

信息检索与Web搜索

第15讲 链接分析

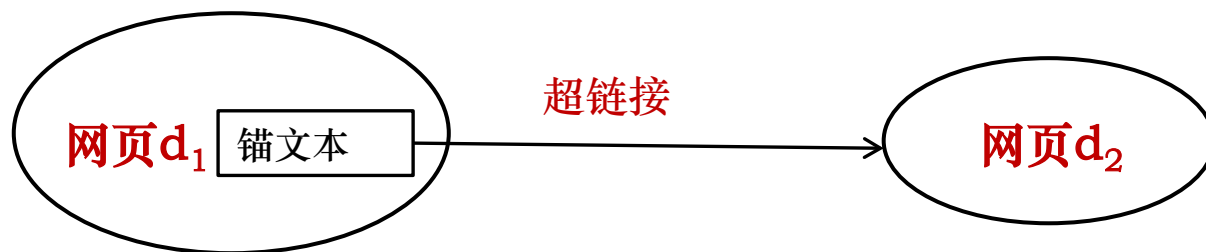
Link Analysis

授课人：高曙明

概述

- **链接分析：** 是对网页的链接结构信息进行分析，以定量刻画网页权威度的技术
- **必要性：** 网页质量差别巨大，造成传统检索结果排序方法效果不佳
- **思想起源：** 引文分析（60年代提出）
- **难点：** 如何定量地区别网页质量好差
- **两个算法：** PageRank, HITS

超链接和锚文本



- 超链接代表了某种质量认可信号
 - 超链 $d_1 \rightarrow d_2$ 表示 d_1 的作者认可 d_2 的质量和相关性
- 锚文本描述了文档 d2 的内容
 - 这里的锚文本定义比较宽泛，包括链接周围的文本
 - 例子：“You can find cheap cars here .”
 - 锚文本：“You can find cheap cars here”

网页文本 VS. 锚文本

□ 后者往往效果好于前者

- 很多网页文本并不包含对自身的精确描述
- 很多网页上大部分是图
- 锚文本是对网页的简洁描述，一般与查询文本更相似，聚集了多个Web网页作者的集体力量

□ 例子：查询 IBM

- IBM的版权页、很多作弊网页、IBM的wikipedia页面匹配上，但可能与IBM 的主页并不匹配!

□ 利用锚文本可以提高搜索效果

指向www.ibm.com的很多锚文本中包含IBM

www.nytimes.com: "IBM acquires Webify"

www.slashdot.org: "New IBM optical chip"

www.stanford.edu: "IBM faculty award recipients"



```
graph TD; A["www.nytimes.com: 'IBM acquires Webify'"] -.-> D["www.ibm.com"]; B["www.slashdot.org: 'New IBM optical chip'"] -.-> D; C["www.stanford.edu: 'IBM faculty award recipients'"] -.-> D;
```

www.ibm.com

锚文本的使用

- 构建索引时将锚文本与目标网页的文本一起使用
- 在计算过程中，锚文本词项被赋予更高的权重
 - 通常也会基于词频计算锚文本词项的权重
 - 诸如Click、here的高频词会受到惩罚
- 需要检测与处理误导性的锚文本

Google炸弹 (Google bomb)

- ❑ Google炸弹是指由于人为恶意构造锚文本而导致的结果很差的搜索
- ❑ 2007年1月Google引入了一个新的权重计算公式来修正了很多Google炸弹的结果
- ❑ 但是还有不少没有解决: [dangerous cult] on Google, Bing, Yahoo
- ❑ 已解决的Google炸弹: [dumb motherf...], [who is a failure?], [evil empire]

PageRank 的起源：引用分析

- 引用分析：科技文献被引用情况的分析
- 一个引用的例子：“Miller (2001) has shown that physical activity alters the metabolism of estrogens.”
- 可以把 “Miller (2001)” 看成是两篇学术文献之间的超链接
- 科技文献领域使用 “超链接” 的两个应用
 - 根据引用频率来度量一篇文档的影响度
 - 根据他人引用的重合率来度量两篇文献的相似度(共引相似度)
 - 在Web上也存在共引相似度：Google中提供的 “find pages like this” 或者 “Similar” 功能

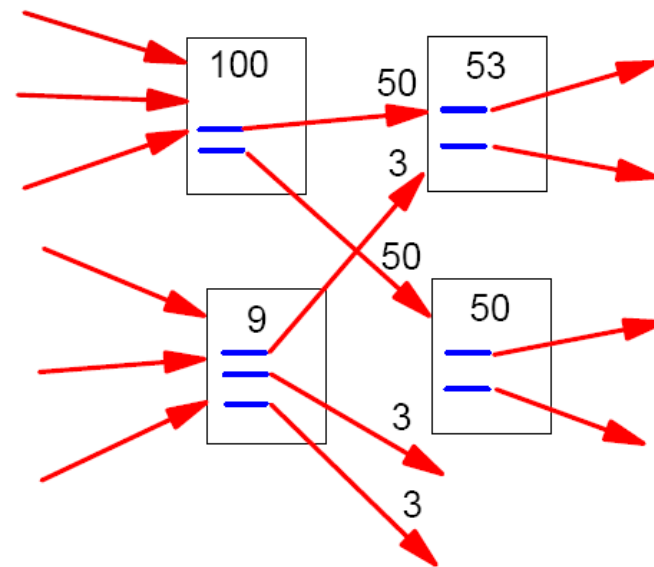
PageRank起源：引用分析

- 在Web上：引用频率 = 入链数
 - 入链数目大并不意味着高质量...
 - ... 主要原因是因为存在大量作弊链接...
- 更好的度量方法：对不同网页来的引用频率进行加权
 - 一篇文档的投票权重来自于它本身的质量（引用频率）
 - PageRank：基于链接分析对Web图中的每个节点所赋予的一个0到1之间的分值

原始的PageRank公式

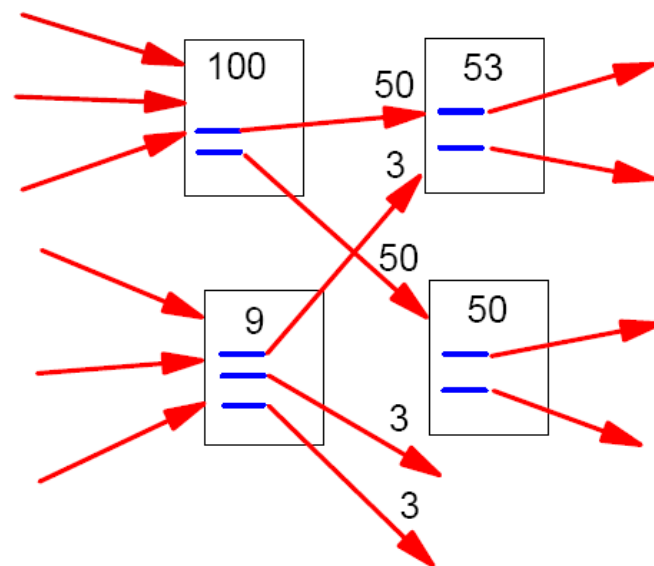
$$R(u) = c \sum_{v \in B_u} \frac{R(v)}{N_v}$$

