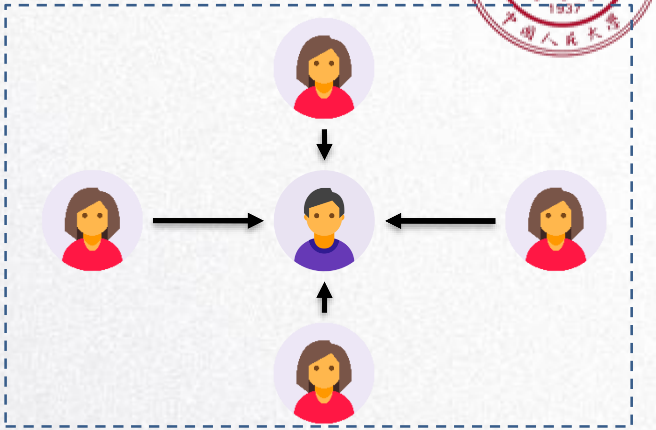
Page rank

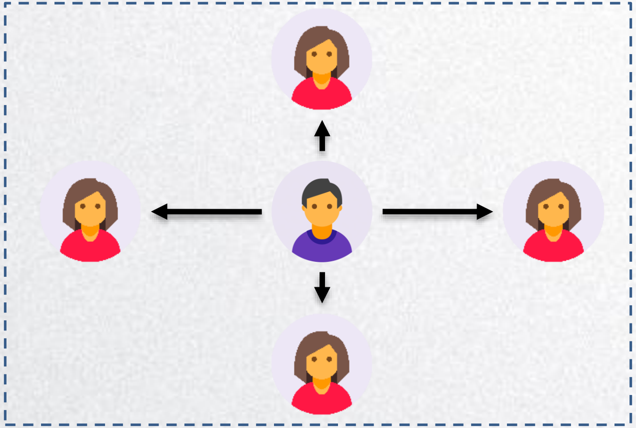
覃雄派

Page rank

- 在网络中，不同节点的“地位”是不平等的
  - 例子：美国高中生恋爱关系图
  - 如果定义有向边：“追求”关系

有向图





- 思考：
  - 右边两图中男生的重要性一样吗?
  - 你怎么解释这种重要性?

Page rank

提纲

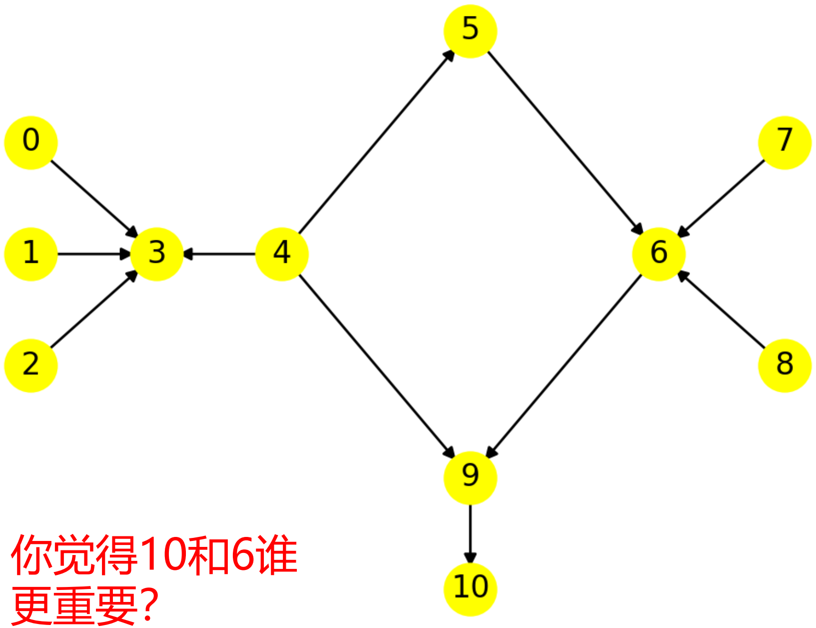
- Page Rank的核心思想
- 实例计算与公式推导
- 收敛性讨论
- Damping Factor
- Personalized page rank

Page rank

- 度量有向图节点的重要性
- 示例：简易版恋爱关系有向图
  - 定义有向边：“追求”关系

有向图

- 基于投票的思路
  - 将每个入边看作一次投票
  - 得到的票数越多，越重要

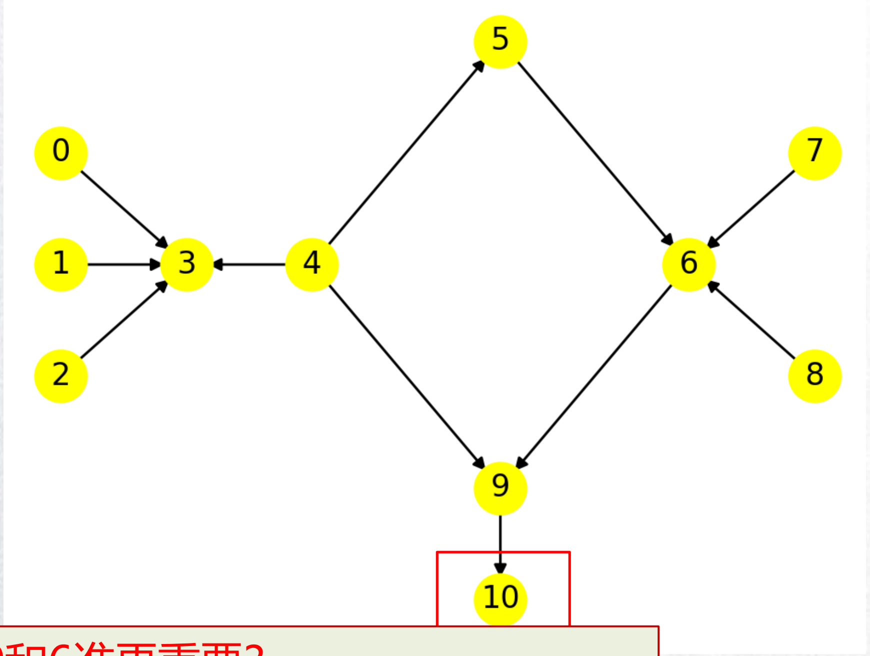


你觉得10和6谁更重要?

Page rank

- PageRank的基本思想：给不同的入边赋上不同的权重
  - 考虑某个节点v
  - 指向v的节点的PageRank值越高，相应入边的权重越高
  - 指向v的节点指向其它节点的数目越多，分摊越多，对v相应入边的权重越低

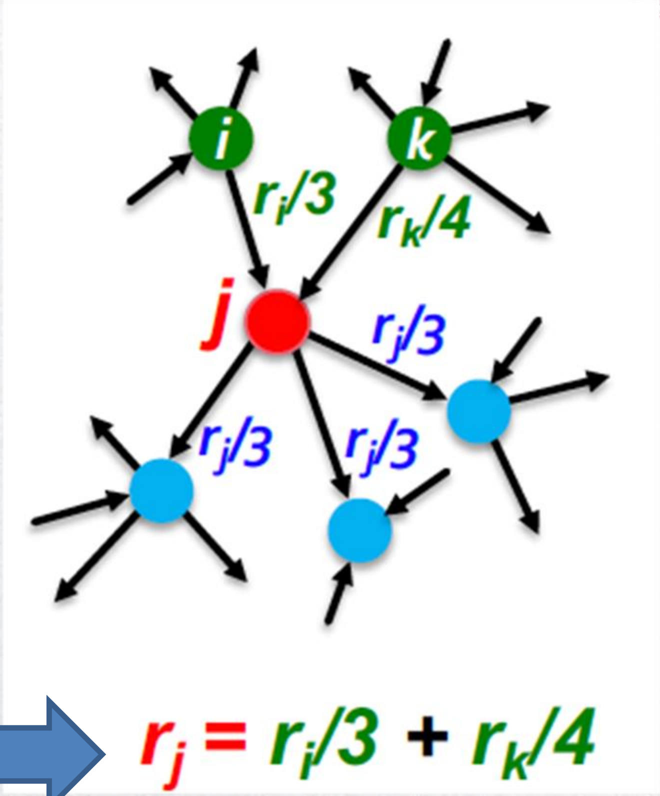
In-Degree Centrality不能表达!



