Lecture #25

03/10/2023

Inverse Power Iteration:

Smallest magnitude eig. value & the corresponding eig. vector

- Assume that A is nonsingular $\iff \lambda_i \neq 0$ for all $i = 1, \ldots, n$

IDEA:

$$\left.\begin{aligned} \lambda_i(A^{-1}) &= \frac{1}{\lambda_i(A)} \\ \text{and} \quad \underline{v}^{(i)}(A^{-1}) &= \underline{v}^{(i)}(A) \end{aligned}\right\}$$

Why?

$$A\underline{v} = \lambda \underline{v} \Rightarrow \underline{v} = \lambda A^{-1} \underline{v} \Rightarrow \boxed{\left(\frac{1}{\lambda}\right) \underline{v} = A^{-1} \underline{v}}$$

$\therefore$ We can simply apply the power iteration algorithm to the matrix A^{-1}

since $\lambda_n(A) = \lambda_1(A^{-1})$

$\therefore$ we should simply do: $\underline{x}_{k+1} = \dfrac{A^{-1}\underline{x}_k}{\|A^{-1}\underline{x}_k\|_2}$

To avoid computing A^{-1}:

$$\left.\begin{aligned} \underline{y}_{k+1} &= A \backslash \underline{x}_k ; \\ \underline{x}_{k+1} &= \underline{y}_{k+1} / \|\underline{y}_{k+1}\|_2 ; \end{aligned}\right\}$$

call: $\underline{y}_{k+1} := A^{-1}\underline{x}_k$

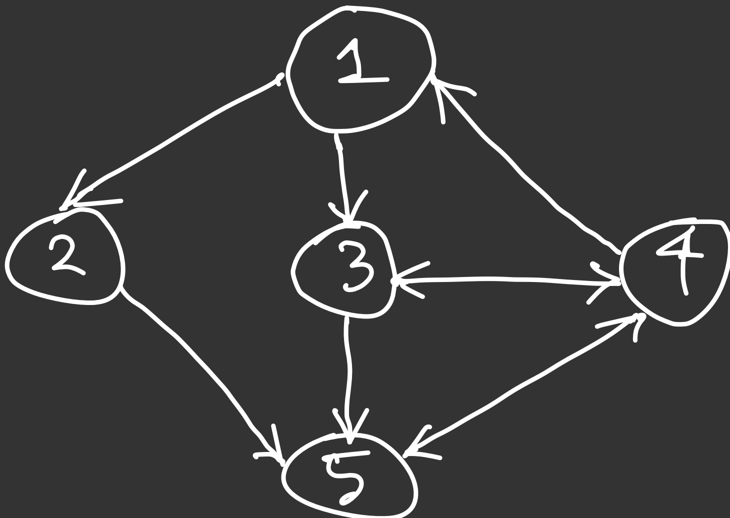
$\Updownarrow$

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

e.g., by LU decomposition

Case study on the application of power iteration:

PageRank Algorithm:



node $\equiv$ webpage ($\neq$ website)

$\text{node } i \xrightarrow[\text{(edge)}]{} \text{node } j$

webpage i contains URL to webpage j, with a single click

called "Directed graph" $\equiv$ a collection of nodes and directed edges/arrows

<u>IDEA:</u> A webpage with more incoming links should be ranked higher/more popular/relevant

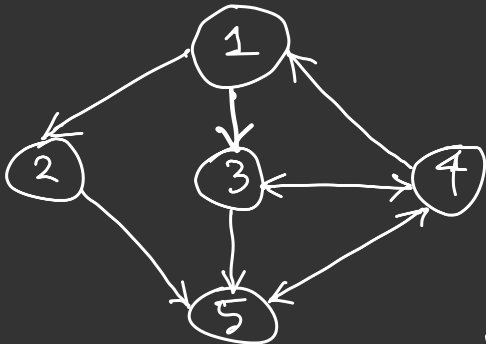
<u>Question:</u> Given a collection of n nodes and edges/arrows between them, what is the pagerank/relative importance vector $\underline{p}$ such that

$$\underbrace{\underline{p} \geq \underline{0}}_{\text{elementwise}}, \quad \mathbb{1}^T \underline{p} = 1$$

} element sum of $\underline{p}$ vector $= 1$

vector of ones $\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$

<u>Example:</u> $\underline{p} = \begin{pmatrix} 0.35 \\ 0.05 \\ 0.20 \\ 0.10 \\ 0.30 \end{pmatrix}$



Given the directed graph, compute

$$\underbrace{A}_{\substack{n \times n \\ \text{matrix}}} = \begin{cases} 1 & \text{if } j \longrightarrow i \text{ exists} \\ 0 & \text{otherwise} \end{cases}$$

