

· 基金项目论文 ·

文章编号: 1000—3428(2005)06—0040—03

文献标识码: A

中图分类号: TP391;TP393

# Web结构分析算法HITS的改进及应用

李 昕, 朱永胜, 武港山

(南京大学计算机软件新技术国家重点实验室, 南京大学计算机科学与技术系, 南京 210093)

**摘 要:** 在网络环境下, 传统检索技术不可避免地存在种种不足, 而超链分析技术可以直接或间接地解决这些问题。该文在介绍网络结构的基础上, 描述了已有的HITS算法及其改进策略, 并提出了该算法的简单实现架构。HITS算法着眼于挖掘超链间的潜在语义关系, 有助于在更深层次上挖掘Web中蕴含的语义信息。

**关键词:** Web; 超链; HITS

## Improvement and Application for HITS Algorithm

LI Xin, ZHU Yongsheng, WU Gangshan

(State Key Laboratory for Novel Software Technology, Nanjing University,

Department of Computer Science and Technology, Nanjing University, Nanjing 210093)

**【Abstract】** Under the circumstances of Web, traditional retrieval technology inevitably has some drawbacks, and the technique of hyperlink analysis can solve the problem directly or indirectly. Based on the introduction of Web structure, this paper describes the existent HITS algorithm and some strategies to improve its performance, and proposes a simple structure to realize this algorithm. HITS algorithm focuses on mining the potential semantic relationship between hyperlinks, so benefits the mining of semantic information contained in Web to a deeper level.

**【Key words】** Web; Hyperlink; HITS

### 1 概述

随着网络技术的不断发展, 网络已经成为十分流行的交互式信息发布媒体。网络上的信息资源数量庞大, 变化多样, 刷新速度快, 这使得人们陷入了浩如烟海的数据中, 面对想要查找的信息往往感到无从下手。为了解决Web信息量飞速增长带来的问题, 多种多样的机器处理方法已经被开发出来帮助用户查找所需的资源<sup>[1]</sup>, 搜索引擎就是其中一种十分重要的工具。

搜索引擎的出现改变了用户访问Web的方式, 在这之前, 用户只能通过URL或其它页面中指向本页面的链接来访问一个页面。搜索引擎则根据网页中出现的词条为网页生成索引, 当用户使用搜索引擎时, 输入查询关键字, 搜索引擎将关键字与索引中的内容进行比对, 并将查询到的页面按照与关键字的相似程度排序返回给用户。搜索引擎的应用使得根据页面的内容访问页面成为可能。在目前所有对网页的访问中, 通过搜索引擎的访问占了相当大的比例<sup>[2]</sup>。

在搜索引擎技术中, 最为重要的部分是如何计算一个文档与查询关键字的相似度, 这同时决定了所查询到文档的排序结果。现有的搜索引擎大部分基于传统的文本检索技术, 通过对文档库中的文档进行抽词建立索引库, 并辅以停用词表, 分析词与词之间的逻辑关系, 从而确定文档与查询关键字的相似度。这种分析技术对传统的、较为稳定的文档库能够收到较好的效果, 而对于网络上发布的变化多样的复杂文本, 则无法得到理想的结果。

常用的搜索引擎评价标准是查准率和查全率, 由于实现技术上的原因, 现今已有的搜索引擎在这两个方面都或多或少存在一些问题。首先, Web中的信息资源数量庞大, 搜索引擎无法列出所有的资源信息, 对于搜索引擎没有列出的资源, 用户无法访问。其次, 某一页面作者可能会省略某些对检索有用的关键词。比如, 一个包含词条“轮子”与“发动机”的页面, 极有可能它的内容是与汽车相关的, 然而如果

用户查询词条“汽车”, 则该页面不会被返回。对于页面的制作者而言, “汽车”这一词条已经在指向该页面上级页面中出现, 在该页面中省略是很正常的。根本问题在于, 页面作者通常假定用户按照他们所设想的路线访问页面, 而不是通过搜索引擎访问其中的单个页面, 这就使得传统的文本检索技术在网络环境下不能够很好地发挥作用。

对Web中超链结构的分析可以直接或间接地解决这些问题。按照文献[3]中超链的定义, 一个超链有两个“端”和一个方向, 从一个“端”(链源)指向另一个“端”(链宿), 链源一般就是一个HTML文档, 链宿可以是任何Web资源。超链本身的语义信息并不明确, 它的语义往往蕴含在其互结构中, 因此超链结构分析对于Web资源的自动化处理起着重要的作用<sup>[4]</sup>。Monika R. Henzinger<sup>[5]</sup>提出, 页面作者使用超链往往是因为他们认为这样会对用户有所帮助, 一部分是出于导航的目的, 另一部分是为了提供与当前页面相关的参考内容。在后一种情况下, 超链的链宿极有可能与当前页面关于同一主题, 搜索引擎可以利用链源与链宿之间的关系来进行相关信息的检索。

