AMS526: Numerical Analysis I (Numerical Linear Algebra for Computational and Data Sciences)

Lecture 16: QR Algorithm without Shifts (NLA§28); QR Algorithm with Shifts (NLA§29)

Xiangmin Jiao

SUNY Stony Brook

Outline

1 QR Algorithm without Shifts

2 QR Algorithm with Shifts

QR Algorithm

- Most basic version of QR algorithm is remarkably simple:

> Algorithm: "Pure" QR Algorithm
>
> $A^{(0)} = A$
>
> **for** $k = 1, 2, \ldots$
>
> $\qquad Q^{(k)} R^{(k)} = A^{(k-1)}$
>
> $\qquad A^{(k)} = R^{(k)} Q^{(k)}$

- With some suitable assumptions, $A^{(k)}$ converge to Schur form of A (diagonal if A is symmetric)
- Similarity transformation of A:

$$A^{(k)} = R^{(k)} Q^{(k)} = \left(Q^{(k)} \right)^T A^{(k-1)} Q^{(k)}$$

- But why it works?

Unnormalized Simultaneous Iteration

- To understand QR algorithm, first consider simple algorithm
- Simultaneous iteration is power iteration applied to several vectors
- Start with linearly independent $v_1^{(0)}, \cdots, v_n^{(0)}$
- We know from power iteration that $A^k v_1^{(0)}$ converge to q_1
- With some assumptions, the space $\left\langle A^k v_1^{(0)}, \ldots, A^k v_n^{(0)} \right\rangle$ should converge to $\langle q_1, \cdots, q_n \rangle$
- Notation: Define initial matrix $V^{(0)}$ and matrix $V^{(k)}$ at step k:

$$V^{(0)} = \left[v_1^{(0)} \middle| \cdots \middle| v_n^{(0)} \right], \; V^{(k)} = A^k V^{(0)} = \left[v_1^{(k)} \middle| \cdots \middle| v_n^{(k)} \right]$$

Unnormalized Simultaneous Iteration

- Define orthogonal basis for column space of $V^{(k)}$ by reduced QR factorization $\hat{Q}^{(k)}\hat{R}^{(k)} = V^{(k)}$
- We assume that
  1. leading $k+1$ eigenvalues are distinct, and
  2. all leading principal submatrices of $\hat{Q}^T V^{(0)}$ are nonsingular where $\hat{Q} = [q_1|\cdots|q_n]$
- We then have columns of $\hat{Q}^{(k)}$ converge to eigenvectors of A:

$$\|q_j^{(k)} - (\pm q_j)\| = O(c^k),$$

  where $c = \max_{1\leq k\leq n} |\lambda_{k+1}|/|\lambda_k|$
- Proof idea: Show that subspace of any leading j columns of $V^{(k)} = A^k V^{(0)}$ converges to subspace of first j eigenvectors of A, so does the subspace of any leading j columns of $\hat{Q}^{(k)}$.

Simultaneous Iteration

- Matrices $V^{(k)} = A^k V^{(0)}$ are highly ill-conditioned
- Orthonormalize at each step rather than at the end

> Algorithm: Simultaneous Iteration
>
> Pick $\hat{Q}^{(0)} \in \mathbb{R}^{n \times n}$
>
> **for** $k = 1, 2, \ldots$
>
> $\qquad Z = A\hat{Q}^{(k-1)}$
>
> $\qquad \hat{Q}^{(k)}\hat{R}^{(k)} = Z$

- Column spaces of $\hat{Q}^{(k)}$ and $Z^{(k)}$ are both equal to column space of $A^k\hat{Q}^{(0)}$, therefore same convergence as before

Simultaneous Iteration $\Longleftrightarrow$ QR Algorithm

Algorithm: Simultaneous Iteration

Pick $\hat{Q}^{(0)} \in \mathbb{R}^{m\times n}$
for $k = 1, 2, \ldots$
$$Z = A\hat{Q}^{(k-1)}$$
$$\hat{Q}^{(k)}\hat{R}^{(k)} = Z$$

Algorithm: "Pure" QR Algorithm

$A^{(0)} = A$
for $k = 1, 2, \ldots$
$$Q^{(k)}R^{(k)} = A^{(k-1)}$$
$$A^{(k)} = R^{(k)}Q^{(k)}$$

