algorithm 2 GC (Estandar)

Require: x_0

Ensure: x^*

- 1: Hacer $g_0 = Qx_0 - b, d_0 = -g_0, k = 0$
 - 2: **while** $\|g_k\| \neq 0$ (No conveja) **do**
 - 3: $q_k = Qd_k$
 - 4: $\alpha_k = \frac{g_k^T g_k}{d_k^T q_k} = -\frac{g_k^T d_k}{d_k^T Qd_k}$
 - 5: $x_{k+1} = x_k + \alpha_k d_k$
 - 6: $g_{k+1} = g_k + \alpha_k q_k = \nabla f(x_{k+1})$
 - 7: $\beta_{k+1} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} = \frac{g_{k+1}^T Qd_k}{d_k^T Qd_k}$
 - 8: $d_{k+1} = -g_{k+1} + \beta_{k+1} d_k$
 - 9: $k = k + 1$
 - 10: **end while**
-

Algoritmo GC no lineal

El Algoritmo GC se usa para resolver el problema de optimización

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} x^T Q x - b^T x \quad (2)$$

donde Q es una matriz simétrica y definida positiva, o equivalentemente, resolver el sistema de ecuaciones

$$Qx = b \quad (3)$$

Será posible usarlo cuando $f(\cdot)$ es una función convexa cualquiera?

Algoritmo GC no lineal

El Algoritmo GC se usa para resolver el problema de optimización

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} x^T Q x - b^T x \quad (2)$$

donde Q es una matriz simétrica y definida positiva, o equivalentemente, resolver el sistema de ecuaciones

$$Qx = b \quad (3)$$

Será posible usarlo cuando $f(\cdot)$ es una función convexa cualquiera?

Algoritmo GC no lineal Fletcher-Reeves

Algorithm 3 GC Fletcher-Reeves

Require: x_0

Ensure: x^*

- 1: $d_0 = -\nabla f(x_0)$, $k = 0$
 - 2: **while** $\|\nabla f(x_k)\| \neq 0$ (No conveja) **do**
 - 3: Calcular α_k usando un método de búsqueda en línea
 - 4: $x_{k+1} = x_k + \alpha_k d_k$
 - 5: Calcular $\nabla f(x_{k+1})$
 - 6: $\beta_{k+1}^{FR} = \frac{\nabla^T f(x_{k+1}) \nabla f(x_{k+1})}{\nabla^T f(x_k) \nabla f(x_k)}$
 - 7: $d_{k+1} = -\nabla f(x_{k+1}) + \beta_{k+1}^{FR} d_k$
 - 8: $k = k + 1$
 - 9: **end while**
-

Sugerencia: `scipy.optimize.line_search`

Algoritmo GC no lineal

Otras variantes

El método de Fletcher-Reeves (FR) converge si el punto inicial está suficientemente cerca del óptimo. Existen variantes del método FR. La diferencia fundamental es la forma de calcular el parámetro β_k

- Polak-Ribiere

$$\beta_{k+1}^{PR} = \frac{\nabla^T f(\mathbf{x}_{k+1})(\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k))}{\nabla^T f(\mathbf{x}_k)\nabla f(\mathbf{x}_k)}$$

- En algunos casos (raros) el algoritmo de PR se puede ciclar infinitamente. Sin embargo, se puede garantizar la convergencia tomando $\beta_{k+1}^+ = \max(0, \beta_{k+1}^{PR})$.

Algoritmo GC no lineal

Otras variantes

El método de Fletcher-Reeves (FR) converge si el punto inicial está suficientemente cerca del óptimo. Existen variantes del método FR. La diferencia fundamental es la forma de calcular el parámetro β_k

- Polak-Ribiere

$$\beta_{k+1}^{PR} = \frac{\nabla^T f(\mathbf{x}_{k+1})(\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k))}{\nabla^T f(\mathbf{x}_k)\nabla f(\mathbf{x}_k)}$$

- En algunos casos (raros) el algoritmo de PR se puede ciclar infinitamente. Sin embargo, se puede garantizar la convergencia tomando $\beta_{k+1}^+ = \max(0, \beta_{k+1}^{PR})$.

