

EL MÉTODO *QR*.

ALAN REYES-FIGUEROA
MÉTODOS NUMÉRICOS II

(AULA 12) 18.AGOSTO.2022

El Metodo QR

El algoritmo QR, (FRANCIS, KUBLANÓVSKAYA), data de la década de los 50s, es una de las joyas del análisis numérico. En su forma más simple, este algoritmo se puede ver como un procedimiento estable para calcular factoraciones QR de las potencias de una matriz A, A^2, A^3, \dots

En su versión más simple, el algoritmo es el siguiente:

Algoritmo: (Método QR puro, sólo calcula autovalores).

Inputs: $A \in \mathbb{R}^{n \times n}$, simétrica.

Outputs: $R \in \mathbb{R}^{n \times n}$, matriz diagonal, con los autovalores de A en la diagonal.

Initialize $A^{(0)} = A$.

for $k = 1, 2, \dots$:

$Q^{(k)}R^{(k)} = A^{(k-1)}$, (factoración QR)

$A^{(k)} = R^{(k)}Q^{(k)}$. (Recombinar en orden reverso)

Tomar una factoración QR, multiplicar los factores Q y R en el orden inverso RQ , y repetir.

El Metodo QR

Bajo supuestos adecuados, este algoritmo simple converge a una forma de Schur para la matriz A (triangular superior si A es arbitraria, y diagonal si A es simétrica), donde en la diagonal de los iterados $A^{(k)}$ se van guardando los autovalores de la matriz A .

Formalmente, si $A \in \mathbb{R}^{n \times n}$ es una matriz a la que queremos calcular los autovalores, durante el k -ésimo paso del método anterior, calculamos la descomposición QR de $A^{(k)}$: $A^{(k)} = Q^{(k)}R^{(k)}$. Luego, $Q^{(k)}$ es ortogonal y $R^{(k)}$ es triangular superior. Al formar $Q^{(k+1)} = R^{(k)}Q^{(k)}$ resulta

$$A^{(k+1)} = R^{(k)}Q^{(k)} = (Q^{(k)})^{-1}Q^{(k)}R^{(k)}Q^{(k)} = (Q^{(k)})^{-1}A^{(k)}Q^{(k)} = (Q^{(k)})^T A^{(k)} Q^{(k)}. \quad (1)$$

Esto muestra que $A^{(k+1)}$ es similar a $A^{(k)}$, y portanto tienen los mismos autovalores. Como esto vale para $k = 0, 1, 2, \dots$, todos los iterados $A^{(k)}$ son similares a $A^{(0)} = A$, y se preservan los autovalores de A .

EN el límite, los iterados $A^{(k)}$ convergen a una matriz diagonal Λ , cuyas entradas son los autovalores de A , esto es

$$\lim_{k \rightarrow \infty} A^{(k)} = \Lambda.$$

El Metodo QR

$$\begin{aligned}\text{De (1)} \quad A &= A^{(0)} = Q^{(0)}A^{(1)}(Q^{(0)})^T = Q^{(0)}Q^{(1)}A^{(2)}(Q^{(1)})^T(Q^{(0)})^T \\ &= \dots \\ &= \underbrace{Q^{(0)}Q^{(1)} \dots Q^{(k)}}_{Q_k} A^{(k+1)} \underbrace{(Q^{(0)}Q^{(1)} \dots Q^{(k)})^T}_{Q_k^T}.\end{aligned}$$

Entonces en el límite, si $Q = \lim_{k \rightarrow \infty} \prod_{j=0}^k Q^{(j)}$, se tiene $A = \lim_{k \rightarrow \infty} Q_k A^{(k+1)} Q_k^T = Q \Lambda Q^T$, y se obtiene una descomposición espectral para A .

El algoritmo QR se puede considerar una versión más sofisticada del método de las potencias. Ambos métodos multiplican repetidamente un vector por la matriz de la que se quieren conocer los valores propios, normalizando después de cada iteración.

Sin embargo, mientras que el método de las potencias solo proporciona el mayor de los valores propios, el método QR usa la descomposición homónima para normalizar y ortogonalizar tras cada iteración. Así cuando $k \rightarrow \infty$, se obtiene la matriz diagonal Λ con todos los autovalores y por tanto Q queda con los autovectores como columnas.

El Metodo QR

Algoritmo: (Método QR puro, autovalores y autovectores).

Inputs: $A \in \mathbb{R}^{n \times n}$, simétrica.

Outputs: $Q \in \mathbb{R}^{n \times n}$, ortogonal con los autovectores de A como columnas, $R \in \mathbb{R}^{n \times n}$ matriz diagonal, con los autovalores de A en la diagonal.

Initialize $A^{(0)} = A$, $Q = I$.

for $k = 1, 2, \dots$:

$Q^{(k)}R^{(k)} = A^{(k-1)}$, (factoración QR)

$A^{(k)} = R^{(k)}Q^{(k)}$, (Recombinar en orden reverso),

$Q = QQ^{(k)}$. (Multiplicar los iterados $Q^{(k)}$).

Return Q , $R = R^{(k)}$.

El Metodo QR

El método QR no es el único de este tipo. Antes de éste, se usaba también el método LR , el cual ejecuta una iteración similar al QR pero basándose en la descomposición LU , en lugar de la QR .

