Numerical Methods I: Eigenvalues and eigenvectors

Georg Stadler
Courant Institute, NYU
stadler@cims.nyu.edu

Oct 29, 2015
Main reference:
Section 5 in Deuflhard/Hohmann
Overview

Conditioning
Eigenvalues and eigenvectors
How hard are they to find?

For a matrix $A \in \mathbb{R}^{n \times n}$ (potentially symmetric and/or positive—can also be complex), we want to find $\lambda$ and $x$ such that

$$Ax = \lambda x.$$ 

- This is a **nonlinear** problem.
- How **difficult** is this? Eigenvalues are the roots of the characteristic polynomial. Also, any polynomial is the characteristic polynomial of a matrix. Thus, for matrices larger than $4 \times 4$, eigenvalues cannot be computed analytically.
- Must use an **iterative** algorithm.
Consider an algebraically simple eigenvalue $\lambda_0$ of $A$:

$$Ax_0 = \lambda_0 x_0.$$ 

Then, there exists a continuously differentiable map

$$\lambda(\cdot): A \rightarrow \lambda(A)$$

in a neighborhood of $A$ such that $\lambda(A) = \lambda_0$. The derivative is

$$\lambda'(A)C = \frac{(Cx_0, y_0)}{(x_0, y_0)},$$

where $y_0$ is an eigenvector of $A^T$ for the eigenvalue $\lambda_0$. 

Conditioning
Conditioning

Compute norm of $\lambda'(A)$ as linear mapping that maps

$$C \mapsto \frac{(Cx_0, y_0)}{(x_0, y_0)}.$$

Use as norm for $C$ the norm induced by the Euclidean norm:

$$\|\lambda'(A)\| = \frac{\|x_0\|\|y_0\|}{|(x_0, y_0)|},$$

i.e., the inverse cosine of the angle between $x_0, y_0$. 

Theorem: The absolute and relative condition numbers for computing a simple eigenvalue $\lambda_0$ are

$$\kappa_{\text{abs}} = \| \lambda'(A) \| = \frac{1}{| \cos(\angle(x_0, y_0)) |},$$

and

$$\kappa_{\text{rel}} = \frac{\| A \|}{| \lambda_0 \cos(\angle(x_0, y_0)) |}.$$

In particular, for normal matrices, $\kappa_{\text{abs}} = 1$. Note that finding non-simple eigenvalues is ill-posed (but can still be done).
Overview

Power method and variants
The power method

Choose starting point \( \mathbf{x} \) and iterate

\[
\mathbf{x}^{k+1} := A\mathbf{x}^k, \quad \mathbf{x}^{k+1} = \frac{\mathbf{x}^{k+1}}{\|\mathbf{x}^{k+1}\|}
\]

Idea: Eigenvector corresponding to largest (in absolute norm) eigenvalue will start dominating, i.e., \( \mathbf{x}^k \) converges to eigenvector for largest eigenvalue \( \lambda \).

- Convergence speed depends on eigenvalues
- Only finds largest eigenvalue \( \lambda_{\text{max}} = \mathbf{x}^T A \mathbf{x} \) upon convergence
The power method—variants

In reverse power method: Having an estimate $\tilde{\lambda}$ for an eigenvalue $\lambda_i$, consider the $(A - \tilde{\lambda}I)^{-1}$ which has the eigenvalues

$$(\lambda_i - \tilde{\lambda})^{-1}, \quad i = 1, \ldots, n.$$

Consider the inverse power iteration

$$(A - \tilde{\lambda}I)x^{k+1} = x^k, \quad x^{k+1} = x^{k+1}/\|x^{k+1}\|$$

- Required matrix-solve in every iteration
- Convergence speed depends on how close $\tilde{\lambda}$ is to $\lambda_i$. 
The power method—variants

Rayleigh quotient iteration: Accelerated version of the inverse power method using of changing shifts:

- Choose starting vector $\mathbf{x}^0$ with $\|\mathbf{x}^0\| = 1$. Compute $\lambda_0 = (\mathbf{x}^0)^T A \mathbf{x}^0$.
- For $i = 0, 1, \ldots$ do

  $$(A - \bar{\lambda}_k I) \mathbf{x}^{k+1} = \mathbf{x}^k, \quad \mathbf{x}^{k+1} = \mathbf{x}^{k+1} / \|\mathbf{x}^{k+1}\|.$$  

- Compute $\lambda_{k+1} = (\mathbf{x}^{k+1})^T A \mathbf{x}^{k+1}$, and go back.
Overview

The QR algorithm
The QR algorithm for finding eigenvalues is as follows ($A^0 := A$), and for $k = 0, 1, \ldots$:

- Compute QR decomposition of $A^k$, i.e., $A^k = QR$.
- $A^{k+1} := RQ$, $k := k + 1$ and go back.

Why should that converge to something useful?
The QR algorithm

- **Similarity transformations** do not change the eigenvalues, i.e., if $B$ is invertible, then

$$A \text{ and } P^{-1}AP$$

have the same eigenvalues.

- $QA^{k+1}Q^T = QRQQ^T = QR = A^k$, i.e., the iterates $A^k$ in the QR algorithm have the same eigenvalues.

- The algorithm is closely related to the Rayleigh coefficient method.

- The algorithms is expensive (QR-decomposition is $O(n^3)$).

- Convergence can be slow.
QR algorithm and Hessenberg matrices

Idea: Find a matrix format that is preserved in the QR-algorithm. Hessenberg matrices $H$ are matrices for which $H_{i,j} = 0$ if $i > j + 1$.

- Hessenberg matrices remain Hessenberg in the QR algorithm.
- An iteration of the QR-algorithm with a Hessenberg matrix requires $O(n^2)$ flops.

Algorithm:

1. Use Givens rotations to transfer $A$ into Hessenberg form. Use transpose operations on right hand side (similarity transformation).
2. Use QR algorithm for the Hessenberg matrix.
The QR algorithm for symmetric matrices
QR algorithm for symmetric matrices

Let’s consider symmetric matrices $A$. Then the Hessenberg form (after application of Given rotations from both sides) is tridiagonal. Theorem: For a symmetric matrix with distinct eigenvalues

$$|\lambda_1| > \ldots |\lambda_n| > 0, \quad \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$$

holds:

1. $\lim_{k \to \infty} Q_k = I$,
2. $\lim_{k \to \infty} R_k = \Lambda$,
3. $a_{i,j}^k = O \left( \left| \frac{\lambda_i}{\lambda_j} \right|^k \right)$ for $i > j$. 
The method also converges in the presence of multiple eigenvalues. Only when $\lambda_i = -\lambda_{i+1}$, the corresponding block does not become diagonal.

One can speed up the method by introducing a shift operator:

\[ A_k - \sigma_k I = QR, \]
\[ A_{k+1} = RQ + \sigma_k I. \]

Again, the shift can be updated during the iteration.
QR algorithm for symmetric matrices

Summary

Complexity: Convert to Hessenberg form using Givens rotations: $4/3n^3$ flops; each QR iteration: $O(n^2)$ flops. Overall, convergence is dominated by the reduction to tridiagonal form.

This method finds all eigenvalues (of a symmetric matrix).

The corresponding eigenvectors can be found from the algorithm as well.