PageRank与图挖掘

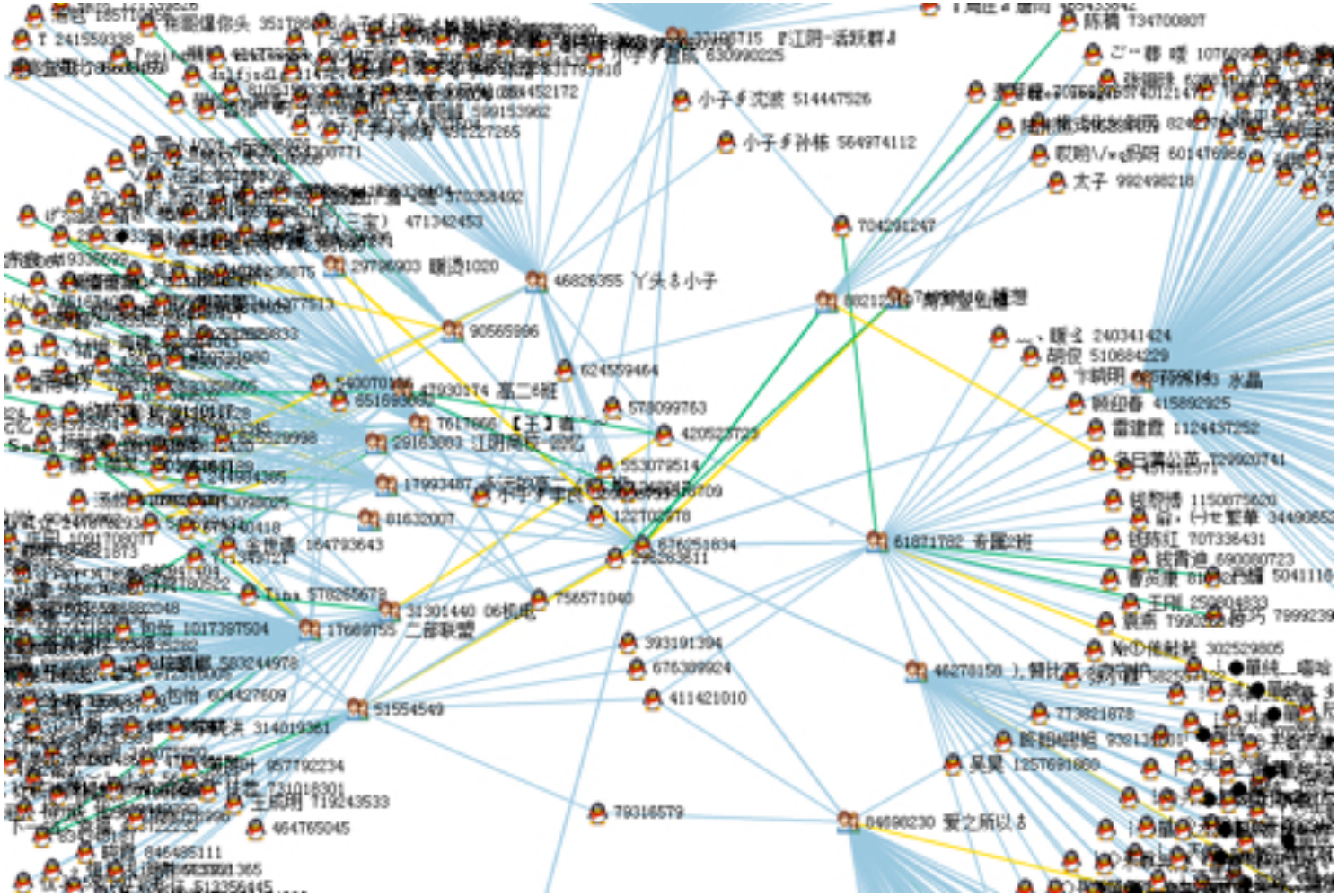
七月在线：寒老师
2016-07-24

图数据：社交网络

Facebook social graph

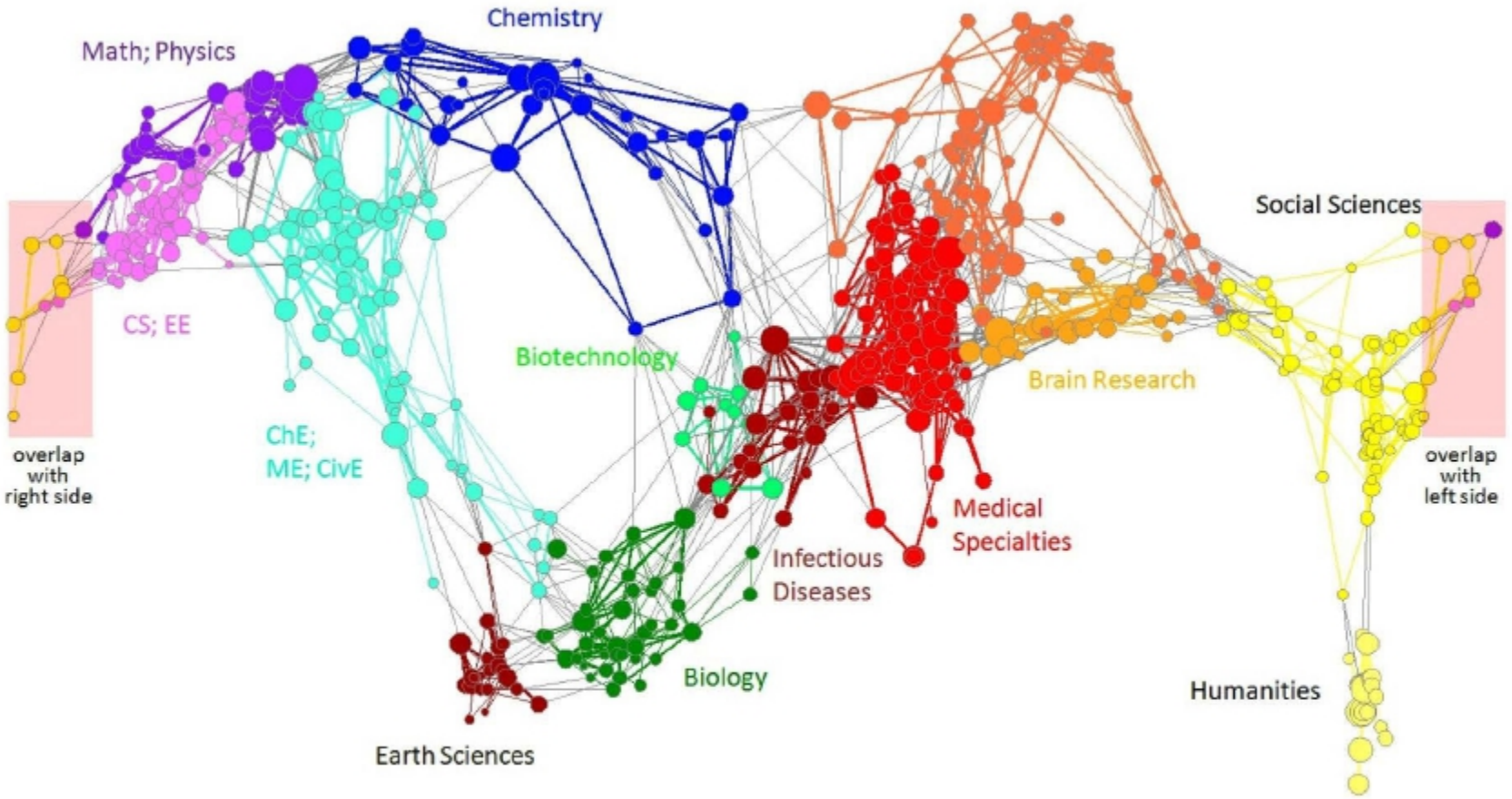
4-degrees of separation [Backstrom-Boldi-Rosa-Ugander-Vigna, 2011]

