CS450: Numerical Analysis

Howard Hao Yang

Assistant Professor, ZJU-UIUC Institute

15/11/2023

Today's Agenda

- Eigenvalue and eigenvectors
- Power methods
- Sensitivity analysis

Eigenvalue Problems: Setup

- Given an $n \times n$ matrix $\boldsymbol{A}$
  - $\boldsymbol{x} \neq \boldsymbol{0}$ is called an eigenvector of if there exists a λ such that

$$\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$$

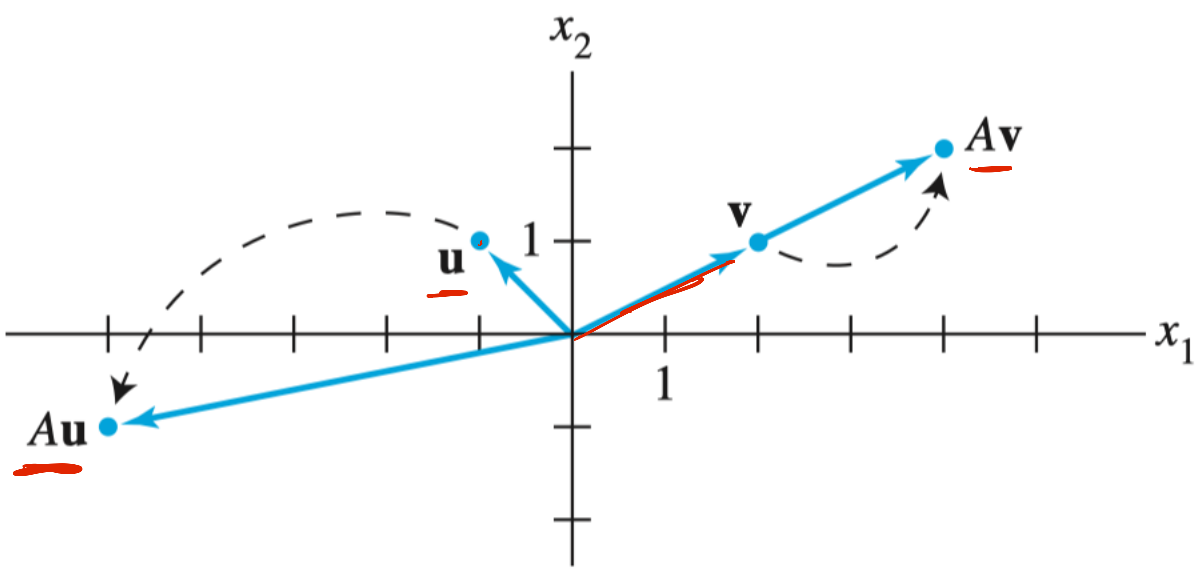
  - In that case, λ is called an eigenvalue
  - An eigenvector of a matrix determines a direction in which the matrix expands or shrinks any vector lying in that direction by a scalar multiple, given by the eigenvalue
  - The set of all eigenvalues $\lambda(\boldsymbol{A})$ is call the spectrum
  - The spectral radius is the magnitude of the biggest eigenvalue:

$$\rho(\boldsymbol{A}) = \max\{|\lambda| : \lambda(\boldsymbol{A})\}$$

Illustrating Example

- Let $\boldsymbol{A} = \begin{bmatrix} 3 & -2 \\ 1 & 0 \end{bmatrix}$, $\boldsymbol{u} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$, and $\boldsymbol{v} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$. The result of $\boldsymbol{Av}$ only stretches $\boldsymbol{v}$ but does not affect its direction

$Av = \begin{bmatrix} 4 \\ 2 \end{bmatrix} = 2\begin{bmatrix} 2 \\ 1 \end{bmatrix} = 2v$



Eigenvalue Transformations

(λ, x)

- What do the following transformations of the eigenvalue problem $\boldsymbol{Ax} = \lambda \boldsymbol{x}$ do?
  - Shift: $\boldsymbol{A} \to \boldsymbol{A} - \sigma \boldsymbol{I}$, correspondingly, $(\boldsymbol{A} - \sigma \boldsymbol{I})\boldsymbol{x} = (\lambda - \sigma)\boldsymbol{x}$
  - Inversion: $\boldsymbol{A} \to \boldsymbol{A}^{-1}$, correspondingly, $\boldsymbol{A}^{-1}\boldsymbol{x} = \lambda^{-1}\boldsymbol{x}$

