Lecture 20
Other Eigenvalue Algorithms; Computing SVD

Songting Luo

Department of Mathematics
Iowa State University

MATH 562 Numerical Analysis II

Outline

Outline

Three Alternative Algorithms

- Jacobi algorithm: earliest known method
- Bisection method: standard way for finding few eigenvalues
- Divide-and-conquer: faster than QR and amenable to parallelization

The Jacobi Algorithm

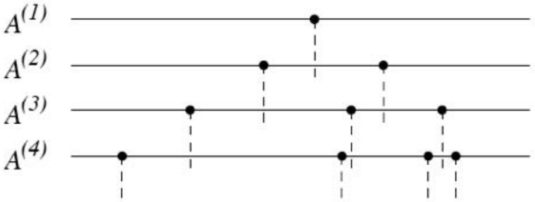
- Diagonalize 2×2 real symmetric matrix by Jacobi rotation
$$\mathbf{J}^T \begin{bmatrix} a & d \\ d & b \end{bmatrix} \mathbf{J} = \begin{bmatrix} \neq 0 & 0 \\ 0 & \neq 0 \end{bmatrix}$$
where $\mathbf{J} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$, and $\tan(2\theta) = 2d/(b-a)$.
- What are its similarity and differences with Givens rotation?
- Iteratively apply transformation to two rows and two corresponding columns of $\mathbf{A} \in \mathbb{R}^{m \times m}$
- Need not tridiagonalize first, but loop over all pairs of rows and columns by choosing greedily or cyclically
- Magnitude of nonzeros shrink steadily, converging quadratic
- In each iteration, $O(m^2)$ Jacobi rotation, $O(m)$ operations per rotation, leading to $O(m^3 \log(|\log \epsilon_{machine}|))$ flops total
- Jacobi method is easy to parallelize (QR algorithm does not scale well), delivers better accuracy than QR algorithm, but far slower than QR algorithm

Method of Bisection

- Idea: Search the real line for roots of $p(x) = det(\mathbf{A} - x\mathbf{I})$
- Finding roots from coefficients is highly unstable, but computing $p(x)$ from given x is stable (e.g., can be computed using Gaussian elimination with partial pivoting)
- Let $\mathbf{A}^{(i)}$ denote principal square submatrix of dimension i for irreducible matrix $\mathbf{A}$ (note: different from notation in QR algorithm)
- Key property: eigenvalues of $\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(m)}$ strictly interlace

$$\lambda_j^{(k+1)} < \lambda_j^{(k)} < \lambda_{j+1}^{(k+1)},$$

for $k = 1, \ldots, m-1$,

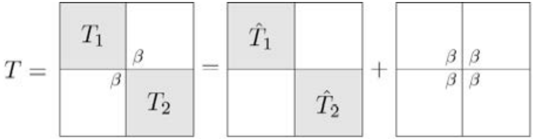


Method of Bisection

- Interlacing property allows us to determine number of negative eigenvalues of $\mathbf{A}$, which is equal to number of sign changes in Sturm sequence
$$1, det(\mathbf{A}^{(1)}), det(\mathbf{A}^{(2)}), \dots, det(\mathbf{A}^{(m)}).$$
- Shift $\mathbf{A}$ to get number of eigenvalues in $(-\infty,\ b)$ and $(-\infty, a)$, and in turn $[a,\ b)$
- Three-term recurrence for determinants for tridiagonal matrices
$$det(\mathbf{A}^{(k)}) = a_{k,k} det(\mathbf{A}^{(k-1)}) - a_{k,k-1}^2 det(\mathbf{A}^{(k-2)})$$
- With shift $x\mathbf{I}$ and $p^{(k)}(x) = det(\mathbf{A}^{(k)} - x\mathbf{I})$:
$$p^{(k)}(x) = (a_{k,k} - x)p^{(k-1)}(x) - a_{k,k-1}^2 p^{(k-2)}(x)$$
- Bisection algorithm can locate eigenvalues in arbitrarily small intervals
- $O(m|\log(\epsilon_{machine})|)$ flops per eigenvalue, always high relative accuracy

Divide-and-Conquer Algorithm

- Split symmetric $\mathbf{T}$ into submatrices



- Sum of 2 block-diagonal matrix and rank-one correction
- Split $\mathbf{T}$ in equal sizes and compute eigenvalues of $\hat{\mathbf{T}}_1$ and $\hat{\mathbf{T}}_2$ recursively
- Solve nonlinear problem to get eigenvalues of $\mathbf{T}$ from those of $\hat{\mathbf{T}}_1$ and $\hat{\mathbf{T}}_2$.

Divide-and-Conquer Algorithm

- Suppose diagonalization $\hat{\mathbf{T}}_1 = \mathbf{Q}_1\mathbf{D}_1\mathbf{Q}_1^T$ and $\hat{\mathbf{T}}_2 = \mathbf{Q}_2\mathbf{D}_2\mathbf{Q}_2^T$ have been computed. We then have

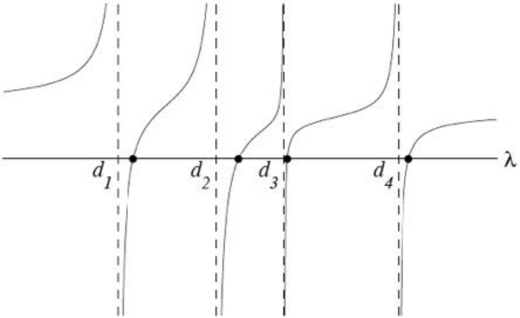
$$\mathbf{T} = \begin{bmatrix} \mathbf{Q}_1 & 0 \\ 0 & \mathbf{Q}_2 \end{bmatrix} \left(\begin{bmatrix} \mathbf{D}_1 & 0 \\ 0 & \mathbf{D}_2 \end{bmatrix} + \beta \mathbf{z}\mathbf{z}^T \right) \begin{bmatrix} \mathbf{Q}_1^T & 0 \\ 0 & \mathbf{Q}_2^T \end{bmatrix}$$

  with $\mathbf{z}^T = (\mathbf{q}_1^T, \mathbf{q}_2^T)$, where $\mathbf{q}_1^T$ is last row of $\mathbf{Q}_1$ and $\mathbf{q}_2^T$ is first row of $\mathbf{Q}_2$
- This is similarity transformation: Find eigenvalues of diagonal matrix plus rank-one correction