- $R(u)$ 和 $R(v)$ 分别是网页 u 、 v 的PageRank值, B_u 是指向网页 u 的网页集合、 N_v 是网页 v 的出链数目
- 一个网页的PageRank等于所有的指向它的网页的PageRank的分量之和(c 为归一化参数), 网页的每条出链上每个分量上承载了相同的PageRank分量

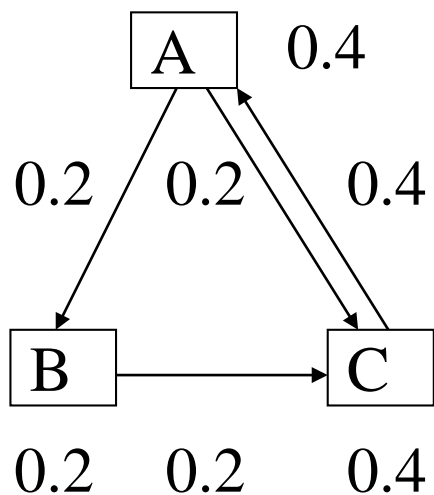


PageRank 的特点

- (1) 一个网页如果它的入链越多，那么它也越重要(PageRank越高);
- (2) 一个网页如果被越重要的网页所指向，那么它也越重要(PageRank越高)。



简单计算的例子($c=1$)



$$R(A) = R(C)$$

$$R(B) = 0.5R(A)$$

$$R(C) = R(B) + 0.5R(A)$$

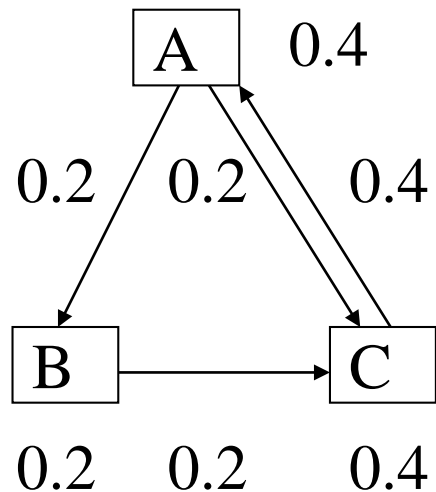
$$R(A) + R(B) + R(C) = 1$$

解上述方程得：

$$R(A) = R(C) = 0.4$$

$$R(B) = 0.2$$

简单计算的例子：迭代法求解



$$R(A)=R(C)$$

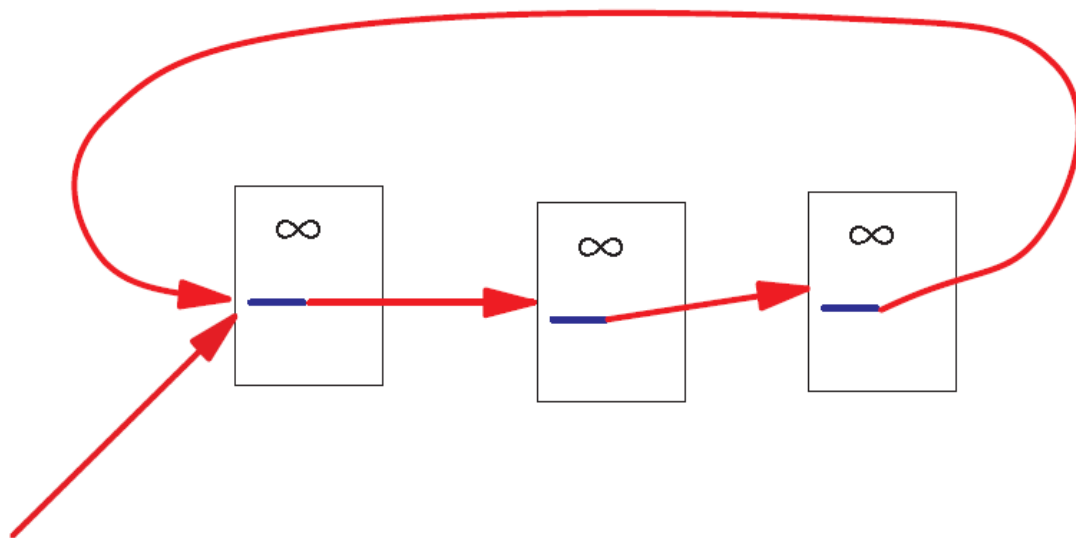
$$R(B)=0.5R(A)$$

$$R(C)=R(B)+0.5R(A)$$

$$R(A)+R(B)+R(C)=1$$

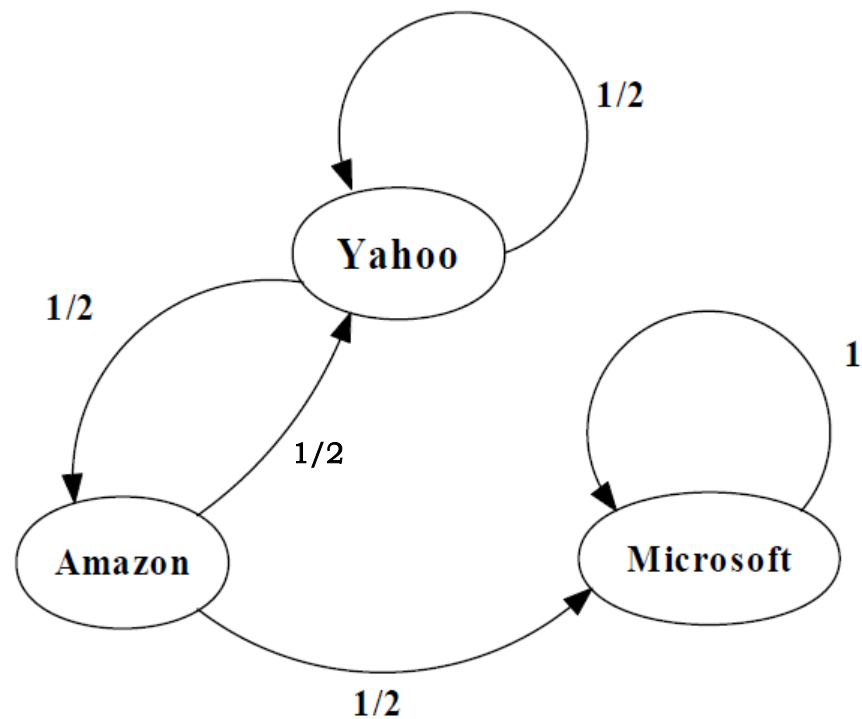
迭代次数	R(A)	R(B)	R(C)
0	1/3	1/3	1/3
1	1/3	1/6	1/2
2	1/2	1/6	1/3
3	1/3	1/4	5/12
...
收敛	2/5	1/5	2/5