- QR algorithm is equivalent to simultaneous iteration with $\hat{Q}^{(0)} = I$
- Replace $\hat{R}^{(k)}$ by $R^{(k)}$ and $\hat{Q}^{(k)}$ by $\underline{Q}^{(k)}$, and introduce new statement $A^{(k)} = \left(\underline{Q}^{(k)}\right)^T A\underline{Q}^{(k)}$ in simultaneous iteration

Simultaneous iteration

$$\underline{Q}^{(0)} = I$$
$$Z = A\underline{Q}^{(k-1)}$$
$$\underline{Q}^{(k)}R^{(k)} = Z$$
$$A^{(k)} = \left(\underline{Q}^{(k)}\right)^T A\underline{Q}^{(k)}$$

QR algorithm

$$A^{(0)} = A$$
$$Q^{(k)}R^{(k)} = A^{(k-1)}$$
$$A^{(k)} = R^{(k)}Q^{(k)}$$
$$\underline{Q}^{(k)} = Q^{(1)}Q^{(2)}\ldots Q^{(k)}$$

Simultaneous Iteration $\Longleftrightarrow$ QR Algorithm

- $\underline{Q}^{(k)} = Q^{(1)} Q^{(2)} \cdots Q^{(k)}$. Let $\underline{R}^{(k)} = R^{(k)} R^{(k-1)} \cdots R^{(1)}$
- Both schemes generate QR factorization $A^k = \underline{Q}^{(k)} \underline{R}^{(k)}$ and projection $A^{(k)} = \left(\underline{Q}^{(k)}\right)^T A \underline{Q}^{(k)}$

Proof by induction. For $k = 0$ it is trivial for both algorithms.
For $k \geq 1$ with simultaneous iteration, $A^{(k)}$ is given by definition, and

$$A^k = A\underline{Q}^{(k-1)} \underline{R}^{(k-1)} = \underline{Q}^{(k)} R^{(k)} \underline{R}^{(k-1)} = \underline{Q}^{(k)} \underline{R}^{(k)}$$

For $k \geq 1$ with QR algorithm,

$$A^k = A\underline{Q}^{(k-1)} \underline{R}^{(k-1)} = \underline{Q}^{(k-1)} A^{(k-1)} \underline{R}^{(k-1)} = \underline{Q}^{(k)} \underline{R}^{(k)}$$

and

$$A^{(k)} = \left(Q^{(k)}\right)^T A^{(k-1)} Q^{(k)} = \left(\underline{Q}^{(k)}\right)^T A \underline{Q}^{(k)}$$

Convergence of QR Algorithm

- Since $\underline{Q}^{(k)} = \hat{Q}^{(k)}$ in simultaneous iteration, column vectors of $\underline{Q}^{(k)}$ converge linearly to eigenvectors if A has distinct eigenvalues
- From $A^{(k)} = \left(\underline{Q}^{(k)}\right)^T A\underline{Q}^{(k)}$, diagonal entries of $A^{(k)}$ are Rayleigh quotients of column vectors of $\underline{Q}^{(k)}$, so they converge linearly to eigenvalues of A
- Off-diagonal entries of $A^{(k)}$ converge to zeros, as they are generalized Rayleigh quotients involving approximations of distinct eigenvectors
- Overall, $A = \underline{Q}^{(k)} A^{(k)} \left(\underline{Q}^{(k)}\right)^T$. For a symmetric matrix, it converges to eigenvalue decomposition of A
- Convergence rate is only linear: columns of $\hat{Q}^{(k)}$ converge to eigenvectors of A $\|q_j^{(k)} - (\pm q_j)\| = O(c^k)$, where $c = \max_{1\leq k\leq n} |\lambda_{k+1}|/|\lambda_k|$

Outline

1 QR Algorithm without Shifts

2 QR Algorithm with Shifts

Simultaneous Inverse Iteration $\Longleftrightarrow$ QR Algorithm

- Similar to inverse iteration, QR algorithm can be sped-up by introducing shifts at each step
- Assume A is real and symmetric. QR algorithm is equivalent to *simultaneous inverse iteration*, applied to "flipped" identity matrix P

$$P = \begin{bmatrix} & & & 1 \\ & & 1 & \\ & \vdots & & \\ 1 & & & \end{bmatrix}$$

