MÉTODO LANCZOS

Carolina Villarroel - Solana Diaz

Universidad Nacional de Córdoba

Facultad de Matemática, Astronomía, Física y Computación

October 2024

Programación

Definiciones

Antes de empezar a explicar el Algoritmo vamos a recordar unas definiciones importantes para este tema:

Matriz Tridiagonal real: Una matriz $T \in \mathbb{R}^{nxn}$ es una matriz tridiagonal real de tamaño n si sus elementos son sólo distintos de cero en la diagonal principal y en las diagonales adyacentes por debajo y por encima de esta

Introducción

Ya teniendo estas deficiones, podemos hablar de cual es la idea de este método. Queremos encontrar los autovalores extremos de un matriz A cuadrada, simétrica, dispersa y de gran tamaño.

Para ello, el algoritmo contruye una sucesión de matrices tridiagonales T_k y una matriz ortogonal Q a partir de A. Donde los autovalores de la matriz T son muy buenas aproximaciones de A.

Deducción del Algoritmo

Supongamos que $T = Q^T AQ$, donde $Q = [q_1|...|q_n]$ y

$$T_k = \begin{bmatrix} \alpha_1 & \beta_1 & & \dots & 0 \\ \beta_1 & \alpha_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & \beta_{k-1} \\ 0 & \dots & & \beta_{k-1} & \alpha_k \end{bmatrix}$$

Igualando las columnas de $AQ = QT$ se deduce que:

$$Aq_k = \beta_{k-1}q_{k-1} + \alpha_k q_k + \beta_k q_{k+1}, \quad \beta_0 q_0 \equiv 0 \quad con \;\; k = 1, ..., n-1 \tag{1}$$

Tener en cuenta que $T_{ij} = q_i^T Aq_j$, igualando coordenada a coordenada en $T = Q^T AQ$, tenemos las entradas de la diagonal son

$$T_{kk} = q_k^T Aq_k \tag{2}$$

Deducción del Algoritmo

Por otro lado, obtenemos una expresión para calcular sucesivos q_k de la siguiente manera. Definimos el siguiente vector de residuos:

$$r_k = \beta_k q_{k+1} = (A - \alpha_k I)q_k - \beta_{k-1} q_{k-1} \tag{3}$$

y en caso de que $r_k \neq 0$, encontramos nuestro vector Lanczos:

$$q_{k+1} = r_k / \beta_k \tag{4}$$

donde

$$\beta_k = \|r_k\|_2 \tag{5}$$

Si encontramos un $r_k = 0$, la iteración termina, pero no sin llegar a un subespacio invariante. Rara vez encontramos un $\beta_k = 0$, pero igualmente se producen buenas aproximaciones incluso antes de encontrar un β_k pequeño.

Deducción del Algoritmo

Lo que se hace es pedir un número m de máximas iteraciones, de forma que si no se llega en m iteraciones a un subespacio invariante, finalizamos el algoritmo y habremos aproximado m autovalores de la matriz dada.

Primer Algoritmo de Lanczos

Algoritmo 1: Algoritmo de Lanczos

Entrada: Matriz $A \in \mathbb{R}^{n \times n}$, un vector aleatorio $q \in \mathbb{R}^n$, y m el número de iteraciones

Salida : Matriz tridiagonal $T_k \in \mathbb{R}^{k \times k}$, y $Q_k = [q_1 \mid \dots \mid q_n] \in \mathbb{R}^{n \times k}$ tal que $AQ_k = Q_k T_k$

$q_1 = q/\|q\|$, $\beta_0 = 1$, $q_0 = 0$.

for $k = 1$ **to** m **do**

  $\alpha_k = q_k^T A q_k$;

  $r_k = Aq_k - \alpha_k q_k - \beta_{k-1} q_{k-1}$

  $\beta_k = \|r_k\|$;

  if $\beta_k \neq 0$ **then**

    $q_{k+1} = r_k/\beta_k$

  end

end

Link a colab: Código

Teorema

Como hemos vimos, en la práctica solemos pedir un número máximo de iteraciones. Pero, en ocasiones el algoritmo puede llegar a terminar de manera prematura. El siguiente teorema nos muestra cuándo termina el teorema de forma autónoma:

Teorema 2.2.1

TEOREMA: La iteración continúa hasta que $k = m$, donde

$$m = rank(K(A, q_1, n)) \tag{6}$$

Es más, para $k = 1, \ldots, m$ se tiene que $AQ_k = Q_k T_k + r_k e_k^T$, donde

$$T_k = \begin{bmatrix} \alpha_1 & \beta_1 & & \ldots & 0 \\ \beta_1 & \alpha_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & \beta_{k-1} \\ 0 & \ldots & & \beta_{k-1} & \alpha_k \end{bmatrix}$$

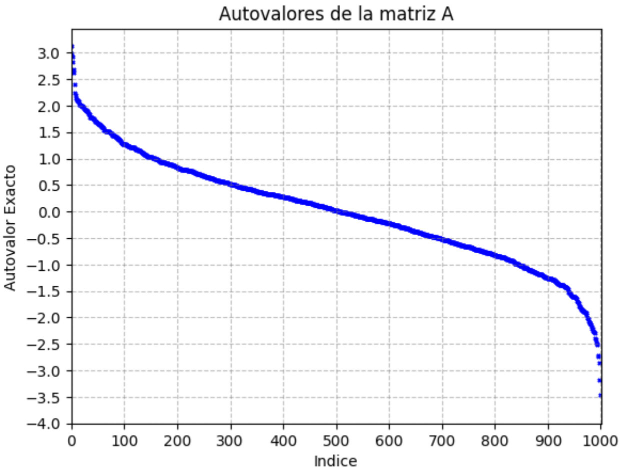
y e_k es el vector canónico de tamaño kx1 con el 1 en su última componente, y $Q = [q_1 \mid \ldots \mid q_k]$ tiene columnas ortonormales que generan $K(A, q_1, k)$.