    $\Rightarrow (A - \sigma I)^{-1} x = (\lambda - \sigma)^{-1} x$
  - Power: $\boldsymbol{A} \to \boldsymbol{A}^{k}$, correspondingly, $\boldsymbol{A}^{k}\boldsymbol{x} = \lambda^{k}\boldsymbol{x}$
  - Polynomial: $\boldsymbol{A} \to \sum_{k=0} a_k \boldsymbol{A}^{k}$, correspondingly, $\sum_{k=0} a_k \boldsymbol{A}^{k}\boldsymbol{x} = \sum_{k=0} a_k \lambda^{k}\boldsymbol{x}$
  - A function that has Taylor expansion: $\boldsymbol{A} \to f(\boldsymbol{A})$, correspondingly, $f(\boldsymbol{A})\boldsymbol{x} = \boldsymbol{f}(\lambda)\boldsymbol{x}$
  - Q: if -1 and 1 are two eigenvalues of $\boldsymbol{A}$, show that 1 is an eigenvalue of $\boldsymbol{A}^2$

Existence and Uniqueness

- Given an $n \times n$ matrix $\boldsymbol{A}$
  - note that

$$\boldsymbol{Ax} = \lambda \boldsymbol{x} \Rightarrow (\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{x} = \boldsymbol{0}$$

$a_n \neq 0$

  - the matrix $\boldsymbol{A} - \lambda \boldsymbol{I}$ is singular, which implies

$$\det(\boldsymbol{A} - \lambda \boldsymbol{I}) = 0$$

$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 = 0$

  - $\det(\boldsymbol{A} - \lambda \boldsymbol{I})$ is call the characteristic polynomial, which has degree n and hence n (potentially complex) roots
  - an $n \times n$ matrix $\boldsymbol{A}$ always has n eigenvalues, but they may not be real and may not be distinct

Example

- Give characteristic polynomial, eigenvalues, and eigenvectors of

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \to \det\left(\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$$

$$\downarrow$$

- The characteristic polynomial of this matrix is $(\lambda - 1)^2$ $\Rightarrow$ $\lambda_1 = \lambda_2 = 1$
- Eigenvectors:

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

which results in $x + y = x$. Hence, $y = 0$. So, we only have a 1-D space of eigenvectors. In this case, AM=2 > GM=1.

Example

- Find the eigenvalues and eigenvectors of the following matrix

$$A = \begin{bmatrix} 2 & 0 & 0 \\ 1 & 1 & 2 \\ 1 & -1 & 4 \end{bmatrix}$$

$\det(A-\lambda I)=0$

- The characteristic polynomial of this matrix is $\underbrace{p_A(\lambda) = (\lambda - 3)(\lambda - 2)^2 = 0}_{\lambda_1 = 3,\ \lambda_2=\lambda_3=2}$

- The eigenvectors are

$$\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\} = \{(0,1,1)^t, (0,2,1)^t, (-2,0,1)^t\}$$

Notion (1): Multiplicity

$(\lambda - \lambda_i)^{n_i}$

- What is the multiplicity of a matrix?
  - Algebraic Multiplicity: multiplicity of the root of the characteristic polynomial
  - Geometric Multiplicity: number of linearly independent eigenvectors (i.e., the dimensionality of the eigen space) $\bar{n}_i$
- In general, AM ≥ GM
- If the inequality holds strictly, i.e., AM > GM, the matrix is called defective

Existence and Uniqueness (Cont'd)

n $n \geq 5$

- Does characteristic polynomial really help obtain eigenvalues algorithmically?
  - Not quite. – Abel's theorem: for $n \geq 5$, there is no general formula for roots of polynomial.

    $\det(A - \lambda I) = 0$
  - Hence, albeit its (extreme) usefulness for theoretical purpose, the characteristic polynomial turns out not to be useful as a means of actually computing eigenvalues for matrices of nontrivial size.
  - For LU and QR, we can obtain exact answers (except rounding).
  - For eigenvalue problems: not possible—we must perform approximations.

Another possible Issue: Round-off

ZJU-UIUC INSTITUTE

- Consider matrix $\boldsymbol{A}$ as follows

$\det(A-\lambda I)$

$$\boldsymbol{A} = \begin{bmatrix} 1 & \epsilon \\ \epsilon & 1 \end{bmatrix}$$

where ϵ is slightly smaller than $\sqrt{\epsilon_{\mathrm{mach}}}$ in a given floating-point system

- A quick computation shows that the exact eigenvalues of $\boldsymbol{A}$ are $1+\epsilon$ and $1-\epsilon$, while computing the characteristic polynomial of $\boldsymbol{A}$ in floating-point arithmetic results in

$\epsilon^2 < \epsilon_{mach}$

$$\det(\boldsymbol{A} - \lambda \boldsymbol{I}) = \lambda^2 - 2\lambda + (1 - \epsilon^2) = \lambda^2 - 2\lambda + 1$$

giving $\lambda = 1$ as different solutions

Power Method: Preliminaries

- Definition (Linear independency): Let $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ be a set of vectors, the set is linearly independent if the only solution to

$$\alpha_1 \boldsymbol{v}_1 + \alpha_2 \boldsymbol{v}_2 + \cdots + \alpha_n \boldsymbol{v}_n = \boldsymbol{0}$$

is $\alpha_1 = \alpha_2 = \cdots = \alpha_n$. Otherwise, the set is linearly dependent.

