Web搜索

目录

- Web搜索基础
  - Web与文档集的不同
  - 近似重复检测
- Web采集
  - 采集器
  - 连接服务器
- 链接分析
  - 锚文本
  - 链接分析：Pagerank
  - 链接分析：HITS

Web搜索引擎简史

- 基于关键词的搜索 1995-1997
  - Altavista, Excite, Infoseek, Inktomi, Lycos
- 支付搜索: Goto (变成 Overture.com →Yahoo!)
  - 谁付的钱多就把谁放在前面：关键词会被拍卖
- 1998+: Google提出基于链接的排序
  - 除了Inktomi外其他早期搜索引擎都灭亡
  - 同期 Goto/Overture的年收入约10亿美元
- Google策略：独立于搜索结果，在页面的一侧添加相关付费广告
  - Yahoo跟着学, 收购了 Overture (用作广告) and Inktomi (用作搜索)
- 2005+：赢得了更多的搜索份额，在欧洲和北美占统治地位
- 2009: Yahoo!和Microsoft进行合作，由Microsoft提供搜索技术，Yahoo！自己负责自己的广告
- 2013：Bing在美国份额突破29.3% (Google: 66.7%)
- Google一家独大(中国百度)，未来走势如何？ 未知

付费广告

Google

算法生成结果

百度

海尔-你的生活智慧 我的智慧生活 官网

Haier

海尔是全球大型家电第一品牌。在互联网时代，海尔通过解决方案和管理模式的破坏性创新，与用户进行设计、生产、服务全流程交互，为用户量身定制个性化的智能家居体验和引领的美好生活解决方案。海尔通过开放平台模式，整合全球一流资源，构建创新生态圈和商业生态圈。

www.haier.com 2014-03 - 品牌推广

- 海尔大型家电第五次蝉联全球第一
- 海尔兄弟新形象征集 好兄弟，一起雷欧

产品体验	智能家电	【海尔商城】	服务支持
匀冷冰箱	智能云洗衣机	海尔冰箱秒杀	下载中心
Mooka智能电视	智能云冰箱	海尔热水器热销	维修与安装
海尔空气盒子	智能云热水器	海尔洗衣机爆款	在线客服

海尔冰箱 海尔商城 一站式销售服务平台 推广链接

海尔商城 eHaier.com

海尔冰箱?海尔,全球白电领导品牌!海尔商城提供家电选购设计,送装同步!

海尔冰箱 - 海尔洗衣机 - 海尔热水器 - 货到付款

www.ehaier.com V3

JD 冰箱,「京东」全网底价!正品行货! www.jd.com V3

冰箱,京东,一站购物终极体验!<闪电发货+火速到达>[京东],值得信赖!

买海尔(Haier)冰箱到<国美在线> 100%正品 7天包退 www.gome.com.cn V

海尔BCD-133EN冰箱133升省电... 现价¥:1128 <国美在线>品质保证 去看看>>

海尔BCD-186KB冰箱186升史上... 现价¥:1298 <国美在线>品质保证 去看看>> 更多

海尔冰箱大全、冰箱报价、海尔冰箱最新报价-太平洋产品报价

太平洋电脑网提供海尔冰箱大全全面服务信息,包含海尔冰箱报价、参数、评测、比较、点评、论坛等,帮您全面了解海尔冰箱。

product.pconline.com.cn/icebox/haier/ 2014-02-27 - 百度快照

海尔冰箱（共585款海尔冰箱）海尔冰箱报价_参数_评论_百度微购

热门品牌: 全部 海尔 统帅 卡萨帝

价格区间: 全部 0-1999 2000-3999 4000-5999 6000-9999 10000-14999 更多

基本全是广告

- 『海尔商城』-全场免运-货到付款
- 『海尔商城』-100%官方正品-天天爆款
- 『温情发送』海尔商城全场通用代金券！

相关电器 展开


海尔空调

容声

格力空调


美的冰箱

相关企业 展开

京东商城

格力

苏宁

国美

相关网站

淘宝

苏宁易购

海尔商城

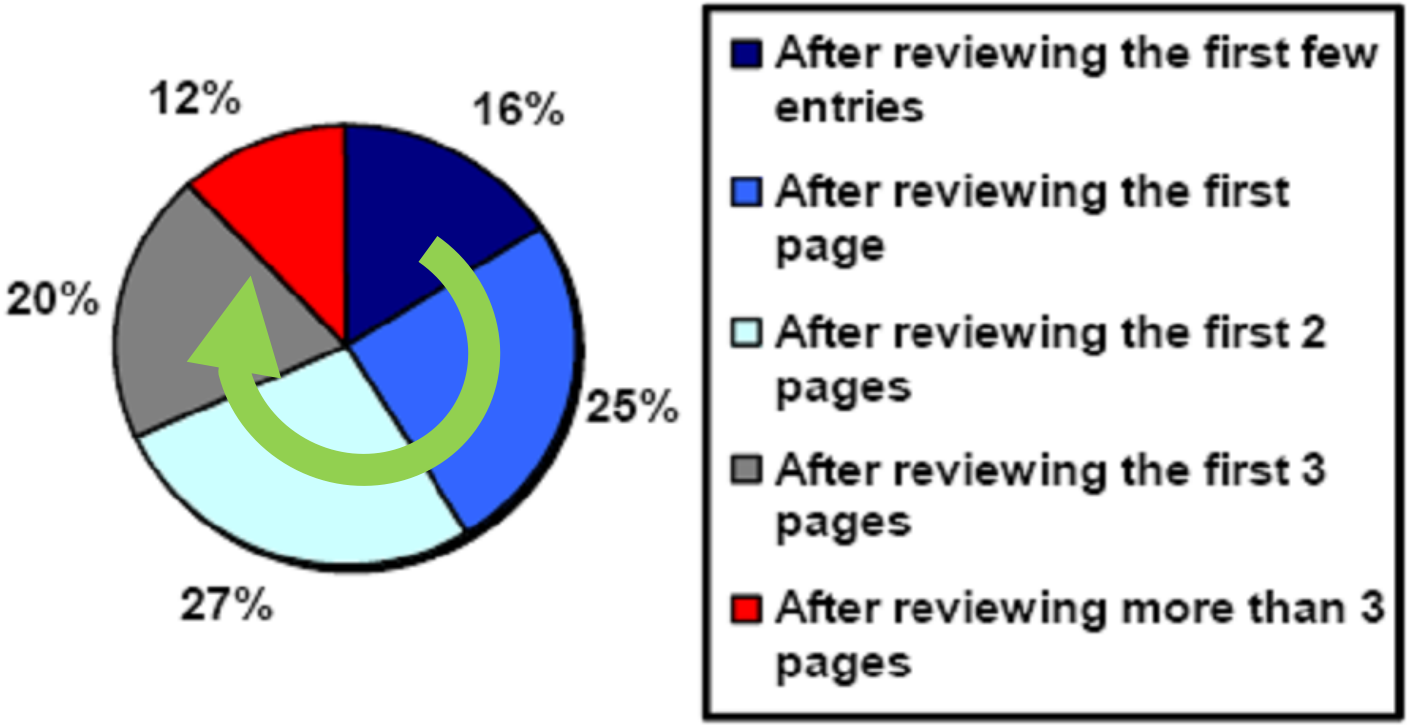

国美电器网上商城

冰箱风云榜

排名 搜索指数

用户会深入搜索结果多深？

"When you perform a search on a search engine and don't find what you are looking for, at what point do you typically either revise your search, or move on to another search engine? (Select one)"



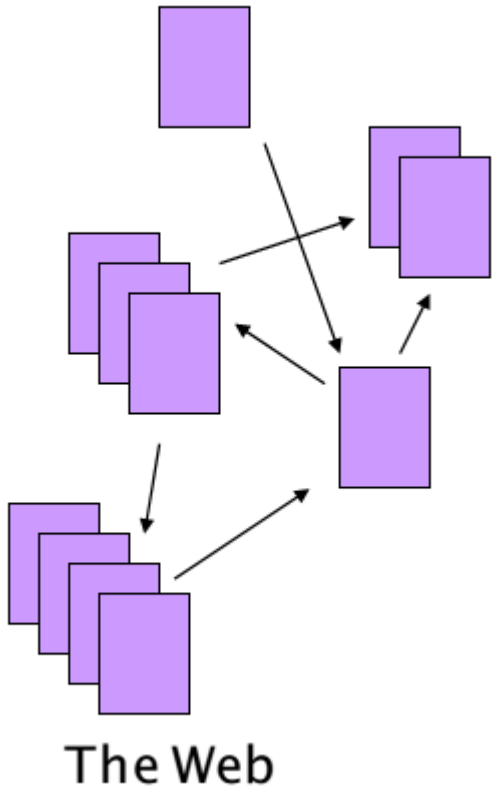
(Source: iprospect.com WhitePaper_2006_SearchEngineUserBehavior.pdf)

对搜索结果的经验性评价

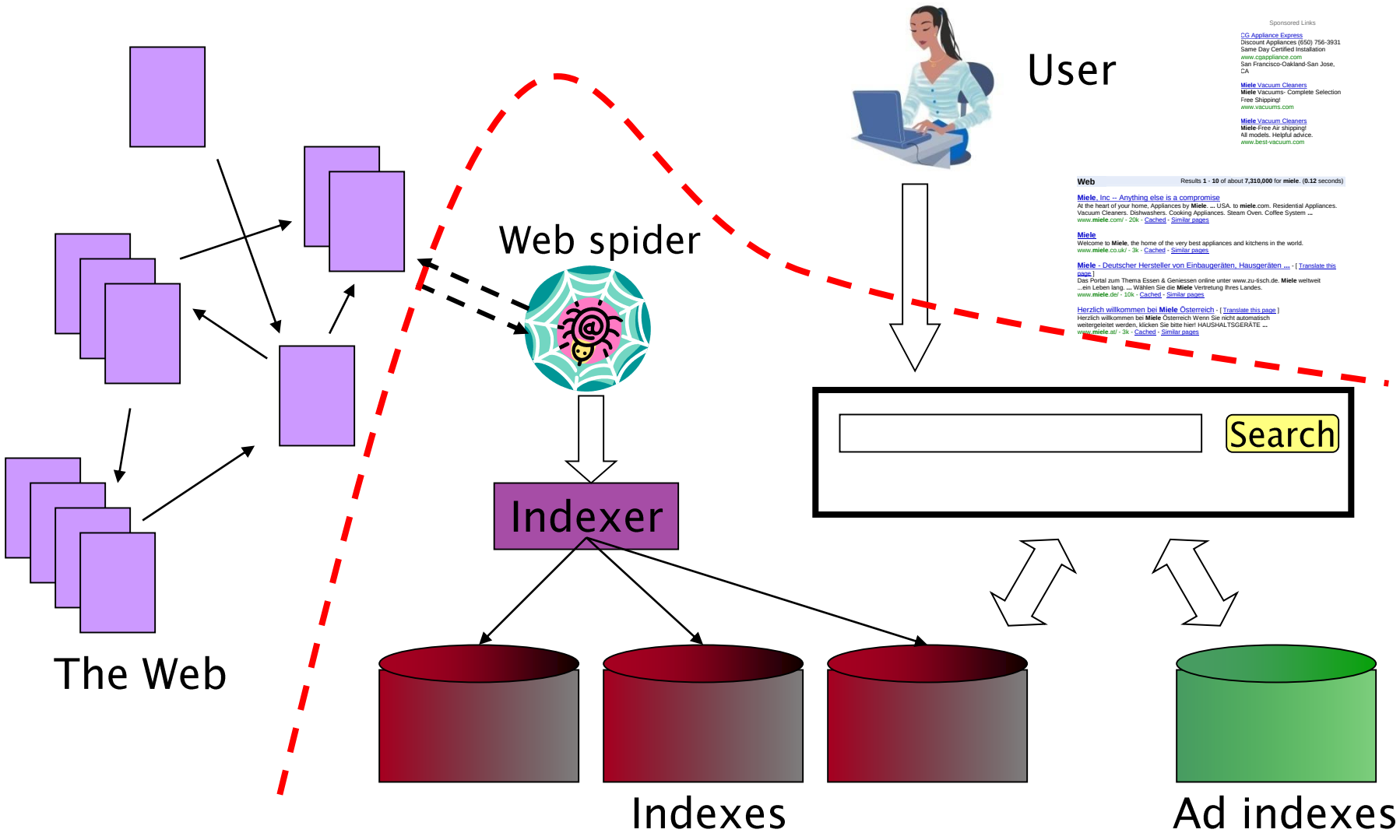
- 页面质量良莠不齐
  - 仅仅相关是不够的
  - 质量也很关键
    - 内容: 可信, 多样, 不重复, 容易维护
    - 页面可读性: 显示得又快又好
    - 无打扰: 无弹出广告等
- 正确率和召回率 Precision vs. recall
  - 在互联网上，召回率不再那么重要
- 什么比较重要？
  - 头版头条的正确率(反例：百度)
  - 全面 – 要能处理模糊的查询词
    - 匹配结果比较少的时候召回率很重要
- 用户的认知可能非学术的，但是是有意义的

