Linear Algebra hw4
PageRank

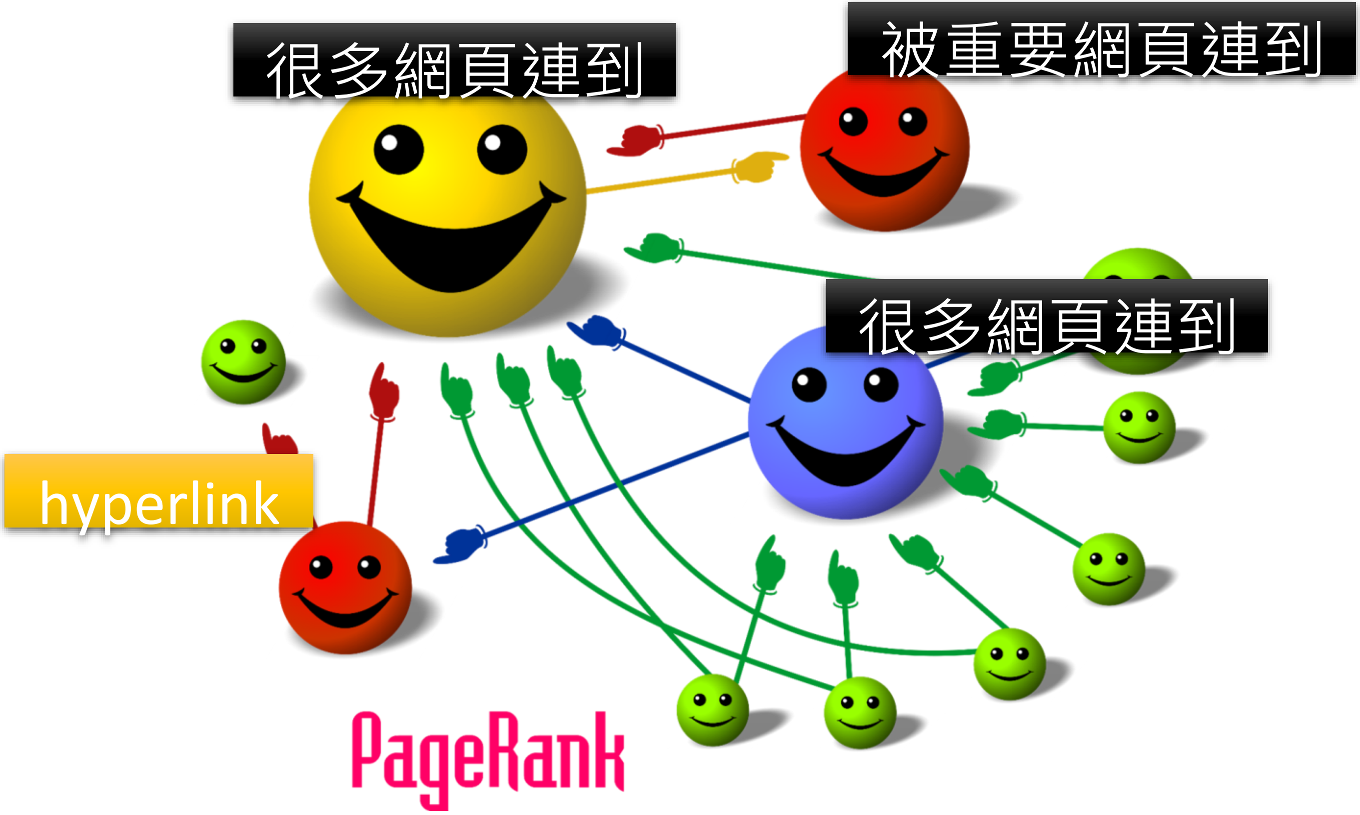
呂元翔

Introduction - PageRank

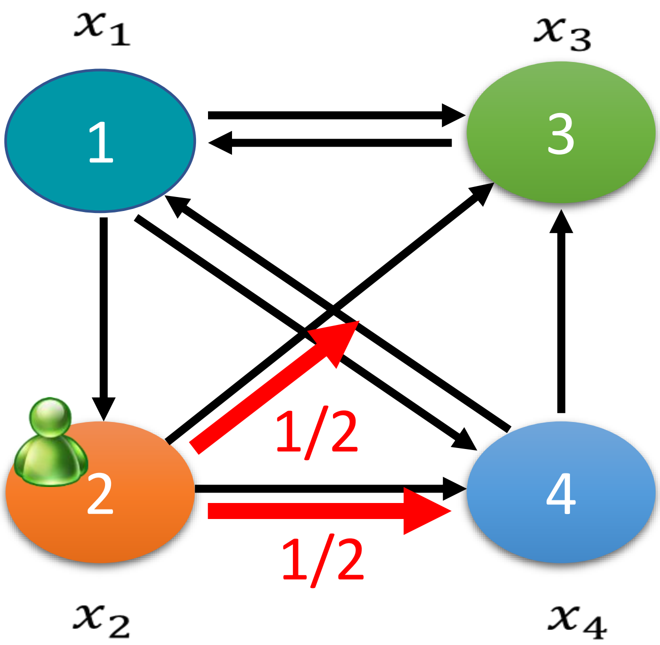
- PageRank is an algorithm to measure the importance of a website.
- It was developed by Larry Page and Sergey Brin in 1996.
- **The more links a website received from other websites, the more important it is.**
- The algorithm iteratively calculate the importance of a website according to the websites that have links to it and their importance.

教授上課影片: Page Rank

Importance



Importance - Formulas



$$x_1 = x_3 + \frac{1}{2}x_4$$

$$x_2 = \frac{1}{3}x_1$$

$$x_3 = \frac{1}{3}x_1 + \frac{1}{2}x_2 + \frac{1}{2}x_4$$

$$x_4 = \frac{1}{3}x_1 + \frac{1}{2}x_2$$

Importance - Formulas

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$Ax = x$$

$$x_1 = x_3 + \frac{1}{2}x_4$$

$$x_2 = \frac{1}{3}x_1$$

$$x_3 = \frac{1}{3}x_1 + \frac{1}{2}x_2 + \frac{1}{2}x_4$$

$$x_4 = \frac{1}{3}x_1 + \frac{1}{2}x_2$$

The solution x is in the eigenspace of eigenvalue $\lambda = 1$

Does it has a unique ranking solution?

1. **There doesn't exist a eigenvalue λ of the matrix that equals 1.**

   Q : How to ensure there exists a λ = 1 ?

