



中國人民大學
RENMIN UNIVERSITY OF CHINA

计算传播理论与实务

2019-2020秋季学期

第四讲 网络分析

PageRank分析

授课教师：范举副教授、塔娜讲师

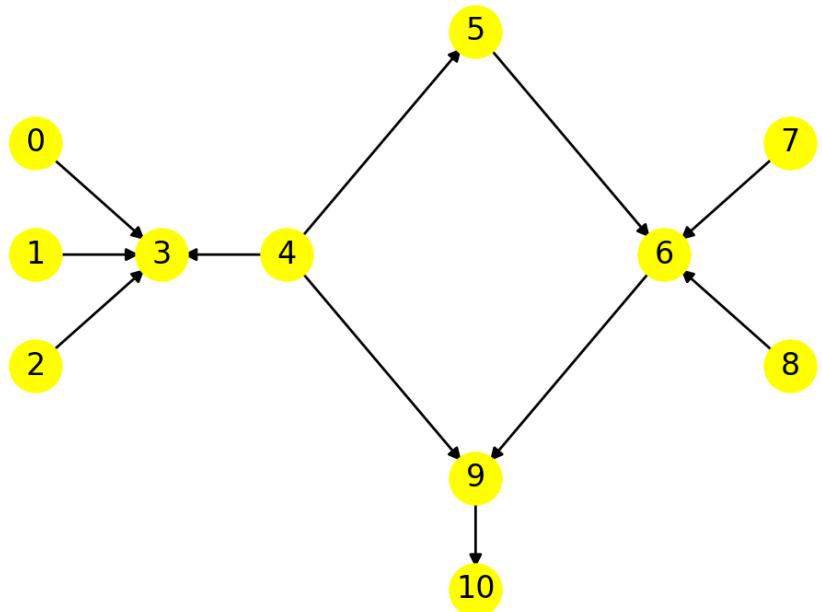
时间：2019年12月9日

考虑有向图

- 示例：简易版恋爱关系有向图
 - 定义有向边：“追求”关系



```
G = nx.DiGraph()
```



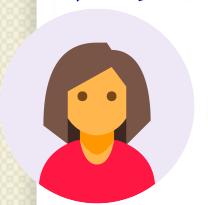
扩展阅读 : Tutorial of NetworkX

https://networkx.github.io/documentation/stable/auto_examples/index.html

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



←

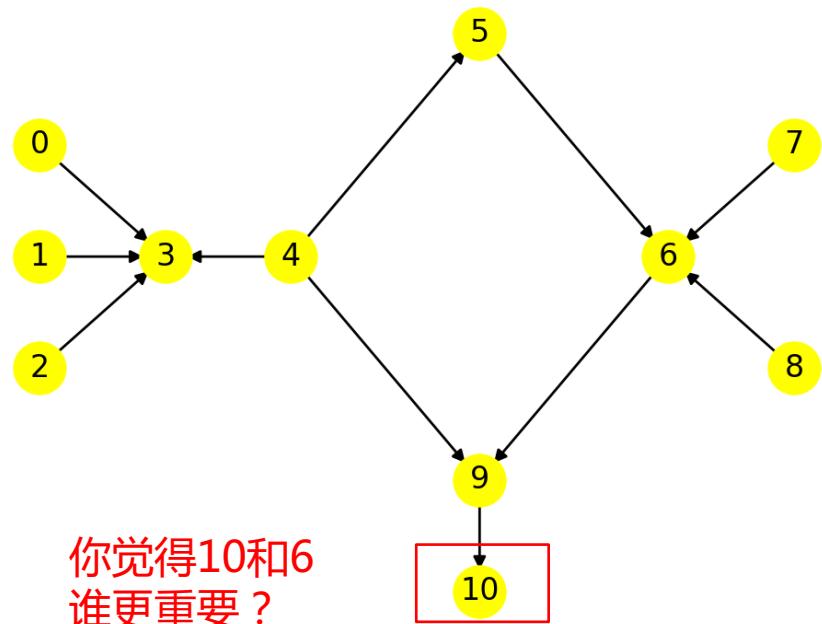


有向图

- 基于投票的思路

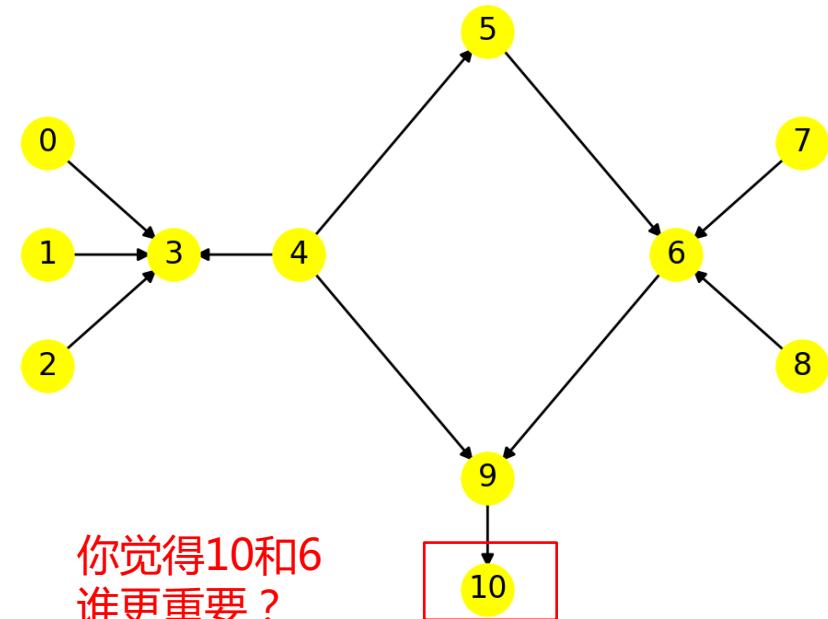
- 将每个入边看作一次投票
 - 得到的票数越多，越重要

In-Degree Centrality!



