Numerical Methods I
Eigenvalue Problems

Aleksandar Donev
Courant Institute, NYU[1]
donev@courant.nyu.edu

[1]MATH-GA 2011.003 / CSCI-GA 2945.003, Fall 2014

October 2nd, 2014

Outline

Eigenvalue Decomposition

- For a square matrix $\mathbf{A} \in \mathbb{C}^{n \times n}$, there exists at least one λ such that

$$\mathbf{A}\mathbf{x} = \lambda \mathbf{x} \quad \Rightarrow \quad (\mathbf{A} - \lambda \mathbf{I})\,\mathbf{y} = \mathbf{0}$$

- Putting the **eigenvectors** $\mathbf{x}_j$ as columns in a matrix $\mathbf{X}$, and the **eigenvalues** λ_j on the diagonal of a diagonal matrix $\mathbf{\Lambda}$, we get

$$\mathbf{A}\mathbf{X} = \mathbf{X}\mathbf{\Lambda}.$$

- A matrix is **non-defective** or **diagonalizable** if there exist n **linearly independent eigenvectors**, i.e., if the matrix $\mathbf{X}$ is invertible:

$$\mathbf{X}^{-1}\mathbf{A}\mathbf{X} = \mathbf{\Lambda}$$
$$\mathbf{A} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1}.$$

- The transformation from $\mathbf{A}$ to $\mathbf{\Lambda} = \mathbf{X}^{-1}\mathbf{A}\mathbf{X}$ is called a **similarity transformation** and it preserves the eigenspace.

Unitarily Diagonalizable Matrices

- A matrix is **unitarily diagonalizable** if there exist n linearly independent **orthogonal eigenvectors**, i.e., if the matrix $\mathbf{X}$ can be chosen to be **unitary (orthogonal)**, $\mathbf{X} \equiv \mathbf{U}$, where $\mathbf{U}^{-1} = \mathbf{U}^{\star}$:

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\star}.$$

  Note that unitary matrices generalize orthogonal matrices to the complex domain, so we use **adjoints** (conjugate transposes) instead of transposes throughout.
- Theorem: A matrix is unitarily diagonalizable iff it is **normal**, i.e., it commutes with its adjoint:

$$\mathbf{A}^{\star}\mathbf{A} = \mathbf{A}\mathbf{A}^{\star}.$$

- Theorem: **Hermitian (symmetric) matrices**, $\mathbf{A}^{\star} = \mathbf{A}$, are unitarily diagonalizable and have **real eigenvalues**.

Left Eigenvectors

- The usual eigenvectors are more precisely called **right eigenvectors**. There is also **left eigenvector** corresponding to a given eigenvalue λ

$$\mathbf{y}^\star \mathbf{A} = \lambda \mathbf{y}^\star \quad \Rightarrow \quad \mathbf{A}^\star \mathbf{y} = \lambda \mathbf{y}.$$

$$\mathbf{Y}^\star \mathbf{A} = \mathbf{\Lambda} \mathbf{Y}^\star$$

- For a matrix that is diagonalizable, observe that

$$\mathbf{Y}^\star = \mathbf{X}^{-1}$$

  and so the left eigenvectors provide no new information.
- For **unitarily diagonalizable** matrices, $\mathbf{Y} = \left(\mathbf{X}^{-1}\right)^\star = \mathbf{U}$, so that the **left and right eigenvectors coincide**.

Non-diagonalizable Matrices

- For matrices that are not diagonalizable, one can use **Jordan form factorizations**, or, more relevant to numerical mathematics, the **Schur factorization** (decomposition):

$$\mathbf{A} = \mathbf{U}\mathbf{T}\mathbf{U}^\star,$$

  where $\mathbf{T}$ is **upper-triangular**.
- The eigenvalues are on the diagonal of $\mathbf{T}$.
- Note: Observe that $\mathbf{A}^\star = (\mathbf{U}\mathbf{T}\mathbf{U}^\star)^\star = \mathbf{U}\mathbf{T}^\star\mathbf{U}^\star$ so for Hermitian matrices $\mathbf{T} = \mathbf{T}^\star$ is real diagonal.
- An important property / use of eigenvalues:

