QR iteration for eigenvalues and eigenvectors

Wednesday, November 20, 2024 10:38 AM

Class 37: November 20, 2024

Recall: Last time, we linked the Gram-Schmidt process with a matrix factorization: A = QR, where Q is unitary (has orthonormal columns) and R is upper triangular. After explaining why the Gram-Schmidt algorithm is the poster child of instability & loss of precision, we noted that it either has to be modified (leading to the Modified and Double Gram-Schmidt algorithms), or we need to do something altogether different:

Problem: Find Q unitary (or Q a product of unitary matrices) such that R = Q* A. This idea leads to two popular algorithms: Householder QR (using Householder reflectors) and Givens QR (using Givens rotations).

To get to the Householder QR algorithm, we went through the following concepts:

- **Orthogonal projector:** we say a matrix is an orthogonal projector if P^2 = P and P is symmetric (hermitian). If P is a projector, (I-P) is also a projector such that R(P) = N(I-P) and N(P) = R(I-P).
- **Orthogonal projection onto a subspace V**: given an orthonormal basis {q1,..,qm} of V, the orthogonal projector onto V is sum_{i=1}^m q_i q*_i = Q Q*.
- **Householder reflector**: Given a unit vector w, the Householder reflector Hw = I - 2 ww* . This operator "flips" the component in the direction of w, leaving its orthogonal complement y the same.

We finally derived the algorithm for Householder QR by transforming A into R one column at a time. We got (H_{n-1} H_{n-2} … H_1) A = R, which means that A = (H1 H2 … H_{n-1}) R = QR.

- Compute $\mathbb{R}$ (and maybe apply $\mathbb{Q}^*$)
- Compute $\underline{\mathbb{R} \text{ and } \mathbb{Q}}$.

$$H_k = \begin{bmatrix} I_{k-1} & 0 \\ 0 & \underbrace{I - 2ww^*}_{n-(k-1)} \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

$$H_{k-1} A = \begin{bmatrix} A_{11} & A_{12} \\ H_w \underbrace{A_{21}}_{\text{often } 0} & H_w A_{22} \end{bmatrix} \qquad \underbrace{A_{22} - 2(A_{22}w)w^*}_{O((n-(k-1))^2)}$$

$\underline{\text{Algo 1:}}$ ($\mathbb{R}$ and apply $\mathbb{Q}^*$) $\rightarrow \sim \frac{4}{3} n^3$ (2GE, 1LU)

Algo 1: [illegible]
1 LU)

Algo 2: (Compute $\mathbb{Q}, \mathbb{R}$) $\rightarrow \sim \frac{8}{3} n^3$ (4 GE, 2 LU)

EIGENVALUES / EIGENVECTORS

▶ $\mathbb{A} = \mathbb{U} \mathbb{D} \mathbb{U}^*$, $\mathbb{U}$ unitary
$\mathbb{D}$ diagonal.

DEF: (Eigenpair) Given $\mathbb{A} \in \mathbb{C}^{n \times n}$, then $\lambda \in \mathbb{C}$ is an eigenvalue of $\mathbb{A}$ iff there is $\vec{x} \neq 0$ s.t.

$$\mathbb{A} \vec{x} = \lambda \vec{x} \leftrightarrow \underline{(\mathbb{A} - \lambda I) \vec{x} = \vec{0}}$$

$x \rightarrow$ eigenvector (for λ)
$(\lambda, x) \rightarrow$ eigenpair of $\mathbb{A}$
set of λ's $\rightarrow$ spectrum

$(\mathbb{A} - \lambda I)$ → $\{0\}$ — λ is not an eigen.
$(\mathbb{A} - \lambda I)$ → $\neq \{\vec{0}\}$ — λ is an eigenvalue.

$\updownarrow$

$\mathbb{A} - \lambda I$ is singular — $\underbrace{\det(\mathbb{A} - \lambda I)}$ $= 0$

$P_{\mathbb{A}}(\lambda) = \det(\mathbb{A} - \lambda I)$ char. poly of $\mathbb{A}$ (deg = n)

▷ eigenvalues = roots of $P_{\mathbb{A}}$.

---

eigenvalues = roots of P_A.

otation: $\mathcal{E}_\lambda = N(A - \lambda I)$ "eigenspace".

Multiplicity of λ:
- $p_A(\lambda) = (\lambda - \lambda_0)^k q(\lambda)$, $q(\lambda_0) \neq 0$.
- $\dim(\mathcal{E}_\lambda)$.

SIMILARITY: A, B $n \times n$, I say they are similar if $\exists P$ invertible s.t.

$$B = P^{-1} A P$$

▷ A, B are the same linear op but P is a change of basis.

I want P to be unitary

$$B = P^* A P$$

cols of P are orthonormal.

Invariants under similarity:

- $P_A(\lambda) = P_B(\lambda)$
- Eigenvalues are the same.

$(\lambda, \vec{x}_A)$ eigenpair for A $\longleftrightarrow$ $(\lambda, P^{-1} x_A)$ eigenpair for B.

(λ, x_A) eigenpair for A $\longleftrightarrow$ $(\lambda, \ldots x_A)$ eigenpair for B.

---

random $A \neq U D U^*$ this is not always possible.

(Schur decomposition): for $A \in \mathbb{C}^{n \times n}$, there exists a unitary U s.t.

$$T = U^* A U$$

where T is upper triangular (Schur form) and eigen of $A \rightarrow$ diagonal (T).

Algo (iterative) that produces $\approx (T, U)$

Spectral theorems (2):

Real case: $(\lambda \in \mathbb{R})$ - A be symmetric (hermitian) $(A = A^*)$ $\longleftrightarrow$ there is an orthonormal basis of eigenvectors w/ $\lambda \in \mathbb{R}$, $A = U D U^*$

Complex case $(\lambda \in \mathbb{C})$ - A be normal $> A A^* = A^* A$. $\longleftrightarrow$ there is orth. basis of complex eigenvec/values $\leftrightarrow A = U D U^*$.

Sketch proof (Schur)

(1) $A = A^* \leftrightarrow T = T^* \rightarrow T = D$

(i) $\mathbb{A} = \mathbb{A}^* \leftrightarrow \mathbb{T} = \mathbb{T}^* \rightarrow \mathbb{T} = \mathbb{D}$

(ii) $\mathbb{A}\mathbb{A}^* = \mathbb{A}^*\mathbb{A} \leftrightarrow \mathbb{T}\mathbb{T}^* = \mathbb{T}^*\mathbb{T}$

$\leftrightarrow \underline{\mathbb{T} = \mathbb{D}}$

$\boxed{\text{Statement: There are no direct methods to find eigendecomposition / all eigenvalues of } \mathbb{A} \text{ (for } n \geq 5)}$

If there was, you would have a formula for roots of $p_{\mathbb{A}}$.

QR iteration:

Start $\mathbb{A}_0 = \mathbb{A}$

while (algo has not converged)

$[\mathbb{Q}_k, \mathbb{R}_k] = qr(\mathbb{A}_k)$

$\mathbb{A}_{k+1} = \mathbb{R}_k \mathbb{Q}_k = \underline{\mathbb{Q}_k^* \mathbb{A}_k \mathbb{Q}_k}$

$\therefore \lim_{k \to \infty} \mathbb{A}_k = \mathbb{T}, \quad \lim_{k \to \infty} (\mathbb{Q}_1 \ldots \mathbb{Q}_k) = \mathbb{U}$