原始PageRank的一个不足



- ❑ 图中存在一个循环通路，每次迭代，该循环通路中的每个节点的PageRank不断增加，但是它们并不指出去，即不将PageRank分配给其他节点！

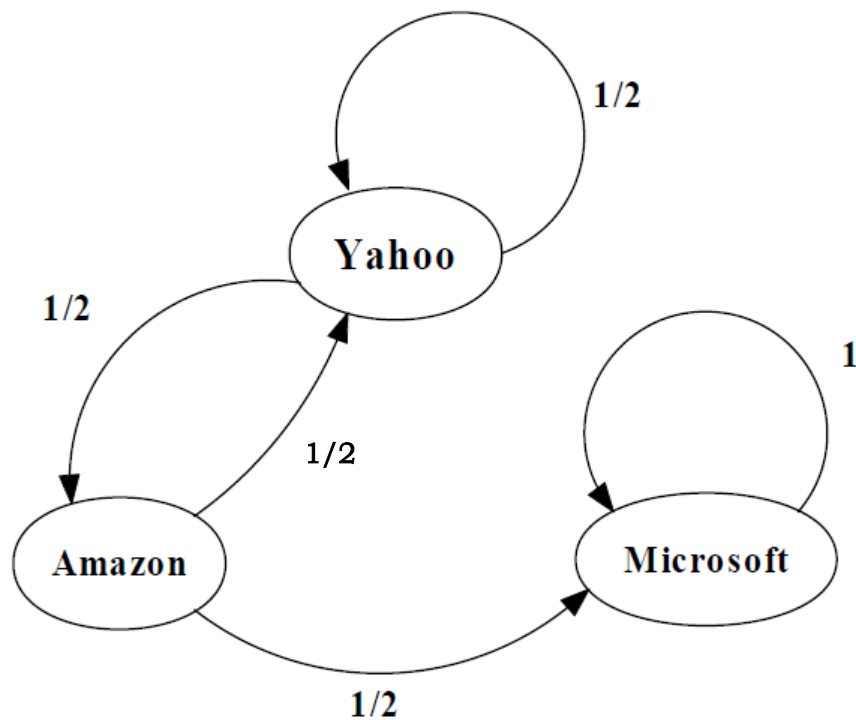
一个例子



$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 1/3 \\ 1/6 \\ 1/2 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

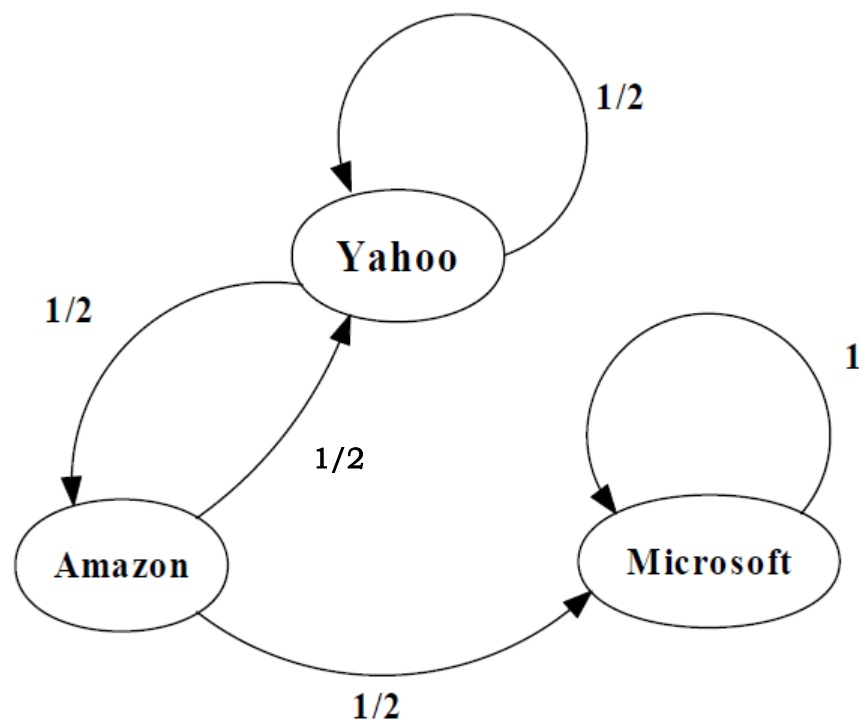
一个例子



$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 1/4 \\ 1/6 \\ 7/12 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/6 \\ 1/2 \end{bmatrix}$$

一个例子



$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 5/24 \\ 1/8 \\ 2/3 \end{bmatrix} \begin{bmatrix} 1/6 \\ 5/48 \\ 35/48 \end{bmatrix} \dots \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \leftarrow$$

基于概率的PageRank

- **基本思想：** 假设有一个网上的随机冲浪者在网上随机冲浪，那么对每个网页存在一个被其访问的概率，并定义该概率值为网页的PageRank
- 一般而言，网页的概率值越大，网页越重要
- 随机冲浪的两种形式
 - **随机游走：** 从当前网页的链出网页中随机选出一个网页作为下一步的访问目标
 - **随机跳转：** 从当前网页跳到Web中的其它任一网页

改进的PageRank公式

□ **随机冲浪模型：** 网页的访问概率

$$R(u) = \frac{\alpha}{N} + (1 - \alpha) \sum_{v \in B_u} \frac{R(v)}{N_v}$$

- 这里前面部分是由随机跳转形成的访问概率，后面部分则是由随机游走形成的访问概率
- 可以证明，PageRank是收敛的。计算时，PageRank很难通过解析方式求解，通常通过迭代方式求解。 α 通常取0.15

基于马尔科夫链的PageRank计算

- **马尔科夫链**: 是一个离散时间随机过程，这个过程中的每一步都需要做一个随机选择，包括 N 个状态
- **通过一个 $N \times N$ 的转移概率矩阵 P 刻画**，其中每个元素的值位于 $[0,1]$ 区间，每一行的元素之和为1
- **Web图上的一个随机冲浪过程可以看成是马尔科夫链**，其中马尔科夫链中的每个状态对应一个网页，而每个转移概率代表从一个网页转移到另一个网页的概率

基于马尔科夫链的PageRank计算

□ 转移概率矩阵的构造

- 令网页间的连接矩阵 $L=\{l_{ij}\}$, P_i 有链接指向 P_j 时, $l_{ij}=1$, 否则 $l_{ij}=0$ 。对 L 的每行进行归一化, 即用 P_i 的出度 N_i 去除, 得到矩阵 $A=\{a_{ij}\}$, $a_{ij}=l_{ij}/N_i$
- 如果 A 的某一行的所有元素全为0, 则用 $1/N$ 代替每个元素
- 将上述矩阵乘以 $1-\alpha$, 并对每个元素加上 α/N