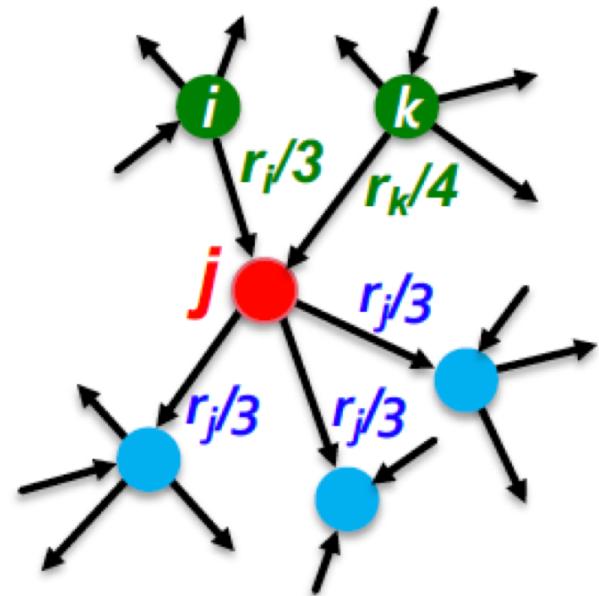
PageRank的基本思想

- In-Degree Centrality的局限性
 - 每个入边的权重相同！
- 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，对 v 相应入边的权重越低



PageRank的基本思想

- 指向 v 的节点的PageRank值越高，相应入边的权重越高
- 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
 - 以右图三个节点 i, j, k 为例



$$\begin{matrix} & i & j & k \\ i & 0 & 1 & 0 \\ j & 0 & 0 & 0 \\ k & 0 & 1 & 0 \end{matrix}$$

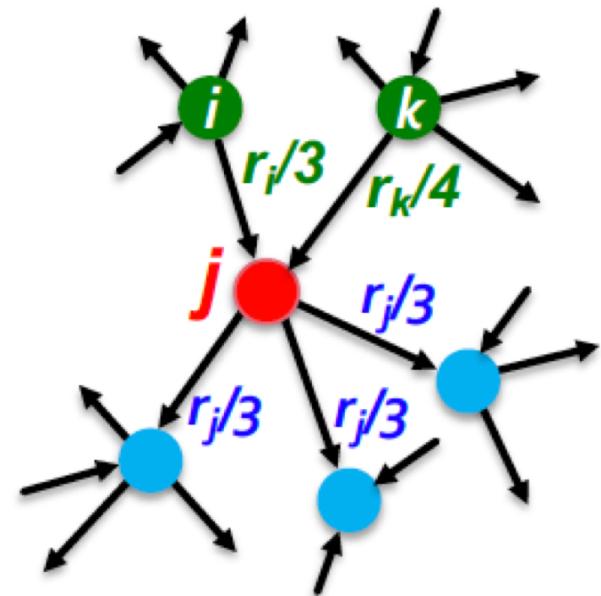
$$r_j = r_i/3 + r_k/4$$

PageRank的基本思想

- 指向 v 的节点的PageRank值越高，相应入边的权重越高
- 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