Divide-and-Conquer Algorithm

- Eigenvalues of $\mathbf{D} + \mathbf{w}\mathbf{w}^T$ are the roots of rational function

$$f(\lambda) = 1 + \sum_{j=1}^{m} \frac{w_j^2}{d_j - \lambda}$$



Divide-and-Conquer Algorithm

- Solve secular equation $f(\lambda) = 0$ with nonlinear solver
- $O(m)$ flops per root, $O(m^2)$ flops for all roots.
- Total cost for divide-and-conquer algorithm is $O(m^2)$
- For computing eigenvalues only, most of operations are spent in tridiagonal reduction, and constant in "Phase 2" is not important
- However, for computing eigenvectors, divide-and conquer reduces phase 2 to $4m^3/3$ flops compared to $6m^3$ for QR.

Outline

Computing SVD

- Intuitive idea for computing SVD of $\mathbf{A} \in \mathbb{R}^{m \times m}$:
  - Form $\mathbf{A}^*\mathbf{A}$ and compute its eigenvalue decomposition $\mathbf{A}^*\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^*$
  - Let $\mathbf{\Sigma} = \sqrt{\mathbf{\Lambda}}$
  - Solve system $\mathbf{U}\mathbf{\Sigma} = \mathbf{A}\mathbf{V}$ to obtain $\mathbf{U}$
- This method can be very efficient if $m \gg n$
- However, it is not very stable, especially for smaller singular values because of the squaring of the condition number
  - For SVD of $\mathbf{A}$, $|\tilde{\sigma}_k - \sigma_k| = O(\epsilon_{machine}\|\mathbf{A}\|)$, where $\tilde{\sigma}_k$ and σ_k denote the computed and exact kth singular value
  - If computed from eigenvalue decomposition of $\mathbf{A}^*\mathbf{A}$, $|\tilde{\sigma}_k - \sigma_k| = O(\epsilon_{machine}\|\mathbf{A}\|^2/\sigma_k)$, which is problematic if $\sigma_k \ll \|\mathbf{A}\|$
- If one is interested in only relatively large singular values, then using eigenvalue decomposition is not a problem. For general situations, a more stable algorithm is desired.

Computing SVD

- Typical algorithm for computing SVD are similar to computation of eigenvalues
- Consider $\mathbf{A} \in \mathbb{R}^{m\times m}$, then hermitian matrix $\mathbf{H} = \begin{bmatrix} \mathbf{0} & \mathbf{A}^* \\ \mathbf{A} & \mathbf{0} \end{bmatrix}$ has eigenvalue decomposition

$$\mathbf{H}\begin{bmatrix} \mathbf{V} & \mathbf{V} \\ \mathbf{U} & -\mathbf{U} \end{bmatrix} = \begin{bmatrix} \mathbf{V} & \mathbf{V} \\ \mathbf{U} & -\mathbf{U} \end{bmatrix}\begin{bmatrix} \mathbf{\Sigma} & \mathbf{0} \\ \mathbf{0} & -\mathbf{\Sigma} \end{bmatrix}$$

  where $\mathbf{A} = \mathbf{U\Sigma V}$ gives the SVD. This approach is stable.
- In practice, such a reduction is done implicitly without forming the large matrix
- Typically done in two or more stages:
  - First, reduce to bidiagonal form by applying different orthogonal transformations on left and right,
  - Second, reduce to diagonal form using a variant of QR algorithm or divide-and-conquer algorithm

Generalized Eigenvalue Problem

- Generalized eigenvalue problem has the form

$$\mathbf{Ax} = \lambda \mathbf{Bx}$$

  where $\mathbf{A}$ and $\mathbf{B}$ are $m \times m$ matrices.
- For example, in structural vibration problems, $\mathbf{A}$ represents the stiffness matrix, $\mathbf{B}$ the mass matrix, and eigenvalues and eigenvectors determine natural frequencies and modes of vibration of structures
- If $\mathbf{A}$ or $\mathbf{B}$ is nonsingular, then it can be converted into standard eigenvalue problem $(\mathbf{B}^{-1}\mathbf{A})\mathbf{x} = \lambda\mathbf{x}$ or $(\mathbf{A}^{-1}\mathbf{B})\mathbf{x} = (1/\lambda)\mathbf{x}$
- If $\mathbf{A}$ and $\mathbf{B}$ are both symmetric, preceding transformation loses symmetry and in turn may lose orthogonality of generalized eigenvectors. If $\mathbf{B}$ is positive definite, alternative transformation is

$$(\mathbf{L}^{-1}\mathbf{A}\mathbf{A}^{-T})\mathbf{y} = \lambda\mathbf{y}, \text{ where } \mathbf{B} = \mathbf{L}\mathbf{L}^T \text{ and } \mathbf{y} = \mathbf{L}^T\mathbf{x}$$

- If $\mathbf{A}$ and $\mathbf{B}$ are both singular or indefinite, then use QZ algorithm to reduce $\mathbf{A}$ and $\mathbf{B}$ into triangular matrices simultaneously by orthogonal transformation (see Golub and van Loan for detail)