$$\mathbf{A}^n = (\mathbf{U}\mathbf{T}\mathbf{U}^\star)(\mathbf{U}\mathbf{T}\mathbf{U}^\star)\cdots(\mathbf{U}\mathbf{T}\mathbf{U}^\star) = \mathbf{U}\mathbf{T}(\mathbf{U}^\star\mathbf{U})\mathbf{T}(\mathbf{U}^\star\mathbf{U})\cdots\mathbf{T}\mathbf{U}^\star$$

$$\mathbf{A}^n = \mathbf{U}\mathbf{T}^n\mathbf{U}^\star$$

Sensitivity of Eigenvalues

- Now consider a perturbation of a diagonalizable matrix $\delta\mathbf{A}$ and see how perturbed the similar matrix becomes:

$$\mathbf{X}^{-1}\left(\mathbf{A}+\delta\mathbf{A}\right)\mathbf{X}=\mathbf{\Lambda}+\delta\mathbf{\Lambda} \quad \Rightarrow$$

$$\delta\mathbf{\Lambda}=\mathbf{X}^{-1}\left(\delta\mathbf{A}\right)\mathbf{X} \quad \Rightarrow$$

$$\|\delta\mathbf{\Lambda}\| \leq \left\|\mathbf{X}^{-1}\right\| \|\delta\mathbf{A}\| \|\mathbf{X}\| = \kappa\left(\mathbf{X}\right)\|\delta\mathbf{A}\|$$

- Conclusion: The conditioning of the eigenvalue problem is related to the **conditioning of the matrix of eigenvectors**.
- If $\mathbf{X}$ is unitary then $\|\mathbf{X}\|_2 = 1$ (from now on we exclusively work with the 2-norm): **Unitarily diagonalizable matrices are always perfectly conditioned!**
- Warning: The **absolute error** in all eigenvalues is of the same order, meaning that the **relative error will be very large** for the smallest eigenvalues.

Sensitivity of Individual Eigenvalues

PERTURBATION ANALYSIS

$$A x = \lambda x$$

$$A(\varepsilon) = A + \varepsilon \cdot \delta A, \qquad \varepsilon << 1$$

DIFFERENTIATE W.R.T ε

$$A(\varepsilon) \cdot x(\varepsilon) = \lambda(\varepsilon) \cdot x(\varepsilon)$$

PRE-MULTIPLY BY y^*

$$\rightarrow (\delta A) \cdot x + A \cdot x' = \lambda' x + \lambda \cdot x'$$

$$y^*(\delta A) x + (y^* A) x' =$$

EQUAL

$$\lambda'(y \cdot x) + (\lambda y^*) x' \Rightarrow$$

$$y^* (\delta A) x = (y \cdot x) \lambda' \Rightarrow$$

$$\lambda' = \frac{y^* (\delta A) x}{(y \cdot x)}$$

← SENSITIVITY OF λ

Sensitivity of Individual Eigenvalues

$$\delta\lambda \approx \frac{\mathbf{y}^\star (\delta\mathbf{A})\,\mathbf{x}}{\mathbf{y}^\star\mathbf{x}}$$

- Recalling the Cauchy-Schwartz inequality:

$$|\mathbf{y}\cdot\mathbf{x}| = \|\mathbf{x}\|\,\|\mathbf{y}\|\cos\theta_{xy} \le \|\mathbf{x}\|\,\|\mathbf{y}\|$$

$$|\delta\lambda| \le \frac{\|\mathbf{x}\|\,\|\delta\mathbf{A}\|\,\|\mathbf{y}\|}{\|\mathbf{x}\|\,\|\mathbf{y}\|\cos\theta_{xy}} = \frac{\|\delta\mathbf{A}\|}{\cos\theta_{xy}}$$

- Defining a conditioning number for a given eigenvalue

$$\kappa(\lambda,\mathbf{A}) = \sup_{\delta\mathbf{A}} \frac{|\delta\lambda|}{\|\delta\mathbf{A}\|} = \frac{1}{\cos\theta_{xy}}$$

- For **unitarily diagonalizable** matrices $\mathbf{y}=\mathbf{x}$ and thus $\kappa(\lambda,\mathbf{A})=1$: **perfectly conditioned**!