Como es de esperar el método QR proporcional una solución mucho más estable, debido a la ortogonalidad de los componentes.

El Metodo QR

Algoritmo: (Método QR práctico).

Inputs: $A \in \mathbb{R}^{n \times n}$, simétrica.

Outputs: $Q \in \mathbb{R}^{n \times n}$, ortogonal con los autovectores de A como columnas, $R \in \mathbb{R}^{n \times n}$ matriz diagonal, con los autovalores de A en la diagonal.

Initialize $A^{(0)} = A$, $Q = I$.

for $k = 1, 2, \dots$:

$\mu^{(k)} = \text{shift}$

$Q^{(k)}R^{(k)} = A^{(k-1)} - \mu^{(k)}I$, (factoración QR)

$A^{(k)} = R^{(k)}Q^{(k)} + \mu^{(k)}I$, (Recombinar en orden reverso),

$Q = QQ^{(k)}$. (Multiplicar los iterados $Q^{(k)}$).

Return Q , $R = R^{(k)}$.

Forma de Hessenberg

Para efectos de mejorar la estabilidad y eficiencia en el cálculo de los autovalores, se propone un esquema en dos fases:

1. Reducir la matriz A a la forma de Hessenberg mediante una secuencia de transformaciones de semejanza unitaria.
2. Calcular los autovalores sobre la forma de Hessenberg.

Para calcular la factoración de Schur $A = QTQ^T$, nos gustaría aplicar transformaciones unitarias de similitud sobre A , de tal manera que se introduzcan ceros bajo la diagonal. Una mala idea es triangularizar la matriz A mediante aplicaciones de reflexiones de Householder de forma directa:

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} \xrightarrow{Q_1^*} \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix}$$

A Q_1^*A

Forma de Hessemberg

Desafortunadamente, para completar la transformación de similitud, también debemos multiplicar Q_1 a la derecha de A :

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix} \xrightarrow{\cdot Q_1} \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}$$

$Q_1^* A$ $Q_1^* A Q_1$

Esto tiene el efecto de reemplazar cada columna de la matriz por una combinación lineal de todas las columnas. El resultado es que los ceros que estaban previamente introducidos son destruidos.

La estrategia correcta para introducir ceros requeridos consiste en ser menos ambicioso y operar sobre menos entradas de la matriz. En el primer paso, seleccionamos un reflector Householder Q_1 que deja la primera fila sin cambios. Cuando se multiplica a la izquierda de A , forma combinaciones lineales de las filas $2, \dots, n$ para introducir ceros

Forma de Hessemberg

en las filas $3, \dots, n$ de la primera columna. Entonces, al multiplicar Q_1 a la derecha de A , esto deja la primera columna sin cambios.

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} \xrightarrow{Q_1^*} \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix} \xrightarrow{\cdot Q_1} \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}$$

A Q_1^*A $Q_1^*AQ_1$

Esta idea se repite para introducir ceros en columnas posteriores. Por ejemplo, el reflector Househ lder, Q_2 , deja la primera y segunda filas y columnas sin cambios:

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix} \xrightarrow{Q_2^*} \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix} \xrightarrow{\cdot Q_2} \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & \times & \times & \times \end{bmatrix}$$

$Q_1^*AQ_1$ $Q_2^*Q_1^*AQ_1$ $Q_2Q_1^*AQ_1Q_2$

Forma de Hessenberg

Después de repetir este proceso $n - 2$ veces, tenemos un producto en la **forma de HESSENBERG**:

$$\underbrace{Q_{m-2}^* \cdots Q_2^* Q_1^*}_{Q^*} A \underbrace{Q_1 Q_2 \cdots Q_{m-2}}_Q = H.$$

$$H = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix}$$

Forma de Hessemberg

Algoritmo: (Reducción de HOUSEHÖLDER a la forma de HESSEMBERG).

Inputs: $A \in \mathbb{R}^{n \times n}$, simétrica.

Outputs: $Q \in \mathbb{R}^{n \times n}$ matriz ortogonal de reflexiones de Householder, $H \in \mathbb{R}^{n \times n}$, matriz en la forma de Hessemberg, tales que $A = Q^T H Q$.

Initialize $Q = I$, $H = A$.

for $k = 1, 2, \dots, n - 2$:

$$\mathbf{x} = H_{k+1:n,k},$$

$$\mathbf{q}_k = \text{sign}(x_1) \|\mathbf{x}\|_2 \mathbf{e}_1 + \mathbf{x},$$

$$\mathbf{q}_k = \mathbf{q}_k / \|\mathbf{q}_k\|_2,$$

$$H_{k+1:n,k:n} = H_{k+1:n,k:n} - 2\mathbf{q}_k(\mathbf{q}_k^T H_{k+1:n,k:n}),$$

$$H_{1:n,k+1:n} = H_{1:n,k+1:n} - 2(H_{1:n,k+1:n} \mathbf{q}_k) \mathbf{q}_k^T.$$

$$Q = Q Q^{(k)}. \text{ (Multiplicar los iterados } Q^{(k)}).$$

Return Q , H .

Obs! En la práctica la matriz Q no se guarda explícitamente. Sólo se construye H .