7.4 锚链网络的建模

- 网页两大特性
  - HTML**标签**，网页之间的**超链**。
- 标签
  - 将网页不同部分以不同形式呈现给用户不同视觉效果的手段（布局，字体、字号的变化）
  - 提示某些内容的重要程度
- 链接
  - 反映网页间形成的“参考”、“引用”和“推荐”之关系。
  - 合理假设，若1网页被较多其它网页链接，则它较被关注/重要/有用
  - 入度--指向该网页的超链数。衡量其重要程度之指标
  - 出度--从该网页链出的超链数。分析网上信息状况有意义。

Google之PageRank算法

- 早期搜索引擎
  - 关键词出现概率；Html标签提示等进行权重修订。
  - 不能反映实际，重要性并不一定反映在关键词中
- 链接流行度(link popularity)
  - 由页面链接数量来决定当前页面的重要性，
  - 防止人为加工的页面欺骗搜索引擎
- PageRank算法
  - 每个链入赋不同权值，上链页面越重要则该链入权越高
  - 即当前页面的重要性由其它页面的重要性决定。

PageRank算法1

$$PR(A) = (1-d) + d(\frac{PR(T_1)}{C(T_1)} + \frac{PR(T_2)}{C(T_2)} + ... \frac{PR(T_n)}{C(T_n)}) = (1-d) + d\sum_{i=1}^{n} \frac{PR(T_i)}{C(T_i)}$$

- PR(X)是页面X的级别或重要程度
- T_i是链向页面A的第i个页面
- $C(T_i)$ 是页面T_i 链出的链接数量
- d是阻尼系数
  - 取值在0到1之间；
  - 用户不可能无限单击下去，常劳累而随机跳入另一页
  - 1-d是页面A本身所具有的网页级别或重要程度

- 随机冲浪模型
  - Random surfer model
  - Sergey Brin& Lawrence Page 提出
  - 用户点击超链动作是一种不关心内容的随机行为
  - 用户点击页面内某一超链接的概率，完全由该页面上所包含超链多少所决定
  - 冲浪到该页的概率是上链各页面上超链被点击概率之和

PageRank算法2

$$PR(A)=\frac{(1-d)}{N}+\frac{d}{N}(\frac{PR(T_1)}{C(T_1)}+\frac{PR(T_2)}{C(T_2)}+.....\frac{PR(T_n)}{C(T_n)})=\frac{(1-d)}{N}+\frac{d}{N}\sum_{i=1}^{n}\frac{PR(T_i)}{C(T_i)}$$

- PageRank算法1 的修订
  - 其中N是互联网上网页的数量。
  - 所有页面的网页级别形成一个概率分布，所有页面的网页级别之和为1。
  - 算法1--所有页面的网页级别之和等于互联网上网页的数量。

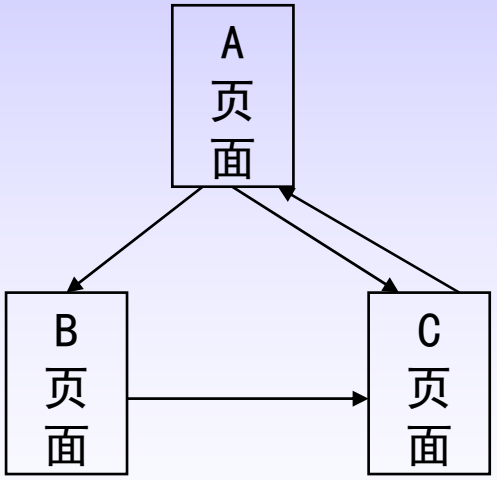
锚链网建模例子

- 按算法1，对3个页面间锚链可列如下方程
- 不妨设d=0. 5

$$PR(A) = 0.5 + 0.5PR(C)$$
$$PR(B) = 0.5 + 0.5PR(A)/2$$
$$PR(C) = 0.5 + 0.5(PR(A)/2 + PR(B))$$

- 解方程

$$PR(A) = 14/13 = 1.0769$$
$$PR(B) = 10/13 = 0.76923$$
$$PR(C) = 15/13 = 1.1538$$
$$PR(A) + PR(B) + PR(C) = 2.99993 = 3$$



PageRank算法

- Google用近似迭代方法计算网页级别
  - 给每个网页赋予一个初值，
  - 利用上面公式，有限次循环运算得到近似的网页级别
  - 实际进行大约100次迭代才能得到整个网络的网页级别