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

- 定义节点*i*的PageRank值为 p_i

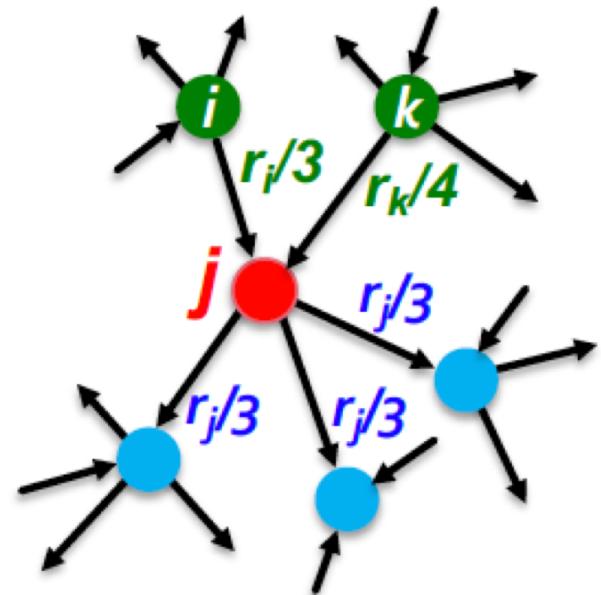


$$r_j = r_i/3 + r_k/4$$

PageRank的基本思想

- 指向 v 的节点的PageRank值越高，相应入边的权重越高
- 如何用数学表达上述想法
 - 写出PageRank值 p_i 的递推公式

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



- 思考：将上面的公式写成矩阵形式

$$r_j = r_i/3 + r_k/4$$

PageRank的矩阵计算

- 定义以下矩阵（向量）

$$p = (p_1, p_2, \dots, p_n)$$

为便于书写，采用行向量表示方法。
也可用列向量表示

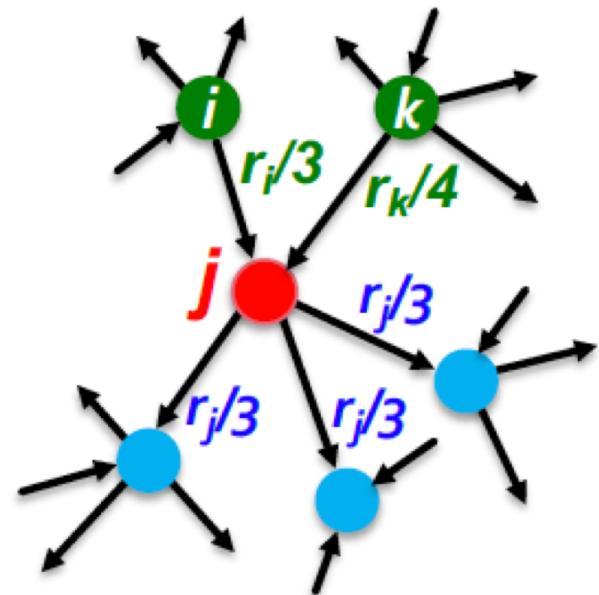
$$A = \begin{bmatrix} L_{11} & L_{12} & \dots & L_{1n} \\ L_{21} & L_{22} & \dots & L_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ L_{n1} & L_{n2} & \dots & L_{nn} \end{bmatrix}$$

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

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

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

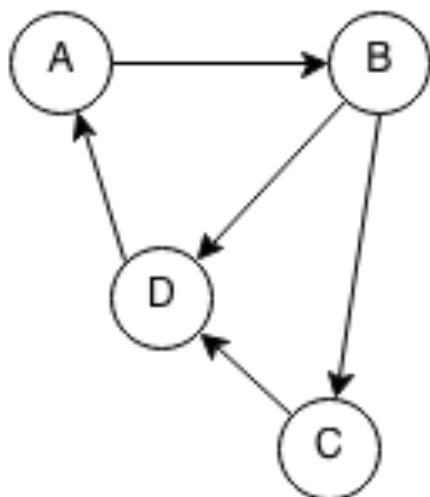
$$p^{t+1} \leftarrow p^t L$$



$$r_j = r_i/3 + r_k/4$$

PageRank的矩阵计算

- 练习：计算下图中节点的PageRank分值
 - 使用Python的numpy库



```
import numpy as np
def pagerank_naive (DiG, max_iter=200) :
    # 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 @
    n = len(DiG)
    p = np.ones(n)/n
    for i in range(max_iter):
        p = p @ L
    print (p)
```