Web文档集

- 没有设计/多人协作
- 分散的内容创作、链接，民主化的发布
- 内容多样：包含真理、谎言、矛盾和大量猜测 …
- 异构：非结构化的(text, html, …)，半结构化的 (XML，有注释的照片)，结构化的(数据库)…
- 规模比之前的文本集大得多… 但是其中有很多重复的记录
- 增长– 最开始每几个月就翻一倍，现在涨速下降但总量依然在扩大
- *内容可能是动态生成的*



Web搜索基本流程



小结：Web与文档集不同

- 规模：海量？动态？
- 相关性度量
  - 返回结果排序不仅仅依赖于q与d的相关性
  - 广告？用户认可？
- 重复？

目录

- Web搜索基础
  - Web与文档集的不同
  - 近似重复检测
- Web采集
  - 采集器
  - 连接服务器
- 链接分析
  - 锚文本
  - 链接分析：Pagerank
  - 链接分析：HITS

重复文档/近似重复文档

- 网上到处都是相同的内容
- 完全**复制** Duplication
  - 可以通过指纹(fingerprints，比如64位Hash)来检测精确匹配
- 大多数情况是**近似重复** Near-Duplication
  - E.g., 两份文本仅仅是日期不同
  - 通过编辑距离计算语法上的相似性
  - 通过一定的阈值来检测近似复制
    - E.g., Similarity > 80% => 文档近似复制
    - 通过阈值来检测是不可传递的 (AB近似，BC近似不能推断AC近似)

相似性计算

- **搭叠Shingles**（*N*元词 *N*-Grams）
  - 给定正整数*k*及文档*d*的一个词项序列，可以定义文档*d*的*k*-shingle为*d*中所有*k*个连续词项构成的序列。
  - a rose is a rose is a rose → 4-Grams

    a_rose_is_a

    rose_is_a_rose

    is_a_rose_is

    a_rose_is_a …
- 直观上看，如果两个文档的shingle集合几乎一样，那么它们就满足近似重复。

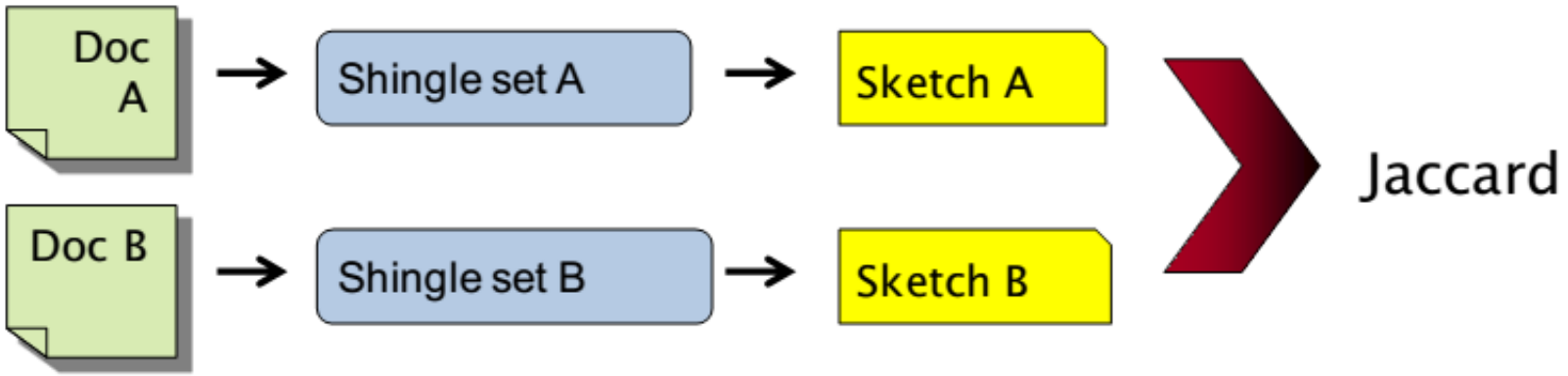
Jaccard系数

- 在文档的Shingle集合上计算 交集大小/并集大

$$\text{Jaccard}(C_i, C_j) = \frac{|C_i \cap C_j|}{|C_i \cup C_j|}$$

Jaccard系数：衡量重复度

- 计算所有文档对之间搭叠的精确交集是非常费时而且难以处理的
- 使用一种聪明的方式从Shingles中选出一个子集(素描***sketch***)来近似计算 (就是抽样Sample)

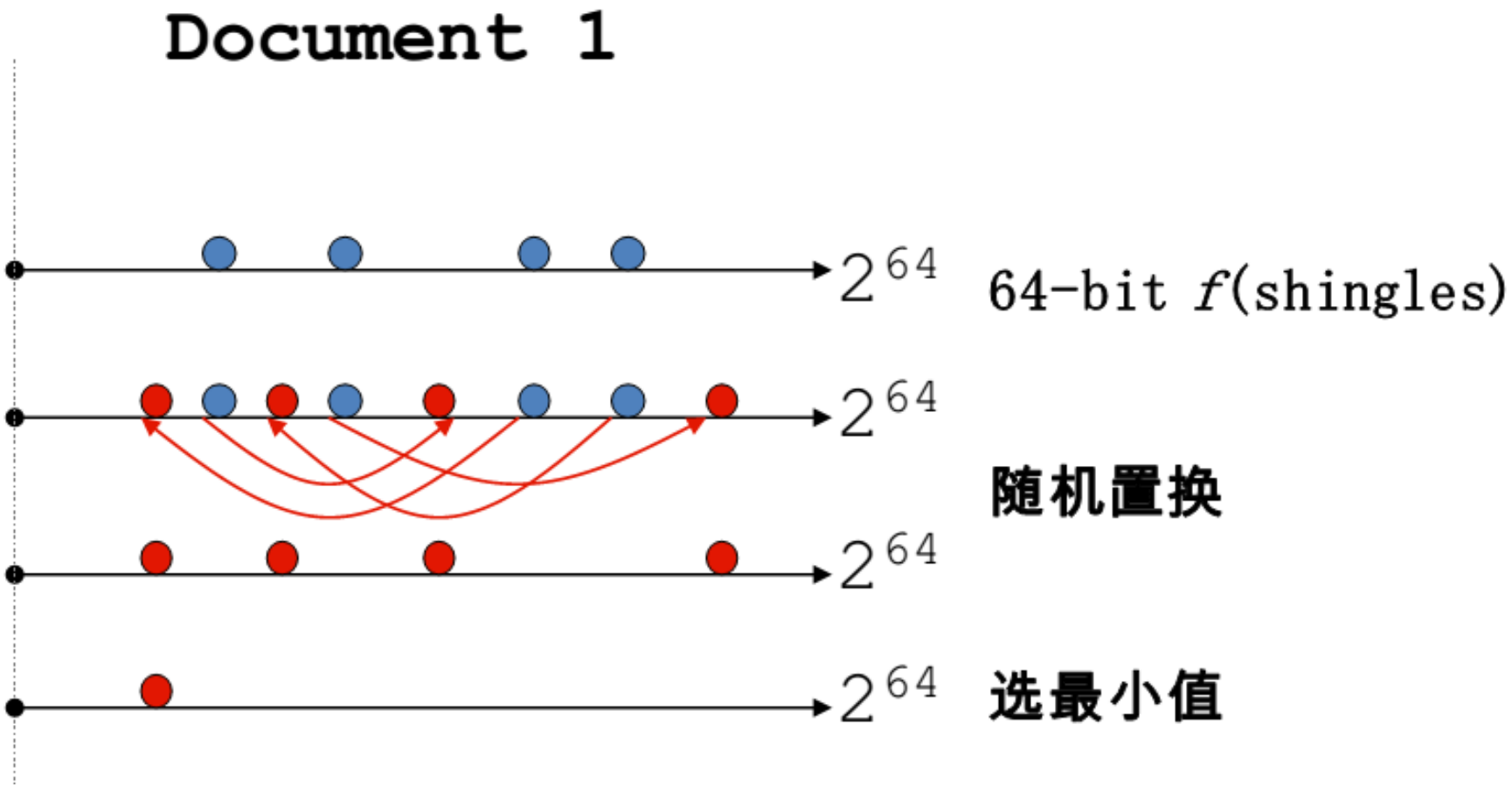


文档的素描(Sketch)

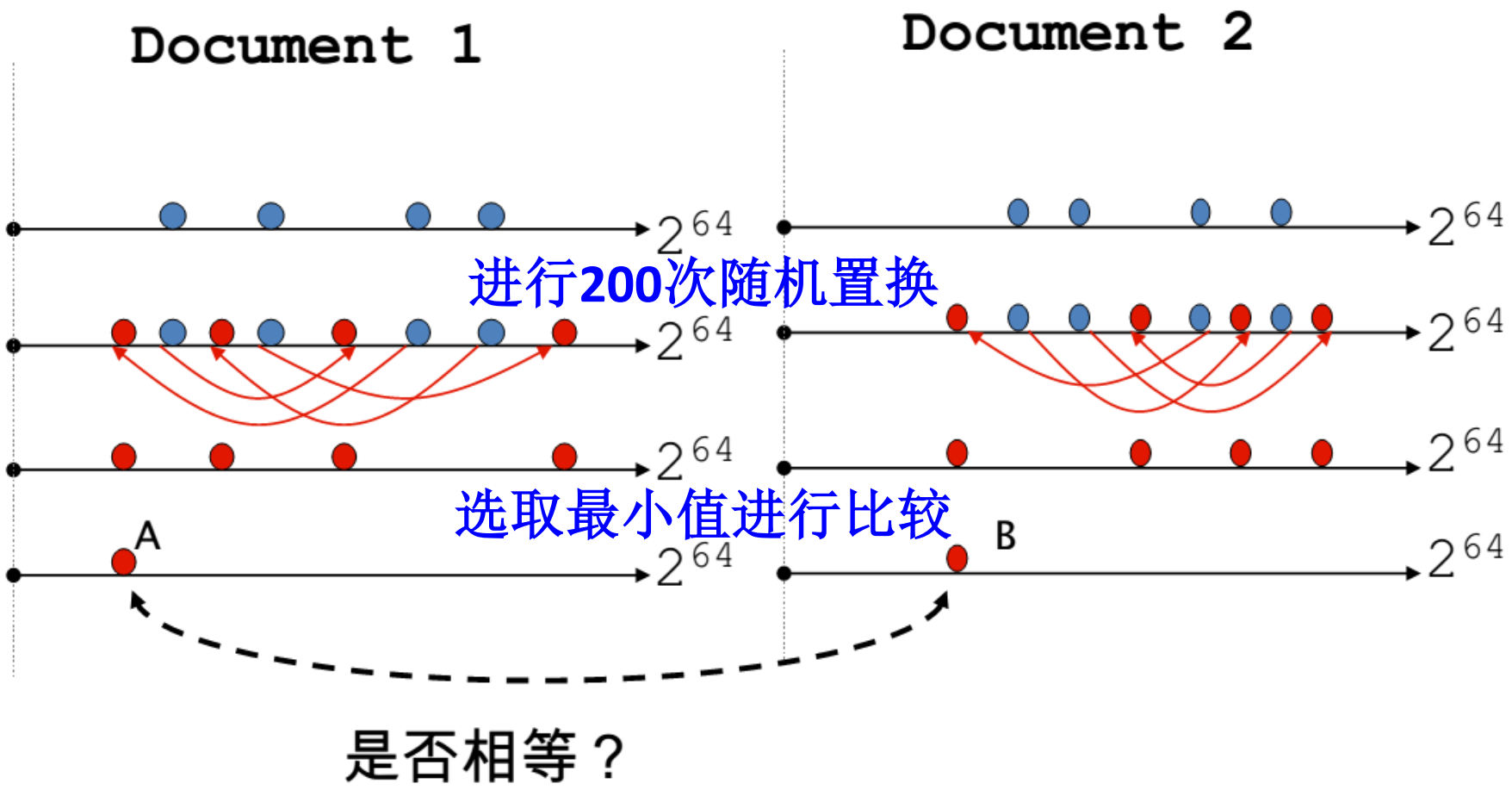
为每篇文档生成素描向量“sketch vector”(大小约为~200)

