Tarea 9: Gradiente Conjugado y Barzilai-Borwein

Optimización I
Maestría en Computación
Centro de Investigación en Matemáticas

Esteban Reyes Saldaña
esteban.reyes@cimat.mx

27 de abril de 2021

Resumen

En esta tarea se implementó el método de Gradiente conjugado y el método de Barzilai-Borwein. El método de Gradiente Conjugado resuelve el problema de minimización de una función cuadrática mediante la diagonalización de su expresión algebraica. Se presenta a continuación una descripción general, así como el pseudocódigo de los métodos implementados. Finalmente se incluyen conclusiones observadas a partir de la experimentación.

1 Introducción

1.1 Gradiente Conjugado

Se quiere resolver mediante un algoritmo iterativo, el problema de optimización sin restricciones

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

donde Q es una matriz simétrica y definida positiva. Notemos que esto es equivalente a resolver el sistema de ecuaciones

$$\begin{aligned} Qx &= b \\ x^* &= Q^{-1}b \end{aligned}$$

para ello, consideremos el caso particular en $\mathbb{R}^2$

$$\min_{(x,y)\in\mathbb{R}^2} f(x,y) = \frac{1}{2}\lambda_1(x-a)^2 + \frac{1}{2}\lambda_2(y-b)^2. \tag{2}$$

con $\lambda_1, \lambda_2 > 0$.

Notemos que en este caso

$$\begin{aligned} f(x,y) &= f_1(x) + f_2(y) \\ \nabla f(x,y) &= \left[f_1'(x), f_2'(y)\right]^T \end{aligned}$$

Además $(x,y)^* = (a,b)$. Sea (x_k, y_k) dado y la actualización en la dirección de máximo descenso,

$$\begin{aligned} x_{k+1} &= x_k - \alpha_x f_1'(x_k) \\ y_{k+1} &= y_k - \alpha_y f_2'(y_k) \end{aligned}$$

donde α_x y α_y son los tamaños de paso.

Supongamos que $f_1'(x_k) \neq 0$ y $f_2'(y_k) \neq 0$. Si además, α_k y α_y son tamaños de paso exacto en cada dirección (se usa descenso coordenado)

$$\begin{aligned} \alpha_k &= \frac{x_k - a}{f_1'(x_k)} \\ x_{k+1} &= x_k - \alpha_x f_1'(x) \\ &= x_k - \frac{x_k - a}{f_1'(x_k)} f_1'(x_k) \\ &= a. \end{aligned}$$

similarmente $y_{k+1} = b$. Luego

$$(x_{k+1}, y_{k+1}) = (x, y)^* = (a, b). \qquad (3)$$

Es decir, para cualquier punto (x_k, y_k) se llega al óptimo en una iteración.

Observaciones

1. El ejemplo anterior es una forma cuadrática orientada en los ejes.

2. Si se aplica máximo descenso coordenado con tamaño de paso exacto, se llega al óptimo, en una iteración.

3. Notar que el problema es separable en cada variable, i.e.,

$$f(x, y) = \frac{1}{2}[x, y]\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix} - [a, b]\begin{bmatrix}x\\ y\end{bmatrix} + cte \qquad (4)$$

Es decir la matrix ed diagonal.

La pregunta que surge ahora es ¿se podrán usar las ideas anteriores al caso general?, es decir, usar las ideas anteriores para resolver

$$\min_{x\in\mathbb{R}^n} f(x) = \frac{1}{2}x^TQx - b^Tx \qquad (5)$$

1.2 Barzilai-Borwein

El método de Barzilai-Borwein se basa en encontrar un tamaño de paso α_k tal que $\alpha_k g_k$ aproxime a $H^{-1}g_k$ sin tener que calcular el Hessiano de la función.

2 Método

2.1 Gradiente Conjugado

Definición 2.1. Sea Q una matriz simétrica y definida positiva. Dos vectoress d_1, d_2 se dicen **conjugados** con respecto a Q o simplemente Q-ortogonales si $d_1^TQd_2 = 0$.

