# 回　顾

内容提要

- 时间序列及其应用：时间序列挖掘的概念
- 时间序列预测的常用方法
- 基于ARMA模型的序列匹配方法
- 基于离散傅立叶变换的时间序列相似性查找
- 基于规范变换的查找方法
- 序列挖掘及其基本方法
- AprioriAll 算法
- AprioriSome 算法
- GSP算法

# 第七章 Web挖掘技术

## 内容提要

- Web挖掘的意义
- Web挖掘的分类
- Web挖掘的含义
- Web挖掘的数据来源
- Web内容挖掘方法
- Web访问信息挖掘方法
- Web结构挖掘方法

# Web挖掘的价值

- **因特网（Web）上蕴藏着大量的信息**
  - 通过简单的浏览或关键词匹配的搜索引擎得到的是孤立而凌乱的“表面信息”，Web挖掘期望发现潜在的关联信息。
- **将Web上的丰富信息转变成有用的知识**
  - 因特网中页面内部、页面之间、超级链接、页面访问等都包含大量对用户可用的信息，而这些信息的深层次含义是很难被用户直接使用的，必须经过浓缩和提炼。
- **对用户个性化挖掘与推荐**
  - 网站信息的个性化是将来的发展趋势。通过Web挖掘，可以达到对用户访问行为、频度、内容等的分析，可以得到关于群体用户访问行为和方式的普遍知识，用以改进Web服务方的设计，提供个性化的服务。

# 第七章 Web挖掘技术

内容提要

- Web挖掘的意义
- Web挖掘的分类
- Web挖掘的含义
- Web挖掘的数据来源
- Web内容挖掘方法
- Web访问信息挖掘方法
- Web结构挖掘方法

# Web挖掘的类型

- Web挖掘依靠它所挖掘的信息来源可以分为：
  - **Web内容挖掘（Web Content Mining）**：对站点的Web页面的各类信息进行集成、概化、分类等，挖掘某类信息所蕴含的知识模式。
  - **Web访问信息挖掘（Web Usage Mining）**：Web访问信息挖掘是对用户访问Web时在服务器方留下的访问记录进行挖掘。通过分析日志记录中的规律，可以识别用户的忠实度、喜好、满意度，可以发现潜在用户，增强站点的服务竞争力。
  - **Web结构挖掘（Web Structure Mining）**：Web结构挖掘是对Web页面之间的链接结构进行挖掘。在整个Web空间里，有用的知识不仅包含在Web页面的内容之中，而且也包含在页面的链接结构之中。对于给定的Web页面集合，通过结构挖掘可以发现页面之间的关联信息，页面之间的包含、引用或者从属关系等。
- Web其它分类
  - 从数据源上
  - 从知识模式上

# 第七章 Web挖掘技术

内容提要

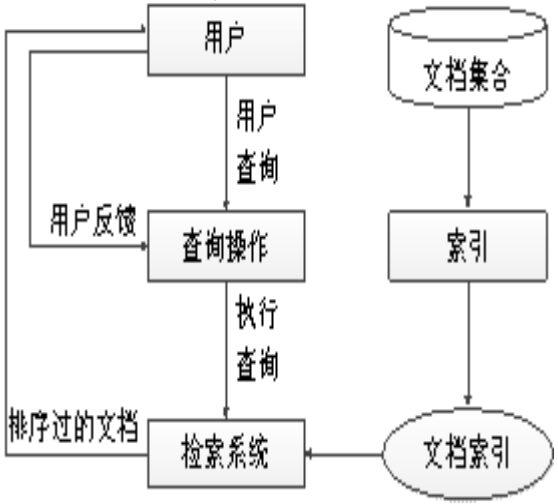
- Web挖掘的意义
- Web挖掘的分类
- Web挖掘的含义
- Web挖掘的数据来源
- Web内容挖掘方法
- Web访问信息挖掘方法
- Web结构挖掘方法

# Web挖掘的含义

- WeB挖掘是一个宽泛的概念，可以简单地描述为：
  - Web挖掘是数据挖掘在Web上的应用，它利用数据挖掘技术从与WWW相关的资源和行为中抽取感兴趣的、有用的模式和隐含信息。
  - 针对Web页面内容、页面之间的结构、用户访问信息、电子商务信息等各种Web数据，应用数据挖掘方法以帮助人们从因特网中提取知识，为访问者、站点经营者以及包括电子商务在内的基于因特网的商务活动提供决策支持。



- 信息检索（Information Retrieval，IR）是搜索的根基，其目的是帮助用户从大规模的文本文档中找到所需信息的研究领域。




用户查询的形式分为：关键词查询、布尔查询、短语查询、临近查询、全文搜索、自然语言查询。

- 信息检索可能经常被说成是Web挖掘的初级阶段，是为了强调Web挖掘不是简单的信息索引或关键词匹配技术，而是实现信息浓缩成知识的过程，它可以支持更高级的商业决策和分析。

# 第七章 Web挖掘技术

内容提要

- Web挖掘的意义
- Web挖掘的分类
- Web挖掘的含义
- Web挖掘的数据来源
- Web内容挖掘方法
- Web访问信息挖掘方法
- Web结构挖掘方法

# Web挖掘的主要数据源

- Web挖掘的数据来源是宽泛的。凡是在Web站点中对用户有价值的数据都可以成为它挖掘的数据源。但是由于差距很大，挖掘的策略和方法存在很大差异。
- 由于这些对象的数据形式及含义的差异，其挖掘技术会不同。一些比较有代表性的数据源有：
  - Web服务器日志数据
  - Web上的电子商务数据
  - Web上的网页
  - Web上的网页之间的链接
  - Web上的多媒体数据

# 服务器日志数据

- 对Web服务器的访问，服务器方将会产生三种日志文件：
  - Server logs：记录用户的访问时间、IP地址以及请求等信息。
  - Error logs：存取请求失败的数据，例如丢失连接、授权失败或超时等
  - Çookie logs：Cookie是由web服务器产生的记号并由客户端持有，用于识别用户和用户的会话。

Server logs的一个格式示意

| Field | Description |
|---|---|
| Date | Date， time， and timezone of request |
| Client IP | Remote host IP and / or DNS entry |
| User name | Remote log name of the user |
| Bytes | Bytes transferred (sent and received) |
| Server | Server name， IP address and port |
| Request | URI query and stem |
| Status | http status code returned to the client |
| Service name | Requested service name |
| Time taken | Time taken for transaction to complete |
| Protocol version | Version of used transfer protocol |
| User agent | Service provider |
| Cookie | Cookie ID |
| Referrer | Previous page |
| … | … |

# 在线市场数据

- 在线市场数据是指和市场活动相关的信息。例如：
  - 交易数据
  - 用户注册等用户数据
  - 商品数据

- 从内容上说，不同目的商务网站有不同的商务信息。但是，这类数据通常是用传统的关系数据库结构来存储数据。

- 在线市场数据是业务数据，是进行业务相关分析的主体。用户的挖掘目标只有结合在线市场数据分析才能达到目的。

# Web页面

- Web页面是网站信息的主体，但是它们的主要信息不可能像关系型数据库那样规整，因此Web页面的内容组织形式的分析是研究Web挖掘的具体方法的基础。
- 目前的Web页面满足HTML或者XML标准，因此可以利用这些标记语言格式实施挖掘。
- 1998年WWW社团提出了XML语言标准（eXtensible Markup Language）。该标准通过把一些描述页面内容的标记（tag）添加到HTML页面中，用于对HTML页面内容进行自描述。基于XML规范的挖掘研究也是一个重要的研究分支。

# Web页面链接

- Web网站是页面的集合，而且是有结构的。页面之间的链接在支持页面的转换和推进。
- Web页面链接形成网站的页面结构，而这种结构是分析网站设计合理性或者对用户的友好性的基础。
  - 设计合理性：网站的脆弱性，如一个网页坏掉其它就无法进入？
  - 用户方便性：对常用网页的“链接次数”
- Web网站的网页结构用户不可能预先知道，而且和用户的点击习惯、兴趣爱好、环境等有关。

