Web 超链分析算法研究*)

朱 炜 王 超 李 俊 潘金贵
(南京大学计算机软件新技术国家重点实验室 南京210093)
(南京大学多媒体技术研究所 南京210093)

Research on Algorithms Analyzing Hyperlinks: A Survey

ZHU Wei WANG Chao LI Jun Pan Jin-Gui
(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093)
(Multimedia Technology Institute of Nanjing University, Nanjing 210093)

Abstract The World Wide Web serves as a huge, widely distributed, global information service center, and expanding in a rapid speed. It is import to find the information the user need precisely and rapidly. In recent years, researchers discovery that rich and import information is contained among hyperlinks, and develop a lot of algorithm using hyperlink to improve the quantity and relevance of the results which search engine returned. This paper presents a review and a comparison of such algorithms existing now. Problems of these algorithms and directions to further research will be discussed.

Keywords PageRank, Authority, Hub, HITS, SALSA, Anchor

1. 引言

万维网 WWW(World Wide Web)是一个巨大的、分布全球的信息服务中心,正在以飞快的速度扩展。1998年 WWW 上拥有约3.5亿个文档[14],每天增加约1百万的文档[6],不到9个月的时间文档总数就会翻一番[14]。Web 上的文档和传统的文档比较,有很多新的特点,它们是分布的、异构的、无结构或者半结构的,这就对传统信息检索技术提出了新的挑战。

传统的 Web 搜索引擎大多数是基于关键字匹配的,返回的结果是包含查询项的文档,也有基于目录分类的搜索引擎。这些搜索引擎的结果并不令人满意。有些站点有意提高关键字出现的频率来提高自身在搜索引擎中的重要性,破坏搜索引擎结果的客观性和准确性。另外,有些重要的网页并不包含查询项。搜索引擎的分类目录也不可能把所有的分类考虑全面,而且目录大多靠人工维护,主观性强,费用高,更新速度慢[2]。

最近几年,许多研究者发现,WWW 上超链结构是个非常丰富和重要的资源,如果能够充分利用的话,可以极大地提高检索结果的质量。基于这种超链分析的思想,Sergey Brin 和 Lawrence Page 在1998年提出了 PageRank 算法[1],同年 J. Kleinberg 提出了 HITS 算法[5],其它一些学者也相继提出了另外的链接分析算法,如 SALSA, PHITS, Bayesian 等算法。这些算法有的已经在实际的系统中实现和使用,并且取得了良好的效果。

本文按照时间顺序详细剖析了各种链接分析算法,对不同的算法进行了比较。对这些算法做了评价和总结,指出了存在的问题和改进方向。

2. Web 超链分析算法

2.1 Google 和 PageRank 算法

搜索引擎 Google 最初是斯坦福大学的博士研究生 Sergey Brin 和 Lawrence Page 实现的一个原型系统[2],现在已经发展成为 WWW 上最好的搜索引擎之一。Google 的体系结构类似于传统的搜索引擎,它与传统的搜索引擎最大的不同处在于对网页进行了基于权威值的排序处理,使最重要的网页出现在结果的最前面。Google 通过 PageRank 元算法计算出网页的 PageRank 值,从而决定网页在结果集中的出现位置,PageRank 值越高的网页,在结果中出现的位置越前。

2.1.1 PageRank 算法 PageRank 算法基于下面2个前提:

前提1 一个网页被多次引用,则它可能是很重要的;一个网页虽然没有被多次引用,但是被重要的网页引用,则它也可能是很重要的;一个网页的重要性被平均地传递到它所引用的网页。这种重要的网页称为权威(Authoritive)网页。

前提2 假定用户一开始随机地访问网页集合中的一个网页,以后跟随网页的向外链接向前浏览网页,不回退浏览,浏览下一个网页的概率就是被浏览网页的 PageRank 值。

简单 PageRank 算法描述如下:u 是一个网页,$F(u)$是 u 指向的网页集合,$B(u)$是指向 u 的网页集合,$N(u)$是 u 指向外的链接数,显然 $N(u)=|F(u)|$,c 是一个用于规范化的因子(Google 通常取0.85),(这种表示法也适用于以后介绍的算法)则 u 的 Rank 值计算如下:

$$R(u)=c\sum_{v\in B(u)}R(v)/N(v)$$

这就是算法的形式化描述,也可以用矩阵来描述此算法,设 A 为一个方阵,行和列对应网页集的网页。如果网页 i 有指

*)本文得到日本邮政省通信放送机构(TAO)国际研究基金资助。朱 炜 硕士研究生,主要研究方向为万维网,信息检索。王 超 硕士研究生,主要研究方向为信息检索,Agent 技术。李 俊 硕士研究生,主要研究方向为 Agent 技术,信息检索,多媒体远程教育。潘金贵 教授,博士生导师,主要研究方向为中间件,Agent 技术,多媒体远程教育。

向网页 j 的一个链接，则 $A_{i,j}=1/N_i$，否则 $A_{i,j}=0$。设 V 是对应网页集的一个向量，有 $V=cAV$，V 为 A 的特征根为 c 的特征向量。实际上，只需要求出最大特征根的特征向量，就是网页集对应的最终 PageRank 值，这可以用迭代方法计算。