Definición 2.2. Un conjunto de vectores $d_0, d_1, \ldots, d_k$ son mutuamente Q-ortogonales si $d_i^TQd_j$ para $i \neq j$.

Definición 2.3. Un conjunto de vectores $d_0, d_1, \ldots, d_k$ son mutuamente Q-ortogonales si $d_i^TQd_j$ para $i \neq j$.

Proposición 1. Sea Q una matriz simétrica definida positiva. Si el conjunto de vectores $d_0, d_1, \ldots, d_k$ son mutuamente Q-ortogonales entonces son linealmente independientes.

2.1.1 Solución mediante Direcciones Conjugadas

1. Sea $\mathcal{D} = \{d_0, d_1, \ldots, d_{n-1}\}$ un conjunto de direcciones conjugadas (previamente conocidas o dadas) con respecto a una matriz simétrica y definida positiva Q.

2. De acuerdo a la proposición anterior el conjunto D es linealmente independiente. Por lo tanto D es una base de $\mathbb{R}^n$.

3. Supongamos adicionalmente que x^* es la solución al problema de optimización.

4. Luego, podemos expresar x^* de forma única, como una combinación lineal usando la base $\mathcal{D}$.

$$x^* = \sum_{i=1}^{n} \alpha_j d_j$$

Premultiplicando por d_i^TQ y por el hecho que $d_i^TQd_j = 0$ para $i \neq j$ (por ser direcciones conjugadas), obtenemos los coeficientes de la combinación lineal

$$\begin{aligned} d_i^TQx^* &= \sum_{j=0}^{n-1} \alpha_j d_i^TQd_j \\ \alpha_i d_i^TQd_i &= d_i^TQx^* \\ \alpha_i &= \frac{d_i^Tb}{d_i^TQd_i} \end{aligned}$$

Finalmente,

$$x^* = \sum_{j=0}^{n-1} \frac{d_j^T b}{d_j^T Q d_j} d_j.$$

2.1.2 Algoritmo Direcciones Conjugadas Iterativo

Dado un conjunto de direcciones conjugadas $\mathcal{D}$, se define $g_k = \nabla f(x_k) = Qx_k - b$.
Dado un punto inicial x_0, generamos la secuencia

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

donde

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

es el minimizador de $f(\cdot)$ a lo largo de la recta $x_k + \alpha d_k$. Luego

$$g_{k+1}^T d_k = 0.$$

Teorema 2.1. Para cualquier $x_0 \in \mathbb{R}^n$, la sucesión $\{x_k\}$ generada usando el algoritmo anterior converge a la solución x^* en a lo sumo n pasos.

2.1.3 Algoritmo Gradiente Conjugado

La idea del Algoritmo Gradiente Conjugado se basa en el Algoritmo de las direcciones conjugadas, pero sin conocer las direcciones conjugadas a priori.
Se comienza seleccionando la primera dirección $d_0 = -g_0$ y el resto

$$d_{k+1} = -g_{k+1} + \beta_{k+1} d_k \tag{6}$$

donde β_{k+1} se selecciona de modo que d_k y d_{k+1} sean Q-conjugados.

Para calcular β_{k+1} basta premultiplicar por $d_k^T Q$ en la igualdad $d_{k+1} = -g_{k+1} + \beta_{k+1} d_k$

$$\begin{aligned} d_{k+1} &= -g_{k+1} + \beta_{k+1} d_k & (7)\\ d_k^T Q d_{k+1} &= -d_k^T Q g_{k+1} + \beta_{k+1} d_k^T Q d_k & (8)\\ \beta_{k+1} &= \frac{g_{k+1}^T Q d_k}{d_k^T Q d_k}. & (9) \end{aligned}$$

Algunas relaciones importantes de Gradiente Conjugado son

- $g_{k+1} d_k = 0$.
- $\alpha_k = -\dfrac{g_k^T d_k}{d_k^T Q d_k}$.
- $\beta_{k+1} = \dfrac{g_{k+1}^T Q d_k}{d_k^T Q d_k}$.
- $d_{k+1} = -g_{k+1} + \beta_{k+1} d_k$.

