

现代信息检索

Modern Information Retrieval

第18讲 链接分析

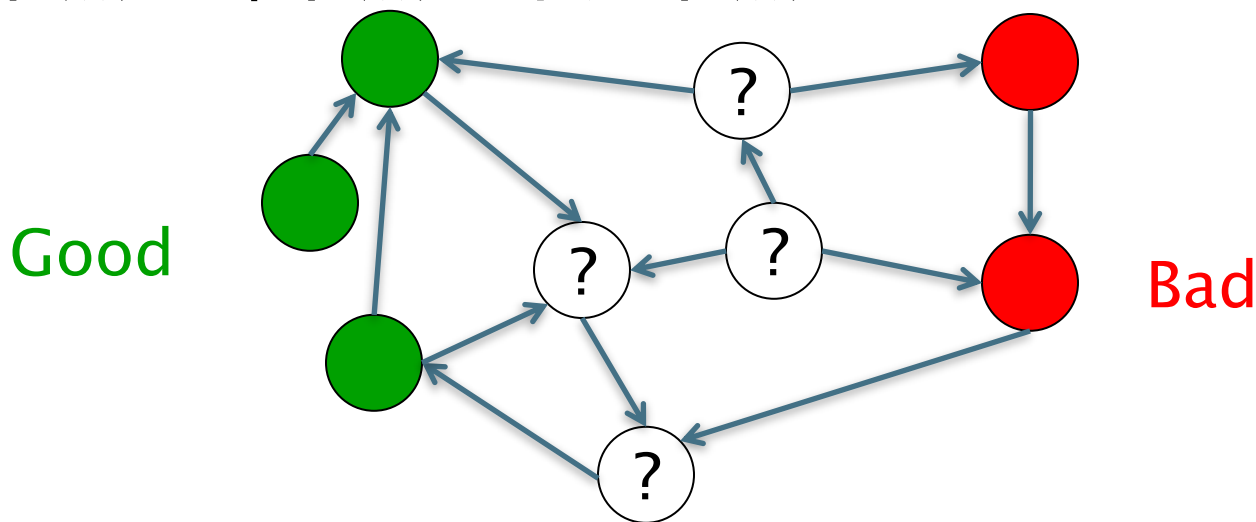
Link Analysis

本讲内容

- 锚文本: Web上的链接相关信息为什么对IR有用?
- 引用分析(Citation analysis): PageRank及其他基于链接排序方法的数学基础
- PageRank : 一个著名的基于链接分析的排序算法(Google)
- HITS : 另一个著名的基于链接分析的排序算法(IBM)

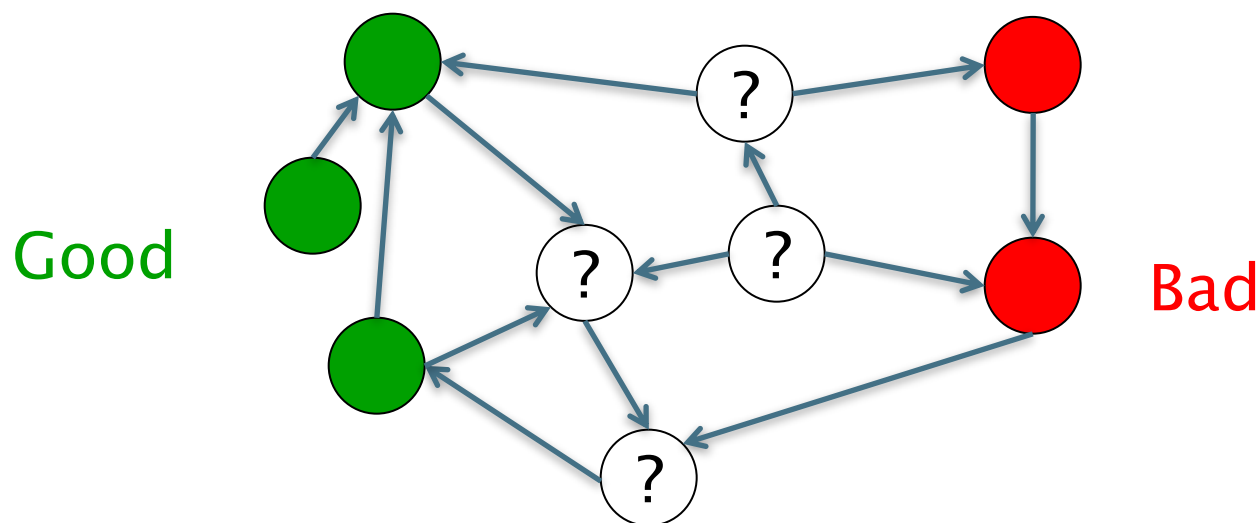
链接无处不在

- 真实性和权威性的有效来源
 - 垃圾邮件 - 哪些电子邮件帐户是垃圾邮件发送者?
 - host质量 - 哪些host质量不好?
 - 电话呼叫记录
- 好节点、坏节点和未知节点