- Theorem: Let $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ be a set of linearly independent vectors in R^n, then for any vector $\boldsymbol{x} \in R^n$, a unique collection of constants $\beta_1, \cdots, \beta_n$ exists with

$$\beta_1 \boldsymbol{v}_1 + \beta_2 \boldsymbol{v}_2 + \cdots + \beta_n \boldsymbol{v}_n = \boldsymbol{x}$$

(1,0)

(1,0)

$x = x_1 e_1 + x_2 e_2$

Power Method: Preliminaries (Cont'd)

ZJU-UIUC INSTITUTE

- Definition (basis): any collection of n linearly independent vectors in R^n is called a basis for R^n
- Theorem: If $\boldsymbol{A}$ is a $n \times n$ matrix and $\lambda_1, \lambda_2, \cdots, \lambda_n$ are distinct eigenvalues of $\boldsymbol{A}$ with associated eigenvectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n$, then $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}$ is a linearly independent set, i.e., they form a basis for R^n

$$\mathbb{R}^3: \left\{\begin{bmatrix}1\\0\\0\end{bmatrix},\begin{bmatrix}0\\1\\0\end{bmatrix},\begin{bmatrix}0\\0\\1\end{bmatrix}\right\} \rightleftharpoons \left\{\begin{bmatrix}1\\0\\0\end{bmatrix},\begin{bmatrix}0\\e^{-j\frac{\pi}{3}}\\0\end{bmatrix},\begin{bmatrix}0\\0\\e^{-j\frac{2\pi}{3}}\end{bmatrix}\right\}$$

Euclidean basis　　　　Fourier basis

Power Method

- Consider an $n \times n$ matrix $\boldsymbol{A}$.
- Assume $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|$ with eigenvectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. Further assume $\|\boldsymbol{x}_i\| = 1$

$\to \mathbb{R}^n$

- More generally, $\boldsymbol{x} = \sum_{i=1}^{n} \alpha_i \boldsymbol{x}_i$ (this is usually the case, why?)
- Then, $\boldsymbol{y} = \boldsymbol{A}^{1000}\boldsymbol{x} = \sum_{i=1}^{n} \alpha_i \lambda_i^{1000} \boldsymbol{x}_i$
- As such, $\frac{\boldsymbol{y}}{\lambda_1^{1000}} = \alpha_1 \boldsymbol{x}_1 + \sum_{i=2}^{n} \alpha_i \left(\frac{\lambda_i}{\lambda_1}\right)^{1000} \boldsymbol{x}_i \approx \alpha_1 \boldsymbol{x}_1$
- Use this as a computational procedure to find $\boldsymbol{x}_1$ is called the power method

Power Method (Cont'd)

- Algorithm details

$A \in \mathbb{R}^{n \times n}$

Algorithm 4.1 Power Iteration

$\boldsymbol{x}_0$ = arbitrary nonzero vector $\in \mathbb{R}^n$
for $k = 1, 2, \ldots$
  $\boldsymbol{x}_k = \boldsymbol{A}\boldsymbol{x}_{k-1}$ { generate next vector }
end

- This algorithm finds the eigenvector corresponding to the maximum eigenvalue (a.k.a. the leading eigenvalue)
- More concretely, this algorithm converges to a multiple of $\boldsymbol{x}_1$

Example (1)

- Consider finding the leading eigenvector of the following matrix

$$\boldsymbol{A} = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \rightarrow 4, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad Ax = \lambda x$$

- Executing the power method with an arbitrarily initialized vector gives

$$\boldsymbol{x}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

k	$\boldsymbol{x}_k^T$		Ratio
0	0	1	
1	1	3	3.000
2	6	10	3.333
3	28	36	3.600
4	120	136	3.778
5	496	528	3.882
6	2016	2080	3.939
7	8128	8256	3.969
8	32640	32896	3.984
9	130816	131328	3.992