- 相同向量个数 $\geq t$ (一般80%) 判定为近似near duplicate
- 对文档D, $\text{sketch}_D[i]$ 如下:
  - 利用f函数把所有的搭叠shingles映射到{0...2^m}得到$f(s)$ (e.g., 利用 fingerprinting为每个搭叠s计算一个m 位的哈希)
  - 利用π_i (对{0...2^m}的随机置换函数，即对集合的对象进行随机排序)对所有搭叠的哈希$f(s)$进行随机置换得到$\pi_i(f(s))$，从而形成一个新的随机序列
  - 对上一步的随机置换序列选择 MIN{$\pi_i(f(s))$}

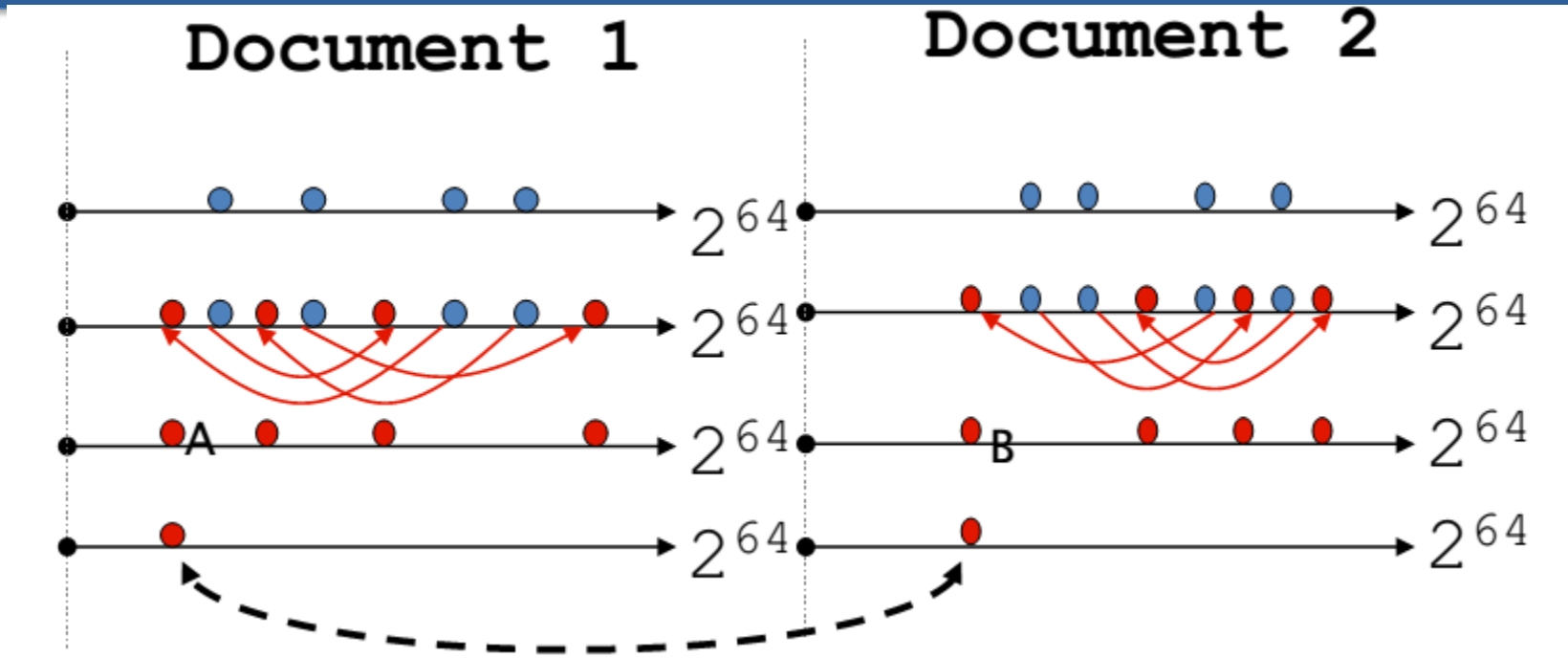
计算 Sketch[*i*] for Doc1



测试 if Doc1.Sketch[*i*] = Doc2.Sketch[*i*]



本质上是对shingle集合进行洗牌、抽样



A = *B* iff the shingle with the MIN value in the union of Doc1 and Doc2 is common to both (i.e., lies in the intersection)

定理19-1: *A* = *B*发生的概率 = 交集大小/并集大小 (Size_of_intersection / Size_of_union)

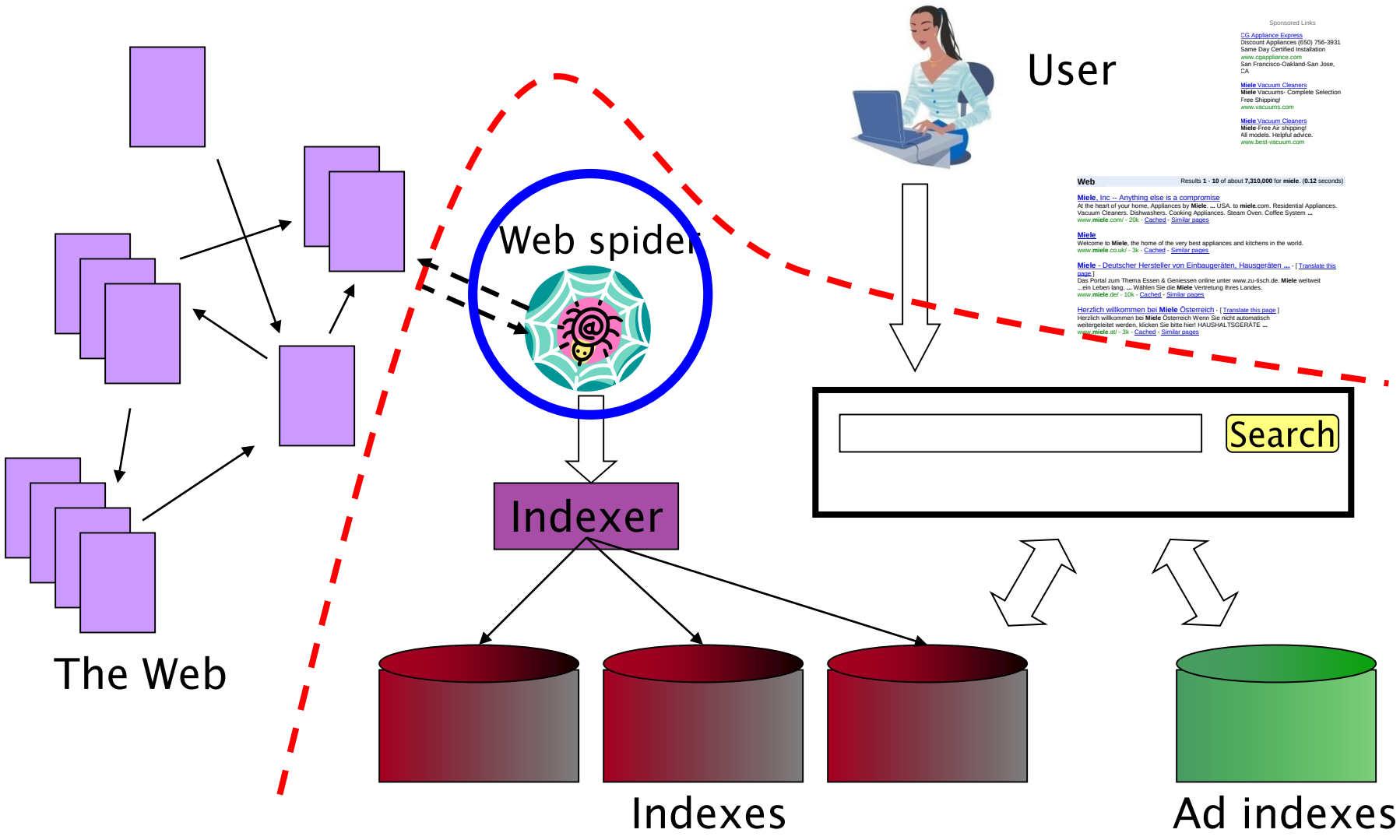
小结：近似重复检测

- Shingle算法的核心思想是将文件相似性问题转换为集合的相似性问题
- 数量较大时，对shingle集合进行抽样，以降低空间和时间计算复杂性
- Shingle抽样主要有三种方法，即Min-Wise，Modm，Mins
  - Mins技术先将shingle和整数集进行映射，然后从中选择最小*s*个元素组成取样集合。
  - 此外，还可以使用shingle的hash值代表shingle进行相似性计算，能够节省一定计算开销。

