EECS 275 Matrix Computation

Ming-Hsuan Yang

Electrical Engineering and Computer Science
University of California at Merced
Merced, CA 95344
http://faculty.ucmerced.edu/mhyang

Lecture 17

Overview

- QR algorithm without shifts
- Simultaneous iteration
- QR algorithm with shifts
- Wilkinson shifts

Reading

- Chapter 28-29 of *Numerical Linear Algebra* by Llyod Trefethen and David Bau
- Chapter 8 of *Matrix Computations* by Gene Golub and Charles Van Loan

The QR algorithm

- The QR algorithm, dating to the early 1960s, is one of the jewels of numerical analysis
- In its simplest form, it can be viewed as a stable procedure for computing QR factorizations of the matrix powers $A, A^2, A^3, \ldots$
- Useful for solving eigenvalue problems

Pure QR algorithm

- "Pure" QR algorithm:

  $A^{(0)} = A$
  for $k = 1, 2, \ldots$ **do**
  $\quad Q^{(k)}R^{(k)} = A^{(k-1)}$ // QR factorization of $A^{(k-1)}$
  $\quad A^{(k)} = R^{(k)}Q^{(k)}$ // Recombine factors in reverse order
  end for

- Take a QR factorization, multiply the computed factors Q and R together in the reverse order RQ, and repeat
- Under suitable assumptions, this simple algorithm converges to a Schur form for the matrix, upper triangular if A is arbitrary, diagonal if A is Hermitian
- Here we assume A is real and symmetric with real eigenvalues λ_j and orthonormal eigenvectors $\mathbf{q}_j$, i.e., interest in the convergence of the matrices $A^{(k)}$ to diagonal form

QR algorithm (cont'd)

- The QR algorithm

$$
\begin{aligned}
Q^{(k)} R^{(k)} &= A^{(k-1)} && \text{// QR factorization of} A^{(k-1)} \\
A^{(k)} &= R^{(k)} Q^{(k)} && \text{// Recombine factors in reverse order}
\end{aligned}
$$

- Carry out similarity transformation ($A \mapsto X^{-1}AX$)
  - triangularize $A^{(k)}$ by forming $R^{(k)} = (Q^{(k)})^\top A^{(k-1)}$
  - multiply on the right by $Q^{(k)}$ gives $A^{(k)} = (Q^{(k)})^\top A^{(k-1)} Q^{(k)}$
- Recall if $X \in \mathbb{C}^{m \times m}$ is nonsingular, then the map $A \mapsto X^{-1}AX$ is a similarity transformation of A
- Also recall an eigendecomposition of a square matrix A is a factorization $A = X \Lambda X^{-1}$ where X is a nonsingular and Λ is diagonal

QR algorithm (cont'd)

- Like Rayleigh quotient iteration, the QR algorithm for real symmetric matrices converges cubically
- However, it must be modified by introducing shifts at each step
- The use of shifts is one of the three modifications required to bring it closer to practical algorithm
  - before starting the iteration, A is reduced to tridiagonal form (e.g., using Hessenberg reduction)
  - instead of $A^{(k)}$, a shifted matrix $A^{(k)} - \mu^{(k)} I$ is factored at each step, where $\mu^{(k)}$ is some eigenvalue estimate
  - whenever possible, and in particular whenever an eigenvalue is found, the problem is deflated by breaking $A^{(k)}$ into submatrices

Practical QR algorithm

- Practical QR algorithm

  $(Q^{(0)})^\top A^{(0)} Q^{(0)} = A \quad // \; A^{(0)}$ is a tridiagonalization of A

  for $k = 1, 2, \ldots$ **do**

  Pick a shift $\mu^{(k)} \quad // \;$ e.g., choose $\mu^{(k)} = A^{(k-1)}_{mm}$

  $Q^{(k)} R^{(k)} = A^{(k-1)} - \mu^{(k)} I \quad // \;$ QR factorization of $A^{(k-1)} - \mu^{(k)} I$

  $A^{(k)} = R^{(k)} Q^{(k)} + \mu^{(k)} I \quad // \;$ Recombine factors in reverse order

  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

  $$\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} = A^{(k)}$$

  and apply the QR algorithm to A_1 and A_2

  end for

- The QR algorithm with well-chosen shifts has been the standard method for computing all eigenvalues of a matrix since the early 1960