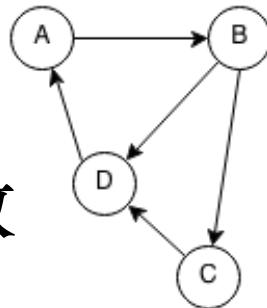
思考： PageRank的结果与 p 的初值有关吗？

- 简单起见，考虑 pagerank_fake 函数
- 考虑以下几种初始值
 - $p^0 = (1,0,0,0)$
 - $p^0 = (0,1,0,0)$
 - $p^0 = (0.25,0.25,0.25,0.25)$
- 请预测最终结果.....

```
import numpy as np
def pagerank_fake (DiG, max_iter=200) :
    # 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 @
    n = len(DiG)
    p = np.ones(n)/n
    for i in range(max_iter):
        p = p @ L
    print (p)
```

思考： PageRank的结果与 p 的初值有关吗？

- 简单起见，考虑 pagerank_fake 函数



- 考虑以下几种初始值
 - $p^0 = (1,0,0,0)$
 - $p^0 = (0,1,0,0)$
 - $p^0 = (0.25,0.25,0.25,0.25)$
- 请预测最终结果.....

```
[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]]
```

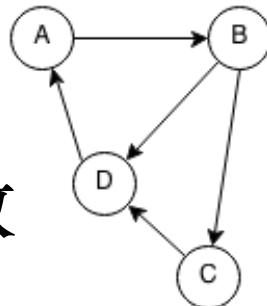
.....

```
[[0.28571429 0.28571429 0.14285714 0.28571428]]  
[[0.28571428 0.28571429 0.14285714 0.28571429]]  
[[0.28571429 0.28571428 0.14285714 0.28571429]]  
[[0.28571429 0.28571429 0.14285714 0.28571429]]  
[[0.28571429 0.28571429 0.14285714 0.28571429]]  
[[0.28571429 0.28571429 0.14285714 0.28571429]]  
[[0.28571429 0.28571429 0.14285714 0.28571429]]  
[[0.28571429 0.28571429 0.14285714 0.28571429]]  
[[0.28571429 0.28571429 0.14285714 0.28571429]]
```

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

思考： PageRank的结果与 p 的初值有关吗？

- 简单起见，考虑 pagerank_fake 函数



- 考虑以下几种初始值
 - $p^0 = (1,0,0,0)$
 - $p^0 = (0,1,0,0)$
 - $p^0 = (0.25,0.25,0.25,0.25)$
- 请预测最终结果.....

```
[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 ]]
```

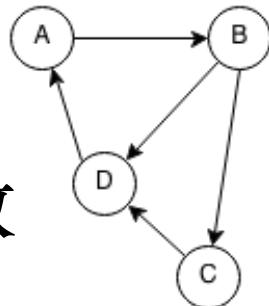
.....

```
[[0.28571429 0.28571429 0.14285714 0.28571429]]  
[[0.28571429 0.28571429 0.14285714 0.28571429]]  
[[0.28571429 0.28571429 0.14285714 0.28571429]]  
[[0.28571429 0.28571429 0.14285714 0.28571429]]  
[[0.28571429 0.28571429 0.14285714 0.28571429]]  
[[0.28571429 0.28571429 0.14285714 0.28571429]]  
[[0.28571429 0.28571429 0.14285714 0.28571429]]  
[[0.28571429 0.28571429 0.14285714 0.28571429]]  
[[0.28571429 0.28571429 0.14285714 0.28571429]]  
[[0.28571429 0.28571429 0.14285714 0.28571429]]
```

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

思考：PageRank的结果与 p 的初值有关吗？

- 简单起见，考虑 pagerank_fake 函数



- 考虑以下几种初始值
 - $p^0 = (1,0,0,0)$
 - $p^0 = (0,1,0,0)$
 - $p^0 = (0.25,0.25,0.25,0.25)$
 - 请预测最终结果.....

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

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 $P^{t+1} = p^t$

$$p = pL$$

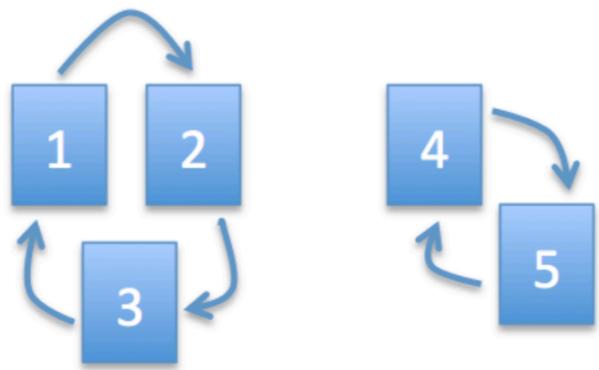
- 这说明稳定状态时， p 是矩阵 L 对应特征值为1的特征向量！
 - 可是.....
 - 我们怎么能确定 L 有为1的特征值？
 - 就算有，特征向量 p 唯一吗？

考虑一个反例.....