你觉得10和6谁更重要?
还真难说；因为6得到了5/7/8的投票；
但是10得到了9的投票，而9得到了6的投票。

Page rank

- PageRank的基本思想：给不同的入边赋上不同的权重
  - 考虑某个节点v
  - 指向v的节点的PageRank值越高，相应入边的权重越高
  - 指向v的节点指向其它节点的数目越多，分摊越多，对v相应入边的权重越低

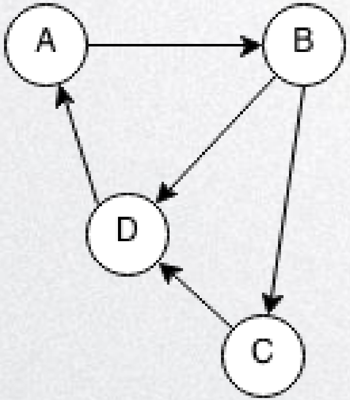


1，i指向j
2，k指向j
3，i的出度为3，以1/3分摊
4，k的出度为4，以1/4分摊

Page rank

- 如何用数学表达上述想法
  - 定义有向图的邻接矩阵$A=\{L_{ij}\}$
  - 其中$L_{ij}=1$表示i到j有边， $L_{ij}=0$表示无边
  - 以下图为例

$$
\begin{array}{c c}
 & \text{To} \quad A \quad B \quad C \quad D \\
\begin{array}{c}\text{From}\\ A \\ B \\ C \\ D\end{array} & A=\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}
\end{array}
$$

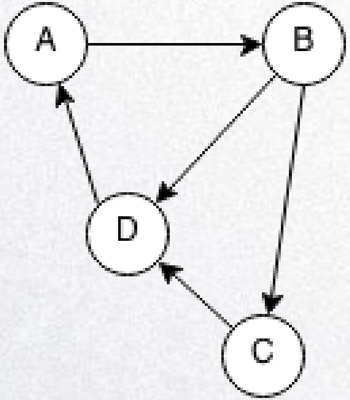


Page rank

- 如何用数学表达上述想法
  - 定义每个节点的出度为m_i，则有

$$m_i = \sum_{j=1}^{n} L_{ij}$$

  - 构造M矩阵如下
  - 对角线上的元素的值为
    - A的某一行的1的sum，即某个节点的出度



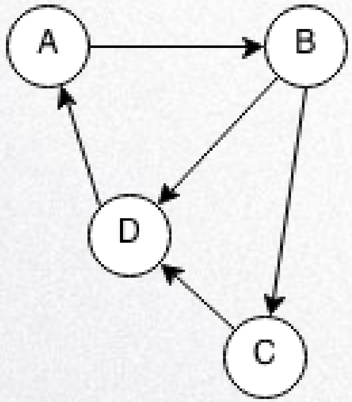
$$\begin{array}{c} A \\ B \\ C \\ D \end{array} \quad \mathrm{M}=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad A=\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Page rank

- 如何用数学表达上述想法
  - 计算$M^{-1}A$

$$M=\begin{bmatrix}1&0&0&0\\0&2&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\qquad A=\begin{bmatrix}0&1&0&0\\0&0&1&1\\0&0&0&1\\1&0&0&0\end{bmatrix}$$

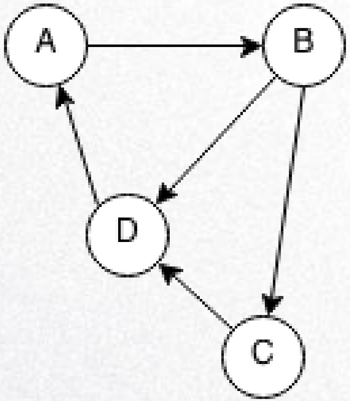
$$M^{-1}A=\begin{bmatrix}1&0&0&0\\0&\frac{1}{2}&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}0&1&0&0\\0&0&1&1\\0&0&0&1\\1&0&0&0\end{bmatrix}=\begin{bmatrix}0&1&0&0\\0&0&\frac{1}{2}&\frac{1}{2}\\0&0&0&1\\1&0&0&0\end{bmatrix}$$



B的出度为2，它的重要性按照1/2进行分摊，如何分摊看后文

D的出度为1，它的重要性按照1/1进行分摊...

Page rank



- 如何用数学表达上述想法
  - 假设已有结点A, B, C, D的初始page rank为
  - $[p_1 \quad p_2 \quad p_3 \quad p_4]$
  - 利用矩阵乘法进行Page Rank值的迭代结算
  - $[p_1 \quad p_2 \quad p_3 \quad p_4] = [p_1 \quad p_2 \quad p_3 \quad p_4]M^{-1}A$
  - $=[p_1 \quad p_2 \quad p_3 \quad p_4]\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$
  - $=\left[p_4 \quad p_1 \quad \frac{1}{2}p_2 \quad \frac{1}{2}p_2\right]$

这里$\frac{1}{m_1}L_{11}$、$\frac{1}{m_2}L_{21}$、$\frac{1}{m_3}L_{31}$都是0
$\frac{1}{m_4}L_{41}$=1

看看新的p_1
$= p_1\frac{1}{m_1}L_{11}+p_2\frac{1}{m_2}L_{21}+p_3\frac{1}{m_3}L_{31}+p_4\frac{1}{m_4}L_{41}$
它是其它各个节点的重要度，根据是否有它们到本节点的连接，分摊到本节点的重要度，累加

p_2, p_3, p_4做类似理解

Page rank

Page rank

- 写出PageRank值p_i的递推公式

$$p_i = \sum_{j \to i} \frac{p_j}{m_j} = \sum_{j=1}^{n} \frac{L_{ji}}{m_j} p_j$$

看看新的p_i
$= \boldsymbol{p_1 \frac{1}{m_1} L_{1i} + p_2 \frac{1}{m_2} L_{2i} + p_3 \frac{1}{m_3} L_{3i} + p_4 \frac{1}{m_4} L_{4i}}$
它是其它各个节点的重要度，根据是否有它们到p_i的连接，分摊到本节点的重要度的累加

将上面的公式写成矩阵形式

Page rank

- Page Rank迭代过程的一般形式
  - 写出PageRank值p_i的递推公式

$$\boldsymbol{p} = (p_1, p_2, \dots, p_n)$$

$$\boldsymbol{A} = \begin{bmatrix} L_{11} & L_{12} & \dots & L_{1n} \\ L_{21} & L_{22} & \dots & L_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ L_{n1} & L_{n2} & \dots & L_{nn} \end{bmatrix}$$

$$\boldsymbol{M} = \begin{bmatrix} m_1 & 0 & \dots & 0 \\ 0 & m_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & m_n \end{bmatrix}$$

$$\boldsymbol{p} \leftarrow \boldsymbol{p}(\boldsymbol{M}^{-1}\boldsymbol{A})$$

Let $\boldsymbol{L} = \boldsymbol{M}^{-1}\boldsymbol{A}$

$$\boldsymbol{p}^{t+1} \leftarrow \boldsymbol{p}^t\boldsymbol{L}$$

$$p_i = \sum_{j\to i} \frac{p_j}{m_j} = \sum_{j=1}^{n} \frac{L_{ji}}{m_j} p_j$$

看看新的p_i

$$= \boldsymbol{p_1 \frac{1}{m_1} L_{1i} + p_2 \frac{1}{m_2} L_{2i} + p_3 \frac{1}{m_3} L_{3i} + p_4 \frac{1}{m_4} L_{4i}}$$

它是其它各个节点的重要度，根据是否有它们到p_i的连接，分摊到本节点的重要度的累加

将上面的公式写成矩阵形式

Page rank

- "Beautiful math tends to be useful, and useful things tend to have beautiful math."

Page rank

Page rank

- 对不同的[$p_1 \quad p_2 \quad p_3 \quad p_4$]初始值进行迭代
  - 迭代函数

```python
import numpy as np
import networkx as nx

def pagerank_naive (DiG, pinit, max_iter=36):
    # Adjacency Matrix
    A = nx.to_numpy_matrix(DiG)
    # Out-Degree -> M -> M^{-1}
    D = np.sum(A,axis=1)
    M = np.diag(D.A1)
    M_I = np.linalg.inv(M)

    L = M_I @ A # Must use Python3 to use @
    p = pinit

    for i in range(max_iter):
        p = p @ L
        print (p)

```

名称	修改日期	类型	大小
01page_rank_simple.py	2022/2/13 15:11	Python File	1 KB

Page rank

- 对不同的[$p_1 \quad p_2 \quad p_3 \quad p_4$]初始值进行迭代
  - 创建一个图

```
DiG = nx.DiGraph()
DiG.name = "Simple Graph"
node_list = ['A','B','C','D']
DiG.add_nodes_from(node_list)
edge_list  = [('A','B'), ('B','C'), ('B','D'), ('C','D'), ('D','A')
              ]
DiG.add_edges_from(edge_list)
```