Ejemplo

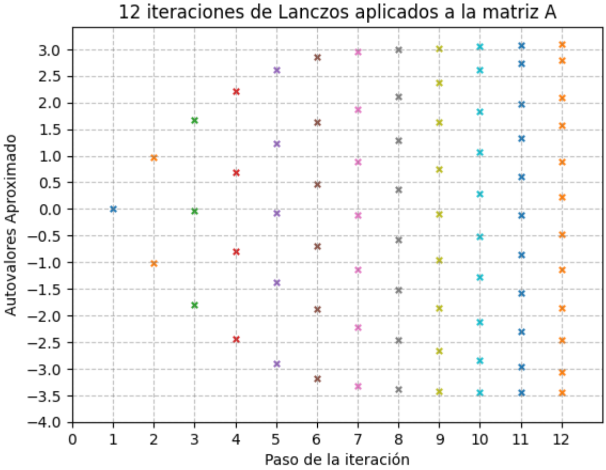
A continuación veremos un ejemplo donde se pretende ilustrar cómo se produce la convergencia de los autovalores cuando crece el número de iteraciones. Vamos a tomar una matriz cuyos autovalores exactos conozcamos, y aplicaremos el algoritmo de Lanczos para un vector inicial q_1.

Consideremos A una matriz diagonal de tamaño 1000 con entradas tomados de una distribución $N(\mu = 0, \sigma^2 = 1)$, que han sido reordenadas posteriormente, de forma que $a_{1,1} \geq a_{2,2} \geq ... \geq a_{1000,1000}$. Tomamos como vector inicial $q_1 = [1, ..., 1]^T$, es decir, el vector columna de 1000 entradas donde todas ellas son 1.

Gráficas


Autovalores de la matriz A
Autovalor Exacto
Indice
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3.0
−3.5
−4.0
0
100
200
300
400
500
600
700
800
900
1000

Gráficas


12 iteraciones de Lanczos aplicados a la matriz A
Autovalores Aproximado
Paso de la iteración
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3.0
−3.5
−4.0
0
1
2
3
4
5
6
7
8
9
10
11
12

Teoría de la Convergencia

Como ya se ha visto, encontrar un $\beta_k = 0$ es un buen evento en el sentido de que se ha calculado un espacio invariante exacto. Rara vez ocurre esto, y sin embargo, los autovalores extremos de las sucesivas matrices T_k son aproximaciones sorprendentemente buenas de los autovalores extremos de A.

Teoría de la Convergencia

Se puede demostrar teóricamente y como vimos en el ejemplo, que la aproximación de los autovalores de A mejora a medida que se incrementa el número máximo de iteraciones. También se puede demostrar que los autovalores extremos (máximo y mínimo) convergen rápidamente, mientras que los autovalores interiores lo hacen de manera más lenta.

Ortogonalidad de las columnas de Q

Aunque el método de Lanczos genera secuencias de vectores ortogonales en el espacio exacto, en la práctica, debido a errores de redondeo, a medida que se realizan más iteraciones se pueden perder ortogonalidad . Esto puede afectar negativamente la calidad de las aproximaciones. Sin embargo, se pueden aplicar técnicas de reortogonalización para resolver este problema. Por eso existe un segundo y tercer algortimo de Lanczos donde mejora la ortogonalización, el cual no veremos.

¿Por que es bueno computacionalmente este algoritmo?

El método es computacionalmente eficiente ya que agarra una matriz densa y de alta dimensión, y produce una matriz tridiagonal de menor tamaño. Lanczos solo requiere operar en ese subespacio de dimensión mucho más pequeña, lo que reduce considerablemente la cantidad de cálculos, ya que es más simple aproximar los autovalores de una matriz tridiagonal que de una matriz de gran tamaño y dispersa.
Además, su enfoque iterativo reduce significativamente la cantidad de cálculos necesarios en comparación con métodos tradicionales de diagonalización.

Conclusión

Cómo conclusión se puede decir que el algoritmo de Lanczos es un método de iteración muy bueno para la aproximación de autovalores y autovectores. Uno de sus principales beneficios del algoritmo es que esta diseñado específicamente para matrices grandes y dispersas, es decir, a diferencias de otros métodos, el algoritmo de Lanczos es más eficiente y adecuado para grandes escalas debido a su baja complejidad computacional por iteración.

Y como lo mencionamos anteriormente el algoritmo tiene una muy buena tasa de convergencia, es decir, que la precisión de las aproximaciones de los autovalores mejora cuando aumenta las iteraciones.