Sensitivity of Eigenvectors

- A priori estimate: The conditioning number for the eigenvector itself depends on the **separation between the eigenvalues**

$$\kappa(\mathbf{x}, \mathbf{A}) = \left(\min_j |\lambda - \lambda_j| \right)^{-1}$$

- This indicates that **multiple eigenvalues require care**. Even for Hermitian matrices eigenvectors are hard to compute.
- If there is a defective (non-diagonalizable) matrix with eigenvalue for which the difference between the algebraic and geometric multiplicities is $d > 0$, then

$$\delta\lambda \sim \|\delta\mathbf{A}\|^{1/(1+d)},$$

which means the conditioning number is infinite: **Defective eigenvalues are very badly conditioned**.

The need for iterative algorithms

- The eigenvalues are roots of the **characteristic polynomial** of **A**, which is generally of order n.
- According to Abel's theorem, there is no closed-form (rational) solution for $n \geq 5$.
- **All eigenvalue algorithms must be iterative!**
  This is a fundamental difference from, example, linear solvers.
- There is an important distinction between iterative methods to:
  - Compute **all eigenvalues** (similarity transformations).
  - Compute **only one or a few eigenvalues**, typically the smallest or the largest one (power-like methods).
- Bounds on eigenvalues are important, e.g., Courant-Fisher theorem for the **Rayleigh quotient**:

$$\min \lambda \leq r_A(\mathbf{x}) = \frac{\mathbf{x}^\star \mathbf{A}\mathbf{x}}{\mathbf{x}^\star \mathbf{x}} \leq \max \lambda$$

The Power Method

- Recall that for a diagonalizable matrix

$$\mathbf{A}^n = \mathbf{X}\mathbf{\Lambda}^n\mathbf{X}^{-1}$$

and assume $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \cdots |\lambda_n|$ and that the columns of $\mathbf{X}$ are normalized, $\|\mathbf{x}_j\| = 1$.

- Any **initial guess** vector $\mathbf{q}_0$ can be represented in the linear basis formed by the eigenvectors

$$\mathbf{q}_0 = \mathbf{X}\mathbf{a}$$

- Recall iterative methods for linear systems: **Multiply a vector with the matrix A** many times:

$$\mathbf{q}_{k+1} = \mathbf{A}\mathbf{q}_k$$

$$\mathbf{q}_n = \mathbf{A}^n\mathbf{q}_0 = \left(\mathbf{X}\mathbf{\Lambda}^n\mathbf{X}^{-1}\right)\mathbf{X}\mathbf{a} = \mathbf{X}\left(\mathbf{\Lambda}^n\mathbf{a}\right)$$

Power Method

- As $n \to \infty$, the **eigenvalue of largest modulus** λ_0 will dominate,

$$\mathbf{\Lambda}^n = \lambda_1^n \text{Diag}\left\{1, \left(\frac{\lambda_2}{\lambda_1}\right)^n, \dots\right\} \to \lambda_1^n \text{Diag}\left\{1, 0, \dots, 0\right\}$$

$$\mathbf{q}_n = \mathbf{X}\left(\mathbf{\Lambda}^n \mathbf{a}\right) \to \lambda_1^n \mathbf{X} \begin{bmatrix} a_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \lambda_1^n \mathbf{x}_1$$

- Therefore the **normalized iterates** converge to the eigenvector:

$$\tilde{\mathbf{q}}_n = \frac{\mathbf{q}_n}{\|\mathbf{q}_n\|} \to \mathbf{x}_1$$

- The **Rayleigh quotient** converges to the eigenvalue:

$$r_A\left(\mathbf{q}_n\right) = \frac{\mathbf{q}_n^\star \mathbf{A} \mathbf{q}_n}{\mathbf{q}_n \cdot \mathbf{q}_n} = \tilde{\mathbf{q}}_n^\star \mathbf{A} \tilde{\mathbf{q}}_n \to \lambda_1$$

An alternative derivation

$$q_0 = \sum_n a_i \cdot x_i$$

$$q_n = A^n q_0 = \sum_n a_i (A^n x_i)$$

$$= \sum a_i (\lambda_i^n x_i) =$$

$$= \sum a_i \lambda_1^n \left(\frac{\lambda_i}{\lambda_1}\right)^n \cdot x_i \xrightarrow[n\to\infty]{as} a_1 \lambda_1^n x_1 + O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^n\right)$$

$$\Rightarrow \boxed{\tilde{q}_n \to x_1 + O(\varepsilon)\, x_2 + O(\varepsilon) x_3 + \dots \quad \text{where} \quad \varepsilon = \left|\frac{\lambda_2}{\lambda_1}\right|^n}$$

