Tarea 9. Optimización, Algoritmos Cuasi-Newton

Oscar Esaú Peralta Rosales
Maestría en Computación
Centro de Investigación en Matemáticas

***Resumen*—Se presentan la implementación de dos algoritmos Cuasi-Newton, el algoritmo DFP (Davidon-Fletcher-Powel) obtenida a través de la aproximación del Hessiano usando la corrección de rango 2 y el algoritmo BFGS (Broyden-Fletcher-Goldfarb-Shanno) obtenida a través de aproximar la matriz inversa del Hessiano mediante la matriz obtenida de la corrección de rango 2. En el apéndice se detalla un problema relacionado al tema.**

***Index Terms*—Gradiente Conjugado No lineal**

I. Introduction

El Método de Newton es un algoritmo de optimización sin restricciones que nos permite obtener un óptimo de una función convexa, donde en cada paso la actualización está determinada por $x_{k+1} = x_k - \alpha_k H_k^{-1} g_k$. Sin embargo este método require realizar el cálculo del Hessiano en cada iteración y su inversa lo cual puede ser muy costoso además de que no garantiza que la dirección $d_k = -H_k^{-1} g_k$ sea de descenso. Una de la soluciones a estos problemas son los algoritmos Cuasi-Newton. Estos métodos al igual que Descenso de Gradiente solo requieren conocer el gradiente de la función objetivo en cada iteración. Así, midiendo los cambios en los gradientes, construyen un modelo de la función objetivo que es suficientemente bueno para producir una convergencia superlineal [1].

II. Metodología

En el algoritmo DFP realiza la aproximación al Hessiano mediante la corrección de rango 2:

$$H_{k+1} = H_{k+1} + \alpha x x^T + \beta y y^T$$

satisfaciendo $H_{k+1} y_i = s_i$ para $i = 0, 1, 2, ..., k$ y $y_k = g_{k+1} - g_k$, $s_k = x_{k+1} - x_k$. Generando las matrices de corrección

$$H_{k+1}^{DFP} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k} \tag{1}$$

$$B_{k+1}^{DFP} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k} \tag{2}$$

y usando la fórmula de Sherman-Morrison-Woodbury se puede reescribir como:

$$B_{k+1}^{DFP} = (I - \rho_k y_k s_k^T) B_k (I - \rho_k s_k y_k^T) + \rho_k y_k y_k^T \tag{3}$$

con $\rho_k = \frac{1}{y_k^T s_k}$.

El algoritmo de BFGS realiza la aproximación a la inversa del Hessiano a partir de (3), obteniendo:

$$H_{k+1}^{BFGS} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T \tag{4}$$

con $\rho_k = \frac{1}{y_k^T s_k}$.

Algorithm 1: Algoritmo DFP

```
Result: x*
k = 0
while ||g_k|| > tol do
    d_k = -B_k^{-1} g_k
    α_k = Calcular usando un método de búsqueda
      en linea
    x_{k+1} = x_k + α_k d_k
    Calcular y_k y s_k
    Calcular B_{k+1} con (3)
    k = k + 1
end
```

El Algoritmo de DFP es mostrado en 1 y el Algoritmo de BFGS en 2, notemos que en el caso de DFP podemos usar directamente la inversa del Hessiano 1.

Algorithm 2: Algoritmo BFGS

```
Result: x*
k = 0
while ||g_k|| > tol do
    d_k = -H_k g_k
    α_k = Calcular usando un método de búsqueda
      en linea
    x_{k+1} = x_k + α_k d_k
    Calcular y_k y s_k
    Calcular H_{k+1} con (4)
    k = k + 1
end
```

La implementación (mostrada en el notebook adjunta a este reporte) es directa a través de los los Algoritmos de DFP y BGFS, a excepción, por el cálculo del tamaño de paso α_k. Para obtención del tamaño de paso se usaron los métodos de Backtracking e Interpolación Cúbica (no hubo diferencia substancial en su uso), con valores de c_1 y c_2 entre 0.3 y 0.4 para la validación de la condición

de Armijo. Como matriz inicial para B_0 y H_0 se usó la matriz identidad.

Las funciones a optimizar fueron las ya conocidas función de Rosembrock con $n = 100$ y la función de Wood. Para la evaluación se realizaron 30 corridas sobre cada función con puntos iniciales aleatorios. Los resultados obtenidos se muestran en la siguiente sección.

III. Resultados

De las 30 ejecuciones indicadas previamente se reportan las siguiente tablas donde se indica el promedio de iteraciones realizadas, el promedio de la norma del gradiente en x^* y el promedio de tiempo de ejecución.