Page ra

- 对不同的[$p_1 \quad p_2 \quad p_3 \quad p_4$]初始值进
  - $p^0 = (1,0,0,0)$
  - $p^0 = (0,1,0,0)$
  - $p^0 = (0.25,0.25,0.25,0.25)$

```python
n= len(DiG)
p = np.ones(n)/n#(0.25,0.25,0.25,0.25)
print(p, ":")
pagerank_naive(DiG,p)

p = [1,0,0,0]
p = np.asarray(p)
print(p, ":")
pagerank_naive(DiG,p)

p = [0,1,0,0]
p = np.asarray(p)
print(p, ":")
pagerank_naive(DiG,p)
```

比喻：初始时所有节点上均有0.25滴红墨水；接着不断让墨水沿着图结构传递给邻居

```
[0.25 0.25 0.25 0.25] :
[[0.25  0.25  0.125 0.375]]
[[0.375 0.25  0.125 0.25 ]]
[[0.25  0.375 0.125 0.25 ]]
[[0.25   0.25   0.1875 0.3125]]
[[0.3125 0.25   0.125  0.3125]]
[[0.3125 0.3125 0.125  0.25  ]]
[[0.25    0.3125  0.15625 0.28125]]
[[0.28125 0.25    0.15625 0.3125 ]]
[[0.3125  0.28125 0.125   0.28125]]
[[0.28125  0.3125   0.140625 0.265625]]
[[0.265625 0.28125  0.15625  0.296875]]
[[0.296875 0.265625 0.140625 0.296875]]
[[0.296875  0.296875  0.1328125 0.2734375]]
[[0.2734375 0.296875  0.1484375 0.28125  ]]
[[0.28125   0.2734375 0.1484375 0.296875 ]]
[[0.296875   0.28125    0.13671875 0.28515625]]
[[0.28515625 0.296875   0.140625   0.27734375]]
[[0.27734375 0.28515625 0.1484375  0.2890625 ]]
[[0.2890625  0.27734375 0.14257812 0.29101562]]
[[0.29101562 0.2890625  0.13867188 0.28125   ]]
[[0.28125    0.29101562 0.14453125 0.28320312]]
[[0.28320312 0.28125    0.14550781 0.29003906]]
[[0.29003906 0.28320312 0.140625   0.28613281]]
[[0.28613281 0.29003906 0.14160156 0.28222656]]
[[0.28222656 0.28613281 0.14501953 0.28662109]]
[[0.28662109 0.28222656 0.14306641 0.28808594]]
[[0.28808594 0.28662109 0.14111328 0.28417969]]
[[0.28417969 0.28808594 0.14331055 0.28442383]]
[[0.28442383 0.28417969 0.14404297 0.28735352]]
[[0.28735352 0.28442383 0.14208984 0.28613281]]
[[0.28613281 0.28735352 0.14221191 0.28430176]]
[[0.28430176 0.28613281 0.14367676 0.28588867]]
[[0.28588867 0.28430176 0.14306641 0.28674316]]
[[0.28674316 0.28588867 0.14215088 0.28521729]]
[[0.28521729 0.28674316 0.14294434 0.28509521]]
[[0.28509521 0.28521729 0.14337158 0.28631592]]
```

Page ra

- 对不同的$[p_1 \quad p_2 \quad p_3 \quad p_4]$初始值进
  - $p^0 = (1,0,0,0)$
  - $p^0 = (0,1,0,0)$
  - $p^0 = (0.25,0.25,0.25,0.25)$

```python
n= len(DiG)
p = np.ones(n)/n#(0.25,0.25,0.25,0.25)
print(p, ":")
pagerank_naive(DiG,p)

p = [1,0,0,0]
p = np.asarray(p)
print(p, ":")
pagerank_naive(DiG,p)

p = [0,1,0,0]
p = np.asarray(p)
print(p, ":")
pagerank_naive(DiG,p)
```

比喻：初始时节点A上有一滴红墨水；接着不断让墨水沿着图结构传递给邻居

```
[1 0 0 0] :
[[0. 1. 0. 0.]]
[[0.  0.  0.5 0.5]]
[[0.5 0.  0.  0.5]]
[[0.5 0.5 0.  0. ]]
[[0.  0.5  0.25 0.25]]
[[0.25 0.  0.25 0.5 ]]
[[0.5  0.25 0.  0.25]]
[[0.25  0.5  0.125 0.125]]
[[0.125 0.25  0.25  0.375]]
[[0.375 0.125 0.125 0.375]]
[[0.375  0.375  0.0625 0.1875]]
[[0.1875 0.375  0.1875 0.25  ]]
[[0.25   0.1875 0.1875 0.375 ]]
[[0.375   0.25    0.09375 0.28125]]
[[0.28125 0.375   0.125   0.21875]]
[[0.21875 0.28125 0.1875  0.3125 ]]
[[0.3125   0.21875  0.140625 0.328125]]
[[0.328125 0.3125   0.109375 0.25    ]]
[[0.25     0.328125 0.15625  0.265625]]
[[0.265625  0.25      0.1640625 0.3203125]]
[[0.3203125 0.265625  0.125     0.2890625]]
[[0.2890625 0.3203125 0.1328125 0.2578125]]
[[0.2578125  0.2890625  0.16015625 0.29296875]]
[[0.29296875 0.2578125  0.14453125 0.3046875 ]]
[[0.3046875  0.29296875 0.12890625 0.2734375 ]]
[[0.2734375  0.3046875  0.14648438 0.27539062]]
[[0.27539062 0.2734375  0.15234375 0.29882812]]
[[0.29882812 0.27539062 0.13671875 0.2890625 ]]
[[0.2890625  0.29882812 0.13769531 0.27441406]]
[[0.27441406 0.2890625  0.14941406 0.28710938]]
[[0.28710938 0.27441406 0.14453125 0.29394531]]
[[0.29394531 0.28710938 0.13720703 0.28173828]]
[[0.28173828 0.29394531 0.14355469 0.28076172]]
[[0.28076172 0.28173828 0.14697266 0.29052734]]
[[0.29052734 0.28076172 0.14086914 0.2878418 ]]
[[0.2878418  0.29052734 0.14038086 0.28125   ]]
```

Page ra

- 对不同的$[p_1 \quad p_2 \quad p_3 \quad p_4]$初始值进
  - $p^0 = (1,0,0,0)$
  - $p^0 = (0,1,0,0)$
  - $p^0 = (0.25,0.25,0.25,0.25)$

```python
n= len(DiG)
p = np.ones(n)/n#(0.25,0.25,0.25,0.25)
print(p, ":")
pagerank_naive(DiG,p)

p = [1,0,0,0]
p = np.asarray(p)
print(p, ":")
pagerank_naive(DiG,p)

p = [0,1,0,0]
p = np.asarray(p)
print(p, ":")
pagerank_naive(DiG,p)
```

比喻：初始时节点B上有一滴红墨水；接着不断让墨水沿着图结构传递给邻居

```
[0 1 0 0] :
[[0.  0.  0.5 0.5]]
[[0.5 0.  0.  0.5]]
[[0.5 0.5 0.  0. ]]
[[0.  0.5  0.25 0.25]]
[[0.25 0.  0.25 0.5 ]]
[[0.5  0.25 0.  0.25]]
[[0.25  0.5  0.125 0.125]]
[[0.125 0.25  0.25  0.375]]
[[0.375 0.125 0.125 0.375]]
[[0.375  0.375  0.0625 0.1875]]
[[0.1875 0.375  0.1875 0.25  ]]
[[0.25   0.1875 0.1875 0.375 ]]
[[0.375   0.25    0.09375 0.28125]]
[[0.28125 0.375   0.125   0.21875]]
[[0.21875 0.28125 0.1875  0.3125 ]]
[[0.3125   0.21875  0.140625 0.328125]]
[[0.328125 0.3125   0.109375 0.25     ]]
[[0.25     0.328125 0.15625  0.265625]]
[[0.265625  0.25      0.1640625 0.3203125]]
[[0.3203125 0.265625  0.125     0.2890625]]
[[0.2890625 0.3203125 0.1328125 0.2578125]]
[[0.2578125  0.2890625  0.16015625 0.29296875]]
[[0.29296875 0.2578125  0.14453125 0.3046875 ]]
[[0.3046875  0.29296875 0.12890625 0.2734375 ]]
[[0.2734375  0.3046875  0.14648438 0.27539062]]
[[0.27539062 0.2734375  0.15234375 0.29882812]]
[[0.29882812 0.27539062 0.13671875 0.2890625 ]]
[[0.2890625  0.29882812 0.13769531 0.27441406]]
[[0.27441406 0.2890625  0.14941406 0.28710938]]
[[0.28710938 0.27441406 0.14453125 0.29394531]]
[[0.29394531 0.28710938 0.13720703 0.28173828]]
[[0.28173828 0.29394531 0.14355469 0.28076172]]
[[0.28076172 0.28173828 0.14697266 0.29052734]]
[[0.29052734 0.28076172 0.14086914 0.2878418 ]]
[[0.2878418  0.29052734 0.14038086 0.28125   ]]
[[0.28125    0.2878418  0.14526367 0.28564453]]
```

Page rank

- 对不同的$[p_1 \quad p_2 \quad p_3 \quad p_4]$初始值进行迭代
  - $p^0 = (1,0,0,0)$
  - $p^0 = (0,1,0,0)$
  - $p^0 = (0.25,0.25,0.25,0.25)$
  - 都收敛到[[0.28125 0.2878418 0.14526367 0.28564453]]左右

  - 这是巧合吗?