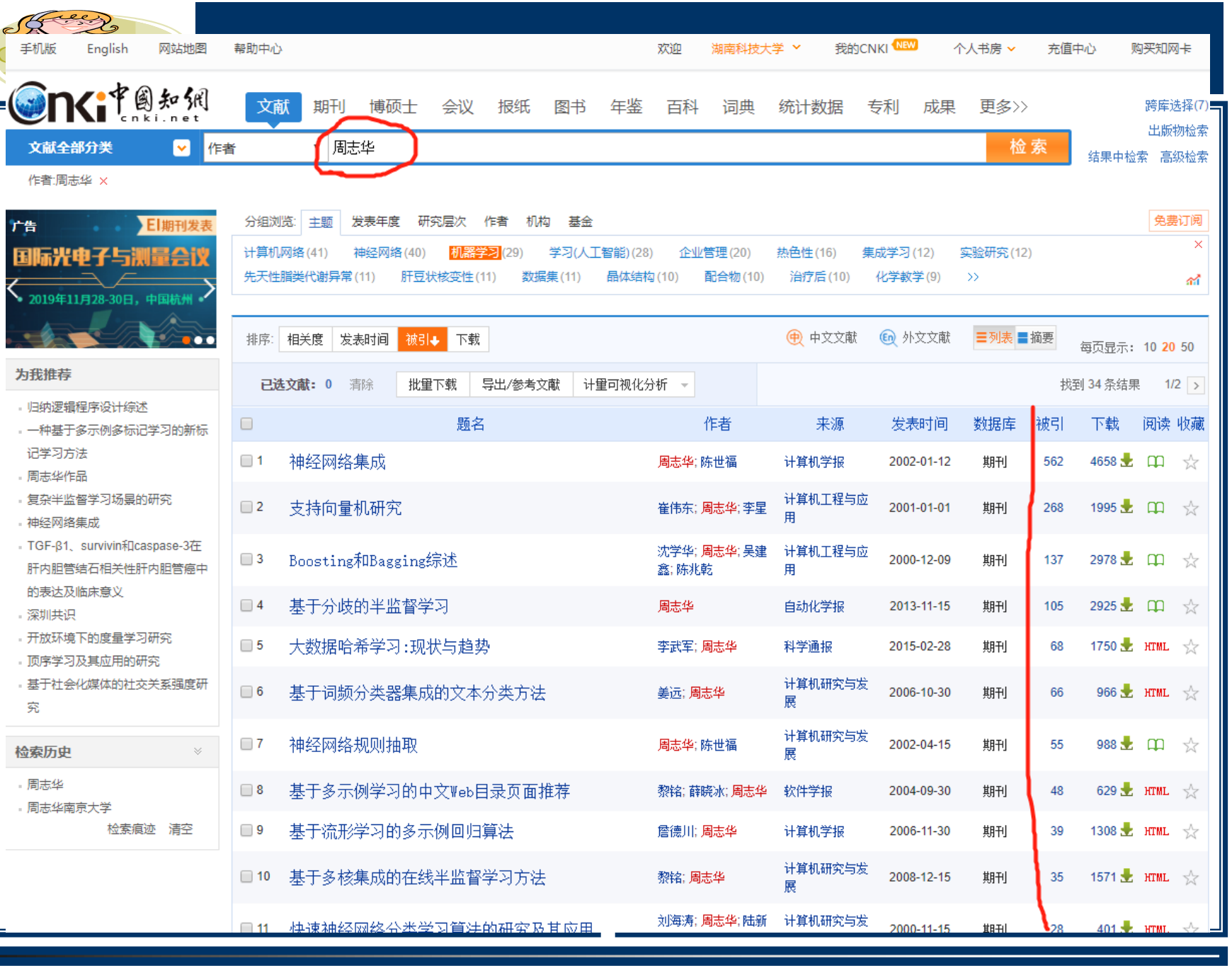
# 第七章 Web挖掘技术

## 内容提要

- Web挖掘的意义
- Web挖掘的分类
- Web挖掘的含义
- Web挖掘的数据来源
- Web内容挖掘方法
- Web访问信息挖掘方法
- Web结构挖掘方法

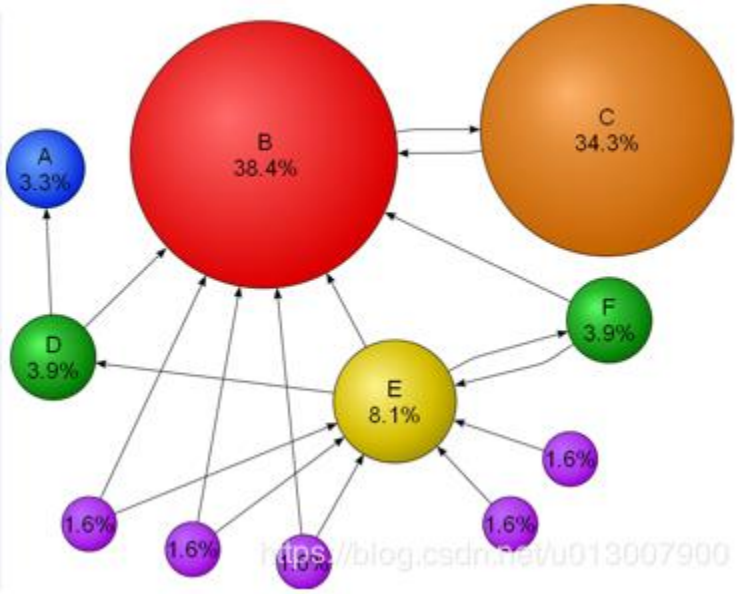
# 页面分级方法-PageRank

- 谷歌的两位创始人，当时还是美国斯坦福大学 (Stanford University) 研究生的佩奇 (Larry Page) 和布林 (Sergey Brin) 借鉴了学术界评判学术论文重要性的通用方法，那就是看论文的引用次数。由此想到网页的重要性也可以根据这种方法来评价。于是PageRank的核心思想就诞生了。


手机版 English 网站地图 帮助中心
欢迎 湖南科技大学 我的CNKI NEW 个人书房 充值中心 购买知网卡
cnki.net
文献 期刊 博硕士 会议 报纸 图书 年鉴 百科 词典 统计数据 专利 成果 更多>>
跨库选择(7)
出版物检索
文献全部分类
作者
周志华
检索
结果中检索 高级检索
作者:周志华
广告
EI期刊发表
国际光电子与测量会议
2019年11月28-30日，中国杭州
分组浏览: 主题 发表年度 研究层次 作者 机构 基金
免费订阅
计算机网络(41) 神经网络(40) 机器学习(29) 学习(人工智能)(28) 企业管理(20) 热色性(16) 集成学习(12) 实验研究(12)
先天性脂类代谢异常(11) 肝豆状核变性(11) 数据集(11) 晶体结构(10) 配合物(10) 治疗后(10) 化学教学(9) >>
排序: 相关度 发表时间 被引 下载
中文文献 外文文献 列表 摘要
每页显示: 10 20 50
已选文献: 0 清除
批量下载 导出/参考文献 计量可视化分析
找到 34 条结果 1/2
为我推荐
归纳逻辑程序设计综述
一种基于多示例多标记学习的新标记学习方法
周志华作品
复杂半监督学习场景的研究
神经网络集成
TGF-β1、survivin和caspase-3在肝内胆管结石相关性肝内胆管癌中的表达及临床意义
深圳共识
开放环境下的度量学习研究
顶序学习及其应用的研究
基于社会化媒体的社交关系强度研究
检索历史
周志华
周志华南京大学
检索痕迹 清空
题名 作者 来源 发表时间 数据库 被引 下载 阅读 收藏
1 神经网络集成 周志华; 陈世福 计算机学报 2002-01-12 期刊 562 4658
2 支持向量机研究 崔伟东; 周志华; 李星 计算机工程与应用 2001-01-01 期刊 268 1995
3 Boosting和Bagging综述 沈学华; 周志华; 吴建鑫; 陈兆乾 计算机工程与应用 2000-12-09 期刊 137 2978
4 基于分歧的半监督学习 周志华 自动化学报 2013-11-15 期刊 105 2925
5 大数据哈希学习:现状与趋势 李武军; 周志华 科学通报 2015-02-28 期刊 68 1750 HTML
6 基于词频分类器集成的文本分类方法 姜远; 周志华 计算机研究与发展 2006-10-30 期刊 66 966 HTML
7 神经网络规则抽取 周志华; 陈世福 计算机研究与发展 2002-04-15 期刊 55 988
8 基于多示例学习的中文Web目录页面推荐 黎铭; 薛晓冰; 周志华 软件学报 2004-09-30 期刊 48 629 HTML
9 基于流形学习的多示例回归算法 詹德川; 周志华 计算机学报 2006-11-30 期刊 39 1308 HTML
10 基于多核集成的在线半监督学习方法 黎铭; 周志华 计算机研究与发展 2008-12-15 期刊 35 1571 HTML
11 快速神经网络分类学习算法的研究及其应用 刘海涛; 周志华; 陆新 计算机研究与发 2000-11-15 期刊 28 401 HTML

# PageRank的核心思想

- 如果一个网页被很多其他网页链接到的话说明这个网页比较重要，也就是PageRank值会相对较高
- 如果一个PageRank值很高的网页链接到一个其他的网页，那么被链接到的网页的PageRank值会相应地因此而提高

# 页面分级方法

- 页面分级借鉴了学术引文分析思想。
  - 在每一篇学术论文的结尾处都会有参考文献列表，这些参考文献主要用来告诉读者，该篇学术论文参考或者引用的论文集。
  - 一篇学术论文被引用次数越高，越能够说明该篇学术论文的参考价值越大。
  - 网页之间的超链接关系与参考文献引用有很多相似的地方，如果页面A上存在指向页面B的链接地址，就可以看作是页面A对页面B的一次引用。

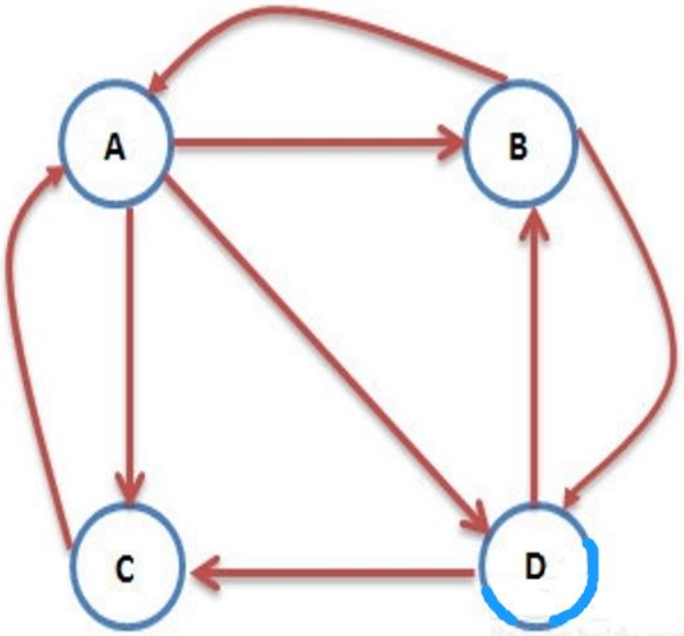
- 定义7-3 设u为一个Web页，$F_u$为所有u指向的页面（扇出）的集合，$B_u$为所有指向u的页面（扇入）的集合。设c（<1）为一个调整参数，那么u页面的PageRank被定义为：

