Algorithms for Symmetric Eigenproblems

Eigenvalues in their own ("eigen") way

Radu Trîmbițaș

"Babeș-Bolyai" University

April 8, 2009

Real Symmetric Matrices

- $A \in \mathbb{R}^{m\times m}$ is symmetric if $A^T = A$
- All eigenvalues of A $\lambda_1, \ldots, \lambda_m$ are real
- A is unitarily diagonalizable — A has a set of m linearly independent orthonormal vectors $q_1, \ldots, q_m$
- Eigenvectors are normalized $\|q_j\| = 1$, and sometimes the eigenvalues are ordered in a particular way
- Initial reduction to tridiagonal form is assumed

Rayleigh Quotient

- The Rayleigh quotient of a symmetric matrix A and a vector $x \neq 0$ is the scalar
$$\rho(x, A) = \frac{x^T A x}{x^T x}$$
- For an eigenvector x, and the corresponding eigenvalue λ, $\rho(x, A) = \lambda$
- For general x, $\rho(x, A) = \alpha$ that minimizes $\|Ax - \alpha x\|_2$
- x eigenvector of $A \Longleftrightarrow \nabla \rho(x, A) = 0$
- $\rho(\cdot, A)$ is smooth and $\nabla \rho(q_j, A) = 0$, therefore quadratically accurate:
$$\rho(x, A) - \rho(q_j, A) = O(\|x - q_j\|^2) \quad \text{as } x \to q_j$$

Rayleigh Quotient Iteration

- Improvement of inverse iteration, set shift σ to last computed Rayleigh quotient

Rayleight Quotient Iteration: Given $x^{(0)}$ with $\|x^{(0)}\|_2 = 1$, and *tol* we iterate

$\rho_0 := \rho(x^{(0)}, A);$
$i := 1;$
repeat
  $y^{(i)} := (A - \rho_{i-1} I)^{-1} x^{(i-1)};$
  $x^{(i)} := y^{(i)} / \|y^{(i)}\|_2;$
  $\rho_i := \rho(x^{(i)}, A);$
  $i := i + 1;$
until convergence ($\|Ax^{(i)} - \rho_i x^{(i)}\|_2 < tol$)

Convergence of Rayleigh Quotient Iteration

- Cubic convergence in Rayleigh quotient iteration:

$$\left\| y^{(i+1)} - (\pm q_J) \right\| = O(\left\| y^{(i)} - (\pm q_J) \right\|^3)$$

and

$$|\rho_{i+1} - \lambda_J| = O(|\rho_i - \lambda_J|^3)$$

- Proof idea: If $y^{(i)}$ is close to an eigenvector, $\left\| y^{(i)} - q_J \right\| \leq \varepsilon$, then the accuracy of the Rayleigh quotient estimate ρ_i is $|\rho_{i+1} - \lambda_J| = O(\varepsilon^2)$. One step of inverse iteration then gives

$$\left\| y^{(i+1)} - q_J \right\| = O\left(|\rho_i - \lambda_J| \left\| y^{(i)} - q_J \right\| \right) = O(\varepsilon^3)$$

QR Iteration for Symmetric Matrices

- First, reduce A to tridiagonal form

"Practical" QR algorithm : Given $(Q^{(0)})^T A^{(0)} Q^{(0)} = A$ ($A^{(0)}$ a tridiagonalization of A) we iterate

$i := 0;$
repeat
  Choose a shift σ_i near an eigenvalue of A;
  Factor $A^{(i)} - \sigma_i I = Q^{(i)} R^{(i)}$; {QR decomposition}
  $A^{(i+1)} := R^{(i)} Q^{(i)} + \sigma_i I;$
  $i := i + 1;$
  if $A^{(i)}_{j,j+1}$ is sufficiently small **then**
    set $A_{j,j+1} := 0$ and $A_{j+1,j} := 0$ to obtain

$$\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} = A^i$$

    now apply the QR algorithm to A_1 and A_2
until convergence

Choosing shift: The Rayleigh Quotient Shift

- Natural choice for σ_i: Rayleigh quotient for last column of $Q^{(i)}$

$$\sigma_i = \frac{\left(q_m^{(i)}\right)^T A q_m^{(i)}}{\left(q_m^{(i)}\right)^T q_m^{(i)}} = \left(q_m^{(i)}\right)^T A q_m^{(i)}$$

- Cubic convergence for last column $q_m^{(i)}$
- Convenient fact: This Rayleigh quotient appears as m, m entry of $A^{(i)}$ since $A^{(i)} = \left(Q^{(i)}\right)^T A Q^{(i)}$
- The Rayleigh quotient shift corresponds to setting $\sigma_i = A_{m,m}^{(i)}$

Choosing shift: The Wilkinson Shift

- The QR algorithm with Rayleigh quotient shift might fail, e.g. with two symmetric eigenvalues
- Break symmetry by the Wilkinson shift

