一种改进 PageRank 的新方法

A New Way to Improved PageRank

李村合　吕克强　（中国石油大学(华东) 计算机与通信工程学院 山东东营 257061）

摘　要： PageRank 是 Web 主题检索最成功的算法之一，但它同时也存在一些问题。PageRank 算法仅仅考虑了 Web 的链接结构，并没有考虑链接所携带的内容信息。针对这种情况，本文提出了根据链接临近文本信息对 PageRank 进行主题矫正计算的方案，最终使用 PageRank 与主题矫正值的和替换整最初的 PageRank。模拟实验结果表明，改进后的算法可以提高 PageRank 算法的查全率。

关键词： PageRank　链接分析　链接文本　搜索引擎　主题漂移

1　引言

随着互联网的高速发展，从 Web 可以获取的信息也在急剧增加。截止目前，Google 检索的页面数已经达到 80 多亿，然而 Google 能检索的网页也仅仅是 Web 的一部分。面对如此浩瀚的信息，越来越多的用户开始用搜索引擎查找他们所需要的信息。

Web 检索算法是搜索引擎的核心技术之一。目前最为有效且应用于实际的排序算法都使用了 Web 页面的超链接结构。其中最著名的是 1998 年 Sergey Brin 和 Lawrence Page 提出的 PageRank 算法[1] 和同年的 Kleinberg 提出的 HITS 算法[2]。与 HITS 算法相比，PageRank 算法独立于主题查询词、计算不占用用户的响应时间、有更好的健壮性，更适合于大规模的 Web 检索。该算法被应用于 Google 搜索引擎中，并取得了成功。

但 PageRank 算法并非十全十美，也存在一些缺陷，最明显的就是主题漂移问题[3]。本文从结合 Web 中超链接文本信息角度，提出了一种新的方法来避免主题漂移现象，提高搜索结果的查全率。

2　PageRank 算法简介

PageRank 算法的最初定义如下：把整个 Web 看成一个有向图 G = (V,E)，其中的 V 是节点(网页)集，E 是有向边的集合(当且仅当存在从页面 u 到页面 v 的链结时存在从节点 u 到节点 v 的边)。则一个页面的重要程度是由指向它的页面的重要程度和指向它的页面链接数量决定的。由其初始定义如下：

$$PR(v) = c \sum_{(u,v)\in E} \frac{PR(u)}{N_u} \quad (1)$$

公式(1)中 N_u 表示节点 u 的出度，而所有 Web 页面的链接情况可以用一个 n × n 的邻接矩阵 A 来表示，当页面 i 到页面 j 有链接时，$a_{ij}=1$；当 i = j 或者当页面 i 到页面 j 无链接时，$a_{ij}=0$。矩阵 M 是将矩阵 A 转置后将各元素除以各行非零元素的和后得到的。M 可以表示为：$m_{ij} = a_{ji} / \sum_{k=1}^{n} a_{jk}$，若向量 R 表示 n 个页面 PageRank 值，根据公式(1)，R = cMR 的关系成立(c 为定量)。这样，R 可以看作 M 矩阵的特征向量。具体计算时，可以给每个网页一个初始的 PageRank 值，然后反复迭代运算，即：

$$PR_{i+1} = cM * PR_i \quad (3)$$

公式(3)中只有当 M 为强连通矩阵时，迭代运算才收敛。然而网页结构并非表现为一个完全牢固的链接图，也就是说不是所有的网页都可以从其他网页通过超链来达到，而 PageRank 值的计算正依赖于此，如果整个网页图中的一组紧密链接的网页如果没有外出的链接就产生等级沉没(Rank Sink)，一个独立的网页如果没有外出的链接就产生等级泄漏(Rank Leak)。所以 Page 等人就提出了下面的改进方案，对上述存在的两个问题进行了有效的解决：采用一种“用户在大多数情况下都顺着当前页面中的链接前进，但偶尔也会