A A

$\frac{3}{1}$ $\frac{10}{3}$ $\frac{36}{10}$ ~ 4

Characterizing (iterative) Algorithms

- How fast can we find the leading eigenvector?
- **Definition** (Rate of Convergence): Suppose $\{\beta_n\}_{n=1}^{\infty}$ is a sequence know to converge to zero and $\{\alpha_n\}_{n=1}^{\infty}$ converges to a number α. If a positive constant exists with

err in iteration n

$$|\alpha_n - \alpha| \leq K\beta_n, \text{ for large } n$$

then we say $\{\alpha_n\}_{n=1}^{\infty}$ converges to α with rate, or order, of convergence $O(\beta_n)$, we can also write it as $\alpha_n = \alpha + O(\beta_n)$

$\alpha_n = \frac{1}{n} + \frac{1}{n^2}$

$\bar{\alpha}_n = \frac{1}{n^2} + \frac{3}{n^3}$ ↓ 0

- Example: $\alpha_n = \frac{n+1}{n^2}$ and $\bar{\alpha}_n = \frac{n+3}{n^3}$, which converges faster?

Power Method: Performance (Cont'd)

ZJU-UIUC INSTITUTE

$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|$

- How fast can we find the leading eigenvector?

- If we denote by $e_k = \|\boldsymbol{x}_1 - \tilde{\boldsymbol{x}}_1^{(k)}\|$, check that $e_{k+1} \approx \frac{|\lambda_2|}{|\lambda_1|} e_k$; which gives the convergence rate of power method as $e_k = O\left(\left(\frac{|\lambda_2|}{|\lambda_1|}\right)^k\right)$ – this is commonly known as the linear convergence rate (why linear?)

$\frac{|\lambda_2|}{|\lambda_1|} < 1$

Illustrative Example (1)

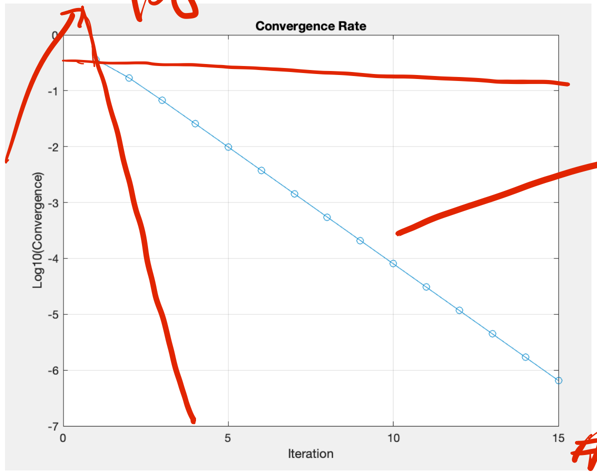
- Run the following code

```
% Example matrix
A = [2, 1; 1, 3];

% Set parameters
max_iterations = 50;
tol = 1e-6;

% Run the power method with convergence plot
[eigenvalue, eigenvector, convergence_rate] = power_method_with_convergence_plot(A, max_iterations, tol);

% Display the result
fprintf('Dominant Eigenvalue: %f\n', eigenvalue);
fprintf('Corresponding Eigenvector: [%f, %f]\n', eigenvector(1), eigenvector(2));
```

$A = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$

```
Converged in 15 iterations.
Dominant Eigenvalue: 3.618034
Corresponding Eigenvector: [0.525731, 0.850651]
>>
```

log(||e||)



slop

$\frac{|\lambda_2|}{|\lambda_1|}$

iterations

$|\lambda_2| \ll |\lambda_1|$

$|\lambda_2| \approx |\lambda_1|$

```
function [eigenvalue, eigenvector, convergence_rate] = power_method_with_convergence_plot(A, max_iterations, tol)
    n = size(A, 1);

    % Initialize a random vector as an initial guess for the eigenvector
    x0 = randn(n, 1);
    x0 = x0 / norm(x0);

    % Initialize arrays to store convergence data
    convergence_data = zeros(max_iterations, 1);

    % Power iteration
    for iterations = 1:max_iterations
        x1 = A * x0;
        eigenvalue = x1' * x0;
        x1 = x1 / norm(x1);

        % Store the convergence data
        convergence_data(iterations) = norm(x1 - x0);

        % Check for convergence
        if norm(x1 - x0) < tol
            fprintf('Converged in %d iterations.\n', iterations);
            eigenvector = x1;

            % Plot the convergence rate
            figure;
            plot(1:iterations, log10(convergence_data(1:iterations)), '-o');
            title('Convergence Rate');
            xlabel('Iteration');
            ylabel('Log10(Convergence)');
            grid on;
            break;
        end

        x0 = x1;
    end

    % If it didn't converge within the specified iterations
    if iterations == max_iterations
        fprintf('Power method did not converge within the specified number of iterations.\n');
        eigenvalue = [];
        eigenvector = [];
    end

    % Return the convergence rate for further analysis if needed
    convergence_rate = convergence_data(1:iterations);
end
```