Así que podemos recalcular α_k y β_k como

$$\begin{aligned} \alpha_k &= -\frac{g_k^T d_k}{d_k^T Q d_k}\\ &= -\frac{g_k^T(-g_k + \beta_k d_{k-1})}{d_k^T Q d_k}\\ &= \frac{g_k^T g_k - g_k^T \beta_k d_{k-1}}{d_k^T Q d_k}\\ &= \frac{g_k^T g_k}{d_k^T Q d_k} \end{aligned}$$

De manera similar,

$$\begin{aligned} \beta_{k+1} &= -\frac{g_{k+1}^T Q d_k}{d_k^T Q d_k}\\ &= -\frac{g_{k+1}^T Q(-g_k + \beta_k d_{k-1})}{d_k^T Q d_k}\\ &= \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} \end{aligned}$$

2.2 Función a Optimizar

Considere el siguiente problema de optimización,

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

donde

$$Q = PDP^T \quad (10)$$

$$P = \prod_{j=1}^{m} H_j \quad (11)$$

$$H_j = I - 2\frac{u_j u_j^T}{u_j^T u_j}. \quad (12)$$

I es la matriz identidad, H_j son matrices ortogonales Householder donde u_j son vectores con entradas generadas aleatoriamente en $(-1, 1)$. D es una matriz diagonal cuyos componentes i-ésimos están definidos por

$$log(d_i) = \left(\frac{i-1}{n-1}\right) ncond, i = 1, 2, \ldots n$$

Note que el parámetro $ncond$ se refiere al número de condición de la matriz Q, por ejemplo, si $ncond = 0$ entonces Q es la matriz identidad.
Para calcular b se genera una solución x^* de manera aleatoria sobre el intervalo $(-1, 1)$, entonces

$$b = Qx^*.$$

2.3 Gradiente de la Función

Al tratarse de una forma cuadrática sabemos que el gradiente de la función está dado por

$$\nabla f(x) = Qx - b$$

2.4 Pseudocódigo

2.4.1 Gradiente Conjugado

Input: x_0
Output: x^*

Haga $g_0 = Qx_0 - b$, $d_0 = -g_0$, $k = 0$
while $||g_k|| \neq 0$ **do**
  $\alpha_k = \dfrac{g_k^T g_k}{d_k^T Q d_k} = -\dfrac{g_k^T d_k}{d_k^T Q d_k}$
  $x_{k+1} = x_k + \alpha_k d_k$
  $g_{k+1} = Qx_{k+1} - b = \nabla f(x_{k+1})$
  $\beta_{k+1} = \dfrac{g_{k+1}^T g_{k+1}}{g_k^T g_k} = -\dfrac{g_{k+1}^T Q d_k}{d_k^T Q d_k}$
  $d_{k+1} = -g_{k+1} + \beta_{k+1} d_k$
  $k = k + 1$
end while

2.4.2 Barzilai Borwein

Input: x_0, τ_0
Output: x^*

Haga $\alpha_0 = 0$, $y_0 = x_0$, $k = 0$
while $||g_k|| \neq 0$ **do**
  if $k > 0$ **then**
    $s_{k-1} = x_k - x_{k-1}$
    $y_{k-1} = g_k - g_{k-1}$
    $\alpha_k = \dfrac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}$
  else
    Calcular $alpha_k$ con paso exacto
  end if
  $x_{k+1} = x_k - \alpha_k g_k$
  $k = k + 1$
end while

2.4.3 Paso Exacto

Input: x_0
Output: x^*

Haga $k = 0$
while $||g_k|| \neq 0$ **do**
  $d_k = -g_k$
  $alpha_k = \dfrac{g_k^T g_k}{g_k^T Q g_k}$
  $x_{k+1} = x_k + \alpha_k d_k$
  $k = k + 1$
end while

3 Resultados

Para comparar resultados en tiempo e iteraciones se utilizaron los métodos de Gradiente Conjugado, Barzilai-Borwein y Gradiente Descendiente con paso Exacto.