超链结构分析的技术极大地增强了检索结果和查询关键字之间的相关性, 所以大部分主要的搜索引擎都声称使用了某种超链结构分析技术。现在已形成的描述网络超链拓扑结构的算法模型主要有PageRank<sup>[6, 7]</sup>、HITS<sup>[8, 9, 10]</sup>等。本文将重点放在HITS算法上, 在简要介绍HITS算法的基础上, 对如何改进算法进行探讨, 并提出简单的实现架构。

**基金项目:** 国家自然科学基金资助项目(60073030); 国家“863”计划基金资助项目(2002AA117010-10); 富士通研究开发中心资助项目

**作者简介:** 李 昕(1980—), 女, 硕士生, 主研方向: 语义网络, 信息检索; 朱永胜, 硕士生; 武港山, 副教授

**定稿日期:** 2004-03-06 **E-mail:** lixin@graphics.nju.edu.cn

## 2 HITS算法

### 2.1 HITS算法思想

Web最早是由研究项目发展起来的,其形成过程十分随意,并没有一个统一规划的结构。但是即便如此,其结构也有一定的规律可循,这就是通常所说的community,在文献[9]和文献[10]中,作者对community的定义和结构作了详细的描述。一个community具有良好的内聚性,因此community内部的页面往往具有相同或相似的主题,正确识别Web中的community对改进搜索引擎的性能有很大意义。

Community中的页面可以分为两种类型:(1)表达某一主题的页面,称为authority;(2)页面指向很多的authority,它的主要功能是把这些authority联结在一起,称为hub;而authority和hub之间相互优化的关系,即为HITS算法的基础。这两种页面具有不同的功能,对于用户而言,也具有不同的意义。如果用户希望了解一个陌生领域的研究内容,hub页面所包含的超链指向各种不同的链宿,能够提供丰富的信息;但如果用户希望查找一个具体的概念或范畴,则authority页面的定位更加准确。因此,HITS算法为每个页面引入两个权值:authority权值和hub权值,最后分别输出一组具有最大authority权值的页面和一组具有最大hub权值的页面。在文献[8]中,Kleinberg证明了该算法的数学模型收敛性,文献[4]和文献[11]也均包含该算法的描述,在此,本文将简单介绍HITS算法的思想。

可以用一张图来描述Web的结构,这张图包括一个节点集合,在一些节点之间存在有向边。取这个节点集合的子集S,S中的所有节点和节点之间的边构成了Web的子图。HITS算法的第一步就是建立这张子图,从中寻找hub和authority。首先,将查询提交给传统的搜索引擎,从搜索引擎返回的页面中选取一定数量的页面作为根集(root set),也可以称为开始集(start set)。然后,在根集的基础上生成基本集(base set),基本集中包括所有引用根集中页面和被根集中页面引用的页面。基本集中的页面和它们之间的超链构成了所要处理的子图,算法的其余部分主要是针对这个基本集进行的。

为基本集中的每一个页面p定义一个非负的authority权值 $X_p$ 和一个非负的hub权值 $Y_p$ ,如果一个页面p的 $X_p$ 较大,则它被认为是一个较好的authority;同样,如果一个页面p的 $Y_p$ 较大,则被认为是一个较好的hub。初始化时,赋给所有的 $X_p$ 和 $Y_p$ 相同的值,然后按照如下规则来计算 $X_p$ 和 $Y_p$ 。用所有指向页面p的页面q的hub权值 $Y_q$ 之和来更新 $X_p$ ,并用所有页面p指向的页面q的authority权值 $X_q$ 之和来更新 $Y_p$ 。公式如下:

$$x_p = \sum_{q \rightarrow p} y_q, \quad y_p = \sum_{p \rightarrow q} x_q \quad (1)$$

