

## Overview

Eigenvalues/PF  
Diagonalization  
Error

PF; refinements  
Eigenvalue apps.

$$|x_1| > |x_2| \dots$$

$$x = \alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_n x_n$$

$$\frac{\Delta^{1000} x}{\lambda_1^{1000}} = \alpha_1 x_1 + \alpha_2 \frac{\lambda_2^{1000}}{\lambda_1^{1000}} x_2 + \dots + \alpha_n \frac{\lambda_n^{1000}}{\lambda_1^{1000}} x_n$$

# Diagonalizability

If we have  $n$  eigenvectors with different eigenvalues, the matrix is diagonalizable.

$$X = \begin{pmatrix} | & & | \\ x_1 & \dots & x_n \\ | & & | \end{pmatrix}$$

$$A x_i = \lambda_i x_i \\ x_i \neq 0$$

$$AX = \begin{matrix} \lambda_1 x_1 & & \\ & \dots & \\ & & \lambda_n x_n \end{matrix}$$

~~$DX$~~

$$X^{-1}AX = D$$

$$| X^{-1} \cdot D = \begin{pmatrix} \lambda_1 & & \\ & \dots & \\ & & \lambda_n \end{pmatrix}$$

$\boxed{X} \quad \boxed{D}$

## Are all Matrices Diagonalizable?

Give characteristic polynomial, eigenvalues, eigenvectors of

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}. \leftarrow \text{not diagonalizable}$$

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} \leftarrow$$

$$\begin{aligned} |x + |y = y & \Rightarrow y = 0 \\ \text{↪ } \begin{matrix} 0x + |y = y \\ y = y \end{matrix} \end{aligned}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} ? \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

## Power Iteration

We can use linear-independence to find the eigenvector with the largest eigenvalue. Consider the eigenvalues of  $A^{1000}$ .

## Power Iteration: Issues?

What could go wrong with Power Iteration?

- Overflow (un-normalised)
- $|\lambda_1| < |\lambda_2|$
- Starting vector has no comp. in  $\vec{x}_1$   
not an issue due to rounding

## What about Eigenvalues?

Power Iteration generates eigenvectors. What if we would like to know eigenvalues?

$$Ax = \begin{pmatrix} \vdots \\ \vdots \end{pmatrix}$$

$$\lambda x = \begin{pmatrix} \vdots \\ \vdots \end{pmatrix}$$

In principle:  $A\vec{x} / \vec{x}$



$$\frac{x^u \cdot Ax^u}{x^u \cdot x^u} = \text{Rayleigh quotient}$$

$$\frac{x^u \cdot Ax^u}{x^u \cdot x^u} = \lambda$$

## Convergence of Power Iteration

What can you say about the convergence of the power method?

Say  $v_1^{(k)}$  is the  $k$ th estimate of the eigenvector  $x_1$ , and

$$e_k = \left\| x_1 - v_1^{(k)} \right\|.$$

The diagram illustrates the convergence of the power method. It shows the decomposition of the normalized power iteration result into the dominant eigenvector and a term representing the error.

$$\frac{A^{1000} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}}{\lambda_1^{1000}} = \alpha_1 x_1 + \underbrace{\frac{1}{\lambda_1^{1000}} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}}_{\text{error } e_k}$$

The diagram shows the expression  $\frac{A^{1000} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}}{\lambda_1^{1000}}$  on the left. This is equal to  $\alpha_1 x_1$  plus a term in parentheses:  $\frac{1}{\lambda_1^{1000}} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$ . The term in parentheses is circled and labeled "error  $e_k$ ".

$$\|e_k\| \approx \left( \frac{\lambda_2}{\lambda_1} \right)^k \|e_{k-1}\|$$

$$|\lambda_1| > |\lambda_2|$$

# Transforming Eigenvalue Problems

Suppose we know that  $Ax = \lambda x$ . What are the eigenvalues of these changed matrices?