Deflation

$A = \cancel{\lambda_1 x_1 x_1^T} + \lambda_2 x_2 x_2^T + \cdots + \lambda_n x_n x_n^T$

$A - \lambda_1 x_1 x_1^T \leftarrow$ power method

- For matrix $\boldsymbol{A}$, we can compute a pair λ_1 and $\boldsymbol{x}_1$ of eigenvalue and eigenvector – how about the rest?
- Suppose $||\boldsymbol{x}_1|| = 1$, we can choose $\boldsymbol{u}_1 = \lambda_1 \boldsymbol{x}_1$, and the matrix $\boldsymbol{A} - \boldsymbol{x}_1 \boldsymbol{u}_1^T$ will have eigenvalues $0, \lambda_2, \ldots, \lambda_n$
- Employ the power method can obtain the other eigenvalues

Power Method: Issues

$x = \alpha_1 x_1 + \cdots + \alpha_n x_n$

$\alpha_1 = 0$

- What can go wrong?
  - Starting vector has no component along $\boldsymbol{x}_1$ (this is not a big problem, due to the random initialization) ✗
  - Overflow in computing, e.g., λ_i^{1000} (this can be addressed by normalization) ✗

Algorithm 4.2 Normalized Power Iteration

$\boldsymbol{x}_0$ = arbitrary nonzero vector
for $k = 1, 2, \ldots$
  $\boldsymbol{y}_k = \boldsymbol{A}\boldsymbol{x}_{k-1}$ { generate next vector }
  $\boldsymbol{x}_k = \boldsymbol{y}_k / \|\boldsymbol{y}_k\|_\infty$ { normalize }
end

- ✓ $|\lambda_1| = |\lambda_2|$, this is a real problem

$|\lambda_2| \approx |\lambda_1|$

Power Method: Response

$|\lambda_2| \approx |\lambda_1|$

$e_{k+1} \approx e_k$

- If we denote by $e_k = \|\boldsymbol{x}_1 - \widetilde{\boldsymbol{x}}_1\|$, check that $e_{k+1} \approx \frac{|\lambda_2|}{|\lambda_1|} e_k$; which gives the convergence rate of power method as $e_k = O\left(\left(\frac{|\lambda_2|}{|\lambda_1|}\right)^k\right)$

$e_{k+1} \approx 0.99\, e_k$

- The convergence rate of power method depends on $\frac{|\lambda_2|}{|\lambda_1|}$, if it is very small, converges very fast; if it is close to one, converges slowly ($\lambda_1 = 1.01,\ \lambda_2 = 1$)
- Recall the shift operation: $\boldsymbol{A} \to \boldsymbol{A} - \sigma\boldsymbol{I}$, correspondingly, $(\boldsymbol{A} - \sigma\boldsymbol{I})\boldsymbol{x} = (\lambda - \sigma)\boldsymbol{x}$; can we manipulate something to $\boldsymbol{A}$ such that $\frac{|\lambda_2 - \sigma|}{|\lambda_1 - \sigma|}$ is relatively small?

λ $(\lambda - \sigma)$

A Detour

- If not the largest, how can I find the eigenvector associated with the smallest eigenvalue of $\boldsymbol{A}$?
- Approach: use power method to $\boldsymbol{A}^{-1}$ $(\lambda_1^{-1}, \cdots, \lambda_n^{-1})$
- Algorithm (adopts a LU factorization for $\boldsymbol{A}$) $A = LU$

$\max \frac{1}{\lambda_i} = \frac{1}{\min \lambda_i}$

$A y_k = x_{k-1}$

1. $L\omega = x_{k-1}$
2. $U y_k = \omega$

Algorithm 4.3 Inverse Iteration

$\boldsymbol{x}_0 =$ arbitrary nonzero vector
for $k = 1, 2, \ldots$
  Solve $\boldsymbol{A}\boldsymbol{y}_k = \boldsymbol{x}_{k-1}$ for $\boldsymbol{y}_k$ { generate next vector }
  $\boldsymbol{x}_k = \boldsymbol{y}_k / \|\boldsymbol{y}_k\|_\infty$ { normalize }
end