- 右图对应的L矩阵，特征值为1的特征向量有几个？

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

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



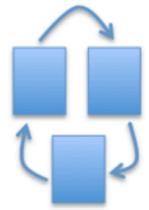
$$L = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

有关节点重要性的
结论正好相反！

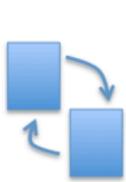
然而，真实的图结构是复杂的

- 可能会存在以下三种不强连通的情况

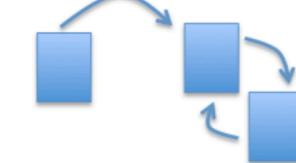
Disconnected components



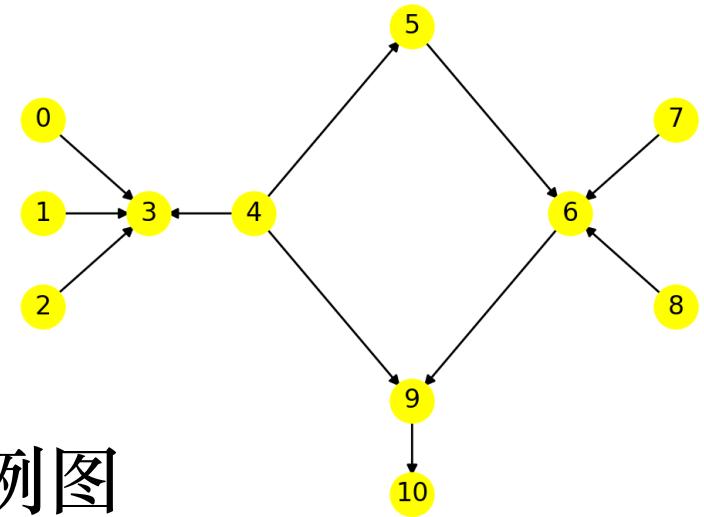
Dangling links



Loops



- 考虑我们的恋爱关系示例图
- 应该如何解决这一问题？



真正的PageRank算法

- 在前面计算的公式的基础上做了“微小”改动

$$p_i = \frac{1 - \alpha}{n} + \alpha \sum_{j \rightarrow i} \frac{p_j}{m_j} = \frac{1 - \alpha}{n} + \alpha \sum_{j=1}^n \frac{L_{ji}}{m_j} p_j$$

$$p = \alpha \mathbf{pL} + \frac{1-\alpha}{n} \mathbf{pE}, E \text{ is the } n \times n \text{ matrix of 1s}$$

α : Damping parameter, 经验上取0.85

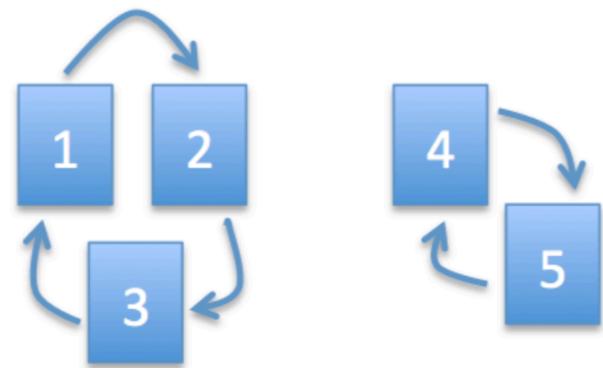
- PageRank计算的过程也称随机游走 (Random Walk)

再次考慮之前的反例.....

