

# Eigenvalues and Eigenvectors

# Few concepts to remember from linear algebra

Let  $\mathbf{A}$  be an  $n \times m$  matrix and the linear transformation  $\mathbf{y} = \mathbf{A}\mathbf{x}$

$$\mathbf{x} \in \mathcal{R}^m \xrightarrow{\mathbf{A}} \mathbf{y} \in \mathcal{R}^n$$

- Rank: maximum number of linearly independent columns or rows of  $\mathbf{A}$
- $\text{Range}(\mathbf{A}) = \{\mathbf{y} = \mathbf{A}\mathbf{x} \mid \forall \mathbf{x}\}$
- $\text{Null}(\mathbf{A}) = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{0}\}$

# Eigenvalue problem

Let  $\mathbf{A}$  be an  $n \times n$  matrix:

$\mathbf{x} \neq \mathbf{0}$  is an eigenvector of  $\mathbf{A}$  if there exists a scalar  $\lambda$  such that

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

where  $\lambda$  is called an eigenvalue.

If  $\mathbf{x}$  is an eigenvector, then  $\alpha \mathbf{x}$  is also an eigenvector. Therefore, we will usually seek for normalized eigenvectors, so that

$$\|\mathbf{x}\| = 1$$

Note: When using Python, `numpy.linalg.eig` will normalize using  $p=2$  norm.

# How do we find eigenvalues?

Linear algebra approach:

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

$$(\mathbf{A} - \lambda \mathbf{I}) \mathbf{x} = \mathbf{0}$$

Therefore the matrix  $(\mathbf{A} - \lambda \mathbf{I})$  is singular  $\implies \det(\mathbf{A} - \lambda \mathbf{I}) = 0$

$p(\lambda) = \det(\mathbf{A} - \lambda \mathbf{I})$  is the characteristic polynomial of degree  $n$ .

In most cases, there is no analytical formula for the eigenvalues of a matrix (Abel proved in 1824 that there can be no formula for the roots of a polynomial of degree 5 or higher)  $\implies$  **Approximate the eigenvalues numerically!**

# Example

$$\mathbf{A} = \begin{pmatrix} 2 & 1 \\ 4 & 2 \end{pmatrix} \quad \det \begin{pmatrix} 2 - \lambda & 1 \\ 4 & 2 - \lambda \end{pmatrix} = 0$$

Solution of characteristic polynomial gives:  $\lambda_1 = 4, \lambda_2 = 0$

To get the eigenvectors, we solve:  $\mathbf{A} \mathbf{x} = \lambda \mathbf{x}$

$$\begin{pmatrix} 2 - (4) & 1 \\ 4 & 2 - (4) \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \mathbf{x} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$\begin{pmatrix} 2 - (0) & 1 \\ 4 & 2 - (0) \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \mathbf{x} = \begin{pmatrix} -1 \\ 2 \end{pmatrix}$$

## Notes:

The matrix  $\mathbf{A}$  is singular ( $\det(\mathbf{A})=0$ ), and  $\text{rank}(\mathbf{A})=1$

The matrix has two distinct real eigenvalues

The eigenvectors are linearly independent

# Diagonalizable Matrices

A  $n \times n$  matrix  $\mathbf{A}$  with  $n$  linearly independent eigenvectors  $\mathbf{u}$  is said to be **diagonalizable**.

$$\mathbf{A} \mathbf{u}_1 = \lambda_1 \mathbf{u}_1,$$

$$\mathbf{A} \mathbf{u}_2 = \lambda_2 \mathbf{u}_2,$$

...

$$\mathbf{A} \mathbf{u}_n = \lambda_n \mathbf{u}_n,$$

In matrix form:

$$\mathbf{A} (\mathbf{u}_1 \quad \dots \quad \mathbf{u}_n) = (\lambda_1 \mathbf{u}_1 \quad \dots \quad \lambda_n \mathbf{u}_n) = (\mathbf{u}_1 \quad \dots \quad \mathbf{u}_n) \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_n \end{pmatrix}$$

This corresponds to a similarity transformation

$$\mathbf{A} \mathbf{U} = \mathbf{U} \mathbf{D} \Leftrightarrow \mathbf{A} = \mathbf{U} \mathbf{D} \mathbf{U}^{-1}$$

