Direct methods for symmetric eigenvalue problems

Imre Pólik, PhD

McMaster University
School of Computational Engineering and Science

February 4, 2008

Outline

1. Introduction
    - Theoretical background
    - Posing the question
    - Perturbation theory
2. Algorithms
    - Direct and iterative methods
    - Transformation to tridiagonal form
    - Counting the eigenvalues
    - Finding the eigenvectors
3. Eigenvalues of a tridiagonal matrix
    - QR method
    - Divide and conquer
    - Bisection with inverse iteration
    - Jacobi method
4. Conclusion

Theoretical background (A is real, symmetric)

- Eigenvalue: $Ax = \lambda x$
- $A - \lambda I$ is singular
- n eigenvalues (with multiplicities)
- Characteristic polynomial: $p(\lambda) = \det(A - \lambda I)$
- If $Q^T Q = I$ then A and $Q^T AQ$ have the same eigenvalues
- If P is nonsingular then A and $P^T AP$ have the same inertia
- Eigendecomposition ($Av_i = \lambda_i v_i$):

$$A = \sum_{i=1}^{n} \lambda_i v_i v_i^T$$

- The v_is should be orthogonal

Posing the question

- Find all the eigenvalues
- Find some eigenvalues
- Find largest/smallest eigenvalue(s)
- Find largest/smallest magnitude eigenvalue(s)
- Find eigenvalue(s) around a given number
- Find the corresponding eigenvectors
- Find the eigenvalues in $[\alpha, \beta]$
- Count the eigenvalues in $[\alpha, \beta]$

Perturbation theory

- Eigenvalues are well-conditioned (Weyl)

$$|\lambda_i(A) - \lambda_i(A+E)| \leq \|E\|_2$$

- Eigenvectors can be ill-conditioned

$$\begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & \varepsilon \\ 0 & \varepsilon & 1 \end{pmatrix} \qquad \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1+\varepsilon \end{pmatrix}$$

- Problem: distance between eigenvalues
- Problem: small eigenvalues
- Shifting: $\lambda(A - \alpha I) = \lambda(A) - \alpha$

Direct and iterative methods

- Eigenvalues are roots of polynomials
- Every method has to iterate
- Direct
  - transforms the matrix to a simpler form (tridiagonal)
  - finds all the eigenvalues and eigenvectors
  - (almost) always works
  - converges quickly
- Iterative
  - only multiplication by A or A^T
  - does not access the elements of A
  - only a few eigenvectors/values

Tridiagonalization (Hessenberg reduction)

- Householder reflections

$$\underbrace{\begin{pmatrix} 1 & 0 \\ 0 & I - 2uu^T \end{pmatrix}}_{Q} \underbrace{\begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix}}_{A} = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \end{pmatrix}$$

- With this Q, due to symmetry:

$$QAQ = \begin{pmatrix} * & * & 0 & 0 \\ * & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \end{pmatrix}$$

- Cost $\frac{4}{3}n^3$, or $\frac{8}{3}n^3$ if Q is needed for the eigenvectors
- LAPACK: `ssytrd`, Matlab: `hess`

Count the eigenvalues in $[\alpha, \beta)$

- Factorize $A - \alpha I = L_\alpha D_\alpha L_\alpha^T$
- Number of negative eigenvalues in D_α gives the number of eigenvalues of A that are smaller than α
- Factorize $A - \beta I = L_\beta D_\beta L_\beta^T$
- Number of negative eigenvalues in D_β gives the number of eigenvalues of A that are smaller than β
- The difference gives the number of eigenvalues in $[\alpha, \beta)$
- Cost: two LDL factorizations (can exploit structure)

Finding the eigenvectors

- Solve $(A - \lambda I)v = 0$
  - λ is only approximate
  - extremely ill-conditioned system
  - only if λ is an exact eigenvalue
- Inverse iteration
  - repeat

$$w_{i+1} = (A - \alpha I)^{-1} v_i$$
$$v_{i+1} = \frac{w_{i+1}}{\|w_{i+1}\|_2}$$

  - fast convergence
  - can be modified to find the eigenvalues, too

Eigenvalues of a tridiagonal matrix

QR method

- Factor $A_i = Q_i R_i$, set $A_{i+1} = R_i Q_i$, repeat
- Shifting: $A_i - \alpha_i I = Q_i R_i$, $A_{i+1} = R_i Q_i + \alpha_i I$
- If α_i is close to an eigenvalue then convergence is fast (quadratic)
- Fastest method for eigenvalues
- Fastest method for eigenvectors if $n \leq 25$
- Total cost: $\frac{4}{3}n^3 + \mathcal{O}(n^2)$, cubic! convergence
- LAPACK: `xSTEQR`, `xSTERF`, Matlab: `eig`

QR method – Choosing the shift

$$A = \begin{pmatrix} a_1 & c_1 & & & \\ c_1 & a_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & c_{n-2} & a_{n-1} & c_{n-1} \\ & & & c_{n-1} & a_n \end{pmatrix}$$

- Single shift: $\alpha = a_n$
- Wilkinson shift:
  α is the eigenvalue of $\begin{pmatrix} a_{n-1} & c_{n-1} \\ c_{n-1} & a_n \end{pmatrix}$ closest to a_n

Divide and conquer

- Partition the tridiagonal matrix into two halves
- Solve the smaller problems recursively
- Assemble the results
- Cost: $\mathcal{O}(n^{2.3})$ on average
- Fastest method for eigenvectors if $n \geq 25$
- Not easy to implement in a stable way
- Efficiently parallelizable
- LAPACK: `xSTEVD`

Bisection with inverse iteration

- Count the eigenvalues in $[\alpha, \beta)$
- Bisect the interval, count the eigenvalues in the subintervals, then repeat
- LDL^T is too expensive in general $\Rightarrow$
- Transform A to tridiagonal form first
- Cost: $\mathcal{O}(nk)$ for k eigenvalues
- Excellent parallelization
- Only linear convergence (due to bisection)
- Compute eigenvectors with inverse iteration
- LAPACK: `xSTEBZ`, `xSTEIN`

Jacobi method

- Transform A to (almost) diagonal form
- Zero out $a_{jk}, j \neq k$ with Givens rotation

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}^T \begin{pmatrix} A_{jj} & A_{jk} \\ A_{kj} & A_{kk} \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

- Costs $\mathcal{O}(n)$ flops per rotation
- How to choose A_{jk}?

$$\operatorname{off}(A) := \sqrt{\sum_{j<k} A_{jk}^2}$$

$$\operatorname{off}(A')^2 = \operatorname{off}(A)^2 - A_{jk}^2$$

- Always the largest A_{jk}: too expensive
- Row-wise: cheap but still efficient (5-10 sweeps)
- In general Jacobi is too slow, but very accurate

Conclusion

- Transform A to tridiagonal form
- Only eigenvalues: use QR
- Eigenvalues and eigenvectors: use D&C
- Small eigenvalues with high accuracy: use Jacobi
- Eigenvalues in $[\alpha, \beta)$: use bisection
- Most algorithms are available in LAPACK

Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, editors.
Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide.
SIAM, Philadelphia, 2000.

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