Simultaneous inverse iteration	"Pure" QR Algorithm
$\hat{Q}^{(0)} = P$	$A^{(0)} = A$
for $k = 1, 2, \dots$	**for** $k = 1, 2, \dots$
$Z = A^{-1}\hat{Q}^{(k-1)}$	$Q^{(k)}R^{(k)} = A^{(k-1)}$
$\hat{Q}^{(k)}\hat{R}^{(k)} = Z$	$A^{(k)} = R^{(k)}Q^{(k)}$

Simultaneous Inverse Iteration $\Longleftrightarrow$ QR Algorithm

- Let $\underline{Q}^{(k)} = Q^{(1)}Q^{(2)}\cdots Q^{(k)}$ and $\underline{R}^{(k)} = R^{(k)}R^{(k-1)}\cdots R^{(1)}$. Then $A^k = \underline{Q}^{(k)}\underline{R}^{(k)}$
- Inverting A^k, we have $A^{-k} = \left(\underline{R}^{(k)}\right)^{-1}\left(\underline{Q}^{(k)}\right)^T$
- Because A^{-k} is symmetric, $A^{-k} = \underline{Q}^{(k)}\left(\underline{R}^{(k)}\right)^{-T}$
- Use "flipped" permutation matrix P and write that last expression as

$$A^{-k}P = \left[\underline{Q}^{(k)}P\right]\left[P\left(\underline{R}^{(k)}\right)^{-T}P\right],$$

  which is QR factorization of $A^{-k}P$
- Therefore, simultaneous inverse iteration applied to $\hat{Q}^{(0)} = P$ is "equivalent" to QR algorithm, in that it produces

$$\hat{Q}^{(k)} = \underline{Q}^{(k)}P \text{ and } \hat{R}^{(k)}\hat{R}^{(k-1)}\cdots\hat{R}^{(1)} = P\left(\underline{R}^{(k)}\right)^{-T}P$$

- Question: How to obtain $A^{(k)}$ in simultaneous inverse iteration?

Simultaneous Inverse Iteration $\Longleftrightarrow$ QR Algorithm

- Let $\underline{Q}^{(k)} = Q^{(1)}Q^{(2)}\cdots Q^{(k)}$ and $\underline{R}^{(k)} = R^{(k)}R^{(k-1)}\cdots R^{(1)}$. Then $A^k = \underline{Q}^{(k)}\underline{R}^{(k)}$
- Inverting A^k, we have $A^{-k} = \left(\underline{R}^{(k)}\right)^{-1}\left(\underline{Q}^{(k)}\right)^T$
- Because A^{-k} is symmetric, $A^{-k} = \underline{Q}^{(k)}\left(\underline{R}^{(k)}\right)^{-T}$
- Use "flipped" permutation matrix P and write that last expression as

$$A^{-k}P = \left[\underline{Q}^{(k)}P\right]\left[P\left(\underline{R}^{(k)}\right)^{-T}P\right],$$

  which is QR factorization of $A^{-k}P$
- Therefore, simultaneous inverse iteration applied to $\hat{Q}^{(0)} = P$ is "equivalent" to QR algorithm, in that it produces

$$\hat{Q}^{(k)} = \underline{Q}^{(k)}P \text{ and } \hat{R}^{(k)}\hat{R}^{(k-1)}\cdots\hat{R}^{(1)} = P\left(\underline{R}^{(k)}\right)^{-T}P$$

- Question: How to obtain $A^{(k)}$ in simultaneous inverse iteration?
  Answer: $A^{(k)} = \left(\underline{Q}^{(k)}\right)^T A\underline{Q}^{(k)} = P\left(\hat{Q}^{(k)}\right)^T A\hat{Q}^{(k)}P$

QR Algorithm with Shifts

- Similar to inverse iteration, we can introduce shifts $\mu^{(k)}$ to accelerate convergence

> Algorithm: QR Algorithm with Shifts
>
> $A^{(0)} = A$
>
> **for** $k = 1, 2, \ldots$
>
> Pick a shift $\mu^{(k)}$
>
> $Q^{(k)} R^{(k)} = A^{(k-1)} - \mu^{(k)} I$
>
> $A^{(k)} = R^{(k)} Q^{(k)} + \mu^{(k)} I$