如果有2个相互指向的网页 a，b，它们不指向其它任何网页，另外有某个网页 c，指向 a，b 中的某一个，比如 a，那么在迭代计算中，a，b 的 rank 值不分布出去而不断地累计。如图1。

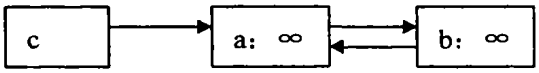


图1

为了解决这个问题，Sergey Brin 和 Lawrence Page 改进了算法，引入了衰退因子 E(u)，E(u)是对应网页集的某一向量，对应 rank 的初始值，算法改进如下：

$$R'(u)=c\sum_{v\in B(u)}R(v)/N(v)+cE(u)$$

其中，$\|R'\|_1=1$，对应的矩阵形式为 $V'=c(AV'+E)$。

另外还有一些特殊的链接，指向的网页没有向外的链接。PageRank 计算时，把这种链接首先除去，等计算完以后再加入，这对原来计算出的网页的 rank 值影响是很小的。

Pagerank 算法除了对搜索结果进行排序外，还可以应用到其它方面，如估算网络流量、向后链接的预测器、为用户导航等[2]。

2.1.2 *算法的一些问题* Google 是结合文本的方法来实现 PageRank 算法的[2]，所以只返回包含查询项的网页，然后根据网页的 rank 值对搜索到的结果进行排序，把 rank 值最高的网页放置到最前面，但是如果最重要的网页不在结果网页集中，PageRank 算法就无能为力了，比如在 Google 中查询 search engines，像 Google，Yahoo，Altivisa 等都是很重要的，但是 Google 返回的结果中这些网页并没有出现。同样的查询例子也可以说明另外一个问题，Google，Yahoo 是 WWW 上最受欢迎的网页，如果出现在查询项 car 的结果集中，一定会有很多网页指向它们，就会得到较高的 rank 值，事实上它们与 car 不太相关。

在 PageRank 算法的基础上，其它的研究者提出了改进的 PageRank 算法。华盛顿大学计算机科学与工程系的 Matthew Richardson 和 Pedro Dominggos 提出了结合链接和内容信息的 PageRank 算法，去除了 PageRank 算法需要的前提2，增加考虑了用户从一个网页直接跳转到非直接相邻的但是内容相关的另外一个网页的情况[3]。斯坦福大学计算机科学系 Taher Haveliwala 提出了主题敏感(Topic-sensitive) PageRank 算法[4]。斯坦福大学计算机科学系 Arvind Arasu 等经过试验表明，PageRank 算法计算效率还可以得到很大的提高[22]。

2.2 HITS 算法及其变种

PageRank 算法中对于向外链接的权值贡献是平均的，也就是不考虑不同链接的重要性。而 Web 的链接具有以下特征：1)有些链接具有注释性，也有些链接是起导航或广告作用。有注释性的链接才用于权威判断。2)基于商业或竞争因素考虑，很少有 Web 网页指向其竞争领域的权威网页。3)权威网页很少具有显式的描述，比如 Google 主页不会明确给出 Web 搜索引擎之类的描述信息。可见平均的分布权值不符合链接的实际情况[17]。J. Kleinberg[5]提出的 HITS 算法中引入了另外一种网页，称为 Hub 网页，Hub 网页是提供指向权威网页链接集合的 Web 网页，它本身可能并不重要，或者说没有几个网页指向它，但是 Hub 网页却提供了指向就某个主题而言最为重要的站点的链接集合。一般来说，好的 Hub 网页指向许多好的权威网页；好的权威网页是有许多好的 Hub 网页指向的 Web 网页。这种 Hub 与 Authoritive 网页之间的相互加强关系，可用于权威网页的发现和 Web 结构和资源的自动发现，这就是 Hub/Authority 方法的基本思想。

2.2.1 HITS(Hyperlink－Induced Topic Search)*算法* 是利用 Hub/Authority 方法的搜索方法，算法如下：将查询 q 提交给传统的基于关键字匹配的搜索引擎。搜索引擎返回很多网页，从中取前 n 个网页作为根集(root set)，用 S 表示。S 满足如下3个条件：

1. S 中网页数量相对较小；
2. S 中网页大多数是与查询 q 相关的网页；
3. S 中网页包含较多的权威网页。

通过向 S 中加入被 S 引用的网页和引用 S 的网页将 S 扩展成一个更大的集合 T。

以 T 中的 Hub 网页为顶点集 Vl，以权威网页为顶点集 V2，Vl 中的网页到 V2中的网页的超链接为边集 E，形成一个二分有向图 SG＝(V1，V2，E)。对 V1中的任一个顶点 v，用 h(v)表示网页 v 的 Hub 值，对 V2中的顶点 u，用 a(u)表示网页的 Authority 值。开始时 h(v)＝a(u)＝1，对 u 执行 I 操作修改它的 a(u)，对 v 执行 O 操作修改它的 h(v)，然后规范 a(u)，h(v)，如此不断地重复计算下面的操作 I、O，直到 a(u)，h(v)收敛。(证明此算法收敛可见文[5])

$$\text{I 操作：}a(u)=\sum_{v_i:(v,u)\in E}h(v) \tag{1}$$

$$\text{O 操作：}h(v)=\sum_{u_i:(v,u)\in E}a(u) \tag{2}$$

每次迭代后需要对 a(u)，h(v)进行规范化处理：

$$a(u)=a(u)/\sqrt{\sum_{q\in V_2}[a(q)]^2}\qquad h(v)=h(v)/\sqrt{\sum_{q\in v_1}[h(q)]^2}$$

