



## Gradiente Conjugado

1. El gradiente conjugado es un algoritmo para la solución numérica de cierto tipo de sistemas de ecuaciones lineales.  
De aquellos cuya matriz del sistema es simétrica y positiva definida.
2. El gradiente conjugado se implementa como un algoritmo iterativo, aplicable a sistemas cuya matriz contiene muchos ceros, y muy grandes para ser manejados por métodos directos u otros métodos como el de descomposición de Choleski.
3. Se aplica a métodos de optimización no restringida, fue desarrollado por Magnus Hestenes y Eduard Stiefel, que lo programaron en una máquina Z4.
4. El método del gradiente biconjugado, es la generalización para matrices no simétricas.
5. Varios métodos de gradiente conjugado buscan mínimos de ecuaciones no lineales.

Descripción del problema.

1. Supongase que queremos resolver el sistema de ecuaciones lineales  $Ax = b$  para el vector  $x$ . Sabemos que la matriz  $A$  es simétrica y positiva definida (i.e. )

$$A^t = A, x^t Ax > 0$$

para todos los vectores  $x$  no cero y

$$x \in \mathbb{R}^N, b \in \mathbb{R},$$

2. El método directo. Decimos que dos vectores no cero, son conjugados con respecto a la matriz  $A$ , si

$$u^t Av = 0$$

Dado que  $A$  es simétrica y positiva definida, el lado derecho define un producto interno

$$u^t Av = \langle u, v \rangle_A := \langle Au, v \rangle = \langle u, A^t v \rangle = \langle u, Av \rangle$$

3. Dos vectores son conjugados si y sólo si, son ortogonales con respecto al producto interno. Suponga que

$$P = \{p_1, \dots, p\}$$

es un conjunto de vectores mutuamente conjugados (con respecto a  $A$ ). Entonces  $P$  forma una base, y podemos expresar el vector solución  $X^*$  en esta base

$$f(a) = \lim_{x \rightarrow a} \frac{f(x) - f(a)}{x - a} =$$

$$X^* = \sum_{i=1}^n \alpha_i p_i,$$

$$\text{Y } AX^* = \sum_{i=1}^n \alpha_i A p_i$$

Multiplicando a la derecha por  $p_k^t$

$$p_k^t A X^* = \sum_{i=1}^n \alpha_i p_k^t A p_i$$

sustituyendo en  $Ax = b$

$$p_k^t b = \sum_{i=1}^n \alpha_i \langle p_k, p_i \rangle A$$

tenemos que  $\forall i \neq k, \langle p_k, p_i \rangle A = 0$

produce  $\langle p_k, b \rangle = \alpha_k \langle p_k, p_i \rangle A$

$$\text{llegando a } \alpha_k = \frac{\langle p_k, b \rangle}{\langle p_k, p_k \rangle A}$$

4. Lo anterior es lo que nos da el método para resolver el sistema  $Ax = b$

encontrando una secuencia de direcciones conjugadas y calculando los coeficientes  $\alpha_k$

El método iterativo

1. Si escogemos los vectores  $p_k$ , adecuadamente, no necesitaremos de todos

ellos, para obtener una buena aproximación a la solución  $x^*$ .

Así que si podemos tomar el método del gradiente conjugado como un método iterativo, para resolver sistemas grandes, que a los métodos directos le lleva mucho más tiempo resolver.

2. Inicio. Tomamos  $x_0$

como la aproximación inicial de  $x^*$  podemos tomar  $x_0 = (0, \dots, 0)$

Iniciando con  $x_0$  buscando la solución y en cada iteración debemos medir que tan cerca estamos de  $x^*$

Esta métrica viene de el hecho que la solución  $x^*$  es también el único minimizador de la función cuadrática  $f(x) = \frac{1}{2} x^t A x - x^t b, x \in \mathbb{R}^N$

3. La existencia de un minimizador único, es dada por  $\nabla^2 f(x) = A$ , cuando A es una matriz positiva definida. Y el minimizador resuelve la ecuación

$$\nabla f(x) = Ax - b,$$

4. Esto sugiere tomar el primer vector base  $p_0$  como el negativo del gradiente de  $f(x = x_0)$

El gradiente es igual a  $Ax - b$  iniciando con  $x_0$  tendremos  $p_0 = b - x_0$

Los otros vectores serán conjugados al gradiente, de ahí el nombre del método de gradiente conjugado.