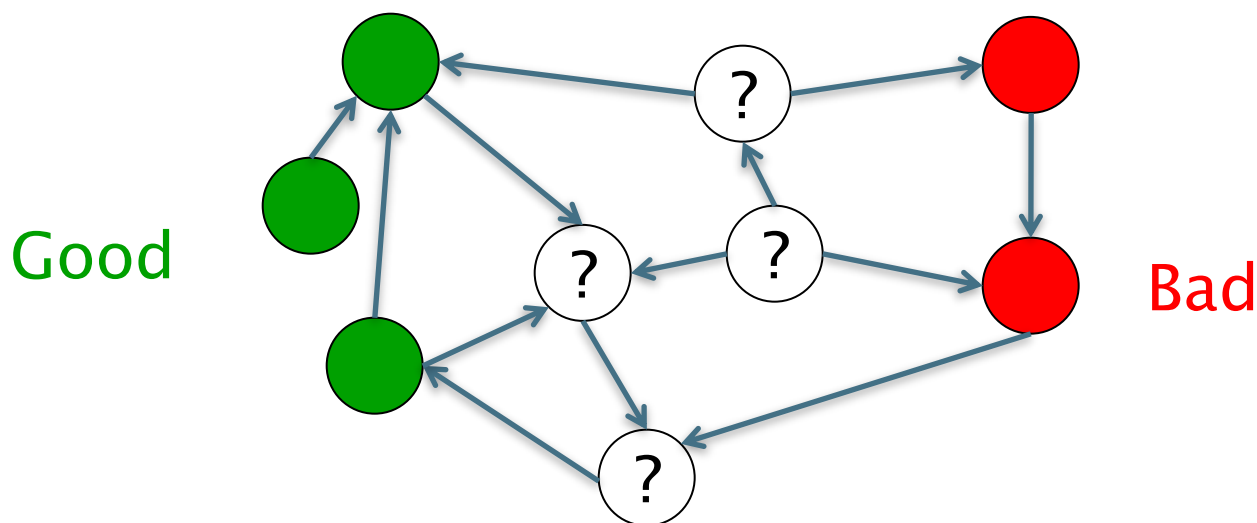
例1：好/坏/未知节点

- 好节点、坏节点和未知节点
 - 好节点不会指向坏节点
 - 所有其他貌似合理的组合



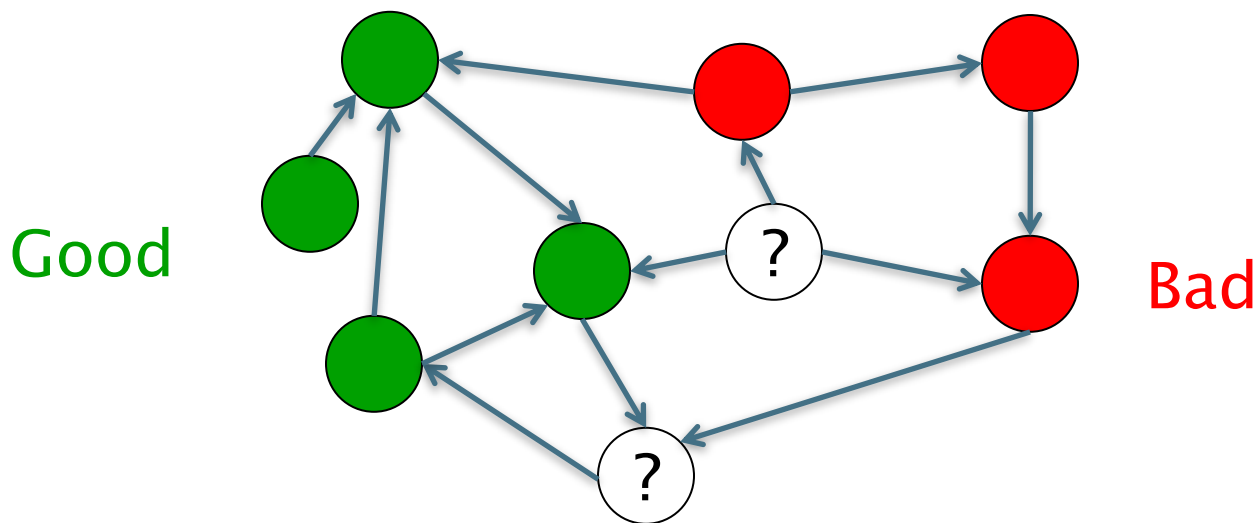
简单的迭代逻辑

- 好节点不会指向坏节点
 - 如果一个节点指向了坏节点，那么这个节点是坏节点
 - 如果一个好节点指向这个节点，那么这个节点是好节点



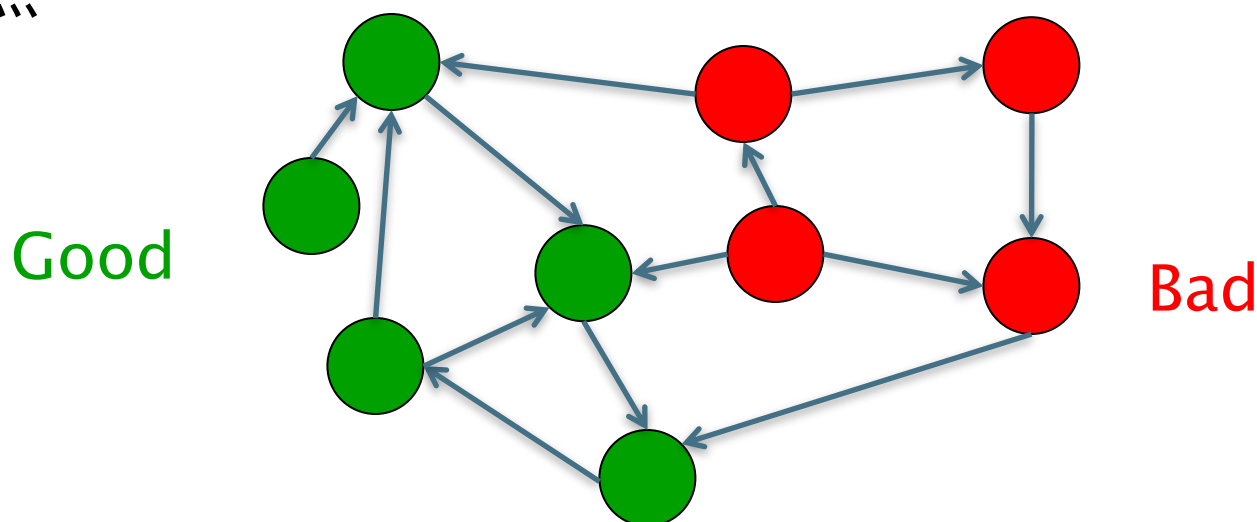
简单的迭代逻辑

- 好节点不会指向坏节点
 - 如果一个节点指向了坏节点，那么这个节点是坏节点
 - 如果一个好节点指向这个节点，那么这个节点是好节点



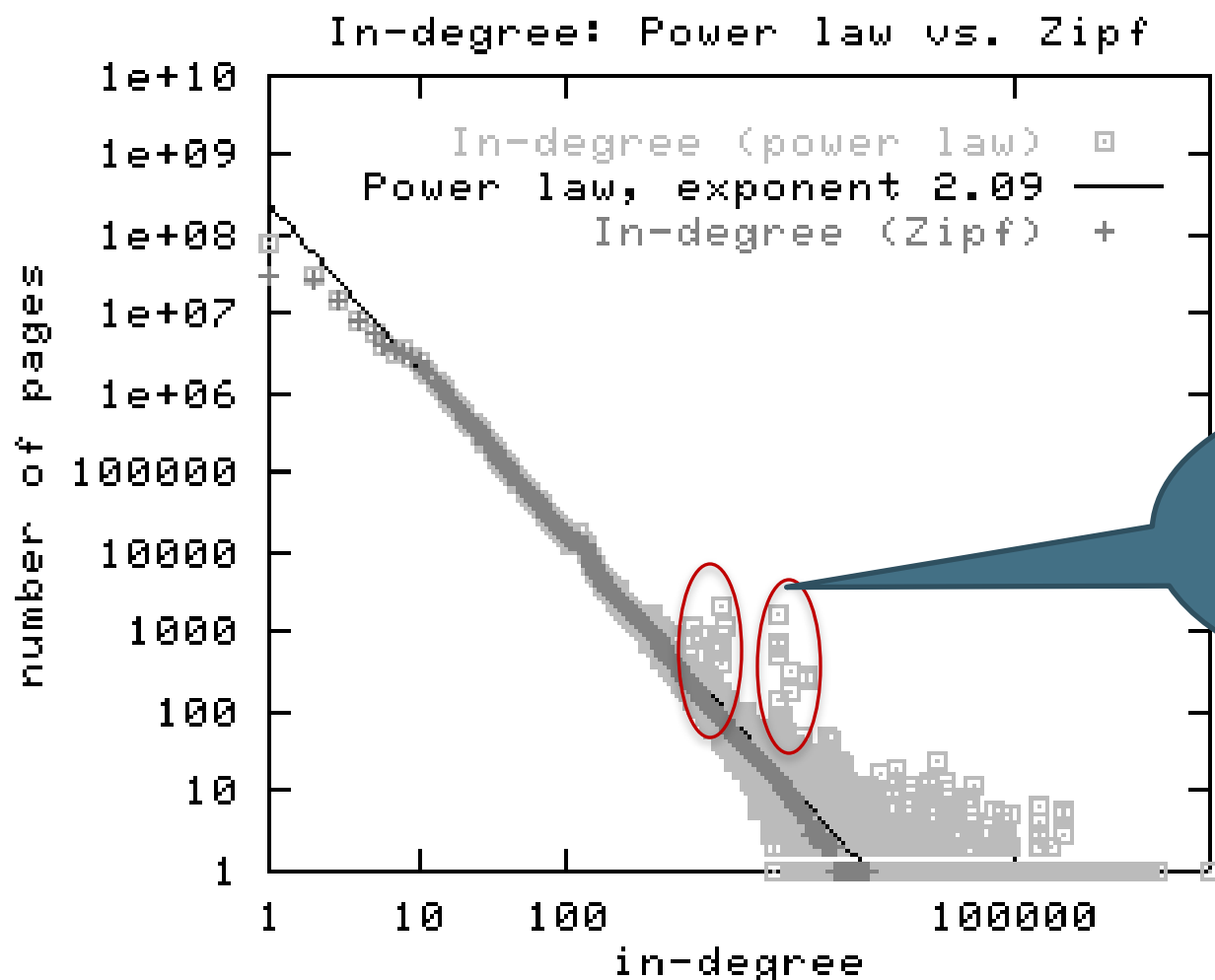
简单的迭代逻辑

- 好节点不会指向坏节点
 - 如果一个节点指向了坏节点，那么这个节点是坏节点
 - 如果一个好节点指向这个节点，那么这个节点是好节点



有时需要概率计算—例如：识别垃圾邮件

例2 在指向页面的连接中——异常模式



垃圾邮件发送者违反了幂律！

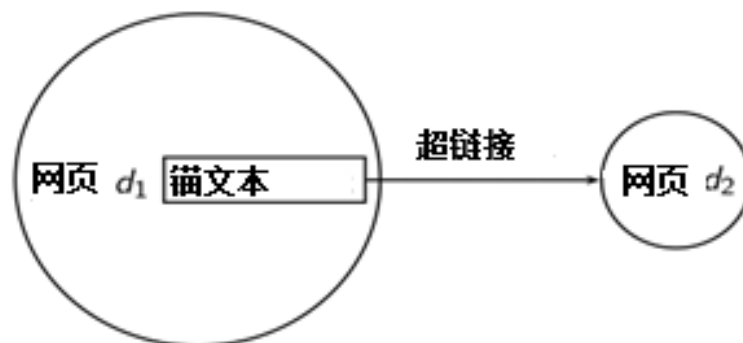
链接分析的其他示例

- 社交网络是群体行为的丰富来源
- 例如：购物者亲和力—Goel+Goldstein 2010
 - 朋友花钱多的消费者，自己也会花不少钱
- <http://www.cs.cornell.edu/home/kleinber/net-works-book/>

为什么我们对链接分析感兴趣？

- 链接分析对目前为止的完全基于文本的IR任务进行了补充
 - （文档）评分和排序
 - 基于链接的聚类—来自链接的主题结构
 - 链接作为分类特征—彼此链接的文档可能是同一主题
- 爬虫
 - 根据已看到的链接，我们下一步要爬取哪里？

Web可以看成一个有向图



- 假设1: 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
- 假设 2: 锚文本描述了文档 d_2 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 here”

$[d_2 \text{ 中 文 本}]$ vs. $[d_2 \text{ 中 文 本}] + [\text{锚文本} \rightarrow d_2]$

- 后者往往效果好于前者
- 例子: 查询 *IBM*
 - 匹配IBM 的版权页
 - 匹配很多作弊网页
 - 匹配IBM的wikipedia页面
 - 可能与IBM 的主页并不匹配! (注: 这里指的是不能匹配关键字)
 - ... 也许 IBM 的主页上大部分都是图
- 而按照 $[\text{锚文本} \rightarrow d_2]$ 来搜索效果会比较好
 - 这种表示下, 出现IBM最多的是其主页 www.ibm.com

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

对锚文本构建索引

- 因此，锚文本往往比网页本身更能揭示网页的内容
- 在计算过程中，锚文本应该被赋予比文档中文本更高的权重

PageRank背后的假设

- 假设1: Web上的链接是网页质量的标志—链出网页的作者认为链向的网页具有很高的质量
- 假设2: 锚文本能够描述链向网页的内容
- 通常情况下假设是成立的，但也有例外