**Power.**  $A \rightarrow A^k$

$$A^2 \vec{x} = AA \vec{x} = A\lambda \vec{x} = \lambda^2 \vec{x}$$

**Shift.**  $A \rightarrow A - \sigma I$

$$(A - \sigma I) \vec{x} = A \vec{x} - \sigma \vec{x} = \lambda \vec{x} - \sigma \vec{x} = (\lambda - \sigma) \vec{x}$$

**Inversion.**  $A \rightarrow A^{-1}$

$$A^{-1} x = A^{-1} \frac{Ax}{\lambda} = \frac{x}{\lambda}$$
$$x = \frac{Ax}{\lambda}$$

$A$  $-51$  $151$  $A - 3I$  $-9 = -5 - 3$  $2 = 5 - 3$  $A$  $0.25$  $\underline{5}$  $\textcircled{A^{-1}}$  $\underline{\frac{1}{0.25} = 4}$  $0.2$

$$A: \quad -3 \quad 0.2 \quad 5$$

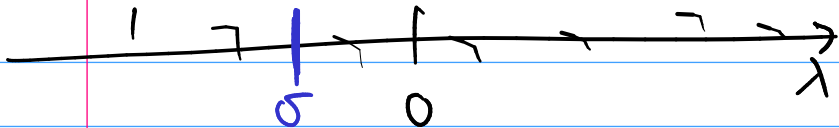
$$(A - 2I): \quad -5 \quad -2.2 \quad 3$$

---

$$A: \quad -3 \quad 0.2 \quad 1000$$

$$A + 2I: \quad -1 \quad 2.2 \quad 1002$$

$$(A + 2I)^{-1}: \quad -1 \quad \frac{1}{2.2} \quad \frac{1}{1002}$$



Shift:  $A - \sigma I$

Invert:

"Inverse Iteration"

Inverse iteration + RQ as shift;  $RQI$

## Inverse Iteration / Rayleigh Quotient Iteration

Describe [inverse iteration](#).

$$\vec{x}_{k+1} = (A - \sigma \mathbf{I})^{-1} \vec{x}_k \quad \vec{x}_0 = ?$$

Describe [Rayleigh Quotient Iteration](#).

$$\sigma = \frac{\vec{x}^T A \vec{x}}{\vec{x}^T \vec{x}}$$

$$\vec{x}_{k+1} = (A - \sigma \mathbf{I})^{-1} \vec{x}_k$$



## Demo: Power Iteration and its Variants

### In-class activity: Eigenvalue Iterations

$$Ax = b$$
$$\Leftrightarrow x = A^{-1}b$$

*l.a. solve()*

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

$$\Leftrightarrow (A - \sigma I) x_{k+1} = x_k$$

## Computing Multiple Eigenvalues

All Power Iteration Methods compute one eigenvalue at a time.  
What if I want *all* eigenvalues?

$$x_{k+1} = Ax \quad A \begin{pmatrix} | & | \\ x & y \\ | & | \end{pmatrix} = \begin{pmatrix} A_1 & A_2 \\ \uparrow & \uparrow \\ & \text{orthogonalize} \end{pmatrix}$$

~~" simultaneous iteration "~~

" orthogonal iteration "

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

$$\left( \begin{array}{c|cccc} \vec{x} & | & | & | & | \\ \hline & & & & \end{array} \right) A Q^{-1} = \left( \begin{array}{c|ccc} \lambda & ? & ? \\ \hline 0 & \boxed{0} & \end{array} \right)$$

$Q$       orth. to  $\vec{x}$       Deflation

## Simultaneous Iteration

What happens if we carry out power iteration on multiple vectors simultaneously?

$$Ax = \lambda x$$

$$y = Jx$$

$$Ay = JAy = 5\lambda x \\ = \lambda y$$

# Outline

Python, Numpy, and Matplotlib  
Making Models with Polynomials  
Making Models with Monte Carlo  
Error, Accuracy and Convergence  
Floating Point  
Modeling the World with Arrays  
    The World in a Vector  
    What can Matrices Do?  
    Graphs  
    Sparsity  
Norms and Errors  
The 'Undo' Button for Linear Operations: LU  
Repeating Linear Operations:  
Eigenvalues and Steady States  
**Eigenvalues: Applications**

Approximate Undo: SVD and Least Squares  
SVD: Applications  
    Solving Funny-Shaped Linear Systems  
    Data Fitting  
    Norms and Condition Numbers  
    Low-Rank Approximation  
Interpolation  
Iteration and Convergence  
Solving One Equation  
Solving Many Equations  
Finding the Best: Optimization in 1D  
Optimization in  $n$  Dimensions