**Example**  $\mathbf{A} = \begin{pmatrix} 2 & 1 \\ 4 & 2 \end{pmatrix} \quad \det \begin{pmatrix} 2 - \lambda & 1 \\ 4 & 2 - \lambda \end{pmatrix} = 0$

Solution of characteristic polynomial gives:  $\lambda_1 = 4, \lambda_2 = 0$

To get the eigenvectors, we solve:  $\mathbf{A} \mathbf{x} = \lambda \mathbf{x}$

$$\begin{pmatrix} 2 - (4) & 1 \\ 4 & 2 - (4) \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \mathbf{x} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad \text{or normalized} \quad \mathbf{x} = \begin{pmatrix} 0.447 \\ 0.894 \end{pmatrix}$$

eigenvector

$$\begin{pmatrix} 2 - (0) & 1 \\ 4 & 2 - (0) \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \mathbf{x} = \begin{pmatrix} -1 \\ 2 \end{pmatrix} \quad (p = 2 \text{ norm}) \quad \mathbf{x} = \begin{pmatrix} -0.447 \\ 0.894 \end{pmatrix}$$

$$\mathbf{A} = \mathbf{U} \mathbf{D} \mathbf{U}^{-1} \quad \mathbf{U} = \begin{pmatrix} 0.447 & -0.447 \\ 0.894 & 0.894 \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} 4 & 0 \\ 0 & 0 \end{pmatrix}$$

**Notes:**

The matrix  $\mathbf{A}$  is singular ( $\det(\mathbf{A})=0$ ), and  $\text{rank}(\mathbf{A})=1$

Since  $\mathbf{A}$  has two linearly independent eigenvectors, the matrix  $\mathbf{U}$  is full rank, and hence, the matrix  $\mathbf{A}$  is diagonalizable.

# Example

The eigenvalues of the matrix:

$$\mathbf{A} = \begin{pmatrix} 3 & -18 \\ 2 & -9 \end{pmatrix}$$

are  $\lambda_1 = \lambda_2 = -3$ .

Select the **incorrect** statement:

- A) Matrix  $\mathbf{A}$  is diagonalizable
- B) The matrix  $\mathbf{A}$  has only one eigenvalue with multiplicity 2
- C) Matrix  $\mathbf{A}$  has only one linearly independent eigenvector
- D) Matrix  $\mathbf{A}$  is not singular

# Let's look back at diagonalization...

- 1) If a  $n \times n$  matrix  $\mathbf{A}$  has  $n$  linearly independent eigenvectors  $\mathbf{x}$  then  $\mathbf{A}$  is diagonalizable, i.e.,

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{U}^{-1}$$

where the columns of  $\mathbf{U}$  are the linearly independent normalized eigenvectors  $\mathbf{x}$  of  $\mathbf{A}$  (which guarantees that  $\mathbf{U}^{-1}$  exists) and  $\mathbf{D}$  is a diagonal matrix with the eigenvalues of  $\mathbf{A}$ .

- 2) If a  $n \times n$  matrix  $\mathbf{A}$  has less than  $n$  linearly independent eigenvectors, the matrix is called defective (and therefore not diagonalizable).
- 3) If a  $n \times n$  **symmetric** matrix  $\mathbf{A}$  has  $n$  distinct eigenvalues then  $\mathbf{A}$  is diagonalizable.

A  $n \times n$  symmetric matrix  $A$  with  $n$  distinct eigenvalues is diagonalizable.

Suppose  $\lambda, \mathbf{u}$  and  $\mu, \mathbf{v}$  are eigenpairs of  $A$

$$\lambda \mathbf{u} = A\mathbf{u}$$

$$\mu \mathbf{v} = A\mathbf{v}$$

$$\lambda \mathbf{u} = A\mathbf{u} \rightarrow \mathbf{v} \cdot \lambda \mathbf{u} = \mathbf{v} \cdot A\mathbf{u}$$

$$\lambda \mathbf{v} \cdot \mathbf{u} = A^T \mathbf{v} \cdot \mathbf{u} = A\mathbf{v} \cdot \mathbf{u} = \mu \mathbf{v} \cdot \mathbf{u} \rightarrow (\mu - \lambda) \mathbf{v} \cdot \mathbf{u} = 0$$

If all  $n$  eigenvalues are distinct  $\rightarrow (\mu - \lambda) \neq 0$

Hence,  $\mathbf{v} \cdot \mathbf{u} = 0$ , i.e., the eigenvectors are orthogonal (linearly independent), and consequently the matrix  $A$  is diagonalizable.