Parámetro	Valor
α_0	0.9
$ncond$	$\{2, 4, 6\}$
n	$10,000$
m	3

Parámetro	max_{iter}	τ_x	τ_f	τ_{grad}
Valor	500	10^{-8}	10^{-8}	10^{-8}

Para cada número de condición se repitió el experimento 30 veces. Los resultados obtenidos fueron

3.1 $ncond = 2$

Algoritmo	Tiempo	Iteraciones
Gradiente Conjugado	5.60 seg	27.0
Barzilai-Borwein	4.11 seg	29.13
Paso Exacto	9.58 seg	28.0

3.2 $ncond = 4$

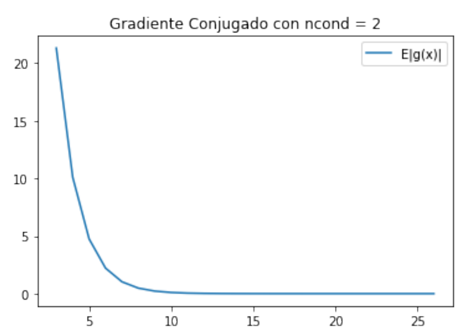
Algoritmo	Tiempo	Iteraciones
Gradiente Conjugado	14.28 seg	68.66
Barzilai-Borwein	11.17 seg	77.5
Paso Exacto	49.69 seg	142.43

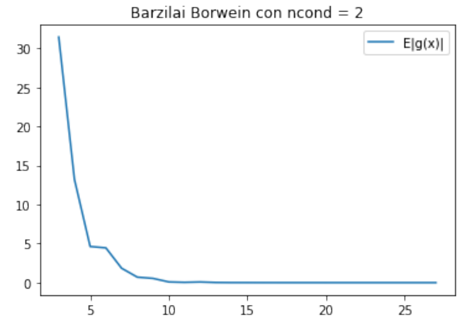
3.3 $ncond = 6$

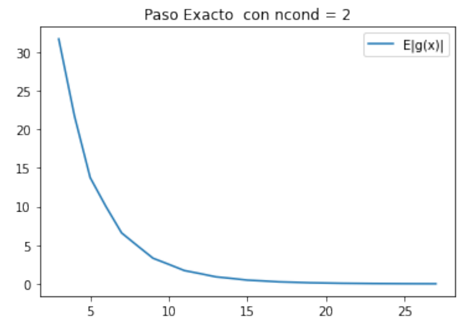
Algoritmo	Tiempo	Iteraciones
Gradiente Conjugado	35.96 seg	173.13
Barzilai-Borwein	28.56 seg	200.3
Paso Exacto	175.02 seg	500.0