式(1)反映了若一个网页由很多好的 Hub 指向，则其权威值会相应增加(即权威值增加为所有指向它的网页的现有 Hub 值之和)。式(2)反映了若一个网页指向许多好的权威页，则 Hub 值也会相应增加(即 Hub 值增加为该网页链接的所有网页的权威值之和)。

和 PageRank 算法一样，可以用矩阵形式来描述算法，这里省略不写。

HITS 算法输出一组具有较大 Hub 值的网页和具有较大权威值的网页。

2.2.2 HITS *的问题* HITS 算法有以下几个问题：

1)实际应用中，由 S 生成 T 的时间开销是很昂贵的，需要下载和分析 S 中每个网页包含的所有链接，并且排除重复的链接。一般 T 比 S 大很多，由 T 生成有向图也很耗时。需要分别计算网页的 A/H 值，计算量比 PageRank 算法大。

2)有些时候，一主机 A 上的很多文档可能指向另外一台主机 B 上的某个文档，这就增加了 A 上文档的 Hub 值和 B 上文档的 Authority，相反的情况也如此。HITS 是假定某一文档的权威值，是由不同的单个组织或者个人决定的，上述情况影响了 A 和 B 上文档的 Hub 和 Authority 值[7]。

3)网页中一些无关的链接影响 A、H 值的计算。在制作网页的时候，有些开发工具会自动地在网页上加入一些链接，这

些链接大多是与查询主题无关的。同一个站点内的链接目的是为用户提供导航帮助，也与查询主题不甚无关，还有一些商业广告，赞助商和用于友情交换的链接，也会降低 HITS 算法的精度[8]。

4)HITS 算法只计算主特征向量，也就是只能发现 T 集合中的主社区(Community)，忽略了其它重要的社区[12]。事实上，其它社区可能也非常重要。

5)HITS 算法最大的弱点是处理不好主题漂移问题(topic drift)[7,8]，也就是紧密链接 TKC(Tightly-Knit Community Effect)现象[8]。如果在集合 T 中有少数与查询主题无关的网页，但是它们是紧密链接的，HITS 算法的结果可能就是这些网页，因为 HITS 只能发现主社区，从而偏离了原来的查询主题。下面讨论的 SALSA 算法中解决了 TKC 问题。

6)用 HITS 进行窄主题查询时，可能产生主题泛化问题[5,9]，即扩展以后引入了比原来主题更重要的新的主题，新的主题可能与原始查询无关。泛化的原因是因为网页中包含不同主题的向外链接，而且新主题的链接具有更加的重要性。

2.2.3 HITS 的变种 HITS 算法遇到的问题大多是因为 HITS 是纯粹的基于链接分析的算法，没有考虑文本内容，继 J. Kleinberg 提出 HITS 算法以后，很多研究者对 HITS 进行了改进，提出了许多 HITS 的变种算法，主要有：

(1)Monika R. Henzinger 和 Krishna Bharat 对 HITS 的改进

对于上述提到的 HITS 遇到的第2个问题，Monika R. Henzinger 和 Krishna Bharat 在[7]中进行了改进。假定主机 A 上有 k 个网页指向主机 B 上的某个文档 d，则 A 上的 k 个文档对 B 的 Authority 贡献值总共为1，每个文档贡献1/k，而不是 HITS 中的每个文档贡献1，总共贡献 k。类似地，对于 Hub 值，假定主机 A 上某个文档 t 指向主机 B 上的 m 个文档，则 B 上 m 个文档对 t 的 Hub 值总共贡献1，每个文档贡献1/m。I,O 操作改为如下

I 操作：$a(u)=\sum_{v_i:(v,u)\in E} h(v)\times au\_wt(v,u)$

O 操作：$h(v)=\sum_{u_i:(v,u)\in E} a(u)\times hub\_wt(v,u)$

调整后的算法有效地解决了问题2，称之为 imp 算法。

在这基础上，Monika R. Henzinger 和 Krishna Bharat 还引入了传统信息检索的内容分析技术来解决4和5，实际上也同时解决了问题3。具体方法如下，提取根集 S 中的每个文档的前1000个词语，串连起来作为查询主题 Q，文档 D_j 和主题 Q 的相似度按如下公式计算：

$$similarity(Q,D_j)=\Sigma(U_{iq},U_{ij})/\sqrt{(\Sigma U_{iq}^2)\times(\Sigma U_{ij}^2)}$$