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

# 简单pagerank模型

- 互联网中的网页的链接可以看作是有向图，如A、B、C、D四个网页。

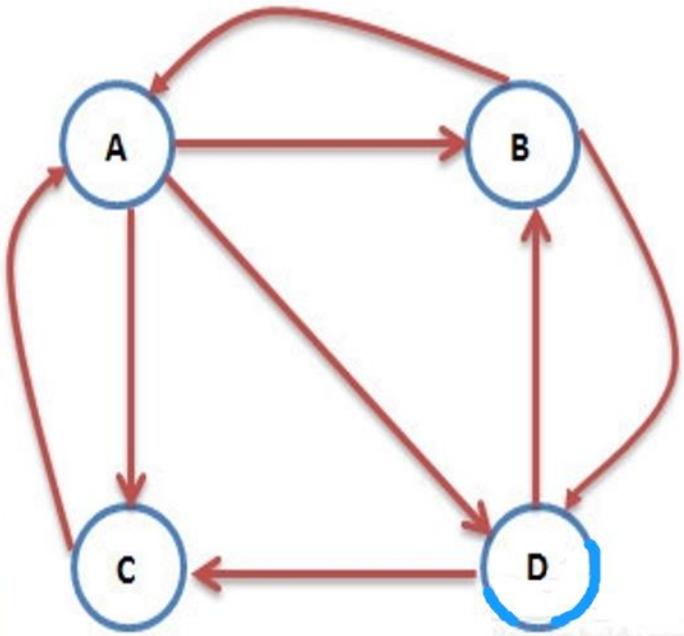
| | 入度 | 出度 |
|---|---|---|
| A | 2 | 3 |
| B | 2 | 2 |
| C | 2 | 1 |
| D | 2 | 2 |




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

转移矩阵

# 简单pagerank模型




## 转移矩阵

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

初始时，假设上网者在每一个网页的概率都是相等的，则初试的概率分布就是一个所有值都为1/n的n维列向量V0

$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}$$

# 简单pagerank模型

- 初始向量V0

$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}$$

转移矩阵

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

- 用V0去右乘转移矩阵M，得到第一步之后上网者的概率分布向量V1，

$$V_1=MV_0=\begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}=\begin{bmatrix} 9/24 \\ 5/24 \\ 5/24 \\ 5/24 \end{bmatrix}$$

# 简单pagerank模型

- 用V1去右乘转移矩阵M，得到第二步之后上网者的概率分布向量V2，

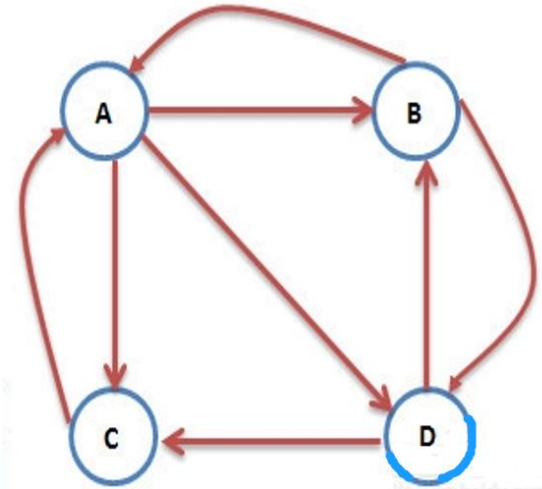
$$\begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 9/24 \\ 5/24 \\ 5/24 \\ 5/24 \end{bmatrix} = \begin{bmatrix} 15/48 \\ 11/48 \\ 11/48 \\ 11/48 \end{bmatrix}$$

- 不断迭代直到稳定：

$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} \begin{bmatrix} 9/24 \\ 5/24 \\ 5/24 \\ 5/24 \end{bmatrix} \begin{bmatrix} 15/48 \\ 11/48 \\ 11/48 \\ 11/48 \end{bmatrix} \cdots \begin{bmatrix} 11/32 \\ 7/32 \\ 7/32 \\ 7/32 \end{bmatrix} \cdots \begin{bmatrix} 3/9 \\ 2/9 \\ 2/9 \\ 2/9 \end{bmatrix}$$

# PageRank算法

- 简单来说分为两步：
- 1、计算每个网页一个PageRank（PR）值
- 2、通过（投票）算法不断迭代，直至达到平稳分布为止。
- PageRank算法：计算每一个网页的PageRank值，然后根据这个值的大小对网页的重要性进行排序。
-