3.4 Promedio de norma de Gradiente

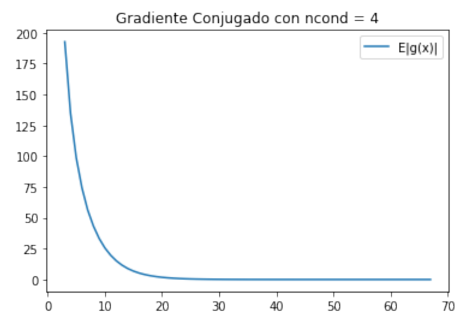
3.4.1 $ncond = 2$

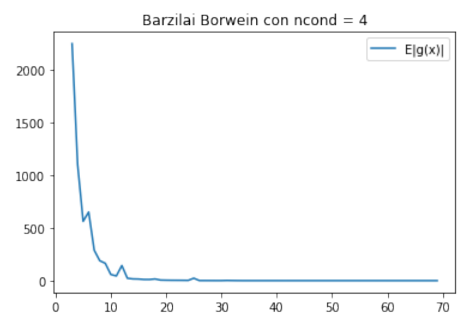


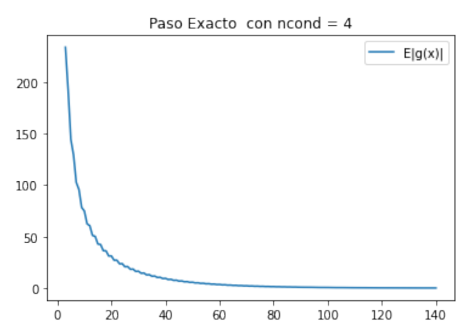




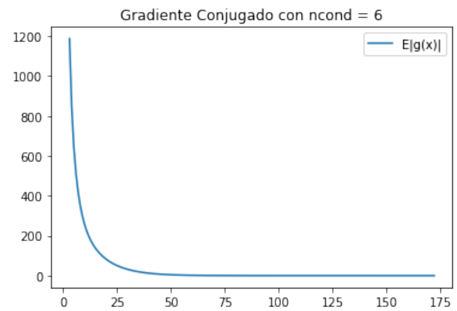
3.4.2 $ncond = 4$

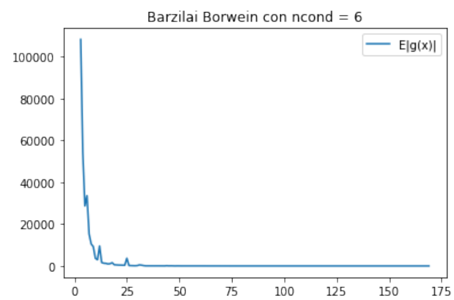


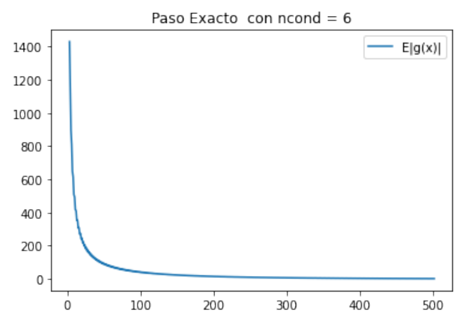




3.4.3 $ncond = 6$







4 Conclusiones

4.1 Gradiente Conjugado

Este método utiliza la idea de diagonalizar la matriz Q para resolver el problema como en el caso base de $\mathbb{R}^2$ a través de usar la Q- conjugación. Una característica particular de este método es que genera vectores $p_0, p_1, \ldots, p_k$ tal que cada uno de ellos cumple ser conjugado con la matriz Q y además este conjunto de vectores son mutuamente ortogonales y linealmente independientes. Para la forma cuadrática dada el algoritmo mostró rendimiento superior en términos de tiempo e iteraciones respecto al algoritmo de paso exacto. Notemos además, que conforme el número de condición se aumenta, el número de iteraciones crece para todos los algoritmos. Esto se ve reflejado en la matriz D y por lo tanto en $Q = P^T DP$, es decir, si $ncond$ crece, las entradas de D crecen en valor absoluto y también crecen las entradas de Q.

4.2 Barzilai-Borwein

Para la forma cuadrática dada el algoritmo fue el mejor en términos de tiempo e iteraciones. Este método utiliza la idea de aproximar la matriz Hessiana utilizando solamente el gradiente. Lo anterior explica por qué el rendimiento fue superior, dado que no solamente trata de utilizar información del gradiente si no que incorpora información aportada por el Hessiano.

4.3 Paso Exacto

Este algoritmo es de los primeros visto en clases. Para el caso de una forma cuadrática dada, se conoce la expresión para el paso α_k. Notemos que de los tres métodos utilizados este fue el más lento respecto a tiempo e iteraciones.