Note that a diagonalizable matrix  $A$  does not guarantee  $n$  distinct eigenvalues.

# Some things to remember about eigenvalues:

- Eigenvalues can have zero value
- Eigenvalues can be negative
- Eigenvalues can be real or complex numbers
- A  $n \times n$  real matrix can have complex eigenvalues
- The eigenvalues of a  $n \times n$  matrix are not necessarily unique. In fact, we can define the multiplicity of an eigenvalue.
- If a  $n \times n$  matrix has  $n$  linearly independent eigenvectors, then the matrix is diagonalizable

# How can we get eigenvalues numerically?

Assume that  $\mathbf{A}$  is diagonalizable (i.e., it has  $n$  linearly independent eigenvectors  $\mathbf{u}$ ). We can propose a vector  $\mathbf{x}$  which is a linear combination of these eigenvectors:

$$\mathbf{x} = \alpha_1 \mathbf{u}_1 + \alpha_2 \mathbf{u}_2 + \cdots + \alpha_n \mathbf{u}_n$$

Then we evaluate  $\mathbf{A} \mathbf{x}$ :

$$\mathbf{A} \mathbf{x} = \alpha_1 \mathbf{A} \mathbf{u}_1 + \alpha_2 \mathbf{A} \mathbf{u}_2 + \cdots + \alpha_n \mathbf{A} \mathbf{u}_n$$

And since  $\mathbf{A} \mathbf{u}_1 = \lambda_1 \mathbf{u}_1$  we can also write:

$$\mathbf{A} \mathbf{x} = \alpha_1 \lambda_1 \mathbf{u}_1 + \alpha_2 \lambda_2 \mathbf{u}_2 + \cdots + \alpha_n \lambda_n \mathbf{u}_n$$

where  $\lambda_i$  is the eigenvalue corresponding to eigenvector  $\mathbf{u}_i$  and we *assume*

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$$

# Power Iteration

Our goal is to find an eigenvector  $\mathbf{u}_i$  of  $\mathbf{A}$ . We will use an iterative process, where we start with an initial vector, where here we assume that it can be written as a linear combination of the eigenvectors of  $\mathbf{A}$ .

$$\mathbf{x}_0 = \alpha_1 \mathbf{u}_1 + \alpha_2 \mathbf{u}_2 + \cdots + \alpha_n \mathbf{u}_n$$

And multiply by  $\mathbf{A}$  to get:

$$\begin{aligned}\mathbf{x}_1 &= \mathbf{A} \mathbf{x}_0 = \alpha_1 \lambda_1 \mathbf{u}_1 + \alpha_2 \lambda_2 \mathbf{u}_2 + \cdots + \alpha_n \lambda_n \mathbf{u}_n \\ \mathbf{x}_2 &= \mathbf{A} \mathbf{x}_1 = \alpha_1 (\lambda_1)^2 \mathbf{u}_1 + \alpha_2 (\lambda_2)^2 \mathbf{u}_2 + \cdots + \alpha_n (\lambda_n)^2 \mathbf{u}_n \\ &\quad \vdots \\ \mathbf{x}_k &= \mathbf{A} \mathbf{x}_{k-1} = \alpha_1 (\lambda_1)^k \mathbf{u}_1 + \alpha_2 (\lambda_2)^k \mathbf{u}_2 + \cdots + \alpha_n (\lambda_n)^k \mathbf{u}_n\end{aligned}$$

Or rearranging...

$$\mathbf{x}_k = (\lambda_1)^k \left[ \alpha_1 \mathbf{u}_1 + \alpha_2 \left( \frac{\lambda_2}{\lambda_1} \right)^k \mathbf{u}_2 + \cdots + \alpha_n \left( \frac{\lambda_n}{\lambda_1} \right)^k \mathbf{u}_n \right]$$

# Power Iteration

$$\mathbf{x}_k = (\lambda_1)^k \left[ \alpha_1 \mathbf{u}_1 + \alpha_2 \left( \frac{\lambda_2}{\lambda_1} \right)^k \mathbf{u}_2 + \cdots + \alpha_n \left( \frac{\lambda_n}{\lambda_1} \right)^k \mathbf{u}_n \right]$$

Assume that  $\alpha_1 \neq 0$ , the term  $\alpha_1 \mathbf{u}_1$  dominates the others when  $k$  is very large.

