SI231b: Matrix Computations

Lecture 13: Eigenvalue Computations

Yue Qiu

qiuyue@shanghaitech.edu.cn

School of Information Science and Technology

ShanghaiTech University

Oct. 27, 2020

Factorize a matrix to a form in which eigenvalues are explicitly displayed

- Diagonalization, $\mathbf{A} = \mathbf{V}\Lambda\mathbf{V}^{-1}$, exists if and only if $\mathbf{A}$ is nondefective.
- Schur decompositon, $\mathbf{A} = \mathbf{Q}\mathbf{T}\mathbf{Q}^H$ always exists.
- Jordan canonical form, $\mathbf{A} = \mathbf{S}\mathbf{J}\mathbf{S}^{-1}$ always exists (will not be introduced in our lecture), where

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_1 & & & \\ & \mathbf{J}_2 & & \\ & & \ddots & \\ & & & \mathbf{J}_k \end{bmatrix}$$

with

$$\mathbf{J}_i = \begin{bmatrix} \lambda_i & & & \\ & \lambda_i & & \\ & & \ddots & \\ & & & \lambda_i \end{bmatrix}, \quad \text{or} \quad \mathbf{J}_i = \begin{bmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix}$$

Outline

Some Facts About Eigenvalues

- Eigenvalues of Hermitian matrices are real
- Eigenvalues of real symmetric matrices are real
- Eigenvectors of real symmetric matrices are also real
- Complex eigenvalues of real matrices appear in conjugate pair.
  - For $\mathbf{A} \in \mathbb{R}^{n \times n}$, if $(\lambda, \mathbf{v})$ is an eigenpair, then also $(\lambda^*, \mathbf{v}^*)$
- Skew-Hermitian matrices ($\mathbf{A} = -\mathbf{A}^H$) have only pure imaginary eigenvalues
- Hermitian/real symmetric matricres are diagonalizable.

The Largest Eigenvalue and Associated Eigenvector

Let $\mathbf{A} \in \mathbb{C}^{n \times n}$ be diagonalizable, i.e., $\mathbf{A} = \mathbf{V}\Lambda\mathbf{V}^{-1}$ with $\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \end{bmatrix}$, and $\Lambda = \mathrm{diag}(\lambda_1,\ \lambda_2,\ \cdots,\ \lambda_n)$. Assume that

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|.$$

The following iteration generates a sequence of $(\lambda^{(k)},\ \mathbf{q}^{(k)})$ that converges to $(\lambda_1,\ \mathbf{v}_1)$.

Power Iteration:

```
random selection q^(0) ∈ C^n
for k = 1, 2, ···
        z^(k) = A q^(k-1)
        q^(k) = z^(k) / ||z^(k)||_2
        λ^(k) = (q^(k))^H A q^(k)
end
```

The Power Iteration can only compute the largest eigenvalue and associated eigenvector with convergence rate

- $|\lambda^{(k)} - \lambda_1| = \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)$
- $\|\mathbf{q}^{(k)} - \mathbf{v}_1\| = \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)$
- can have slow convergence when λ_2 is close to λ_1 in magnititude, i.e., $\left|\frac{\lambda_2}{\lambda_1}\right|$ is close to 1.
- convergence can be made faster by using a shift μ with

  $$\left|\frac{\lambda_1 - \mu}{\mu - \lambda_j}\right| < \left|\frac{\lambda_2}{\lambda_1}\right|,$$

  together with *Inverse Iteration*. Here λ_1 and λ_j are the closest and second closest eigenvalues to μ.

Suppose μ is not an eigenvaue of $\mathbf{A}$, the inverse iteration is given by

Inverse Iteration:

```
random selection q^(0) ∈ C^n
for k = 1, 2, ···
        z = (A − μI)^{-1} q^(k−1)        solve (A − μI)z = q^(k−1)
        q^(k) = z / ||z||_2
        λ^(k) = (q^(k))^H A q^(k)
end
```

$$
\begin{aligned}
&\texttt{random selection } \mathbf{q}^{(0)} \in \mathbb{C}^n \\
&\texttt{for } k = 1,\ 2,\ \cdots \\
&\qquad \mathbf{z} = (\mathbf{A} - \mu\mathbf{I})^{-1}\mathbf{q}^{(k-1)} \qquad \texttt{solve } (\mathbf{A} - \mu\mathbf{I})\mathbf{z} = \mathbf{q}^{(k-1)} \\
&\qquad \mathbf{q}^{(k)} = \frac{\mathbf{z}}{\|\mathbf{z}\|_2} \\
&\qquad \lambda^{(k)} = (\mathbf{q}^{(k)})^H \mathbf{A}\mathbf{q}^{(k)} \\
&\texttt{end}
\end{aligned}
$$

- compute the eigenvalue closest to μ
- convergence rate

$$\left|\frac{\mu - \lambda_j}{\mu - \lambda_k}\right|$$

where λ_j and λ_k are the closest and second closest eigenvalues to μ.

Efficiency per iteration vs Number of iterations?