Algoritmo GC no lineal

Otras variantes

El método de Fletcher-Reeves (FR) converge si el punto inicial está suficientemente cerca del óptimo. Existen variantes del método FR. La diferencia fundamental es la forma de calcular el parámetro β_k

- Hestenes-Stiefel

$$\beta_{k+1}^{HS} = \frac{\nabla^T f(\mathbf{x}_{k+1})(\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k))}{(\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k))^T \mathbf{d}_k}$$

Algoritmo GC no lineal

Fletcher-Reeves Polak-Ribiere

- Es posible garantizar la convergencia si $|\beta_k| \leq \beta_k^{FR}$.
- Lo anterior, una variante adecuada sería: FR-PR para $k \geq 2$

$$\beta_k = \begin{cases} -\beta_k^{FR} & \text{si } \beta_k^{PR} < -\beta_k^{FR} \\ \beta_k^{PR} & \text{si } |\beta_k^{PR}| \leq \beta_k^{FR} \\ \beta_k^{FR} & \text{si } \beta_k^{PR} > \beta_k^{FR} \end{cases}$$

Nota: La idea anterior se basa en que para cualquier $|\beta_k| \leq \beta_k^{FR}$ se puede demostrar convergencia global para $k \geq 2$.

Descenso de Fletcher-Reeves

$$\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_{k+1}^{FR} \mathbf{d}_k$$

- Si α_k es un tamaño de paso exacto entonces $\mathbf{g}_{k+1}^T \mathbf{d}_k = 0$, luego

$$\mathbf{g}_{k+1}^T \mathbf{d}_{k+1} = -\|\mathbf{g}_{k+1}\|^2 + \beta_{k+1}^{FR} \mathbf{g}_{k+1}^T \mathbf{d}_k = -\|\mathbf{g}_{k+1}\|^2 < 0$$

es decir, \mathbf{d}_{k+1} es de descenso.

- Si α_k no es un tamaño de paso exacto, el termino $\beta_{k+1}^{FR} \mathbf{g}_{k+1}^T \mathbf{d}_k$ podría dominar al primero, y no se garantiza descenso.

Descenso de Fletcher-Reeves

Para el caso de tamaño de paso no exacto, se puede garantizar descenso si se usan las condiciones fuertes de Wolfe

$$f_{k+1} \leq f_k + c_1 \alpha_k \mathbf{g}_k^T \mathbf{d}_k \quad (4)$$

$$|\mathbf{g}_{k+1}^T \mathbf{d}_k| \leq -c_2 \mathbf{g}_k^T \mathbf{d}_k \quad (5)$$

con $0 < c_2 < c_1 < 1$. Las condiciones anteriores garantizan la convergencia, seleccionando $0 < c_2 < \frac{1}{2}$

Descenso de Fletcher-Reeves

Proposicion

Si el algoritmo de GC no lineal usa el tamaño de paso inexacto satisfaciendo las condiciones fuertes de Wolfe con $0 < c_2 < \frac{1}{2}$. Entonces el algoritmo produce una direccion de descenso \mathbf{d}_k , que satisface

$$-\frac{1}{1 - c_2} \leq \frac{\mathbf{g}_k^T \mathbf{d}_k}{\|\mathbf{g}_k\|^2} \leq \frac{2c_2 - 1}{1 - c_2}$$

para $k = 0, 1, \dots$

Ver la demostracion en el Nocedal (por induccion)

El problema de optimizacion

$$\min_x \quad f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} - \mathbf{b}^T \mathbf{x}$$

Puede ser resuelto en a lo mas 'n' iteraciones usando Gradiente Conjugado.

Sin embargo, la convergencia de GC depende del numero de condicion

$$\kappa(\mathbf{Q}) = \|\mathbf{Q}\|_2 \|\mathbf{Q}^{-1}\|_2 = \frac{\lambda_n}{\lambda_1}$$

de la matriz \mathbf{Q} , es decir,

$$\|\mathbf{x}_k - \mathbf{x}^*\|_{\mathbf{Q}} \leq 2 \left(\frac{\sqrt{\kappa(\mathbf{Q})} - 1}{\sqrt{\kappa(\mathbf{Q})} + 1} \right)^k \|\mathbf{x}_0 - \mathbf{x}^*\|_{\mathbf{Q}}$$

aunque la cota anterior esta sobreestimada, nos da una idea de como acelerar el GC. Se puede ver que si $\kappa(\mathbf{Q}) \approx 1$ entonces la convergencia podría ser mas rapida, en particular, si $\kappa(\mathbf{Q}) = 1$, converge en una iteracion.

Orden del Tema

- 1 Gradiente Conjugado Lineal
- 2 Gradiente Conjugado No Lineal
- 3 **Gradiente Conjugado Precondicionado**
Precondicionamiento
Precondicionadores

La idea general del precondicionamiento consiste en transformar el sistema

$$Qx = b$$

de modo que la matriz del sistema Q tenga un mejor número de condición. Para ello podríamos premultiplicar el sistema por una matriz M^{-1} de tal forma que $M^{-1} \approx Q^{-1}$.