Since  $|\lambda_1| > |\lambda_2|$ , we have  $\left( \frac{\lambda_2}{\lambda_1} \right)^k \ll 1$  when  $k$  is large

Hence, as  $k$  increases,  $\mathbf{x}_k$  converges to a multiple of the first eigenvector  $\mathbf{u}_1$ , i.e.,

$$\lim_{k \rightarrow \infty} \frac{\mathbf{x}_k}{(\lambda_1)^k} = \alpha_1 \mathbf{u}_1 \quad \text{or} \quad \mathbf{x}_k \rightarrow \alpha_1 (\lambda_1)^k \mathbf{u}_1$$

# How can we now get the eigenvalues?

If  $\mathbf{x}$  is an eigenvector of  $\mathbf{A}$  such that

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

then how can we evaluate the corresponding eigenvalue  $\lambda$ ?

$$\lambda = \frac{\mathbf{x}^T \mathbf{A} \mathbf{x}}{\mathbf{x}^T \mathbf{x}}$$

Rayleigh coefficient

# Normalized Power Iteration

$$\mathbf{x}_k = (\lambda_1)^k \left[ \alpha_1 \mathbf{u}_1 + \alpha_2 \left( \frac{\lambda_2}{\lambda_1} \right)^k \mathbf{u}_2 + \cdots + \alpha_n \left( \frac{\lambda_n}{\lambda_1} \right)^k \mathbf{u}_n \right]$$

$\mathbf{x}_0$  = arbitrary nonzero vector

$$\mathbf{x}_0 = \frac{\mathbf{x}_0}{\|\mathbf{x}_0\|}$$

for  $k = 1, 2, \dots$

$$\mathbf{y}_k = \mathbf{A} \mathbf{x}_{k-1}$$

$$\mathbf{x}_k = \frac{\mathbf{y}_k}{\|\mathbf{y}_k\|}$$

# Normalized Power Iteration

$$\mathbf{x}_k = (\lambda_1)^k \left[ \alpha_1 \mathbf{u}_1 + \alpha_2 \left( \frac{\lambda_2}{\lambda_1} \right)^k \mathbf{u}_2 + \cdots + \alpha_n \left( \frac{\lambda_n}{\lambda_1} \right)^k \mathbf{u}_n \right]$$

What if the starting vector  $\mathbf{x}_0$  have no component in the dominant eigenvector  $\mathbf{u}_1$  ( $\alpha_1 = 0$ )?

# Normalized Power Iteration

$$\mathbf{x}_k = (\lambda_1)^k \left[ \alpha_1 \mathbf{u}_1 + \alpha_2 \left( \frac{\lambda_2}{\lambda_1} \right)^k \mathbf{u}_2 + \cdots + \alpha_n \left( \frac{\lambda_n}{\lambda_1} \right)^k \mathbf{u}_n \right]$$

What if the first two largest eigenvalues (in magnitude) are the same,  $|\lambda_1| = |\lambda_2|$ ?

$$\mathbf{x}_k = (\lambda_1)^k \alpha_1 \mathbf{u}_1 + (\lambda_1)^k \left( \frac{\lambda_2}{\lambda_1} \right)^k \alpha_2 \mathbf{u}_2 + (\lambda_1)^k \left[ \cdots + \alpha_n \left( \frac{\lambda_n}{\lambda_1} \right)^k \mathbf{u}_n \right]$$

# Potential pitfalls

1. Starting vector  $\mathbf{x}_0$  may have no component in the dominant eigenvector  $\mathbf{u}_1$  ( $\alpha_1 = 0$ ). This is usually unlikely to happen if  $\mathbf{x}_0$  is chosen randomly, and in practice not a problem because rounding will usually introduce such component.
2. Risk of eventual overflow (or underflow): in practice the approximated eigenvector is normalized at each iteration (Normalized Power Iteration)
3. First two largest eigenvalues (in magnitude) may be the same:  $|\lambda_1| = |\lambda_2|$ . In this case, power iteration will give a vector that is a linear combination of the corresponding eigenvectors:
  - If signs are the same, the method will converge to correct magnitude of the eigenvalue. If the signs are different, the method will not converge.
  - This is a “real” problem that cannot be discounted in practice.