其中 $q \rightarrow p$ 表示页面q指向页面p。

可以给页面标号 $\{1, 2, \dots, n\}$ 并且定义它们的 $n \times n$ 阶邻接矩阵,如果页面i指向页面j,则矩阵中的项 $(i, j)$ 为1,否则为0。同样把所有的authority权值和hub权值定义为向量, $x = (x_1, x_2, \dots, x_n)$ ,  $y = (y_1, y_2, \dots, y_n)$ ,则式(1)的矩阵形式为

$$x \leftarrow A^T y, \quad y \leftarrow Ax \quad (2)$$

将式(2)进一步展开,可以得到

$$x \leftarrow A^T y \leftarrow A^T Ax \leftarrow (A^T A)x, \quad y \leftarrow Ax \leftarrow AA^T y \leftarrow (AA^T)y \quad (3)$$

因此向量x, y均可由式(3)经过多次迭代而得。根据线

性代数的理论,迭代序列经过标准化最终将收敛于矩阵的特征向量,即上文计算的hub权值和authority权值是页面集合的固有特征,并不是由初始向量和参数的选择决定的<sup>[1]</sup>。

### 2.2 一些对HITS算法的改进

虽然基于链接的算法可以带来很好的结果,但是由于HITS算法完全不考虑页面文本的内容,在实际应用中也出现了一定的问题,如主题漂移等。这主要是由于算法认为页面中的所有超链具有同等价值引起的,根据算法描述,只要两个页面之间存在超链,则邻接矩阵中对应的值即为1,这完全忽视了超链之间的差异,引起了算法结果的偏差。通过引入页面文本的语义信息可以解决这个问题,已经有不少研究者对算法进行了改进,在一定程度上改善了这些偏差。

Krishna Bharat和Monika R. Henzinger通过对超链引入相关权值(relevance weight)的方法来修改authority权值和hub权值的计算方法,如果超链的相关权值小于一定的阈值,则认为该超链对页面权值的影响可以忽略不计,该超链将从子图中删除<sup>[11]</sup>。此外,文献[4]中也提出了将大的hub页面分裂成较小单元的思想。一个页面中往往包含很多的链接,这些链接很可能并不是关于同一主题的。在这样的情况下,将hub页面分成连续的子集进行处理可以收到更好的效果,这些子集被称为pagelet。单个的pagelet比整个hub页面更为集中地指向一个主题,因此为每个pagelet计算一个权重可以收到更好的检索结果。而在HITS算法的应用实例Clever系统中,作者通过在超链的周围文字中匹配查询关键字并计算词频的方法来计算超链的权值,用计算出的权值来代替邻接矩阵中相应的值,从而达到引入语义信息的目的<sup>[12]</sup>。

在这些思想的基础上,并经过对其不足之处的改进,本文提出了计算超链权值的解决方案。

## 3 本文对HITS算法的改进

### 3.1 超链权值计算方案

Clever系统利用超链文字及其周围文字计算超链的权值,将页面文本的语义信息引入HITS算法,收到了一定的效果<sup>[12]</sup>。在某些情况下,超链的周围文字是对链宿页面内容的简单描述和评价,这时使用周围文字中的词频信息能够提高算法的精度。然而,网络上的页面形式十分复杂,很多时候超链周围文字无法代表链宿页面的内容,甚至与链宿页面的内容大相径庭。因此,本文认为,在使用超链周围文字的词频信息时,应该通过适当的系数控制其对超链权值影响。

本文认为,在页面的文本中,最能够代表链宿页面语义信息的是超链文字(本文仅考虑以文本为载体的超链,对以图像、动画等为载体的超链暂时不作考虑)。超链文字是超链的载体,通常可以作为链宿页面内容的标题,因而能够很好地反映链宿页面的语义信息。至于超链周围文字,虽然能够在一定程度上反映超链的语义,但在相当多的情况下,这些文字无法代表链宿页面的主题信息。如果认为超链周围文字和超链文字在评价链宿页面语义信息上具有同等的地位,反而会降低算法的准确度。本文通过引入加权系数 $\alpha$ 来控制超链周围文字在超链权值中所占的比例。

在计算超链权值时,需要将文本中的语义信息进行量化,这样才能够使语义信息这一概念具有可计算性。本文使用查询关键字在超链文字中出现的次数,即词频信息进行语义信息的量化。为了方便描述,定义从页面p指向页面q关于查询关键字k的超链权值为 $w(p, q, k)$ ,这个数值随着查询关键字在超链文字和周围文字中出现数量的增多而增大;定义