Luego, obtendríamos el sistema equivalente

$$\mathbf{M}^{-1}\mathbf{Q}x = \mathbf{M}^{-1}b$$

donde suponemos que \mathbf{M} es positiva definida, por ejemplo podríamos seleccionar una matriz diagonal. Si $\mathbf{M}^{-1} \approx \mathbf{Q}^{-1}$ entonces

$$\mathbf{M}^{-1}\mathbf{Q} \approx \mathbf{I}$$

por lo que $\kappa(\mathbf{M}^{-1}\mathbf{Q}) \approx 1$.

Definamos

$$\mathbf{M} \stackrel{def}{=} \mathbf{C}^T \mathbf{C}$$

El sistema anterior, puede ser reescrito de la siguiente forma

$$\mathbf{M}^{-1} \mathbf{Q} \mathbf{x} = \mathbf{M}^{-1} \mathbf{b} \quad (6)$$

$$\mathbf{C}^{-1} \mathbf{C}^{-T} \mathbf{Q} \mathbf{C}^{-1} \hat{\mathbf{x}} = \mathbf{C}^{-1} \mathbf{C}^{-T} \mathbf{b} \quad (7)$$

$$\mathbf{C}^{-T} \mathbf{Q} \mathbf{C}^{-1} \hat{\mathbf{x}} = \mathbf{C}^{-T} \mathbf{b} \quad (8)$$

con

$$\mathbf{C}^{-1} \hat{\mathbf{x}} = \mathbf{x} \quad (9)$$

$$\hat{\mathbf{x}} = \mathbf{C} \mathbf{x} \quad (10)$$

- El nuevo sistema de ecuaciones (8), es equivalente a resolver un problema de optimization cuadrático.
- El siguiente problema se hubiera obtenido también sustituyendo (9), $\mathbf{x} = \mathbf{C}^{-1}\hat{\mathbf{x}}$, en función $f(\cdot)$, es decir $g(\hat{\mathbf{x}}) = f(\mathbf{x}) = f(\mathbf{C}^{-1}\hat{\mathbf{x}})$

$$\min_{\hat{x}} g(\hat{x}) = \frac{1}{2} \hat{x}^T \mathbf{C}^{-T} \mathbf{Q} \mathbf{C}^{-1} \hat{x} - (\mathbf{C}^{-T} \mathbf{b})^T \hat{x} \quad (11)$$

$$= \frac{1}{2} \hat{x}^T \hat{\mathbf{Q}} \hat{x} - \hat{\mathbf{b}}^T \hat{x} \quad (12)$$

Donde

$$\hat{\mathbf{Q}} = \mathbf{C}^{-T} \mathbf{Q} \mathbf{C}^{-1} \quad (13)$$

$$\hat{\mathbf{b}} = \mathbf{C}^{-T} \mathbf{b} \quad (14)$$

$$\hat{x} = \mathbf{C} x \quad (15)$$

$$\mathbf{C}^{-1} \hat{x} = x \quad (16)$$

- La idea ahora es aplicar al algoritmo GC al problema de minimización de $g(\hat{x})$, Ecuación (12).
- Luego, hacer sustitución de variables para llevar las ecuaciones a los términos equivalentes que corresponden a la minimización de $f(x)$.
- Es decir, convertir de la notación nueva a la original, i.e., $\hat{Q}, \hat{b}, \hat{x} \rightarrow Q, b, x$ mediante las ecuaciones (13)-(15).

Require: \hat{x}_0

Ensure: \hat{x}^*

1: Hacer $\hat{g}_0 = \hat{Q}\hat{x}_0 - \hat{b}$, $\hat{d}_0 = -\hat{g}_0$, $k = 0$

2: **while** $\|\hat{g}_k\| \neq 0$ (No conveja) **do**

3: $\hat{\alpha}_k = \frac{\hat{g}_k^T \hat{g}_k}{\hat{d}_k^T \hat{Q} \hat{d}_k}$

4: $\hat{x}_{k+1} = \hat{x}_k + \hat{\alpha}_k \hat{d}_k$

5: $\hat{g}_{k+1} = \hat{g}_k + \hat{\alpha}_k \hat{Q} \hat{d}_k$

6: $\hat{\beta}_{k+1} = \frac{\hat{g}_{k+1}^T \hat{g}_{k+1}}{\hat{g}_k^T \hat{g}_k}$

7: $\hat{d}_{k+1} = -\hat{g}_{k+1} + \hat{\beta}_{k+1} \hat{d}_k$

8: $k = k + 1$

9: **end while**

Por ejemplo para $\hat{\alpha}_k$ tenemos

$$\hat{\alpha}_k = \frac{\hat{\mathbf{g}}_k^T \hat{\mathbf{g}}_k}{\hat{\mathbf{d}}_k^T \hat{\mathbf{Q}} \hat{\mathbf{d}}_k} = \frac{\mathbf{g}_k^T \mathbf{M}^{-1} \mathbf{g}_k}{\mathbf{d}_k^T \mathbf{Q} \mathbf{d}_k}$$