$U_{iq}=FREQ_{iq}\times IDF_i$，$U_{ij}=FREQ_{ij}\times IDF_i$，$FREQ_{iq}$=项 i 在查询 Q 中的出现次数，$FREQ_{ij}$=项 i 在文档 D_j 中的出现次数，IDF_i 是 WWW 上包含项 i 的文档数目的估计值。

在 S 扩展到 T 后，计算每个文档的主题相似度，根据不同的阈值进行刷选，可以选择所有文档相似度的中值，根集文档相似度的中值，最大文档相似度的分数，如1/10，作为阈值。根据不同阈值进行处理，删除不满足条件的文档，再运行 imp 算法计算文档的 A/H 值，这些算法分别称为 med，startmed，maxby10。

在此改进的算法中，计算文档的相似度时间开销会很大。

(2)ARC 算法

IBM Almaden 研究中心的 Clever 工程组提出了 ARC (Automatic Resource Compilation)算法，对原始的 HITS 做了改进，赋予网页集对应的连结矩阵初值时结合了链接的锚(anchor)文本，适应了不同的链接具有不同的权值的情况。

ARC 算法与 HITS 的不同主要有以下3点：

①由根集 S 扩展为 T 时，HITS 只扩展与根集中网页链接路径长度为1的网页，也就是只扩展直接与 S 相邻的网页，而 ARC 中把扩展的链接长度增加到2，扩展后的网页集称为增集(Augment Set)。

②HITS 算法中，每个链接对应的矩阵值设为1，实际上每个链接的重要性是不同的，ARC 算法考虑了链接周围的文本来确定链接的重要性。考虑链接 p→q，p 中有若干链接标记，文本1(a href="q")锚文本(/a)文本2，设查询项 t 在文本1，锚文本，文本2，出现的次数为 n(t)，则 w(p,q)=1+n(t)。文本1和文本2的长度经过试验设为50字节[10]。构造矩阵 W，如果有网页 i→j，$W_{i,j}=w(i,j)$，否则 $W_{i,j}=0$，H 值设为1，Z 为 W 的转置矩阵，迭代执行下面3个操作：(1)A=WH；(2)H=ZA；(3)规范化 A，H。

③ARC 算法的目标是找到前15个最重要的网页，只需要 A/H 的前15个值相对大小保持稳定即可，不需要 A/H 整个收敛，这样2中迭代次数很小就能满足，文[10]中指出迭代5次就可以，所以 ARC 算法有很高的计算效率，开销主要是在扩展根集上。

(3)Hub 平均(Hub-Averaging-Kleinberg)算法

Allan Borodin 等在文[11]指出了一种现象，设有 M+1 个 Hub 网页，M+1个权威网页，前 M 个 Hub 指向第一个权威网页，第 M+1个 Hub 网页指向了所有 M+1个权威网页。显然根据 HITS 算法，第一个权威网页最重要，有最高的 Authority 值，这是我们希望的。但是，根据 HITS，第 M+1个 Hub 网页有最高的 Hub 值，事实上，第 M+1个 Hub 网页既指向了权威值很高的第一个权威网页，同时也指向了其它权威值不高的网页，它的 Hub 值不应该比前 M 个网页的 Hub 值高。因此，Allan Borodin 修改了 HITS 的 O 操作：

O 操作：$h(v)=\sum_{v_i:(v,u)\in E} a(u)/n$，n 是(v,u)的个数

调整以后，仅指向权威值高的网页的 Hub 值比既指向权威值高又指向权威值低的网页的 Hub 值高，此算法称为 Hub 平均(Hub-Averaging-Kleinberg)算法。

(4)阈值(Threshhold-Kleinberg)算法

Allan Borodin 等在文[11]中同时提出了3种阈值控制的算法，分别是 Hub 阈值算法，Authority 阈值算法，以及结合2者的全阈值算法。

计算网页 p 的 Authority 时，不考虑指向它的所有网页 Hub 值对它的贡献，只考虑 Hub 值超过平均值的网页的贡献，这就是 Hub 阈值方法。

Authority 阈值算法和 Hub 阈值方法类似，不考虑所有 p 指向的网页的 Authority 对 p 的 Hub 值贡献，只计算前 K 个权威网页对它 Hub 值的贡献，这是基于算法的目标是查找最重要的 K 个权威网页的前提。

同时使用 Authority 阈值算法和 Hub 阈值方法的算法，就是全阈值算法。

2.3 SALSA 算法

PageRank 算法是基于用户随机的向前浏览网页的直觉知识，HITS 算法考虑的是 Authoritive 网页和 Hub 网页之间的加强关系。实际应用中，用户大多数情况下是向前浏览网页，但是很多时候也会回退浏览网页。基于上述直觉知识，R.

Lempel 和 S. Moran 提出了 SALSA(Stochastic Approach for Link-Structure Analysis)算法[8],考虑了用户回退浏览网页的情况,保留了 PageRank 的随机漫游和 HITS 中把网页分为 Authoritive 和 Hub 的思想,取消了 Authoritive 和 Hub 之间的相互加强关系。

具体算法如下:

1. 和 HITS 算法的第一步一样,得到根集并且扩展为网页集合 T,并除去孤立节点。

2. 从集合 T 构造无向图 G'=(Vh,Va,E)

$V_h=\{s_h|s\in C$ and out-degree$(s)>0\}$(G' 的 Hub 边).

