SI231b: Matrix Computations

Lecture 17: QR Iteration for Eigenvalue Computations

Yue Qiu

qiuyue@shanghaitech.edu.cn

School of Information Science and Technology

ShanghaiTech University

Nov. 11, 2021

Unnormalized Simultaneous Iteration

Define $\mathrm{V}^{(0)}$ to be the $n \times r$ matrix,

$$\mathrm{V}^{(0)} = \begin{bmatrix} v_1^{(0)} & v_2^{(0)} & \cdots & v_r^{(0)} \end{bmatrix}.$$

After k steps of applying A , we obtain

$$\mathrm{V}^{(k)} = \mathrm{A}^k \mathrm{V}^{(0)} = \begin{bmatrix} v_1^{(k)} & v_2^{(k)} & \cdots & v_r^{(k)} \end{bmatrix}.$$

Assume

1. The leading $r+1$ eigenvalues are distinct in absolute value;

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_r| > |\lambda_{r+1}| \geq |\lambda_{r+2}| \geq \cdots |\lambda_n|$$

2. All the leading principle sub-matrices $\mathrm{Q}^T \mathrm{V}^{(0)}$ are nonsingular.
   - Q is the matrix with $\mathrm{q}_1, \mathrm{q}_2, \cdots, \mathrm{q}_r$ as columns;
   - $\mathrm{q}_1, \mathrm{q}_2, \cdots, \mathrm{q}_r$ are eigenvectors associated with eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_r$.

```
choose V^(0) with r linear independent columns
for k = 1, 2, ···
    V^(k) = AV^(k-1)
    Q^(k)R^(k) = V^(k)   reduced QR factorization
end
```

Under the assumptions, we have as $k \to \infty$,

- For real symmetric matrix A (Q has orthonormal columns)

$$\|q_j^{(k)} - (\pm q_j)\| = \mathcal{O}(C^k),$$

for $1 \leq j \leq r$, where $C < 1$ is the constant

$$C = \max_{1 \leq k \leq r} \frac{|\lambda_{k+1}|}{|\lambda_k|}$$

- For unsymmetric matrix A (Q does not have orthonormal columns)

$$\mathcal{R}(Q^{(k)}) \to \mathcal{R}(Q)$$

For Unnormalized Simultaneous Iteration, as $k \to \infty$, the vectors $q^{(1)}, q^{(2)}, \cdots, q^{(r)}$ all converge to multiples of the same dominant eigenvector q_1. Therefore, they form an ill-conditioned basis of span $\left\{q^{(1)}, q^{(2)}, \cdots, q^{(r)}\right\}$.

The remedy is simple, we should build orthonormal basis at each iteration $\rightsquigarrow$ **Simultaneous Iteration/Subspace Iteration**

Subspace Iteration:

```
random selection Q^(0) with orthonormal columns
for k = 1, 2, ···
      Z_k = AQ^(k-1)
      Z_k = Q^(k) R^(k)   reduced QR factorization
end
```

- Z_k and $Q^{(k)}$ has the same column space
- equal to the column space of $A^k Q^{(0)}$

- $\mathcal{R}(\mathrm{Q}^{(k)})$ converge to subspace associated with r largest eigenvalues in magnititude (dominant invariant subspace).
- $\lambda\left(\left(\mathrm{Q}^{(k)}\right)^H \mathrm{A}\mathrm{Q}^{(k)}\right) \to \{\lambda_1,\ \lambda_2,\ \cdots,\lambda_r\}$
- $\left|\lambda_i^{(k)} - \lambda_i\right| = \mathcal{O}\left(\left|\frac{\lambda_{r+1}}{\lambda_i}\right|^k\right),\ i = 1,\ 2,\ \cdots,\ r$
- also called simultaneously iteration or orthogonal iteration
- when $r = n$, it coincides with QR iteration

QR Iteration:

$$
\begin{array}{l}
A^{(0)} = A \\
\texttt{for } k = 1, \; 2, \; \cdots \\
\qquad Q^{(k)}R^{(k)} = A^{(k-1)} \quad \texttt{QR factorization of } A^{(k-1)} \\
\qquad A^{(k)} = R^{(k)}Q^{(k)} \\
\texttt{end}
\end{array}
$$

Facts:

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

Challenges of QR Iteration

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

- too computationally expensive!

Improvement:

Perform a similarity transform A to obtain a form $\mathrm{A}^{(0)} = (\mathrm{Q}^{(0)})^H \mathrm{A} \mathrm{Q}^{(0)}$

- the QR decomposition of $\mathrm{A}^{(0)}$ should be computationally cheap
- $\mathrm{A}^{(k)}$ $(k = 1, 2, \cdots)$ should have similar structure with $\mathrm{A}^{(0)}$ so that the QR decomposition at each iteration is computationally cheap

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

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

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

- by using Givens rotations

QR Iteration with Hessenberg Reduction:

```
A = Q^H HQ,  A^(0) = H,   H is upper Hessenberg
for k = 1, 2, ···
      Q^(k) R^(k) = A^(k−1)   QR factorization of A^(k−1)
      A^(k) = R^(k) Q^(k)
end
```

Key: $A^{(k)}$ 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 $A = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}$.