跳跃到完全无关的页面里"的浏览模型。并将"偶尔"固定为 15% 来计算。也就是说用户在 85% 的情况下沿着链接前进,但在 15% 的情况下会突然跳跃到无关的页面中去。这样推移概率矩阵可以表示成:

$$M' = cM + (1-c) * [1/N] \quad (4)$$

其中,[1/N]是所有要素为 1/N 的 N 阶方阵,c = 0.85(=1 -0.15)。

根据上述的变形,原先求矩阵 M 的特征向量问题变成了求矩阵 M'的最大特征值对应特征向量的问题。此时迭代形式如下:

$$PR_{i+1} = cM * PR_i + (1-c) * [1/N]_{n*1} \quad (5)$$

这样 PageRank 的定义由公式(1)变为:

$$PR(v) = (1-c) * (1/N) + c \sum_{(u,v)\in E} \frac{PR(u)}{N_u} \quad (6)$$

3 PageRank 算法的改进策略

从公式(6)看出, PageRank 值的迭代计算过程,实际上就是页面 u 把自己的 PageRank 值 PR(u)值平均分给了它指向的 N_u 个页面,页面上所有的链接都看作是等价的。在实际的网上冲浪过程中,用户在浏览页面的时候带有目的性和兴趣性。用户的目的和兴趣集中体现在主题词上。因此,我们考虑使用主题词来指导 PageRank 值的修正。目前使用了主题词有文献[4]的智能冲浪算法,但是该算法对每个主题词都要进行一次迭代计算,对于海量的网上信息意义不明显。超链接 u→v 所携带的信息能够体现网页 u 对 v 的评价,而且这种评价比较客观。我们考虑在 PR 计算完成后使用超链接 u→v 所携带的信息对页面 v 的 PR 值进行修正。我们可以定义 $PR_q(v)$ 为在查询此为 q 的消减下页面 v 的 PR 修正值。

我们根据以下三个前提条件对 PR 进行修正。

条件 1:超链接 u→v 所携带的信息能够体现网页 u 对 v 的评价[5];

条件 2:查询结果页面排序是依赖于主题词的[2];

条件 3:标准 PR(v)值体现了网页 v 全局重要重要性,原因在于标准 PR(v)值计算过程中没有依赖于任何查询词。而修正值 $PR_q(v)$ 体现了网页 v 相对于主题词 q 重要性。

设 L 为所有超链接的集合,$L = \{L_1, L_2 \cdots L_n\}$, L 中的任意元素 L_i 为一个三元组(sor,des, text),其中 sor 为链接 L_i 的起始页面,des 为链接 L_i 的目标页面,text 为 L_i 链接附近的信息。对于主题词 q, 可以计算出 text 中包含主题词 q 的链接集合,记为 $L_q = \{L_{q1}, L_{q2} \cdots L_{qn}\}$, $Count(L_q)$ 为 L_q 中元素的个数。对于任意页面 v,可以获得指向页面 u 的集合 $L_v = \{L_{v1}, L_{v2} \cdots L_{vn}\}$, $Count(L_v)$ 为 L_v 中元素的个数。我们可以求集合 L_q 和 L_v 的交集 $L_q \cap L_v$, $Count(L_q \cap L_v)$ 为 $L_q \cap L_v$ 中元素的个数。显然 $Count(L_q \cap L_v) \leq Count(L_v)$ 且 $Count(L_q \cap L_v) \leq Count(L_q)$。我们设

$$k_q = (Count(L_q \cap L_v)/Count(L_q)) * (Count(L_q \cap L_v)/Count(L_v)) \quad (7)$$

k_q 为页面 v 的链接敏感变量。所有指向页面 v 的链接中包含词 q 的链接数量越多 k_q 越大;所有包含 q 的链接中指向 v 的页面比例越大 k 越大。一下面的一个简单的 Web 图例说明一下:

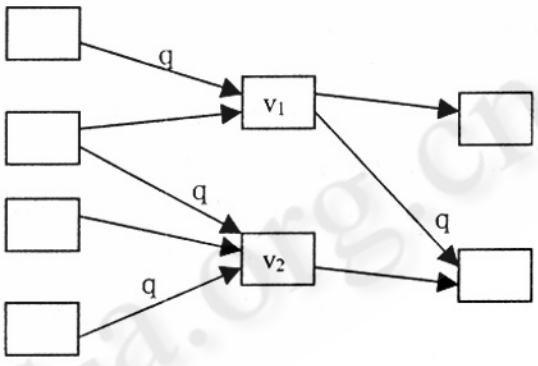

图 1　简单 Web 示意图

Web 内的链接的数量为 8,与主题词 q_i 相关的链接的数量为 4。指向 v1 的链接的水量为 2,指向链接 v1 且包含 q 的链接的数量为 1;指向 v2 的链接的水量为 3,指向链接 v2 且包含 q 的链接的数量为 2。

$$k_q(v1) = (Count(L_q \cap L_v)/ Count(L_q)) * (Count(L_q \cap L_v)/ Count(L_v)) = (1/4) * (1/2) = 1/8$$

$$k_q(v2) = (Count(L_q \cap L_v)/ Count(L_q)) * (Count(L_q \cap L_v)/ Count(L_v)) = (2/4) * (2/3) = 1/3$$