Page rank

Page rank

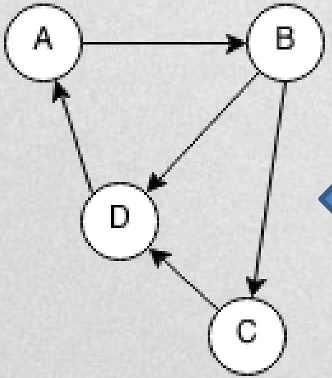
- PageRank分值稳定代表了什么?
- 分值稳定为什么重要?
  - 度量节点重要性需要分值稳定
- 分值会稳定到什么状态?
  - 分值稳定意味着$P^{t+1}=p^t$

$$\boldsymbol{p}=\boldsymbol{pL}$$

- 这说明稳定状态时
  - p是矩阵L对应特征值为1的特征向量!
- 可是……
  - 1.我们怎么能确定L有为1的特征值?
  - 2.就算有，特征向量p唯一吗?

Page rank

- 马尔科夫链视角下的PageRank
- 马尔科夫链的稳态概率分布（Stationary Distribution）
  - 一个马尔科夫链存在唯一的稳态分布，当且仅当它是**不可约的遍历链**
  - 上述定理的数学证明课上不展开
  - 进一步学习：《应用随机过程》，林元烈著



直观解释

强连通的有向图

每个节点都有入链和出链

Page rank

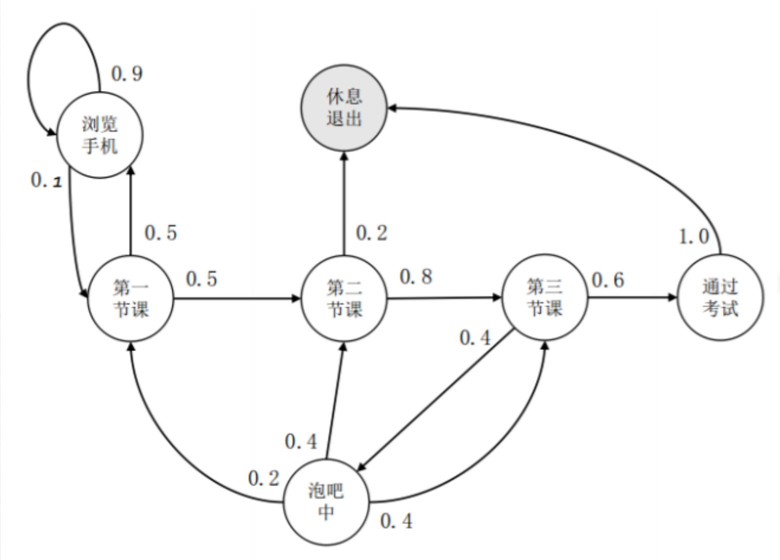
- 马尔科夫链视角下的PageRank
- 一个离散时间的马尔科夫链是一个带有马尔科夫性的随机变量的序列 $X_1, X_2, X_3, \ldots$，即转移到下一个状态的概率只取决于当前状态，而与历史状态无关，用公式表示如下：
  - $\Pr(X_{n+1} = x | X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = \Pr(X_{n+1} = x | X_n = x_n)$
  - 其中： $\Pr(X_{n+1} = x | X_n = x_n)$ 称为状态转移概率；
  - 随机变量 X_i 可能的取值构成了一个可数的集合 S ，称为状态空间

练习：请建立马尔科夫链与PageRank的关联
（下页继续）

马尔科夫性

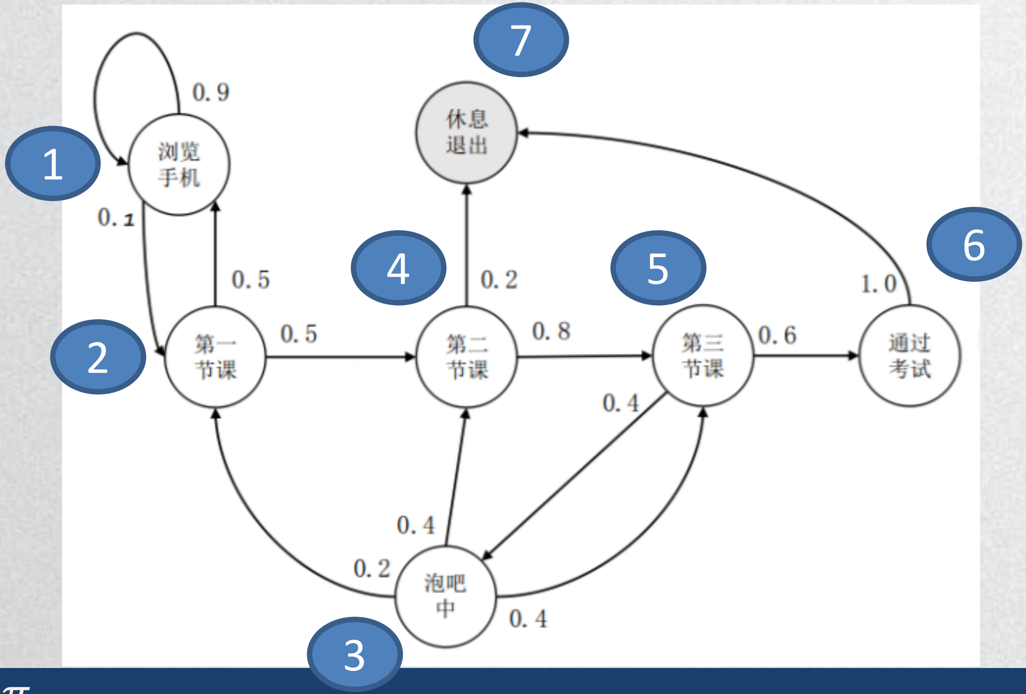
Page rank

- 马尔科夫链
- 练习 - 请写出该图对应的：
  - 状态空间S
  - 状态转移概率(矩阵)T
  - 如果时间点1时，某学生选择上了“第一节课”，请写出$X1$的分布
  - 如果$X1$符合上述分布，写出$X2$的分布
  - 如果时间$n\to\infty$，变量X_n的分布会稳定吗？（参考前一页的说明）



Page rank

- 马尔科夫链
- 练习 - 请写出该图对应的：
  - 状态空间S...



1.状态空间S

{1, 2, 3, 4, 5, 6, 7}

2.状态转移概率(矩阵)T

$$\begin{bmatrix} 0.9 & 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0.4 & 0.4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.8 & 0 & 0.2 \\ 0 & 0 & 0.4 & 0 & 0 & 0.6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

3. 如果时间点1时，某学生选择上了第一节课，请写出$X1$的分布

(0, 1, 0, 0, 0, 0, 0)

4. 如果$X1$符合上述分布，写出$X2$的分布

$X2$ =$X1$ * T 得到(0.5, 0, 0, 0.5, 0, 0, 0)

解读一下？

5. 如果时间$n \to \infty$，变量Xn的分布会稳定吗？（参考前一页的说明）

不稳定，7只有入链，没有有出链

不是不可约的遍历链

Page rank

- PageRank的性质
- 一个看似“反常识”的结论
  - Random Walk → Deterministic Answer
- 马尔科夫链的稳态概率分布（Stationary Distribution）
  - 一个马尔科夫链存在唯一的稳态分布，当前仅当它是不可约的遍历链

$$\boldsymbol{p} = \boldsymbol{pL}$$

- 稳定状态时：p是矩阵L对应特征值为1的特征向量！

Page rank

- 统计思维 Statistical Thinking

数据科学的必备能力之一：统计思维

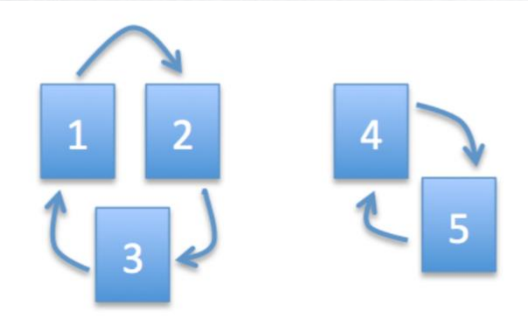
Random → Deterministic Distribution

Page rank

Page rank

我们怎么能确定L有1的特征值？
就算有，特征向量p唯一吗？

- 考虑一个反例……
- 该图对应的L矩阵，特征值为1的特征向量有几个？



$$\boldsymbol{L}=\begin{bmatrix}0&0&1&0&0\\1&0&0&0&0\\0&1&0&0&0\\0&0&0&0&1\\0&0&0&1&0\end{bmatrix}$$