$V_a=\{s_a|s\in C$ and in-degree$(s)>0\}$(G' 的 Authority 边).

$E=\{(s_h,r_a)|s\rightarrow r$ in $T\}$

这就定义了2条链,Authority 链和 Hub 链。

3. 定义2条马尔可夫链的变化矩阵,也是随机矩阵,分别是 Hub 矩阵 H,Authority 矩阵 A。

$$H_{i,j}=\sum_{k:k\in F(i)\cap F(j)}\frac{1}{|F(i)|}\times\frac{1}{B(k)}$$

$$A_{i,j}=\sum_{k:k\in B(i)\cap B(j)}\frac{1}{|B(i)|}\times\frac{1}{F(k)}$$

4. 求出矩阵 H,A 的主特征向量,就是对应的马尔可夫链的静态分布。

5. A 中值大的对应的网页就是所要找的重要网页。

SALSA 算法没有 HITS 中相互加强的迭代过程,计算量远小于 HITS。SALSA 算法只考虑直接相邻的网页对自身 A/H 的影响,而 HITS 是计算整个网页集合 T 对自身 AH 的影响。

实际应用中,SALSA 在扩展根集时忽略了很多无关的链接,比如:①同一站点内的链接,因为这些链接大多只起导航作用。②CGI 脚本链接。③广告和赞助商链接。

试验结果表明,对于单主题查询 java,SALSA 有比 HITS 更精确的结果,对于多主题查询 abortion,HITS 的结果集中于主题的某个方面,而 SALSA 算法的结果覆盖了多个方面,也就是说,对于 TKC 现象,SALSA 算法比 HITS 算法有更高的健壮性。

2.3.1 BFS(Backword Forward Step)算法 SALSA 算法计算网页的 Authority 值时,只考虑网页在直接相邻网页集中受欢迎的程度,忽略其它网页对它的影响。HITS 算法考虑的是整个图的结构,特别是经过 n 步以后,网页 i 的 Authority 的权重是 $|BF^n(i)|/|BF^n|$,$BF^n(i)$ 为离开网页 i 的 $(BF)^n$ 的路径的数目,也就是说网页 j<>i,对 i 的权值贡献等于从 i 到 j 的 $(BF)^n$ 路径的数量。如果从 i 到 j 包含有一个回路,那么 j 对 i 的贡献将会呈指数级增加,这并不是算法所希望的,因为回路可能不是与查询相关的。

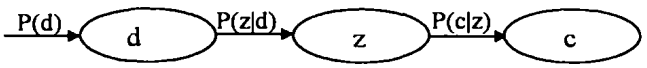
因此,Allan Borodin 等[11]提出了 BFS(Backward Forward Step)算法,既是 SALSA 的扩展情况,也是 HITS 的限制情况。基本思想是,SALSA 只考虑直接相邻网页的影响,BFS 扩展到考虑路径长度为 n 的相邻网页的影响。在 BFS 中,$BF^n(i)$ 被指定表示能通过 $(BF)^n$ 路径到达 i 的结点的集合,这样 j 对 i 的贡献依赖于 j 到 i 的距离。BFS 采用指数级降低权值的方式,结点 i 的权值计算公式如下:

$$a_i=2^{n-1}|B(i)|+2^{n-2}|BF(i)|+2^{n-3}|BFB(i)|+\cdots\cdots+|BF^n(i)|$$

算法从结点 i 开始,第一步向后访问,然后继续向前或者向后访问邻居,每一步遇到新的结点加入权值计算,结点只有在第一次被访问时加入进去计算。

2.4 PHITS

D. Cohn and H. Chang 提出了计算 Hub 和 Authority 的统计算法 PHITS(Probabilistic analogue of the HITS)[12]。他们提出了一个概率模型,在这个模型里面一个潜在的因子或者主题 z 影响了文档 d 到文档 c 的一个链接,他们进一步假定,给定因子 z,文档 c 的条件分布 P(c|z)存在,并且给定文档 d,因子 z 的条件分布 P(z|d)也存在。



$$P(d,c)=P(d)\times P(c|d)$$

其中 $P(c|d)=\sum_z P(c|z)\times P(z|d)$

根据这些条件分布,提出了一个似然函数(likelihood function)L:

$L=\prod_{(d,c)\in M}P(d,c)$,M 是对应的连结矩阵

然后,PHITS 算法使用 Dempster 等提出的 EM 算法[20]分配未知的条件概率使得 L 最大化,也就最好地解释了网页之间的链接关系。算法要求因子 z 的数目事先给定。Allan Borodin 指出,PHITS 中使用的 EM 算法可能会收敛于局部的最大化,而不是真正的全局最大化[11]。D. Cohn 和 T. Hofmann 还提出了结合文档内容和超链接的概率模型[13]。