# Error

$$\mathbf{x}_k = (\lambda_1)^k \left[ \alpha_1 \mathbf{u}_1 + \underbrace{\alpha_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k \mathbf{u}_2 + \cdots + \alpha_n \left(\frac{\lambda_n}{\lambda_1}\right)^k \mathbf{u}_n}_{\text{Error}} \right]$$

We can see from the above that the rate of convergence depends on the ratio  $\left|\frac{\lambda_2}{\lambda_1}\right|$ , that is:

$$\|(\lambda_1)^{-k} \mathbf{x}_k - \alpha_1 \mathbf{u}_1\| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)$$

# Convergence and error

$$\mathbf{x}_k = \mathbf{u}_1 + \underbrace{\left(\frac{\alpha_2}{\alpha_1}\right) \left(\frac{\lambda_2}{\lambda_1}\right)^k}_{\mathbf{e}_k} \mathbf{u}_2 + \dots$$

$$\frac{\mathbf{e}_{k+1}}{\mathbf{e}_k} \approx \left| \frac{\lambda_2}{\lambda_1} \right|$$

Power method has linear convergence, which is quite slow.

# Clicker question

Suppose you are given a matrix with eigenvalues 3, 4, and 5. You use (normalized) power iteration to approximate one of the eigenvectors  $\|\mathbf{x}\|$ . For simplicity, assume  $\|\mathbf{x}\| = 1$ . Your initial guess  $\mathbf{x}_0$  has a norm of the error  $\|\mathbf{x} - \mathbf{x}_0\| = 0.3$ .

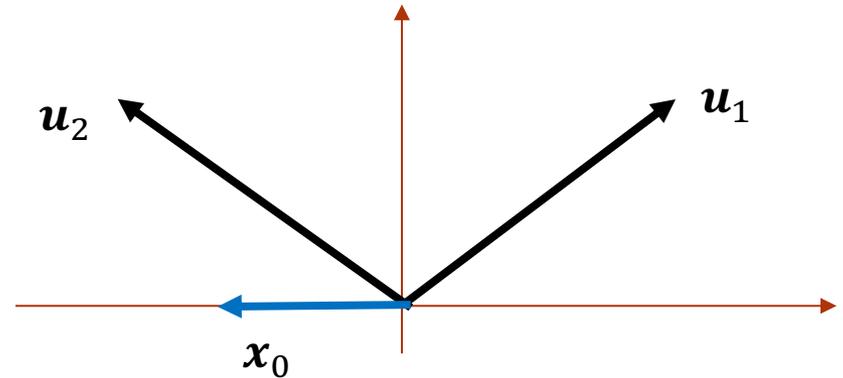
How big will the error be after three rounds of normalized power iteration?

(Note that for normalized power iteration, all vectors under consideration have norm 1, so the absolute and the relative error are the same.)

- A)* 0.1536
- B)* 0.192
- C)* 0.09
- D)* 0.027

# Clicker question

The matrix  $\mathbf{A} = \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}$  has eigenvalues  $(4, 2)$  and corresponding eigenvectors  $\mathbf{u}_1 = (1, 1)$  and  $\mathbf{u}_2 = (-1, 1)$ .



Suppose we want to use the normalized power iteration, starting from  $\mathbf{x}_0 = (-0.5, 0)$ . Select the correct statement

- A) Normalized power iteration will not converge
- B) Normalized power iteration will converge to the eigenvector corresponding to the eigenvalue 2.
- C) Normalized power iteration will converge to the eigenvector corresponding to the eigenvalue 4.

# Clicker question

Suppose  $\mathbf{x}$  is an eigenvector of  $\mathbf{A}$  such that

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

What is an eigenvalue of  $\mathbf{A}^{-1}$ ?

