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. 09, 2022

The subspace iteration is **equivalent** to QR iteration when applied to a full set of vectors ($r = n$).

Subspace Iteration

$$\begin{aligned}
\underline{\mathbf{Q}}^{(0)} &= \mathbf{I} \\
\mathbf{Z} &= \mathbf{A}\underline{\mathbf{Q}}^{(k-1)} \\
\mathbf{Z} &= \underline{\mathbf{Q}}^{(k)}\mathbf{R}^{(k)} \\
\mathbf{A}^{(k)} &= (\underline{\mathbf{Q}}^{(k)})^T\mathbf{A}\underline{\mathbf{Q}}^{(k)}
\end{aligned}$$

QR Iteration

$$\begin{aligned}
\underline{\mathbf{A}}^{(0)} &= \mathbf{A} \\
\mathbf{A}^{(k-1)} &= \mathbf{Q}^{(k)}\mathbf{R}^{(k)} \\
\mathbf{A}^{(k)} &= \mathbf{R}^{(k)}\mathbf{Q}^{(k)} \\
\underline{\mathbf{Q}}^{(k)} &= \mathbf{Q}^{(1)}\mathbf{Q}^{(2)}\cdots\mathbf{Q}^{(k)}
\end{aligned}$$

Theorem [Equivalence of Subspace iteration with QR iteration]

The above subspace iteration and QR iteration generate identical sequences of matrices $\mathbf{R}^{(k)}$, $\mathbf{Q}^{(k)}$, and $\mathbf{A}^{(k)}$ defined by the QR factorization of the k-th power of $\mathbf{A}$

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

with

$$\mathbf{A}^{(k)} = (\underline{\mathbf{Q}}^{(k)})^T\mathbf{A}\underline{\mathbf{Q}}^{(k)},$$

where

$$\underline{\mathbf{R}}^{(k)} = \mathbf{R}^{(k)}\mathbf{R}^{(k-1)}\cdots\mathbf{R}^{(1)}$$

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

- too computationally expensive!

Improvement:

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

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

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 (how?).

- by using Givens rotations

QR Iteration with Hessenberg Reduction:

$\mathbf{A} = \mathbf{Q}^H\mathbf{H}\mathbf{Q}$, $\mathbf{A}^{(0)} = \mathbf{H}$, $\mathbf{H}$ `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
```

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

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

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

where $\mathbf{Q}_1 = \begin{bmatrix} 1 & \\ & \tilde{\mathbf{Q}}_1 \end{bmatrix}$

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:

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

Example:

Consider the following matrix

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

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

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

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

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

Shifted QR Iteration:

$$\mathbf{A} = \mathbf{Q}^H \mathbf{H} \mathbf{Q}, \quad \mathbf{A}^{(0)} = \mathbf{H}, \quad \mathbf{H} \texttt{ is upper Hessenberg}$$

```
for k = 1, 2, ···
```

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

$$\mathbf{A}^{(k)} = \mathbf{R}^{(k)} \mathbf{Q}^{(k)} + \mu_k \mathbf{I}$$

```
end
```

Facts:

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

Selection of Shift

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

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

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

  The Wilkinson shift is chosen as the eigenvalue of $\mathbf{B}$ 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, $\mathbf{A} = \mathbf{A} - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^H$
    - 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