2.5 贝叶斯算法

Allan Borodin 等提出了完全的贝叶斯统计方法来确定 Hub 和 Authoritive 网页[11]。假定有 M 个 Hub 网页和 N 个 Authority 网页,可以是相同的集合。每个 Hub 网页有一个未知的实数参数 e_i,表示拥有超链的一般趋势,一个未知的非负参数 h_i,表示拥有指向 Authority 网页的链接的趋势。每个 Authoritive 网页 j,有一个未知的非负参数 a_j,表示 j 的 Authority 的级别。

统计模型如下,Hub 网页 i 到 Authority 网页 j 的链接的先验概率如下给定:

$P(i,j)=\mathrm{Exp}(a_j h_i+e_i)/(1+\mathrm{Exp}(a_j h_i+e_i))$

Hub 网页 i 到 Authority 网页 j 没有链接时,$P(i,j)=1/(1+\mathrm{Exp}(a_j h_i+e_i))$

从以上公式可以看出,如果 e_i 很大(表示 Hub 网页 i 有很高的趋势指向任何一个网页),或者 h_i 和 a_j 都很大(表示 i 是个高质量 Hub,j 是个高质量的 Authority 网页),那么 i→j 的链接的概率就比较大。

为了符合贝叶斯统计模型的规范,要给 2M+N 个未知参数 (e_i,h_i,a_j) 指定先验分布,这些分布应该是一般化的,不提供信息的,不依赖于被观察数据的,对结果只能产生很小影响的。Allan Borodin 等在文[11]中指定 e_i 满足正态分布 $N(\mu,\delta^2)$,均值 $\mu=0$,标准方差 $\delta=10$,指定 h_i 和 a_j 满足 Exp(1)分布,即 $x>=0$,$P(h_i>=x)=P(a_j>=x)=\mathrm{Exp}(-x)$。

接下来就是标准的贝叶斯方法处理和 HITS 中求矩阵特征根的运算。

2.5.1 简化的贝叶斯算法 Allan Borodin 同时提出了简化的上述贝叶斯算法,完全除去了参数 e_i,也就不再需要正态分布的参数 μ,δ 了。计算公式变为:$P(i,j)=a_j h_i/(1+a_j h_i)$,Hub 网页到 Authority 网页 j 没有链接时,$P(i,j)=1/(1+a_j h_i)$。

Allan Borodin 指出简化的贝叶斯产生的效果与 SALSA

算法的结果非常类似。

2.6 Reputation

上面的所有算法，都是从查询项或者主题出发，经过算法处理，得到结果网页。多伦多大学计算机系 Alberto Mendelzon，Davood Rafiei 提出了一种反向的算法，输入为某个网页的 URL 地址，输出为一组主题，网页在这些主题上有声望(reputation)[16]。比如输入 www.gamelan.com，可能的输出结果是"java"，具体的系统可以访问 htpp://www.cs.toronto.edu/db/topic。

给定一个网页 p，计算在主题 t 上的声望，首先定义2个参数，渗透率 $P_p(t)$ 和聚焦率 $F_t(p)$，为简单起见，网页 p 包含主题项 t，就认为 p 在主题 t 上。

$$P_p(t)=I(p,t)/N(t) \quad F_t(p)=I(p,t)/In(p)$$

$I(p,t)$ 是指向 p 而且包含 t 的网页数目，$In(p)$ 是指向 p 的网页数目，$N(t)$ 是包含 t 的网页数目。结合非条件概率，引入 $L(p)=In(p)/N_w$，$M(T)=n(T)/N_w$，N_w 是 Web 上网页的数目。P 在 t 上的声望计算如下：

$$RM(p,t)=(P_p(t)-L(p))/L(p)$$
$$=(F_t(P)-M(t))/M(t)$$

指定 $LM(p,t)$ 是既指向 p 又包含 t 的概率，即 $LM(p,t)=I(p,t)/N_w$，显然有

$$RM(p,t)=LM(t,p)/[L(p)\times M(t)]-1$$

我们可以从搜索引擎(如 Altavista)的结果得到 $I(p,t)$，$N(t)$，$In(p)$，Web 上网页的总数估计值 N_w 某些组织会经常公布，在计算中是个常量不影响 RM 的排序，RM 最后如此计算：$RM(p,t)=[N_w\times I(t,p)]/[In(p)\times N(t)]-1$

给定网页 p 和主题 t，RM 可以如上计算，但是多数的情况只给定网页 p，需要提取主题后计算。算法的目标是找到一组 t，使得 RM(p,t)有较大的值。TOPIC 系统中是抽取指向 p 的网页中的锚文本的单词作为主题(上面已经讨论过锚文本能很好描述目标网页，精度很高)，避免了下载所有指向 p 的网页，而且 RM(p,t)的计算很简单，算法的效率较高。主题抽取时，还忽略了用于导航、重复的链接的文本，同时也过滤了停止字(stop word)，如"a"，"the"，"for"，"in"等。

Reputation 算法也是基于随机漫游模型的(random walk)，可以说是 PageRank 和 SALSA 算法的结合体。

3. 链接算法的分类及其评价

链接分析算法可以用来提高搜索引擎的查询效果，可以发现 WWW 上的重要的社区，可以分析某个网站的拓扑结构，声望，分类等，可以用来实现文档的自动分类等。归根结底，能够帮助用户在 WWW 海量的信息里面准确找到需要的信息。这是一个正在迅速发展的研究领域。

