Lecture 19
QR Algorithm with Shifts

Songting Luo

Department of Mathematics
Iowa State University

MATH 562 Numerical Analysis II

Outline

Outline

Simultaneous Inverse Iteration ⇔QR Algorithm

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

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

Simultaneous inverse iteration

$\hat{\mathbf{Q}}^{(0)} = \mathbf{P}$

for $k = 1, 2, \ldots$

$\quad \mathbf{Z} = \mathbf{A}^{-1}\hat{\mathbf{Q}}^{(k-1)}$

$\quad \hat{\mathbf{Q}}^{(k)}\hat{\mathbf{R}}^{(k)} = \mathbf{Z}$

"Pure" QR Algorithm

$\mathbf{A}^{(0)} = \mathbf{A}$

for $k = 1, 2, \ldots$

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

$\quad \mathbf{A}^{(k)} = \mathbf{R}^{(k)}\mathbf{Q}^{(k)}$

Simultaneous Inverse Iteration $\Leftrightarrow$QR Algorithm

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

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

  which is QR factorization of $\mathbf{A}^{-k}\mathbf{P}$
- Therefore, simultaneous inverse iteration applied to $\hat{\mathbf{Q}}^{(0)} = \mathbf{P}$ is "equivalent" to QR algorithm, in that it produces $\hat{\mathbf{Q}}^{(k)} = \underline{\mathbf{Q}}^{(k)}\mathbf{P}$ and $\hat{\mathbf{R}}^{(k)}\hat{\mathbf{R}}^{(k-1)}\cdots\hat{\mathbf{R}}^{(1)} = \mathbf{P}(\underline{\mathbf{R}}^{(k)})^{-T}\mathbf{P}$
- Question: How to obtain $\mathbf{A}^{(k)}$ in simultaneous inverse iteration?

Simultaneous Inverse Iteration $\Leftrightarrow$QR Algorithm

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

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

  which is QR factorization of $\mathbf{A}^{-k}\mathbf{P}$
- Therefore, simultaneous inverse iteration applied to $\hat{\mathbf{Q}}^{(0)} = \mathbf{P}$ is "equivalent" to QR algorithm, in that it produces $\hat{\mathbf{Q}}^{(k)} = \underline{\mathbf{Q}}^{(k)}\mathbf{P}$ and $\hat{\mathbf{R}}^{(k)}\hat{\mathbf{R}}^{(k-1)} \cdots \hat{\mathbf{R}}^{(1)} = \mathbf{P}(\underline{\mathbf{R}}^{(k)})^{-T}\mathbf{P}$
- Question: How to obtain $\mathbf{A}^{(k)}$ in simultaneous inverse iteration?
- Answer: $\mathbf{A}^{(k)} = (\underline{\mathbf{Q}}^{(k)})^T\mathbf{A}\underline{\mathbf{Q}}^{(k)} = \mathbf{P}(\hat{\mathbf{Q}}^{(k)})^T\mathbf{A}\hat{\mathbf{Q}}^{(k)}\mathbf{P}$

QR Algorithm with Shifts

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

Algorithm: QR Algorithm with Shifts

$\mathbf{A}^{(0)} = \mathbf{A}$
for $k = 1, 2, \ldots$
  Pick a shift $\mu^{(k)}$
  $\mathbf{Q}^{(k)}\mathbf{R}^{(k)} = \mathbf{A}^{(k-1)} - \mu^{(k)}\mathbf{I}$
  $\mathbf{A}^{(k)} = \mathbf{R}^{(k)}\mathbf{Q}^{(k)} + \mu^{(k)}\mathbf{I}$

Properties of QR Algorithm with Shift

- From $\mathbf{Q}^{(k)}\mathbf{R}^{(k)} = \mathbf{A}^{(k-1)} - \mu^{(k)}\mathbf{I}$ and $\mathbf{A}^{(k)} = \mathbf{R}^{(k)}\mathbf{Q}^{(k)} + \mu^{(k)}\mathbf{I}$, we have

$$\mathbf{A}^{(k)} = (\mathbf{Q}^{(k)})^T\mathbf{A}^{(k-1)}\mathbf{Q}^{(k)}$$

  Then by induction, $\mathbf{A}^{(k)} = (\underline{\mathbf{Q}}^{(k)})^T\mathbf{A}\underline{\mathbf{Q}}^{(k)}$

- However, instead of $\mathbf{A}^k = \underline{\mathbf{Q}}^{(k)}\underline{\mathbf{R}}^{(k)}$, we have

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

  which can be shown by induction

- In other words, $\underline{\mathbf{Q}}^{(k)}$ is orthogonalization of $\prod_{j=k}^{1}(\mathbf{A} - \mu^{(k)}\mathbf{I})$
- If shifts are good estimates of eigenvalues, then last column of $\underline{\mathbf{Q}}^{(k)}$ converges to corresponding eigenvector

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

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

$$\mu^{(k)} = r(\mathbf{q}_m^{(k)}) = (\mathbf{q}_m^{(k)})^T \mathbf{A} \mathbf{q}_m^{(k)}$$

- As in Rayleigh quotient iteration, last column $\mathbf{q}_m^{(k)}$ converges cubically
- This Rayleigh quotient appears as (m, m) entry of $\mathbf{A}^{(k)}$ since $\mathbf{A}^{(k)} = (\underline{\mathbf{Q}}^{(k)})^T \mathbf{A} \underline{\mathbf{Q}}^{(k)}$
- Rayleigh quotient shift corresponds to setting $\mu^{(k)} = A_{mm}^{(k)}$

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

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

$$\mu = a_m - sign(\delta) b_{m-1}^2 / (|\delta| + \sqrt{\delta^2 + b_{m-1}^2})$$

  where $\delta = (a_{m-1} - a_m)/2$; if $\delta = 0$, set $sign(\delta) = 1$ (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 $\mathbf{A}$ at the beginning. The tridiagonal structure is preserved by $\mathbf{A}^{(k)}$
- deflation of $\mathbf{A}$ into submatrices when $\mathbf{A}^{(k)}$ is separable

Algorithm: "Practical" QR Algorithm

$(\mathbf{Q}^{(0)})^T\mathbf{A}^{(0)}\mathbf{Q}^{(0)} = \mathbf{A}$ { tridiagnolization of $\mathbf{A}$}
for $k = 1, 2, \ldots$
    Pick a shift $\mu^{(k)}$
    $\mathbf{Q}^{(k)}\mathbf{R}^{(k)} = \mathbf{A}^{(k-1)} - \mu^{(k)}\mathbf{I}$
    $\mathbf{A}^{(k)} = \mathbf{R}^{(k)}\mathbf{Q}^{(k)} + \mu^{(k)}\mathbf{I}$
    If any off-diagonal element $a^{(k)}_{j,j+1}$ is sufficiently close to zero, set $a_{j,j+1} = a_{j+1,j} = 0$ to obtain $\begin{bmatrix} \mathbf{A}_1 & \\ & \mathbf{A}_2 \end{bmatrix} = \mathbf{A}^{(k)}$ and apply QR algorithm to $\mathbf{A}_1$ and $\mathbf{A}_2$

Stability and Accuracy

Theorem

QR algorithm is backward stable

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

where $\tilde{\mathbf{\Lambda}}$ is computed $\mathbf{\Lambda}$ and $\tilde{\mathbf{Q}}$ 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 shown that $|\tilde{\lambda}_j - \lambda_j| \leqslant \|\delta\mathbf{A}\|_2$, and therefore,

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

where $\tilde{\lambda}_j$ are the computed eigenvalues