图数据：社交网络



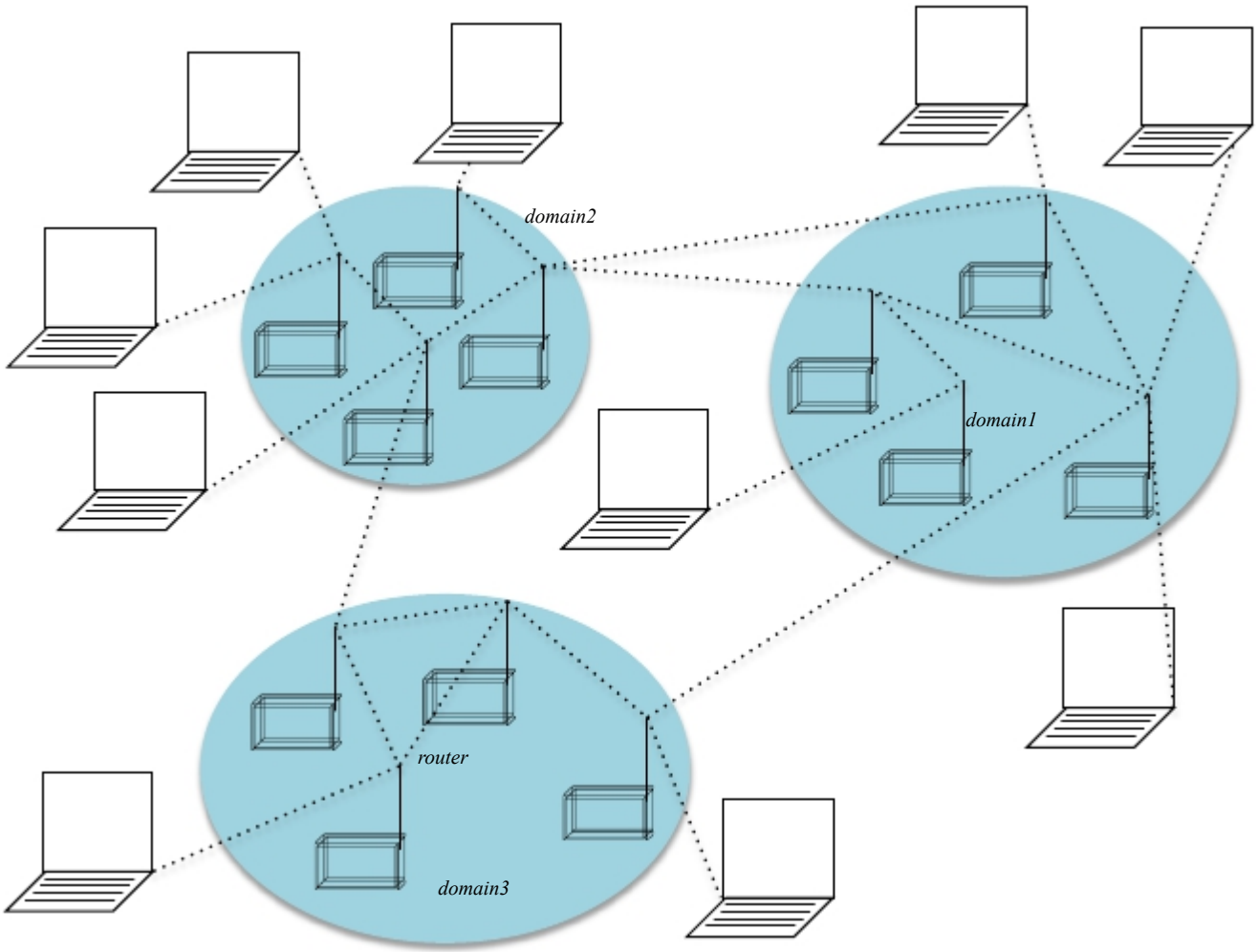
QQ群关系网

图数据：学科知识识网



科学分支交叉的状况(根据论文引用情况)

图数据：通信网



