提高 PageRank 算法效率的方法初探*)

On An Improving Strategy for the PageRank Algorithm

刘 悦 程学旗 李国杰

(中国科学院计算技术研究所 北京100080)

Abstract In this paper, we research and analysis the PageRank algorithm in detail. Based on analysising the structure of the Web Graph, we propose an improving strategy for the algorithm, Reducing the iterative times of the algorithm and got a relatively higher efficiency.

Keywords PageRank algorithm, Search engine, Link analysis, Web Graph

1. 引言

Web 是人类历史上承载数据最丰富的信息库,但在 Web 上查找所需要的信息却很困难,由于 Web 的海量规模、异构、动态等特性,使得 Web 文本检索表现出更大的挑战性,广泛地引起了各方面的研究兴趣。该领域当前的流派大体可以分为:经典 IR 流派,METADATÁ 流派,数据库流派和链接分析流派。

本文所讨论的 PageRank 算法是链接分析流派中的一个典型代表,在页面质量的计算过程中采用链接分析技术,也是第二代搜索引擎的重要特点,各种算法有一个共同的基本思想:它们认为更多地被其他页面链接的页面是质量更好的页面,并且从更重要的页面出发的链接有更大的权重,这个循环定义,通过迭代算法巧妙地打破了循环,除了本文讨论的 PageRank 算法,另外一个比较著名的算法就是 IBM 的 HITS 算法。

2. PageRank 算法简介

PageRank 算法是 Standford 大学的研究人员开发的 Google 搜索引擎的页面质量评价算法。Google 的研究人员观察到在 Web 图这一有向图中,不同的结点被访问到的概率是不同的,它们为用户在 Web 上的浏览行为建立了一个模型:即以概率 d 顺着朝链接点击访问,或者以概率1−d 从一个新的页面开始访问,在该模型下,某一个页面 t 被访问到的概率只与指向它的页面的概率有关。

为了能更清楚地讨论,下面我们对 PageRank 算法本身以及算法中涉及的一些基本概念先做一个说明。

定义1 有向图 G(V;E)称为 Web 图,若 V 表示有限页面顶点集合,E 是由 V 中不同元素组成的有序对的集合,它表示页面之间的超链接:

$\forall \langle v,w\rangle \in E$

表示从页面 v 指向页面 w 的一个超链接。

定义2 设 G=(V;E)是一个 Web 图,且$\forall t\in V$,称如下集合为页面 t 的前集,记为 $^{\cdot}t$:

$\{y|\langle y,t\rangle \in E, \forall y\in V\}$。

定义3 设 G=(V;E)是一个 Web 图,且$\forall t\in V$,称如下集合为页面 t 的后集,记为 $t^{\cdot}$:

$\{y|\langle t,y\rangle \in E, \forall y\in V\}$

定义4 $\forall t\in V$,记$|{}^{\cdot}t|$为 t 的前集中的元素的个数。$|t^{\cdot}|$为 t 的后集中的元素的个数。

在 PageRank 算法中页面 t 被访问到的概率 Pr(t)(通常称为 rank 值)是按照下式给出的:

$$Pr(t)=(1-d)/MAX+d(\Sigma(Pr(t_i))/|t_i^{\cdot}|) \quad (1)$$

其中 $t_i\in {}^{\cdot}t$,MAX 为页面的总数,d 称为影响因子。它一般根据经验进行指定。

概率 Pr(t)反映了 t 的重要程度,在 PageRank 算法中将之用作页面质量的评价参数,它采用的方法就是根据式(1)进行迭代运算,直到计算出的值收敛为止。

3. 对算法的改进

从上面介绍的算法思想我们可以看出它的计算过程很简单,只需要对每一个页面应用公式(1)计算一个 rank 值就可以了,由于我们所处理的 Web 图中结点的数量是巨大的(我们用 TREC Web Track 中的 WT10g 的数据做实验,就必须处理1692097个页面),而其中所涉及到的超链接的数量就更巨大了。如果只是简单地迭代,算法的效率是低下的。

通过分析数据我们对算法从以下几个方面进行了改进:

a)在进行迭代之前,对页面进行预处理,过滤掉噪音链接,标记出不参与迭代的页面。通过分析数据我们发现并不是所有的页面都存在前集和后集,对于前集为空集的页面,PageRank 算法给出的式(1)对它们是没有迭代效果的,所以这样的页面可以不参加迭代,而是直接指定一个固定的 rank 值。剩下的页面,为了加快查找的效率,按页面号进行排序。

b)对于后集中结点数量很大的结点的预处理。对于前面给出的式(1),如果$|t_i^{\cdot}|$很大的话,$Pr(t_i)/|t_i^{\cdot}|$的值就会很小,直观上讲,也就是说在 Web 图上页面 t_i 对页面 t 的 rank 值的贡献很小。所以对于这样一类结点,我们也没有必要简单地迭代,在改进的算法中,我们预先给定一个阈值,对于后集中结点数量大于阈值的结点,指定其 rank 值为1/MAX。这样进一步缩减了迭代的规模。

c)规范化和收敛性的判定规则。根据概率本身的含义,每一次迭代计算出来的结果规范化之后的 Σ 和最好为1,但是

*)本论文得到国家973课题资助(课题编号 G1998030413)。刘 悦 博士研究生,研究兴趣:海量信息处理,知识检索与数据挖掘,算法设计与分析,Petri 网理论与应用等。程学旗 副研究员,主要研究领域为:Internet 高性能软件、智能信息处理、知识检索与算法分析、信息安全计算语言学等。李国杰 中国工程院院士,博士生导师,研究员。