目录

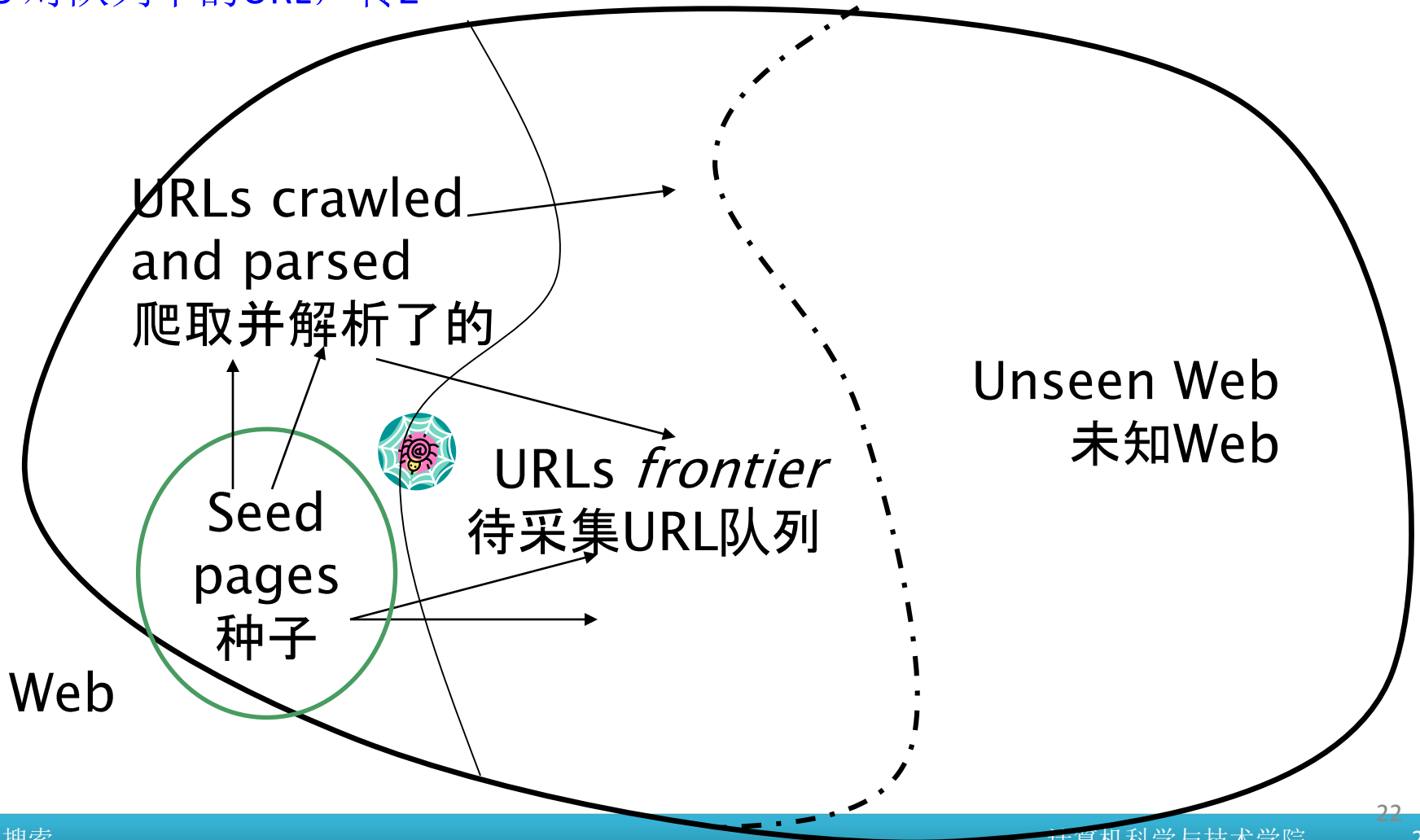
- Web搜索基础
  - Web与文档集的不同
  - 近似重复检测
- Web采集
  - 采集器
  - 连接服务器
- 链接分析
  - 锚文本
  - 链接分析：Pagerank
  - 链接分析：HITS

Web搜索基本流程



Crawling picture

1 从已知的种子URL开始
2 获取页面并进行解析：1) 提取页面中包含的链接；2) 把链接放入URL队列
3 对队列中的URL，转2



采集器必须具有的功能

- **礼貌性**: Web服务器有显式或隐式的策略控制采集器的访问
  - 只爬允许爬的内容、尊重 robots.txt
- **鲁棒性**：能从采集器陷阱中跳出，能处理Web服务器的其他恶意行为
- **分布式**：可以在多台机器上分布式运行
- **可扩展性**：添加更多机器后采集率应该提高
- **性能和效率**：充分利用不同的系统资源，包括处理器、存储器和网络带宽
- 优先抓取“有用的网页”
- **新鲜度**：对原来抓取的网页进行更新
- **功能可扩展性**：支持多方面的功能扩展，例如处理新的数据格式、新的抓取协议等

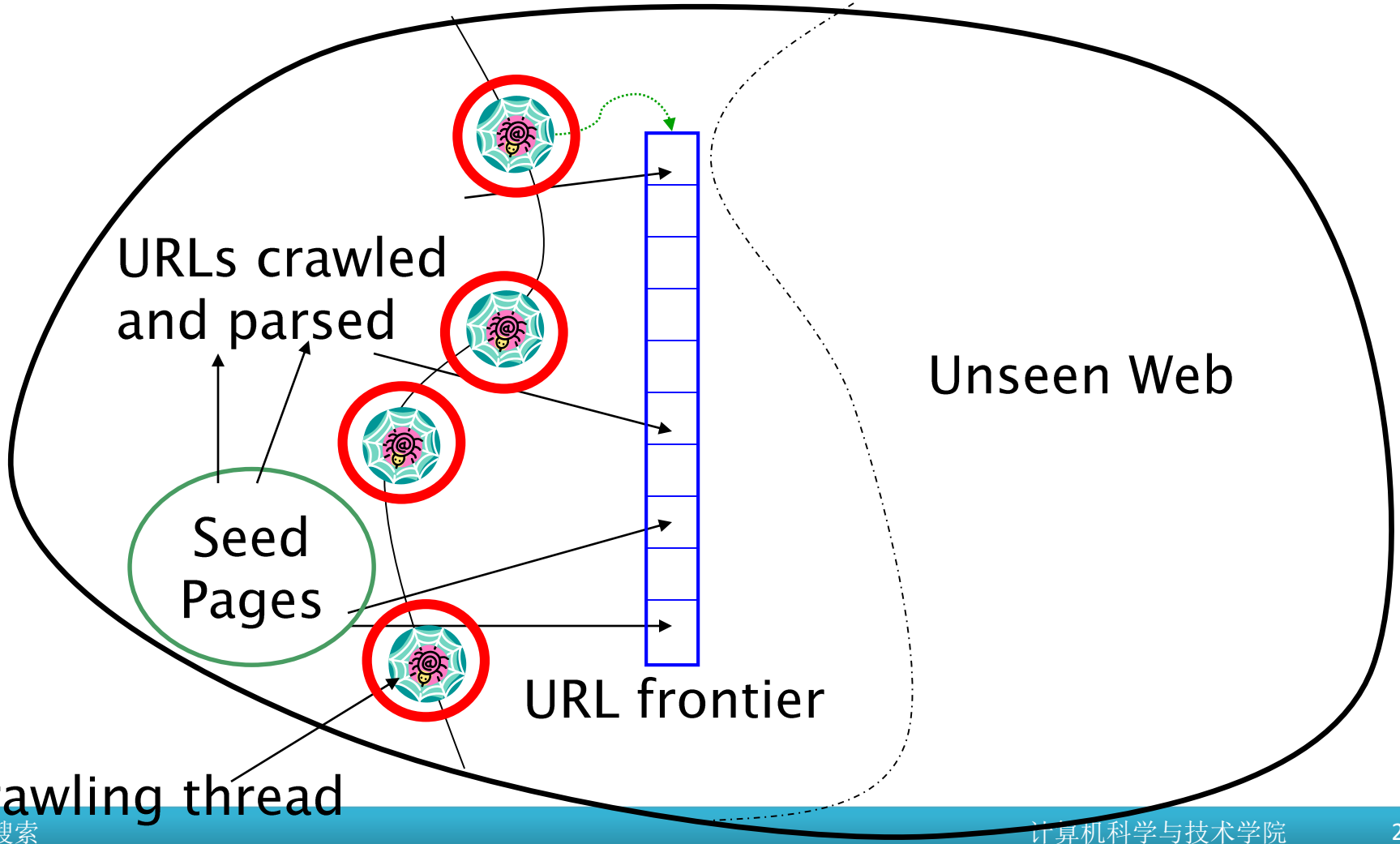
礼貌性

- 显式的礼貌：根据网站站长的说明，选择允许爬取的部分进行爬取
  - 按robots.txt说的做，如下面写法的意思是：任何robot都不能访问“/yoursite/temp/”开头的网址，除了名叫“searchengine”的:
    - User-agent: *
    - Disallow: /yoursite/temp/
    - User-agent: searchengine
    - Disallow:

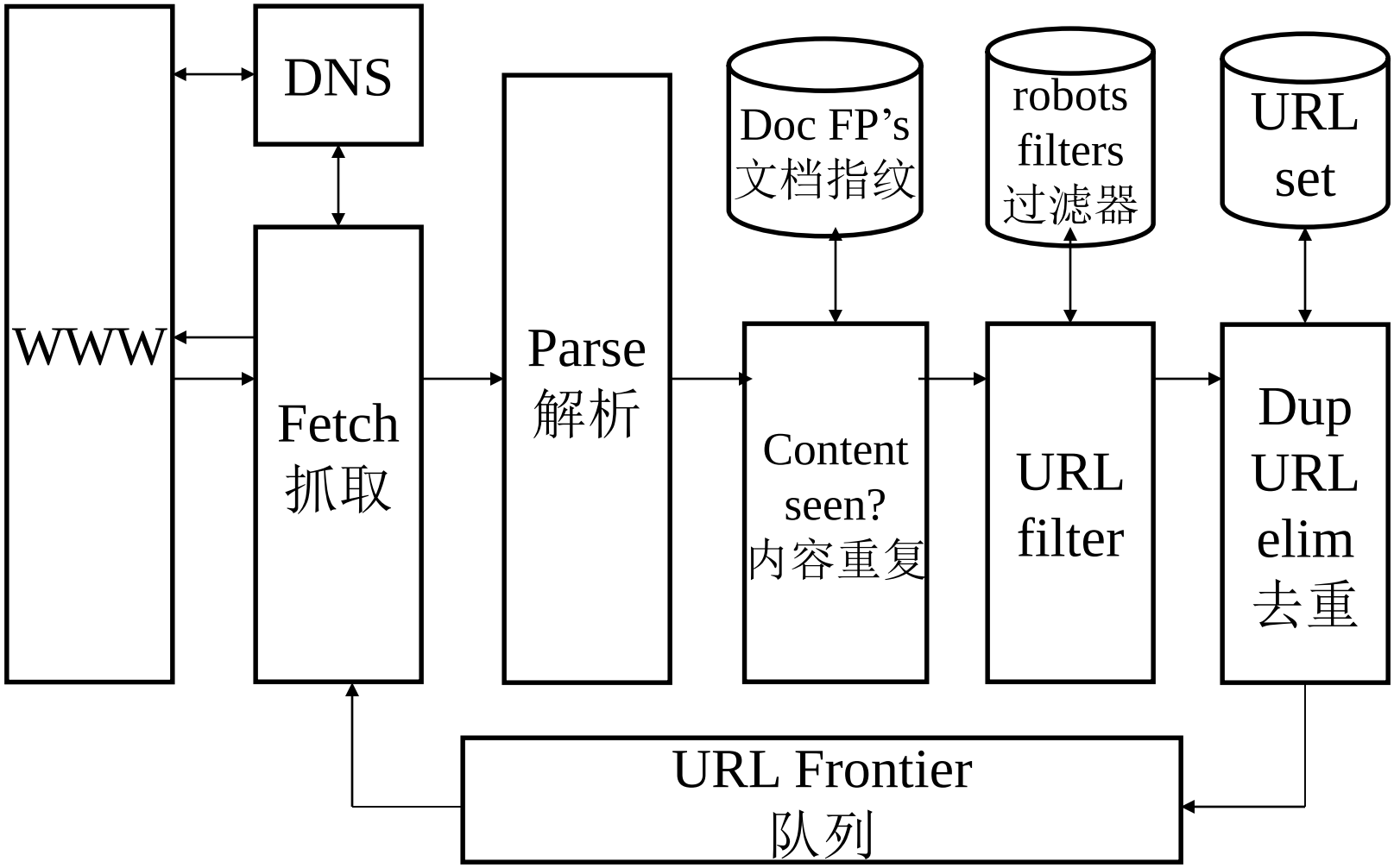
- 隐式的礼貌:即使没有特别的说明，也不应该频繁的访问同一个网站
- **Robots.txt** 源于1994年的协议，对爬取过程进行限制
  http://www.robotstxt.org/orig.html 关于Robots.txt的说明