页面 v2 的对主题词 q 链接敏感变量 kq2 比页面 v1 的对主题词 q 链接敏感变量 kq1 要大一些。

根据前提 3,可以得出 $PR_q(v) = PR(v) * f(k_q)$, f(k)为 k_q 在[0,1]上的递增函数。可以令 $f(k_q) = k_q$。

$PR_q(v) = PR(v) * e^{kq}$。因此，在主题词为 q 时，页面 v 的改进的 PR 值 PR′可以用下面的公式表示：

$$PR'(v) = PR(v) + PR_q(v) = PR(v)(1 + e^{kq}) \quad (8)$$

当主题词为多个词语时，设 $Q = \{q_1, q_2, q_3 \cdots q_n\}$，可以令 $K_Q = k_{q1} + k_{q2} + \cdots + k_{qn}$然后再计算 PR′(v)。

4　PageRank 算法的改进策略

输入：网络图对应的邻接矩阵 A、超链接集合 L；

输出：改进的页面 PR′值。

步骤：

(1) 根据邻接矩阵 A 经过迭代迭代计算标准 PR；

(2) 根据查询的主题词 q，计算出包含 q 的超链接的集合 L_q。根据 L_q 计算出 L_q 中的元素的指向的页面的集合 V_q，根据公式(7)计算出每个页面的主题敏感变量 k_q；

(3) 根据步骤(2)中 k_q 计算出每个页面的 PR_q，然后与 PR 相加得出 PR′。

5　实验分析

为了验证改进的算法的查全率，我们做了模拟实验，实验方案采取文献[6]中的方法。我们开源搜索引擎 Nutch 对 http://mil.news.sohu.com/域名下的页面进行抓取分析，获取到有效网页 123,245 张。然后分别使用标准的 PageRank 和改进的 Page 分别计算标准的 PR 和改进后的 PR′，然后使用军事领域的 10 个专题词进行查询，对于查询结果分别按照 PR 和改进后的 PR′排序，计算返回结果的查全率。结果如图 2。

由模拟实验结果可得，该改进的 PageRank 算法在一定程度上提高搜索引擎返回结果的查全率。减少了查询结果的主题漂移。

6　结束语

本文从超链接的特点出发，根据链接锚文本的信息对链接进行统计，提出了对修正 PageRank 算法改进，经过模拟实验证实，可以提高搜索引擎的查全率。在 Web 检索过程中如果能够将文本和链接结构有机的综合到一起，必然可以进一步的提高准确率，此类算法应该是 Web 检索的努力方向。

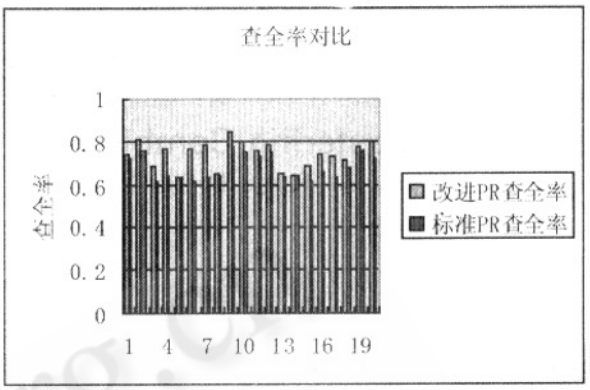

图 2　查询结果比较图

参考文献

1 Sergey Brin , Larry Page. The anatomy of a large - scale hypertextual Web search engine. Computer Networks and ISDN Systems, 1998, 30(4):107 - 117.

2 S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, S. Rajagopalan. Automatic resource compilation by analyzing hyperlink structure and associated text. Computer Networks and ISDN Systems, 1998, 30(4):65 - 74.

3 Haveliwala T H. Topic - sensitive PageRank. IEEE Transactions on Knowledge and Data Engineering, 2003, 15: pp. 784 - 796.

4 Matthew Richardson, Pedro Domingos. The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank, Advances in Neural Information Processing Systems, 2002, 14: 1441 - 1448.

5 张敏、高剑峰、马少平，基于链接描述文本及其上下文的 Web 信息检索. 计算机研究与发展，2004，41(1):221 - 226.

6 黄德才、戚华春，PageRank 算法研究，计算机工程，2006,32(4):145 - 147.