这样得到的每一个页面的 rank 值就非常小(小数点之后6位才有数据),不利于收敛性的判定。在改进后的算法中,为了让计算所得的结果具有可比性,又不至于太小,我们给出了如下的规范化规则:

在迭代开始时,我们已经给每一个页面指定了一个初始的 rank 值(由于它对最终的结果没有影响,因此是任意指定的)。我们将所有页面的初始 rank 值相加,得到一个常量 M。

1. 对第 i 次迭代所得的每一个页面的 rank 值求和,得到 M'

2. 令 K=M/M'

3. 对第 i 次迭代所得的每一个页面的 rank 值做如下规范化:

$rank^{(i)}[j]=k * rank^{(i)}[j]$

j=1..MAX

4. END 规范化

对于收敛性的判定,我们采用的判定性规则是:预先指定一个充分小的 δ,整个过程都在用下面的公式(2)对相邻两次的迭代结果进行比较,如果满足公式(2),就停止迭代。

$$MAX(rank^{(i)}-rank^{(i+1)})-MIN(rank^{(i)}-rank^{(i+1)})<\delta \quad (2)$$

从式(2)我们可以看出,我们对于收敛性的判定不仅仅局限于所有页面的 rank 值都有减小收敛的趋势,而且要求所得的结果波动性要小(MAX 与 MIN 的差足够小,保证了这一点)。

4. 改进后的算法

算法1:modipagerank(改进后的 pagerank 算法)

输入:待处理的 Web 图

输出:每一个页面结点都有一个 rank 值的 Web 图

step1:给每个页面指定一初始的 rank 值

Step2:按子算法1对每一个页面进行预处理

Step3:对每一个页面,利用公式(1)进行计算

Step4:根据规范化规则对计算结果进行规范化

Step5:利用公式(2)判定收敛性,若收敛,转 step6,否则,重复执行 step3,step4

Step6:END

子算法1:页面预处理算法

输入:Web 图中的每一个页面,阈值 a

输出:做了标记的 Web 图

step1:利用快速排序对 Web 图页面前集对应的邻接表 in—link 根据页面号进行排序。

Step2:利用快速排序对 Web 图页面后集对应的表 out—link 根据页面号进行排序。

Step3:对于 Web 图的每一个页面,do

1)根据其页面编号利用二分查找在页面前集对应的邻接表 in_link 中进行查找,若未找到,则标记为不参与迭代页面。

2)根据其页面编号利用二分查找在页面后集对应的表 out_link 中查找,若页面后集中结点的个数大于阈值 a,则令其 rank 值为1/MAX。

step4:END.

算法分析 设:in_link 表的规模=k;out_link 表的规模=s;n 是 Web 图的页面总数。

定理1 子算法1的时间复杂性是 O(nlogm)(m=max(k,s))。

证明:因为 step1和 step2使用的是快速排序,所以它们的时间复杂性分别为:O(k log k)和 O(s log s)。Step3中使用的是二分查找,所以查找的复杂性是 log k 和 log s。对每一个页面都要查找,所以复杂性为 n * (log k+log s)。综上所述子算法1的复杂性是 O(nlogm)。证毕。

定理2 算法1:modipagerank 的时间复杂性是 O(nm-logm)。

证明:该算法的时间复杂性是很显然的。因为对于公式(1)的计算其复杂性是 O(m log m),而对于每一个页面我们都要应用公式(1)进行计算,所以最坏情况下的复杂性为 nm log m,由于是迭代运算,所以应该乘一个常系数 α,α 是收敛是可能的迭代次数,即算法1的时间复杂性为 αnm log m,所以在最坏情况下该算法的时间复杂性为 O(nm log m)。证毕。

5. 实验结果

我们用改进的算法在 TREC Web Track 中的 WT10g 的数据上进行了实验,实验的结果表明我们的改进方案是有效的,首先在预处理阶段,经过预处理之后,in_link 和 out_link 中只保留了1/5左右的有用超链接。经过20次左右的迭代,我们就得到了波动比较小、已经收敛的运行结果。

进一步的研究工作 在算法的改进过程中,我们发现影响算法的效率的关键问题是有用超链接的选取。而 TREC Web Track 中的 WT10g 的数据集合中涉及的数据量有限,所以可以考虑:

a)扩大实验集合,增加有用超链接的提取。

b)在改进算法中,依然没有考虑指向封闭数据集合之外的超链接,它们可能对于算法效率有很大的影响。

感谢冯国臻博士在算法的实现过程中的许多有益建议。感谢 J. Kleinberg 博士提供的论文资料。

参 考 文 献

1 Page L, Brin S, Motwani R, Windograd T. The Pagerank citation ranking: Bring order to the web. 1998

2 Brin S. Page L. Google search engine. http://google. standford. edu

3 Kleinberg J. Authoritative sources in a hyperlinked environment. Proc 9th ACM-SIAM SODA, 1998

4 冯国臻. 基于结构分析的大规模 WWW 文本信息检索技术的研究:[中科院计算所博士论文]

5 Brin S, Page L. The anatomy of alarge scale hypertextual web search engine Proc 7th www, 1998

6 Knuth D E. The art of computer programming, "fundomental Algorithm"

7 Knuth D E. The art of computer programming, "sorting and searching"

8 施伯乐,等. 数据结构