$y_k = A^{-1} y_{k-1}$

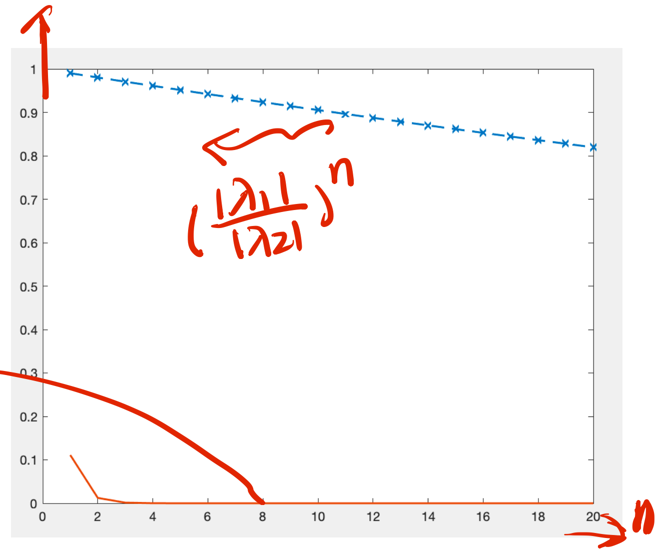
$y_k = y_k / \|y_k\|_\infty$

Power Method Strikes Back

$\lambda_1 \approx \lambda_2 \quad \lambda_3 \quad \cdots \quad \lambda_n$

- How does the previous observation help us accelerate the power method?
- Given an approximated value $\boxed{\sigma}$ to λ_1, such that $|\lambda_1 - \sigma| < |\lambda_i - \sigma|$ for $i \neq 1$, (e.g., $\lambda_1 = 1.01$, $\lambda_2 = 1$, $\sigma = 1.009$)
- Can you design an algorithm to get $\boldsymbol{x}_1$ but run faster than the vanilla power method?
- Apply power method to $(\boldsymbol{A} - \sigma \boldsymbol{I})^{-1}$
- Why it gives you $\boldsymbol{x}_1$?
- What is the convergence rate?

$\left(\left|\frac{\lambda_1 - \sigma}{\lambda_2 - \sigma}\right|\right)^n$



Power Method Strikes Back (Cont'd)

of σ satisfies $\underline{|\lambda_1-\sigma|} < |\lambda_2-\sigma|, |\lambda_3-\sigma|, \ldots, |\lambda_n-\sigma|$

eigenvalues of $(A-\sigma I)^{-1}$: $(\lambda_1-\sigma)^{-1}, (\lambda_2-\sigma)^{-1}, \ldots, (\lambda_n-\sigma)^{-1}$

- Apply power method to $(A - \sigma I)^{-1}$ $\rightarrow$ $(\lambda_1-\sigma)^{-1}, \underline{x}$
- Why it gives you x_1?

$$(A-\sigma I)^{-1}x = (\lambda_1-\sigma)^{-1}x$$

$$\Rightarrow \underbrace{(\lambda_1-\sigma)}x = (A-\sigma I)x$$

$$\lambda_1 x - \cancel{\sigma x} = Ax - \cancel{\sigma x} \Rightarrow \lambda_1 x = Ax \rightarrow x = x_1, \quad Ax_1 = \lambda_1 x_1$$

$$\left|\frac{\lambda_2}{\lambda_1}\right|^k \quad \checkmark$$

- What is the convergence rate?

$$\hat{e}_{k+1} = \frac{\frac{1}{|\lambda_2-\sigma|}}{\frac{1}{|\lambda_1-\sigma|}} \cdot \hat{e}_k = \frac{|\lambda_1-\sigma|}{|\lambda_2-\sigma|} \cdot \hat{e}_k \Rightarrow \hat{e}_k = O\left(\left|\frac{\lambda_1-\sigma}{\lambda_2-\sigma}\right|^k\right)$$

Power Method Strikes Back (Cont'd)

- The remaining question… how to find an approximation to λ_1?
- For any vector $\boldsymbol{x}$, the Rayleigh quotient provides an estimate of some eigenvalues of $\boldsymbol{A}$:

$A x_1 = \lambda_1 x_1$

$$\rho_{\boldsymbol{A}}(\boldsymbol{x}) = \frac{\boldsymbol{x}^H \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^H \boldsymbol{x}} \rightarrow \lambda_1$$

- If $\boldsymbol{x}$ is an eigenvector of $\boldsymbol{A}$, then $\rho_{\boldsymbol{A}}(\boldsymbol{x})$ is the associated eigenvalue

Rayleigh Quotient Iteration

ZJU-UIUC INSTITUTE

- In summary, compute the shift σ using Rayleigh quotient as $\sigma_k = \dfrac{\boldsymbol{x}_k^T \boldsymbol{A} \boldsymbol{x}_k}{\boldsymbol{x}_k^T \boldsymbol{x}_k}$
- Then, apply inverse iteration with that shift: $\boldsymbol{x}_{k+1} = (\boldsymbol{A} - \sigma_k \boldsymbol{I})^{-1} \boldsymbol{x}_k$
- More concretely