Updated crawling picture
改进后的采集器

- 多个爬虫



采集器基本架构

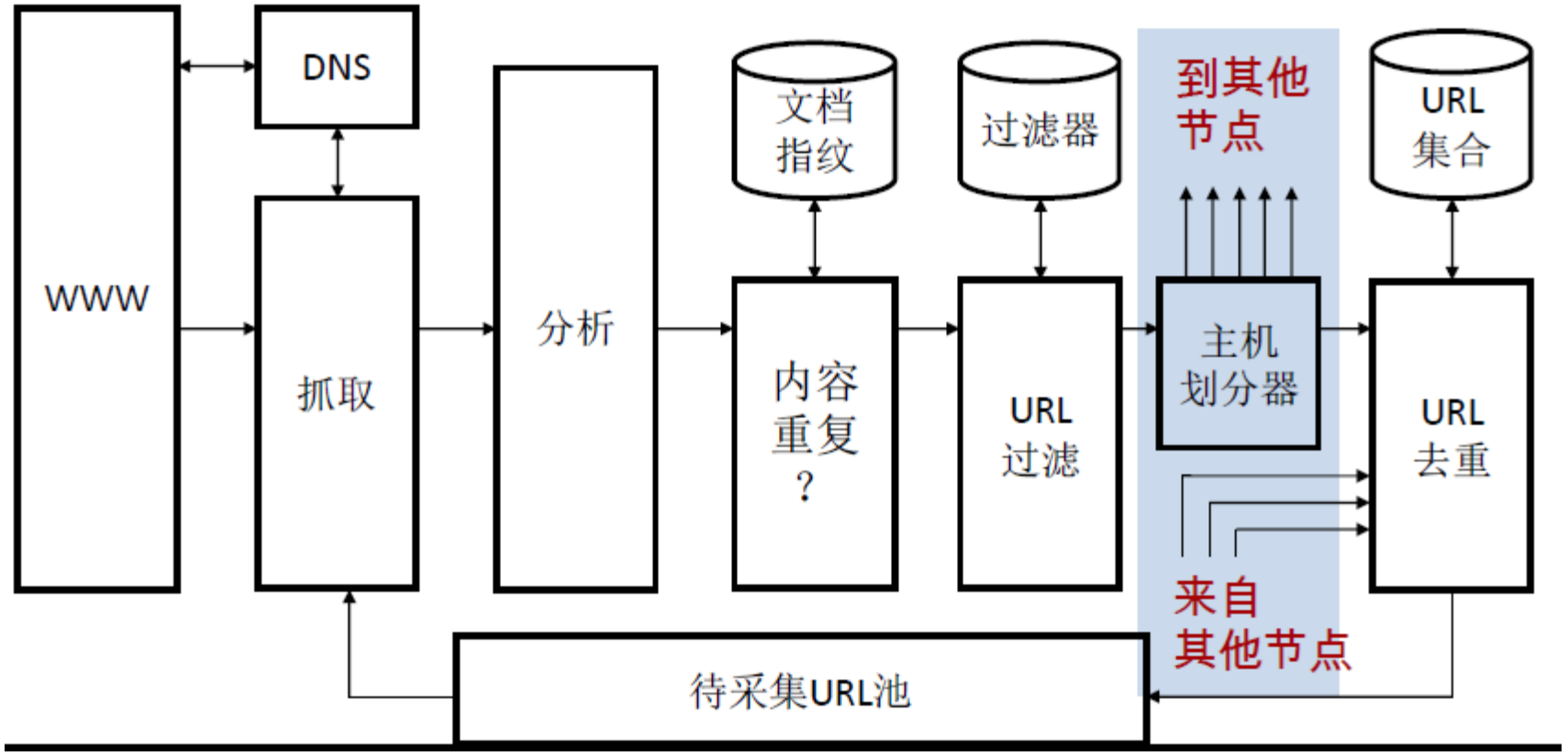

WWW
DNS
Fetch
抓取
Parse
解析
Doc FP's
文档指纹
Content
seen?
内容重复
robots
filters
过滤器
URL
filter
URL
set
Dup
URL
elim
去重
URL Frontier
队列

采集器分布化

- 在分布式系统环境下不同节点的不同进程中运行多个采集线程
  - 地理位置分布的采集系统
- 把要采集的主机分配到每个节点
  - 通过Hash函数或其他针对性的策略
- 不同节点之间怎么通讯？

节点间通信

- 通过过滤检测的URL需要发送到每个节点上进行查重处理



小结：采集器

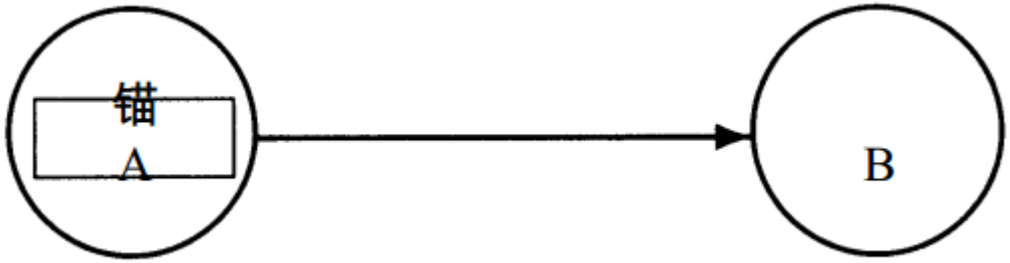
- 礼貌性: Web服务器有显式或隐式的策略控制采集器的访问
  - 只爬允许爬的内容、尊重robots.txt
- 鲁棒性：能从采集器陷阱中跳出，能处理Web服务器的其他恶意行为
- 分布式: 可以在多台机器上分布式运行
- ……

目录

- Web搜索基础
  - Web与文档集的不同
  - 近似重复检测
- Web采集
  - 采集器
  - 连接服务器
- 链接分析
  - 锚文本
  - 链接分析：Pagerank
  - 链接分析：HITS

Web→Web图

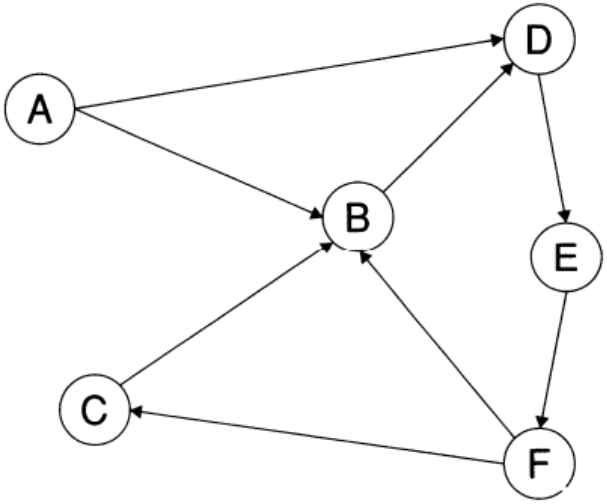
- 可以将整个静态Web看成是静态HTML网页通过超链接互相连接而成的有向图，其中每个网页是图的顶点，而每个超链接则代表一个有向边。



- 包含两个顶点A、B的Web图，每个顶点代表一个网页，A网页上有一个超链接指向B。将所有这样的顶点和有向边集合称为Web图。

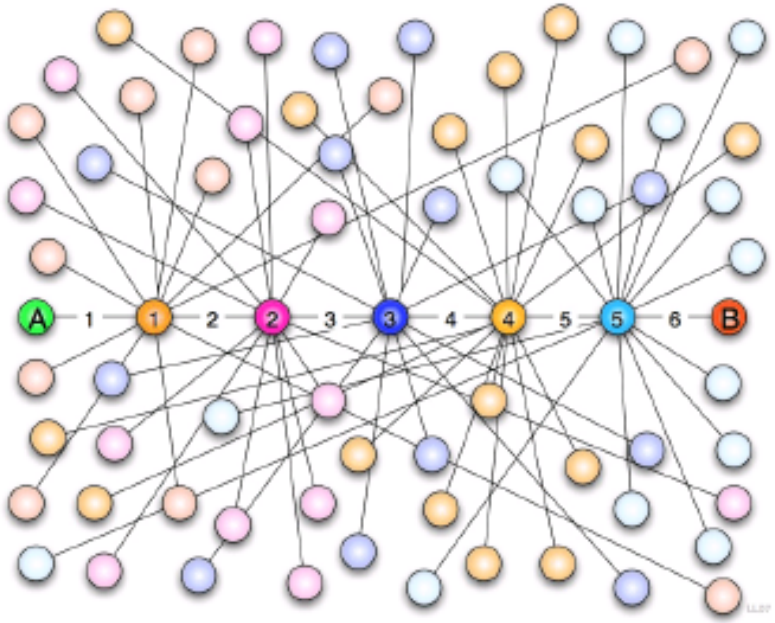
Web特性→Web图特性

- 该有向图可能不是一个强连通(strongly connected)图，既从一个网页出发，沿着超链接前进，有可能永远不会到达另外某个网页。
- 将指向某个网页的链接称为入链接(in-link)，而从某个网页指出去的链接称为出链接(out-link)。一个网页的入链接数目被称为其入度(in-degree)，在一系列研究中得到的网页的平均入度大概从8到15左右不等。同样，某个网页的出链接数目为其出度(out-degree)。



6个网页(分别以A到F标识)，网页B的入度为3、出度为1。该图不是强连通图，因为B不可能到A

Web特性→Web图特性小世界网络



It's a small world

- **It is a 'small world'**
  - Millions of people. Yet, separated by "**six degrees**" of acquaintance relationships
  - Popularized by Milgram's famous experiment
- **Mathematically**
  - Diameter of graph is small ($\log N$) as compared to overall size
  - Property seems interesting given 'sparse' nature of graph but … This property is 'natural' in 'pure' random graphs

Huge graph with small distance

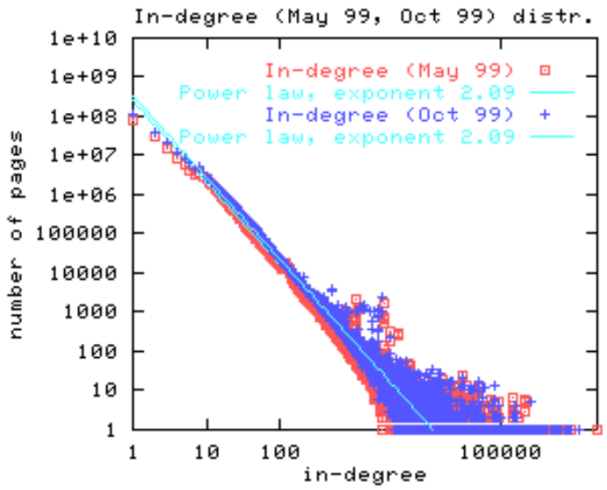
Web特性→Web图特性无标度网络

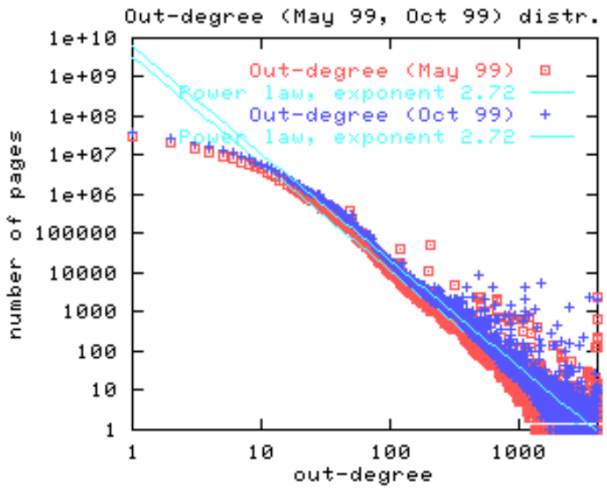
- 站点大小Site Sizes (以页面数量计算)服从**power law** 分布
  - 跨越不同的规模
  - a 在1.6-1.9之间
- 节点的度Connections per Node i服从**power law** 分布
  - Study at Notre Dame University reported
  - a = 2.45 for outdegree distribution
  - a = 2.1 for indegree distribution

$$网页数目 \approx \frac{1}{i^a}$$

Degree distributions of the WWW analyzed in [Broder et al '00]