5. El residuo en el paso k-esimo  $r_k = b - Ax_k$  se observa que

$r_k = -\nabla f(x) = b - Ax_k$  así, por el método del gradientes, se

requiere que se mueva en esta dirección. Sin embargo, requerimos que sea en las direcciones conjugadas  $p_k$ , lo que construimos

$$p_k = r_k - \sum_{i < k} \frac{p_i^t A r_k}{p_i^t A p_i} p_i$$

6. Siguiendo estas direcciones la siguiente aproximación será

$$x_{k+1} = x_k + \alpha_k p_k.$$

con 
$$\alpha_k = \frac{p_k^t (b - Ax_k)}{p_k^t A p_k} = \frac{p_k^t r_k}{p_k^t A p_k}$$

El algoritmo, se detalla a continuación para resolver  $Ax = b$

donde la matriz A es positiva definida y simétrica, el vector  $x_0$  es la la aproximación inicial a  $x^*$



## Pasos del Algoritmo

$$r_0 = b - Ax_0$$

If  $r_0$  es suficientemente pequeño, entonces regrese  $x_0$  como resultado.

$$p_0 = r_0, k = 0$$

Hacer

$$\alpha_k = \frac{r_k^t r_k}{p_k^t A p_k}$$

$$x_{k+1} = x_k + \alpha_k p_k.$$

$$r_{k+1} = r_k - \alpha_k A p_k$$

if  $r_{k+1}$  es suficientemente pequeño, then salir del loop

$$\beta_k = \frac{r_{k+1}^t r_{k+1}}{r_k^t r_k}$$

$$p_{k+1} = r_{k+1} + \beta_k p_k$$

$$k = k + 1$$

fin de hacer

regresar  $x_{k+1}$  como resultado.

El cálculo de alfa y beta.

En el algoritmo alfa es escogido tal que

$$r_{k+1} \perp r_k$$

el denominador es simplificado desde

$$\alpha_k = \frac{r_k^t r_k}{p_k^t A p_k}$$

Dado que  $r_{k+1} = p_{k+1} - \beta_k p_k$  inicialmente  $\beta_k = \frac{r_{k+1}^t A p_k}{p_k^t A p_k}$ ,

usando  $r_{k+1} = r_k - \alpha_k A p_k$ , y equivalentemente  $A p_k = \frac{1}{\alpha_k} (r_k - r_{k+1})$

el numerador de beta es reescrito

$$r_{k+1}^t A p_k = \frac{1}{\alpha_k} r_{k+1}^t (r_k - r_{k+1}) = -\frac{1}{\alpha_k} r_{k+1}^t r_{k+1}$$

Como  $r_k$   $r_{k+1}$  son ortogonales, llegamos a

$$p_k^t A p_k = \frac{1}{\alpha_k} r_k^t (r_k - r_{k+1}) = \frac{1}{\alpha_k} r_k^t r_k$$

## Ejemplo Numérico

Resolver el sistema dado por  $Ax = \begin{bmatrix} 4 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

Con  $x_0 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ . Este sistema proviene de la función cuadrática  $f(x_1, x_2) = 4x_1^2 + 2x_1x_2 + 3x_2^2 - x - 2x_2$ .

La solución exacta es  $x^* = \begin{bmatrix} \frac{1}{11} \\ \frac{7}{11} \end{bmatrix} = \begin{bmatrix} 0.0909 \\ 0.6364 \end{bmatrix}$ .

Primero calculamos el vector residuo

$$r_0 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} - \begin{bmatrix} 4 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} -8 \\ -3 \end{bmatrix}$$

Calculamos alfa inicial, con  $p_0$  igual  $r_0$

$$\alpha_0 = \frac{r_0^t r_0}{p_0^t A p_0} = \frac{73}{331},$$

ahora calculamos  $x_1 = x_0 + \alpha_0 p_0$ ,

$$x_0 = \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \frac{73}{331} \begin{bmatrix} -8 \\ -3 \end{bmatrix} = \begin{bmatrix} 0.2356 \\ 0.3384 \end{bmatrix}.$$

Este resultado completa la 1ra iteración.

Ahora calculamos el nuevo residuo

$$r_1 = r_0 - \alpha_0 A p_0 = \begin{bmatrix} -8 \\ -3 \end{bmatrix} - \frac{73}{331} \begin{bmatrix} 4 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} -8 \\ -3 \end{bmatrix} = \begin{bmatrix} -0.281 \\ 0.7492 \end{bmatrix}.$$

Así continuamos hasta que el número de iteraciones  $k$  sea igual a un  $n$  prefijado o que  $r_k$  sea pequeño.

## Complemento de la Práctica

1. Verifique que su programa dé los resultados mostrados en el ejemplo. Si coinciden con un máximo de 20 iteraciones se llega al resultado.
2. Busque dónde se puede aplicar la paralelización del algoritmo, haga el intento de paralelizarlo.
3. Intente escribir el algoritmo en C++ y con opemp.

## Aplicación

Este método puede ser aplicado a un sistema arbitrario con matriz  $n \times m$  usando una matriz de las ecuaciones normales

$$A^t A$$

y el vector del lado derecho  $A^t b$ , puesto que

$$A^t A$$

es simétrica y semidefinida positiva para cualquier matriz  $A$ .

el resultado es el gradiente conjugado sobre las ecuaciones normales

$$x^t A x = A^t b$$