LECTURE 24

Consider a sequence of vectors $\{q_k\}$ in $\mathbb{C}^n$ such that

$$\lim_{k\to\infty} q_k = q_*$$

DEFN (Linear Convergence)

We say $\{q_k\}$ converges to q_* linearly if there exists a constant $c \in [0,1)$ s.t.

$$\lim_{k\to\infty} \frac{\| q_{k+1} - q_* \|}{\| q_k - q_* \|} = c.$$

EXAMPLE

$\{2^{-k}\} = \{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \dots\}$ converges to 0 linearly, i.e.

$$\lim_{k\to\infty} \frac{|2^{-(k+1)} - 0|}{|2^{-k} - 0|} = \frac{1}{2}$$

DEFN (Q-Rate of Convergence)

We say that the q-rate of convergence is p if there exists a constant $c \geq 0$ s.t.

$$\lim_{k\to\infty} \frac{\| q_{k+1} - q_* \|}{\| q_k - q_* \|^p} = c.$$

q-quadratic convergence when $p=2$
q-cubic convergence when $p=3$

EXAMPLE

$$\{2^{-2^k}\} = \{2^{-2}, 2^{-4}, 2^{-8}, 2^{-16}, \dots\}$$

converges to 0 q-quadratically, i.e.

$$\lim_{k\to\infty} \frac{|2^{-2^{k+1}} - 0|}{|2^{-2^k} - 0|^2} = \lim_{k\to\infty} \frac{|2^{-2^{k+1}}|}{\underbrace{|2^{-2^k} \cdot 2^{-2^k}|}_{2^{-2\cdot 2^k} = 2^{-2^{k+1}}}} = 1$$

RATE OF CONVERGENCE FOR POWER ITERATION

Recall the estimate generated by power iteration of the form

$$q_k = \frac{\lambda_1^k}{|\lambda_1^k|} \frac{c_1 v_1 + c_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k v_2 + \dots + c_n \left(\frac{\lambda_n}{\lambda_1}\right)^k v_n}{\left\| c_1 v_1 + c_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k v_2 + \dots + c_n \left(\frac{\lambda_n}{\lambda_1}\right)^k v_n \right\|}$$

As $k \to \infty$

$$\left\| q_k - \frac{\lambda_1^k}{|\lambda_1^k|} \frac{c_1 v_1}{\|c_1 v_1\|} \right\| \longrightarrow \frac{|c_2|}{\|c_1 v_1\|} \left| \frac{\lambda_2}{\lambda_1} \right|^k \|v_2\|$$

Consequently

$$\lim_{k\to\infty} \frac{\left\| q_{k+1} - \tilde{c}_{k+1} \frac{v_1}{\|v_1\|} \right\|}{\left\| q_k - \tilde{c}_k \frac{v_1}{\|v_1\|} \right\|} = \frac{|\lambda_2|}{|\lambda_1|}$$

where $\{\tilde{c}_k\}$ denotes a sequence of complex signs (i.e. $\tilde{c}_k = e^{i\theta_k}$ for some θ_k.)

REMARKS

* Power iteration converges linearly.
* Convergence is especially slow when $|\lambda_1| \approx |\lambda_2|$

INVERSE ITERATION

Suppose σ is a good estimate for an eigenvalue λ_j of A.

Let's say the eigenvalue λ_k second closest to σ is significantly further away, i.e $|\lambda_j - \sigma| \ll |\lambda_k - \sigma|$.

BASIC OBSERVATION

Suppose (λ, v) is an eigenpair of A. Then

$$Av = \lambda v \iff (A - \sigma I)v = (\lambda - \sigma)v$$

$$\iff (A - \sigma I)^{-1} v = (\lambda - \sigma)^{-1} v,$$

that is $((\lambda - \sigma)^{-1}, v)$ is an eigenpair of $(A - \sigma I)^{-1}$.

Consequently the largest eigenvalue (in modulus) of $(A - \sigma I)^{-1}$ is significantly larger than second largest, i.e.

$$\underbrace{\frac{1}{|\lambda_j - \sigma|}}_{\text{largest eigenvalue of } (A-\sigma I)^{-1}} \gg \underbrace{\frac{1}{|\lambda_k - \sigma|}}_{\text{second largest}}$$

Inverse iteration is power iteration applied to $(A-\sigma I)^{-1}$

ALGORITHM (Inverse Iteration)

* Given $A \in \mathbb{C}^{n\times n}$, $q_0 \in \mathbb{C}^n$ s.t. $\|q_0\|=1$ and $\sigma \in \mathbb{C}$
* Produce an estimate for the eigenpair (λ_j, v_j) where λ_j is the eigenvalue of A closest to σ.