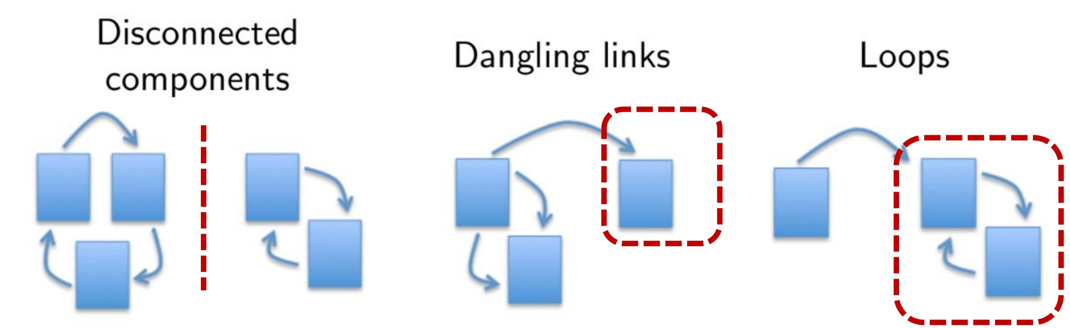
$$\boldsymbol{p}=\begin{bmatrix}\frac{1}{3}&\frac{1}{3}&\frac{1}{3}&0&0\end{bmatrix}\Rightarrow\boldsymbol{p}=\boldsymbol{pL}$$

$$\boldsymbol{p}=\begin{bmatrix}0&0&0&\frac{1}{2}&\frac{1}{2}\end{bmatrix}\Rightarrow\boldsymbol{p}=\boldsymbol{pL}$$

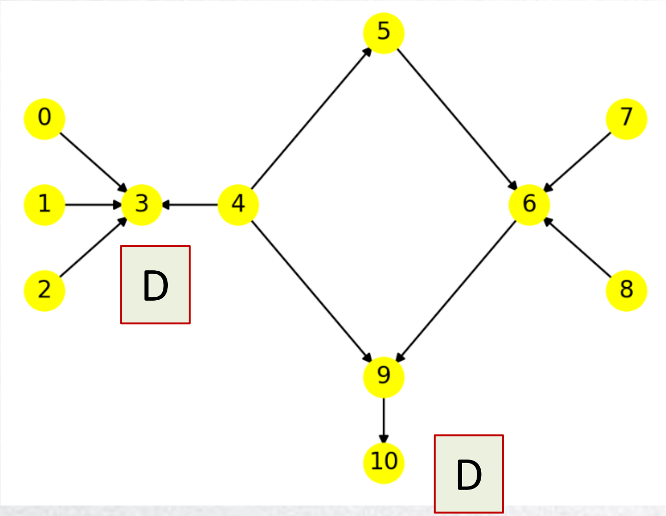
有关节点重要性的结论不符合直觉！

Page rank

- 真实的图结构是复杂的
- 可能会存在以下三种不强连通的情况



- 考虑我们的恋爱关系示例图



应该如何解决这一问题？
即怎么样在此基础上，强制构造强连通？

Page rank

- 真正的PageRank算法
- 在前面计算的公式的基础上做了**“微小”改动**

$$p = \alpha\, \boldsymbol{pL} + \frac{1-\alpha}{n}\, \boldsymbol{pE},$$ ***E* is the n × n matrix of 1s**

α: Damping parameter, 经验上取0.85

$$p = p\left(\ 0.85\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} + 0.15\begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}\right)$$

以0.15的比例，在每个节点上按照1/4跳转到本节点和另外3个节点

Page rank

- 真正的PageRank算法
- 在前面计算的公式的基础上做了“微小”改动

$$p = \alpha\, \boldsymbol{pL} + \frac{1-\alpha}{n}\, \boldsymbol{pE},$$ ***E* is the n × n matrix of 1s**

α: Damping parameter, 经验上取0.85

展开

$$p = p\,0.85\begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & \frac{1}{2} & \frac{1}{2}\\ 0 & 0 & 0 & 1\\ 1 & 0 & 0 & 0\end{bmatrix} + p0.15\begin{bmatrix}\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4}\\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4}\\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4}\\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4}\end{bmatrix}$$

$$p0.15\begin{bmatrix}\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4}\\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4}\\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4}\\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4}\end{bmatrix} = [p_1 \quad p_2 \quad p_3 \quad p_4]\,0.15\begin{bmatrix}\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4}\\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4}\\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4}\\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4}\end{bmatrix}$$

$$=0.15\left[\frac{p_1+p_2+p_3+p_4}{4} \quad \frac{p_1+p_2+p_3+p_4}{4} \quad \frac{p_1+p_2+p_3+p_4}{4} \quad \frac{p_1+p_2+p_3+p_4}{4}\right]$$

$$=0.15\left[\frac{1}{4} \quad \frac{1}{4} \quad \frac{1}{4} \quad \frac{1}{4}\right] = \frac{0.15}{4}\,[1 \quad 1 \quad 1 \quad 1]$$

Page rank

- 真正的PageRank算法
- 在前面计算的公式的基础上做了“微小”改动

$$p = \alpha\, \boldsymbol{pL} + \frac{1-\alpha}{n}\boldsymbol{e},$$ $\boldsymbol{e}$为元素都为1的行向量

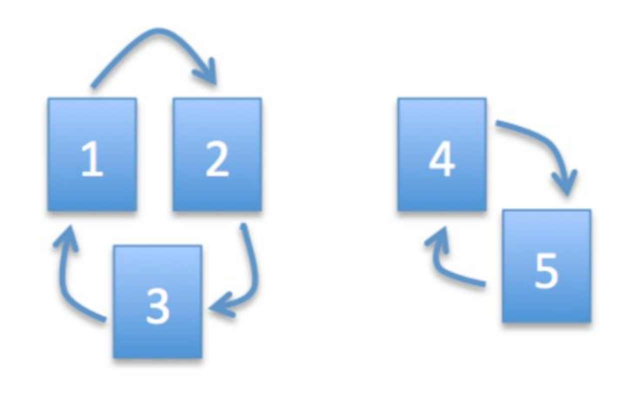
α: Damping parameter, 经验上取0.85

PageRank计算的过程也称随机游走（Random Walk）

Page rank

- 再次考虑之前的反例……考虑$\alpha = 0.85$

$$p = \alpha\, \boldsymbol{pL} + \frac{1-\alpha}{n}\boldsymbol{pE}$$



$$\boldsymbol{L} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$= \frac{0.15}{5} \cdot \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} + 0.85 \cdot \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0.03 & 0.03 & 0.88 & 0.03 & 0.03 \\ 0.88 & 0.03 & 0.03 & 0.03 & 0.03 \\ 0.03 & 0.88 & 0.03 & 0.03 & 0.03 \\ 0.03 & 0.03 & 0.03 & 0.03 & 0.88 \\ 0.03 & 0.03 & 0.03 & 0.88 & 0.03 \end{pmatrix}.$$

Now only one eigenvector of A with eigenvalue 1: $p = \boxed{\begin{pmatrix} 0.2 \\ 0.2 \\ 0.2 \\ 0.2 \\ 0.2 \end{pmatrix}}$.