   Sol: Use Column-Stochastic Matrix

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{1}{2} \\ 0 & 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

An example of column-stochastic matrix

summation = 1

non-negative

1. **The dimension of eigenspace is > 1**

   Q : How to ensure the dimension of the eigenspace = 1?

   Sol : Use Column-Stochastic matrix and "Positive"

$$\mathbf{M} = (1-m)\mathbf{A} + m\mathbf{S}$$

$$\begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1n} \\ S_{21} & S_{22} & \cdots & S_{2n} \\ \vdots & \vdots & & \vdots \\ S_{n1} & S_{n2} & \cdots & S_{nn} \end{bmatrix} = \begin{bmatrix} 1/N & & 1/N \\ 1/N & & 1/N \\ & \ddots & \\ 1/N & & 1/N \end{bmatrix}$$

ref : https://en.wikipedia.org/wiki/Stochastic_matrix

Power method

Find x^*, such that $x^* = Mx^*$

M is very large

$$x^1 = Mx^0$$

$$x^2 = Mx^1$$

$$\vdots$$

$$x^k = Mx^{k-1}$$

Start from x^0

$$x^0 = \begin{bmatrix} 1/n \\ \vdots \\ 1/n \end{bmatrix}$$

all positive, sum to 1

If $k \to \infty$

$$x_k = x^*$$

HW4

$$\mathbf{R} = \begin{bmatrix} \text{PageRank}(p_1) \\ \text{PageRank}(p_2) \\ \vdots \\ \text{PageRank}(p_N) \end{bmatrix}$$