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

  where $\delta = (a_{m-1} - a_m)/2$ and $B = \begin{bmatrix} a_{m-1} & b_{m-1} \\ b_{m-1} & a_m \end{bmatrix}$ is the lower-right submatrix of $A^{(k)}$
- σ is the eigenvalue of B that is closest to a_m
- Always convergence with this shift, in worst case quadratically

- Idea: a 2×2 real symmetric matrix can be diagonalized by a Jacobi rotation:
$$J \begin{bmatrix} a & d \\ d & b \end{bmatrix} J^T = \begin{bmatrix} \neq 0 & 0 \\ 0 & \neq 0 \end{bmatrix}$$
where
$$J = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \qquad \tan 2\theta = \frac{2d}{a-b}$$
- Jacobi's method does not start by reducing A to tridiagonal form but it produces a sequence of orthogonally similar matrices, which eventually converge to a diagonal matrix with eigenvalues on the diagonal.
- We start from $A_0 = A$; $A_{i+1} = J_i^T A_i J_i$; applying this formula successively we get
$$\begin{aligned} A_m &= J_{m-1}^T A_{m-1} J_{m-1} = J_{m-1}^T J_{m-2}^T A_{m-2} J_{m-2} J_{m-1} = \dots \\ &= J_{m-1}^T \dots J_0^T A_0 J_0 \dots J_{m-1} = J^T A J \end{aligned}$$

The Jacobi Algorithm

- The Jacobi (Givens) rotation at jth step is

$$J_i = R(j,k,\theta) = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & \ddots & & & & & \\ & & & \cos\theta & & -\sin\theta & & & \\ & & & & \ddots & & & & \\ & & & \sin\theta & & \cos\theta & & & \\ & & & & & & \ddots & & \\ & & & & & & & 1 & \\ & & & & & & & & 1 \end{bmatrix}$$

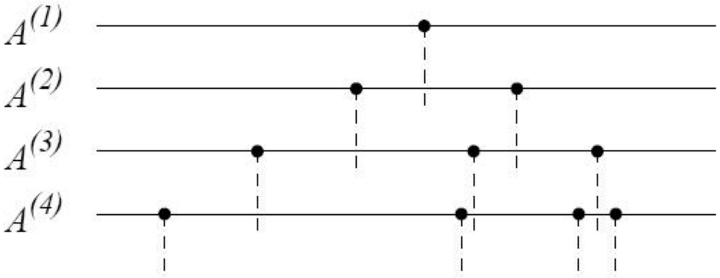
where θ is chosen to zero out the j,k and k,j entries of A_{i+1}.

- Loop over all pairs of rows/columns, quadratic convergence
- $O(m^2)$ steps, $O(m)$ operations per step $\Longrightarrow O(m^3)$ operation count

The Method of Bisection

- Idea: Search the real line for roots of $p(x) = \det(A - xI)$
- Finding roots from coefficients highly unstable, but $p(x)$ could be computed by elimination
- Important property: Eigenvalues of principal upper-left square submatrices $A^{(1)}, \ldots, A^{(m)}$ interlace

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



The Method of Bisection

- Because of the interlacing property: The number of negative eigenvalues of A equals the number of sign changes in the Sturm sequence

$$1, \det(A^{(1)}), \det(A^{(2)}), \ldots, \det(A^{(m)})$$

- Shift A to get number of eigenvalues in $(-\infty, b)$ and twice for $[a, b)$
- Three-term recurrence for the determinants:

$$\det(A^{(k)}) = a_k \det(A^{(k-1)}) - b_{k-1}^2 \det(A^{(k-2)})$$

- With shift xI and $p^{(k)}(x) = \det(A^{(k)} - xI)$:

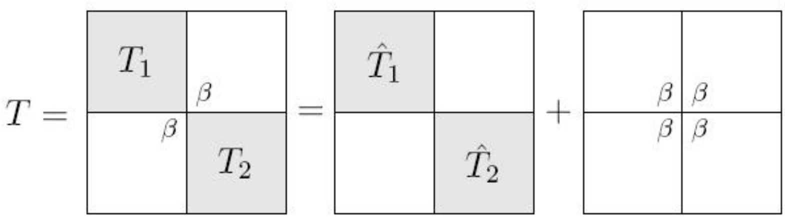
$$p^{(k)}(x) = (a_k - x)p^{(k-1)}(x) - b_{k-1}^2 p^{(k-2)}(x)$$

- $O(m \log(\epsilon_{\text{machine}}))$ flops per eigenvalue, always high relative accuracy

The Divide-and-Conquer Algorithm

Cuppen[1], Gu and Eisenstadt [2]

- Split symmetric tridiagonal T into submatrices:



- This is the sum of a 2×2 block-diagonal matrix and a rank-one correction
- Split T in equal sizes and compute eigenvalues of $\widehat{T}_1$, $\widehat{T}_2$ recursively
- Solve nonlinear problem to get eigenvalues of T from those of $\widehat{T}_1$, $\widehat{T}_2$