Unnormalized simultaneous iteration

- Idea: apply the power iteration to several vectors at once (also known as block power iteration)
- Suppose we start with a set of n linearly independent vectors $\mathbf{v}_1^{(0)}, \ldots, \mathbf{v}_n^{(0)}$
- As $A^k \mathbf{v}_1^{(0)}$ converges as $k \to \infty$ (under suitable assumptions) to the eigenvector corresponding to the largest eigenvalue of A in absolute value
- The space $\langle A^k \mathbf{v}_1^{(0)}, \ldots, A^k \mathbf{v}_n^{(0)} \rangle$ should converge (under suitable assumptions) to the space $\langle \mathbf{q}_1, \ldots, \mathbf{q}_n \rangle$ spanned by the eigenvectors $\mathbf{q}_1, \ldots, \mathbf{q}_n$ of A corresponding to the n largest eigenvalues in absolute value

Unnormalized simultaneous iteration (cont'd)

- In matrix notation, define $V^{(0)}$ to be the $m \times n$ initial matrix

$$V^{(0)} = \begin{bmatrix} \mathbf{v}_1^{(0)} & \cdots & \mathbf{v}_n^{(0)} \end{bmatrix}$$

  and define $V^{(k)}$ to the result after k applications of A:

$$V^{(k)} = A^k V^{(0)} = \begin{bmatrix} \mathbf{v}_1^{(k)} & \cdots & \mathbf{v}_n^{(k)} \end{bmatrix}$$

- Extract a well-behaved basis for this space by computing a reduced QR factorization of $V^{(k)}$

$$\hat{Q}^{(k)} \hat{R}^{(k)} = V^{(k)}$$

  where $\hat{Q}^{(k)}$ and $\hat{R}^{(k)}$ have dimensions $m \times n$ and $n \times n$, respectively
- As $k \to \infty$, the columns should converge to eigenvectors $\pm\mathbf{q}_1$, $\pm\mathbf{q}_2, \ldots, \pm\mathbf{q}_n$

Analysis of simultaneous iteration

- If we expand $\mathbf{v}_j^{(0)}$ and $\mathbf{v}_j^{(k)}$ in the eigenvectors of A, we have

$$\begin{aligned} \mathbf{v}_j^{(0)} &= a_{1j}\mathbf{q}_1 + \cdots + a_{mj}\mathbf{q}_m \\ \mathbf{v}_j^{(k)} &= \lambda_1^k a_{1j}\mathbf{q}_1 + \cdots + \lambda_m^k a_{mj}\mathbf{q}_m \end{aligned}$$

- Simple convergence results will hold if two conditions are satisfied
  - the leading $n+1$ eigenvalues are distinct in absolute value:

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| > |\lambda_{n+1}| \geq |\lambda_{n+2}| \geq \cdots \geq |\lambda_m|$$

  - the collection of expansion coefficients a_{ij} is in nonsingular. Define $\hat{Q}$ as the $m \times n$ matrix whose columns are the eigenvectors $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n$. We assume the following

    All the leading principal submatrices of $\hat{Q}^\top V^{(0)}$ are nonsingular

  - namely, the upper-left 1×1, 2×2, $\ldots$, $n \times n$ submatrices are nonsingular

Simultaneous iteration

- As $k \to \infty$, the vectors $\mathbf{v}_1^{(k)}, \ldots, \mathbf{v}_n^{(k)}$ all converge to multiples of the same dominant eigenvector $\mathbf{q}_1$ of A
- Thus, although the space they span $\langle \mathbf{v}_1^{(k)}, \ldots, \mathbf{v}_j^{(k)} \rangle$ converges to something useful, these vectors constitute a highly ill-conditioned basis of that space
- Need to orthonormalize at each step rather than once for all
- Use a different sequence of matrices $Z^{(k)}$ rather than $V^{(k)}$ as before

$\hat{Q}^{(0)} \in \mathbb{R}^{m \times n}$ with orthonormal columns
for $k = 1, 2, \ldots$ **do**
  $Z = A\hat{Q}^{(k-1)}$
  $\hat{Q}^{(k)}\hat{R}^{(k)} = Z$ // reduced QR factorization of Z
end for

