Eigenvalues and Eigenvectors

Power Iteration

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

Assume that $\alpha_1 \neq 0$, the term $\alpha_1 \boldsymbol{u}_1$ 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, $\boldsymbol{x}_k$ converges to a multiple of the first eigenvector $\boldsymbol{u}_1$, i.e.,

How can we now get the eigenvalues?

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

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

then how can we evaluate the corresponding eigenvalue λ?

Power Iteration

Normalized Power Iteration

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

$\boldsymbol{x_0}$ = arbitrary nonzero vector

$$\boldsymbol{x_0} = \frac{\boldsymbol{x_0}}{\|\boldsymbol{x_0}\|}$$

for $k = 1,2, \ldots$

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

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

Normalized Power Iteration

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

What if the starting vector $\boldsymbol{x_0}$ have no component in the dominant eigenvector $\boldsymbol{u}_1$ ($\alpha_1 = 0$)?

Normalized Power Iteration

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

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

1) λ_1 and λ_2 both positives

Normalized Power Iteration

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

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

2) λ_1 and λ_2 both negative

Normalized Power Iteration

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

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

3) λ_1 and λ_2 opposite signs

Potential pitfalls

1. Starting vector $\boldsymbol{x_0}$ may have no component in the dominant eigenvector $\boldsymbol{u}_1$ ($\alpha_1 = 0$). This is usually unlikely to happen if $\boldsymbol{x_0}$ 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

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

Convergence and error

Example

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.)

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

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

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

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

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

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

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

Think about this question...

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

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

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)
```

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

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

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

What is an eigenvalue of $\boldsymbol{A}^2 + \sigma\boldsymbol{B}$?

Eigenvalues of a Shifted Inverse Matrix

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

Eigenvalues of a Shifted Inverse Matrix

Convergence summary

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

λ_1: largest eigenvalue (in magnitude)
λ_2: second largest eigenvalue (in magnitude)
λ_n: smallest eigenvalue (in magnitude)
λ_{n-1}: second smallest eigenvalue (in magnitude)
λ_c: closest eigenvalue to σ
λ_{c2}: second closest eigenvalue to σ