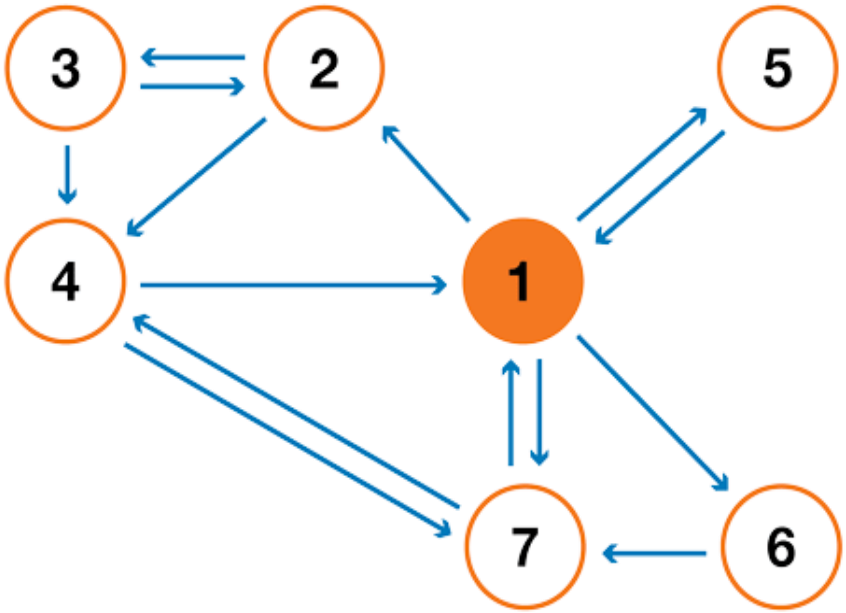



$$\begin{bmatrix} 3/9 \\ 2/9 \\ 2/9 \\ 2/9 \end{bmatrix}$$

A、B、C、D

# 终止点问题

- 所有的网页浏览都是收敛的吗？
- 要满足收敛性，需要具备一个条件：图是强连通的，从任意网页可以到达其他任意网页，即形成一个马尔科夫的过程。



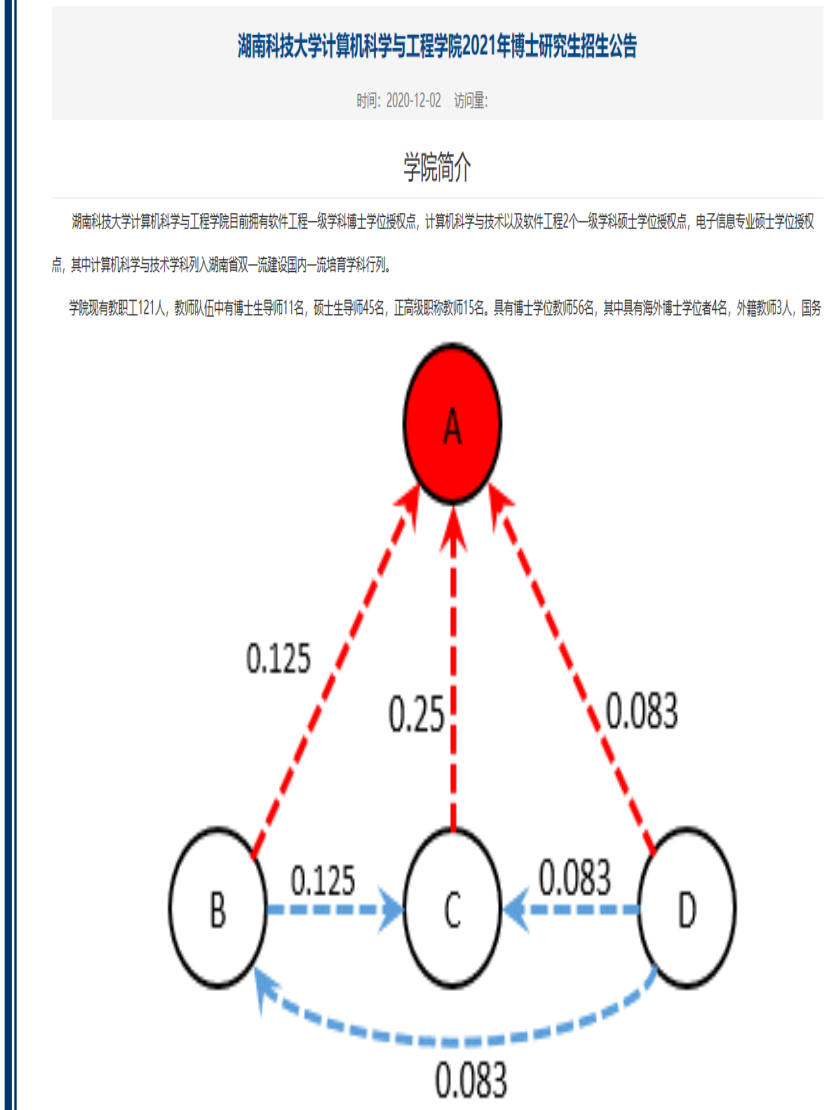

Trajectory： 1

# 页面等级值设定的垂直链接问题

假如某些节点只有入度而没有出度的话，在等级值的流动分配过程中，这些只有入度的页面会不断的积累其他页面传递过来的等级值，造成等级值的滞留。

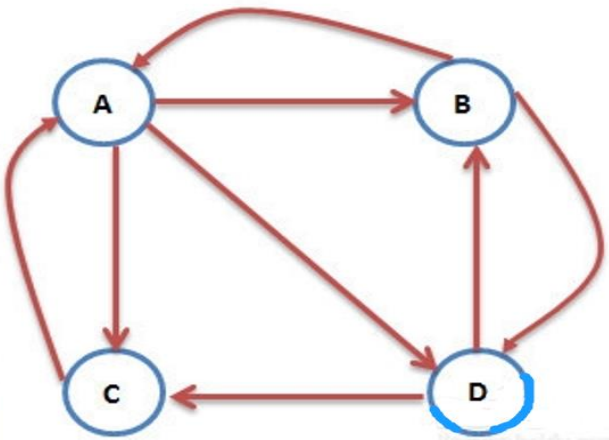
这种只拥有入度却没有出度的的页面被称为垂悬链接。

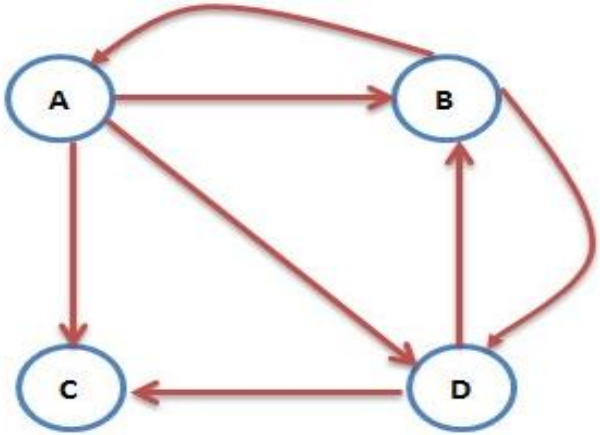
图7-5就是一个存在悬垂链接的例子。



# 简单pagerank模型

- 假设把上例中C到A的链接丢掉，C变成了一个终止点，得到下面这个图：



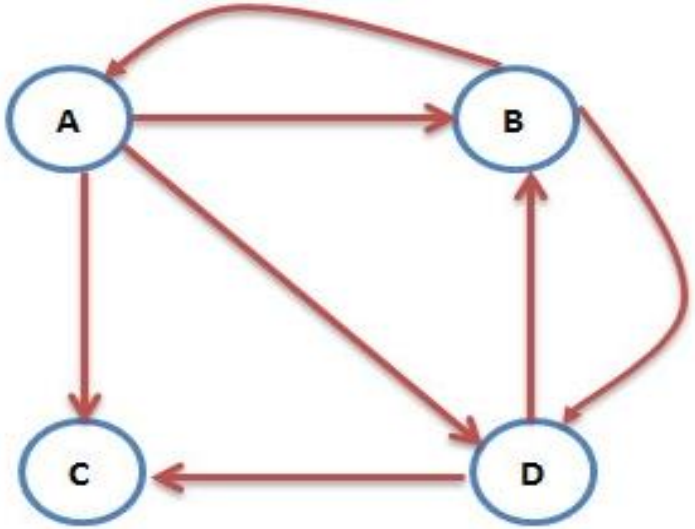

$$M=\begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$

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

# 简单pagerank模型

- 假设把上例中C到A的链接丢掉，C变成了一个终止点，得到下面这个图：
- 转移矩阵为：

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




- 连续迭代下去，最终所有元素都为0：
-

$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} \begin{bmatrix} 3/24 \\ 5/24 \\ 5/24 \\ 5/24 \end{bmatrix} \begin{bmatrix} 5/48 \\ 7/48 \\ 7/48 \\ 7/48 \end{bmatrix} \begin{bmatrix} 21/288 \\ 31/288 \\ 31/288 \\ 31/288 \end{bmatrix} \cdots \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

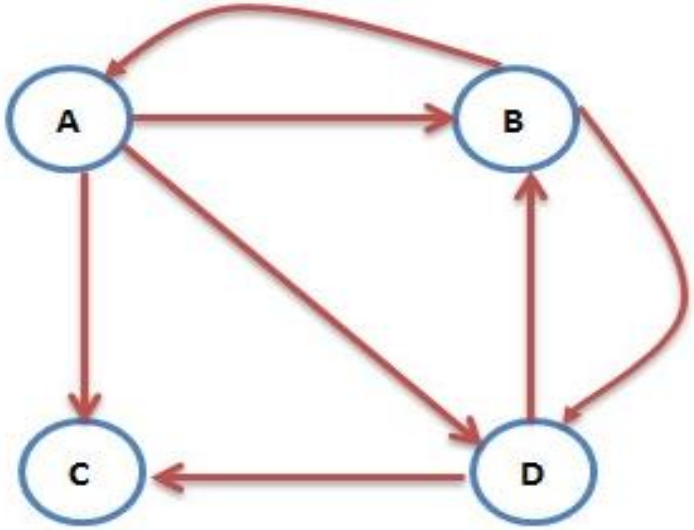
# 随机冲浪模型

- 上网者不是点击连接，而是在地址栏输入另外一个地址，而在地址栏输入跳转到各个网页的概率是1/n。假设上网者每一步查看当前网页的概率为a，那么他从浏览器地址栏跳转的概率为(1-a)，于是原来的迭代公式转化为：

$$V' = \alpha MV + (1-\alpha)e$$

$$V_1 = \alpha M V_0 + (1-\alpha)e = 0.8 \times \begin{bmatrix} 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 1 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} + 0.2 \times \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 9/60 \\ 13/60 \\ 25/60 \\ 13/60 \end{bmatrix}$$

# 随机冲浪模型




$$V' = \alpha MV + (1-\alpha)e$$

$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} \begin{bmatrix} 9/60 \\ 13/60 \\ 25/60 \\ 13/60 \end{bmatrix} \begin{bmatrix} 41/300 \\ 53/300 \\ 153/300 \\ 53/300 \end{bmatrix} \begin{bmatrix} 15/4500 \\ 707/4500 \\ 2543/4500 \\ 707/4500 \end{bmatrix} \cdots \begin{bmatrix} 15/148 \\ 19/148 \\ 95/148 \\ 19/148 \end{bmatrix}$$

# PageRank算法-描述

**算法7-3** 基于随机冲浪的PageRank算法

输入：页面链接网络$G$

输出：页面等级值向量$R$

(1) 根据页面链接网络$G$生成移转概率矩阵$M$;

(2) 设点击概率$d$; 等级值向量初始$R_0$; 迭代终止条件 $\varepsilon$;

(3) $i$=1;

(4) Repeat

(5) 计算$R_{i+1}=M*R_i$;

(6) 计算$|R_{i+1}-R_i|$; 两个向量的逐分量和

(7) Until $|R_{i+1}-R_i| < \varepsilon$;

(8) 输出$R_{i+1}$, 作为最终等级值向量.

# 链接分析算法：HITS算法

- HITS（Hyperlink-Induced Topic Search）是遵照寻找权威页面和中心页面的典型方法，基于一组给定的关键字，可以找到相关的页面。
- 所谓权威页面是指包含需求信息的最佳资源页面。是指与某个领域或者某个话题相关的高质量网页，比如搜索引擎领域，Google和百度首页即该领域的高质量网页，比如视频领域，优酷和土豆首页即该领域的高质量网页。



# 链接分析算法：HITS算法

- HITS（Hyperlink-Induced Topic Search）是遵照寻找权威页面和中心页面的典型方法，基于一组给定的关键字，可以找到相关的页面。
- 所谓中心页面是一个包含权威页面连接的页面。比如hao123首页

# 权威页面和中心页面

- HITS算法的目的即是通过一定的技术手段，在海量网页中找到与用户查询主题相关的高质量“Authority”页面和“Hub”页面，尤其是“Authority”页面，因为这些页面代表了能够满足用户查询的高质量内容，搜索引擎以此作为搜索结果返回给用户。

- 算法基本思想：相互增强关系
  - 基本假设1：一个好的“Authority”页面会被很多好的“Hub”页面指向；
  - 基本假设2：一个好的“Hub”页面会指向很多好的“Authority”页面；
  - 进行多轮迭代计算，每轮迭代计算更新每个页面的两个权值，直到权值稳定不再发生明显的变化为止。

# 权威页面和中心页面

**算法7-3 HITS**

输入：把www看作一个引导图W；查询请求q；支持s。

输出：权威页面的集合A；中心页面的集合H。

（1）BEGIN

（2） $R=SE(W, q)$；//利用q得到页面的根集合$R$

（3） $B= R \cup$ {指向$R$的连接}$\cup${来自$R$的连接}；

（4） $G(B, L)=$ 由$B$导出的W的子图；

（5） $G(B, L1)$=删除G中相同站点的连接；

（6） $x_p=\sum_q Y_q$；// $<q,p> \in L1$，得到权威页面的权重；

（7） $y_p=\sum_q X_q$；// $< q,p > \in L1$，得到中心页面的权重；

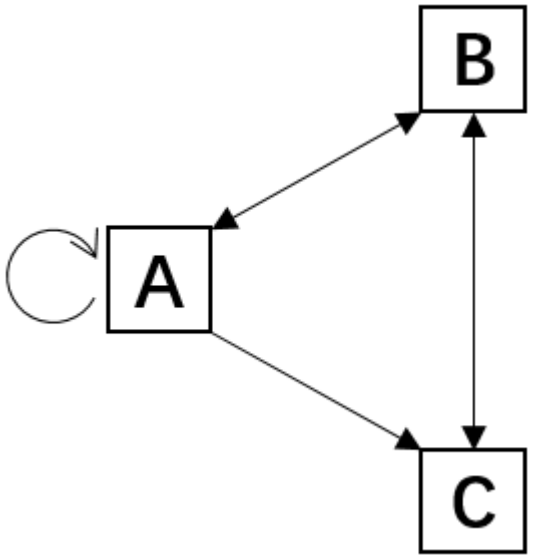
（8） $A=\{p|p$为具有最高$x_p$值的页面$\}$；

（9） $H=\{p|p$为具有最高$y_p$值的页面$\}$；

（10）END

# 实例

- 有三个网页A，B，C及其链接：
- 




阵构造邻接矩阵

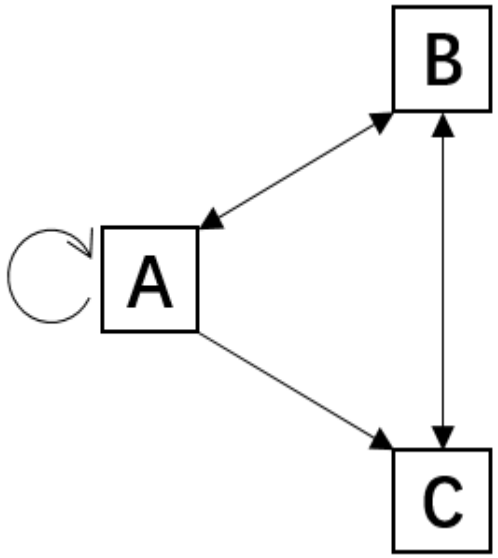
$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

- 每个节点都有一个Hub分数和
  以有一个Hub向量h和Authori
  个元素都初始化为1 √‾ n，其

$$h_0 = \begin{bmatrix} \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \end{bmatrix}, \quad a_0 = \begin{bmatrix} \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \end{bmatrix}$$