- The column spaces of $\hat{Q}^{(k)}$ and $Z^{(k)}$ are the same, both being equal to the column space of $A^k\hat{Q}^{(0)}$

Simultaneous iteration $\Longleftrightarrow$ QR Algorithm

- The QR algorithm is equivalent to simultaneous iteration applied to a full set of $n = m$ initial vectors, namely, the identity, $\hat{Q}^{(0)} = I$
- Since the matrices $\hat{Q}^{(k)}$ are now square, we are dealing with full QR factorization and can drop the hats on $\hat{Q}^{(k)}$ and $\hat{R}^{(k)}$
- Will replace $\hat{R}^{(k)}$ by $R^{(k)}$ but $\hat{Q}^{(k)}$ by $\underline{Q}^{(k)}$ to distinguish the Q matrices of simultaneous iteration from those of the QR algorithm

Simultaneous iteration and unshifted QR algorithm

Simultaneous iteration

$$\underline{Q}^{(0)} = I \quad (1)$$
$$Z = A\underline{Q}^{(k-1)} \quad (2)$$
$$Z = \underline{Q}^{(k)}R^{(k)} \quad (3)$$
$$A^{(k)} = (\underline{Q}^{(k)})^\top A\underline{Q}^{(k)} \quad (4)$$

Unshifted QR algorithm

$$A^{(0)} = A \quad (5)$$
$$A^{(k-1)} = Q^{(k)}R^{(k)} \quad (6)$$
$$A^{(k)} = R^{(k)}Q^{(k)} \quad (7)$$
$$\underline{Q}^{(k)} = Q^{(1)}Q^{(2)}\cdots Q^{(k)} \quad (8)$$

For both algorithms, we define one $m \times m$ matrix $\underline{R}^{(k)}$

$$\underline{R}^{(k)} = R^{(k)}R^{(k-1)}\cdots R^{(1)} \quad (9)$$

Theorem

The above processes generate identical sequences of matrices $\underline{R}^{(k)}$, $\underline{Q}^{(k)}$, and $A^{(k)}$, namely, those defined by the QR factorization of the k-th power of A

$$A^k = \underline{Q}^{(k)}\underline{R}^{(k)} \quad (10)$$

together with the projection

$$A^{(k)} = (\underline{Q}^{(k)})^\top A\underline{Q}^{(k)} \quad (11)$$

Simultaneous iteration and QR algorithm

Proof.

The case for $k = 0$ is trivial. For both simultaneous iteration and the QR algorithm imply $A^0 = \underline{Q}^{(0)} = \underline{R}^{(0)} = I$ and $A^{(0)} = A$ from which the results are immediate.

For $k \geq 1$ for simultaneous iteration

$$A^k = A \underbrace{A^{k-1}}_{(10)} = \underbrace{A\underline{Q}^{(k-1)}}_{(2)(3)} \underline{R}^{(k-1)} = \underline{Q}^{(k)} R^{(k)} \underline{R}^{(k-1)} = \underline{Q}^{(k)} \underline{R}^{(k)}$$

For $k \geq 1$ for the QR algorithm

$$A^k = A \underbrace{A^{k-1}}_{(10)} = \underbrace{A\underline{Q}^{(k-1)}}_{(11)} \underline{R}^{(k-1)} = \underline{Q}^{(k-1)} \underbrace{A^{(k-1)}}_{(6)(8)} \underline{R}^{(k-1)} = \underline{Q}^{(k)} \underline{R}^{(k)}$$

Finally,

$$\underbrace{A^{(k)}}_{(7)} = \underbrace{R^{(k)}}_{(6)} Q^{(k)} = (Q^{(k)})^\top \underbrace{A^{(k-1)}}_{(11)} Q^{(k)} = (\underline{Q}^{(k)})^\top A \underline{Q}^{(k)}$$