A Naive Try

Let Q_1 be the Householder reflection matrix that reflects a_1 to $-\text{sign}(a_1(1))\|a_1\|_2 e_1$,

$$A = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \rightarrow \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & \times & \times & \times \\ & \times & \times & \times \end{bmatrix}}_{Q_1 A} \rightarrow \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ \color{red}{\times} & \times & \times & \times \\ \color{red}{\times} & \times & \times & \times \\ \color{red}{\times} & \times & \times & \times \end{bmatrix}}_{Q_1 A Q_1^H}$$

Mission failed!

Less Ambitious Try

Let $\tilde{a}_1 = A(2:n, 1)$ and Q_1 be the Householder reflection matrix that reflects $\tilde{a}_1$ to $-\text{sign}(\tilde{a}_1(1))\|\tilde{a}_1\|_2 e_1$,

$$A = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \rightarrow \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & \times & \times & \times \end{bmatrix}}_{Q_1 A} \rightarrow \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & \times & \times & \times \end{bmatrix}}_{Q_1 A Q_1^H}$$

Repeat the above procedure to the 2nd column of $Q_1 A Q_1^H \cdots$

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

Hessenberg Reduction:

$$
\begin{array}{l}
\texttt{for}\ k = 1 : n-2 \\
\qquad \mathrm{x} = \mathrm{A}\texttt{(k+1:n, k)} \\
\qquad \mathrm{v}_k = \texttt{sign}(\mathrm{x}(1))\|\mathrm{x}\|_2 \mathrm{e}_1 + \mathrm{x} \\
\qquad \mathrm{v}_k = \dfrac{\mathrm{v}_k}{\|\mathrm{v}_k\|_2} \\
\qquad \mathrm{A}(k+1:n, k:n) = \mathrm{A}(k+1:n, k:n) - 2\mathrm{v}_k(\mathrm{v}_k^H \mathrm{A}(k+1:n, k:n)) \\
\qquad \mathrm{A}(1:n, k+1:n) = \mathrm{A}(1:n, k+1:n) - 2(\mathrm{A}(1:n, k+1:n)\mathrm{v}_k)\mathrm{v}_k^H \\
\texttt{end}
\end{array}
$$

Example:

Consider the following matrix

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$A^{(0)} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}}_{Q^{(0)}} \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{R^{(0)}}$$

$$A^{(1)} = R^{(0)} Q^{(0)} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = A^{(0)}$$

No convergence of $A^{(k)}$ observed.

To make QR iteration converge, i.e., $A^{(k)}$ converge to a upper triangular matrix, **shift** is required.

Shifted QR Iteration:

$$
\begin{array}{l}
\mathsf{A} = \mathsf{Q}^H \mathsf{H} \mathsf{Q}, \ \ \mathsf{A}^{(0)} = \mathsf{H}, \quad \texttt{H is upper Hessenberg} \\
\texttt{for } k = 1,\ 2,\ \cdots \\
\qquad \mathsf{Q}^{(k)} \mathsf{R}^{(k)} = \mathsf{A}^{(k-1)} - \mu_k \mathsf{I} \quad \texttt{QR factorization of } \mathsf{A}^{(k-1)} - \mu_k \mathsf{I} \\
\qquad \mathsf{A}^{(k)} = \mathsf{R}^{(k)} \mathsf{Q}^{(k)} + \mu_k \mathsf{I} \\
\texttt{end}
\end{array}
$$

Facts:

- $\mathsf{A}^{(k)}$ has same eigenvalues with A (requires a proof)
- shift μ_k may differ from iteration to iteration

Selection of Shift

- Raleigh Quotient shift: $\mu_k = \mathrm{A}^{(k-1)}(n, n)$
  - no guarantee on convergence
  - if converged, order of convergence is cubic
- Wilkinson shift

  Denote the lower-rightmost 2×2 matrix of $\mathrm{A}^{(k-1)}$ by

$$\mathrm{B} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

  The Wilkinson shift is chosen as the eigenvalue of B that is closer to d.
  - always converge for Hermitian/real symmetric matrices with cubic convergence rate (quadratic convergence for the worst case)

References

1. J. H. Wilkinson. Global convergence of tridiagonal QR algorithm with origin shifts. *Linear Algebra and its Applications*, 1(3): 409 – 420, 1968.

Brief Summary

- Power iteration
  - compute the largest eigenvalue in magnitude
  - convergence may be slow if $|\lambda_2|$ is close to $|\lambda_1|$
  - deflation technique (making a nonzero eigenvalue to zero) can be used to compute the second largest eigenvalue im magnititude
    - For real symmetric/Hermitian case, $\mathsf{A} = \mathsf{A} - \lambda_1 \mathsf{v}_1 \mathsf{v}_1^H$
    - complicated for unsymmetric/non-Hermitian case, investigate by yourself if interested.
- Inverse iteration (with shift)
  - compute the smallest eigenvalue in magnitude
  - when coming with shift μ, it computes the eigenvalues closest to μ

Brief Summary

- Subspace iteration
  - A block version of the power iteration, or power iteration applied to a subspace
  - compute a few largest eigenpairs in magnititude
  - inverse iteration can also be applied in the subspace iteration
  - when starting with full space, it coincides with QR iteration.
- QR iteration
  - compute all eigenvalues/eigenvectors
  - to reduce computational complexity, Hessenberg reduction is required before the iteration
  - shift is required to obtain convergence

Readings

You are supposed to read

- Gene H. Golub and Charles F. Van Loan. *Matrix Computations*, Johns Hopkins University Press, 2013. 1997.

  Chapter 7.3, 8.2, 8.3