Algoritmo QR para cálculo de autovalores

Partimos de una matriz A de tamaño $n \times n$.

1. Factorizamos $A_0 = Q_0 R_0$
2. Calculo la nueva matriz $A_1 = R_0 Q_0$
3. Factorizamos $A_1 = Q_1 R_1$
4. Calculo nueva matriz $A_2 = R_1 Q_1$

$\vdots$

Veamos en qué resulta:

$$A_0 = Q_0 R_0 \longrightarrow Q_0^T A_0 = R_0$$

QR algorithm

$$A_1 = R_0 Q_0 = \underline{Q_0^T A_0} Q_0$$

$$A_1 = Q_1 R_1 \longrightarrow Q_1^T A_1 = R_1$$

$$A_2 = R_1 Q_1 = Q_1^T A_1 Q_1 = Q_1^T Q_0^T A_0 Q_0 Q_1$$

$$\vdots$$

$$A_K = R_{K-1} Q_{K-1} = \dots = Q_{K-1}^T Q_{K-2}^T \dots Q_0^T A_0 Q_0 \dots Q_{K-2} Q_{K-1}$$

Nos fijamos ahora en un par de detalles

① Como el producto de mat. ortogonales resulta en una matriz ortogonal, podemos escribir

$$A_K = Q^T A_0 Q \qquad (Q = Q_0 \dots Q_{K-2} Q_{K-1})$$

② A_K y A_0 son semejantes → tendrán los mismos autovalores

A_1 y A_0 son semejantes (similar matrices) → mismos autovalores

Explicación:

$$A_1 v = \lambda v$$

$$Q_0^T A_0 Q_0 v = \lambda v \rightarrow A_0 Q_0 v = \lambda Q_0 v \rightarrow A_0 u = \lambda u$$

Same eigenvalue, different eigenvector.

Cuando $K \to \infty$ (y se cumplen las condiciones de convergencia que no cubriremos aquí), obtengo una matriz A_K que será triangular superior. Los autovalores de una matriz triangular superior están sobre la diagonal. De modo que solamente tengo que leer los autovalores en la diagonal.

Esto ~~th~~ puede verse como que QR me da la factorización Schur de una matriz

$$A = U T U^T$$

Donde $U \rightarrow$ ortogonal

$T \rightarrow$ triangular superior.

$A_K \to T$ cuando $K \to \infty$

La matriz U la puedo recuperar si en cada iteración utilizo el siguiente pseudo-código:

Pseudo código QR

$A_0 = A$

$U_0 = I$

for $i = 1$, max_iterations

$\quad Q_{i-1}, R_{i-1} = \text{factorización } QR(A_{i-1})$

$\quad A_i = R_{i-1} Q_{i-1}$

$\quad U_i = U_{i-1}\, Q_{i-1}$ $\quad \# Q_0 \ldots Q_{K-2} Q_{K-1}$

Si $i \to \infty$ se cumple que $U_i A_i U_i^T = A_0$

Convergencia

Asumiendo que $|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n|$ (autovalores de diferente magnitud) los elementos de A_K bajo la diagonal tienden a cero como:

$$|a_{ij}^{(K)}| = O\left(\left|\lambda_i / \lambda_j\right|^K\right), \quad i > j$$

Ejemplo

$$A_{20} = \begin{bmatrix} a_{11} & & \\ \left|\frac{\lambda_2}{\lambda_1}\right|^{20} & a_{22} & \\ \left|\frac{\lambda_3}{\lambda_1}\right|^{20} & \left|\frac{\lambda_3}{\lambda_2}\right|^{20} & a_{33} \end{bmatrix}$$

Conclusiones

1. Si los autovalores son parecidos en magnitud, el algoritmo converge despacio.

2. El algoritmo es caro ya que hay que hacer QR en cada paso (coste n^3)