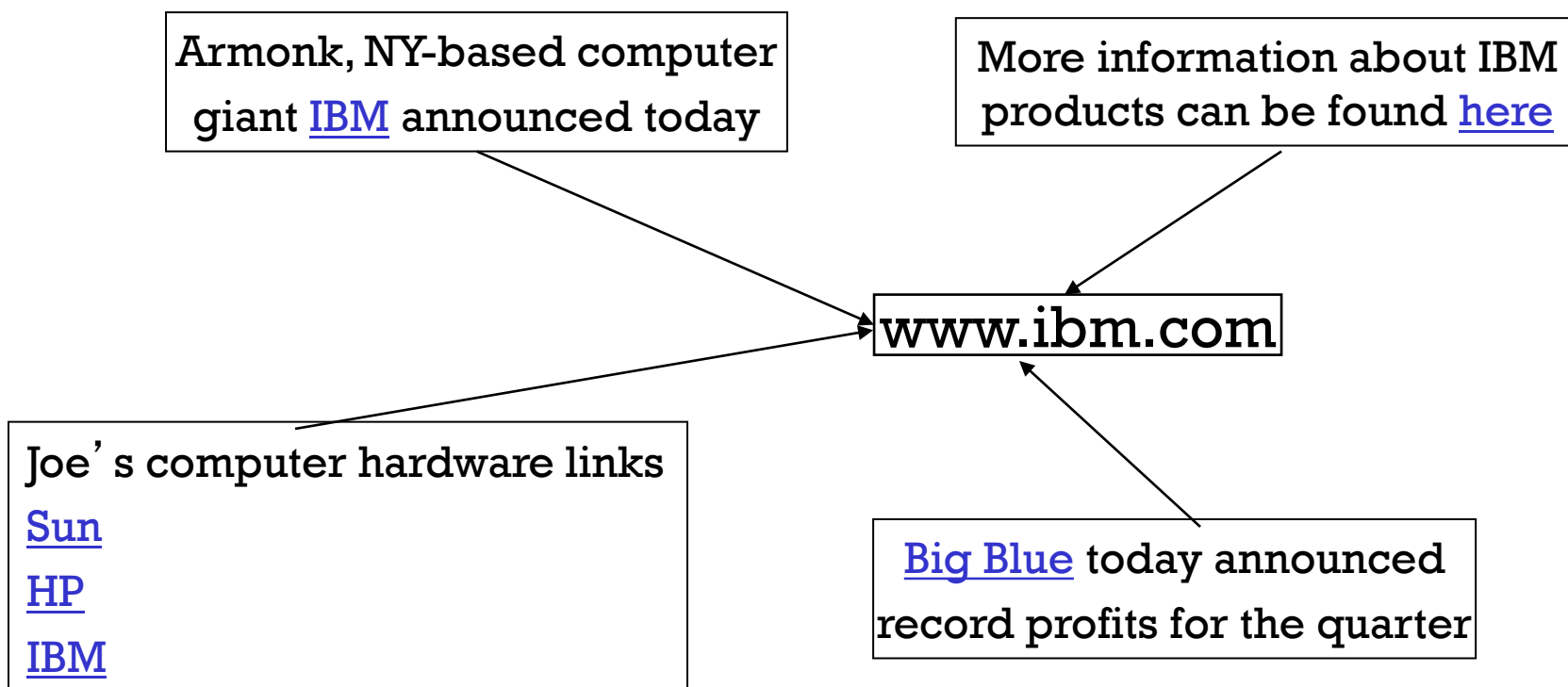
Google炸弹(Google bomb)

- 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
 - 用户群体有意创建链接误导搜索引擎
- 一些知名的Google炸弹
 - [dangerous cult] – 指向 Church of Scientology网站
 - [who is a failure?] – 指向小布什网站
 - [evil empire] – 指向美国政府网站

Google会不断给排序模型打补丁修正新出现的Google炸弹带来的误导性结果

锚文本索引

- 将从指向文档D的链接的锚文本（也可能包含锚文本附近的文本）包含在D的索引中



锚文本索引

- 有时会产生低于期望的效果，例如：垃圾邮件过滤应用全然失败（total failure）
- 可以根据锚页面网站的权威性对锚文本进行加权
 - 例如：如果我们假设来自espn.com或yahoo.com的内容具有权威性，则对来自它们的锚文本给予更多信任
 - 增加外站锚文本的权重（非裙带关系评分）

链接服务器 (Connectivity servers)

低成本地获取所有链接信息

Connectivity Server

- 支持网络图上的快速查询
 - 哪些URL指向给定的URL?
 - 给定URL指向哪些URL?
- 将映射存储在内存中
 - 指向外部链接的URL，指向内部链接的URL
- 应用
 - 链接分析
 - 网络图分析
 - 连接性，爬虫优化
 - 爬虫控制

Boldi and Vigna 2004

- https://www.ics.uci.edu/~djp3/classes/2008_01_01_I_NF141/Materials/p595-boldi.pdf
- Webgraph – 算法集和一个Java实现
- 基本目标 – 维护内存中的节点邻接表
 - 为此，压缩邻接表是关键

邻接表

- 一个节点的邻居集合
- 假设每个URL由一个整数表示
- 例如：对于40亿页的网站，每个节点需要32位…
并且现在肯定 > 40亿页
- 这需要64位来表示每个超链接
- Boldi/Vigna的方法降低到平均3位/链接
 - 进一步的工作达到2位/链接

邻接表压缩

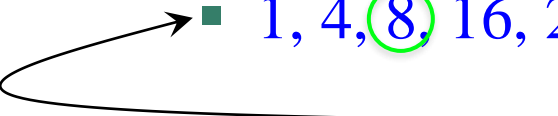
- 压缩中利用到的属性：
 - 相似度（邻接表之间）
 - 位置（一个页面中的许多链接都连接到“附近”的页面）
 - 在已排序的邻接表中使用间隔编码
 - gap value的分布

Boldi/Vigna方法的主要思想

- 考虑所有URL的按字典顺序排列的列表，例如：
 - www.stanford.edu/alchemy
 - www.stanford.edu/biology
 - www.stanford.edu/biology/plant
 - www.stanford.edu/biology/plant/copyright
 - www.stanford.edu/biology/plant/people
 - www.stanford.edu/chemistry

Boldi/Vigna

- 每个URL都有一个邻接表
- 主要思想: 由于模板的缘故, 一个节点的邻接列表类似于字典顺序中的7个先前的URL之一... 否则重新编码
- 根据这 7 个中的之一表示邻接表
- 例如: 考虑以下这些邻接表
 - 1, 2, 4, 8, 16, 32, 64
 - 1, 4, 9, 16, 25, 36, 49, 64
 - 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144
 - 1, 4, 8, 16, 25, 36, 49, 64



Encode as (-2), remove 9, add 8

间隔编码 (gap encodings)

- 给出整数 x, y, z 的已排序列表, 用 $x, y-x, z-y$ 来对 x, y, z 进行表示
- 使用编码来压缩整数
 - γ 编码——获得编码位数 = $1 + 2 \lfloor \lg x \rfloor$
 - δ (Delta) 编码: 对 γ 编码中一元编码的进一步编码
 - 信息论边界: $1 + \lfloor \lg x \rfloor$ 比特
 - ζ (Zeta) 编码: 适用于幂律中的整数 [Boldi, Vigna: Data Compression Conf. 2004]

BV算法的主要优势

- 仅依赖于位置的规范顺序
 - 字典顺序对web十分适用
- 邻接查询可以被非常高效地回答
 - 要获取外部邻居，需要回溯到链的原型
 - 在实践中，这条链通常很短（因为相似性主要基于host内部）
 - 编码过程中也可以明确限制链的长度
- 易于实现one-pass算法
 - 顺序读取，不需要无限缓冲。读取复杂度与网页数量是线性关系

提纲

- ① 上一讲回顾
- ② 锚文本
- ③ 引用分析
- ④ PageRank
- ⑤ HITS: Hub节点&Authority节点

PageRank的起源：引用分析(1)

- 引用分析：科技文献中的引用分析
- 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
- 在科技文献领域使用这些“超链接”的一个应用：
 - 根据他人引用的重合率来度量两篇文献的相似度，这称为共引相似度
 - 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(2)

- 另一个应用: 引用频率可以用度量一篇文档的影响度
 - 最简单的度量指标: 每篇文档都看成一个投票单位, 引用可以看成是投票, 然后计算一篇文档被投票的票数。当然这种方法不太精确。
- 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引用因子
 - 会不会出现循环计算? 答案是否定的, 实际上可以采用良好的形式化定义

PageRank的起源: 引用分析(3)

- 更好的度量方法: 加权的引用频率
- 这就是PageRank的基本思路
- PageRank 最早起源于1960年代Pinsker和Narin提出的引用分析
- 引用分析不是小事情, 在美国, 任何教职人员的薪水取决于其发表文章的影响力!

