Eigenvalues and eigenvectors

Schur decomposition. QR iteration.

Schur decomposition

Let a compex-valued matrix $\mathbf{A} \in \mathbb{C}^{m\times m}$.

$\mathbf{A}$ can be reduced to an upper triangular form via a unitary matrix $\mathbf{Q} \in \mathbb{C}^{m\times m}$, $\qquad (Q^H Q = \widehat{1})$

$$Q^H A Q = \mathbf{T} \quad \text{is upper triangular}$$

$$\mathbf{T} = \begin{bmatrix} t_{11} & \times & \times & \cdots \\ & t_{22} & \times & \cdots \\ & & \ddots & \\ 0 & & & t_{mm} \end{bmatrix}$$

Proof: GvL $\S7.1$

Schur decomposition

- Note that t_{kk} are eigenvalues of $\mathbf{T}$.
- t_{kk} are eigenvalues of A since $\lambda(\mathbf{A}) = \lambda(\mathbf{Q}^H \mathbf{A} \mathbf{Q})$.

<u>NB:</u> Δ form, not diag.
Avoids geometric / algebraic multiplicity of eigenvalues.
Always exists.
Involves complex matrices even if $\mathbf{A}$ is real-valued.

Real Schur decomposition

Let $\mathbf{A} \in \mathbb{R}^{m \times m}$ is real-valued.

Cannot have a triangular real $\mathbf{T}$ matrix because eigenvalues can be complex conjugate.

$\Rightarrow$ ***Real Schur factorization***: block-triangular $\mathbf{T}$.

Real Schur decomposition

Let $\mathbf{A} \in \mathbb{R}^{m\times m}$. $\exists \mathbf{Q} \in \mathbb{R}^{m\times m} \leftarrow$ orthogonal, s.t.

$$\mathbf{Q}^T\mathbf{A}\mathbf{Q} = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1m} \\ 0 & R_{22} & \cdots & R_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_{mm} \end{bmatrix}$$

where R_{kk} is $\begin{cases} \text{either } [\times] \\ \text{or } 2\times 2 \text{ w/ complex conjugate eigenvalues.} \end{cases}$

Proof: GvL §7.4

Schur factorization

Question:

How to transform $\mathbf{A}$ into a Schur form?
Need to use a similarity transform.

Idea 1

$$\mathbf{A} = \mathbf{QR} \qquad \text{however} \quad \lambda(A) \neq \lambda(R)$$

Idea 2

$$\mathbf{A} = \mathbf{QR}$$

$$\text{form } \mathbf{A}_1 = \mathbf{RQ} \equiv \mathbf{Q}^T \mathbf{A} \mathbf{Q}\,.$$

Then $\lambda(\mathbf{A}) = \lambda(\mathbf{A}_1)$.

Schur decomposition

Algorithm:

Start with $\mathbf{A}_0 = \mathbf{A}$. Then for $k = 1, 2, \cdots$

$$\begin{cases} \mathbf{A}_{k-1} = \mathbf{Q}_k \mathbf{R}_k & \qquad \text{QR factorization} \\ \mathbf{A}_k = \mathbf{R}_k \mathbf{Q}_k \end{cases}$$

Note that $\mathbf{Q}_k \in \mathbb{R}^{m \times m}$ is $\perp$ and $\mathbf{R}_k \in \mathbb{R}^{m \times m}$ is Δ

$\mathbf{A}_k$ "converges" to a Δ form i.e. subdiagonal elements of $\mathbf{A}_k \to 0$.

Proof: GvL chap 7.

Convergence of the QR iteration

There is not a real element-wise convergence (upper Δ may vary with k).

$$\text{if} \quad |\lambda_1| > |\lambda_2| > \cdots > |\lambda_m|,$$

$$\text{then} \qquad \underset{l>s\ k\text{-th iteration}}{|a_{ls}^{(k)}|} \leqslant \text{const} \times \left(\frac{\lambda_l}{\lambda_s}\right)^k$$

Computational complexity of the QR iteration

Naïve implementation: Each step is $O(m^3)$ for the QR factorization.

Computational complexity of the QR iteration

Naïve implementation: Each step is $O(m^3)$ for the QR factorization.

Practical implementations:

1. Reduce $\mathbf{A}$ to the ***upper Hessenberg form***.
2. Iterate.

Computational complexity: Hessenberg matrices

First, reduce $\mathbf{A}$ to the ***upper Hessenberg form***: the lower triangular part only has a single subleading diagonal.

$$\mathbf{U}^T\mathbf{A}\mathbf{U} = \mathbf{H} = \begin{bmatrix} \times & \times & \times & \cdots & \times \\ \times & \times & \times & \cdots & \times \\ 0 & \times & \times & \cdots & \times \\ & \ddots & \ddots & \ddots & \\ & & 0 & \times & \times \end{bmatrix}$$

Reduction to the Hessenberg form

Here $\mathbf{U}$ is a sequences of Householder reflections:

$$\begin{bmatrix} \times & \times & \cdots & \times \\ \times & \times & \cdots & \times \\ \times & \times & \cdots & \times \\ \vdots & \vdots & & \\ & & & \\ \times & \times & & \end{bmatrix} \underset{G_1}{\rightarrow} \begin{bmatrix} \times & \times & \cdots & \times \\ \times & \times & \cdots & \times \\ 0 & \times & \cdots & \times \\ \vdots & \vdots & & \\ & & & \\ 0 & \times & & \end{bmatrix} \underset{G_2}{\rightarrow} \begin{bmatrix} \times & \times & \cdots & \times \\ \times & \times & \cdots & \times \\ 0 & \times & \cdots & \times \\ \vdots & 0 & & \\ \vdots & \vdots & & \\ 0 & 0 & & \end{bmatrix} \rightarrow \cdots$$

Reduction requires $O(m^3)$ flops.

QR step with Hessenberg matrices

Then, annihilating the subleading diagonal is $m-1$ Givens rotations:

$$\begin{bmatrix} \times & \times & \times & \\ \times & \times & \times & \\ & \times & \times & \\ & & \times & \ddots \\ & & & \end{bmatrix} \to \begin{bmatrix} \times & \times & \times & \\ 0 & \times & \times & \\ & \times & \times & \\ & & \times & \ddots \\ & & & \end{bmatrix} \to \cdots$$

Then, the k-th step is $O(m^2)$ flops.

Convergence rate: shifts

$$|a_{s,s-1}^{(k)}| \leqslant \text{const}\left(\frac{\lambda_s}{\lambda_{s-1}}\right)^k, \qquad \text{slow if } \lambda_s \approx \lambda_{s-1}$$

Take

$$\widetilde{\mathbf{H}}_k = \mathbf{H}_k - \lambda_* \widehat{\mathbf{1}}$$

$$\lambda(\widetilde{\mathbf{H}})_k = \lambda(\mathbf{H}_k) - \lambda_*$$

$$\Rightarrow \text{conv. rate of} \quad \widetilde{h}_{s,s-1}^{(k)} \quad \text{is} \quad \left(\frac{\lambda_s - \lambda_*}{\lambda_{s-1} - \lambda_*}\right)^k$$

Convergence rate: shifts

1. take λ_* close to λ_s (e.g. $\lambda_* = h^k_{ss}$)
2. do several iterations
3. shift back
$$\widetilde{\mathbf{H}}_k + \lambda_* \widehat{\mathbf{1}}$$