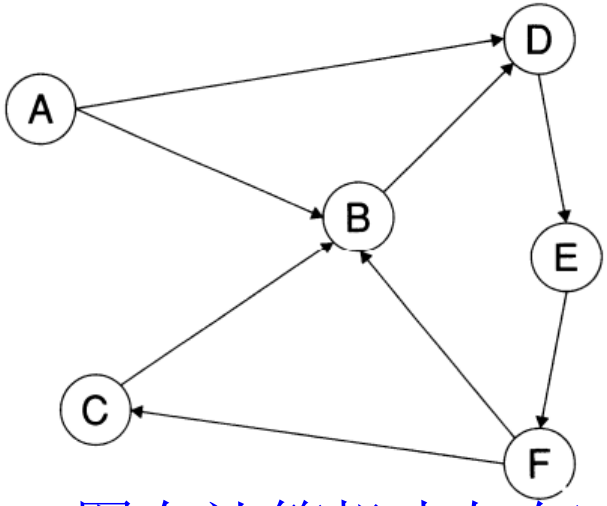
⇒ Web a digraph, study both in- and out-degree distributions





连接服务器

- 支持Web图上的快速查询
  - 哪些URL指向给定的URL？
  - 给定的URL指向哪些URL？
- 在内存中存储了映射表
  - URL到出链, URL到入链
- 应用
  - 采集控制
  - Web图分析
    - 连通性Connectivity, 采集优化
  - 链接分析Link analysis



Web图在计算机中如何表示？

邻接表

- 假定每个网页都用唯一的整数来表示。
- 建立一个类似于倒排索引的邻接表(adjacency table)，其每行都对应一个网页，并按照其对应的整数大小来排序。
- 任一网页p对应的行中包含的也是一系列整数的排序结果，每个整数对应的是链向p的网页编号。这张邻接表允许应答类似于“哪些网页指向p？”的查询。
- 以同样的方法，可以建立所有p所指向的网页的邻接表。

小结：连接服务器

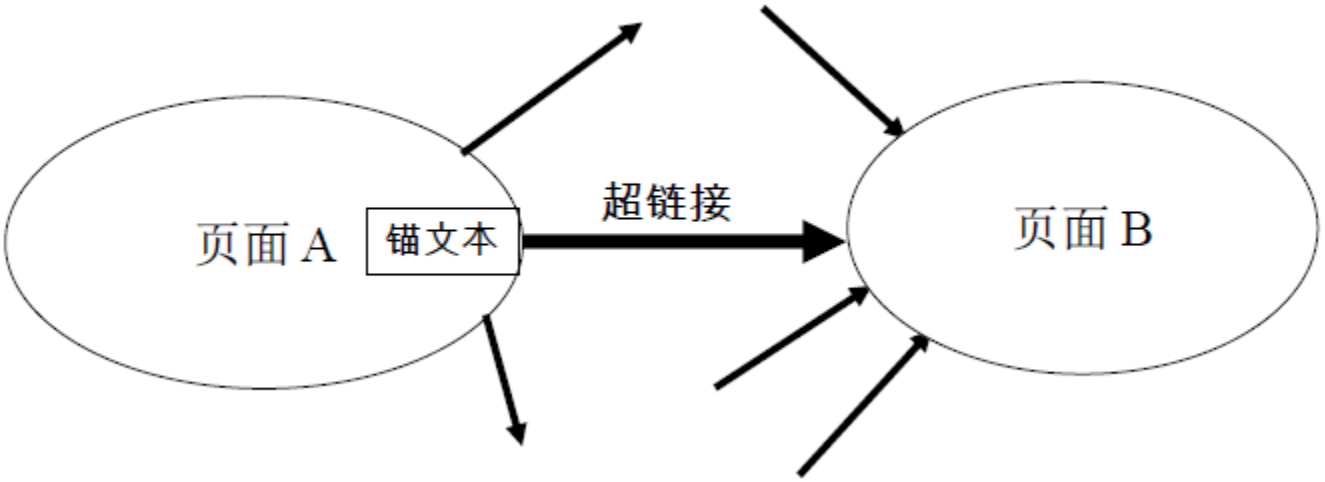
- 哪些URL指向给定的URL？
- 给定的URL指向哪些URL？
- 类似于倒排索引的邻接表

目录

- Web搜索基础
  - Web与文档集的不同
  - 近似重复检测
- Web采集
  - 采集器
  - 连接服务器
- 链接分析
  - 锚文本
  - 链接分析：Pagerank
  - 链接分析：HITS

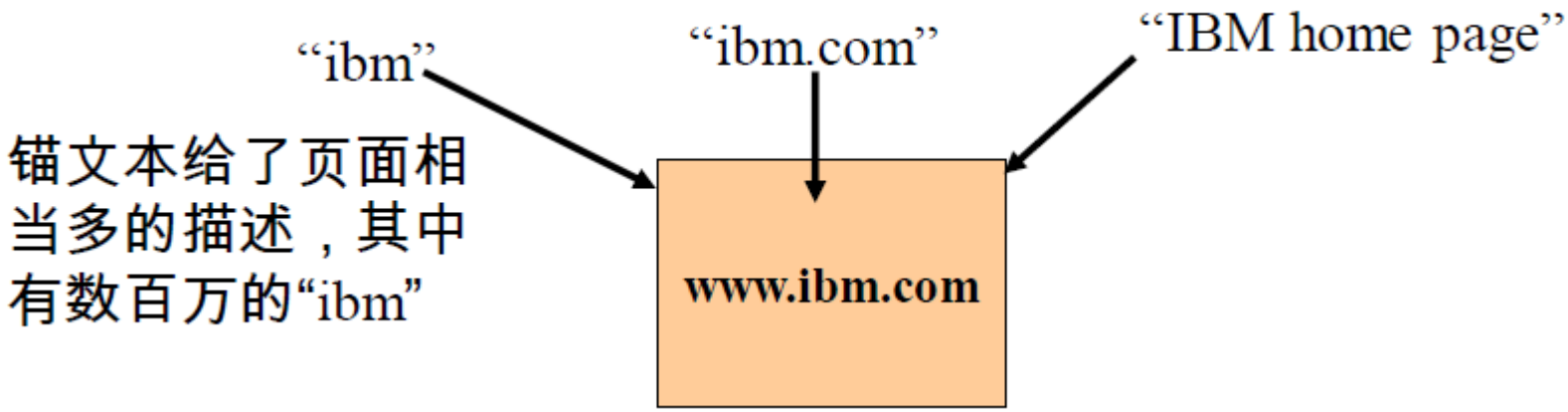
Web是有向图

- 假设**1:** *A*到*B*的超链接表示*A*的作者对*B*的认可
- 假设**2:** 指向页面*B*的锚文本是对*B*的一个很好的描述



锚文本

- 超链接周围还有一些文本，这些文本通常被嵌在<a>标签(称为锚)中
- 对于IBM如何在如下三者间进行辨别
  - IBM's 主页(基本上都是图片)
  - IBM's 版权声明页('ibm' 词频很高)
  - 竞争对手的垃圾信息页面(任意高词频)



- 中文网站锚文本举例
- 雷暴（Thunderstorms）是伴有雷击和闪电的局地对流性天气。

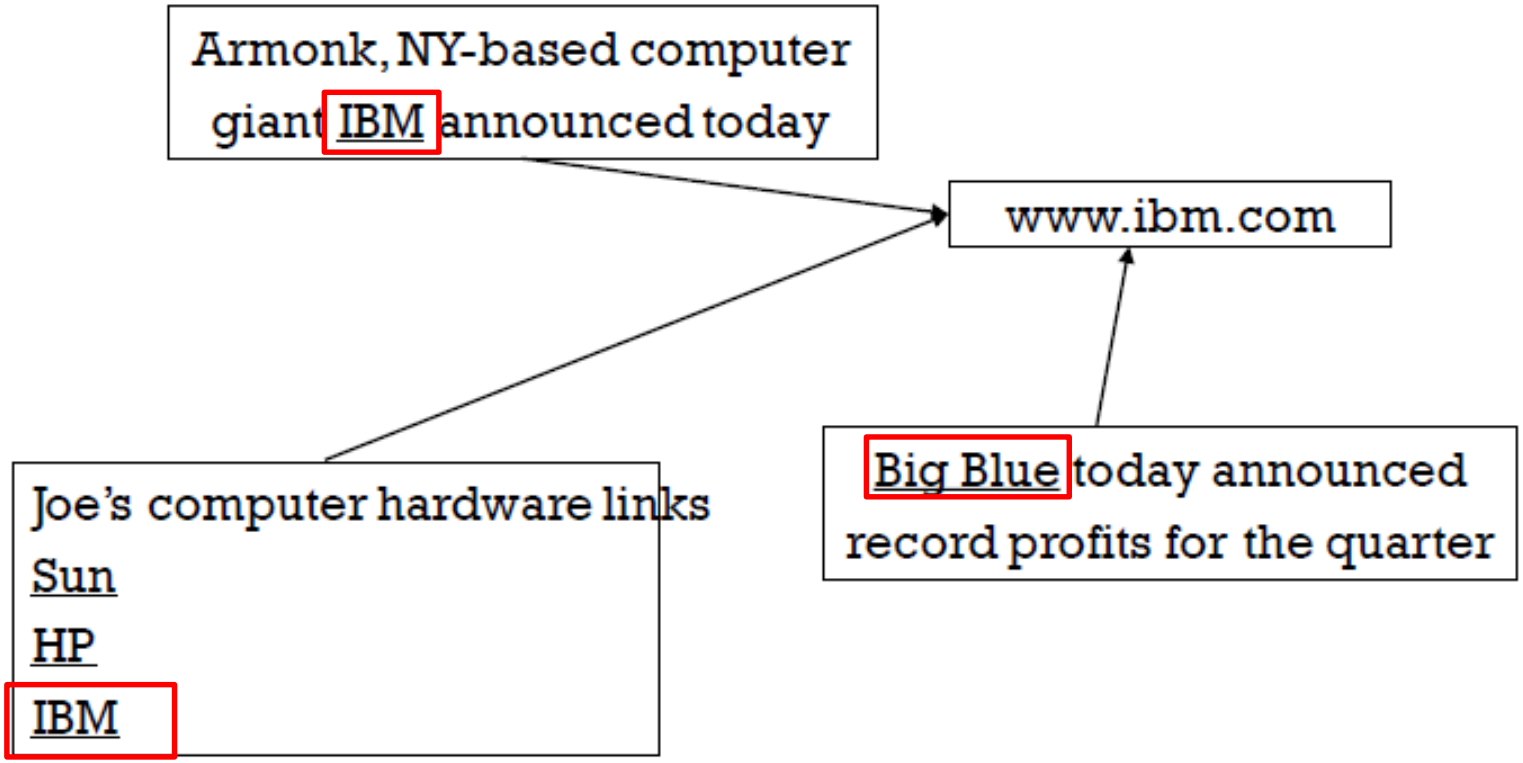
雷击

Sohu.com	京ICP证030367号
Baidu.com	京ICP证030173号
g.cn	ICP证合字B2-20070004号
cntv.cn	京ICP证060535号
xinhuanet.com	京ICP证010042号