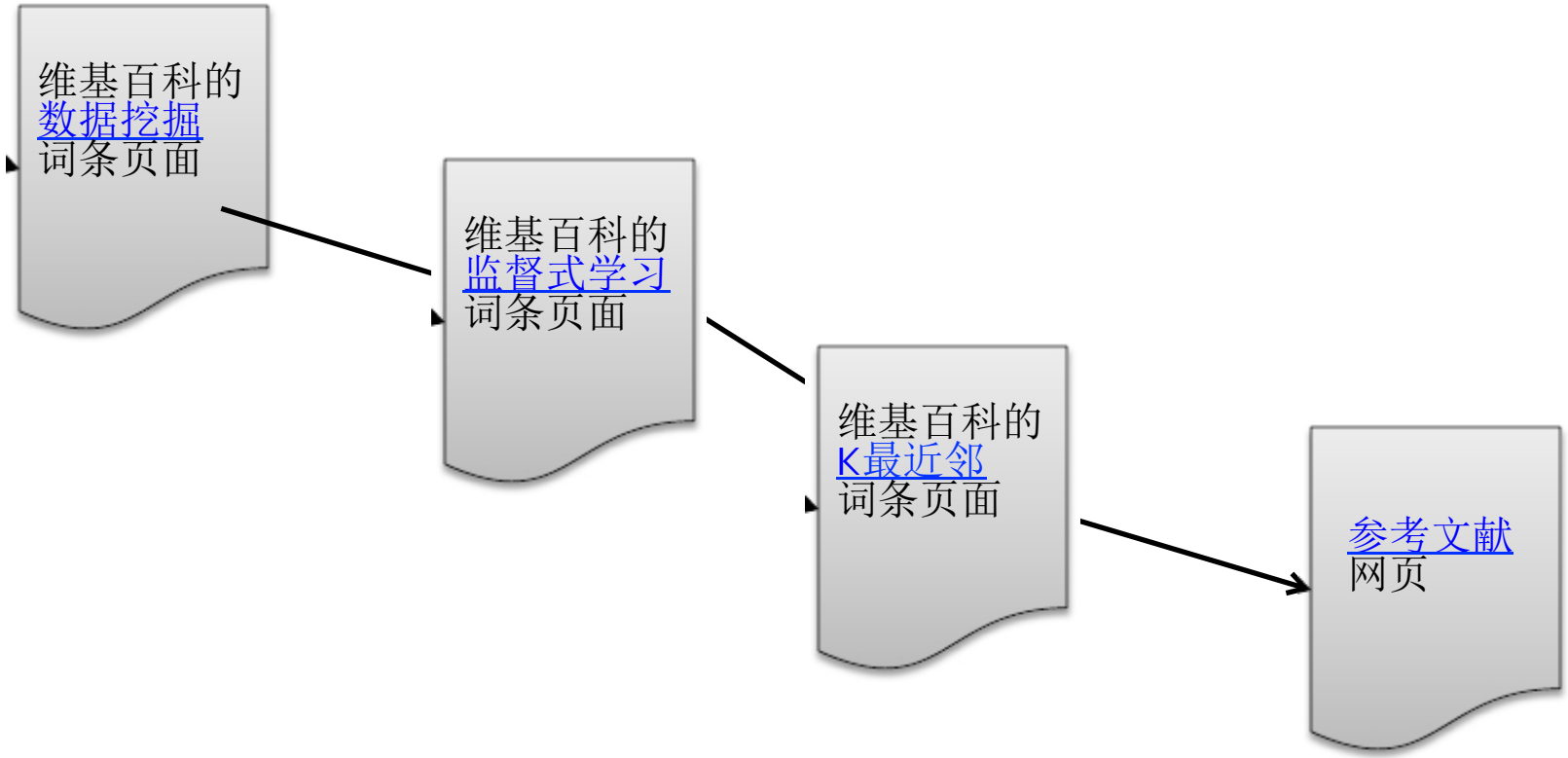
Internet

将web(众多网页)虚拟成有向图

Web作为一张有向图

节点 → 网页

边 → 超链接



更为广泛的一些问题

如何组织网页?