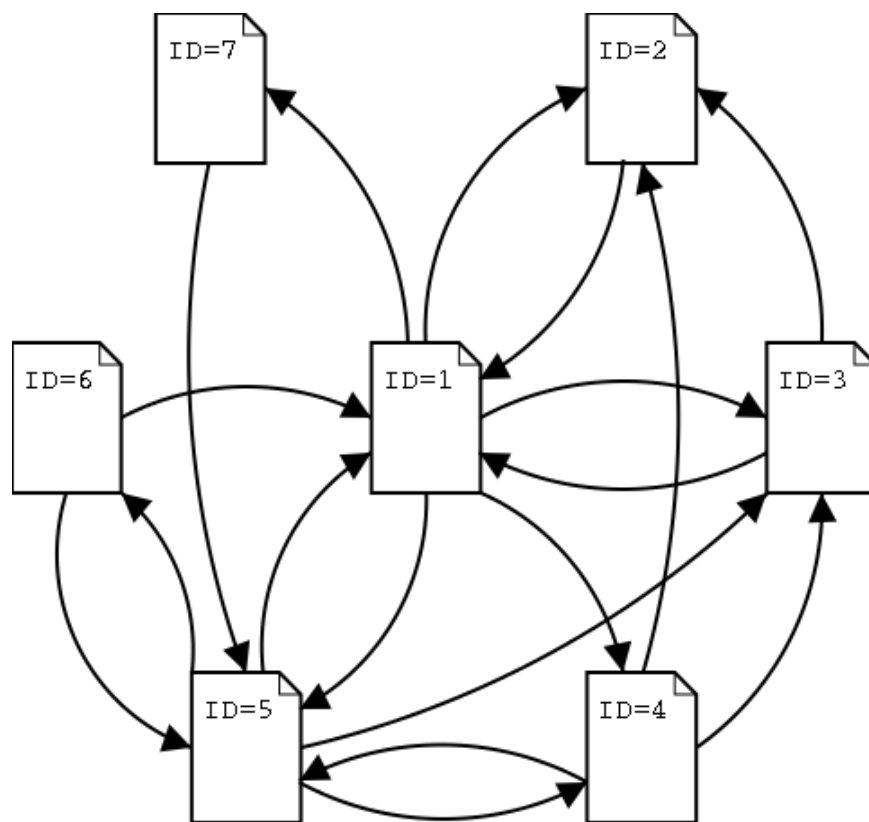
- 上述转移概率的计算对于包含出链的节点也考虑到具有随机跳转的可能: 即认为冲浪者将以 α 的概率选择随机跳转, 以 $1-\alpha$ 的概率选择随机游走

基于马尔科夫链的PageRank计算

$$\begin{bmatrix} R(A) \\ R(B) \\ R(C) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0.5 & 0 & 0 \\ 0.5 & 1 & 0 \end{bmatrix} \begin{bmatrix} R(A) \\ R(B) \\ R(C) \end{bmatrix} \rightarrow \begin{bmatrix} R(A) \\ R(B) \\ R(C) \end{bmatrix} = \begin{bmatrix} 0.033 & 0.033 & 0.933 \\ 0.483 & 0.033 & 0.033 \\ 0.483 & 0.933 & 0.033 \end{bmatrix} \begin{bmatrix} R(A) \\ R(B) \\ R(C) \end{bmatrix}$$

- 令 $R = cA^T R \iff c^{-1}R = A^T R$ ，则根据马尔科夫链的相关定理，特征向量 R 是随机冲浪过程的稳态概率，因此也就是所有Web网页的PageRank值
- 采用幂迭代法求解特征向量，需要设定初始状态分布向量

实例1



Page ID	OutLinks
1	2,3,4,5,7
2	1
3	1,2
4	2,3,5
5	1,3,4,6
6	1,5
7	5

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

计算过程

$$P = \begin{pmatrix} 0 & 1/5 & 1/5 & 1/5 & 1/5 & 0 & 1/5 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/3 & 1/3 & 0 & 1/3 & 0 & 0 \\ 1/4 & 0 & 1/4 & 1/4 & 0 & 1/4 & 0 \\ 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad R = cP^T R, \text{ 令 } c=1, \text{ 解得}$$

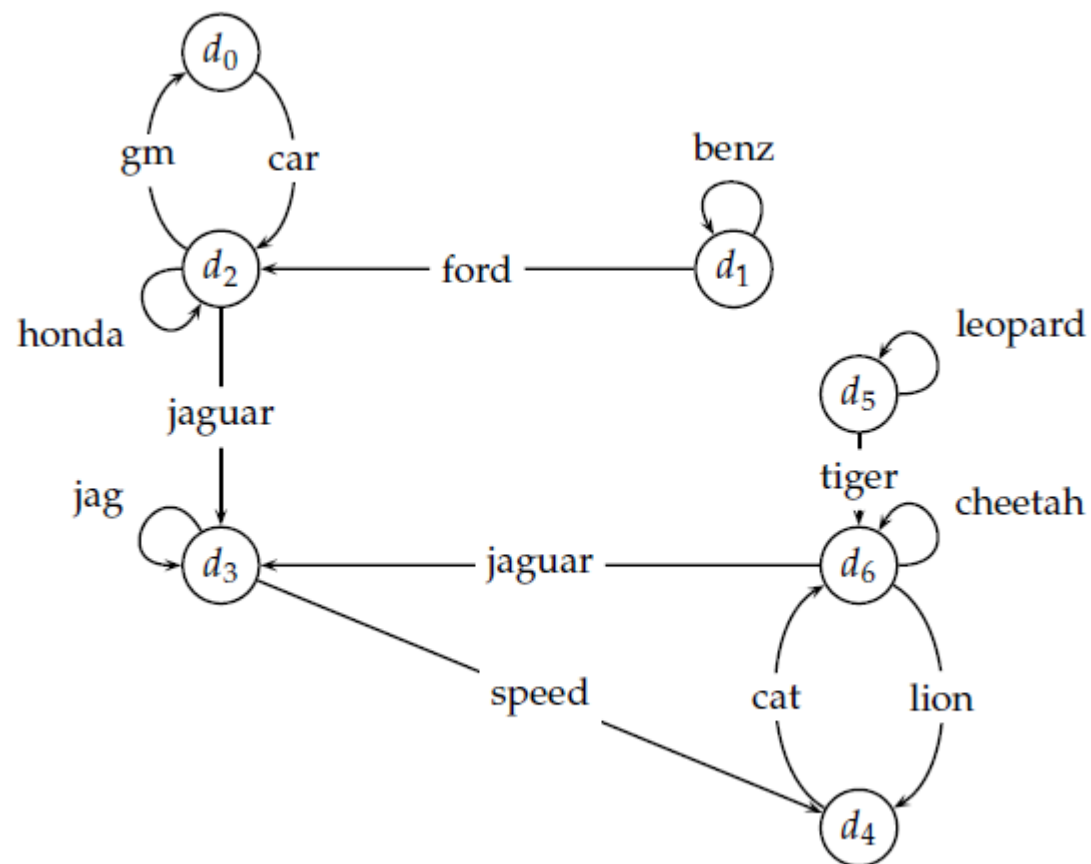
$$R = \begin{pmatrix} 0.69946 \\ 0.38286 \\ 0.32396 \\ 0.24297 \\ 0.41231 \\ 0.10308 \\ 0.13989 \end{pmatrix}$$

Normalized =

$$\begin{pmatrix} 0.303514 \\ 0.166134 \\ 0.140575 \\ 0.105431 \\ 0.178914 \\ 0.044728 \\ 0.060703 \end{pmatrix}$$

实例2

□ 一个部分汽车网页的Web图



计算结果

□ 相应的转移概率矩阵为($\alpha = 0.14$):

0.02	0.02	0.88	0.02	0.02	0.02	0.02
0.02	0.45	0.45	0.02	0.02	0.02	0.02
0.31	0.02	0.31	0.31	0.02	0.02	0.02
0.02	0.02	0.02	0.45	0.45	0.02	0.02
0.02	0.02	0.02	0.02	0.02	0.02	0.88
0.02	0.02	0.02	0.02	0.02	0.45	0.45
0.02	0.02	0.02	0.31	0.31	0.02	0.31

□ 相应的PageRank向量为:

$$\vec{x} = (0.05 \quad 0.04 \quad 0.11 \quad 0.25 \quad 0.21 \quad 0.04 \quad 0.31)$$