- 考慮 $\alpha = 0.85$

$$\begin{aligned}
 &= \frac{0.15}{5} \cdot \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} + 0.85 \cdot \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \\
 &= \begin{pmatrix} 0.03 & 0.03 & 0.88 & 0.03 & 0.03 \\ 0.88 & 0.03 & 0.03 & 0.03 & 0.03 \\ 0.03 & 0.88 & 0.03 & 0.03 & 0.03 \\ 0.03 & 0.03 & 0.03 & 0.03 & 0.88 \\ 0.03 & 0.03 & 0.03 & 0.88 & 0.03 \end{pmatrix}.
 \end{aligned}$$

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

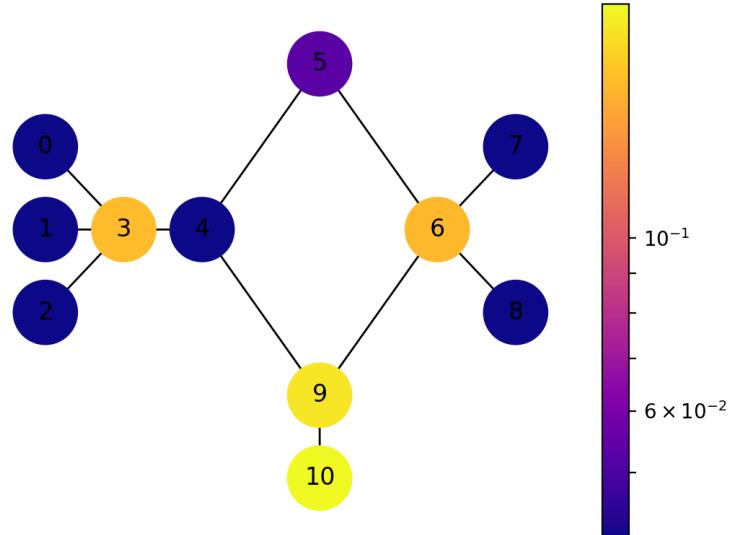


$$L = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

使用PageRank计算恋爱图的 Centrality

```
DiG, dpos =  
gen.di_school_dating_graph()  
print('DiGraph PageRank')  
print(nx.pagerank(DiG, alpha=0.85))  
draw(DiG, dpos, nx.pagerank(DiG,  
alpha=0.85), 'DiGraph PageRank')
```

DiGraph PageRank



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)
$$p = pL$$
 - 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
- 稳定状态时， p 是矩阵 L 对应特征值为1的特征向量！

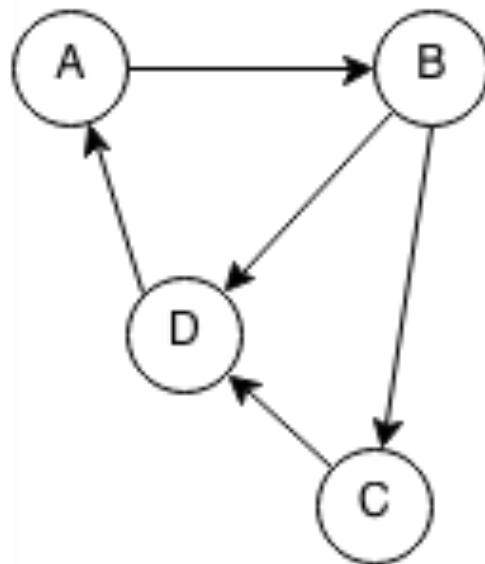
统计思维 Statistical Thinking

数据科学的必备能力之一：统计思维



Random → Deterministic
Distribution

练习



- 计算左侧图结构中节点的 pagerank 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

$$p_i = \frac{1 - \alpha}{n} + \alpha \sum_{j \rightarrow i} \frac{p_j}{m_j} = \frac{1 - \alpha}{n} + \alpha \sum_{j=1}^n \frac{L_{ji}}{m_j} p_j$$

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

PageRank在Web Search中的应用

SirsiDynix e-Library™

中国人民大学图书馆馆藏查询系统

检索/主页 院系资料查询 经济学分馆 苏研院分馆 法学图书馆 新书推荐
返回 说明 退出登录

所有字段 和

著者 和

题名 和

主题 和

丛书 和

期刊名 和

出版社

馆别: ALL
作品语种: 任何
格式: 任何
类型: 任何
馆藏位置: 任何
文献类别1: 任何
文献类别2: 任何
馆藏类别3: 任何

Baidu 百度 陈跃国 数据科学

网页 资讯 视频 图片 知道 文库 贴吧 采购 地图 更多»

百度为您找到相关结果约8,090个

...[陈跃国、杜小勇出版教材《数据科学概论》](#)... 中国人民大学信息学院

2018年4月9日 - 近日,《[数据科学概论](#)》一书正式出版。该书由信息学院覃雄派、[陈跃国](#)、杜小勇历经5年授课、2年编写而成,系中国人民大学出版社推出的[“数据科学”](#)与大数据...