QR Iteration:

$$\mathbf{A}^{(0)} = \mathbf{A}$$

```
for k = 1, 2, ...
    Q^(k) R^(k) = A^(k-1)    QR factorization of A^(k-1)
    A^(k) = R^(k) Q^(k)
end
```

$\mathbf{A}^{(0)} = \mathbf{A}$

for $k = 1,\ 2,\ \cdots$

$\quad \mathbf{Q}^{(k)}\mathbf{R}^{(k)} = \mathbf{A}^{(k-1)}$ QR factorization of $\mathbf{A}^{(k-1)}$

$\quad \mathbf{A}^{(k)} = \mathbf{R}^{(k)}\mathbf{Q}^{(k)}$

end

Facts:

- $\mathbf{A}^{(k)}$ is similar to $\mathbf{A}$ (why?)
- Eigenvalues of $\mathbf{A}^{(k)}$ should be easier to compute than that of $\mathbf{A}$.
- $\mathbf{A}^{(k)}$ should converge fast (expected) to a form whose eigenvalues are easily computed.

Challenges of QR Iteration

For an $n \times n$ matrix $\mathbf{A}$, each iteration requires $\mathcal{O}(n^3)$ flops to compute the QR factorization.

- too computational expensive!

Motivation: perform similarity transform $\mathbf{A}$ to an upper Hessenberg form (zeros below the first subdiagonal), i.e., $\mathbf{Q}^H\mathbf{A}\mathbf{Q} = \mathbf{H}$ where

$$\mathbf{H} = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \end{bmatrix}$$

Advantage: QR factorization of an upper Hessenberg matrix requires $\mathcal{O}(n^2)$ flops (why?).

QR Iteration with Hessenberg Reduction:

$$
\begin{aligned}
&\mathbf{A} = \mathbf{Q}^H \mathbf{H} \mathbf{Q}\,, \ \ \mathbf{A}^{(0)} = \mathbf{H} \\
&\texttt{for} \ \ k = 1, \ 2, \ \cdots \\
&\qquad \mathbf{Q}^{(k)} \mathbf{R}^{(k)} = \mathbf{A}^{(k-1)} \quad \texttt{QR factorization of} \ \ \mathbf{A}^{(k-1)} \\
&\qquad \mathbf{A}^{(k)} = \mathbf{R}^{(k)} \mathbf{Q}^{(k)} \\
&\texttt{end}
\end{aligned}
$$

Key: $\mathbf{A}^{(k)}$ is of upper Hessenberg form (how to preserve?)

- by using Givens rotations to compute the QR factorization (how to prove?)

Benefit: $\mathcal{O}(n^2)$ flops for QR factorization.

For an $n \times n$ matrix $\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \end{bmatrix}$.

A Naive Try

Let $\mathbf{Q}_1$ be the Householder reflection matrix that reflects $\mathbf{a}_1$ to $-\text{sign}(\mathbf{a}_1(1))\|\mathbf{a}_1\|_2\mathbf{e}_1$,

$$\mathbf{A} = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \rightarrow \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & \times & \times & \times \\ & \times & \times & \times \end{bmatrix}}_{\mathbf{Q}_1\mathbf{A}} \rightarrow \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ {\color{red}\times} & \times & \times & \times \\ {\color{red}\times} & \times & \times & \times \\ {\color{red}\times} & \times & \times & \times \end{bmatrix}}_{\mathbf{Q}_1\mathbf{A}\mathbf{Q}_1^H}$$

Mission failed!

Less Ambitious Try

Let $\tilde{\mathbf{a}}_1 = \mathbf{A}(2:n,1)$ and $\mathbf{Q}_1$ be the Householder reflection matrix that reflects $\tilde{\mathbf{a}}_1$ to $-\text{sign}(\tilde{\mathbf{a}}_1(1))\|\tilde{\mathbf{a}}_1\|_2\mathbf{e}_1$,

$$\mathbf{A} = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \rightarrow \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & \times & \times & \times \end{bmatrix}}_{\mathbf{Q}_1\mathbf{A}} \rightarrow \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & \times & \times & \times \end{bmatrix}}_{\mathbf{Q}_1\mathbf{A}\mathbf{Q}_1^H}$$

Repeat the above procedure to the 2nd column of $\mathbf{Q}_1\mathbf{A}\mathbf{Q}_1^H \cdots$

Given an $n \times n$ matrix $\mathbf{A}$, the following algorithm reduces $\mathbf{A}$ to an upper Hessenberg form.

Hessenberg Reduction:

for $k = 1 : n-2$

$\quad \mathbf{x} = \mathbf{A}(\mathtt{k+1:n,\ k})$

$\quad \mathbf{v}_k = \mathtt{sign}(\mathbf{x}(1))\|\mathbf{x}\|_2\mathbf{e}_1 + \mathbf{x}$

$\quad \mathbf{v}_k = \dfrac{\mathbf{v}_k}{\|\mathbf{v}_k\|_2}$

$\quad \mathbf{A}(k+1:n, k:n) = \mathbf{A}(k+1:n, k:n) - 2\mathbf{v}_k(\mathbf{v}_k^H\mathbf{A}(k+1:n, k:n))$

$\quad \mathbf{A}(1:n, k+1:n) = \mathbf{A}(1:n, k+1:n) - 2(\mathbf{A}(1:n, k+1:n)\mathbf{v}_k)\mathbf{v}_k^H$

end

Readings

You are supposed to read

- Lloyd N. Trefethen and David Bau III. *Numerical Linear Algebra*, SIAM, 1997.

  Lecture 25, 26