- 初次尝试: 人为分类
  - 逐级分类的网站
    - Yahoo, DMOZ, LookSmart
- 二次尝试: 网页搜索
  - 信息检索研究:

    在一堆小的可信任的数据集中查找相关文档

    例如报纸文章，专利

但是: 网络太大了，而且充斥着不安全网页，无关信息，等等

网络搜索的2个挑战

网络搜索需要应对的2个问题:

(1) 网页包含太多信息

哪些是可信的?

提示: 可信任度高的页面会通过超链接互相关联!

(2) 例如搜索“报纸”，什么算是最佳搜索结果

没有明确的答案

提示: 真正对报纸有研究的网页可能就会指向多种我们需要的报纸

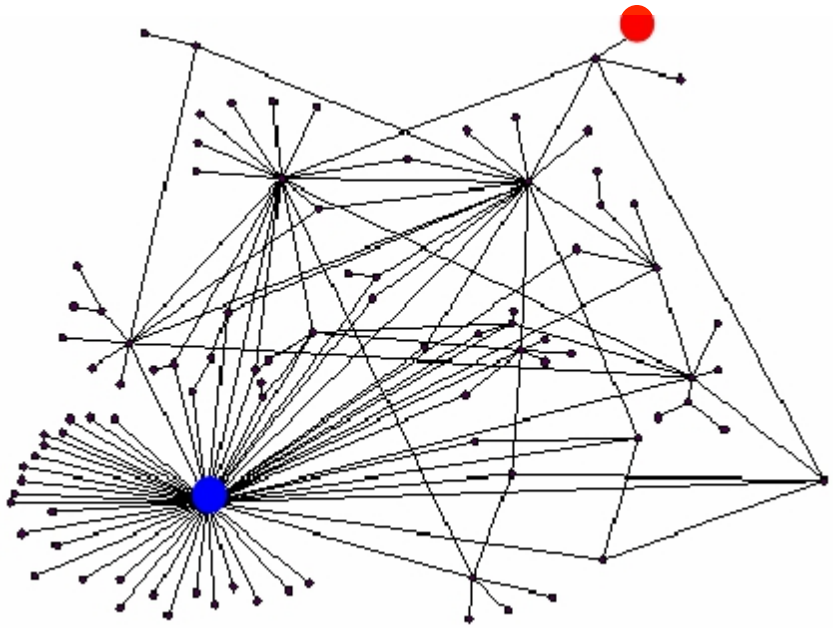
对图中的节点按权重排名

并不是每个网页都是同样重要的

http://php.itcast.cn/ vs. www.tsinghua.edu.cn/

Web图节点之间的连接关系
有巨大的多样性.

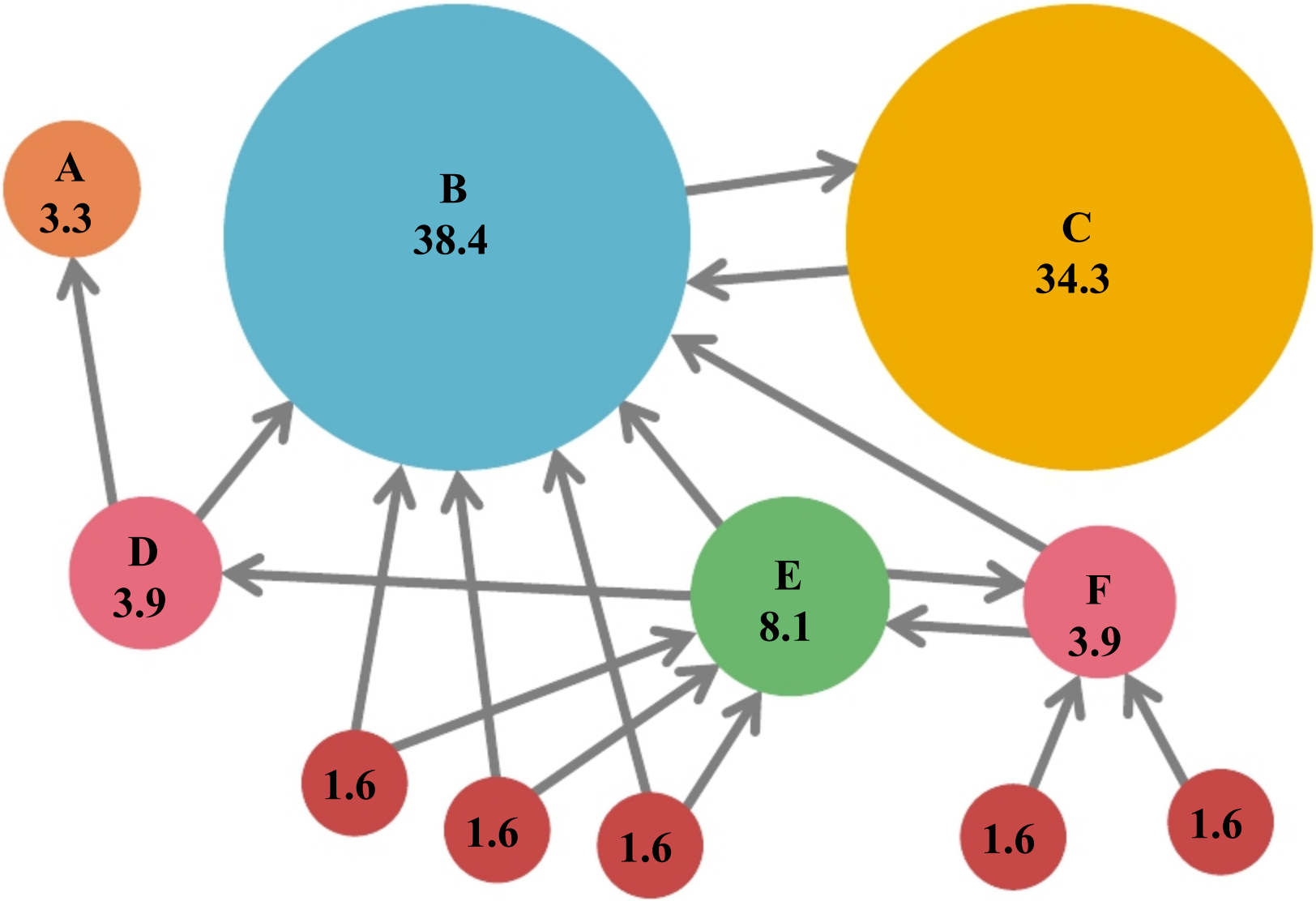
我们将根据图的链接结构
进行节点排名!