Now $\tilde{q}_n^* A \tilde{q}_n = x_1^* A x_1 + O(\varepsilon^2) \sum_{i,j \neq 1} x_i^* A x_j$

$$\Rightarrow \boxed{\hat{\lambda}_n'' = \lambda_1 + O(\varepsilon^2)}$$

Power Method Implementation

Start with an initial guess $\mathbf{q}_0$, and then iterate:

1. Compute **matrix-vector product** and normalize it:

$$\mathbf{q}_k = \frac{\mathbf{A}\mathbf{q}_{k-1}}{\|\mathbf{A}\mathbf{q}_{k-1}\|}$$

2. Use Raleigh quotient to obtain **eigenvalue estimate**:

$$\hat{\lambda}_k = \mathbf{q}_k^\star \mathbf{A}\mathbf{q}_k$$

3. **Test for convergence**: Evaluate the residual

$$\mathbf{r}_k = \mathbf{A}\mathbf{q}_k - \hat{\lambda}_k \mathbf{q}_k$$

and terminate if the error estimate is small enough:

$$\left|\lambda_1 - \hat{\lambda}_k\right| \approx \frac{\|\mathbf{r}_k\|}{\cos\theta_{xy}} < \varepsilon$$

Convergence Estimates

- The normalized iterates converge to the eigenvector **linearly**:

$$\|\mathbf{q}_k - (\pm\mathbf{x}_1)\| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)$$

- If **A** is normal the eigenvalue estimate converges a bit faster but still linearly

$$\left\|\hat{\lambda}_k - \lambda_1\right\| \sim O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right)$$

- The power method is fast when the **dominant eigenvalue is well-separated from the rest** (even if it is degenerate).
- This conclusion is rather general for all iterative methods: Convergence is good for **well-separated** eigenvalues, bad otherwise.
- The power method is typically too slow to be used in practice and there are more sophisticated alternatives (**Lanczos/Arnoldi iteration**).

Inverse Power Iteration

- Observe that applying the power method to $\mathbf{A}^{-1}$ will find the largest of λ_j^{-1}, i.e., the **smallest eigenvalue** (by modulus).
- If we have an **eigenvalue estimate** $\mu \approx \lambda$, then doing the power method for the matrix

$$(\mathbf{A} - \mu\mathbf{I})^{-1}$$

will give the **eigenvalue closest to** μ**.**

- Convergence will be faster if μ is much closer to λ then to other eigenvalues.
- Recall that in practice $(\mathbf{A} - \mu\mathbf{I})^{-1}\,\mathbf{q}$ is computed by **solving a linear system**, not matrix inversion (one can **reuse an** LU **factorization**)!
- Finally, if we have an estimate of **both** the eigenvalue and the eigenvector, we can use **Rayleigh Quotient Iteration** (see homework).

Estimating all eigenvalues / eigenvectors

- Iterative methods akin the power method are not suitable for estimating all eigenvalues.
- Basic idea: Build a sequence of matrices $\mathbf{A}_k$ that all share eigenvalues with $\mathbf{A}$ via **similarity transformations**:

$$\mathbf{A}_{k+1} = \mathbf{P}^{-1}\mathbf{A}_k\mathbf{P}, \text{ starting from } \mathbf{A}_1 = \mathbf{A}.$$

- A numerically stable and good way to do this is to use the QR factorization:

$$\mathbf{A}_k = \mathbf{Q}_{k+1}\mathbf{R}_{k+1}$$

$$\mathbf{A}_{k+1} = \mathbf{Q}_{k+1}^{-1}\mathbf{A}_k\mathbf{Q}_{k+1} = \left(\mathbf{Q}_{k+1}^{-1}\mathbf{Q}_{k+1}\right)\mathbf{R}_{k+1}\mathbf{Q}_{k+1} = \mathbf{R}_{k+1}\mathbf{Q}_{k+1}.$$

- Note that the fact the $\mathbf{Q}$'s are orthogonal is crucial to keep the **conditioning** from getting worse.