# 实例

- 有三个网页A，B，C及其链接：
- 阵构造邻接矩阵




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

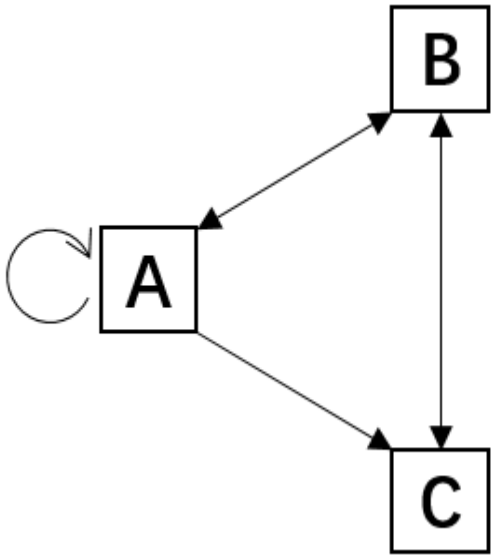
$$\mathbf{h}_0 = \begin{bmatrix} \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \end{bmatrix}, \quad \mathbf{a}_0 = \begin{bmatrix} \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \end{bmatrix}$$

$$\mathbf{h}_1 = A\mathbf{a}_0$$

$$\mathbf{a}_2 = A^T\mathbf{h}_1$$

# 实例

- 有三个网页A，B，C及其链接：
- 交替更新hh和aa的值：




$$\mathbf{h}_1 = A\mathbf{a}_0$$
$$\mathbf{a}_2 = A^T\mathbf{h}_1$$

| 0 | 1 | 2 | 3 | 4 | … |
|---|---|---|---|---|---|
| $h_0$ | $h_1$ | | $h_3$ | | … |
| $a_0$ | | $a_2$ | | $a_4$ | … |

# HITS算法与PageRank算法比较

- HITS算法和PageRank算法可以说是搜索引擎链接分析的两个最基础且最重要的算法。
- 1.HITS算法是与用户输入的查询请求密切相关的，而PageRank与查询请求无关。
- 2.HITS算法必须在接收到用户查询后实时进行计算，计算效率较低；而PageRank则可以在爬虫抓取完成后离线计算，在线直接使用计算结果，计算效率较高；
- 3.HITS算法的计算对象数量较少，只需计算扩展集合内网页之间的链接关系；而PageRank是全局性算法，对所有互联网页面节点进行处理；

# HITS算法与PageRank算法比较

- 5.HITS算法存在主题泛化问题，所以更适合处理具体化的用户查询；而PageRank在处理宽泛的用户查询时更有优势；
- 6.HITS算法在计算时，对于每个页面需要计算两个分值，而PageRank只需计算一个分值即可；
- 7.从链接反作弊的角度来说，PageRank从机制上优于HITS算法，而HITS算法更易遭受链接作弊的影响。
- 8.HITS算法结构不稳定，当对“扩充网页集合”内链接关系作出很小改变，则对最终排名有很大影响；而PageRank相对HITS而言表现稳定，其根本原因在于PageRank计算时的“远程跳转”。

# Web访问信息的一些概念

- W3C国际组织已经为Web访问信息定义了一些基本概念：
  - **定义7-4** 用户（User）：用户被定义为一个通过浏览器访问一个或者多个Web服务器的访问者。一个用户可以通过几台PC机或者使用多个浏览器来访问，因此识别用户是任务之一。
  - **定义7-5** 页面文件（Page File）：一个页面文件是通过HTTP请求发给用户的文件。页面文件有静态的和动态的，动态页面文件由Web服务器动态生成响应用户的请求。
  - **定义7-6** 页面视图（Page View）：一个页面视图由一个集合的页面文件组成，页面视图通常与一个用户的行为相关（如一次鼠标点击）。由框架（frame）、图片、和script等组成。
  - **定义7-7** 客户端浏览器（Client Browser）：是指具有一个独立IP地址的，用户通过其访问Web服务器的浏览器软件。客户端包括代理服务器软件。
  - **定义7-8** Web服务器（Web Server）：是指运行在互联网服务提供方主机上的WWW服务软件，目的是响应客户端发来的HTTP请求。
  - **定义7-9** 点击流（Click Stream）：亦称连续HTTP请求序列。

# Web访问信息的一些概念

- W3C国际组织已经为Web访问信息定义了一些基本概念：
  - **定义7-10** 一次访问用户（One User at a Time）：是指某一个通过一个客户端浏览器发出连续HTTP请求序列的对一个Web服务器进行访问的访问者。如果一个真实的用户每隔一段较长的时间对一个Web服务器发出一个连续HTTP请求序列，那么对该Web服务器而言就有多个一次访问用户进行了访问。
  - **定义7-11** 用户访问会话（User Session）：是指由一个用户发出的对Web世界的一次连续HTTP请求序列。
  - **定义7-12** 服务器用户访问会话（Server Session）：简称用户访问事务（User Transaction）是指一次访问用户的对一个Web服务器的一次访问。由该一次访问用户所请求的页面序列顺序组成。
  - **定义7-13** 访问片断（Episode）：任何有意义的用户访问会话或用户访问事务的子集，被称为访问片断。

# Web站点的结构的对象描述

- 一个Web站点的拓扑结构$M$：

$$M =< P, L_{PageViewLink}, L_{PageLink} >$$

  - 其中$P$为所有页面视图$Page_View$的集合：

$$P = \{Page\_View_1, Page\_View_2, \cdots, Page\_View_n\}$$

    - **一个页面视图*Page_View*由一组框架*Frame*组成：**

$$Page\_View_i = \{Frame_{i1}, Frame_{i2}, \cdots, Frame_{im}\}$$

      - 每个框架由一个页面文件组成：

$$Frame_j = \{Page\_File_h\}$$

  - $L_{pageLink}$为所有页面的链接的集合：

$$L_{PageLink} = \{PageLink_1, PageLink_2, \cdots, PageLink_q\}$$

  - $L_{pageViewLink}$为全部页面视图之间超链关系的集合：

$$L_{PageViewLink} = \{PageViewLink_1, PageViewLink_2, \cdots, PageViewLink_p\}$$

# Web站点结构的预处理

- 通过相应的搜索算法对Web网站进行遍历以找到PageLink，PageViewSet，PageViewLink的集合。
- 算法7-4：生成PageViewSet和PageViewLink算法

**算法7-4** GPVS（Generating Page View Set）
输入： index.htm。
输出： PageViewSet， PageViewLinkSet。

```
（1）PageViewSet← GetFirstPageVeiw(/index.htm);
（2）PageViewLinkSet←NULL;
（3）FOR each pageview ∈ PageViewSet DO BEIGIN
（4）      PageSet← GetAllPage(pageview);
（5）      FOR each p ∈ PageSet DO BEIGIN//每个pageview由一些页面组成
（6）          LinkSet← GetAllHyperLink(p);
（7）          FOR each l∈LinkSet DO BEIGIN //l必须为站点内的地址
（8）              newpageview← Substitute(pageview， l); //根据超链得到一个新的pageview
（9）              PageViewSet← PageViewSet ∪ {newpageview};
（10）             PageViewLinkSet←PageViewLinkSet∪{<pageview，newpageview>};
（11）         END
（12）     END
（13）END. // PageViewSet 集合增量递增，每次从PageViewSet集合中变量pageview只取新的值
```

# 第七章 Web挖掘技术

## 内容提要

- Web挖掘的意义
- Web挖掘的分类
- Web挖掘的含义
- Web挖掘的数据来源
- Web内容挖掘方法
- Web访问信息挖掘方法
- Web结构挖掘方法

# Web访问信息挖掘的特点

- **Web访问数据容量大、分布广、内涵丰富和形态多样**
  - 一个中等大小的网站每天可以记载几兆的用户访问信息。
  - 广泛分布于世界各处。
  - 访问信息形态多样。
  - 访问信息具有丰富的内涵。

- **Web访问数据包含决策可用的信息**
  - 每个用户的访问特点可以被用来识别该用户和网站访问的特性。
  - 同一类用户的访问，代表同一类用户的个性。
  - 一段时期的访问数据代表了群体用户的行为和群体用户的共性。
  - Web访问信息数据是网站的设计者和访问者进行沟通的桥梁。
  - Web访问信息数据是开展数据挖掘研究的良好的对象。

# Web访问信息挖掘的意义

- 通过分析日志文件，可以发现用户访问页面的特征、页面被用户访问的规律、用户频繁访问的页组等，以便其合理、有效地优化站点的结构，最终为用户提供一个方便快捷信息获取环境。

- 有三方面的应用具有代表性：
  - Web服务方主要根据自己的领域知识设计Web页面的结构，而群体用户根据各自的访问兴趣访问这些页面，那么服务方的结构设计是否合理？怎样的设计以便利于群体用户的访问，更加吸引访问者？这些问题的解决是Web访问信息挖掘的主要目的。
  - 群体用户的访问存在哪些特点？如果掌握了这些特点，那么就可以利用其开展进一步的商务活动。
  - 对于每一个新的Web站点的访问者，都会在曾经访问的群体用户中找到一些最相似的相同的访问者，那么那些访问者的访问就可以给这个新的访问者提供推荐，以便利于该访问者的进一步访问。

# Web访问信息挖掘的数据源

- 由于Web世界的分布性，用户访问行为被广泛地分布记录在Web服务器、用户客户端，和代理服务器中。在各个分布地点的不同的用户访问信息表征了不同类型的用户访问行为。挖掘工作必须针对数据的特点来决定相应的挖掘任务。用户访问信息的分布简单归结为：
  - 服务器方：一般地，在一个Web服务器上，服务器日志记录了多个用户对单个站点的用户访问行为。
  - 客户方：一般地，在客户端计算机上，客户端的代理记录了单个用户对单个站点或单个用户对多个站点的用户访问行为。客户端的Cache记录了用户访问内容。客户端的BookMark也记录了单个用户对单个站点的访问偏好。
  - 客户端代理服务器：代理服务器记录了多个用户对多个站点的访问行为，同时代理服务器内部的Cache记录了多个用户对多个站点的访问内容。