Cuadro I
Tabla comparativa de resultados de 30 ejecuciones con la función de Rosembrock

Método	***Iters***	$\|\|\nabla g(x*)\|\|$	***Tiempo***
DFP	247.10	4.9450e-09	3.65s
BFGS	244.53	6.1253e-9	3.57s

Cuadro II
Tabla comparativa de resultados de 30 ejecuciones con la función de Wood

Método	***Iters***	$\|\|\nabla g(x*)\|\|$	***Tiempo***
DFP	26.26	2.4940e-09	0.0086s
BFGS	26.70	2.2028e-9	0.0051s

IV. Conclusiones

Como se observa en los resultados el número de iteraciones y el tiempo de ejecución para el algoritmo BFGS es menor que el algoritmo DFP, mostrando una ligera mejora tal cual como se menciona en la literatura. El problema detectado fue el encontrar un algoritmo y correctos valores de c_1 y c_2 para la obtención de tamaño de paso, pues con valores no tan adecuados no se llega a la convergencia de estos e incluso se provocan errores numéricos.

V. Apéndice

Problemas

1. Las matrices de corrección de rango 1 se escriben

$$B_{k+1}^{RS1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k} \tag{5}$$

$$H_{k+1}^{RS1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{y_k^T (s_k - H_k y_k)} \tag{6}$$

donde $y_k = g_{k+1} - g_k$ y $s_k = x_{k+1} - x_k$

- Derive la matriz H_{k+1}^{RS1} a partir de B_{k+1}^{RS1} usando la fórmula de Sherman-Morrison
- Si H_k es una matriz definida positiva y $y_k^T(s_k - H_k y_k) > 0$ muestre que H_{k+1} es definida positiva.

Solución

La fórmula de Sherman-Morrison dice como sigue: sea $A \in \mathcal{R}^{nxn}$ es una matriz cuadrada e invertible y u, $v \in \mathcal{R}^n$ son dos vectores columna entonces $A + uv^T$ es invertible ssi $1 + v^T A^{-1} u \neq 0$ y

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1} u v^T A^{-1}}{1 + v^T A^{-1} u} \tag{7}$$

Usando (7) definimos (omitimos el super índice RS1):

$$B_{k+1} = B_k + uv^T$$

con $u = \alpha(y_k - B_k s_k)$, $v = (y_k - B_k s_k)$ y $\alpha = \frac{1}{(y_k - B_k s_k)^T s_k}$. Luego:

$$B_{k+1}^{-1} = (B_k + uv^T)^{-1}$$

$$H_{k+1} = B_k^{-1} - \frac{B_k^{-1} u v^T B_k^{-1}}{1 + v^T B_k^{-1} u}$$

sustituyendo u y v

$$H_{k+1} = B_k^{-1} - \frac{B_k^{-1}\alpha(y_k - B_k s_k)(y_k - B_k s_k)^T B_k^{-1}}{1 + (y_k - B_k s_k)^T B_k^{-1}\alpha(y_k - B_k s_k)}$$

$$H_{k+1} = H_k - \frac{\alpha(H_k y_k - s_k)(y_k H_k - s_k)^T}{\alpha(\alpha^{-1} + (y_k - B_k s_k)^T H_k (y_k - B_k s_k))}$$

$$H_{k+1} = H_k - \frac{(H_k y_k - s_k)(y_k H_k - s_k)^T}{y_k B_k s_k + (y_k - B_k s_k)^T H_k (y_k - B_k s_k)}$$

$$H_{k+1} = H_k - \frac{(H_k y_k - s_k)(y_k H_k - s_k)^T}{y_k^T s_k - s_k^T B_k s_k + y_k^T H_k y_k - 2 s_k^T y_k + s_k B_k s_k}$$

$$H_{k+1} = H_k - \frac{(H_k y_k - s_k)(y_k H_k - s_k)^T}{y_k^T H_k y_k - y_k^T s_k}$$

y finalmente al factorizar los signos llegamos a:

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{y_k^T (s_k - H_k y_k)} \tag{8}$$

Por otro lado sabemos que por definición H_k es una matriz simétrica definida positiva. El término $y_k^T(s_k - H_k y_k)$ es mayor a cero, por tanto, solo queda concentrarnos que ver que el término $(H_k y_k - s_k)(y_k H_k - s_k)^T$ es definida o semidefinida positiva.

Sabemos que para una matriz cuadrada X el producto XX^T es una matriz simétrica cuya diagonal son los términos al cuadrado de la diagonal de X. Por definición una matriz simétrica es definida o semidefinida positiva si todos los pivotes son mayor o mayor o igual a cero respectivamente. Por tanto $(H_k y_k - s_k)(y_k H_k - s_k)^T$ es simétrica y al menos semidefinida positiva y H_{k+1} entonces también es simétrica y definida positiva.

Referencias

[1] Jorge Nocedal, Stephen J. Wright, "Numerical Optimization," Second Edition, Springer.