www.miibeian.gov.cn

索引锚文本

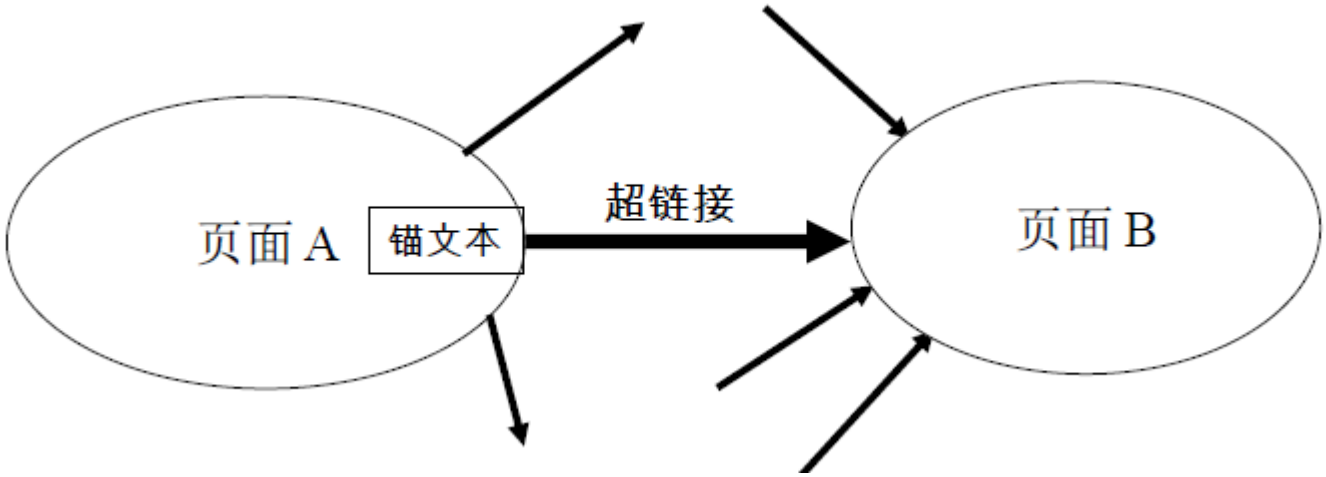
- 在索引文档*D*的时候，也索引指向文档*D*的锚文本。



- 锚文本的使用有时候会产生一些有趣的副作用-e.g.,Big Blue
  - 搜索Big Blue时会出现IBM主页，但是主页里面是没有Big Blue这个词的，出现的原因是很多人提到IBM的时候会使用这一绰号
- 可以根据锚文本所在页面的权威性来确定锚文本的权重
  - E.g., 我们认为cnn.com 和yahoo.com 的内容是权威的，然后就相信它们的锚文本

小结：锚文本

- Web上随处可见的一个现象是，很多网页的内容并不包含对自身的精确描述。
- 因此，**Web**搜索者不一定要使用网页中的词项来对网页进行查询，而使用锚文本。
- 锚文本周围窗口中的文本(**extended anchor text**)也可以当成锚文本一样来使用。



目录

- Web搜索基础
  - Web与文档集的不同
  - 近似重复检测
- Web采集
  - 采集器
  - 连接服务器
- 链接分析
  - 锚文本
  - 链接分析：Pagerank
  - 链接分析：HITS

Lary Page

拉里·佩奇

在 Google+ 上有 8,192,928 个关注者

劳伦斯·爱德华·“拉里”·佩奇，搜索引擎Google的创始人之一，现为Google公司的产品总监兼任CEO。 2011年一月施密特卸任CEO，拉里·佩奇接替。 维基百科

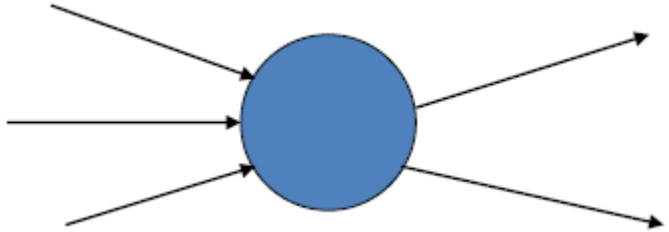
生于： 1973 年 3 月 26 日（41 岁），美国密歇根州东兰辛

身高： 1.81 米

民族	犹太人
国籍	美国
母校	密歇根大学 (B.S.) 斯坦福大学 (M.S.)

PageRank

- 对Web图中的每个节点赋一个0-1间的分值
- 查询词无关的排序
- 第一代版本
  - 使用链接的数目作为流行程度的最简单度量
- 两个基本的改进建议
  - 无向流行度
    - 赋予每个页面一个分数：即出链数 + 入链数 (3+2=5)
  - 有向流行度
    - 页面分数 = 入链数 (3)



查询处理

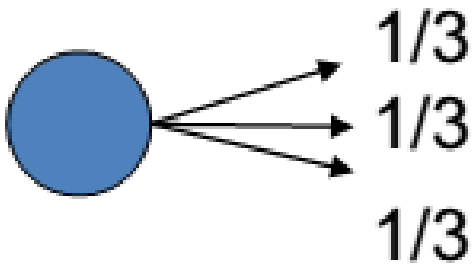
- 先检索出所有满足文本查询词的页面(例如venture capital，互联网行业非常重要的给新网站投钱的)
  - 然后把这些页面按照链接的流行度进行排序(前页的两种计算方法)
  - 更复杂的 – 把链接流行度当作静态得分，结合文本匹配的分数进行综合排序

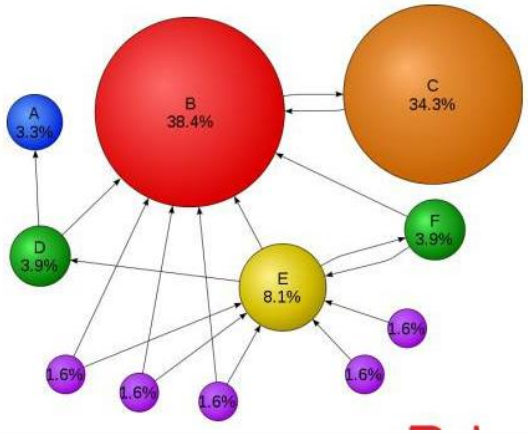
简单流行度的作弊

- 思考: 在如下两种计算方式下怎么作弊能使你的网站得分更高?
  - 无向流行度:
    - 页面分数 = 出链数 + 入链数
  - 有向流行度:
    - 页面分数 = 入链数

Pagerank打分

- 假设一个浏览者在网络上随机行走
  - 从一个随机的页面开始
  - 每一步从当前页等概率地选择一个链接，进入链接所在页面
- 在稳定状态下，每个页面都有一个访问概率 – 用这个概率作为页面的分数

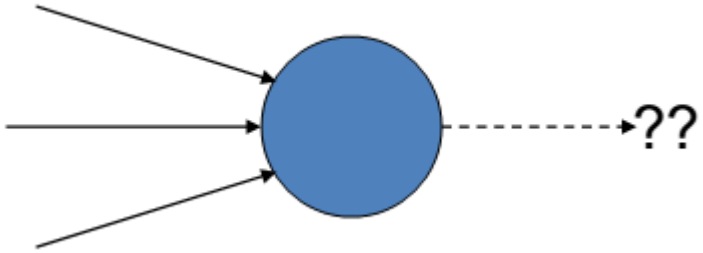




- 当冲浪者在 Web 上进行节点间的随机游走时，某些节点的访问次数会比其它节点更多。
- 直观地看，这些访问频繁的节点具有很多从其它频繁访问节点中指向的入链接。
- PageRank的思路：
  - 在随机游走过程中访问越频繁的网页越重要。

有缺陷

- 互联网上有很多Dead End
  - Dead End即网页不存在出链，那该怎么办？
  - 这样就没法计算长期游走情况下的访问概率了！



随机跳转(Teleporting)

- 遇到dead end时
  - 随机跳转到一个页面，如果网页总数是N，那么随机跳转的概率是$1/N$
- 在非dead end时
  - 以α (值较小，一般10%或20%)的概率跳转到一个随机页面
  - 以1- α的概率从页面的出链中选择一个
- 随机跳转的结果
  - 不会再困在一个地方
  - 将会有一个比率表示所有网页在长期的情况下被访问的概率

那么怎么计算这个概率呢？

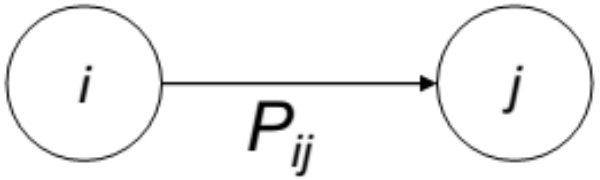
PageRank

- 当冲浪者采用这种混合过程（随机游走 + 随机跳转）时，他就会以一个稳定的概率 $\pi(v)$访问每个节点 v，其中 $\pi(v)$依赖于
  - (i) Web 图的结构；
  - (ii) α 的值。
- 称 $\pi(v)$为 v 的 **PageRank**值
- 将采用马尔科夫链(离散时间随机过程discrete-time stochastic process)理论来说明

Markov链

- 一个Markov链有 N 个状态(N个Web网页)，以及一个 $N \times N$ 的**转移概率矩阵 P**
- 每一步，只能处在一个状态
- $1 <= i, j <= N$，转移概率矩阵的P_{ij} 给出了从状态i到下一个状态j的条件转移概率
- **P**中每一行的元素之和为1，即从该页面跳转到所有出链的概率之和是1

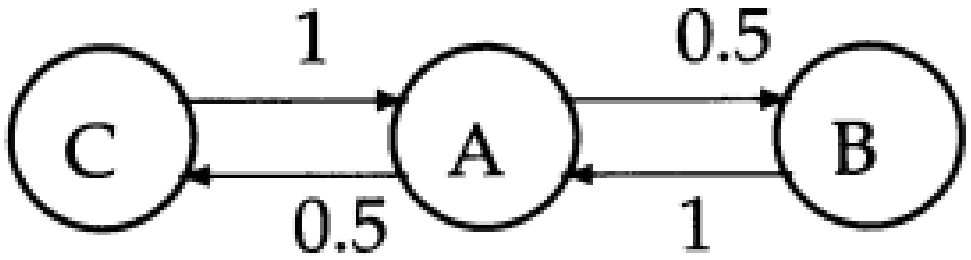
$$\forall i, \sum_{j=1}^{N} P_{ij} = 1$$



- 满足上述性质的非负矩阵被称为随机矩阵(Stochastic Matrix)。重要性质：最大特征值是1，与该特征值对应的有一个主左特征向量(Principal Left Eigenvector)

- 马尔科夫链中，下一个状态的分布仅仅依赖于当前的状态，而和如何到达当前状态无关
- 该马尔科夫链的转移概率矩阵**P**为：

$$\begin{pmatrix} 0 & 0.5 & 0.5 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$