# 服务器方访问信息

- 一个Web服务器日志（Server log）反映出多个用户对单个站点的访问行为。
- 一个从实际Web服务器上采集的Log文件片段：

| IP Address | User ID | Time | Method/URI/Protocol | Stauts | Size |
|---|---|---|---|---|---|
| 159.226.219.52 | - - | 10/Dec/1998:12:34:16 -0600 | "GET /images/lchzhi.gif HTTP/1.1" | 200 | 44851 |
| 159.226.219.52 | - - | 10/Dec/1998:12:34:32 -0600 | "GET /graduate.htm HTTP/1.1" 200 | 7403 | |
| 159.226.219.52 | - - | 10/Dec/1998:12:34:32 -0600 | "GET /images/sxwys2.jpg HTTP/1.1" | 200 | 18481 |
| 203.141.89.99 | - - | 10/Dec/1998:12:34:48 -0600 | "GET /result.htm HTTP/1.0" 200 | 12302 | |
| 159.226.219.52 | - - | 10/Dec/1998:12:34:58 –0600 | "GET /structure.htm HTTP/1.1" 200 | 367 | |
| 159.226.219.52 | - - | 10/Dec/1998:12:34:58 –0600 | "GET /struc-index.htm HTTP/1.1" 200 | 4370 | |
| 159.226.219.52 | - - | 10/Dec/1998:12:34:58 –0600 | "GET /struc-content.htm HTTP/1.1" | 200 | 12047 |
| 159.226.219.52 | - - | 10/Dec/1998:12:34:58 –0600 | "GET /images/znkfsys.jpg HTTP/1.1" | 200 | 22574 |

# 代理服务器端访问信息

- 代理服务器端的访问信息包括用户访问日志和在Cache中被访问的页面信息。
- 一个代理服务器日志的例子（基于WindowsNT4.0的代理服务器）：

200.121.2.88， HEAD\SWANG， Mozilla/4.0 (compatible; MSIE 4.0; Windows 95)， Y， 99-3-28， 15:57:44， W3Proxy， NTPROXY， -， www.ict.ac.cn， 159.226.39.2， 80， 200， 582， 1376， http， tcp， GET， http://www.ict.ac.cn/cjc/cjcw2.html ， -， Inet， 304， 0

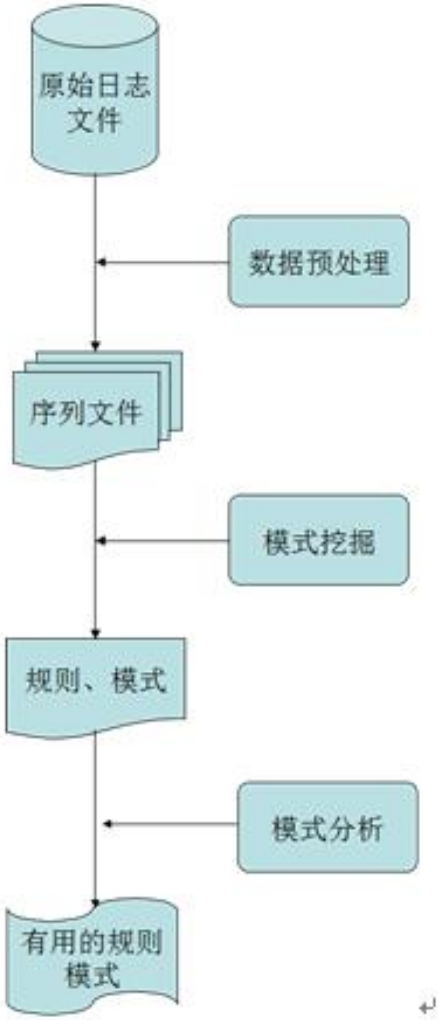
200.121.2.88， HEAD\SWANG， Mozilla/4.0 (compatible; MSIE 4.0; Windows 95)， Y， 99-3-28， 15:57:44， W3Proxy， NTPROXY， -， www.ict.ac.cn， 159.226.39.2， 80， 270， 2101， 1254， http， tcp， GET， http://www.ict.ac.cn/cjc/introc.html ， -， VCache， 304， 0

200.121.2.88， HEAD\SWANG Mozilla/4.0 (compatible; MSIE 4.0; Windows 95)， Y， 99-3-28， 15:57:44， W3Proxy， NTPROXY， -， www.ict.ac.cn， 159.226.39.2， 80， 171， 449， 1110， http， tcp， GET， http://www.ict.ac.cn/cjc/star.gif， - ， Inet， 304， 0

200.121.2.88， HEAD\SWANG， Mozilla/4.0 (compatible; MSIE 4.0; Windows 95)， Y， 99-3-28， 15:57:44， W3Proxy， NTPROXY ， - ， www.ict.ac.cn ， 159.226.39.2 ， 80 ， 211 ， 455 ， 826 ， http ， tcp ， GET ， http://www.ict.ac.cn/cjc/INTROCG.JPG， -， Inet， 304， 0

# Web访问信息挖掘的过程

- Web访问信息挖掘的研究和应用体系



# Web访问信息挖掘的预处理

- Web访问信息挖掘的基础和最烦琐的工作是数据的预处理。预处理用户访问信息是整个数据准备的核心工作，也是开展下一阶段Web访问信息挖掘的基础。
- 预处理阶段主要的工作包括：
  - **数据清洗：**由于数据表示、写入的对象差异以及用户的兴趣和挖掘算法对数据要求的不同，对于Web日志中的数据需要确定合理的数据清洗策略 。
  - **识别用户**
  - 划分**访问事务**和**访问片断**。

# 数据清洗

- 合并数据：在给定挖掘时间段后，数据清洗需要合并Web服务器上的多个日志文件，并且解析每个文件，将其转化到数据库或特定格式的数据文件中。
- 剔除不相关的数据：在Web日志中一些存取记录可能对挖掘来说是不必要的，例如图形文件，压缩文件等的存取可能对面向文本挖掘的用户不需要考虑，所以应该被剔除。通过检查后缀gif、jpeg、zip、ps等就可以实现。
- 代理访问的处理：由于搜索引擎或其他一些自动代理的存在，日志中存在大量的由它们发出的请求。因此从日志中识别代理（Agent）或网络爬虫（Crawler or Spider）对站点的访问是必须的。
- 正规化URI（Uniform Resource Identifier）：由于各种默认情况的存在，需要进一步正规化URI。
- 数据项解析：CGI数据项必须被解析在不同的域中，并被解析为<名字，值>对的形式。

# 识别用户

- Web上的用户需要识别吗？
  - 在Web日志等文件中，访问IP是一定有的，但是并不能直接对用到用户：
    - 不同的用户可能使用同一个计算机或代理服务器去访问Web服务器。
    - 不同的IP地址可能代表同一个用户。
    - 同一个用户有可能会使用不同的代理服务器或者在计算机。
- 如何识别用户？很困难，“IP+环境”有一定可行性。

| Time | IP | URL | Ref | Envir. |
|---|---|---|---|---|
| 0:01 | 1.2.3.4 | A | - | IE5;Win2k |
| 0:09 | 1.2.3.4 | B | A | IE5;Win2k |
| 0:10 | 2.3.4.5 | C | - | IE6;WinXP;SP1 |
| 0:12 | 2.3.4.5 | B | C | IE6;WinXP;SP1 |
| 0:15 | 2.3.4.5 | E | C | IE6;WinXP;SP1 |
| 0:19 | 1.2.3.4 | C | A | IE5;Win2k |
| 0:22 | 2.3.4.5 | D | B | IE6;WinXP;SP1 |
| 0:22 | 1.2.3.4 | A | - | IE6;WinXP;SP2 |
| 0:25 | 1.2.3.4 | E | C | IE5;Win2k |
| 0:25 | 1.2.3.4 | C | A | IE6;WinXP;SP2 |
| 0:33 | 1.2.3.4 | B | C | IE6;WinXP;SP2 |
| 0:58 | 1.2.3.4 | D | B | IE6;WinXP;SP2 |
| 1:10 | 1.2.3.4 | E | D | IE6;WinXP;SP2 |
| 1:15 | 1.2.3.4 | A | - | IE5;Win2k |
| 1:16 | 1.2.3.4 | C | A | IE5;Win2k |
| 1:17 | 1.2.3.4 | F | C | IE6;WinXP;SP2 |
| 1:26 | 1.2.3.4 | F | C | IE5;Win2k |
| 1:30 | 1.2.3.4 | B | A | IE5;Win2k |
| 1:36 | 1.2.3.4 | D | B | IE5;Win2k |

User 1

| | | | |
|---|---|---|---|
| 0:01 | 1.2.3.4 | A | - |
| 0:09 | 1.2.3.4 | B | A |
| 0:19 | 1.2.3.4 | C | A |
| 0:25 | 1.2.3.4 | E | C |
| 1:15 | 1.2.3.4 | A | - |
| 1:26 | 1.2.3.4 | F | C |
| 1:30 | 1.2.3.4 | B | A |
| 1:36 | 1.2.3.4 | D | B |

User 2

| | | | |
|---|---|---|---|
| 0:10 | 2.3.4.5 | C | - |
| 0:12 | 2.3.4.5 | B | C |
| 0:15 | 2.3.4.5 | E | C |
| 0:22 | 2.3.4.5 | D | B |

User 3

| | | | |
|---|---|---|---|
| 0:22 | 1.2.3.4 | A | - |
| 0:25 | 1.2.3.4 | C | A |
| 0:33 | 1.2.3.4 | B | C |
| 0:58 | 1.2.3.4 | D | B |
| 1:10 | 1.2.3.4 | E | D |
| 1:17 | 1.2.3.4 | F | C |

# 用户识别技术介绍

| 方法 | 解释 | 优点 | 缺点 |
| --- | --- | --- | --- |
| 客户或代理IP地址 | 假定一个IP对应一用户。 | 客户端不增功能；简单。 | 适应性差 |
| 嵌入会话ID | 服务器方可跟踪返回的ID来区分用户。 | 可以识别一次访问用户，屏蔽代理服务。 | Web服务器上增加会话识别功能。 |
| 用户注册 | 注册才能进行访问。 | 精度最高。 | 用户适用受限。 |
| Cookie | Cookie功能使每个客户端有一个标识符。 | 可以识别一次访问用户，而且能够跟踪。 | 客户端关闭Cookie功能就没法应用。 |
| 客户端发送 | 在客户端放入软件，当用户浏览页面是就发送用户信息到服务端。 | 精度较高 | 需要用户允许装入相应的软件工具才能使用。 |
| 修改的浏览器 | 由被增强的浏览器发送回用户的浏览信息到服务端。 | 精度较高 | 需要修改或者更新客户端浏览器软件 |