Properties of QR Algorithm with Shift

- From $Q^{(k)}R^{(k)} = A^{(k-1)} - \mu^{(k)}I$ and $A^{(k)} = R^{(k)}Q^{(k)} + \mu^{(k)}I$, we have
$$A^{(k)} = \left(Q^{(k)}\right)^T A^{(k-1)} Q^{(k)}$$
- Then by induction, $A^{(k)} = \left(\underline{Q}^{(k)}\right)^T A\underline{Q}^{(k)}$
- However, instead of $A^k = \underline{Q}^{(k)}\underline{R}^{(k)}$, we now have
$$\left(A - \mu^{(k)}I\right)\left(A - \mu^{(k-1)}I\right)\cdots\left(A - \mu^{(1)}I\right) = \underline{Q}^{(k)}\underline{R}^{(k)},$$
which can be shown by induction
- In other words, $\underline{Q}^{(k)}$ is orthogonalization of $\prod_{j=k}^{1}(A - \mu^{(k)}I)$
- If $\mu^{(k)}$ are good estimates of eigenvalues, then last column of $\underline{Q}^{(k)}$ converges to corresponding eigenvector

Choosing $\mu^{(k)}$: Rayleigh Quotient Shift

- Natural choice of $\mu^{(k)}$ is Rayleigh quotient for last column of $\underline{Q}^{(k)}$

$$\mu^{(k)} = r(q_n^{(k)}) = \left(q_n^{(k)}\right)^T A q_n^{(k)}$$

- As in Rayleigh quotient iteration, last column $q_n^{(k)}$ converges cubically
- This Rayleigh quotient appears as (n, n) entry of $A^{(k)}$ since $A^{(k)} = \left(\underline{Q}^{(k)}\right)^T A\underline{Q}^{(k)}$
- Rayleigh quotient shift corresponds to setting $\mu^{(k)} = A_{nn}^{(k)}$

Choosing $\mu^{(k)}$: Wilkinson Shift

- QR algorithm with Rayleigh quotient shift might fail sometimes, e.g., $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, for which $A^{(k)} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ and μ is always 0
- Wilkinson breaks symmetry by considering lower-rightmost 2×2 submatrix of $A^{(k)}$: $B = \begin{bmatrix} a_{n-1} & b_{n-1} \\ b_{n-1} & a_n \end{bmatrix}$
- Choose eigenvalue of B closer to a_n, with arbitrary tie-breaking:

$$\mu = a_n - \text{sign}(\delta) b_{n-1}^2 \Big/ \left(|\delta| + \sqrt{\delta^2 + b_{n-1}^2} \right)$$

  where $\delta = (a_{n-1} - a_n)/2$; if $\delta = 0$, set $\mathrm{sign}(\delta)$ to 1 (or -1)
- QR algorithm always converges with this shift; quadratically in worst case, and cubically in general

"Practical" QR Algorithm

- Practical QR algorithm involves two additional components:
  - tridiagonalization of A at the beginning. The tridiagonal structure is preserved by $A^{(k)}$ (Exercise 28.2)
  - *deflation* of A into submatrices when $A^{(k)}$ is separable

Algorithm: "Practical" QR Algorithm

$\left(Q^{(0)}\right)^T A^{(0)} Q^{(0)} = A$ {tridiagnolization of A}

for $k = 1, 2, \ldots$

Pick a shift $\mu^{(k)}$

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

$A^{(k)} = R^{(k)} Q^{(k)} + \mu^{(k)} I$

If any off-diagonal element $a_{j,j+1}^{(k)}$ is sufficiently close to zero

set $a_{j,j+1} = a_{j+1,j} = 0$ to obtain $\begin{bmatrix} A_1 & \\ & A_2 \end{bmatrix} = A^{(k)}$ and

apply QR algorithm to A_1 and A_2

Stability and Accuracy

Theorem

QR algorithm is backward stable

$$\tilde{Q}\tilde{\Lambda}\tilde{Q} = A + \delta A, \quad \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{machine})$$

where $\tilde{\Lambda}$ is computed Λ and $\tilde{Q}$ is exactly orthogonal matrix

- Its combination with Hessenberg reduction is also backward stable
- Furthermore, eigenvalues are always well conditioned for **normal** matrices: it can be show that $|\tilde{\lambda}_j - \lambda_j| \leq \|\delta A\|_2$, and therefore,

$$\frac{|\tilde{\lambda}_j - \lambda_j|}{\|A\|} = O(\epsilon_{\text{machine}})$$

  where $\tilde{\lambda}_j$ are the computed eigenvalues
- However, sensitivity of eigenvectors depends on distances between adjacent eigenvalues, so error in eigenvectors may be arbitrarily large