Algorithm 4.4 Rayleigh Quotient Iteration

1 $\boldsymbol{x}_0$ = arbitrary nonzero vector

for $k = 1, 2, \ldots$

2 $\sigma_k = \boldsymbol{x}_{k-1}^T \boldsymbol{A} \boldsymbol{x}_{k-1} / \boldsymbol{x}_{k-1}^T \boldsymbol{x}_{k-1}$ { compute shift }

3 Solve $(\boldsymbol{A} - \sigma_k \boldsymbol{I})\, \boldsymbol{y}_k = \boldsymbol{x}_{k-1}$ for $\boldsymbol{y}_k$ { generate next vector }

$\boldsymbol{x}_k = \boldsymbol{y}_k / \|\boldsymbol{y}_k\|_\infty$ { normalize }

end

$y_k = (A - \sigma_k I)^{-1} x_k$

$x_{k+1} = (A - \sigma_k I)^{-1} x_k$

(Detour) Notation: Rayleigh Quotient

- For any vector $\boldsymbol{x}$, the Rayleigh quotient provides an estimate of some eigenvalues of $\boldsymbol{A}$:

$$\rho_{\boldsymbol{A}}(\boldsymbol{x}) = \frac{\boldsymbol{x}^H \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^H \boldsymbol{x}}$$

- If $\boldsymbol{x}$ is an eigenvector of $\boldsymbol{A}$, then $\rho_{\boldsymbol{A}}(\boldsymbol{x})$ is the associated eigenvalue
- Courant-Fischer Minimax Theorem: if $\boldsymbol{A} \in \boldsymbol{R}^{n \times n}$ is symmetric, then

$$\lambda_k(\boldsymbol{A}) = \max_{\dim(S)=k} \min_{0 \neq x \in S} \frac{\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}$$

Example (2)

ZJU-UIUC INSTITUTE

- Consider finding the leading eigenvector of the following matrix

$$A = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \leftarrow \quad 4, \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

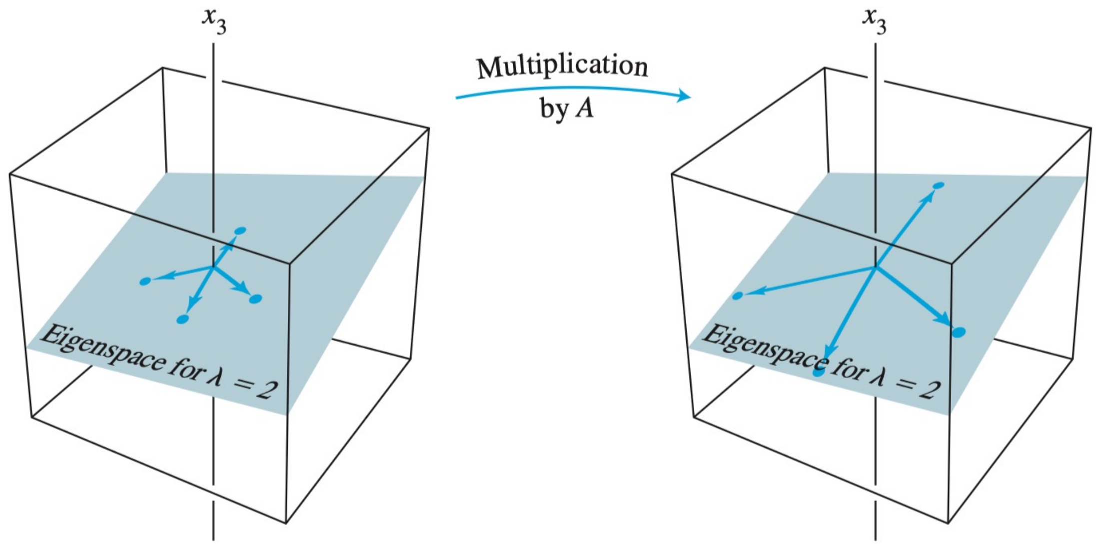
- Executing the power method with an arbitrarily initialized vector gives

k	x_k^T		σ_k
0	0.807	0.397	3.792
1	0.924	1.000	3.997
2	1.000	1.000	4.000

Invariant Subspace

- For an $n{\times}n$ matrix $\boldsymbol{A}$, a subspace S of R^n is said to be an invariant subspace if $\boldsymbol{A}S \subset S$, i.e., if $\boldsymbol{x} \in S$ implies $\boldsymbol{A}\boldsymbol{x} \in S$.
- The eigenspace $S_\lambda = \{\boldsymbol{x}: \boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}\}$ corresponding to eigenvalue λ is an invariant subspace.