info.ruc.edu.cn/news_c... - 百度快照

[覃雄派、陈跃国、杜小勇编著《数据科学概论》的主要特...](#) 新浪博客

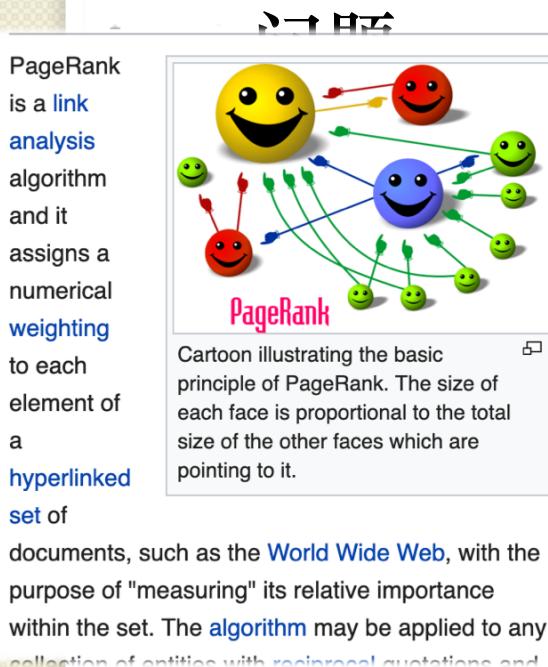
2018年3月27日 - 覃雄派、[陈跃国](#)、杜小勇编著《[数据科学](#)概论》的主要特点 (2018-03-27 15:59:18) 转载▼ 分类: [数据科学](#) 概论 (1)中国人民大学数据库研究团队基于长期...

blog.sina.com.cn/s/blog... - 百度快照

- **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）**

PageRank在Web Search中的应用

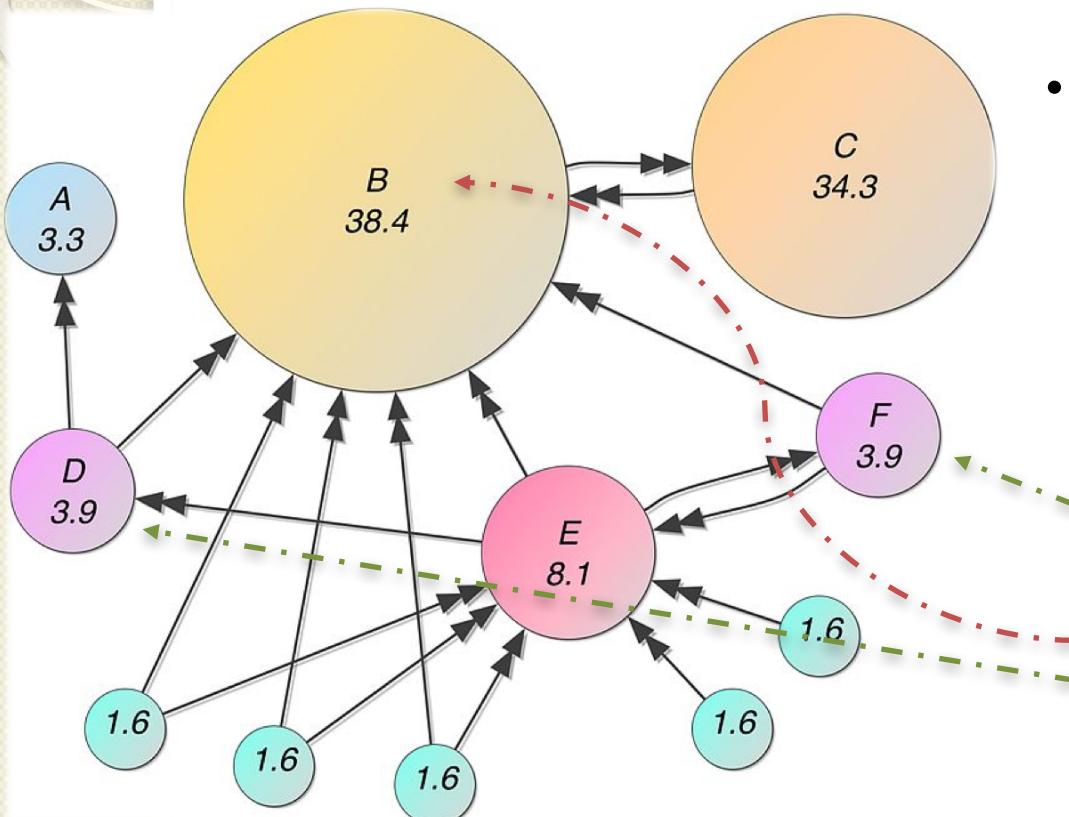
- PageRank由谷歌公司的两个创始人Larry Page和Sergei Brin提出，主要解决Web Page的排序