Convergence of the QR algorithm

- Qualitative understanding of (10) and (11) is the key
- First, (10) explains why the QR algorithm can be expected to find eigenvectors: it constructs orthonormal bases for successive powers A^k
- Second, (11) explains why the algorithm finds eigenvalues
- It follows from (11) that the diagonal elements of $A^{(k)}$ are Rayleigh quotients of A corresponding to the columns of $\underline{Q}^{(k)}$
- As these columns converge to eigenvectors, the Rayleigh quotients converge to the corresponding eigenvalues
- Meanwhile, (11) implies that the off-diagonal elements of $A^{(k)}$ correspond to generalized Rayleigh quotients involving approximations of distinct eigenvectors of A on the left and the right
- Since these approximations must become orthogonal as they converge to distinct eigenvectors, the off-diagonal elements of $A^{(k)}$ must converge to zero

QR algorithm with shifts

- What makes QR iteration works in practice is the introduction of $A \to A - \mu I$ at each step
- An implicit connection to the Rayleigh quotient iteration
- The "pure" QR algorithm is equivalent to simultaneous iteration applied to the identity matrix
- In particular, the first column of the result evolves according to the power iteration applied to $\mathbf{e}_1$
- The "pure" QR algorithm is also equivalent to simultaneous inverse iteration applied to a "flipped" identity matrix P, and the m-th column of the result evolves according to inverse iteration applied to $\mathbf{e}_m$
- Let $Q^{(k)}$ be the orthogonal factor at the k-th step of the QR algorithm, the accumulated product of these matrices

$$\underline{Q}^{(k)} = \prod_{j=1}^{k} Q^{(j)} = \begin{bmatrix} \mathbf{q}_1^{(k)} & \mathbf{q}_2^{(k)} & \cdots & \mathbf{q}_m^{(k)} \end{bmatrix}$$

is the same orthogonal matrix that appears at step k of simultaneous iteration

QR algorithm with shifts (cont'd)

- Another way to put this is to say $\underline{Q}^{(k)}$ is the orthogonal factor in a QR factorization

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

- If we invert this formula, we calculate

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

  for the second equality we have used the fact that A^{-1} is symmetric. Let P denote the $m \times m$ permutation matrix that reverse row or column order

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

- PA swap rows of A, and AP swaps columns of A

QR algorithm with shifts (cont'd)

- Since $P^2 = I$, we have

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

  where the first factor is this product, $\underline{Q}^{(k)}P$, is orthogonal, and the second factor, $P(\underline{R}^{(k)})^{-\top}P$, is upper triangular (start with lower triangular matrix $(\underline{R}^{(k)})^{-\top}$, flip it top-to-bottom, then flip again left-to-right)
- Can be interpreted as a QR factorization of $A^{-k}P$
- In other words, we effectively carry out simultaneous iteration on A^{-1} applied to the initial matrix P, which is to say, simultaneous inverse iteration on A
- In particular, the first column of $\underline{Q}^{(k)}P$, i.e., the last column of $\underline{Q}^{(k)}$, is the result of applying k steps of inverse iteration to the vector $\mathbf{e}_m$

Connection with shifted inverse iteration

- The QR algorithm is both simultaneous iteration and simultaneous inverse iteration: the symmetry is perfect
- However, there is a huge difference between power iteration and inverse iteration as the latter can be accelerated arbitrarily through the use of shifts
- The better we can estimate an eigenvalue $\mu \approx \lambda_J$, the more we can accomplish by a step of inverse iteration with shifted matrix $A - \mu I$
- This corresponds to shifts in the simultaneous iteration and inverse iteration
- Let $\mu^{(k)}$ denote the eigenvalue estimate chosen at the k-th step of the QR algorithm, the relationship between steps $k-1$ and k of the shifted QR algorithm is

$$\begin{aligned} A^{(k-1)} - \mu^{(k)} I &= Q^{(k)} R^{(k)} \\ A^{(k)} &= R^{(k)} Q^{(k)} + \mu^{(k)} I \end{aligned}$$

Connection with shifted inverse iteration (cont'd)