PageRank 面对的 Spamming 问题

- **SEO** (Search Engine Optimization): 通过正当或者作弊等手段提高网站的检索排名(包括PageRank排名)
- 因此, 实际中的PageRank实现必须应对这种作弊, 实际实现复杂得多。实际中往往有多个因子(比如内容相似度)的融合

HITS算法 (Hyperlink-Induced Topic Search)

□ 两类重要网页

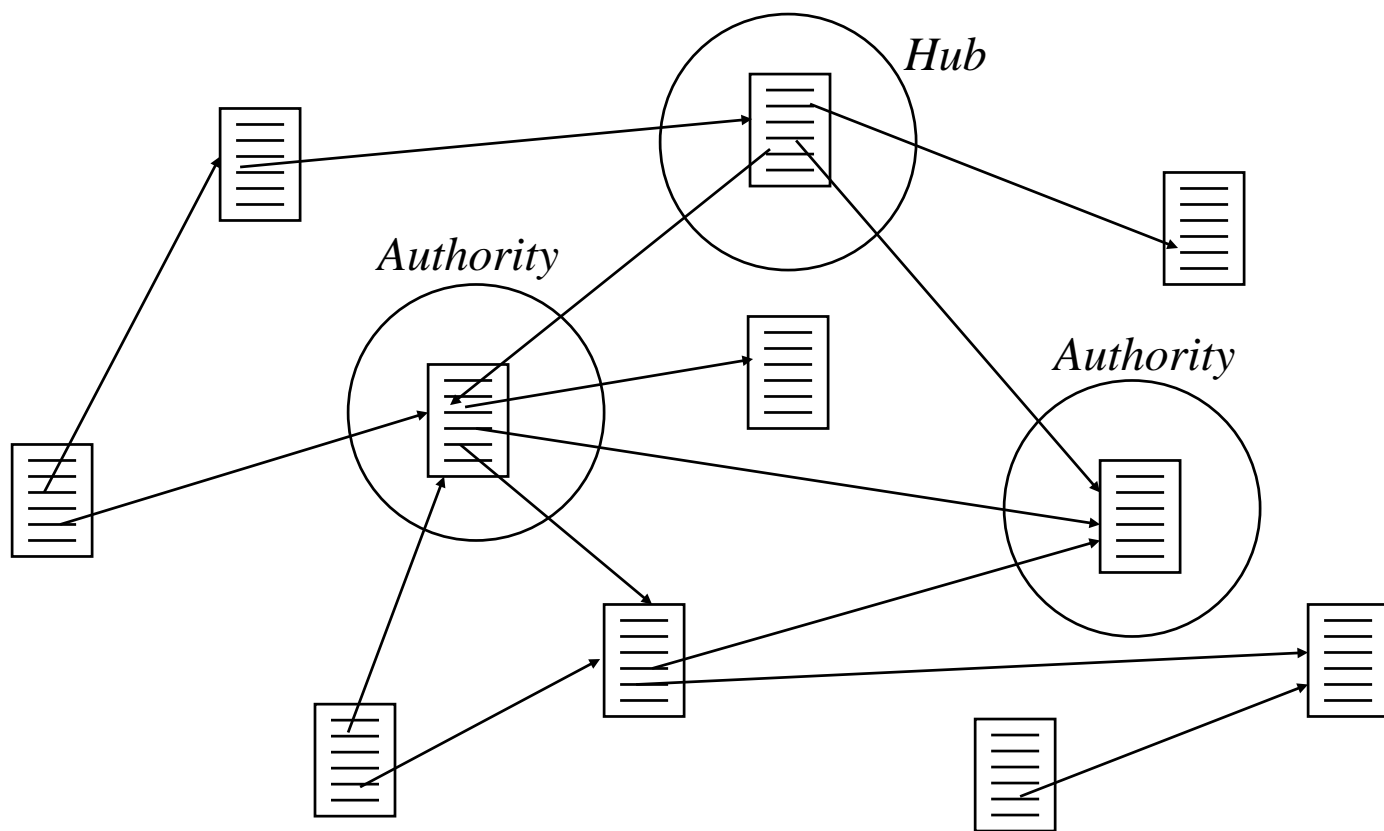
- **权威网页**：提供了重要的、可信任的有用信息的网页，一般由权威机构或权威人士发布
- **导航网页**：提供许多主题相关的权威网页链接的网页，一般由人工编辑

□ **算法目标**：针对一个查询主题，定量确定出相关的导航网页和权威网页

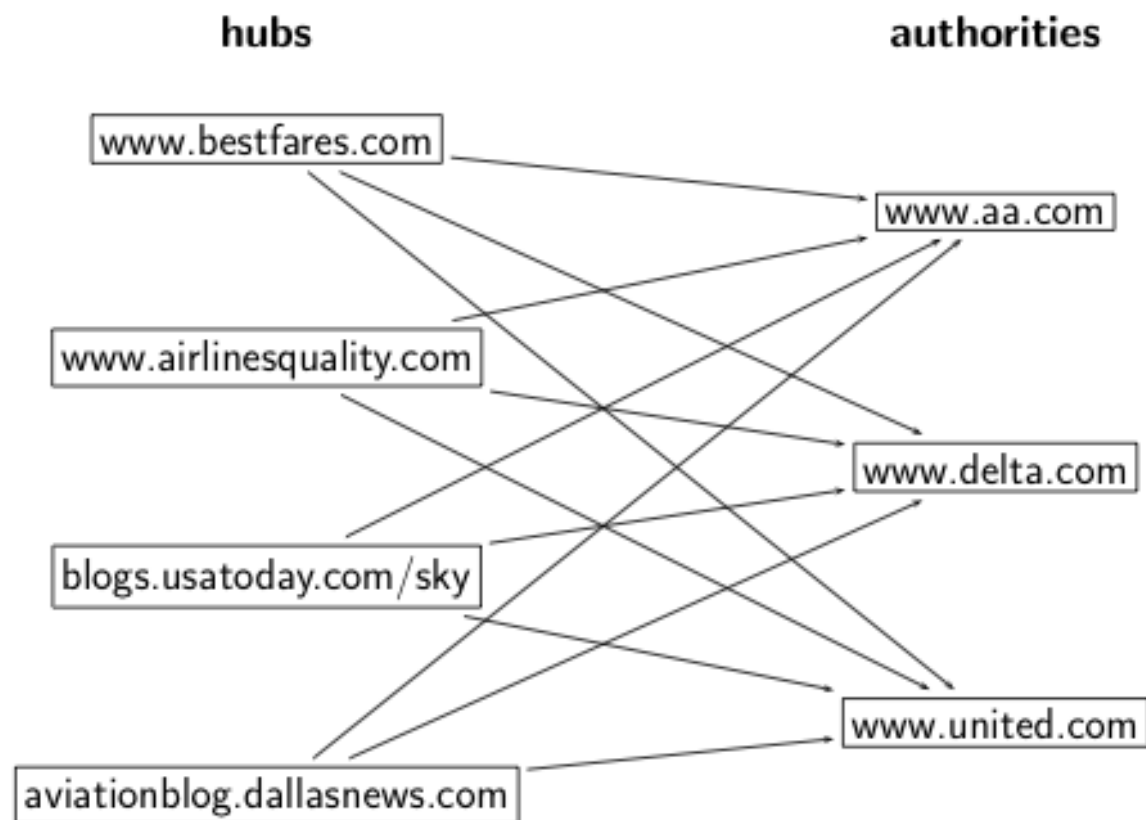
□ **基本思路**：对每个网页计算两个值Hub和 Authority

导航网页与权威网页的关系

- 导航网页指向许多权威网页
- 权威网页被许多导航型网页所链接



实例



查询[Chicago Bulls]的权威网页

0.85	www.nba.com/bulls
0.25	www.essex1.com/people/jmiller/bulls.htm "da Bulls"
0.20	www.nando.net/SportServer/basketball/nba/chi.html "The Chicago Bulls"
0.15	Users.aol.com/rynecub/bulls.htm "The Chicago Bulls Home Page "
0.13	www.geocities.com/Colosseum/6095 "Chicago Bulls"