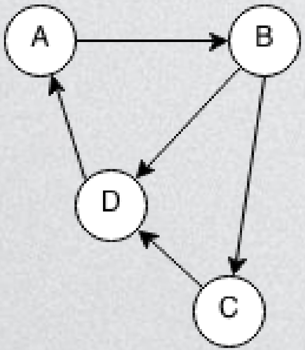
比较合理

Page rank

- 计算该图结构中节点的page rank分值
- 考虑$\alpha = 0.8$且$p^{(0)} = (1,0,0,0)$
  - 计算$p^{(1)}$和$p^{(2)}$
  - 计算收敛时的p

$$p = \alpha\, \boldsymbol{pL} + \frac{1-\alpha}{n}\, \boldsymbol{pE} \rightarrow$$

$$p = \boldsymbol{p}(\alpha\, \boldsymbol{L} + \frac{1-\alpha}{n}\, \boldsymbol{E}),\ E \text{ is the n × n matrix of 1s}$$



$$L = M^{-1}A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$p = p(0.80 \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} + \frac{0.20}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}) = p \begin{bmatrix} 0.05 & 0.85 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.45 & 0.45 \\ 0.05 & 0.05 & 0.05 & 0.85 \\ 0.85 & 0.05 & 0.05 & 0.05 \end{bmatrix}$$

$p^{(0)} = (1,0,0,0)$代入即可

名称	修改日期	类型	大小
02page_rank_validate.py	2022/2/13 14:48	Python File	1 KB

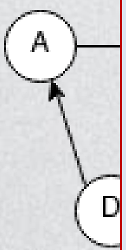
Page rank

- 计算该图
- 考虑α =
  - 计算p
  - 计算收

```
[0.05 0.85 0.05 0.05]
[0.09 0.09 0.39 0.43]
[0.394 0.122 0.086 0.398]
[0.3684 0.3652 0.0988 0.1676]
[0.18408 0.34472 0.19608 0.27512]
[0.270096 0.197264 0.187888 0.344752]
[0.3258016 0.2660768 0.1289056 0.279216 ]
[0.2733728  0.31064128 0.15643072 0.2595552 ]
[0.25764416 0.26869824 0.17425651 0.29940109]
[0.28952087 0.25611533 0.1574793  0.29688451]
[0.2875076  0.2816167  0.15244613 0.27842957]
[0.27274365 0.28000608 0.16264668 0.28460358]
[0.27768287 0.26819492 0.16200243 0.29211978]
[0.28369582 0.27214629 0.15727797 0.28687992]
[0.27950393 0.27695666 0.15885852 0.28468089]
[0.27774471 0.27360315 0.16078266 0.28786948]
[0.28029558 0.27219577 0.15944126 0.28806739]
[0.28045391 0.27423647 0.15887831 0.28643132]
[0.27914505 0.27436313 0.15969459 0.28679723]
[0.27943779 0.27331604 0.15974525 0.28750092]
```

→

, E is the n × n matrix of 1s



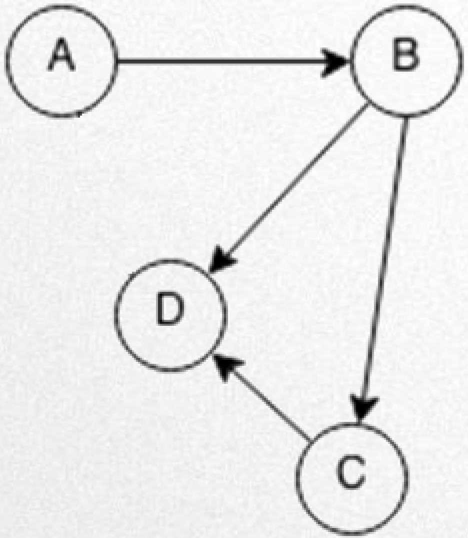
$$\begin{matrix} 0 & 0 \\ \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \\ 0 & 0 \end{matrix}$$

$$\begin{bmatrix} 0.05 & 0.85 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.45 & 0.45 \\ 0.05 & 0.05 & 0.05 & 0.85 \\ 0.85 & 0.05 & 0.05 & 0.05 \end{bmatrix}$$

名称	修改日期	类型	大小
02page_rank_validate.py	2022/2/13 14:48	Python File	1 KB

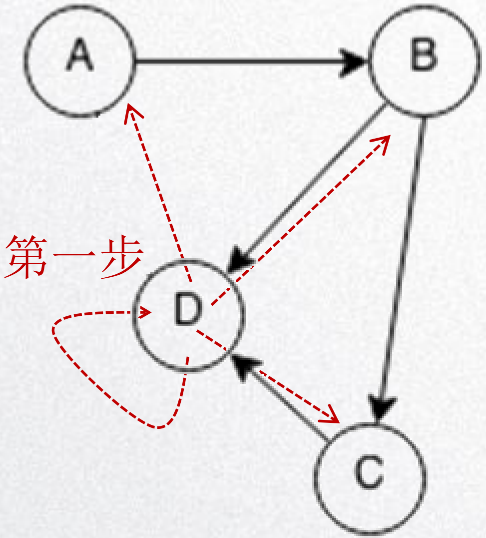
Page rank

- 练习
- 请计算以下图的PageRank值
  - 请写出邻接矩阵，设$\alpha = 0.8$
  - 假设初值为$p^{(0)} = (1,0,0,0)$



Page rank

- 练习
- 请计算以下图的PageRank值
  - 请写出邻接矩阵，设$\alpha = 0.8$
  - 假设初值为$p^{(0)} = (1,0,0,0)$



- 解决方案
  - 将Dangling节点
  - 与所有节点都建立一条边
  - 修改邻接矩阵

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ \boxed{0 \quad 0 \quad 0 \quad 0} \end{bmatrix} \Rightarrow A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ \boxed{1 \quad 1 \quad 1 \quad 1} \end{bmatrix}$$

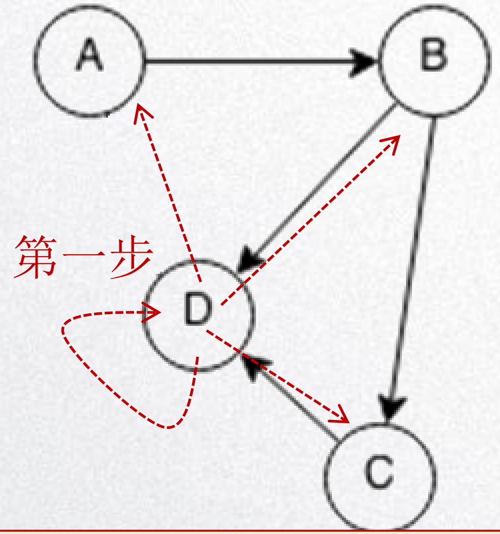
注意$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$

$$L = M^{-1}A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 1 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}$$

Page rank

- 练习
- 请计算以下图的PageRank值
  - 请写出邻接矩阵，设$\alpha = 0.8$
  - 初值为$p^{(0)} = (1,0,0,0)$

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$



$$L = M^{-1}A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{1}{4} \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}$$

$$p = p(0.80 \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix} + \frac{0.20}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix})$$

$p^{(0)} = (1,0,0,0)$代入即可

$$\begin{bmatrix} 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0.4 & 0.4 \\ 0 & 0 & 0 & 0.8 \\ 0.2 & 0.2 & 0.2 & 0.2 \end{bmatrix} + \begin{bmatrix} 0.05 & 0.05 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.05 & 0.05 \end{bmatrix} = \begin{bmatrix} 0.05 & 0.85 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.45 & 0.45 \\ 0.05 & 0.05 & 0.05 & 0.85 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}$$

练习题

$$p^{(0)} = (1,0,0,0)$$

$$\begin{bmatrix} 0.05 & 0.85 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.45 & 0.45 \\ 0.05 & 0.05 & 0.05 & 0.85 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}$$

- *前几次迭代的结果如下*
  - $p^{(1)}$ =[0.05 0.85 0.05 0.05] →sum to 1
  - $p^{(2)}$ =[0.06 0.1 0.4 0.44] →sum to 1
  - $p^{(3)}$ =[0.138 0.186 0.178 0.498] →sum to 1
  - $p^{(4)}$ =[0.1496 0.26 0.224 0.3664] →sum to 1
  - $p^{(5)}$ =[0.12328 0.24296 0.22728 0.40648] →sum to 1
  - .
  - .
  - .

```python
max_iter=5
p = [1,0,0,0]
p = np.asarray(p)
L = [ [0.05,0.85,0.05,0.05],
      [0.05,0.05,0.45,0.45],
      [0.05,0.05,0.05,0.85],
      [0.25,0.25,0.25,0.25]]
for i in range(max_iter):
    p = p @ L
    print (p)