将链接视作投票

- 想法: 将链接视作投票
  - 我们认定一个页面的链接越多越重要
    - 入链(指向该页面的)？　出链(该页面指出去的)？
- 将所有指向本页面的链接视作投票:
  - www.stanford.edu has 23,400 in-links
  - www.joe-schmoe.com has 1 in-link
- 所有入链对投票的影响程度一致吗?
  - 不一致！ 重要度高的页面指过来的入链作用更大
  - 化为一个递归的问题

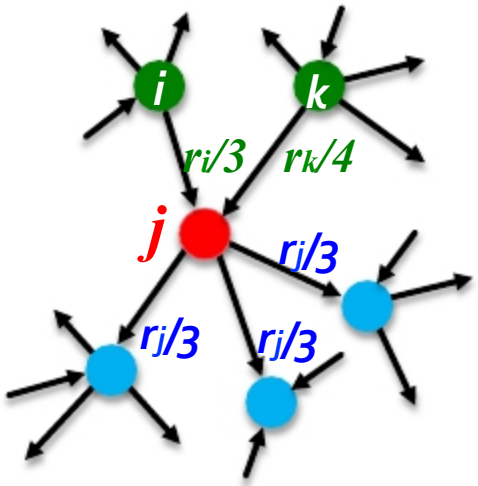
PageRank分数示例



简单的递归公式

- 每条链接的投票影响度与其来源网页的重要性比例
- 如果页面*j* 重要度为 r_j 且有 *n* 条出链，
则每条出链通过投票，能传递 r_j / n 的重要度
- 页面 *j*自身的重要度取决于它的所有入链传递给它的重要度之和。

$r_j = r_i/3 + r_k/4$

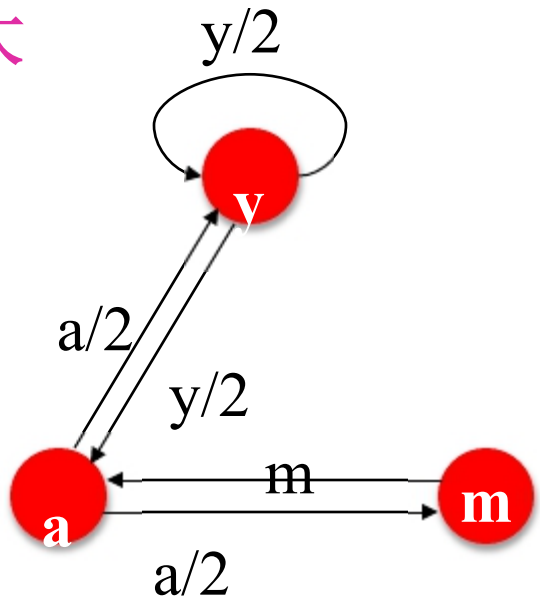


PageRank与“流”模型

- 来自重要度高的网页的出链对重要度影响大
- 一个网页若被另外的重要网页指向，那么它的重要性也相应很高
- 为每个网页*j*定义一个重要度*rj*

$$r_j = \sum_{i \to j} \frac{r_i}{d_i}$$

其中 d_i 为节点i的点出度(out-degree)



“流”公式:

$r_y = r_y/2 + r_a/2$

$r_a = r_y/2 + r_m$

$r_m = r_a/2$

求解“流”方程

- 3个方程，3个未知数，没有常数
  - 解不唯一 ，任何一组解的倍数也是解

Flow equations:

$r_y = r_y/2 + r_a/2$

$r_a = r_y/2 + r_m$

$r_m = r_a/2$

- 我们添加一个附加限制条件:

$$r_a + r_y + r_m = 1$$

解得: $r_a = 2/5 \quad r_y = 2/5 \quad r_m = 1/5$

- 对于低维度的方程组，我们直接用消元法可解，
  对于网页数极多的实际情况，我们需要别的方法
- 我们需要新的公式！

PageRank的矩阵方程

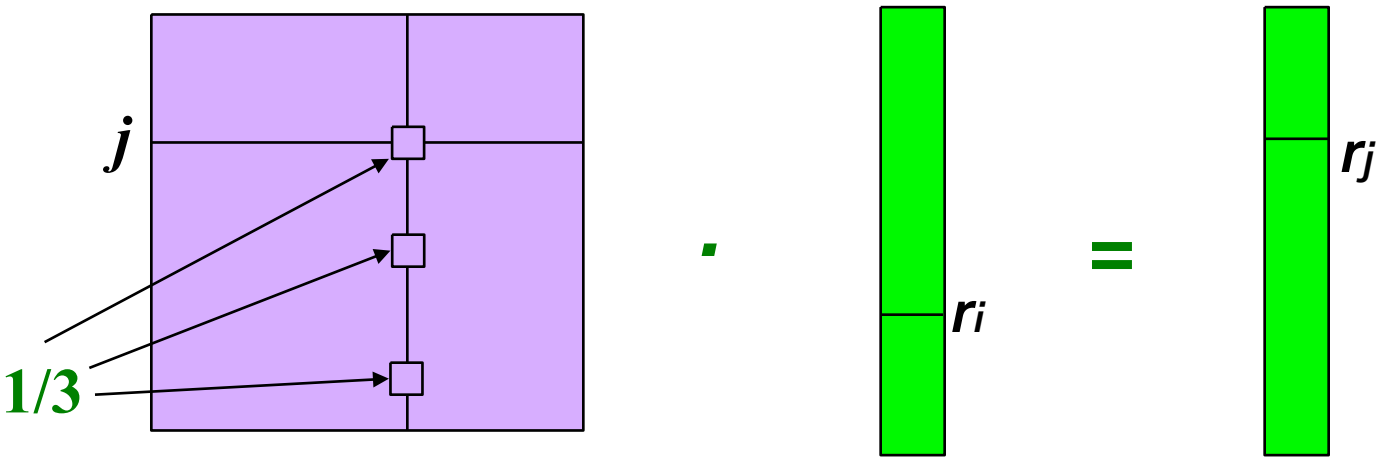
- 随机邻接矩阵
  - 假定页面i有d_i个出链
  - 如果$i \to j$，则$M_{ji}=\frac{1}{d_i}$否则$M_{ji}=0$
    - M是一个列随机矩阵
    - 每一列和为1
- 网页重要度向量r:
  - r_i表明第i个页面的重要度
  - $\sum_i r_i = 1$
- “流”公式用矩阵可表示成:
  - $r = M \cdot r$

$$r_j = \sum_{i \to j} \frac{r_i}{d_i}$$

PageRank的矩阵方程解释

- “流”公式：$r_j = \sum_{i \to j} \frac{r_i}{d_i}$
- “流”公式的矩阵形式：**r = M·r**

假设页面*i*有三个外链，*j*是其中一个



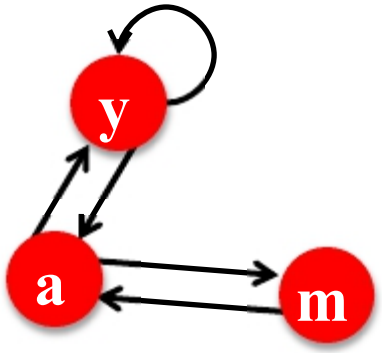
特征向量公式

- “流”公式的矩阵形式：$r = M \cdot r$

- 所以其实重要度向量*r*是网页随机矩阵M的一个特征向量
  - 事实上是对应特征值1的主特征向量
  - 又因为M为列随机矩阵，M最大的特征向量就应该是1

- 我们通过幂迭代可以有效求解出r

“流”矩阵方程示例



	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

$$r = M \cdot r$$

$$r_y = r_y/2 + r_a/2$$

$$r_a = r_y/2 + r_m$$

$$r_m = r_a/2$$

$$\begin{bmatrix} y \\ a \\ m \end{bmatrix} = \begin{bmatrix} ½ & ½ & 0 \\ ½ & 0 & 1 \\ 0 & ½ & 0 \end{bmatrix} \begin{bmatrix} y \\ a \\ m \end{bmatrix}$$

幂迭代方法

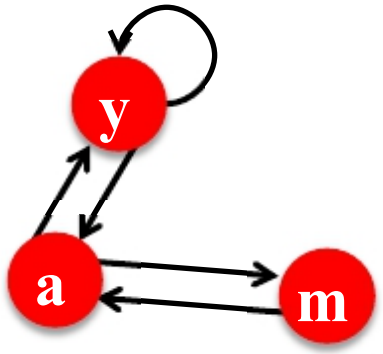
- 将一个网页关系图模拟成一个有向图，图的节点是网页，而边为超链接
- 幂迭代 ：一个简单的迭代方法
  - 假设: 总共有***N*** 个网页
  - 初始化: $\mathbf{r}^{(0)} = [1/N,....,1/N]^T$
  - 迭代: $\mathbf{r}^{(t+1)} = \mathbf{M} \cdot \mathbf{r}^{(t)}$
  - 停止迭代条件： $|\mathbf{r}^{(t+1)} - \mathbf{r}^{(t)}|_1 < \varepsilon$
    - $|\mathbf{x}|_1 = \Sigma_i|x_i|$ 是1范数，绝对值之和，即曼哈顿距离

$$r_j = \sum_{i \to j} \frac{r_i}{d_i}$$

其中 d_i 为节点i的点出度(out-degree)