(Ben Shaul et al, WWW8)

[Chicago Bulls] 的权威网页



查询[Chicago Bulls]的导航网页

1.62	www.geocities.com/Colosseum/1778 "Unbelieveabulls!!!!!"
------	--

1.24	www.webring.org/cgi-bin/webring?ring=chbulls "Chicago Bulls"
------	---


0.74	www.geocities.com/Hollywood/Lot/3330/Bulls.html "Chicago Bulls"
------	---

0.52	www.nobull.net/web_position/kw-search-15-M2.html "Excite Search Results: bulls "
------	---

0.52	www.halcyon.com/wordltd/bball/bulls.html "Chicago Bulls Links"
------	---

(Ben Shaul et al, WWW8)

[Chicago Bulls]导航型网页的例子

**COAST TO COAST TICKETS**
great tickets from nice people

Returning Customer

City Guide | 1

[Minnesota Timberwolves Tickets](#)
[New Jersey Nets Tickets](#)
[New Orleans Hornets Tickets](#)
[New York Knicks Tickets](#)
[Oklahoma City Thunder Tickets](#)
[Orlando Magic Tickets](#)
[Philadelphia 76ers Tickets](#)
[Phoenix Suns Tickets](#)
[Portland Trail Blazers Tickets](#)
[Sacramento Kings Tickets](#)
[San Antonio Spurs Tickets](#)
[Toronto Raptors Tickets](#)
[Utah Jazz Tickets](#)
[Washington Wizards Tickets](#)
[NBA All-Star Weekend](#)
[NBA Finals Tickets](#)
[NBA Playoffs Tickets](#)
[All NBA Tickets](#)

Official Website Links:
[Chicago Bulls \(official site\)](#)
<http://www.nba.com/bulls/>

Fan Club - Fan Site Links:
[Chicago Bulls](#)
Chicago Bulls Fan Site with Bulls Blog, News, Bulls Forum, Wallpapers and all your basic Chicago Bulls essentials!!
<http://www.bullscentral.com>
[Chicago Bulls Blog](#)
The place to be for news and views on the Chicago Bulls and NBA Basketball!
<http://chi-bulls.blogspot.com>

News and Information Links:
[Chicago Sun-Times \(local newspaper\)](#)
<http://www.suntimes.com/sports/basketball/bulls/index.html>
[Chicago Tribune \(local newspaper\)](#)
<http://www.chicagotribune.com/sports/basketball/bulls/>
[Wikipedia - Chicago Bulls](#)
All about the Chicago Bulls from Wikipedia, the free online encyclopedia.
http://en.wikipedia.org/wiki/Chicago_Bulls

Merchandise Links:
[Chicago Bulls watches](#)
http://www.sportswatches.com/NBA_watches/Chicago-Bulls-watches.html

Event Selections
Sporting Events
[MLB Baseball Tickets](#)
[NFL Football Tickets](#)
[NBA Basketball Tickets](#)
[NHL Hockey Tickets](#)
[NASCAR Racing Tickets](#)
[PGA Golf Tickets](#)
[Tennis Tickets](#)
[NCAA Football Tickets](#)

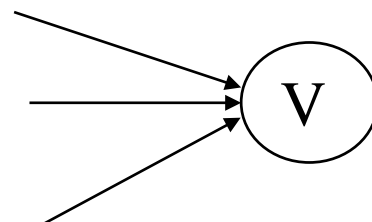
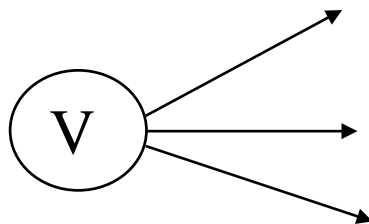
计算方法

□ 基于hub和authority之间的相互关系，可以定义

$$\begin{aligned} h(v) &\leftarrow \sum_{v \mapsto y} a(y) \\ a(v) &\leftarrow \sum_{y \mapsto v} h(y). \end{aligned}$$



$$\begin{aligned} h(v) &= \sum a(y_i) \\ a(v) &= \sum h(y_i) \end{aligned}$$



HITS算法的收敛性

- 令A表示所处理Web子集的邻接矩阵， \vec{h} 和 \vec{a} 分别表示所有网页的hub值和authority值向量，则根据上面的公式可得：

$$\begin{array}{lcl} \vec{h} \leftarrow A\vec{a} & \longrightarrow & \vec{h} \leftarrow AA^T\vec{h} \\ \vec{a} \leftarrow A^T\vec{h}, & \longrightarrow & \vec{a} \leftarrow A^TA\vec{a}. \end{array} \quad \longrightarrow \quad \begin{array}{lcl} \vec{h} & = & (1/\lambda_h)AA^T\vec{h} \\ \vec{a} & = & (1/\lambda_a)A^TA\vec{a}. \end{array}$$

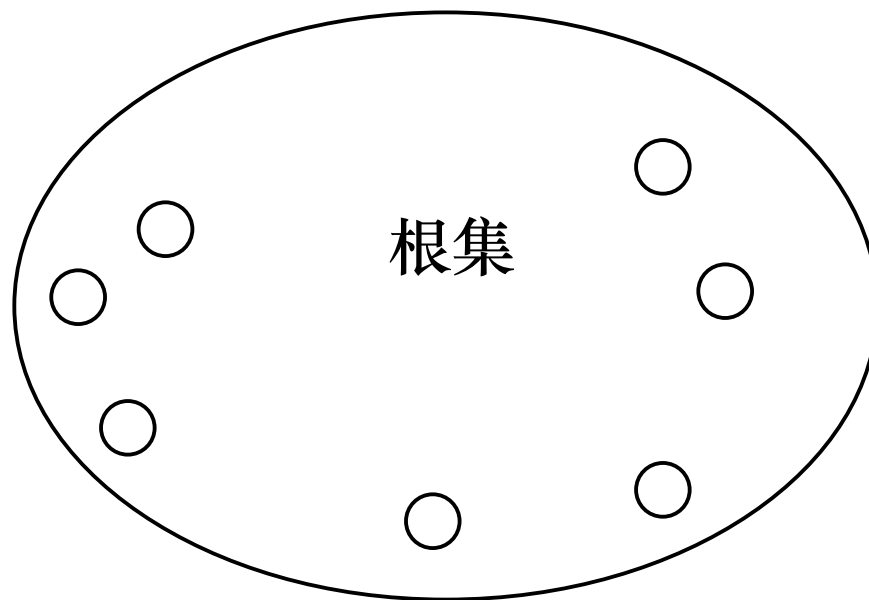
- 根据矩阵理论， \vec{h} 和 \vec{a} 最后会收敛于某个稳态向量