```

名称	修改日期	类型	大小
02page_rank_validate.py	2022/2/13 14:48	Python File	1 KB

Page rank

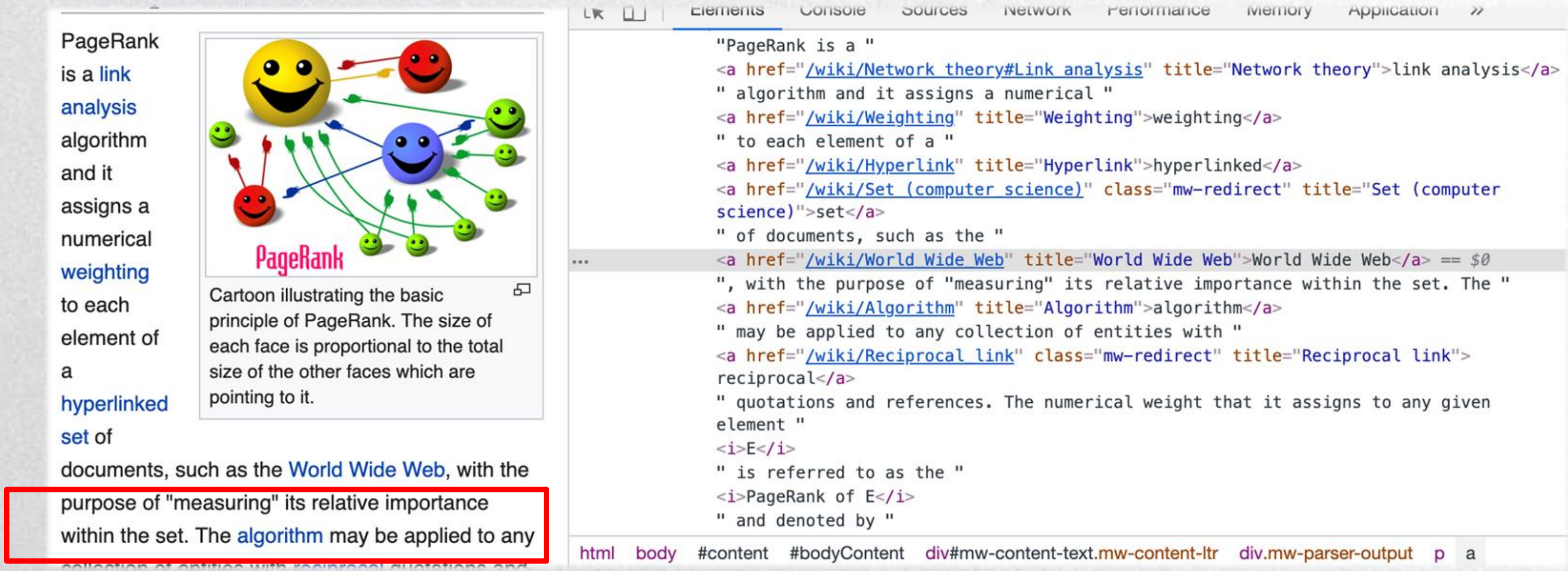
Page rank

- PageRank在Web Search中的应用

- **覆盖主题：**单一 vs. 多元
- **内容源：**专家学者 vs. 普罗大众
- **质量评估标准：**清晰 vs. 复杂
- **用户查询：**结构化（精确但有门槛）、关键词（易用却可能有歧义）

Page rank

- PageRank在Web Search中的应用
- PageRank由谷歌公司的两个创始人Larry Page和Sergei Brin提出，主要解决Web Page的排序问题



Page rank

- Node Centrality
- 1. 基于几何图形的度量方法
  - Degree Centrality
  - Closeness Centrality
- 2. 基于路径的度量方法
  - Betweenness Centrality
- 3. PageRank算法
  - 矩阵运算形式（为什么要有damping factor?）
  - 马尔科夫链的数学性质
  - 个性化PageRank算法（后文介绍）

L有为1的特征值
特征向量p唯一

⇕

马尔科夫链存在唯一的稳态分布

Page rank

Page rank

- Page rank计算Python实例分析
- 使用PageRank计算恋爱图的Centrality

“0501-图数据入门、中心度”目录下

名称	类型	大小	修改日期
01graph_high_school_love.py	Python File	3 KB	2021/10/25 16:00

https://aksakalli.github.io/2017/07/17/network-centrality-measures-and-their-visualization.html

Page rank

- Page rank计算Python实例分析

注意是有向图

```python
def school_dating_digraph():
    students = set(range(11))
    G = nx.DiGraph()
    G.name = "Simple Dating Graph"
    G.add_nodes_from(students)
    dating_rel = [(0,3), (1,3), (2,3), (3,4), (4,5), (4,9),
                  (5,6), (6,7), (6,8), (6,9), (9,10)]
    G.add_edges_from(dating_rel)
    # You may want to try automatic layout
    #pos = nx.spring_layout();
    pos = {0: [0.1, 0.6], 1: [0.1, 0.5], 2: [0.1, 0.4], 3: [0.2, 0.5],
           4: [0.3, 0.5], 5: [0.45, 0.7], 6: [0.6, 0.5], 7: [0.7, 0.6],
           8: [0.7, 0.4], 9: [0.45, 0.3], 10: [0.45, 0.2]}
    return G, pos
```

Page rank

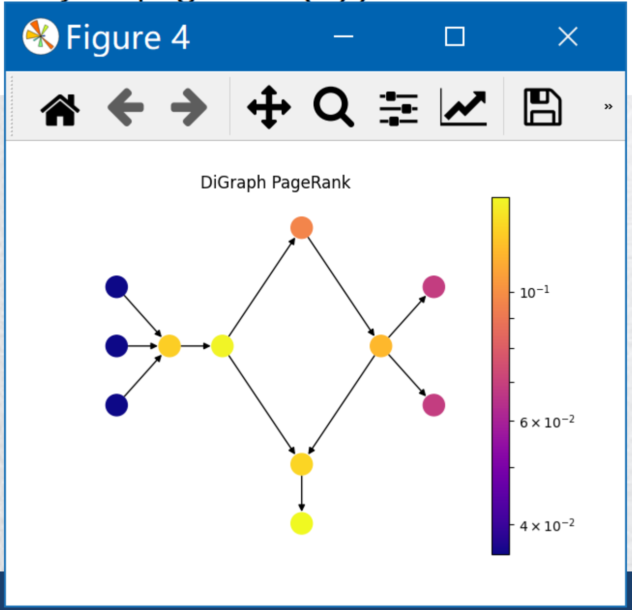
- Page rank计算Python实例分析

```python
G,pos = school_dating_digraph()
draw(G, pos, nx.pagerank(G, alpha=0.85), 'DiGraph PageRank',4)
print("pagerank",nx.pagerank(G))
```

Page rank

- Page rank计算Python实例分析

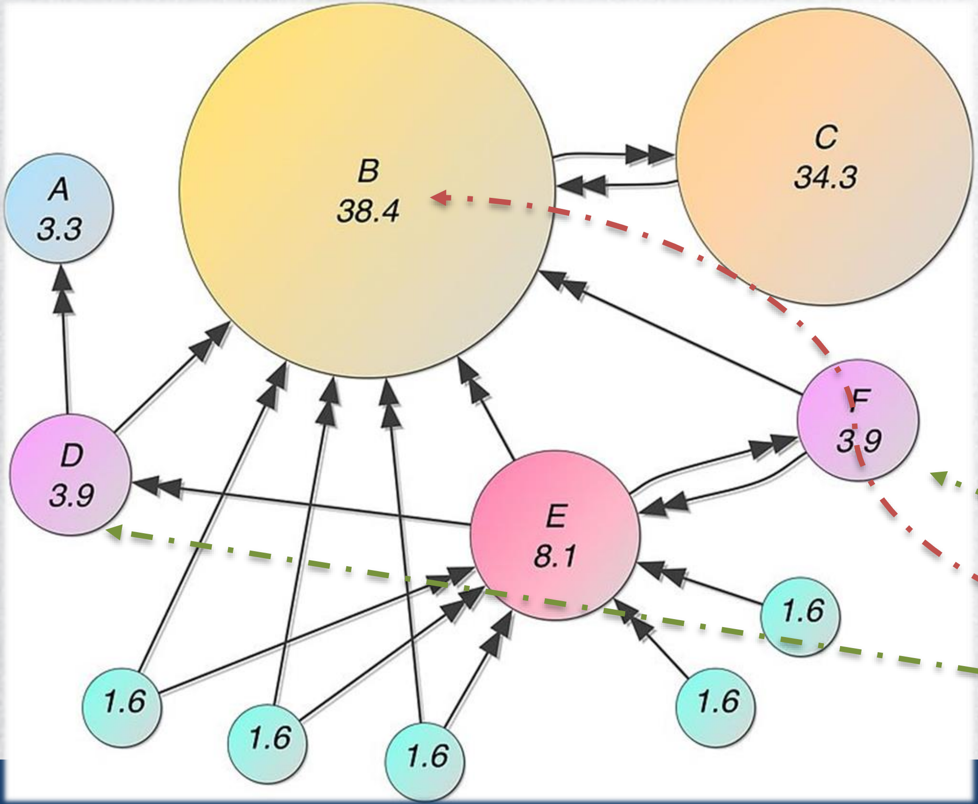
```python
G,pos = school_dating_digraph()
draw(G, pos, nx.pagerank(G, alpha=0.85), 'DiGraph PageRank',4)
print("pagerank",nx.pagerank(G))
```



Page rank

Page rank

- Personalized PageRank



- 考虑新的场景
  - 如果用户已经收藏了网页D和F，希望最后算出的分数反映出这种偏好
  - 与已收藏网页D和F相关的（直接或间接指向）的网页得到更高的分数
- 个性化
  - 那些都是很好的可我偏偏不喜欢——《白马啸西风》