The basic QR method

- The behavior of the QR iteration can be understood most transparently as follows [following Trefethen and Bau]:
- Observation: The range of the matrix $\mathbf{A}^k$ converges to the space spanned by the eigenvectors of $\mathbf{A}$,
  with the eigenvectors corresponding to the largest eigenvalues dominating as $k \to \infty$ (so this is ill-conditioned).
- Recall: The columns of $\mathbf{Q}$ in $\mathbf{A} = \mathbf{QR}$ form an **orthonormal basis** for the range of $\mathbf{A}$.
- Idea: Form a **well-conditioned basis for the eigenspace** of $\mathbf{A}$ by factorizing:

$$\mathbf{A}^k = \tilde{\mathbf{Q}}_k \tilde{\mathbf{R}}_k$$

  and then calculate

$$\mathbf{A}_k = \tilde{\mathbf{Q}}_k^{-1} \mathbf{A} \tilde{\mathbf{Q}}_k = \tilde{\mathbf{Q}}_k^{\star} \mathbf{A} \tilde{\mathbf{Q}}_k.$$

- It is not too hard to show that this produces the **same sequence** of matrices $\mathbf{A}_k$ as the QR algorithm.

Why the QR algorithm works

- Summary: The columns of $\tilde{\mathbf{Q}}_k$ converge to the eigenvectors, and

$$\mathbf{A}_k = \tilde{\mathbf{Q}}_k^\star \mathbf{A} \tilde{\mathbf{Q}}_k.$$

- We can recognize the above as a matrix of Rayleigh quotients, which for diagonalizable matrices

$$(\mathbf{A}_k)_{ij} = \tilde{\mathbf{q}}_i^\star \mathbf{A} \tilde{\mathbf{q}}_j \to \lambda_i \delta_{ij} = \begin{cases} \lambda_i & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

  showing that (under suitable assumptions):

$$\mathbf{A}_k \to \mathbf{\Lambda}$$

- It can also be shown that

$$\tilde{\mathbf{Q}}_k = \mathbf{Q}_1 \mathbf{Q}_2 \cdots \mathbf{Q}_k \to \mathbf{X}$$

More on QR algorithm

- The convergence of the basic QR algorithm is closely related to that of the power method: It is only fast if all eigenvalues are **well-separated**.
- For more general (non-diagonalizable) matrices in complex arithmetic, the algorithm converges to the **Schur decomposition** $\mathbf{A} = \mathbf{UTU}^\star$,

$$\mathbf{A}_k \to \mathbf{T} \text{ and } \tilde{\mathbf{Q}}_k \to \mathbf{U}.$$

- It is possible to implement the algorithm entirely using real arithmetic (no complex numbers).
- There are several key improvements to the basic method that make this work in practice: **Hessenberg matrices** for faster QR factorization, **shifts** and **deflation** for acceleration.
- There are other sophisticated algorithms as well, such as the **divide-and-conquer algorithm**, and the best are implemented in the library LAPACK (MATLAB).

Eigenvalues in MATLAB

- In MATLAB, sophisticated variants of the **QR algorithm** (LAPACK library) are implemented in the function *eig*:

$$\Lambda = eig(A)$$

$$[X, \Lambda] = eig(A)$$

- For large or sparse matrices, iterative methods based on the **Arnoldi iteration** (ARPACK library), can be used to obtain a few of the largest/smallest/closest-to-μ eigenvalues:

$$\Lambda = eigs(A, n_{eigs})$$

$$[X, \Lambda] = eigs(A, n_{eigs})$$

- The **Schur decomposition** is provided by $[U, T] = schur(A)$.

Conclusions/Summary

- Eigenvalues are **well-conditioned** for **unitarily diagonalizable matrices** (includes Hermitian matrices), but ill-conditioned for nearly non-diagonalizable matrices.
- Eigenvectors are **well-conditioned** only when e**igenvalues are well-separated**.
- Eigenvalue algorithms are **always iterative**.
- The **power method** and its variants can be used to find the **largest or smallest eigenvalue**, and they converge fast if there is a **large separation** between the target eigenvalue and nearby ones.
- Estimating **all eigenvalues and/or eigenvectors** can be done by combining the power method with QR factorizations.
- MATLAB has high-quality implementations of sophisticated variants of these algorithms.