HITS算法步骤

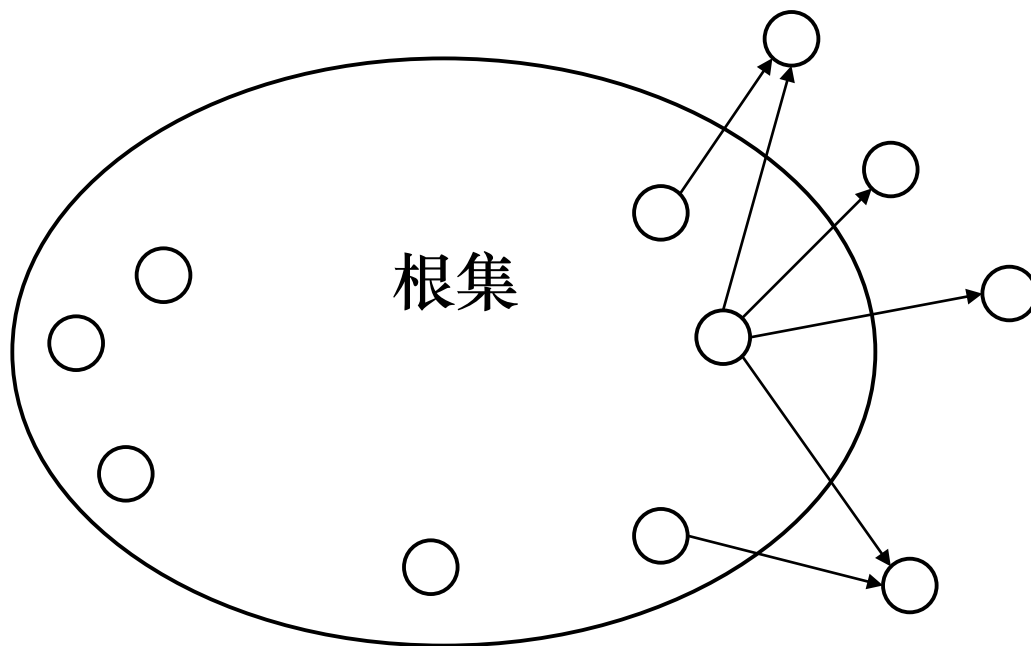
□ 确定Web子集

- 根据给定查询进行搜索，搜索的结果称为根集(root set)
- 将所有链向根集和根集链出的网页加入到根集
- 扩展后的更大的集合称为基本集(base set)，即为Web子集

根集和基本集 (1)

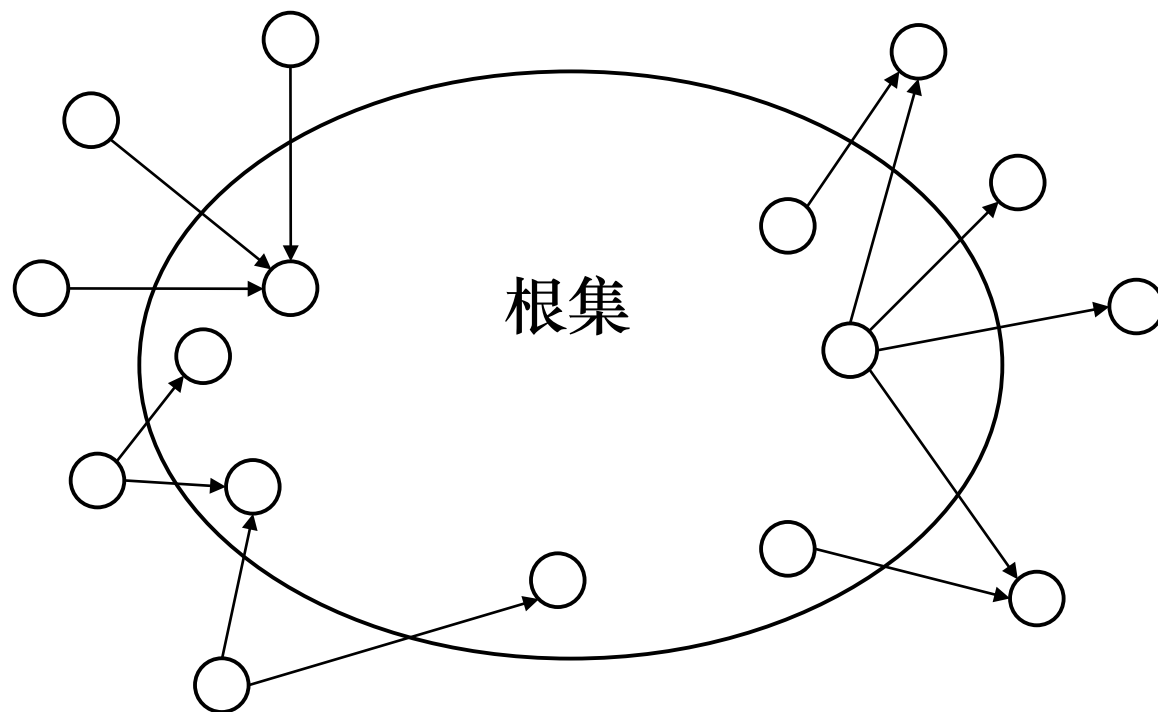


根集和基本集 (2)