上面我们从历史的角度总结了链接分析算法的发展历程，较为详细地介绍了算法的基本思想和具体实现，对算法存在的问题也做了讨论。这些算法有的处于研究阶段，有的已经在具体的系统实现了。这些算法大体可以分为3类，基于随机漫游模型的，比如 PageRank，Reputation 算法，基于 Hub 和 Authority 相互加强模型的，如 HITS 及其变种，基于概率模型的，如 SALSA，PHITS，基于贝叶斯模型的，如贝叶斯算法及其简化版本。所有的算法在实际应用中都结合传统的内容分析技术进行了优化。一些实际的系统实现了某些算法，并且获得了很好的效果，Google 实现了 PageRank 算法，IBM Almaden Research Center 的 Clever Project 实现了 ARC 算法，多伦多大学计算机系实现了一个原型系统 TOPIC，来计算指定网页有声望的主题。

AT&T 香农实验室的 Brian Amento 指出，用权威性来评价网页的质量和人类专家评价的结果是一致的，并且各种链接分析算法的结果在大多数的情况下差别很小[15]。但是，Allan Borodin 也指出没有一种算法是完美的，在某些查询下，结果可能很好，在另外的查询下，结果可能很差[11]。所以应该根据不同查询的情况，选择不同的合适的算法。

基于链接分析的算法，提供了一种衡量网页质量的客观方法，独立于语言，独立于内容，不需人工干预就能自动发现 Web 上重要的资源，挖掘出 Web 上重要的社区，自动实现文档分类。但是也有一些共同的问题影响着算法的精度。

1. 根集的质量。根集质量应该是很高的，否则，扩展后的网页集会增加很多无关的网页，产生主题漂移，主题泛化等一系列的问题，计算量也增加很多。算法再好，也无法在低质量网页集找出很多高质量的网页。

2. 噪音链接。Web 上不是每个链接都包含了有用的信息，比如广告、站点导航、赞助商、用于友情交换的链接，对于链接分析不仅没有帮助，而且还影响结果。如何有效地去除这些无关链接，也是算法的一个关键点。

3. 锚文本的利用。锚文本有很高的精度，对链接和目标网页的描述比较精确。上述算法在具体的实现中利用了锚文本来优化算法。如何准确充分地利用锚文本，对算法的精度影响很大。

4. 查询的分类。每种算法都有自身的适用情况，对于不同的查询，应该采用不同的算法，以求获得最好的结果。因此，对于查询的分类也显得非常重要。

当然，这些问题带有很大的主观性，比如，质量不能精确地定义，链接是否包含重要的信息也没有有效的方法能准确地判定，分析锚文本又涉及到语义问题，查询的分类也没有明确界限。如果算法要取得更好的效果，在这几个方面需要继续做深入的研究，相信在不久的将来会有更多的有趣和有用的成果出现。

参考文献

1 Page L, Brin S, Motwani R, Winograd T. The PageRank Citation Ranking : Bringing Order to the WEB. Jan 1998 and July 2001 at http://www.db.stanford.edu/~backub/PageRanksub.ps

2 Brin S, Page L. The anatomy of a large-scale hypertextual WEB search engine. In: Proc. of the Seventh Intl. World Wide WEB Conf. 1998

3 Richardson M, Domingos P. The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank, volume 14. MIT Press, Cambridge, MA, 2002

4 Haveliwala T H. Topic-Sensitive PageRank. In: Proc. of the Eleventh Intl. World Wide WEB Conf. 2002

5 Kleinberg J. Authoritative sources in a hyperlinked environment. In: Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, 1998. Extended version in Journal of the ACM 46(1999). Also appears as IBM Research Report RJ 10076, May 1997

6 Chakrabarti S, et al. Hypersearching the WEB. Scientific American, June 1999

7 Henzinger M R, Bharat K. Improved algorithms for topic distillation in a hyperlinked environment. In: Proc. of the 21'st Intl. ACM SIGIR Conf. on Research and Development in IR, Aug. 1998

(下转第140页)

属性为文件全名流。

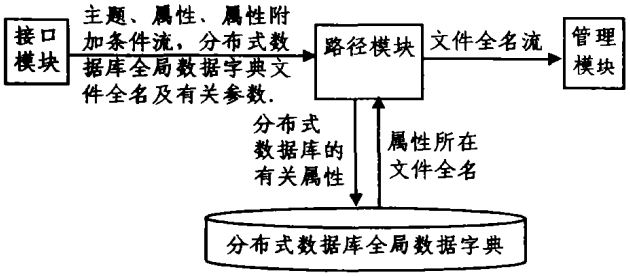


图5

3.2.5 管理模块 如图6所示。管理模块根据接口模块提供的主题、属性及属性附加条件流和路径模块提供的文件全名流，负责创建和发送 aglet 到远程主机，接收局部主机的 aglet 和 aglet 的检索和加工结果，并将检索和加工结果信息装入数据仓库的关系表中。管理模块连续传输 aglet，通过连续传输在目标主机形成它的副本。一收到 aglet，管理模块就重构 aglet 并以它作为参考对象，产生它的执行文本。