PageRank：求解方法

- 幂迭代 ：
- 设: $r_j = 1/N$
- 1: $r'_j = \sum_{i \to j} \frac{r_i}{d_i}$
- 2: $r = r'$
- 未收敛： 回到1



	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

$r_y = r_y/2 + r_a/2$

$r_a = r_y/2 + r_m$

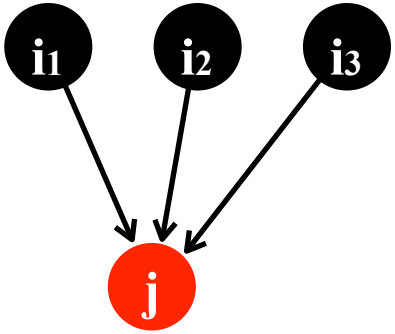
$r_m = r_a/2$

$$\begin{pmatrix} r_y \\ r_a \\ r_m \end{pmatrix} = \begin{matrix} 1/3 & 1/3 & 5/12 & 9/24 & & 6/15 \\ 1/3 & 3/6 & 1/3 & 11/24 & \ldots & 6/15 \\ 1/3 & 1/6 & 3/12 & 1/6 & & 3/15 \end{matrix}$$

Iteration 0, 1, 2,

随机游动(random walk)的解释

- 假定我们现在有一个随机的网页浏览者：
  - 在时间t，浏览者在页面i上浏览
  - 在下一个时间t+1，浏览者随意挑选一个i的出链到下一个页面浏览
  - 随着i的出链到达页面j
  - 以上的操作无限进行着



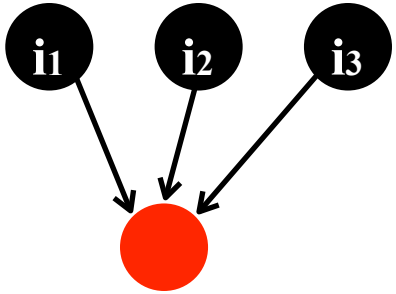
$$r_j = \sum_{i \to j} \frac{r_i}{d_{out}(i)}$$

- 设定：
  - *p(t)*为一个向量，其第*i*个元素代表浏览者在时间*t*浏览页面*i*的概率
  - 所以*p(t)*可视作一个网页的概率分布

平稳分布

- 浏览者时刻t+1在哪呢？
  - *浏览者随机跟随一个出链到达下一个页面：*

$$P(t+1) = M \cdot p(t)$$



- 假设随机游动达到一个状态：

$$P(t+1) = M \cdot p(t) = p(t)$$

则此时***p(t)***是随机游动的平稳分布

- 而我们原始重要度向量***r***满足 $r = M \cdot r$：
  - *所以，r是随机游动的一个平稳分布*

存在性和唯一性

- 随机漫步(又名马尔可夫过程)得出的一个核心结论:

对于满足特定条件的图，其平稳状态是唯一的，而且无论在时间t=0时起始概率分布是怎么样的，最终都会达到这样一个平稳状态。

PageRank：3个问题

$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{d_i}$$

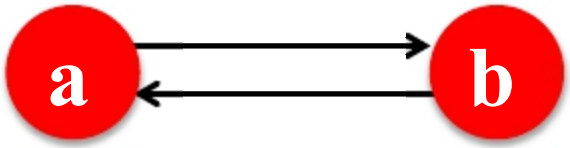
或者写成

$$r = Mr$$

- 按照这个公式迭代一定收敛吗？
- 它会收敛到我们想要的结果吗？
- 我们得到的结果合理吗？

收敛吗？

- 陷阱(“Spider trap”)问题:



$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{d_i}$$

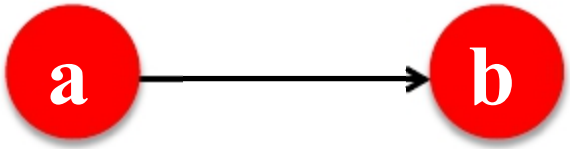
例子:

$$\begin{matrix} r_a \\ r_b \end{matrix} = \begin{matrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{matrix}$$

Iteration 0, 1, 2,

它收敛到我们想要的结果吗？

- 终结点(“Dead end”)问题:



$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{d_i}$$

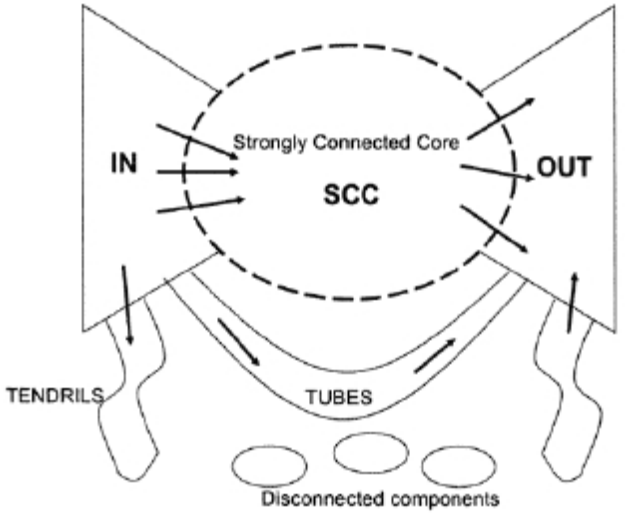
例子:

$$\begin{matrix} r_a \\ r_b \end{matrix} = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{matrix}$$