初值选为1

迭代次数	PR(A)	PR(B)	PR(C)
0	1	1	1
1	1	0.75	1.125
2	1.0625	0.76563	1.1484
3	1.0742	0.76855	1.1528
4	1.0764	0.7691	1.1537
5	1.0768	0.76921	1.1538
6	1.0769	0.76923	1.1538
7	1.0769	0.76923	1.1538

初值选为1.5

迭代次数	PR(A)	PR(B)	PR(C)
0	1.5	1.5	1.5
1	1.25	0.8125	1.2188
2	1.1094	0.77734	1.166
3	1.083	0.77075	1.1561
4	1.0781	0.76952	1.1543
5	1.0771	0.76928	1.1539
6	1.077	0.76924	1.1539
7	1.0769	0.76923	1.1538
8	1.0769	0.76923	1.1538

Clever的Hits算法

- IBM研究院在Clever系统：
  - HITS: Hyperlink-Induced Topic Search
  - 用该算法计算一个网页重要性。
- 两种类型网页：
  - 权威型网页--Authority：对某个特定检索，该网页提供**最好的相关信息**；
  - 目录型网页--Hub：该网页提供很多指向其它**高质量权威型**网页的超链。
- 对每个网页上定义“目录型权值”和“权威型权值”两参数
- Hits算法基本思想
  - **好Hub型网页指向好的Authority网页**
  - **好的Authority网页是被好的Hub型网页所指向的网页**

1）将查询q提交给基于关键字查询的检索系统，从返回结果页面的集合总取前n个网页（如n=200），作为根集合（root set），记为S，则S满足：

- S中的网页数量较少
- S中的网页是与查询q相关的网页
- S中的网页包含较多的Authority网页

2）将S扩展为基本集合(base set) T，T包含由S指出或指向S的网页。可以设定一个上限如 1000—5000个网页

3）开始**权重传播**。在集合T中计算每个网页的目录型权值和权威型权值。

- 用目录型网页和权威型网页相互评价的办法进行递归计算。
- 对某个网页p，用xp表示网页p的权威型权值；
- 用yp表示其目录型权值，并且用如下公式进行计算

$$x_p = \sum_{q \ such \ that \ q->p} y_q$$

$$y_p = \sum_{q \ such \ that \ p->q} x_q$$

Hits算法

这样的递归式也容易用矩阵方法表示。令所有选出来的网页都进行标号，我们得到所有网页的编号集{1,2,…,n}。令相邻矩阵A 为一个 $n \times n$ 的矩阵，如果存在一个从网页i链接到网页j 的超链，就令矩阵中的第（i,j）个元素置为 1，其它各项置为 0。同时，我们将所有网页的权威型权值x和目录型权值y都表示成向量形式 $x = (x_1, x_2, \ldots x_n)$，$y = (y_1, y_2, \ldots y_n)$。由此我们可以得到计算x和y的简单矩阵公式：$y = A \cdot x$，$x = A^T \cdot y$，其中 A^T 是A的转置矩阵。进一步，我们有：

$$x = A^T \cdot y = A^T Ax = (A^T A)x$$

$$y = A \cdot x = AA^T y = (AA^T)y$$

经过一定次数的递归运算后，会得到集合中每个网页的权威型权值和目录型权值。按照这两个不同的权值，分别取出前 k 个返回给用户。

根据 Clever 系统自己的测试数据，对于返回给用户的前 10 个检索结果，Clever 系统在50%的情况下获得了高于 Yahoo!和 AltaVista 的用户评价。

PageRank与Hits算法

- 它们都利用了网页和超链组成的有向图，根据相互链接的关系进行递归的运算。
- 但是，两者又有很大的区别，主要在于运算的时机
  - Google是在网页搜集告一段落时，离线的使用一定的算法计算每个网页的权值，在检索时只需要从数据库中取出这些数据即可，而不用做额外的运算，这样做的好处是检索的速度快，但丧失了检索时的灵活型。
  - Clever使用即时分析运算策略，每得到一个检索，它都要从数据库中找到相应的网页，同时提取出这些网页和链接构成的有向子图，再运算获得各个网页的相应链接权值。这种方法虽然灵活性强，并且更加精确，但在用户检索时进行如此大量的运算，检索效率显然不高。

许多改进形式

- 上述算法基于“网页”级别
- 最近，微软提出了基于“块”间连接的PageRank算法(这是一种更细的划分)
- 提高网站排名方法，比如：网页之间的互相连接，不可见关键词的堆积（字体颜色与背景颜色一致）

Thank you!