SI231b: Matrix Computations

Lecture 14: Eigenvalue Computations

Yue Qiu

qiuyue@shanghaitech.edu.cn

School of Information Science and Technology

ShanghaiTech University

Oct. 29, 2020

QR Iteration:

$A^{(0)} = A$

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

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

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

end

Facts:

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

Outline

- QR Iteration with Hessenberg Reduction
- QR Iteration with Shifts
- Subspace Iteration
- Brief Summary

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

- too computational 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., $Q^H AQ = H$ 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
for k = 1, 2, ···
      Q^(k) R^(k) = A^(k-1)   QR factorization of A^(k-1)
      A^(k) = R^(k) Q^(k)
end
```

$$\mathrm{A} = \mathrm{Q}^H \mathrm{H} \mathrm{Q}, \quad \mathrm{A}^{(0)} = \mathrm{H}$$

Key: $\mathrm{A}^{(k)}$ 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 $\mathrm{A} = \begin{bmatrix} \mathrm{a}_1 & \mathrm{a}_2 & \cdots & \mathrm{a}_n \end{bmatrix}$.

A Naive Try

Let Q_1 be the Householder reflection matrix that reflects a_1 to $\mathrm{sign}(\mathrm{a}_1(1))\|\mathrm{a}_1\|_2 \mathrm{e}_1$,

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

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:

```
for k = 1 : n-2
    x= A(k+1:n, k)
    v_k = sign(x(1))||x||_2 e_1 + x
    v_k = v_k / ||v_k||_2
    A(k+1 : n, k : n) = A(k+1 : n, k : n) - 2v_k(v_k^H A(k+1 : n, k : n))
    A(1 : n, k+1 : n) = A(1 : n, k+1 : n) - 2(A(1 : n, k+1 : n)v_k)v_k^H
end
```

$$
\begin{aligned}
&\texttt{for}\ k = 1 : n-2\\
&\quad x = A(k+1:n,\ k)\\
&\quad v_k = \mathrm{sign}(x(1))\|x\|_2 e_1 + x\\
&\quad v_k = \frac{v_k}{\|v_k\|_2}\\
&\quad A(k+1 : n, k : n) = A(k+1 : n, k : n) - 2v_k(v_k^H A(k+1 : n, k : n))\\
&\quad A(1 : n, k+1 : n) = A(1 : n, k+1 : n) - 2(A(1 : n, k+1 : n)v_k)v_k^H\\
&\texttt{end}
\end{aligned}
$$

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:

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

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

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

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

`end`

Facts:

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

Selection of Shift

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

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

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

  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.

Power Iterations for a Set of Vectors

From the Power Iteration, we know that

- $A^k q_0$ converges to the eigenvector associated with the largest eigenvalue in magnititude.
- if we start with a set of linearly independent vectors $\{q_1,\ q_2,\ \cdots, q_r\}$, then $A^k\{q_1,\ q_2,\ \cdots, q_r\}$ should converge (under suitable assumptions) to a subspace spanned by eigenvectors of A associated with r largest eigenvalues in magnititude.

Suppose there is a gap between the r largest eigenvalues in magnititude and λ_{r+1}, i.e, $|\lambda_1| \geq |\lambda_2| \geq \cdots |\lambda_r| > |\lambda_{r+1}|$

Subspace Iteration:

```
random selection Q^(0) with orthonormal columns
for k = 1, 2, ...
      Z_k = A Q^(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 $A^k Q^{(0)}$

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

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, 7.4, 8.2, 8.3