

Optimización

Método del Gradiente Conjugado No-Lineal

Tarea 10

5 de mayo de 2021

Erika Rivadeneira Pérez
erika.rivadeneira@cimat.mx
Matemáticas Aplicadas - CIMAT

1. Resumen

En el presente reporte se implementan el método del gradiente conjugado no-lineal con cuatro opciones de calcular β vistas en clase: Fletcher-Reeves (FR), Polak-Ribiere (PR), Hestenes-Stiefel (HS) y la que combina Fletcher-Reeves con Polak-Ribiere en una función dada. Luego, se comparan las gráficas de las iteraciones vs. la función objetivo, de las iteraciones vs la norma del gradiente. Además, de las iteraciones y tiempo de ejecución de cada método y una imagen original con su respectiva imagen con el ruido filtrado.

2. Introducción

El método del gradiente conjugado debido a Fletcher y Reeves (1964) combina las características de la convergencia cuadrática del método de las direcciones conjugadas con las del método del gradiente. El método supone una importante mejora del método del gradiente con sólo un pequeño incremento en el esfuerzo de cálculo. El método del gradiente conjugado, esencialmente, combina la información obtenida del vector gradiente con la información acerca del vector gradiente de iteraciones previas. Lo que hace el método es calcular la nueva dirección de búsqueda utilizando una combinación lineal del gradiente en la etapa considerada y el de la etapa anterior. La principal ventaja del método es que necesita almacenar muy poca cantidad de información con lo que puede ser programado fácilmente incluso en calculadoras.

Para la implementación del método del gradiente conjugado para el caso no-lineal se hace uso de un parámetro β el cual puede ser calculado de cuatro

maneras diferentes. Estas maneras son por Fletcher-Reeves (FR), Polak-Ribiere (PR), Hestenes-Stiefel (HS) y la combinación de FR y PR.

En la siguiente sección se menciona el algoritmo considerado para la implementación del gradiente conjugado no-lineal y así obtener y comparar los resultados deseados.

3. Metodología

Se desea resolver el siguiente problema de optimización no lineal

$$\min_X \sum_{i,j} \left[(x_{i,j} - g_{i,j})^2 + \lambda \sum_{(l,m) \in \Omega_{i,j}} \sqrt{(x_{i,j} - x_{l,m})^2 + \mu} \right] \quad (1)$$

donde $\mu = 0,01$, $\lambda > 0$ es un parámetro de regularización dado; g es la función a la cual se le desea filtrar el ruido, por ejemplo una imagen; $\Omega_{i,j}$ es el conjunto de índices formados por los vecinos del píxel (i, j) , es decir $\Omega_{i,j} = \{(i+1, j), (i-1, j), (i, j+1), (i, j-1)\}$. Observe que si la imagen original es de tamaño $m \times n$ entonces la imagen con el ruido filtrado será $X^* = [x_{i,j}]_{1 \leq i \leq m, 1 \leq j \leq n}$ donde X^* es el minimizador del problema.

La minimización de esta función se la realiza mediante el uso del método del gradiente conjugado para el caso no-lineal, cuyo algoritmo se muestra a continuación

Gradiente Conjugado No-Lineal

Input: x_0

- 1: $d_0 = -\nabla f(x_0), k = 0$
- 2: **while** $\|\nabla f(x_k)\| \neq 0$ no converja **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: Calcular β_{k+1}
- 7: $d_{k+1} = -\nabla f(x_{k+1}) + \beta_{k+1} d_k$
- 8: $k = k + 1$
- 9: **end while**

Output: x^*

El cálculo de β_k se lo puede realizar de las siguientes maneras:

- Mediante Fletcher-Reeves:

$$\beta_{k+1}^{FR} = \frac{\nabla^T f(x_{k+1}) \nabla f(x_{k+1})}{\nabla^T f(x_k) \nabla f(x_k)}$$