Iteration 0, 1, 2,

PageRank：问题

2 个问题:



(1) 有一些页面是
终结点 (没有任何出链/out-links)

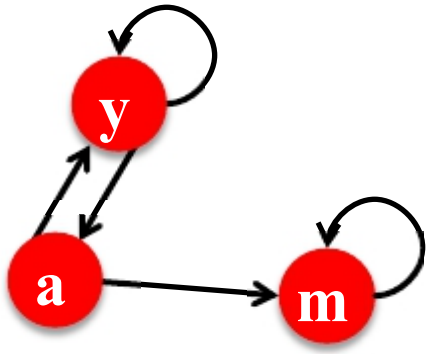
- 这样的页面导致我们传递的重要度“泄露”了

(2) 陷阱问题
(外链组成环形结构)

最终这个陷阱会像天体中的“黑洞”一样吸收掉所有的重要度

陷阱问题

- 幂迭代:
  - 设定$r_j = 1$
  - $$r_j = \sum_{i \to j} \frac{r_i}{d_i}$$
    - 进行迭代



	y	a	m
y	½	½	0
a	½	0	0
m	0	½	1

$r_y = r_y/2 + r_a/2$

$r_a = r_y/2$

$r_m = r_a/2 + r_m$

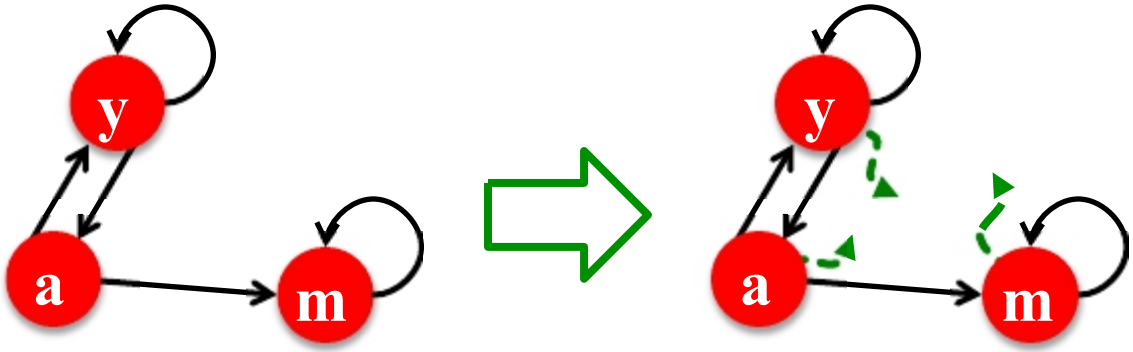
- 例子:

$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{matrix} 1/3 & 2/6 & 3/12 & 5/24 & & 0 \\ 1/3 & 1/6 & 2/12 & 3/24 & \cdots & 0 \\ 1/3 & 3/6 & 7/12 & 16/24 & & 1 \end{matrix}$$

Iteration 0, 1, 2,

解决方法：随机传送(random teleport)

- **Google解决方法：**
  在每个时间节点，浏览者对于下个网页有两种选择
  - 以概率β，随机跟随一个外链到下个网页
  - 以概率1-β，随机跳到某个网页
  - 通常β的取值在0.8到0.9之间
- 即使图中存在陷阱，
  在几次尝试之后，浏览者也会离开陷阱

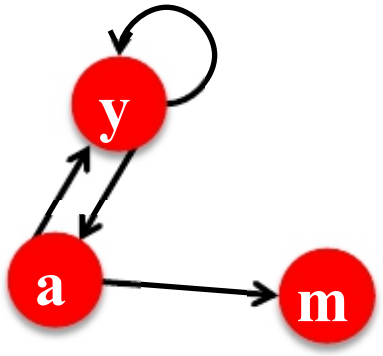


终结点问题

- 幂迭代:
  - 设定$r_j = 1$

$$r_j = \sum_{i \to j} \frac{r_i}{d_i}$$

  - 进行迭代



	y	a	m
y	½	½	0
a	½	0	0
m	0	½	0

$r_y = r_y/2 + r_a/2$

$r_a = r_y/2$

$r_m = r_a/2$

Example:

$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{matrix} 1/3 & 2/6 & 3/12 & 5/24 & & 0 \\ 1/3 & 1/6 & 2/12 & 3/24 & \cdots & 0 \\ 1/3 & 1/6 & 1/12 & 2/24 & & 0 \end{matrix}$$

Iteration 0, 1, 2,

图算法案例1 && 2

见课上ipython notebook

工作与面试要点

欢迎课上一起交流讨论

动手试试

把github挖据的项目中user改一改
对热门的项目(Tensorflow)挖据一下

感谢大家！

恳请大家批评指正！