概率向量

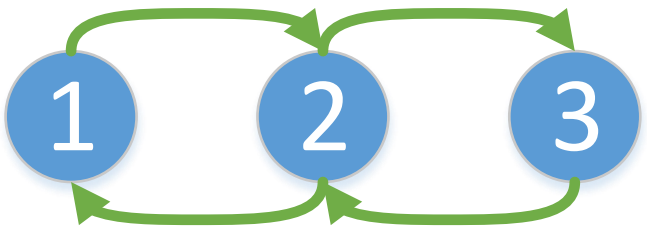
- 马尔科夫链的状态概率分布可以看成一个概率向量(probability vector)，其中的每个元素都在[0,1]之间，并且所有的元素之和为1(一行)。
- 如果一个N维的概率向量的每个分量对应马尔科夫链中的一个状态，那么该向量就可以被看成是在状态上的一个概率分布。(一行)
- 将Web图上的一个随机冲浪过程看成是马尔科夫链，其中的每个状态对应一个网页，而每个转移概率代表从一个网页跳转到另外一个网页的概率
- 一个概率(行)向量$\vec{x} = (x_1, \ldots, x_N)$代表随机游走到哪一个地方

$$\sum_{j=1}^{N} x_j = 1$$

邻接矩阵A→概率转移矩阵P

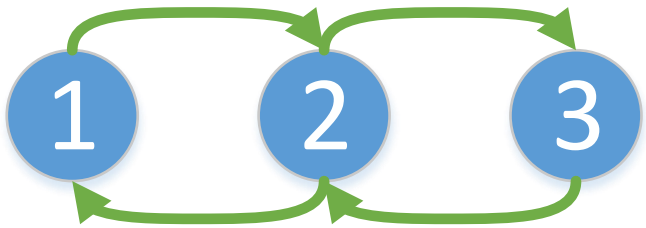
- Web图的邻接矩阵**A**可以如下定义：如果存在网页i到网页j的一条链接，那么$A_{ij}=1$，否则 $A_{ij}=0$。

$$A=\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \end{array}$$



  - 如果某一行没有1(即没有出链)，则用$1/N$代替每个元素(随机选择其它任一网页)。
  - 其它行的处理如下
    - 用每行中的1的个数去除每个1，因此如果某行有3个1，则每个1用1/3代替(归一化)；
    - 上面处理后的结果矩阵乘以$1-\alpha$；
    - 对上面得到的矩阵中的每个元素都加上$\alpha * 1/N$。

$$A = \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



$$P = (1-\alpha)\begin{bmatrix} 0 & 1 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \end{bmatrix} + \frac{\alpha}{N} \overset{\alpha=0.5}{=} \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} \\ 0 & \frac{1}{2} & 0 \end{bmatrix} + \frac{0.5}{3} = \begin{bmatrix} \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \frac{5}{12} & \frac{1}{6} & \frac{5}{12} \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{bmatrix}$$

概率向量的变化

- 在当前这一步的概率向量是$\vec{\mathbf{x}} = (x_1, \ldots, x_N)$那么下一步的概率向量是多少?
- 回想一下，转移概率矩阵$\mathbf{P}$告诉我们在状态i如何转移到其它状态
- 下一步的概率向量就是 $\vec{\mathbf{x}}\mathbf{P}$
- 两步之后$\vec{\mathbf{x}}\mathbf{P}^2$, 然后$\vec{\mathbf{x}}\mathbf{P}^3$ ， ….
- “最终” 意味着当k很大时, $\vec{\mathbf{x}}\mathbf{P}^k = \vec{\pi}$
- 最终访问频率收敛于固定的、稳态概率 $\vec{\pi}$

遍历Markov链

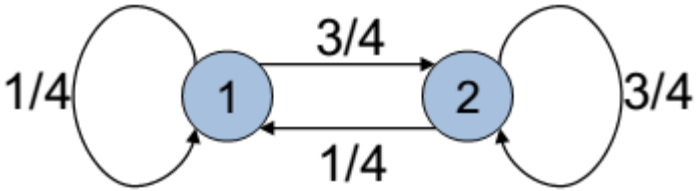
- 满足遍历性的必要条件：

  - 不可约(irreducibility)，任意两个状态之间都存在非零概率转移序列
  - 对任意的初始状态，经过有限时间T_0的跳转后，在$T>T_0$时刻处于其它任意状态的概率都大于0
  - 非周期性：不存在两个状态子集之间的循环往复
- 对任意的遍历Markov链，都存在一个唯一的稳态概率向量$\vec{\pi}$，它是**P**的主左特征向量。
  - 稳态概率分布
  - 该访问率是与起点无关的
  - $\pi(i)\ \ i=1...N$ 是结点i的稳态概率

计算a的一种方法：幂迭代(power iteration)

- 回忆一下，不管从什么地方开始，最终都会到达稳定状态$\vec{\pi}$
- 从任意分布开始 (例如 $\vec{x} = (1, \ldots, 0)$)
- 一步之后，到达 $\vec{x}\mathbf{P}$
- 两步之后 $\vec{x}\mathbf{P}^2$, 然后 $\vec{x}\mathbf{P}^3$ ， ....
- “最终” 意味着当k很大时，$\vec{x}\mathbf{P}^k = \vec{\pi}$
- 算法: 给 $\vec{x}$ 乘上 $\mathbf{P}$ 的k次方，k不断增加，直到乘积看起来已经稳定 (比如比较 $\vec{x}\mathbf{P}^k$ 与$\vec{x}\mathbf{P}^{k-1}$ 之间的差值)

稳态概率



- 稳态看起来就像一个概率向量 $\vec{\pi}=(\pi_1,...,\pi_N)$
  - π_i 就是在状态 i 的概率
  - 在上面这个例子中, $\pi_1=\frac{1}{4}$ and $\pi_2=\frac{3}{4}$
- 怎么计算稳态概率？
  - 假设 $\vec{\pi}=(\pi_1,...,\pi_N)$ 表示稳态概率
  - 如果我们当前状态是 $\vec{\pi}$ ，那么下一步的分布应该是 $\vec{\pi}\mathbf{P}$
  - 因为已经是稳态了，所以 $\vec{\pi}\mathbf{P}=\vec{\pi}$
    - 解矩阵等式可以得到 $\vec{\pi}$
    - $\vec{\pi}$ 是**P** 的主左特征向量(对应于**P**最大特征值的特征向量)
    - 转移概率矩阵的最大特征值是1

PageRank小结

- 预处理
  - Web图→邻接矩阵**A**→概率转移矩阵 **P**
  - 由**P**计算 $\vec{\pi}$
  - 元素 π_i 是一个0和1之间的数: 即页面 i 的PageRank
- 查询处理
  - 检索满足查询要求的页面
  - 按PageRank排序
  - 排序与查询词是无关的

几点事实

- Google确实用了Pagerank，但是它排序并不仅仅依靠PageRank
  - 还用了很多复杂的特征
  - 大量使用基于机器学习的排序
- Pagerank对于爬虫的爬取策略还是很有用的

目录

- Web搜索基础
  - Web与文档集的不同
  - 近似重复检测
- Web采集
  - 采集器
  - 连接服务器
- 链接分析
  - 锚文本
  - 链接分析：Pagerank
  - 链接分析：HITS

Jon Kleinberg

乔恩·克莱因伯格

乔恩·克莱因伯格是美国计算机科学家，康奈尔大学计算机科学教授，2006年获得国际数学联盟颁发的奈望林纳奖。学生昵称他为“反叛王”。克莱因伯格以解决重要而且实际的问题并能够从中发现深刻的数学思想而著称。他的研究跨越了从计算机网络路由到数据挖掘到生物结构比对等诸多领域。 维基百科

生于： 1971 年 10 月，美国马萨诸塞州波士顿

教育背景： 麻省理工学院 (1996 年)， 康乃尔大学 (1993 年)

所获奖项： 奈望林纳奖， 麦克阿瑟奖

超链导向的主题搜索

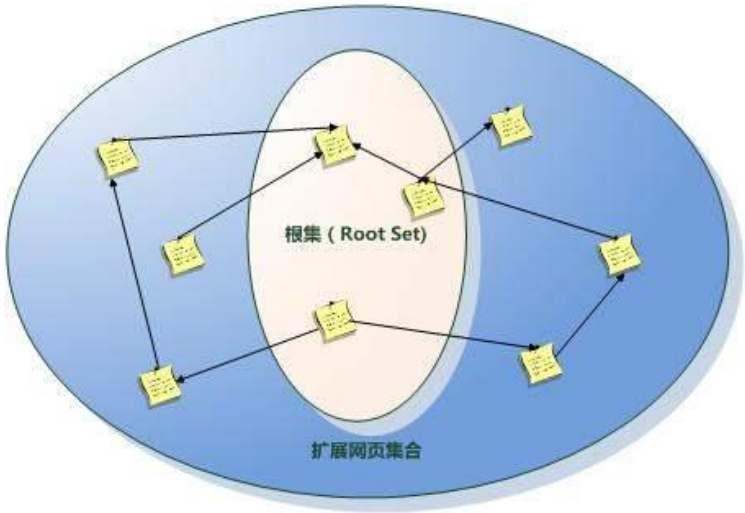
- Hyperlink- Induced Topic Search (HITS)
- 对每个网页给出两个得分：一个得分被称为hub值，另外一个被称为authority值
- 作为查询的响应，与排序过的相关网页列表不同，可以找到两个互相联系的页面集合
  - 导航页Hub page很好的指向某一主题的列表
    - e.g., “Bob’s list of cancer-related links.”
  - 权威页Authority page在针对某一主题的好Hub页中经常出现
- 相对于寻找特定页面，更加适合于泛主题搜索
  - E.g., wish to learn about leukemia(白血病)

导航Hubs 和权威Authorities

- 针对某一主题的好Hub页会指向很多关于这个主题的Authority页面
- 关于某一主题的好Authority页面会被很多针对这一主题的好Hub页指向
- 循环定义Circular definition – 导致可以迭代求解页面的Hub值和Authority值

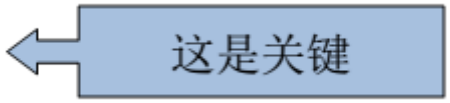
HITS步骤：确定基本集

- 给一个查询词 (如 browser)，使用一个文本索引取出所有包含 browser 的页面称为**根集合**.
- 再在根集合中添加满足下面任一要求的页面
  - 指向根集合中的一个页面
  - 被根集合中的一个页面指向的页面
- 得到的集合称为基本集 **base set**.



HITS步骤：精选出Hub页和Authority页

- 对于基本集中的每一个页面x计算Hub分$h(x)$和Authority分$a(x)$
- 初始化：所有的 x, $h(x)$←1; $a(x)$←1;
- 迭代更新 $h(x)$, $a(x)$;

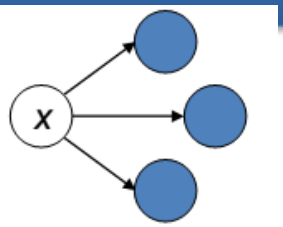


- 迭代之后
  - 输出具有最高 $h()$ 的页面作为Top Hub页
  - 最高 $a()$ 的页面作为Top Authority页

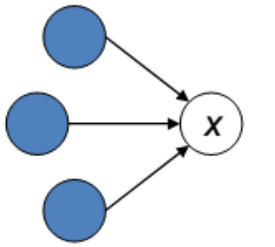
HITS步骤：迭代更新

- 对所有 x 重复如下步骤:

$$h(x) \leftarrow \sum_{x \mapsto y} a(y)$$



$$a(x) \leftarrow \sum_{y \mapsto x} h(y)$$



- 先重新计算所有网页的hub值，接着根据更新后的hub 值又来计算所有网页的authority值
- 接着又根据更新后的authority值重新计算所有网页的hub值，如此可以反复迭代下去
- 为了避免*h*()和*a*()太大，每次迭代之后都可以按一定比例缩小。缩放并不会影响最后结果：因为只关心相对的分数

应该迭代多少次?

- 宣称: 迭代一些次数后分数会收敛
- 实际上, 适当的缩放, $h()$ 和 $a()$ 会陷入一个稳定状态!
- 只需要 $h()$ 和 $a()$ 的相对顺序，而不需要它们的值
- 实践中发现，大概5次迭代后就会稳定

小结：HITS

- 超链导向的主题搜索

  Hyperlink-Induced Topic Search (HITS)
- 对每个网页给出两个得分：一个得分被称为hub值，另外一个被称为authority值
- HITS步骤
  - 确定基本集
  - 精选出Hub页和Authority页
  - 迭代更新
- h是AA^t的特征向量，a是 A^tA的特征向量