Page rank

- Personalized PageRank
- 在前面计算的公式的基础上做了“微小”改动

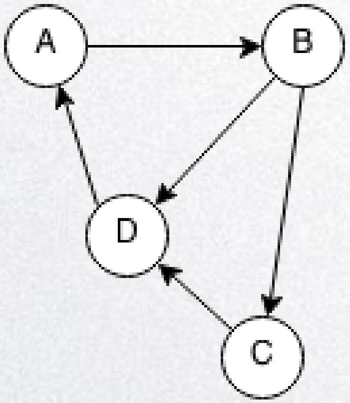
$$p = \alpha\, \boldsymbol{pL} + (1-\alpha)\, \boldsymbol{p}^{(\mathbf{0})}$$

α: Damping parameter, 经验上取0.85

- 计算的过程也称Random Walk with **Restart**
- **比如α=0.85，$\boldsymbol{p}^{(\mathbf{0})}$=(1,0,0,0)**

$$\mathrm{L} = M^{-1}A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$p = \alpha\, \boldsymbol{pL} + (1-\alpha)\, \boldsymbol{p}^{(\mathbf{0})} = 0.85p \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} + 0.2\mathbf{(1,0,0,0)}$$



Page rank

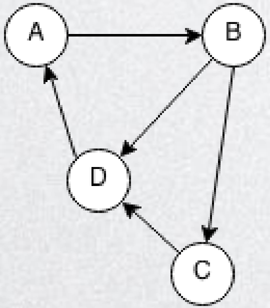
- Personalized page rank代码

名称	类型	大小	修改日期
01page_rank_simple.py	Python File	1 KB	2021/10/26 18:32
02page_rank_validate.py	Python File	1 KB	2021/10/26 21:21

```python
# personalized page rank
p = [1,0,0,0]
p = np.asarray(p)
p0 = p
alpha = 0.85
max_iter = 25
L = [ [0,1,0,0],
     [0,0,0.5,0.5],
     [0,0,0,1],
     [1,0,0,0]]
for i in range(max_iter):
    p = alpha *p @ L + (1-alpha)*p0
    print (p)
```

Page rank

- Personalized PageRank与p的初值有关吗?
- 实验测试，考虑$\alpha = 0.85$



- 考虑以下几种初始值
  - $p^0 = (1,0,0,0)$
  - $p^0 = (0,1,0,0)$
  - $p^0 = (0.5,0.5,0,0)$
- 请预测最终结果……

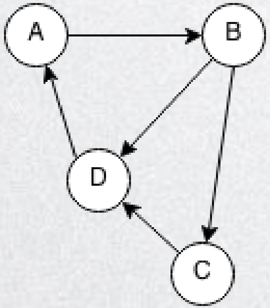
```
[1. 0. 0. 0.]
[[0.15 0.85 0.   0.  ]]
[[0.15     0.1275   0.36125 0.36125]]
[[0.4570625 0.1275    0.0541875 0.36125   ]]
[[0.4570625  0.38850312 0.0541875  0.10024687]]
[[0.23520984 0.38850312 0.16511383 0.2111732 ]]
[[0.32949722 0.19992837 0.16511383 0.30546058]]
[[0.40964149 0.28007264 0.08496956 0.22531631]]
[[0.34151886 0.34819527 0.11903087 0.19125499]]
[[0.31256675 0.29029103 0.14798299 0.24915923]]
[[0.36178535 0.26568173 0.12337369 0.24915923]]
[[0.36178535 0.30751754 0.11291474 0.21778237]]

... ... ... ... ... ... ... ... ... ... ... ...

[[0.34727498 0.29518373 0.12545309 0.23208821]]
[[0.34727498 0.29518373 0.12545309 0.23208821]]
[[0.34727498 0.29518373 0.12545309 0.23208821]]
[[0.34727498 0.29518373 0.12545309 0.23208821]]
[[0.34727498 0.29518373 0.12545309 0.23208821]]
[[0.34727498 0.29518373 0.12545309 0.23208821]]
[[0.34727498 0.29518373 0.12545309 0.23208821]]
[[0.34727498 0.29518373 0.12545309 0.23208821]]
[[0.34727498 0.29518373 0.12545309 0.23208821]]
[[0.34727498 0.29518373 0.12545309 0.23208821]]
[[0.34727498 0.29518373 0.12545309 0.23208821]]
```

Page rank

- Personalized PageRank与p的初值有关吗?
- 实验测试，考虑$\alpha = 0.85$



- 考虑以下几种初始值
  - $p^0 = (1,0,0,0)$
  - $p^0 = (0,1,0,0)$
  - $p^0 = (0.5,0.5,0,0)$
- 请预测最终结果……

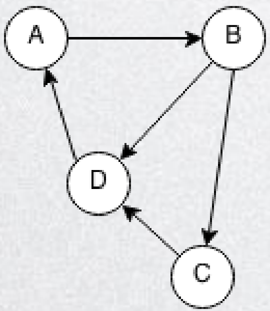
```
[0. 1. 0. 0.]
[[0.     0.15   0.425 0.425]]
[[0.36125 0.15     0.06375 0.425   ]]
[[0.36125    0.4570625 0.06375    0.1179375]]
[[0.10024687 0.4570625  0.19425156 0.24843906]]
[[0.2111732  0.23520984 0.19425156 0.35936539]]
[[0.30546058 0.32949722 0.09996418 0.26507801]]
[[0.22531631 0.40964149 0.14003632 0.22500588]]
[[0.19125499 0.34151886 0.17409764 0.29312851]]
[[0.24915923 0.31256675 0.14514552 0.29312851]]
[[0.24915923 0.36178535 0.13284087 0.25621456]]
[[0.21778237 0.36178535 0.15375877 0.26667351]]

... ... ... ... ... ... ... ... ... ... ... ...

[[0.23208821 0.34727498 0.14759187 0.27304495]]
[[0.23208821 0.34727498 0.14759187 0.27304495]]
[[0.23208821 0.34727498 0.14759187 0.27304495]]
[[0.23208821 0.34727498 0.14759187 0.27304495]]
[[0.23208821 0.34727498 0.14759187 0.27304495]]
[[0.23208821 0.34727498 0.14759187 0.27304495]]
[[0.23208821 0.34727498 0.14759187 0.27304495]]
[[0.23208821 0.34727498 0.14759187 0.27304495]]
[[0.23208821 0.34727498 0.14759187 0.27304495]]
```

Page rank

- Personalized PageRank与$\boldsymbol{p}$的初值有关吗？
- 实验测试，考虑$\alpha = 0.85$



- 考虑以下几种初始值
  - $p^0 = (1,0,0,0)$
  - $p^0 = (0,1,0,0)$
  - $p^0 = (0.5,0.5,0,0)$

- 请预测最终结果……

```
[0.5 0.5 0.  0. ]
[[0.075  0.5    0.2125 0.2125]]
[[0.255625 0.13875  0.2125   0.393125]]
[[0.40915625 0.29228125 0.05896875 0.23959375]]
[[0.27865469 0.42278281 0.12421953 0.17434297]]
[[0.22319152 0.31185648 0.1796827  0.2852693 ]]
[[0.3174789  0.26471279 0.13253901 0.2852693 ]]
[[0.3174789  0.34485707 0.11250294 0.22516109]]
[[0.26638693 0.34485707 0.14656425 0.24219175]]
[[0.28086299 0.30142889 0.14656425 0.27114387]]
[[0.30547229 0.31373354 0.12810728 0.25268689]]
[[0.28978386 0.33465145 0.13333675 0.24222794]]

... ... ... ... ... ... ... ... ... ... ... ...

[[0.28968159 0.32122935 0.13652248 0.25256658]]
[[0.28968159 0.32122935 0.13652248 0.25256658]]
[[0.28968159 0.32122935 0.13652248 0.25256658]]
[[0.28968159 0.32122935 0.13652248 0.25256658]]
[[0.28968159 0.32122935 0.13652248 0.25256658]]
[[0.28968159 0.32122935 0.13652248 0.25256658]]
[[0.28968159 0.32122935 0.13652248 0.25256658]]
```

Page rank

- Personalized PageRank与p的初值有关吗?
- 实验测试，考虑$\alpha = 0.85$
- 考虑以下几种初始值
  - $p^0 = (1,0,0,0)$
  - $p^0 = (0,1,0,0)$
  - $p^0 = (0.5,0.5,0,0)$

[[0.34727498 0.29518373 0.12545309 0.23208821]]

[[0.23208821 0.34727498 0.14759187 0.27304495]]

[[0.28968159 0.32122935 0.13652248 0.25256658]]

- 你能发现什么规律?

特殊偏好的节点得分较高?

你能证明这个规律吗?

Page rank