Para $\hat{\beta}_{k+1}$ se obtiene

$$\hat{\beta}_{k+1} = \frac{\hat{\mathbf{g}}_{k+1}^T \hat{\mathbf{g}}_{k+1}}{\hat{\mathbf{g}}_k^T \hat{\mathbf{g}}_k} = \frac{\mathbf{g}_{k+1}^T \mathbf{M}^{-1} \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{M}^{-1} \mathbf{g}_k}$$

Defininiendo \mathbf{z}_k como la solucion del siguiente sistema de ecuaciones,

$$\mathbf{M} \mathbf{z}_k \stackrel{def}{=} \mathbf{g}_k$$

entonces

$$\mathbf{z}_k = \mathbf{M}^{-1} \mathbf{g}_k$$

- 1: Hacer $\mathbf{g}_0 = \mathbf{Q}\mathbf{x}_0 - \mathbf{b}$,
- 2: Resolver $\mathbf{M}\mathbf{z}_0 = \mathbf{g}_0$ para \mathbf{z}_0
- 3: $\mathbf{d}_0 = -\mathbf{z}_0$, $k = 0$
- 4: **while** $\|\mathbf{g}_k\| \neq 0$ (No conveja) **do**
- 5: $\alpha_k = \frac{\mathbf{g}_k^T \mathbf{z}_k}{\mathbf{d}_k^T \mathbf{Q} \mathbf{d}_k}$
- 6: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$
- 7: $\mathbf{g}_{k+1} = \mathbf{g}_k + \alpha_k \mathbf{Q} \mathbf{d}_k$.
- 8: Resolver el sistema $\mathbf{M}\mathbf{z}_{k+1} = \mathbf{g}_{k+1}$ para \mathbf{z}_{k+1} ,
- 9: $\beta_{k+1} = \frac{\mathbf{g}_{k+1}^T \mathbf{z}_{k+1}}{\mathbf{g}_k^T \mathbf{z}_k}$
- 10: $\mathbf{d}_{k+1} = -\mathbf{z}_{k+1} + \beta_{k+1} \mathbf{d}_k$.
- 11: $k = k + 1$
- 12: **end while**

Notar que si en el algoritmo *Gradiente Conjugado Precondicionado* se hace $\mathbf{M} = \mathbf{I}$ entonces se obtiene el *Algoritmo de Gradiente Conjugado* estandar, pues en este caso se tiene $\mathbf{g}_k = \mathbf{z}_k$.

Orden del Tema

- 1 Gradiente Conjugado Lineal
- 2 Gradiente Conjugado No Lineal
- 3 Gradiente Conjugado Precondicionado**
Precondicionamiento
Precondicionadores

- 1 M debe ser simetrica y positiva definida .
- 2 M debe ser tal que el sistema $Mz_k = g_k$ se pueda resolver eficientemente.
- 3 M^{-1} debe aproximar Q^{-1} en el sentido que $\|I - M^{-1}Q\| < 1$

Usando la descomposicion $\mathbf{Q} = \mathbf{L} + \mathbf{D} + \mathbf{L}^T$ de la matriz positiva definida \mathbf{Q} , donde \mathbf{L} es la matriz triangular inferior y \mathbf{D} es una matriz diagonal, entonces podemos tener opciones para \mathbf{M} dadas

- 1 $\mathbf{M} = \mathbf{D}$: Precondicionamiento Jacobi ,
- 2 $\mathbf{M} = \mathbf{L} + \mathbf{D}$: Precondicionamiento Gauss-Seidel,
- 3 $\mathbf{M} = \mathbf{L} + \frac{1}{\omega}\mathbf{D}$: Precondicionamiento SOR, con $\omega \in (0, 2)$

Tarea

Implementar los Algoritmos de GC lineal y no lineal. Probar los algoritmos anteriores con las siguientes funciones:

$$f(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2 + x_1x_2, f(x, y) = xe^{-x^2-y^2} \text{ y}$$

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \text{ (La Función de Rosembrock)}$$

- 1 Ejecute los algoritmos, adecuadamente, para cada una de las funciones desde 2 puntos iniciales.
- 2 Reporte la siguiente información para algunas de las iteraciones (incluya algunas de las iteraciones finales): k (Iteración), $|\nabla f(\mathbf{x}_k)|$ (Norma del Gradiente), $|\nabla \mathbf{d}_k|$ (Norma de la dirección de descenso) $f(\mathbf{x}_k)$ (Valor de la función), \mathbf{x}_k (Punto encontrado).
- 3 Grafique $(\mathbf{x}_k, f(\mathbf{x}_k))$ y $(\mathbf{x}_k, |\nabla f(\mathbf{x}_k)|)$