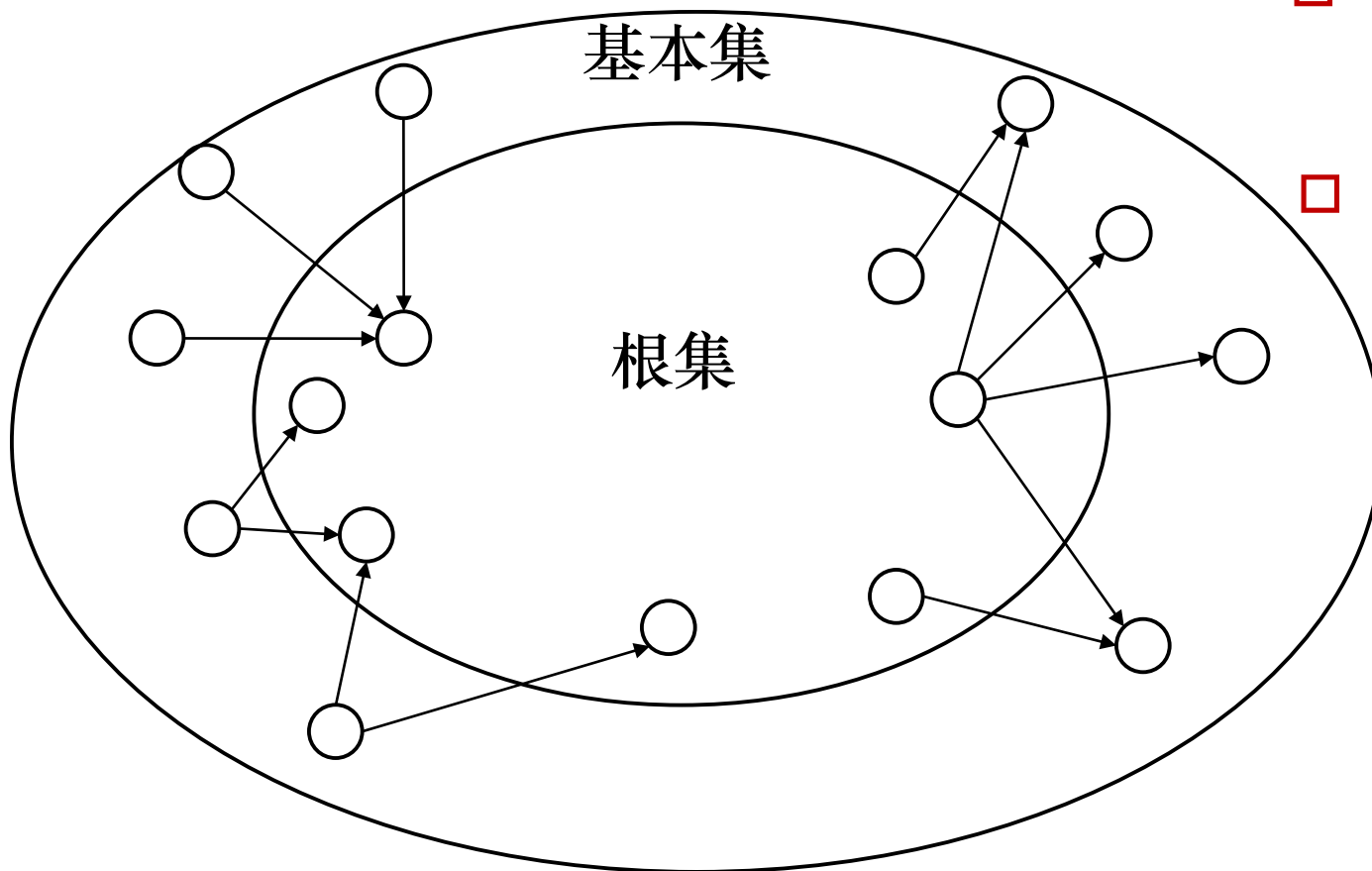
根集中节点链向的网页节点

根集和基本集 (3)



指向根集节点的所有节点

根集和基本集 (4)



- 根集往往包含 200-1000个节点
- 基本集可以达到 5000个节点

HITS算法步骤

□ 迭代计算Web子集中每个网页的h和a

Initialize for all $p \in S$: $a_p = h_p = 1$

For $i = 1$ to k :

For all $p \in S$: $a_p = \sum_{q:q \rightarrow p} h_q$ (*update auth. scores*)

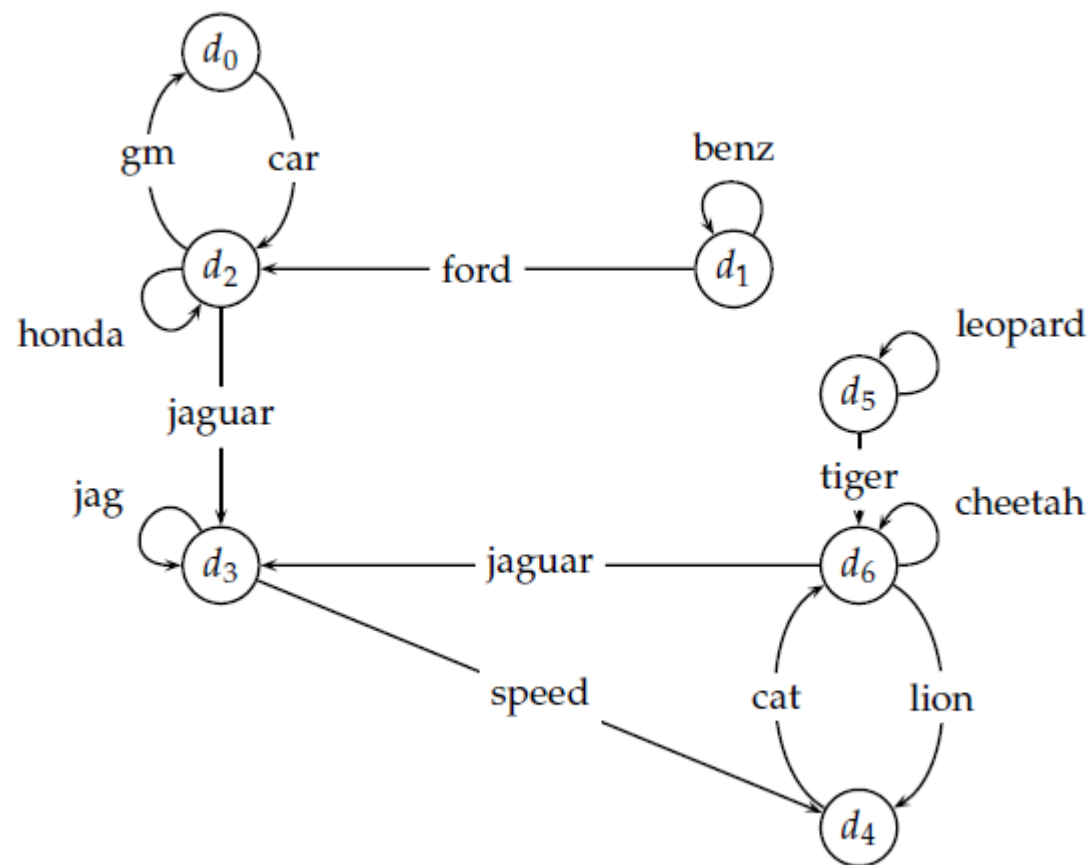
For all $p \in S$: $h_p = \sum_{q:p \rightarrow q} a_q$ (*update hub scores*)

For all $p \in S$: $a_p = a_p / c$ $c: \sum_{p \in S} (a_p / c)^2 = 1$ (*normalize a*)

For all $p \in S$: $h_p = h_p / c$ $c: \sum_{p \in S} (h_p / c)^2 = 1$ (*normalize h*)

实例

□ 一个部分汽车网页的Web图



计算结果

□ Web子集的邻接矩阵及其 \vec{h} 和 \vec{a} 为:

0	0	1	0	0	0	0
0	1	1	0	0	0	0
1	0	1	2	0	0	0
0	0	0	1	1	0	0
0	0	0	0	0	0	1
0	0	0	0	0	1	1
0	0	0	2	1	0	1

$$\vec{h} = (0.03 \quad 0.04 \quad 0.33 \quad 0.18 \quad 0.04 \quad 0.04 \quad 0.35)$$

$$\vec{a} = (0.10 \quad 0.01 \quad 0.12 \quad 0.47 \quad 0.16 \quad 0.01 \quad 0.13)$$

PageRank vs. HITS

- 网页的PageRank与查询主题无关，可以事先算好，因此适合于大型搜索引擎的应用。
- HITS算法的计算与查询主题相关，检索之后再计算，因此，不适合于大型搜索引擎。

参考资料

□ 《信息检索导论》第21章

□ <http://ifnlp.org/ir>

- American Mathematical Society article on PageRank (popular science style)
- Jon Kleinberg' s home page (main person behind HITS)
- A Google bomb and its defusing
- Google' s official description of PageRank: *PageRank reflects our view of the importance of web pages by considering more than 500 million variables and 2 billion terms. Pages that believe are important pages receive a higher PageRank and are more likely to appear at the top of the search results.*

课后作业

□ 见课程网页:

<http://10.76.3.31>