A)  $\lambda$

B)  $-\lambda$

C)  $1/\lambda$

D)  $-\frac{1}{\lambda}$

E) Can't tell without knowing  $\lambda$

# Inverse Power Method

Previously we learned that we can use the Power Method to obtain the largest eigenvalue and corresponding eigenvector, by using the update

$$\mathbf{x}_{k+1} = \mathbf{A} \mathbf{x}_k$$

Suppose there is a single smallest eigenvalue of  $\mathbf{A}$ . With the previous ordering

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots > |\lambda_n|$$

the smallest eigenvalue is  $\lambda_n$ . When computing the eigenvalues of the inverse matrix  $\mathbf{A}^{-1}$ , we get the following ordering

$$\left| \frac{1}{\lambda_n} \right| > \left| \frac{1}{\lambda_{n-1}} \right| \geq \dots \geq \left| \frac{1}{\lambda_1} \right|$$

And hence we can use the Power Method update on the matrix  $\mathbf{A}^{-1}$  to compute the dominant eigenvalue  $\frac{1}{\lambda_n}$ , i.e.,

$$\mathbf{x}_{k+1} = \mathbf{A}^{-1} \mathbf{x}_k$$

# Clicker question

Which code snippet is the best option to compute the smallest eigenvalue of the matrix  $A$ ?

A) 

```
x = x0/la.norm(x0)
for k in range(30):
    x = la.solve(A, x)
    x = x/la.norm(x)
```

B) 

```
x = x0/la.norm(x0)
for k in range(30):
    x = la.inv(A)@x
    x = x/la.norm(x)
```

C) 

```
x = x0/la.norm(x0)
for k in range(30):
    P, L, U = sla.lu(A)
    y = sla.solve_triangular(L, np.dot(P.T, x), lower=True)
    x = sla.solve_triangular(U, y)
    x = x/la.norm(x)
```

D) 

```
x = x0/la.norm(x0)
P, L, U = sla.lu(A)
for k in range(30):
    y = sla.solve_triangular(L, np.dot(P.T, x), lower=True)
    x = sla.solve_triangular(U, y)
    x = x/la.norm(x)
```

E) I have no idea!

# Inverse Power Method

Note that the update

$$\mathbf{x}_{k+1} = \mathbf{A}^{-1} \mathbf{x}_k$$

can be instead written as

$$\mathbf{A} \mathbf{x}_{k+1} = \mathbf{x}_k$$

Where  $\mathbf{x}_k$  is known and we need to solve for  $\mathbf{x}_{k+1}$  (we are just solving a linear system of equations!). Since the matrix  $\mathbf{A}$  does not change from iteration to the next, we can factorize the matrix once and then perform a series of backward and forward substitutions.

Recall  $\mathbf{PA} = \mathbf{LU}$  and  $\mathbf{A} \mathbf{x} = \mathbf{b}$  resulting in  $\mathbf{LU} \mathbf{x} = \mathbf{Pb}$

Hence we can efficiently solve

$$\begin{aligned} \mathbf{L} \mathbf{y} &= \mathbf{P} \mathbf{x}_k \\ \mathbf{U} \mathbf{x}_{k+1} &= \mathbf{y} \end{aligned}$$

# Cost of computing eigenvalues using inverse power iteration

```
x = x0/la.norm(x0)
for k in range(30):
    x = la.solve(A, x)
    x = x/la.norm(x)
```

```
x = x0/la.norm(x0)
for k in range(30):
    x = la.inv(A)@x
    x = x/la.norm(x)
```

```
x = x0/la.norm(x0)
for k in range(30):
    P, L, U = sla.lu(A)
    y = sla.solve_triangular(L, np.dot(P.T, x), lower=True)
    x = sla.solve_triangular(U, y)
    x = x/la.norm(x)
```

```
x = x0/la.norm(x0)
P, L, U = sla.lu(A)
for k in range(30):
    y = sla.solve_triangular(L, np.dot(P.T, x), lower=True)
    x = sla.solve_triangular(U, y)
    x = x/la.norm(x)
```

# Clicker question

What is the approximated cost of computing the largest eigenvalue using Power Method?