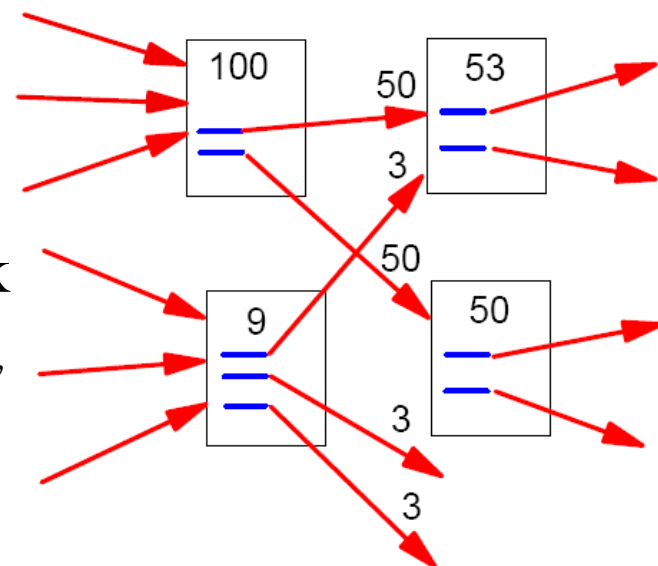
提纲

- ① 上一讲回顾
- ② 锚文本
- ③ 引用分析
- ④ PageRank
- ⑤ HITS: Hub节点&Authority节点

原始的PageRank公式

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

$R(u)$ 和 $R(v)$ 是分别是网页 u 、 v 的PageRank值， B_u 指的是指向网页 u 的网页集合、 N_v 是网页 v 的出链数目。

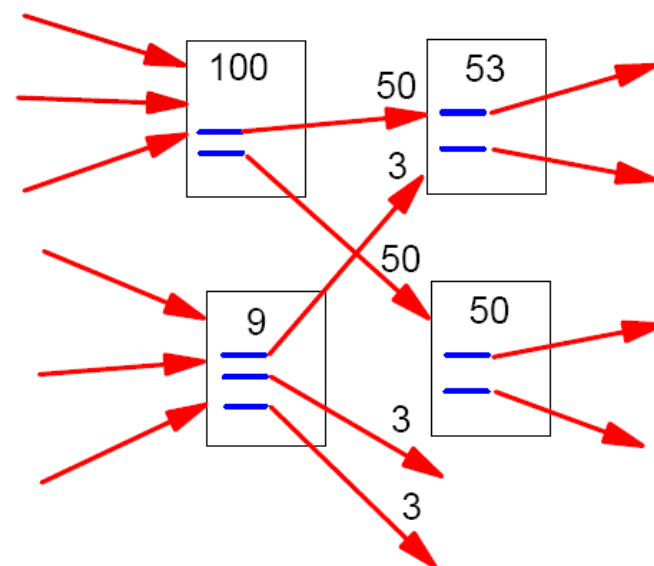


一个网页的PageRank等于所有的指向它的网页的PageRank的分量之和(c 为归一化参数)。网页的每条出链上每个分量上承载了相同的PageRank分量。

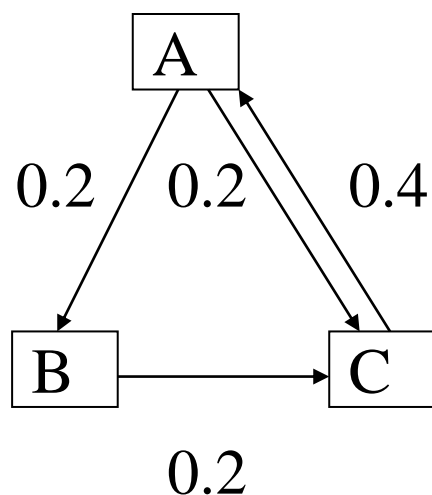
PageRank的特点

- (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)。

类比: (1) 打电话; (2) 微博粉丝



简单计算的例子($c=1$)



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

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

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

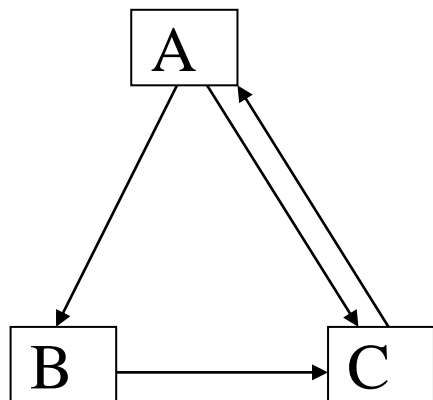
$$R(A)+R(B)+R(C)=1$$

解上述方程得：

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

$$R(B)=0.2$$

简单计算的例子($c=1$): 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

转化成矩阵形式

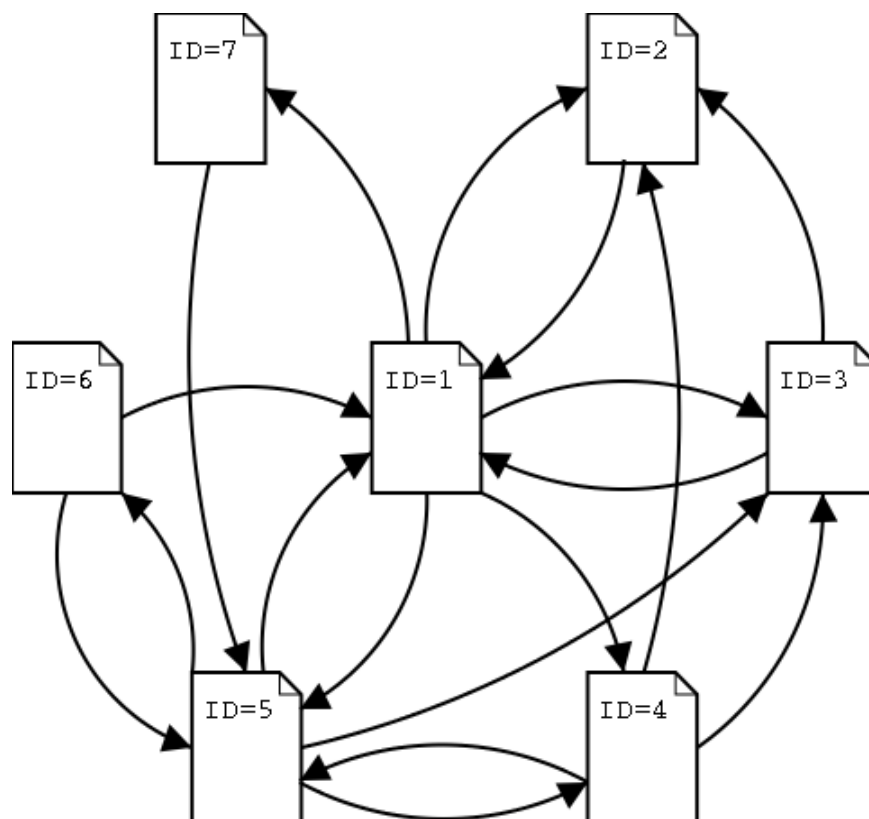
- 令 \mathbf{R} 表示所有 N 个网页的 PageRank 组成的列向量，令网页间的连接矩阵 $\mathbf{L} = \{l_{ij}\}$ ， P_i 有链接指向 P_j 时， $l_{ij} = 1$ ，否则 $l_{ij} = 0$ 。对 \mathbf{L} 的每行进行归一化，即用 P_i 的出度 N_i 去除得到矩阵 $\mathbf{A} = \{a_{ij}\}$ ， $a_{ij} = l_{ij}/N_i$ ，则有 (\mathbf{A}^T 表示 \mathbf{A} 的转置矩阵)：

$$\mathbf{R} = c\mathbf{A}^T\mathbf{R} \iff c^{-1}\mathbf{R} = \mathbf{A}^T\mathbf{R}$$

根据线性代数中有关特征向量和特征值的理论， \mathbf{R} 是矩阵 \mathbf{A}^T 的 c^{-1} 特征值对应的特征向量

$$\begin{aligned} R(A) &= R(C) \\ R(B) &= 0.5R(A) \\ R(C) &= R(B) + 0.5R(A) \end{aligned} \quad \Rightarrow \quad \begin{bmatrix} R(A) \\ R(B) \\ R(C) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0.5 & 0 & 0 \\ 0.5 & 1 & 0 \end{bmatrix} \begin{bmatrix} R(A) \\ R(B) \\ R(C) \end{bmatrix}$$

一个稍微复杂的例子



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

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

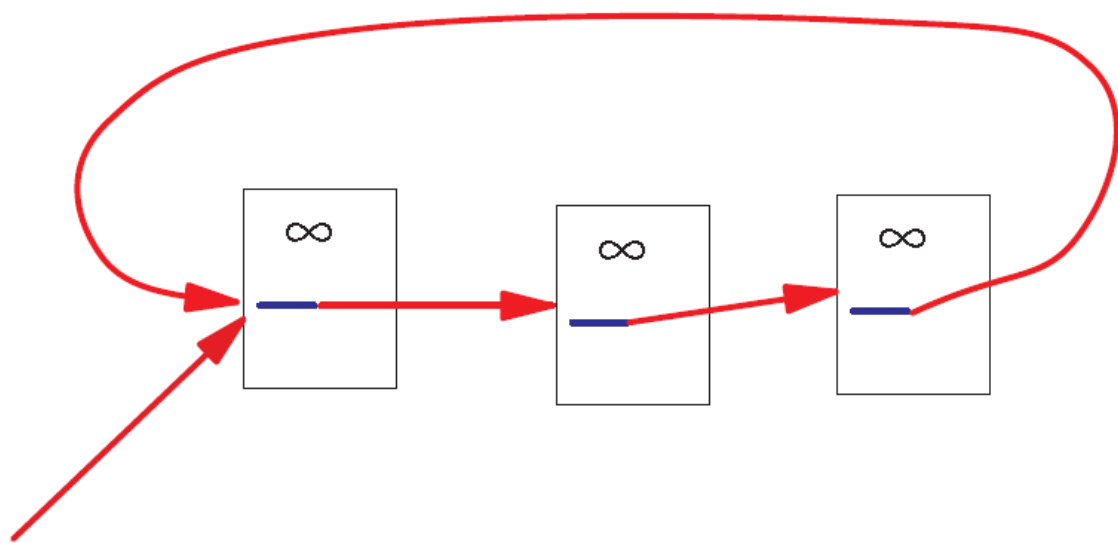
计算过程

$$\text{则归一化后 } A = \begin{pmatrix} 0 & 1/5 & 1/5 & 1/5 & 1/5 & 0 & 1/5 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/3 & 1/3 & 0 & 1/3 & 0 & 0 \\ 1/4 & 0 & 1/4 & 1/4 & 0 & 1/4 & 0 \\ 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} = cA^T R, \text{ 令 } c=1, \text{ 解得}$$