# 识别用户访问事务

- 用户的访问事务主要是通过考虑用户访问发生时间等来界定。

**定义7-1** 设$L$为用户访问日志，其中的一个项$l \in L$包括用户的IP地址$l.ip$，用户的标识符$l.uid$，被存取页的URI地址$l.url$，长度为$l.length$以及存取访问的时间$l.time$，存取访问的时长$l.timelength$，访问事务被定义为：

$$t =< ip_t, uid_t, \{(l_1^t.ip, l_1^t.uid, l_1^t.url, l_1^t.time, l_1^t.timelength, l_1^t.length) \\ ,..., (l_m^t.ip, l_m^t.uid, l_m^t.url, l_m^t.time, l_m^t.timelength, l_m^t.length)\} >$$

$$where, \\ for\, 1 \le k \le m,\ l_k^t \in L,\ l_k^t.ip = ip_t,\ l_k^t.uid = uid_t, \\ l_k^t.time - l_{k-1}^t.time \le C, l_{k-1}^t.timelength = l_k^t.time - l_{k-1}^t.time$$

这里$C$是一个固定的时间窗。

- 时间窗C大小的界定是一个经验值（有人建议30分钟较为合适）。

# 用户访问事务识别的例子

例如：假设C是30，前面的用户对应的事务

User 1

| | | | |
|---|---|---|---|
| 0:01 | 1.2.3.4 | A | - |
| 0:09 | 1.2.3.4 | B | A |
| 0:19 | 1.2.3.4 | C | A |
| 0:25 | 1.2.3.4 | E | C |
| 1:15 | 1.2.3.4 | A | - |
| 1:26 | 1.2.3.4 | F | C |
| 1:30 | 1.2.3.4 | B | A |
| 1:36 | 1.2.3.4 | D | B |

User 2

| | | | |
|---|---|---|---|
| 0:10 | 2.3.4.5 | C | - |
| 0:12 | 2.3.4.5 | B | C |
| 0:15 | 2.3.4.5 | E | C |
| 0:22 | 2.3.4.5 | D | B |

User 3

| | | | |
|---|---|---|---|
| 0:22 | 1.2.3.4 | A | - |
| 0:25 | 1.2.3.4 | C | A |
| 0:33 | 1.2.3.4 | B | C |
| 0:58 | 1.2.3.4 | D | B |
| 1:10 | 1.2.3.4 | E | D |
| 1:17 | 1.2.3.4 | F | C |

事务1

| | | | |
|---|---|---|---|
| 0:01 | 1.2.3.4 | A | - |
| 0:09 | 1.2.3.4 | B | A |
| 0:19 | 1.2.3.4 | C | A |
| 0:25 | 1.2.3.4 | E | C |

事务2

| | | | |
|---|---|---|---|
| 1:15 | 1.2.3.4 | A | - |
| 1:26 | 1.2.3.4 | F | C |
| 1:30 | 1.2.3.4 | B | A |
| 1:36 | 1.2.3.4 | D | B |

事务3

| | | | |
|---|---|---|---|
| 0:10 | 2.3.4.5 | C | - |
| 0:12 | 2.3.4.5 | B | C |
| 0:15 | 2.3.4.5 | E | C |
| 0:22 | 2.3.4.5 | D | B |

事务4

| | | | |
|---|---|---|---|
| 0:22 | 1.2.3.4 | A | - |
| 0:25 | 1.2.3.4 | C | A |
| 0:33 | 1.2.3.4 | B | C |
| 0:58 | 1.2.3.4 | D | B |
| 1:10 | 1.2.3.4 | E | D |
| 1:17 | 1.2.3.4 | F | C |

# 用户会话片段的识别

- 如果再进步进行做精细化的数据预处理，就要考虑用户会话划分。
- 会话或者称会话片段（Section）是指相对独立的、有挖掘价值的访问记录集合。根据W3C的标准，会话访问片断被定义为用户访问事务的有意义的子集。
- 会话识别（Session Identification）就是将用户所有的访问页面分解为一个个的会话，便于进行知识挖掘。
- 有两个成熟的技术
  - 1．导航内容片断：用户到达一个内容页之前是经历哪些导航页的。例如，一个用户访问事务为：*M*1，*M*2，*M*3，*C*1，*M*4，*M*5，*M*6，*C*2，*M*7，*M*8，*C*3，*M*9，*M*10，*M*11，*M*12，*C*4，其中*M*为导航页，*C*为内容页。识别导航内容片断就是要从用户访问事务中识别出：
    - **片断1：*M*1，*M*2，*M*3，*C*1。**
    - **片断2：*M*4，*M*5，*M*6，*C*2。**
    - **片断3：*M*7，*M*8，*C*3。**
    - **片断4：*M*9，*M*10，*M*11，*M*12，*C*4。**
  - 2．最大前向访问序列：所谓用户最大前向访问序列是指在用户访问回退之前一直被访问的页面序列。
    - **每个最大前向访问序列就构成一个访问片段。**
    - **定义该片断的优点是有利于发现用户感兴趣的事务。**

# 在Web访问挖掘中的常用方法

1. 路径分析
   - 路径分析最常用的应用是用于判定在一个Web站点中最频繁访问的路径，这样的知识对于一个电子商务网站或者信息安全评估是非常重要的。
2. 关联规则发现
   - 使用关联规则发现方法可以从Web访问事务集中，找到一般性的关联知识。
3. 序列模式发现
   - 在时间戳有序的事务集中，序列模式的发现就是指找到那些如“一些项跟随另一个项”这样的内部事务模式。
4. 分类
   - 发现分类规则可以给出识别一个特殊群体的公共属性的描述。这种描述可以用于分类新的项。
5. 聚类
   - 从Web Usage数据中聚集出具有相似特性的那些客户。在Web事务日志中，聚类顾客信息或数据项，就能够便于开发和执行未来的市场战略。

# Web访问信息挖掘的要素构成

1. 数据来源
   - 数据的来源分为服务器，代理服务器，和客户端。
2. 数据类型
   - 数据的类型主要分为结构,内容,访问信息,用户概貌文件。
3. 用户的数量
   - 用户的数量表现为：或者数据集只由一个用户的信息构成，或者数据由多个用户的信息构成。
4. 站点的数量
   - 在数据集中的Web站点的个数表现为：或者在数据集中只记录单个站点的信息，或者记录多个站点的信息。
5. 服务对象
   - Web访问信息挖掘的结果由Web服务方进行应用。应用的结果即服务对象可以是当个单个用户，或群体用户。单个用户即意味着个性化。
6. 挖掘手段
   - Web访问信息挖掘所采用的各种数据挖掘方法，例如关联规则发现，聚类，分类，统计等等。

# 利用Web访问信息挖掘实现用户建模

- 由于Web网站的特性，对网站的经营者和设计者而言，无法直接了解用户的特性。然而对访问者个人特性和群体用户特性的了解对Web网站的服务方而言显得尤为重要。幸运的是可以通过数据挖掘的方法得到用户的特性。
- “用户建模”（Modelling Users）是指根据访问者对一个Web站点上Web页面的的访问情况，可以模型化用户的自身特性。在识别出用户的特性后就可以开展针对性的服务。用户建模主要有三种途径。
  - 推断匿名访问者的人口统计特性
  - 在不打扰用户的情况下，得到用户概貌文件
  - 根据用户的访问模式来聚类用户

# 利用Web访问信息挖掘发现导航模式

- 发现导航模式（Discovering Navigation Patterns）是Web访问信息挖掘的一个重要的研究领域。用户的导航模式是指群体用户对Web站点内的页面的浏览顺序模式。
- 用户导航模式的主要应用在改进站点设计和个性化推销等方面。
  - 1. 改进Web站点的结构设计
  - 2. 个性化行销（Direct Marketing）：
  - 3. 利用关联规则发现算法发现导航模式
  - 4. 利用模板发现导航模式
  - 5. 利用超文本概率文法发现导航模式

# 利用Web访问信息挖掘进行个性化服务

在Web站点开展个性化（Personalization）服务的总的思路和步骤是：

- 模型化页面和用户；
- 分类页面和用户；
- 在页面和对象之间进行匹配；
- 判断当前访问的类别以进行推荐。

而且，个性化系统一般分为两个部分：离线部分和在线部分。

表7-11个性化方法的比较

| 方法 | 特点 | 缺点 |
| --- | --- | --- |
| 离线聚类和动态链接结合 | 可以实时个性化地为用户提供推荐。 | 随着用户访问长度的增加，可供推荐的元素会趋于零。 |
| 基于关键词学习 | 引入时间特性为用户提供推荐。 | 需要用户人工干预，无法做到自动。 |
| 识别感兴趣的链接 | 建立代理服务器识别用户的访问兴趣提供推荐。 | 用户兴趣的实效性考虑不够。 |
| 自动定制不同用户访问界面 | 利用用户建模技术自动定制不同的用户访问界面。 | “推论”依赖于用户所在的领域，适应性不好。 |
| 利用客户端代理进行个性化 | 客户端的代理，完全为个人服务。 | 冗余搜索过大。 |
| 聚类推荐 | 可以实时个性化地为用户提供推荐。 | 聚类的个数是人为事先给定的，不能随着每个用户的访问特性而动态调整。 |

# 利用Web访问信息挖掘进行商业智能发现

表7-12　　商业智能方法的比较

| 方法 | 特点 |
| --- | --- |
| Buchner | 其贡献在于首次在Web访问信息挖掘的基础上提出了商业智能的发现的框架；其不足在于发现的知识局限于用户确实发生的购买行为，而对用户潜在的购买兴趣无法发现。 |
| Yun C. | 优点是挖掘了迁移和购买行为之间的内在关系。缺点是发现的知识局限于用户确实发生的购买行为，对用户潜在购买兴趣无法发现。 |
| SurfAid，Accrue，NetGenesis，Aria，Hitlist，WebTrends | 优点是通过分析页面的点击率来为推断商业智能提供Web流量分析。缺点是无法发现高级的商业职能。 |