*A)  $k n$*

*B)  $n^2 + k n$*

*C)  $k n^2$*

*D)  $n^3 + k n^2$*

*E) NOT A*

# Clicker question

Suppose  $\mathbf{x}$  is an eigenvector of  $\mathbf{A}$  such that  $\mathbf{A}\mathbf{x} = \lambda_1\mathbf{x}$  and also  $\mathbf{x}$  is an eigenvector of  $\mathbf{B}$  such that  $\mathbf{B}\mathbf{x} = \lambda_2\mathbf{x}$ . What is an eigenvalue of

What is an eigenvalue of  $(\mathbf{A} + \frac{1}{2}\mathbf{B})^{-1}$ ?

A)  $\frac{\lambda_1}{2\lambda_1 + \lambda_2}$

B)  $\frac{\lambda_2}{2\lambda_1 + \lambda_2}$

C)  $\frac{2}{2\lambda_1 + \lambda_2}$

D)  $\frac{\lambda_1}{2\lambda_2 + \lambda_1}$

E)  $\frac{\lambda_2}{2\lambda_2 + \lambda_1}$

# Iclicker question

Suppose  $\mathbf{x}$  is an eigenvector of  $\mathbf{A}$  such that  $\mathbf{A} \mathbf{x} = \lambda \mathbf{x}$ , but  $\lambda$  is not the largest or smallest eigenvalue. We want to compute the eigenvalue  $\lambda$  that is close to a given number  $\sigma$ . Which of the following modified matrices will give such eigenvalue?

A)  $(\mathbf{A} - \sigma \mathbf{I})$

B)  $(\mathbf{A} - \sigma \mathbf{I})^{-1}$

C)  $(1 - \sigma) \mathbf{A}$

D)  $\frac{1}{\sigma} \mathbf{A}$

E) *I still have no clue how to answer to these iclicker questions...*

# Eigenvalues of a Shifted Inverse Matrix

Suppose the eigenpairs  $(\mathbf{x}, \lambda)$  satisfy  $\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$ .

We can describe the eigenvalues for the shifted inverse matrix as

$$(\mathbf{A} - \sigma\mathbf{I})^{-1}\mathbf{x} = \bar{\lambda} \mathbf{x}$$

$$\mathbf{I}\mathbf{x} = \bar{\lambda}(\lambda\mathbf{I} - \sigma\mathbf{I})\mathbf{x}$$

$$\bar{\lambda} = \frac{1}{\lambda - \sigma}$$

Hence the eigensystem problem is

$$(\mathbf{A} - \sigma\mathbf{I})^{-1}\mathbf{x} = \frac{1}{\lambda - \sigma} \mathbf{x}$$

# Eigenvalues of a Shifted Inverse Matrix

We use the update

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

To obtain the eigenpair  $(\mathbf{x}, \lambda)$  that satisfy  $\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$  such that  $\lambda$  is an eigenvalue close to the number  $\sigma$

We can factorize the matrix  $\mathbf{B} = (\mathbf{A} - \sigma \mathbf{I})$  such that  $\mathbf{PB} = \mathbf{LU}$  and then efficiently solve

$$\begin{aligned} \mathbf{L} \mathbf{y} &= \mathbf{P} \mathbf{x}_k \\ \mathbf{U} \mathbf{x}_{k+1} &= \mathbf{y} \end{aligned}$$

# Convergence summary

	Method	Cost	Convergence $\ e_{k+1}\ /\ e_k\ $
Power Method	$\mathbf{x}_{k+1} = \mathbf{A} \mathbf{x}_k$	$k n^2$	$\left  \frac{\lambda_2}{\lambda_1} \right $
Inverse Power Method	$\mathbf{A} \mathbf{x}_{k+1} = \mathbf{x}_k$	$n^3 + k n^2$	$\left  \frac{\lambda_n}{\lambda_{n-1}} \right $
Shifted Inverse Power Method	$(\mathbf{A} - \sigma \mathbf{I}) \mathbf{x}_{k+1} = \mathbf{x}_k$	$n^3 + k n^2$	$\left  \frac{\lambda_c - \sigma}{\lambda_{c2} - \sigma} \right $

$\lambda_1$ : largest eigenvalue (in magnitude)

$\lambda_2$ : second largest eigenvalue (in magnitude)

$\lambda_n$ : smallest eigenvalue (in magnitude)

$\lambda_{n-1}$ : second smallest eigenvalue (in magnitude)

$\lambda_c$ : closest eigenvalue to  $\sigma$

$\lambda_{c2}$ : second closest eigenvalue to  $\sigma$