Tarea 6-Optimización 1

1st Daniel Vallejo Aldana
Departamento de Matemáticas
Universidad de Guanajuato
daniel.vallejo@cimat.mx

***Resumen*—En el presente trabajo se presentará el algoritmo de gradiente conjugado lineal, y se probará con una función descrita en lineas posteriores para eliminar el ruido a dicha función.**

***Index Terms*—Gradiente conjugado, modelo cuadrático**

I. Introducción

Se quiere resolver el problema de optimización sin restricciones

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

Para una matriz Q la cual es simétrica y positiva definida.

El algoritmo de gradiente conjugado se basa en, las direcciones conjugadas pero sin conocer dichas direcciones. Decimos que dos direcciones d_i y d_j son Q-conjugadas si

$$d_i^tQd_j = 0$$

En este algoritmo se comienza dando la dirección inicial $d_0 = -g_0$ y posteriormente se realiza la actualización de las direcciones d_k como

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

Donde la selección de la β_{k+1} es tal que d_{k+1} y d_k son Q-conjugadas.

II. Método/Algoritmo

Antes de presentar el método de gradiente conjugado se presentará la aproximación que se utilizó para encontrar el producto Qd_k sin la necesidad de conocer explicitamene la matriz Q.

Notamos que al derivar la expresión $f(x) = \frac{1}{2}x^tQx + -b^tx$ respecto a x se tiene que

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

Por lo tanto tenemos entonces que $Qd_k = \nabla f(d_k)+b$, luego vemos que $b = -\nabla f(0)$ con 0 el vector de 0's en $\mathbb{R}^n$. De esta forma se obtiene una forma de calcular Qd_k sin necesidad de contar con la matriz Q de forma explícita.

Utilizando lo anterior se tiene que el algoritmo de gradiente conjugado está dado de la siguiente manera.

1: **Input**: x_0, $g_0 = Qx_0 - b$, $d_0 = -g_0$
2: **Output**: x^*
3: $k = 0$
4: **while** $\|g_k\| \neq 0$
5: $\alpha_k = \frac{g_k^tg_k}{d_k^t(\nabla f(d_k)+b)}$
6: $x_{k+1} = x_k + \alpha_k d_k$
7: $g_{k+1} = \nabla f(x_k + 1)$
8: $\beta_{k+1} = \frac{g_{k+1}^tg_{k+1}}{g_k^tg_k}$
9: $d_{k+1} = -g_{k+1} + \beta_{k+1}d_k$
10: k=k+1
11: **end while**

El algoritmo anterior se utilizó para resolver el siguiente problema de optimización

$$x^* = \arg\min_x \sum_{i,j}\left((x_{i,j} - g_{i,j})^2 + \lambda \sum_{(l,m)\in\Omega_{i,j}}(x_{i,j} - x_{l,m})^2\right)$$

Calculando la derivada parcial de la función anterior respecto a la variable $x_{k,s}$ tenemos la siguiente expresión

$$\frac{\partial f}{\partial x_{k,s}} = 2(x_{k,s} - g_{k,s}) + \lambda \sum_{(l,m)\in\Omega_{i,j}} 2(x_{k,s} - x_{l,m})$$

Lo anterior se cumple para $1 \leq k \leq m$ y $1 \leq s \leq n$ y $\lambda \in \{1, 100, 1000\}$.

III. ANÁLISIS DE RESULTADOS

Se generaron 3 gráficas correspondientes a los tres valores posibles para $\lambda \in \{1, 100, 1000\}$, los parámetros del algoritmo fueron una matriz de ceros de tamaño $m \times n$ y una tolerancia de 1×10^{-3} para que el algoritmo terminara en un tiempo razonable para $\lambda = 1000$. Las

gráficas generadas se presentan en el Apéndice A.

IV. CONCLUSIONES

Con base en los resultados obtenidos en el presente trabajo podemos concluir que el algoritmo de gradiente conjugado da buenos resultados al momento de resolver el problema cuadrático de optimización presentado en este trabajo, sin embargo, computacional mente fue costoso para el caso $\lambda = 1000$ ya que el número de iteraciones que se requieren para la precisión dada son de alrededor de 1000.

APÉNDICES

Apéndice A
Solución del problema de optimización cuadrática

A continuación se presenta la gráfica de la función original generada.

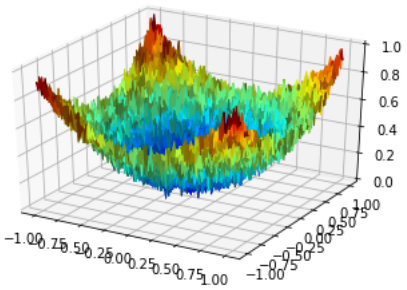

Figura 1. La gráfica anterior muestra la función original la cual se quiere suavizar

Caso 1: $\lambda = 1$

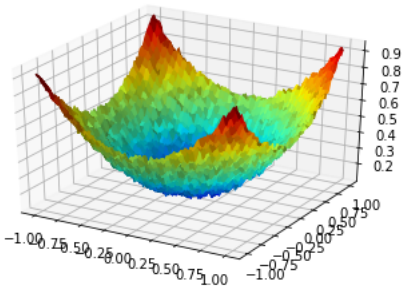

Figura 2. Función suavizada para $\lambda = 1$, 15 iteraciones en total

Caso 1: $\lambda = 100$

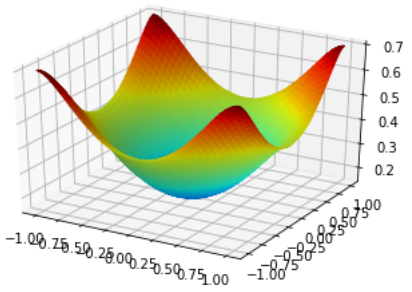

Figura 3. Función suavizada para $\lambda = 100$, 299 iteraciones en total

Caso 1: $\lambda = 1000$

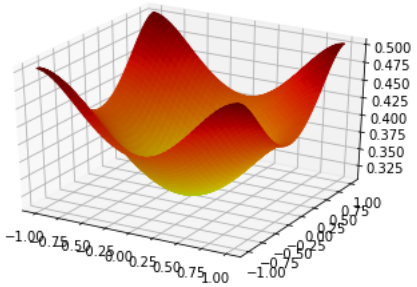

Figura 4. Función suavizada para $\lambda = 1000$, 1000 iteraciones en total con una precisión de 1×10^{-2}

REFERENCIAS

[1] Nocedal Jorge, *Numerical Optimization*, Segunda edición, Springer, 2006