Compute an LU factorization of $(A-\sigma I)$

for $k=1,\dots,m$

  Solve $L\hat{x} = q_{k-1}$ by forward ~~back~~ substitution

  Solve $Ux = \hat{x}$ by ~~forward~~ back substitution

  $q_k = x/\|x\|$

end

$v \Leftarrow q_m$, $\lambda = q_m^* A q_m$

REMARKS

* At the kth iteration the linear system
$$(A-\sigma I)x = q_{k-1}$$
must be solved.
* This can be done efficiently if an LU factorization for $(A-\sigma I)$ is computed initially.

* Inverse iteration is commonly to compute eigenvectors given eigenvalues.

Rate of convergence

Still linear, but with a small constant.

$$\lim_{k\to\infty} \frac{\left\| q_{k+1} - \tilde{c}_{k+1} \frac{v_j}{\|v_j\|} \right\|}{\left\| q_k - \tilde{c}_k \frac{v_j}{\|v_j\|} \right\|} = \frac{1/|\lambda_k - \sigma|}{1/|\lambda_j - \sigma|} = \frac{|\lambda_j - \sigma|}{|\lambda_k - \sigma|}$$

RAYLEIGH ITERATION

Rayleigh quotient

$$r(q) = \frac{q^* A q}{q^* q}$$

gives a very good estimate for an eigenvalue of A if q is close to an eigenvector of A.

In particular

$$r(v_j) = \frac{v_j^* A v_j}{v_j^* v_j} = \frac{v_j^*(\lambda_j v_j)}{v_j^* v_j} = \lambda_j$$

where (λ_j, v_j) is an eigenpair.

__THM__ (Accuracy of Rayleigh Quotient)
Let (λ_j, v_j) be an eigenpair of $A \in \mathbb{C}^{n\times n}$ and $q \in \mathbb{C}^n$ s.t. $\|v_j\| = \|q\| = 1$. Then

$$|r(q) - \lambda_j| \leq 2\|A\|\,\|q - v_j\|$$

__PROOF__

$$\begin{aligned}
r(q) - \lambda_j &= r(q) - r(v_j) \\
&= q^* A q - v_j^* A v_j \\
&= (q^* A q - q^* A v_j) + (q^* A v_j - v_j^* A v_j) \\
&= q^* A (q - v_j) + (q - v_j)^* A v_j
\end{aligned}$$

$\Longrightarrow$ (By triangular inequality and submultiplicative properties)

$$\begin{aligned}
|r(q) - \lambda_j| &\leq |q^* A (q - v_j)| + |(q - v_j)^* A v_j| \\
&\leq \underbrace{\|q^*\|}_{1}\,\|A\|\,\|q - v_j\| + \underbrace{\|(q - v_j)^*\|}_{\|q-v_j\|}\,\|A\|\,\underbrace{\|v_j\|}_{1} \\
&= 2\|A\|\,\|q - v_j\|
\end{aligned}$$

$\square$

Rayleigh iteration is the inverse iteration but with shifts chosen as Rayleigh quotients.

ALGORITHM (Rayleigh Iteration)

* Given $A \in \mathbb{C}^{n \times n}$ and $q_0 \in \mathbb{C}^n$ s.t. $\|q_0\| = 1$.
* Produce an estimate (λ, v) for an eigenpair of A.

for $k = 1, \ldots, m$

$\quad \sigma_{k-1} = q_{k-1}^* A q_{k-1}$

$\quad$ Solve $(A - \sigma_{k-1} I)x = q_{k-1}$ for x

$\quad q_k = x / \|x\|$

end

$v = q_m, \quad \lambda = q_m^* A q_m$

Rate of Convergence

Suppose $\operatorname{span}\{q_k\} \longrightarrow \operatorname{span}\{v_j\}$ as $k \to \infty$.

Then

$$\lim_{k \to \infty} \frac{\left\| q_{k+1} - \tilde{c}_{k+1} \frac{v_j}{\|v_j\|} \right\|}{\left\| q_k - \tilde{c}_k \frac{v_j}{\|v_j\|} \right\|^2} = c,$$

that is the q-rate of convergence is quadratic.