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

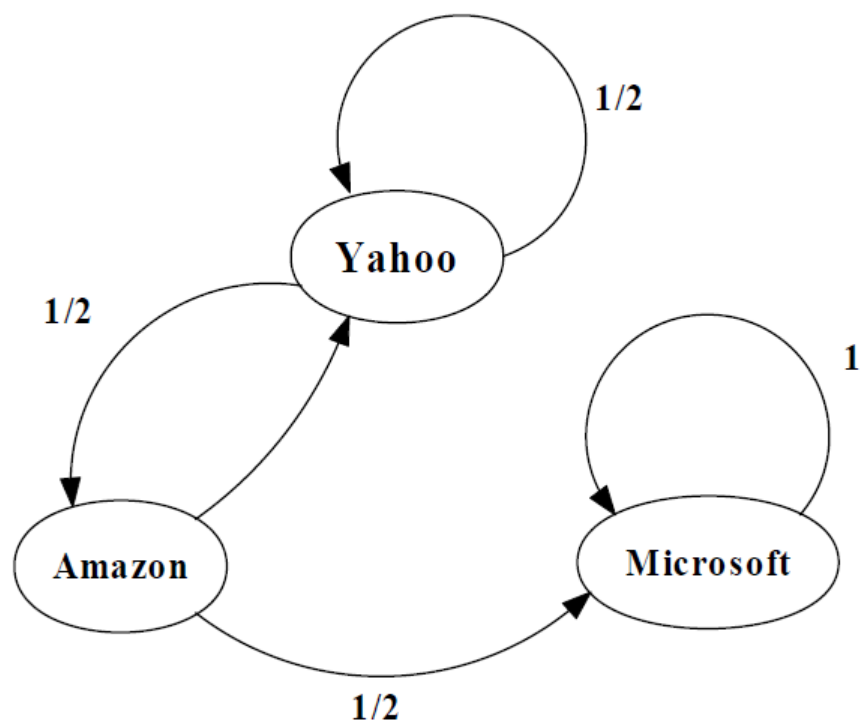
$$\text{Normalized} = \begin{pmatrix} 0.303514 \\ 0.166134 \\ 0.140575 \\ 0.105431 \\ 0.178914 \\ 0.044728 \\ 0.060703 \end{pmatrix}$$

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！

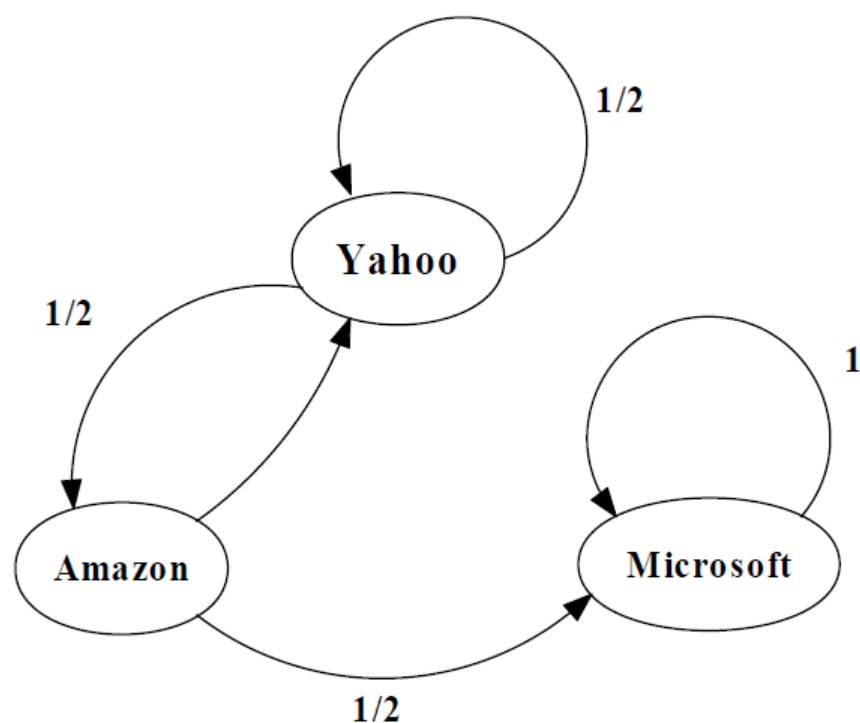
一个例子



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

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

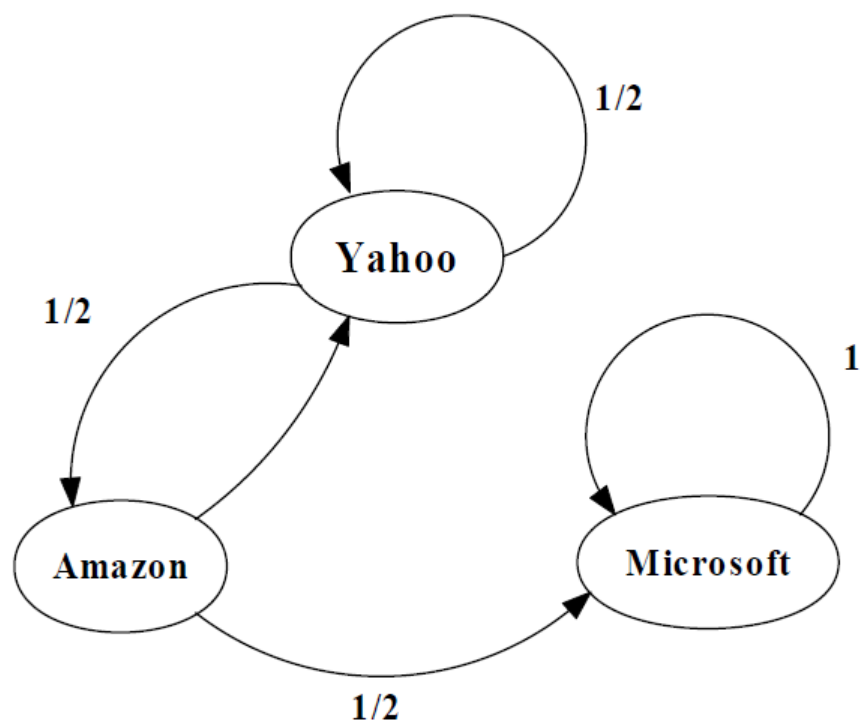
一个例子



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

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

一个例子



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

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

改进的PageRank公式

随机冲浪或随机游走(Random Walk)模型：到达 u 的概率由两部分组成，一部分是直接随机选中的概率 $(1-d)$ 或 $(1-d)/N$ ，另一部分是从指向它的网页顺着链接浏览的概率，则有

$$R(u) = (1-d) + d \sum_{v \in B_u} \frac{R(v)}{N_v} \quad \text{或} \quad R(u) = \frac{(1-d)}{N} + d \sum_{v \in B_u} \frac{R(v)}{N_v}$$