# 利用协作推荐的方法实现实时个性化推荐

基于协作筛方法的Web站点实时个性化系统的结构如图7-3所示。

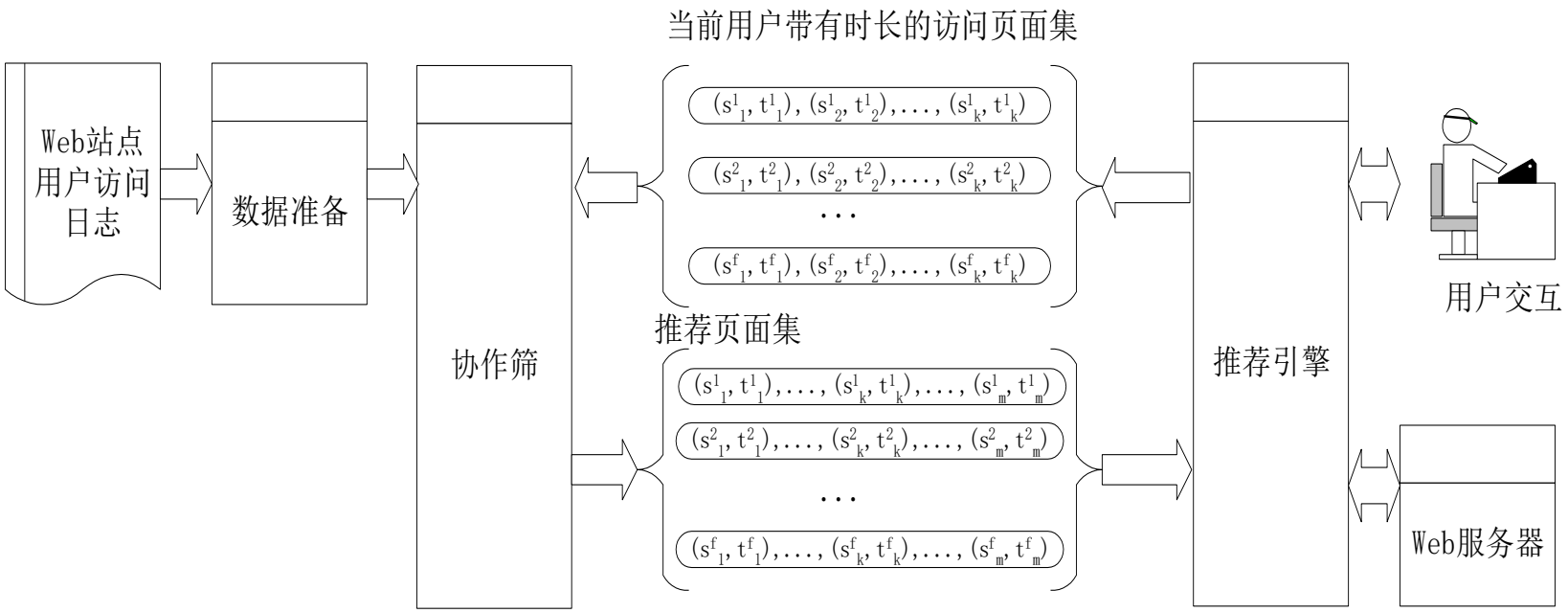



图7-3　基于协作筛方法的Web站点实时个性化系统

- 整个处理过程分为两部分：
  - 离线部分：包括数据准备、得到推荐池、建立协作筛。
  - 在线部分：推荐引擎。

# 数据分析团队的组建

金字塔式

领导者：首席数据官或项目经理

数据科学家+数据工程师+业务专家+各个业务部分

矩阵式

没有具体负责人

以工作划分小团体

职能岗位

项目经理

业务专家

数据工程师

数据建模人员

可视化人员

评估人员

# 数据分析人才培养的难题

- 数理要求高
- 跨学科综合能力
- 国内技术资料少
- 实践机会少
- 数据分析是一门入门容易但精通难的学科
- 数据分析人员需要掌握行业知识以了解业务流程、理解数据背后的隐含信息以合理解读数据、从变化的角度和时间维度把握需求以确定使用哪些数据，这是数据分析的基础
- 数据分析的主要流程是：明确分析目标、数据收集、数据预处理、建模分析、结果评估、结论整理及建议、预测分析

# 第七章 Web挖掘技术

## 内容提要

- Web挖掘的意义
- Web挖掘的分类
- Web挖掘的含义
- Web挖掘的数据来源
- Web内容挖掘方法
- Web访问信息挖掘方法
- Web结构挖掘方法

# Web内容挖掘的主要方法

- 一种Web内容挖掘的分类方法是分为代理人方法和数据库方法。
  - 代理人方法是指使用软件（代理）来完成内容挖掘。
    - 搜索引擎软件就是此类方法的初级阶段。智能检索代理超越了简单的检索机制，可以综合运用数据挖掘等技术实现内容的分析。
    - 页面的分类、聚类。
    - 增加用户的喜好信息来指导检索。
  - 数据库方法是指将Web数据描述为一个数据库系统。
    - Web网站看做是一个多级的异构的数据库系统，可以通过查询语言来获得Web的信息、进而完成信息的抽取和挖掘。

# 文本挖掘是Web内容挖掘的基础

- 文本挖掘（TD）的方式和目标是多种多样的，如：
  - 关键词检索：最简单的方式，它和传统的搜索技术类似。
  - 挖掘项目关联：聚焦在页面的信息（包括关键词）之间的关联信息挖掘上。
  - 信息分类和聚类：利用数据挖掘的分类和聚类技术实现页面的分类，将页面在一个更高层次上进行抽象和整理。
  - 自然语言处理：揭示自然语言处理技术中的语义，实现Web内容的更精确处理。
- 难度也是巨大的
  - 语法层面：容易形式化
  - 语义层面：取决于NLP等技术的进展

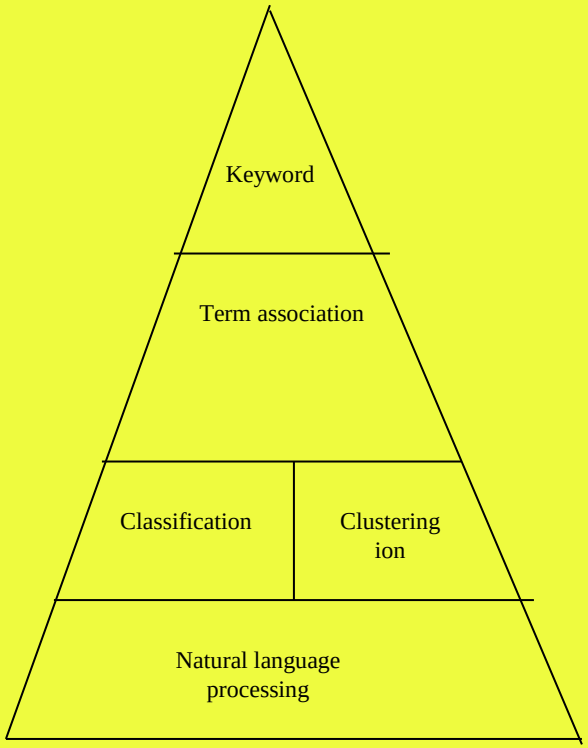



图7-1文本挖掘体系示意

# 虚拟的Web视图

- 利用HTML或者XML进行内容挖掘是可行的。
- 另一个被研究的解决方案被称为VMV（Virtual Web View）的视图机制，Web中的感兴趣的结构被浓缩在这个视图中。
- VMV是将Web中大量无结构数据组成一个MLDB（Multiple Layered Database）。这个数据库是多层次的，每层索引都比它下一层要小。对于最底层来说，需要了解Web文档结构，而最高层则有着完善的结构并可以通过类似SQL的查询语言进行访问或挖掘。
- 等级概念（近意词组、词汇和语义联系等）将帮助归纳过程来架构更高层的MLDB。

# Web页面内文本信息挖掘典型方法

- 对页面进行摘要和分类。
  - 页面摘要：对每一个页面得到相应的摘要信息。
  - 页面分类：分类器输入的是一个Web页面集（训练集），再根据页面文本信息内容进行监督学习，然后把学成的分类器用于分类页面。
- 典型方法是TFIDF向量表示法
  - 不考虑词间的次序和文本的结构。构造二维表：
    - **每一列对应一个词。列集（特征集）为辞典中的所有有区分价值的词，所以整个列集可能有几十万列之多。**
    - **每一行对应一个页面。该页面中的所有词对应到列集（特征集）上。**
    - **表中数据：如果行页面的列词汇不出现，则其值为0；如果出现k次，那么其值就为k；页面中的词如果不出现在列集上，可以被放弃。这种方法可以表征出页面中词的频度。**
- 对中文页面来说，还需先分词然后再进行以上两步处理。
  - 这样构造的二维表表示的是Web页面集合的词的统计信息，最终就可以采用数据挖掘方法进行关联规则、分类及聚类挖掘。
  - 在挖掘之前，一般要先进行特征子集的选取，以降低维数。

# Web页面内多媒体信息挖掘

- 多媒体挖掘是一个大研究分支，总的挖掘过程是先要应用多媒体信息特征提取工具，形成特征2维表，然后就可以采用传统的数据挖掘方法进行挖掘。
  - 在特征提取阶段，利用多媒体信息提取工具进行特征提取。一般地，信息提取工具能够抽取出image和video的文件名、URL、父URL、类型、键值表、颜色向量等。对这些特征可以进行如下挖掘操作：
  - 关联规则发现：例如，如果图像是“大”的而且与关键词“天空”有关，那么它是蓝色的概率为68%。
  - 分类：根据提供的某种类标，针对特征集，利用决策树可以进行分类。

# 第七章 Web挖掘技术

## 内容提要

- Web挖掘的意义
- Web挖掘的分类
- Web挖掘的含义
- Web挖掘的数据来源
- Web内容挖掘方法
- Web访问信息挖掘方法
- Web结构挖掘方法

# Thank you !!!