$$A = \begin{bmatrix} 4 & -1 & 6 \\ 2 & 1 & 6 \\ 2 & -1 & 8 \end{bmatrix} \Rightarrow$$



Invariant Subspace (Cont'd)

- For an $n \times n$ matrix $\boldsymbol{A}$, a subspace S of R^n is said to be an invariant subspace if $\boldsymbol{A}S \subset S$, i.e., if $\boldsymbol{x} \in S$ implies $\boldsymbol{Ax} \in S$.
- The eigenspace $S_\lambda = \{\boldsymbol{x}: \boldsymbol{Ax} = \lambda \boldsymbol{x}\}$ corresponding to eigenvalue λ is an invariant subspace.
- More generally, if $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_p$ are eigenvectors of $\boldsymbol{A}$, then span($[\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_p]$) is an invariant subspace.
- Recall: matrix $\boldsymbol{P} \in R^{n \times n}$ is called a projection matrix if it satisfies $\boldsymbol{P}^2 = \boldsymbol{P}$ (which is also known as idempotent), span($\boldsymbol{P}$) is also an invariant subspace

Sensitivity

- Suppose $\boldsymbol{A}$ is not defective. Suppose that $\boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X} = \boldsymbol{D}$. For a certain perturbation in the input, i.e., $\boldsymbol{A} \to \boldsymbol{A} + \boldsymbol{E}$, how much would it affect the eigenvalues $\boldsymbol{D}$?
- Recall: in linear system or linear least square, the sensitivity depends on $\boldsymbol{A}$, would it be similar case here?
- Note that $\boldsymbol{X}^{-1}(\boldsymbol{A} + \boldsymbol{E})\boldsymbol{X} = \boldsymbol{D} + \boldsymbol{F}$
  - $\boldsymbol{A} + \boldsymbol{E}$ and $\boldsymbol{D} + \boldsymbol{F}$ have the same eigenvalues
  - $\boldsymbol{D} + \boldsymbol{F}$ is not necessarily diagonal

Sensitivity (Cont'd)

- Suppose $\boldsymbol{v}$ is an eigen vector of the perturbed matrix $\boldsymbol{A}+\boldsymbol{E}$ with an eigenvalue μ, satisfying $(\boldsymbol{D}+\boldsymbol{F})\boldsymbol{v}=\mu\boldsymbol{v}$, it gives

$$\boldsymbol{F}\boldsymbol{v}=(\mu\boldsymbol{I}-\boldsymbol{D})\boldsymbol{v} \Rightarrow (\mu\boldsymbol{I}-\boldsymbol{D})^{-1}\boldsymbol{F}\boldsymbol{v}=\boldsymbol{v} \text{ (when is it invertible?)}$$

$$\Rightarrow \|\boldsymbol{v}\| \le \left\|(\mu\boldsymbol{I}-\boldsymbol{D})^{-1}\right\| \cdot \|\boldsymbol{F}\| \cdot \|\boldsymbol{v}\|$$

$$\Rightarrow \left\|(\mu\boldsymbol{I}-\boldsymbol{D})^{-1}\right\|^{-1} \le \|\boldsymbol{F}\|$$

- Let λ_k be the entry in $\boldsymbol{D}$ that is closest to μ, then

$$|\lambda_k-\mu| = \|(\mu\boldsymbol{I}-\boldsymbol{D})^{-1}\|^{-1} \le \|\boldsymbol{F}\| = \|\boldsymbol{X}^{-1}\boldsymbol{E}\boldsymbol{X}\| \le \|\boldsymbol{X}^{-1}\|\|\boldsymbol{E}\|\|\boldsymbol{X}\| = \kappa(\boldsymbol{X})\|\boldsymbol{E}\|$$

Sensitivity (Cont'd)

- Let λ_k be the entry in $\boldsymbol{D}$ that is closest to μ, then

$$|\lambda_k - \mu| \leq \kappa(\boldsymbol{X})\|\boldsymbol{E}\|$$

- The conditioning depends on the eigenvectors
  - The solutions are sensitive to input perturbations if the eigenvectors are nearly linearly dependent
  - If $\boldsymbol{X}$ is orthogonal (e.g., for symmetric matrix $\boldsymbol{A}$), then eigenvalues are always well-conditioned
  - This bound is in terms of all eigenvalues, so may overestimate for each individual eigenvalue