$\ell(p_i, p_j)$ 1/#ofOutGoingLinksfromPj, if p_j has a link to p_i
0, else

$$\mathbf{R} = \left[(1-d) \begin{bmatrix} 1/N & & 1/N \\ 1/N & \ddots & 1/N \\ 1/N & & 1/N \end{bmatrix} + d \begin{bmatrix} \ell(p_1,p_1) & \ell(p_1,p_2) & \cdots & \ell(p_1,p_N) \\ \ell(p_2,p_1) & \ddots & & \\ \vdots & & \ell(p_i,p_j) & \\ \ell(p_N,p_1) & & & \ell(p_N,p_N) \end{bmatrix} \right] \mathbf{R}$$

uniform

Original transition matrix

HW4

Initialize R with $\begin{bmatrix} 1/N \\ 1/N \\ \vdots \\ 1/N \end{bmatrix}$. **In hw4 we set d = 0.85 and repeat the multiplication for 50 times**

Repeat

$$\mathbf{R}' = \left[(1-d)\begin{bmatrix} 1/N & & 1/N \\ 1/N & & 1/N \\ & \ddots & \\ 1/N & & 1/N \end{bmatrix} + d \begin{bmatrix} \ell(p_1,p_1) & \ell(p_1,p_2) & \cdots & \ell(p_1,p_N) \\ \ell(p_2,p_1) & \ddots & & \\ \vdots & & \ell(p_i,p_j) & \\ \ell(p_N,p_1) & & & \ell(p_N,p_N) \end{bmatrix} \right] \mathbf{R}$$

uniform

Original transition matrix

Until it R converge.

HW4

colab link: https://colab.research.google.com/drive/1mRRYa-2BB9T36g0apvveusbDPZ3m4hD_?usp=sharing

Implement the PageRank algorithm and turn a 2-dimensional list into the ordering of web pages.

Download a repository called “ `Linear_Algebra_2023_fall_hw4` “ from the github link in colab.

1.
```
❯ ls -R
PageRank.py     inputs          outputs

./inputs:
summary1.json   summary2.json

./outputs:
answer1.json    answer2.json
```

1. You will get a 2-dimensional list, describing the topology of a network.
   - E.g. [[0,3],[1,2,3],[2],[0,1]]. The first list [0,3] in this list means the webpage with id 0 has incoming links from 0 and 3. The second list [1,2,3] is the websites webpage 1 has links from, etc.
   - Our list have a length of 300. Each sublist has at least a number from 0~299.

HW4

3. You will also get a PageRank.py python file that helps you start. Try

    $python PageRank.py <input_file> <output_file>

    Your can name your <output_file> whatever you like, but need to be a .json file.

   Modify the PageRank.py file so that it outputs the correct ordering.

   Your <output_file> contains a list, like [2,0,3,1], which should be the ordering of the web pages based on pagerank. The web page with a higher pagerank score comes first.

   Float will be acceptable [2.0, 0.0, 3.0,1.0].

Grading

PageRankpy (6%)

- 不要抄作業，不要交別人的答案，作弊一律0分計算
- 僅需要上傳一個檔案到 NTU COOL
- **PageRank.py** (This is your modified PageRank.py file.)
- **DEADLINE: 2023/12/01(五) 23:59 (GMT+8:00)**
- **遲交、改格式、改檔名、改檔案、各種奇怪的錯誤無法改作業：每次分數×0.7，每次修改請完整按照格式繳交。**

Q & A

- 若有作業相關問題請到NTU COOL作業討論版發問
- 你的問題很可能也是其他同學的問題：)