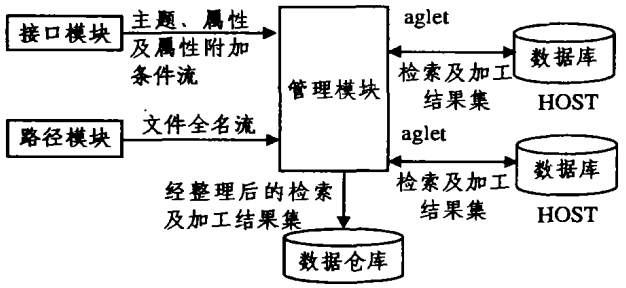


图6

1)aglet。是一个可移动的 Java 对象，它能访问计算机网络中具有 aglet 使能的主机。它根据用户的要求在分布式数据库有关站点完成信息检索和加工，并把结果反馈给管理模块。它有自主性，因为它抵达某一主机后运行它自身的执行线程；它有反应性，能对到来的消息作出反应。

2)identifier。是 aglet 的标识符，每个 aglet 与一个 identifier 相关联。aglet 的标识符是全球唯一，终身不变的。

4 实验系统设计和结果分析

我们利用 ORACLE8i 采用重构法建立了一个分布式考生信息数据库实验原型。按照定义的格式，通过数据仓库设计系统图形工具定义主题、属性及属性附加条件流，我们一一验证了构造数据仓库的九条规则，并进行了组合测试，建立数据仓库移动代理系统都能正确运行，并能产生准确结果。

结束语 利用数据仓库进行决策支持已成为决策支持系统开发者关注的热点，关键是如何直接为决策支持系统开发者提供一个操作简单、效率较高的基于分布式数据库建立数据仓库的系统工具。本文在研究数据仓库设计概念和理论的基础上，提出了以层次模型构建元数据，提出了以元数据为核心的建立数据仓库的九条规则，并构建了数据仓库设计系统图形工具原型和建立数据仓库移动代理系统原型。通过这个系统创建数据仓库，不仅可以免去开发者去学习一些繁杂的概念、理论，而且还能为开发者节约大量的时间和精力。甚至，有兴趣的决策者自身也可以通过这个系统构建数据仓库。

参 考 文 献

1 Lange D B, Oshima M. Programming and Deploying Java Mobile Agents with Aglets. Addison-Wesley, 1998

2 http://www.cs.dartmouth.edu/~agent

3 Aglets Software Development Kit(asdk). http://aglets.trl.ibm.co.jp/

4 邵佩英. 分布式数据库系统及其应用. 科学出版社, 2000

5 [美]Giovinazzo W A, 著. 潇湘工作室译. 面向对象数据仓库设计. 人民邮电出版社, 2000

6 袁鹏飞. Oracle 8i 数据库高级应用开发技术. 人民邮电出版社, 2000

(上接第93页)

8 Lempel R, Moran S. The Stochastic Approach for Link-Structure Analysis (SALSA) and the TKC Effect. In: Porc. 9 th Intl. World Wide WEB Conf. 2000

9 Chakrabarti S, et al. Mining the WEB's link structure. IEEE Computer, Aug. 1999

10 Chakrabarti S, et al. Automatic resource compilation by analyzing hyperlink structure and associated text. In: Proc. 7th Intl. WWW Conf. 1998

11 Borodin A, et al. Finding Authorities and Hubs From Link Structures on the World Wide WEB. In: Proc. 10th Intl. WWW Conf. 2001

12 Cohn D, Chang H. Learning to probabilistically identify authoritative documents. In: Proc 17 th Intl. Conf. on Machine Learning, 2000

13 Cohn D, Hofmann T. The Missing Link—A Probabilistic Model of Document Content and Hypertext Connectivity. Advances in Neural Information Processing Systems (NIPS), 2000, 13

14 Baeza-Yates R, Ribeiro-Neto B. Moderm Information Retrieval. Addison Wesley, New York, NY, USA, 1999

15 Amento B, Terveen L, Hill W. Does "Authority" Mean Quality? Predicting Expert Quality Ratings of WEB Documents. In: 23rd Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval, 2000

16 Mendelzon A, Rafiei D. What do the Neighbours Think? Computing WEB Page Reputations IEEE Data Engineering Bulletin, 2000, 23(3): 9~16

17 韩家炜, 孟小峰, 王静, 李盛恩. WEB 挖掘研究. 计算机研究与发展, 2001, 38(4)

18 Google Inc. Google search engine. http://www.Google.com

19 Topic system. htpp://www.cs.toronto.edu/db/topic

20 Dempster A, Laird N, Rubin D. Maximun likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B, 1977, 39: 1~38

21 IBM Almaden Research Center Clever Project. http://www.almaden.ibm.com/cs/k53/clever.html

22 Arasu A, et al. PageRank Computation and the Structure of the WEB: Experiments and Algorithms. In: 11th Intl. World Wide WEB Conf. 2002