- Suppose diagonalizations $\widehat{T}_1 = Q_1 D_1 Q_1^T$ and $\widehat{T}_2 = Q_2 D_2 Q_2^T$ have been computed. We then have

$$T = \begin{bmatrix} Q_1 & \\ & Q_2 \end{bmatrix} \left(\begin{bmatrix} D_1 & \\ & D_2 \end{bmatrix} + \beta z z^T \right) \begin{bmatrix} Q_1^T & \\ & Q_2^T \end{bmatrix}$$

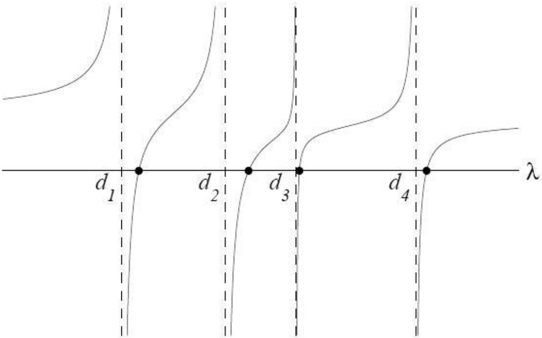
  with $z^T = (q_1^T, q_2^T)$, where q_1^T is the last row of Q_1 and q_2^T is the first row of Q_2

- This is a similarity transformation $\Longrightarrow$ Find eigenvalues of diagonal matrix plus rank-one correction

The Divide-and-Conquer Algorithm

- Eigenvalues of $D + ww^T$ are the roots of the rational function

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



The Divide-and-Conquer Algorithm

- Solve the 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:

$$O\left(m^2 + 2\frac{m^2}{2^2} + 4\frac{m^2}{4^2} + \cdots + m\frac{m^2}{m^2}\right) = O(m^2)$$

- For computing eigenvalues only, most of the operations are spent in the tridiagonal reduction, and the 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

General structure

- All the algorithms for symmetric eigenproblems w.r.t a matrix A, except Jacobi, have the following structure
  1. Reduce A to tridiagonal form T with an orthogonal matrix Q_1: $A = Q_1 T Q_1^T$.
  2. Find the eigendecomposition of T: $T = Q_2 \Lambda Q_2^T$, where Λ is the diagonal matrix of eigenvalues and Q_2 is the orthogonal matrix whose columns are eigenvectors.
  3. Combine this decompositions to get $A = (Q_1 Q_2)\Lambda(Q_1 Q_2)^T$. The columns of $Q = Q_1 Q_2$ are the eigenvectors of A.

General structure - SVD

- All the algorithms for the SVD of a general matrix G, except Jacobi, have the following structure:
  1. Reduce G to bidiagonal form B with orthogonal matrices U_1 and V_1: $G = U_1 B V_1^T$. This means B is nonzero only on the main diagonal and first superdiagonal.
  2. Find the SVD of B: $B = U_2 \Sigma V_2^T$, where Σ is the diagonal matrix of singular values, and U_2 and V_2 are orthogonal matrices whose columns are the left and right singular vectors, respectively.
  3. Combine these decompositions to get $G = (U_1 U_2) \Sigma (V_1 V_2)^T$. The columns of $U = U_1 U_2$ and $V = V_1 V_2$ are the left and the right singular vectors of G, respectively.

Converting SVD to EWD

- There are three ways to convert the problem of finding the SVD of a bidiagonal matrix B to finding the eigenvalues and eigenvectors of a symmetric tridiagonal matrix:
  1. $A = \begin{bmatrix} 0 & B^T \\ B & 0 \end{bmatrix}$, $T_{ps} \equiv P^T A P$ is symmetric tridiagonal, $P = [e_1, e_{n+1}, e_2, \ldots, e_n, e_{2n}]$ permutation matrix ("perfect shuffle"). If (α_i, x_i) is an eigenpair of T_{ps}, then $\alpha_i = \pm\sigma_i$, where σ_i is a singular value of B, and $Px_i = \frac{1}{\sqrt{2}} \begin{bmatrix} u_i \\ \pm u_i \end{bmatrix}$,where u_i and v_i are left and right singular vectors of B, respectively.
  2. $T_{BB^T} = BB^T$, $\sigma_B = \sqrt{\lambda_T}$, left singular vectors of B are the ev of T_{BB^T} — *not numerically stable*
  3. $T_{B^TB} = B^TB$, $\sigma_B = \sqrt{\lambda_T}$, right singular vectors of B are the ev of T_{B^TB} — *not numerically stable*

References

J. J. Cuppen:
A divide and conquer method for the symmetric tridiagonal eigen problem,
Numer. Math. **36** (1981), pp. 177–195, 332–345, parts I and II.

M. Gu, S. C. Eisenstadt:
A divide and conquer algorithm for the symmetric tridiagonal eigenproblem,
SIAM J. Matrix Anal. Appl. **16**(1995), 172–191

Lloyd. N. Trefethen, David Bau III:
Numerical Linear Algebra,
SIAM, Philadelphia, 1997

James Demmel:
Applied Numerical Linear Algebra,
SIAM, Philadelphia, 1997.