- Mediante Polak-Ribiere:

$$\beta_{k+1}^{PR} = \frac{\nabla^T f(x_{k+1}) (\nabla f(x_{k+1}) - \nabla f(x_k))}{\nabla^T f(x_k) \nabla f(x_k)}$$

- Mediante Hestenes-Stiefel:

$$\beta_{k+1}^{HS} = \frac{\nabla^T f(x_{k+1}) (\nabla f(x_{k+1}) - \nabla f(x_k))}{(\nabla f(x_{k+1}) - \nabla f(x_k))^T d_k}$$

- Mediante Fletcher-Reeves Polak-Ribiere:

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

Para la implementación del método se consideraron como criterios de paro una tolerancia de 1×10^{-3} y terminar las iteraciones cuando $\|g_{k+1} - g_k\| < 1 \times 10^{-3}$.

4. Resultados

La función *GCFR* del programa realizado recibe los siguientes parámetros:

1. $X_0 = g$: el punto inicial.
2. $f = fun$: la función objetivo
3. *grad*: el gradiente
4. *metodo*: un string con las opciones: 'PR' (Polak-Ribiere), 'FR' (FletcherReeves), 'HS' (Hestenes-Stiefel) y 'FRPR' (la mezcla de Fletcher-Reeves y Polak-Ribiere).



Figura 1: Imagen original

Se tomaron como variables globales a: la tolerancia aceptada ($tol = 1 \times 10^{-3}$), $\lambda = 10$, la función g del modelo de optimización (1), el cual fue una imagen dada mostrada en la Figura (1).

A continuación se muestran las gráficas de las iteraciones vs. la función objetivo, iteraciones vs. la norma del gradiente y las imágenes resultantes después de eliminar el ruido de la figura (1) de cada método de obtención de β .

- Mediante Fletcher-Reeves: ver figuras (2), (3) y (4)

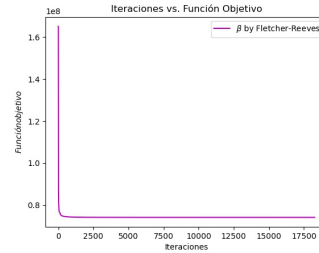


Figura 2: Iteraciones vs. Función objetivo por **FR**

- Mediante Polak-Ribiere: ver figuras (5), (6) y (7)
- Mediante Hestenes-Stiefel: ver figuras (8), (9) y (10)
- Mediante Fletcher-Reeves Polak-Ribiere: ver figuras (11), (12) y (13)

Por otro lado, en la tabla (1) se muestran las iteraciones y el tiempo de cada método.