- This implies
$$A^{(k)} = (Q^{(k)})^\top A^{(k-1)} Q^{(k)}$$
and by induction
$$A^{(k)} = (\underline{Q}^{(k)})^\top A \underline{Q}^{(k)}$$
which is unchanged from that in the QR algorithm without shifts
- However, the equation, $A^k = \underline{Q}^{(k)} \underline{R}^{(k)}$, in QR algorithm without shifts, is no longer true. Instead, we have the factorization
$$(A - \mu^{(k)} I)(A - \mu^{(k-1)} I) \cdots (A - \mu^{(1)} I) = \underline{Q}^{(k)} \underline{R}^{(k)}$$
a shifted variation on simultaneous iteration
- In other words, $\underline{Q}^{(k)} = \prod_{j=1}^{k} Q^{(j)}$ is an orthogonalization of $\prod_{j=1}^{k} (A - \mu^{(j)} I)$
- The first column of $\underline{Q}^{(k)}$ is the result of applying shifted power iteration to $\mathbf{e}_1$ using the shifts $\mu^{(j)}$ and the last column is the result of applying k steps of shifted inverse iteration to $\mathbf{e}_m$ with the same shifts
- If the shifts are good eigenvalue estimates, this last column of $\underline{Q}^{(k)}$ converges quickly to an eigenvector

Connection with Rayleigh quotient iteration

- To estimate the eigenvalue corresponding to the eigenvector approximated by the last column of $\underline{Q}^{(k)}$, it is natural to apply the Rayleigh quotient to this last column

$$\mu^{(k)} = \frac{(\mathbf{q}_m^{(k)})^\top A \mathbf{q}_m^{(k)}}{(\mathbf{q}_m^{(k)})^\top \mathbf{q}_m^{(k)}} = (\mathbf{q}_m^{(k)})^\top A \mathbf{q}_m^{(k)}$$

- If this number is chosen as the shift at every step, the eigenvalue and eigenvector estimates $\mu^{(k)}$ and $\mathbf{q}_m^{(k)}$ are identical to those that are computed by the Rayleigh quotient iteration with $\mathbf{e}_m$
- Thus, the QR algorithm has cubic convergence in the sense that $\mathbf{q}_m^{(k)}$ converges cubically to an eigenvector
- Notice that the Rayleigh quotient $r(\mathbf{q}_m^{(k)})$ appears as the m, m entry of $A^{(k)}$, so it comes for free!

$$A_{mm}^{(k)} = \mathbf{e}_m^\top A^{(k)} \mathbf{e}_m = \mathbf{e}_m^\top \underline{Q}^{(k)\top} A \underline{Q}^{(k)} \mathbf{e}_m = \mathbf{q}_m^{(k)\top} A \mathbf{q}_m^{(k)}$$

- Thus the above equation of $\mu^{(k)}$ is simply setting $\mu^{(k)} = A_{mm}^{(k)}$, which is known as the Rayleigh quotient shift

Wilkinson shift

- Although the Rayleigh quotient shift gives cubic convergence, convergence is not guaranteed for all initial condition
- Consider
$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$
The unshifted QR algorithm does not converge
$$\begin{aligned} A &= Q^{(1)}R^{(1)} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ A^{(1)} &= R^{(1)}Q^{(1)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = A \end{aligned}$$
- The Rayleigh quotient shift $\mu = A_{mm}$, however, has no effect either, since $A_{mm} = 0$
- The problem arises because of the symmetry of the eigenvalues, one is $+1$ and the other is -1
- Need an eigenvalue estimate that can break the symmetry

Wilkinson shift (cont'd)

- Let B denote the lower rightmost 2×2 submatrix $A^{(k)}$

$$B = \begin{bmatrix} a_{m-1} & b_{m-1} \\ b_{m-1} & a_m \end{bmatrix}$$

- The Wilkinson shift is defined as that eigenvalue of B that is closer to a_m where in the case of a tie, one of the two eigenvalues of B is chosen arbitrarily
- A numerically stable formula for the Wilkinson shift is

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

  where $\delta = (a_{m-1} - a_m)/2$ and if $\delta = 0$, $\text{sign}(\delta)$ can be arbitrarily set to 1 or -1

Wilkinson shift (cont'd)

- Like Rayleigh quotient shift, the Wilkinson shift achieves cubic convergence in the generic case
- It can be shown that it achieves at least quadratic convergence in the worst case
- The QR algorithm with Wilkinson shift always converges

Other eigenvalue algorithms

- Jacobi algorithm: one of the oldest ideas for computing eigenvalues
- Bisection method: when one does not need all the eigenvalues (e.g., largest 10%)
- Divide-and-conquer algorithm