上述两个公式中，后一个公式所有网页PageRank的和为1，前一个公式的PageRank和为 $N(1-d)+d$ 。

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。 d 通常取0.85

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。

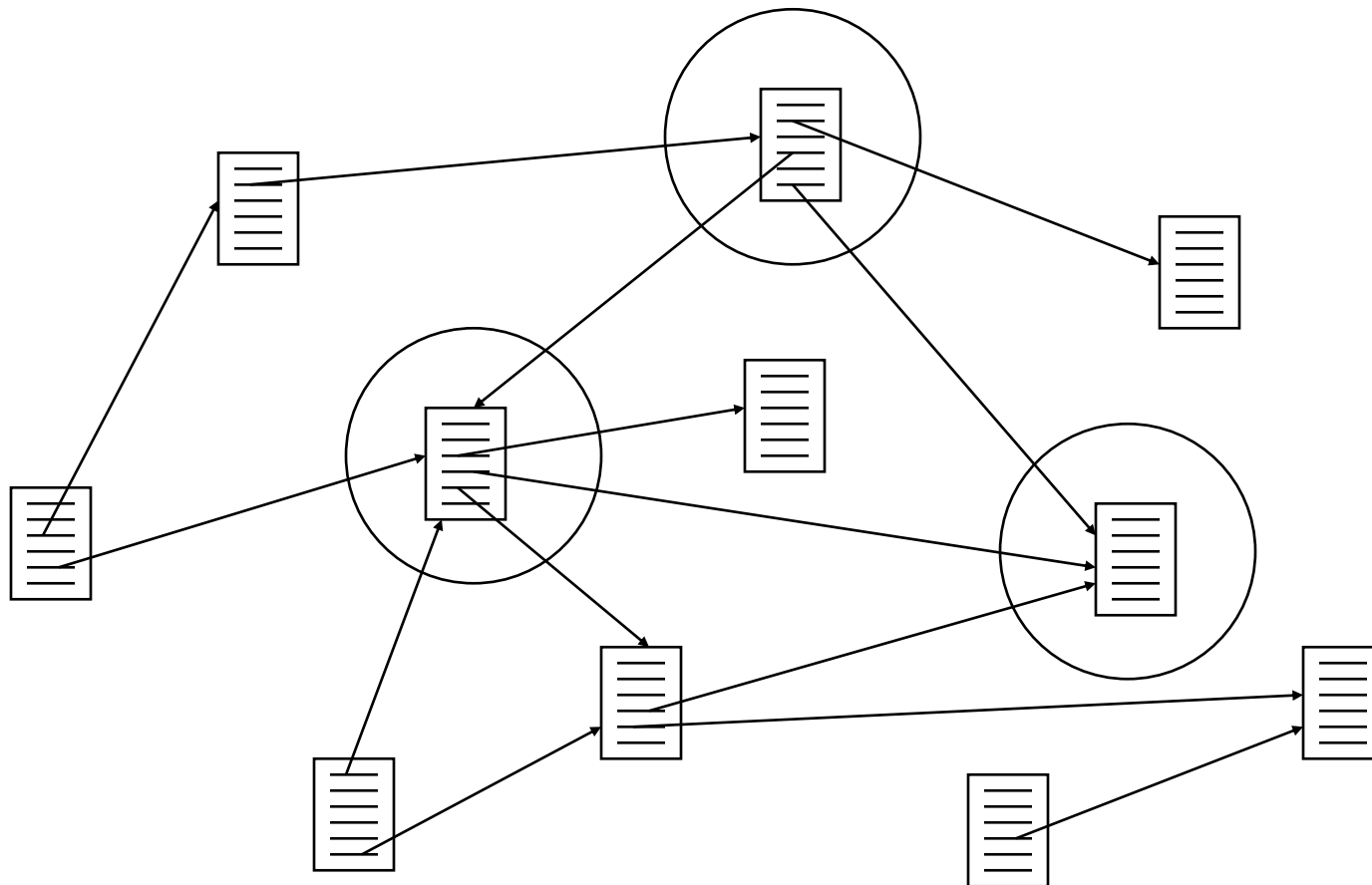
提纲

- ① 上一讲回顾
- ② 锚文本
- ③ 引用分析
- ④ PageRank
- ⑤ HITS: Hub节点 & Authority节点

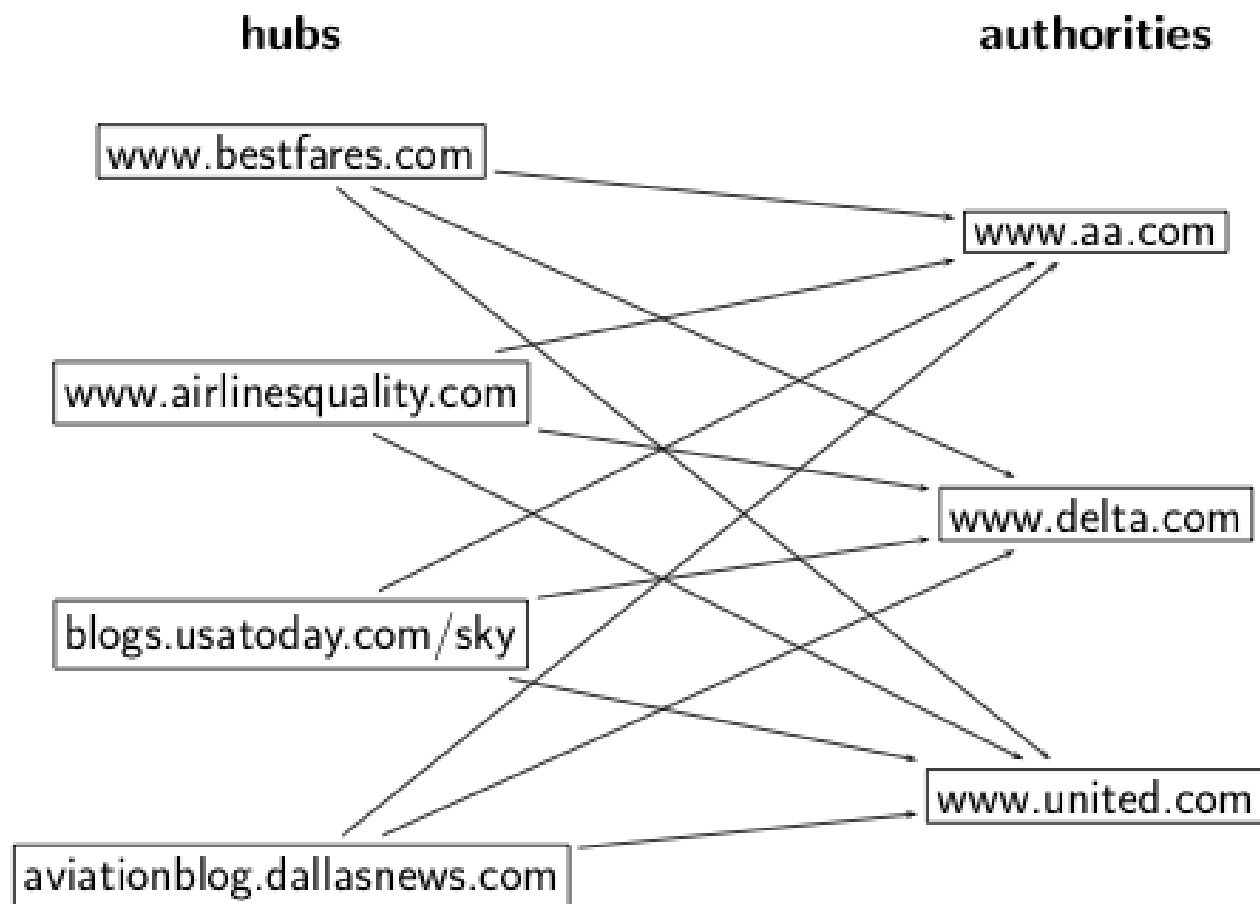
IBM的HITS算法

- HITS(Hyperlink-Induced Topic Search)
- 每个网页计算两个值
 - Hub: 作为目录型或导航型网页的权重
 - Authority: 作为权威型网页的权重

Hub & Authority



例子



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, WWW 08)

[Chicago Bulls]的权威网页

Navigation Bar: NBA D-LEAGUE WNBA GLOBAL TEAMS MOBILE NBA TICKETS FANTASY NBATV STORE VIDEO

Header: bulls.com THE OFFICIAL SITE OF THE CHICAGO BULLS Delivered by at&t

Secondary Navigation: TICKETS TEAM NEWS SCHEDULE FEATURES GAME NIGHT INSIDE THE BULLS HISTORY STORE SEARCH

Left Sidebar:

For!!! Golf with the Bull!
 Tickets for the Chicago Bulls/Verizon Wireless **For!!! Golf** are now on sale! Join Bulls' personalities including current players, coaches, legends, broadcasters and entertainment teams on August 17 at the White Pines Golf Club in Bensenville, IL.

- 2009.10: [James & Michael Smith](#)
- [Mookie Blaylock](#) | [David Lee](#) | [Trent](#)
- [2009](#) | [News](#) | [Gigs](#) | [Events](#) | [Sponsor](#)

Bottom Left: Verizon FAN POLL

Center Section:

Photo of a man speaking at a press conference. Text below: **Draft Workouts**

Right Sidebar:

BULLSEYE POWERED BY KIA KIA MOTORS

CALENDAR	TICKETS
SEASON TICKETS	TICKET EXCHANGE
GROUP TICKETS	E-NEWSLETTER

SEASON TICKETS

CHICAGO BULLS PRESENTED BY **HARRIS**

查询[Chicago Bulls]的导航型网页

1.62 www.geocities.com/Colosseum/1778
“Unbelieveabulls!!!!”