This matrix is called "Adjacency matrix" associated with the graph

Example:

$$\underbrace{A}_{5 \times 5} = \begin{array}{cc} & \text{FROM} \\ \text{TO} & \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

of size # of nodes x # of nodes

in-degree = # of incoming arrows to each node

$$= A\mathbb{1} = \text{row sum of } A$$

$$= \begin{pmatrix} 1 \\ 1 \\ 2 \\ 2 \\ 3 \end{pmatrix}$$

out-degree = # of outgoing arrows from each node

$$= \mathbb{1}^T A = \text{column sum of } A$$

$$= \begin{pmatrix} 2 & 1 & 2 & 3 & 1 \end{pmatrix}$$

Now we divide each column of A by its column sum:

$\Longleftrightarrow$ re-scale the total out-degree to be unity

$$A_{ij} \mapsto S_{ij} = \frac{A_{ij}}{\sum_i A_{ij}}$$

In our example:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} \mapsto S = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1/3 & 0 \\ 1/2 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 1/2 & 0 & 1 \\ 0 & 1 & 1/2 & 1/3 & 0 \end{bmatrix}}_{\text{column-normalized version of } A}$$

If there exists a "dangling node" which has NO outgoing links, then:

$$A = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & 0 \\ \vdots & \vdots & \vdots & \vdots & 0 \\ \vdots & \vdots & \vdots & \vdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & 0 \end{bmatrix}$$

all zeros column

If we ever end up in a webpage which has NO outgoing links, then make a uniformly random jump to any other webpage.

$$S = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & 1/n \\ \vdots & \vdots & \vdots & \vdots & 1/n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & 1/n \end{bmatrix}$$

random surfer model

uniform probability/likelihood

Matrix S has some properties:

$$S_{ij} \geq 0 \quad \text{for all } i, j = 1, \ldots, n$$

and $$\underbrace{\sum_i S_{ij} = 1}_{\text{each column sum} = 1} \quad \text{for all } j = 1, \ldots, n$$

Such a matrix S is called a "column stochastic matrix"

next pg.

Matrix S has probabilistic interpretation:

Suppose $\underline{p}_0 \leftarrow$ arbitrary weight/probability vector.

Now consider the matrix-vector recursion:

$$\underbrace{\underbrace{\underline{p}_{k+1} = \underset{\substack{\uparrow \\ \text{Transition probability matrix}}}{S}\ \underline{p}_k}_{\text{Markov chain}}}_{\text{linear matrix-vector } \underline{\text{fixed point recursion}}}$$

Question: What happens to this recursion as $k \to \infty$?

If it converges then $\underbrace{\underline{p} = S\underline{p}}_{\text{fixed point equation}}$

$\therefore$ Solving for $\underline{p}$ from $\underline{p} = S\underline{p}$ is the same as saying "compute the eig. vector $\underline{p}$ associated with eig. value 1".

Question: But how do we convince ourselves that S has 1 as an eig. value?

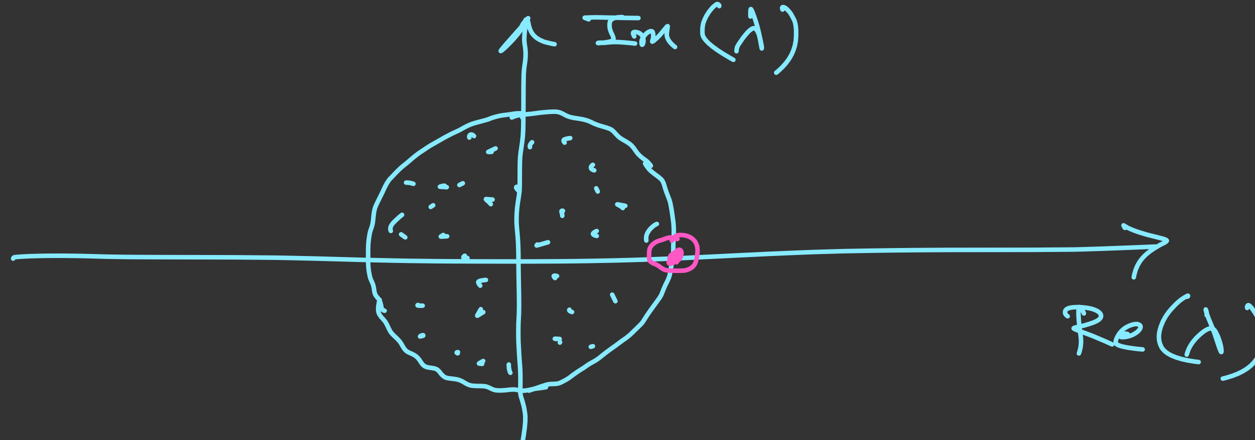
Theorem: Consider any column stochastic matrix $S \in \mathbb{R}^{n \times n}$

Then:

- S has an eig. value 1
- All eig. values of S must have magnitude / modulus ≤ 1

$\Leftrightarrow$ spectral radius of S is

$$\rho(S) := \max_{i=1,\dots,n} |\lambda_i| = 1$$



However, we need additional assumptions on S to guarantee that the eig. value 1 is simple (i.e., has algebraic multiplicity = 1).