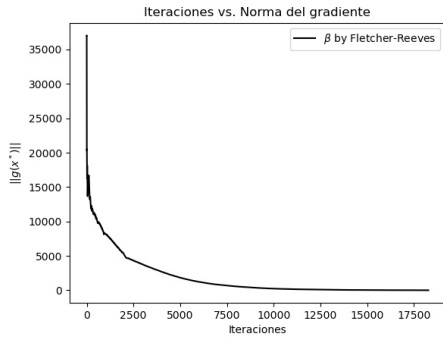


Figura 3: Iteraciones vs. $\|g(x^*)\|$ por **FR**

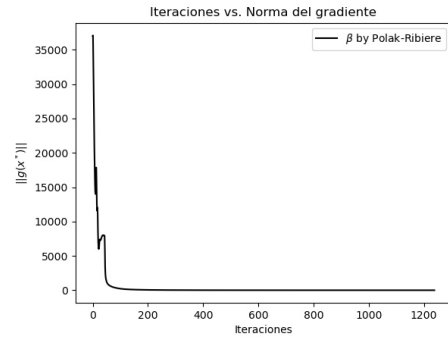


Figura 6: Iteraciones vs. $\|g(x^*)\|$ por **PR**



Figura 4: Imagen con ruido filtrado por **FR**



Figura 7: Imagen con ruido filtrado por **PR**

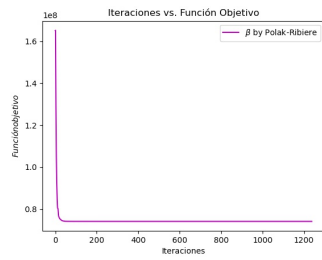


Figura 5: Iteraciones vs. Función objetivo por **PR**

	PR	FR	HS	FRPR
Itera.	1239	18274	280	1324
Tiempo	200.07	2651.98	52.62	200.64

Tabla 1: Iteraciones y tiempo de compilación de cada método



Figura 8: Imagen con ruido filtrado por **HS**

5. Conclusiones y Discusión

Todos los métodos de obtención del parámetro β convergieron eventualmente. La tabla (1) muestra

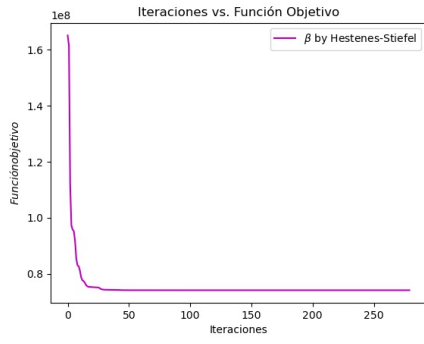


Figura 9: Iteraciones vs. Función objetivo por **HS**

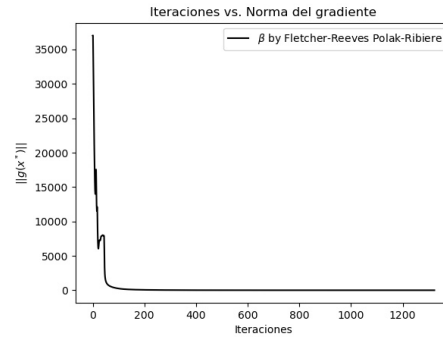


Figura 12: Iteraciones vs. $\|g(x^*)\|$ por **PRFR**

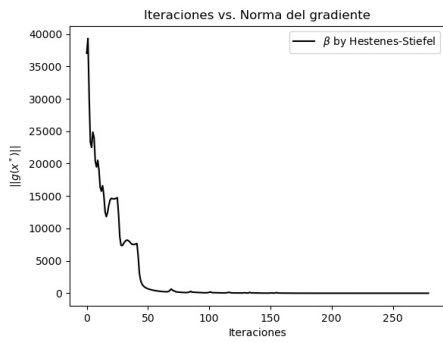


Figura 10: Iteraciones vs. $\|g(x^*)\|$ por **HS**

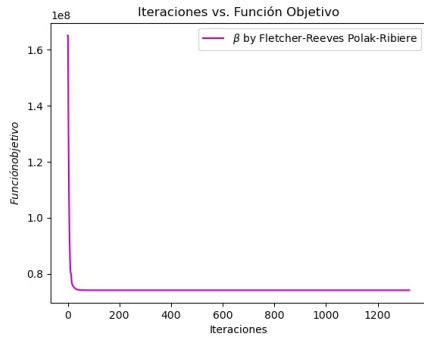


Figura 11: Iteraciones vs. Función objetivo por **PRFR**



Figura 13: Imagen con ruido filtrado por **FRPR**

Referencias

- [1] J. Nocedal and S. J. Wright. Numerical Optimization. Springer Series in Operation Research, 2000.

que Fletcher-Reeves fue el método que más tiempo se demoró en converger y el que menos tiempo ocupó fue el de Hestenes-Stiefel. Este último fue el método más eficiente debido a que convergió en menor tiempo y con menos iteraciones.

Finalmente, es importante mencionar que el método del gradiente conjugado no-lineal es muy eficiente para filtrar el ruido de imágenes, esto se evidencia en el cambio de la figura original (1) a las figuras (4), (7), (10) y (13) las cuales ya filtran el ruido.