PageRank is a "
link analysis
" algorithm and it assigns a numerical "
weighting
" to each element of a "
hyperlinked
set
" of documents, such as the "
World Wide Web == \$0
", with the purpose of "measuring" its relative importance within the set. The "
algorithm
" may be applied to any collection of entities with "
reciprocal
" quotations and references. The numerical weight that it assigns to any given element "
<i>E</i>
" is referred to as the "
<i>PageRank of E</i>
" and denoted by "

```
html body #content #bodyContent div#mw-content-text.mw-content-ltr div.mw-parser-output p a
```

Personalized PageRank



- 考虑新的场景
 - 如果用户已经收藏了网页D和F，希望最后算出的分數反映出这种偏好
 - 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數
- 个性化
 - 那些都是很好的可我偏偏不喜欢——《白马啸西风》

Personalized PageRank

- 在前面计算的公式的基础上做了“微小”改动

$$p_i = (1 - \alpha)p_i^{(0)} + \alpha \sum_{j \rightarrow i} \frac{p_j}{m_j} = (1 - \alpha)p_i^{(0)} + \alpha \sum_{j=1}^n \frac{L_{ji}}{m_j} p_j$$

$$p = \alpha \mathbf{p}L + (1 - \alpha) \mathbf{p}^{(0)}$$

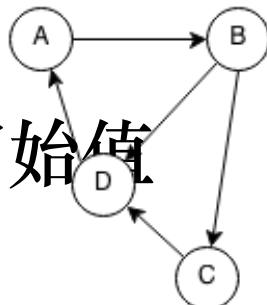


α : Damping parameter, 经验上取0.85

- 计算的过程也称Random Walk with Restart

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

- $p^0 = (1,0,0,0)$

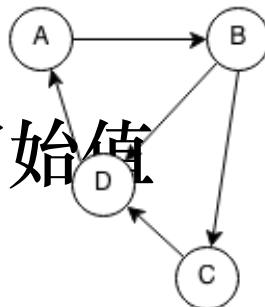
- $p^0 = (0,1,0,0)$

- $p^0 = (0.5, 0.5, 0, 0)$

- 请预测最终结果.....

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

- $p^0 = (1,0,0,0)$

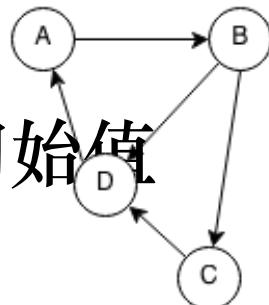
- $p^0 = (0,1,0,0)$

- $p^0 = (0.5, 0.5, 0, 0)$

- 请预测最终结果.....

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.....

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

- $\circ p^0 = (1,0,0,0)$ [[0.34727498 0.29518373 0.12545309 0.23208821]]

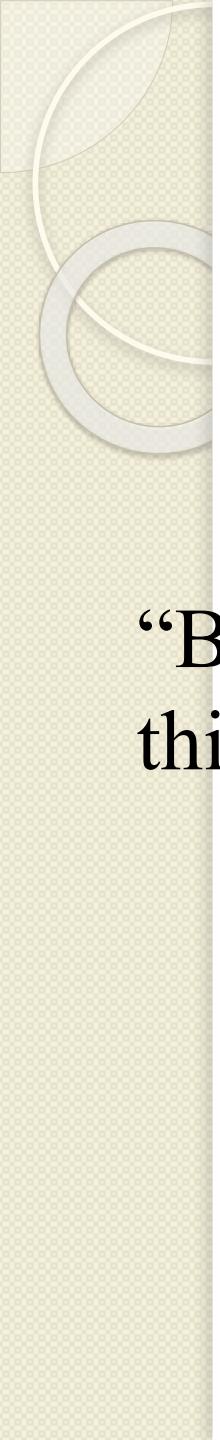
- $\circ p^0 = (0,1,0,0)$ [[0.23208821 0.34727498 0.14759187 0.27304495]]

- $\circ p^0 = (0.5,0.5,0,0)$ [[0.28968159 0.32122935 0.13652248 0.25256658]]

- 你能发现什么规律？

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?）
 - 马尔科夫链的数学本质
 - 个性化PageRank算法



“Beautiful math tends to be useful, and useful things tend to have beautiful math.”