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

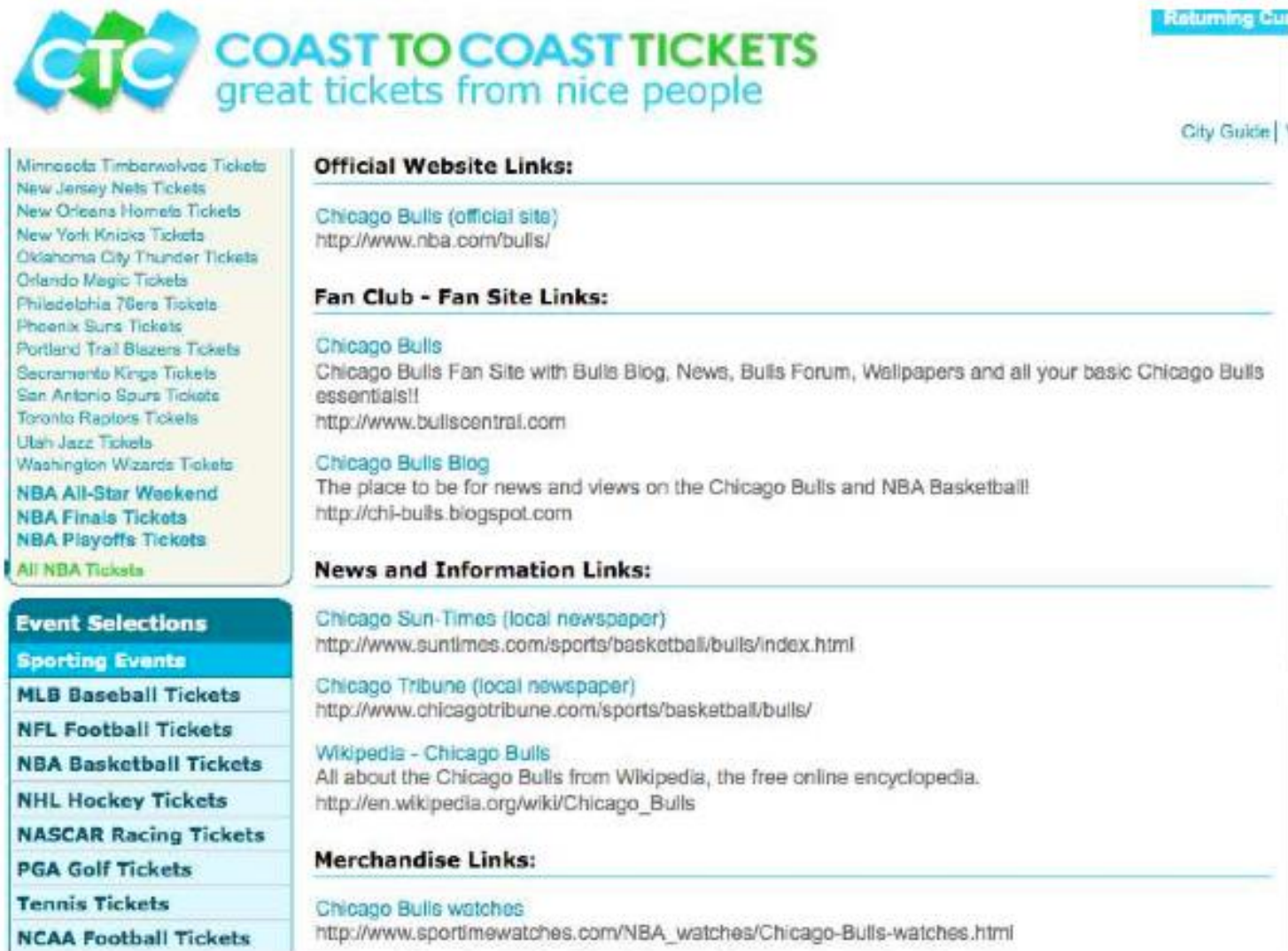
0.74 www.geocities.com/Hollywood/Lot/3330/Bulls.html
“Chicago Bulls”

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



The screenshot shows the 'COAST TO COAST TICKETS' website with the tagline 'great tickets from nice people'. The layout includes a top navigation bar with a 'Returning Customer' link and a 'City Guide' link. The main content area is divided into several sections:

- Left Sidebar:** A list of NBA teams with links to their tickets, including Minnesota Timberwolves, New Jersey Nets, New Orleans Hornets, New York Knicks, Oklahoma City Thunder, Orlando Magic, Philadelphia 76ers, Phoenix Suns, Portland Trail Blazers, Sacramento Kings, San Antonio Spurs, Toronto Raptors, Utah Jazz, and Washington Wizards. It also includes links for 'NBA All-Star Weekend', 'NBA Finals Tickets', 'NBA Playoffs Tickets', and 'All NBA Tickets'.
- Official Website Links:** A section titled 'Official Website Links:' containing a link to the Chicago Bulls official site at <http://www.nba.com/bulls/>.
- Fan Club - Fan Site Links:** A section titled 'Fan Club - Fan Site Links:' containing two links: 'Chicago Bulls' (Chicago Bulls Fan Site with Bulls Blog, News, Bulls Forum, Wallpapers and all your basic Chicago Bulls essentials!!) at <http://www.bullscentral.com>, and 'Chicago Bulls Blog' (The place to be for news and views on the Chicago Bulls and NBA Basketball!) at <http://chi-bulls.blogspot.com>.
- News and Information Links:** A section titled 'News and Information Links:' containing three links: 'Chicago Sun-Times (local newspaper)' at <http://www.suntimes.com/sports/basketball/bulls/index.html>, 'Chicago Tribune (local newspaper)' at <http://www.chicagotribune.com/sports/basketball/bulls/>, and 'Wikipedia - Chicago Bulls' (All about the Chicago Bulls from Wikipedia, the free online encyclopedia.) at http://en.wikipedia.org/wiki/Chicago_Bulls.
- Merchandise Links:** A section titled 'Merchandise Links:' containing a link to 'Chicago Bulls watches' at http://www.sportswatches.com/NBA_watches/Chicago-Bulls-watches.html.
- Event Selections:** A section titled 'Event Selections' with a sub-section 'Sporting Events' containing links for 'MLB Baseball Tickets', 'NFL Football Tickets', 'NBA Basketball Tickets', 'NHL Hockey Tickets', 'NASCAR Racing Tickets', 'PGA Golf Tickets', 'Tennis Tickets', and 'NCAA Football Tickets'.

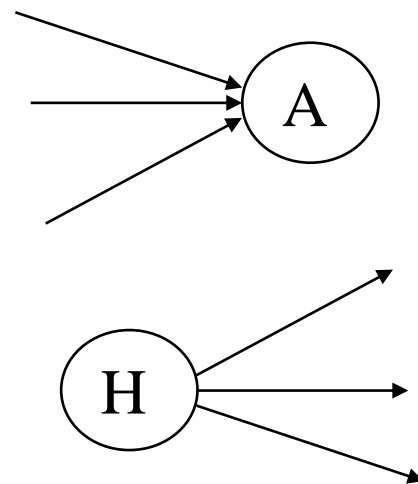
计算方法

$$A(p) = \sum H(q_i)$$

(其中 q_i 是所有链接到 p 的页面)

$$H(p) = \sum A(r_i)$$

(其中 r_i 是所有页面 p 链接到的页面)



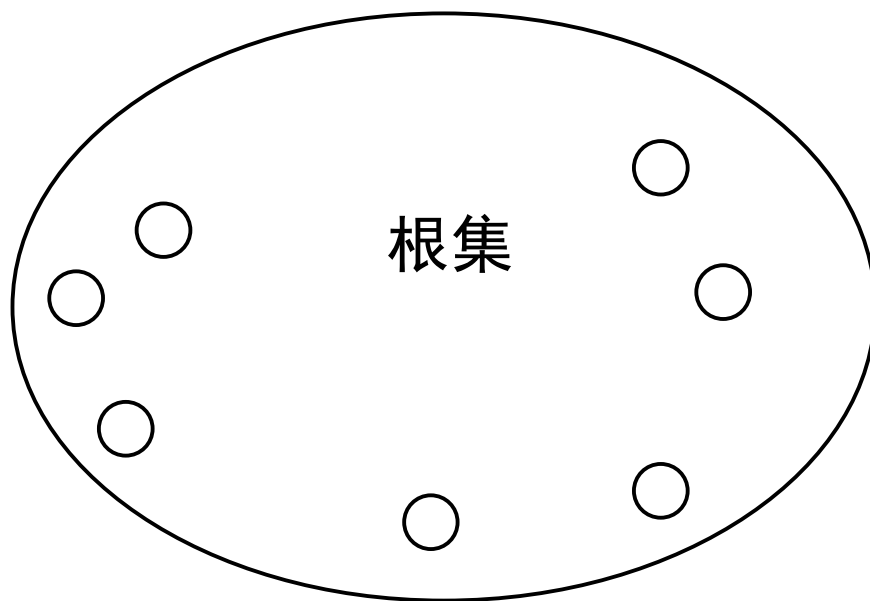
- (1) 一个网页被越重要的导航型网页指向越多，那么它的Authority越大；
- (2) 一个网页指向的高重要度权威型网页越多，那么它的Hub越大。