JUSTIFICATION

$$\frac{\left\| q_{k+1} - \tilde{c}_{k+1} \frac{v_j}{\|v_j\|} \right\|}{\left\| q_k - \tilde{c}_k \frac{v_j}{\|v_j\|} \right\|} = \frac{|r(q_k) - \lambda_j|}{|r(q_k) - \underbrace{\lambda_\ell}_{\text{second closest to } r(q_k)}|} = O\left(\left\| q_k - \tilde{c}_k \frac{v_j}{\|v_j\|} \right\| \right)$$

QR ALGORITHM

Given a Hessenberg matrix $A \in \mathbb{C}^{n\times n}$.
Generates a sequence $\{A_k\}$ satisfying

(i) $A_0 = A$

$\boxed{\text{FOR } k \geq 0}$ (ii) $\underbrace{A_k = Q_{k+1} R_{k+1}}_{\text{QR Factorization of } A_k}$ and $A_{k+1} = R_{k+1} Q_{k+1}$

ALGORITHM (QR Algorithm)

* Given $A \in \mathbb{C}^{n\times n}$ in Hessenberg form.
* Produce a sequence $\{A_k\}$ such that typically $\underbrace{\lim_{k\to\infty} A_k \text{ is upper triangular.}}_{\text{This will be justified later}}$

$A_0 = A$
for $k = 0, 1, \ldots$
    Compute a QR factorization $A_k = Q_{k+1} R_{k+1}$
    $A_{k+1} = R_{k+1} Q_{k+1}$
end

EXAMPLE

$A = \begin{bmatrix} 3 & 4 \\ 2 & 1 \end{bmatrix}$ has the eigenvalues $\lambda_1 = 5,\ \lambda_2 = -1$

Apply the QR algorithm to A

$$A_0 = A = \underbrace{\begin{bmatrix} -0.83 & -0.55 \\ -0.55 & 0.83 \end{bmatrix}}_{Q_1} \underbrace{\begin{bmatrix} -3.61 & -3.88 \\ 0 & -1.39 \end{bmatrix}}_{R_1}$$

$$A_1 = R_1 Q_1 = \begin{bmatrix} 5.15 & -1.23 \\ 0.77 & -1.15 \end{bmatrix}$$

$$A_2 = R_2 Q_2 = \begin{bmatrix} 4.95 & 2.14 \\ 0.14 & -0.95 \end{bmatrix}$$

$$A_3 = R_3 Q_3 = \begin{bmatrix} 5.01 & -1.97 \\ 0.03 & -1.01 \end{bmatrix}$$

REMARKS

(i) A_k and A_{k+1} are unitarily similar

$$A_k = Q_{k+1} R_{k+1} \text{ and } A_{k+1} = R_{k+1} Q_{k+1}$$

$$\Longrightarrow$$

$$R_{k+1} = Q^*_{k+1} A_k \text{ and } A_{k+1} = R_{k+1} Q_{k+1}$$

$$\Longrightarrow$$

$$A_{k+1} = Q^*_{k+1} A_k Q_{k+1}$$

(ii) QR factorization can be computed so that

$$A_k \text{ is Hessenberg} \Longrightarrow A_{k+1} \text{ is Hessenberg}$$

implying that the sequence $\{A_k\}$ is Hesenberg

THM (Invariance of Hessenberg Form)
Let $A_k \in \mathbb{C}^{n\times n}$ be non-singular and Hessenberg. Then the matrix $A_{k+1} \in \mathbb{C}^{n\times n}$ generated by the QR algorithm is also Hessenberg.

PROOF
First note that product of a Hessenberg matrix with an upper triangular matrix from left or right is Hessenberg.

Now

$$A_k = Q_{k+1} R_{k+1} \Longrightarrow Q_{k+1} = \underbrace{A_k}_{\text{Hessenberg}} \underbrace{R_{k+1}^{-1}}_{\text{upper triangular}}$$

$$\Longrightarrow Q_{k+1} \text{ is Hessenberg}$$

Furthermore

$$A_{k+1} = \underbrace{R_{k+1}}_{\text{upper triangular}} \underbrace{Q_{k+1}}_{\text{Hessenberg}} \Longrightarrow A_{k+1} \text{ is Hessenberg} \quad \square$$

(iii) The QR factorization

$$A_k = Q_{k+1} R_{k+1}$$

can be computed at a cost of $O(n^2)$ (by HH reflectors, since A_k is Hessenberg) and the multiplication

$$A_{k+1} = R_{k+1} Q_{k+1}$$

can be performed at a cost of $O(n^2)$. (since Q_k is formed by HH reflectors)

$\therefore$ Therefore the cost of each QR iteration

$$A_k \longrightarrow A_{k+1} = Q_{k+1}^* A_k Q_{k+1}$$

is $O(n^2)$.