HITS算法也是收敛的，也可以通过迭代的方式计算。

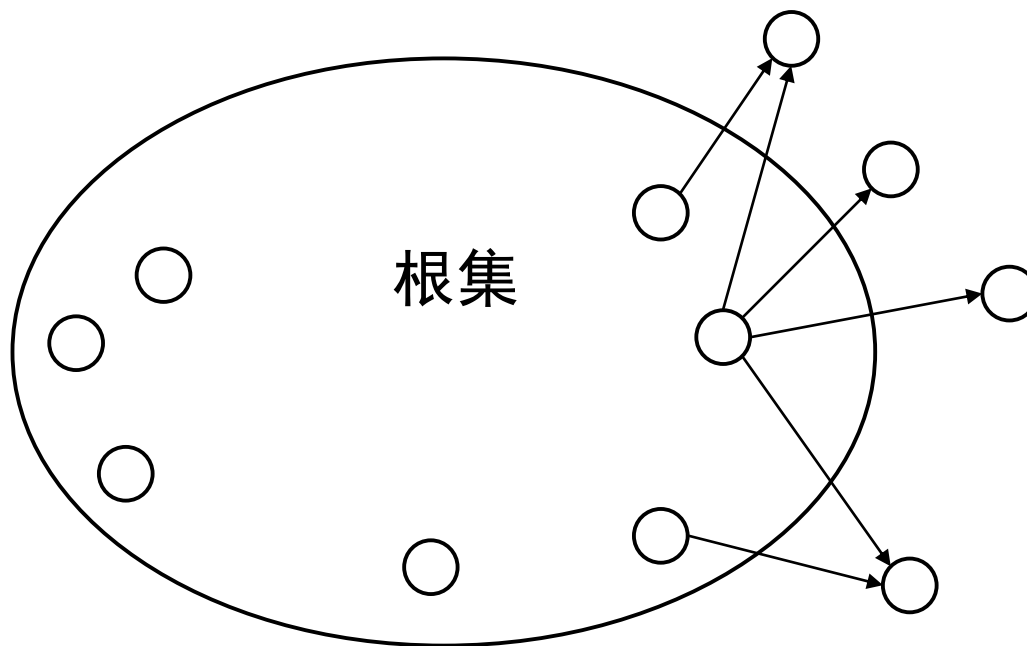
HITS算法的实际计算过程

- 首先进行Web搜索；
- 搜索的结果称为根集(**root set**)； (注：从搜索结果中选择一部分排名靠前的网页作为根集，也叫做种子集合)
- 将所有链向种子集合和种子集合链出的网页加入到种子集合；
- 新的更大的集合称为基本集(**base set**)；
- 最后，在基本集上计算每个网页的hub值和authority值 (该基本集可以看成一个小Web图)。

根集和基本集 (1)

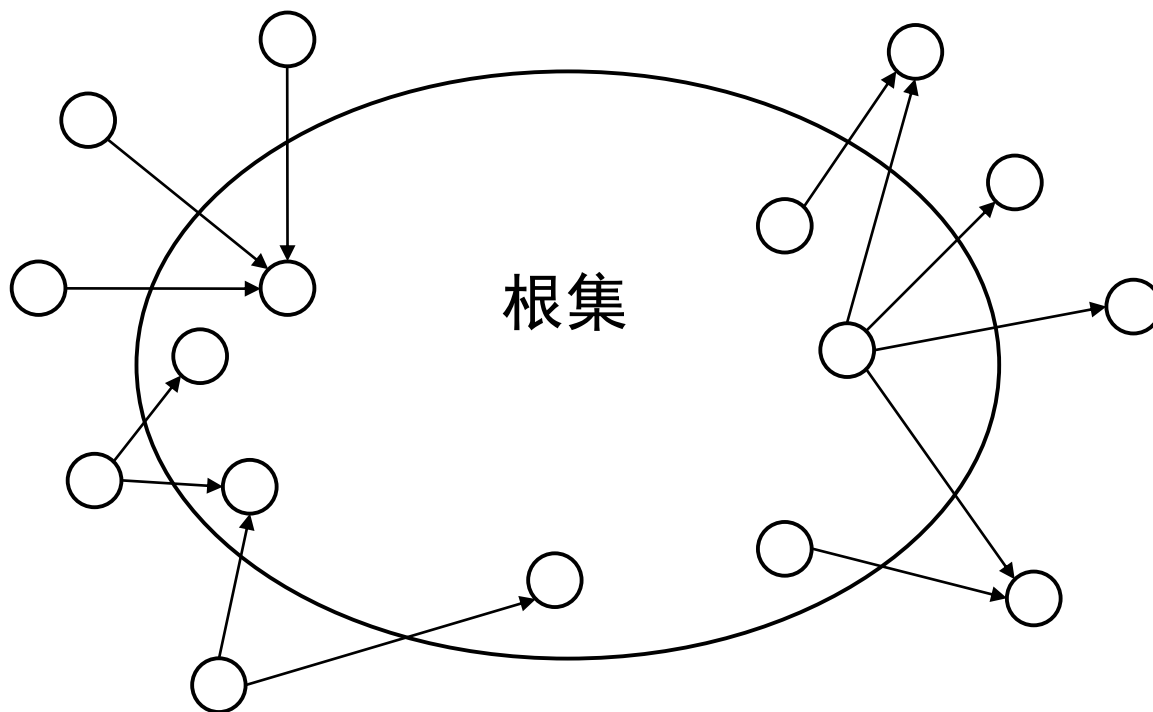


根集和基本集 (2)



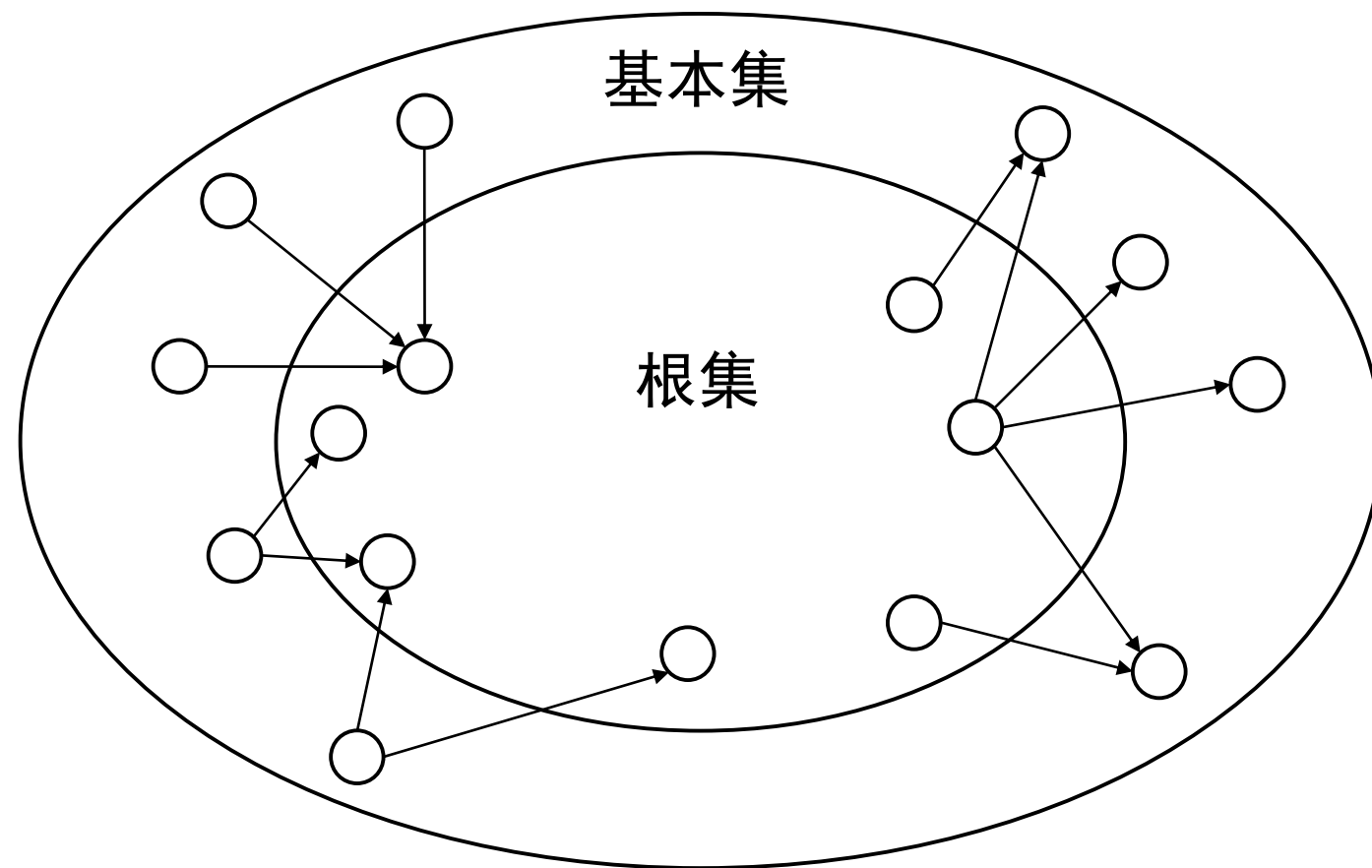
根集中节点链向的网页节点

根集和基本集 (3)



指向根集节点的那些节点

根集和基本集 (4)



基本集

根集和基本集 (5)

- 根集（注：种子集合）往往包含200-1000个节点
- 基本集可以达到5000个节点

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。
 - 网页的Pagerank是一种静态评分，需要与查询相关的评分结合进行网页排序
- HITS算法的计算与查询主题相关，检索之后再行计算，因此，不适合于大型搜索引擎。

本讲内容

- 锚文本: Web上的链接相关信息为什么对IR有用?
- 引用分析(Citation analysis): PageRank及其他基于链接排序方法的数学基础
- PageRank : 一个著名的基于链接分析的排序算法(Google)
- HITS : 另一个著名的基于链接分析的排序算法(IBM)

参考资料

- 《信息检索导论》 第21章

Kleinberg, Jon (1999). [Authoritative sources in a hyperlinked environment](#). *Journal of the ACM*. **46** (5): 604–632.

<http://www2004.org/proceedings/docs/1p309.pdf>

<http://www2004.org/proceedings/docs/1p595.pdf>

<http://www2003.org/cdrom/papers/refereed/p270/kamvar-270-xhtml/index.html>

<http://www2003.org/cdrom/papers/refereed/p641/xhtmll/p641-mccurley.html>

[The WebGraph framework I: Compression techniques \(Boldi et al. 2004\)](#)