$t(k)$ 为查询关键字在超链文字中出现次数,  $st(k)$ 为查询关键字在周围文字中出现的次数, 系数 $\alpha$ 用于控制周围文字的语义信息在超链权值中的比例, 可用式(4)来计算权值 $w$ 的值:

$$w(p, q, k) = 1 + t(k) + \alpha * st(k) \quad (4)$$

其中, 系数 $\alpha$ 的值可以根据不同页面集进行调整。根据式(4)计算出的 $w$ 值是大于1的, 在迭代过程中得到的向量会不断增大。然而, 本文所关心的只是它们之间的相对大小, 而不是权值的绝对数值, 因此, 为了把结果向量的数值控制在一定范围内, 可以在每次迭代后进行标准化。

### 3.2 超链权值在HITS算法中的应用

所有超链的权值计算完成后, 就可以根据公式

$$x \leftarrow A'y \leftarrow A'Ax \leftarrow AA'x, \quad y \leftarrow Ax \leftarrow AA'y \leftarrow AA'A'y \quad (5)$$

进行迭代得到authority权值向量 $x$ 和hub权值向量 $y$ 。其中邻接矩阵 $A$ 中每一项的值是这样定义的, 如果存在超链从页面 $p$ 指向页面 $q$ , 则 $A$ 中对应项的值为 $w(p, q, k)$ , 否则对应项的值为0。经过 $n$ 次迭代后, 输出 $x$ 向量中值最大的一组页面作为authority页面,  $y$ 向量中值最大的一组页面作为hub页面, 其中结果的数量可以根据具体的应用要求定制。迭代次数 $n$ 的选择来自于矩阵特征向量的理论, 经过足够数量的迭代, 结果向量最终将收敛于矩阵 $A'A$ 的特征向量。

## 4 系统架构概述

HITS算法涉及到基本集的选取问题, 理论上应该将所有指向根集中页面和被根集中页面指向的页面全部包括进来, 然而在实际应用中, 考虑到系统开销, 对根集中每一个页面, 仅从两类页面中各选取10个生成基本集。至于根集本身, 本文取传统搜索引擎返回结果的前20个页面形成根集。下文涉及的页面集合即指这里的基本集。系统处理流程如图1所示。

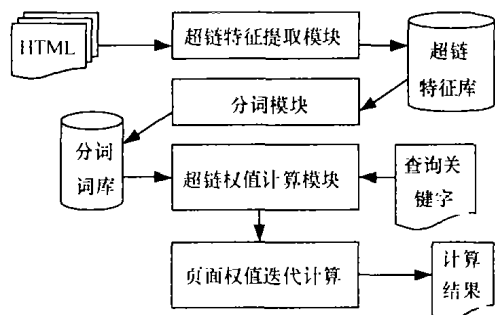


图1 系统处理流程

由图1可以看出, 系统主要包括以下几个部分:

#### (1) 超链特征提取模块

为了实现上文描述的超链权值计算方法, 需要从页面中将每个超链的信息提取出来。涉及到的信息共有以下4种: 链源页面标识, 链宿页面标识, 超链文字和超链周围文字。其中, 链源页面和链宿页面用于确定矩阵 $A$ 中的相应元素 $A(p, q)$ , 超链文字和周围文字用于计算超链权值 $w(p, q, k)$ 。

本文通过解析HTML文档来提取所需的超链特征。HTML解析的过程是将HTML文档的流式数据结构化的过程。根据HTML的语法定义, 依次对输入的HTML文档作词法分析和语法分析, HTML解析器将解析的结果以DOM树的形式输出。链源页面标识、链宿页面标识和超链文字可以通过直接分析DOM树中的节点得到。超链的上下文信息提取需要分析超链上下文窗口中包括的所有节点。上下文窗口在网页上的直观表现即超链前后的文本或者媒体块; 在DOM树上则表现为该超链节点的兄弟节点及其子树或者其

父节点的兄弟节点及其子树。窗口的大小决定上下文所能包括的字符个数, 当字符个数达到一定数量时就认为到达了窗口的边界。

经过该模块的处理, 超链权值计算所需的超链特征信息被提取出来, 形成超链特征库。

#### (2) 分词模块

对于中文文本而言, 为了获得超链文字及其周围文字中的词频信息, 需要将这些文本中的词抽取出来, 与查询关键字进行比对, 这一模块的主要功能就是从文字中抽词并确定词频信息。由于抽词的结果对算法的精度有较大的影响, 这一模块对于提高中文检索的精度十分重要。带词典的抽词算法一般具有较好的效果, 但是容易受到词典的限制, 对某些特殊领域内的特殊用词无法抽出。因此, 可以采用与不带词典的抽词算法相结合方式来克服这一缺点。

该模块的处理结果生成分词词库, 其中包括一系列的索引, 指明每个超链所对应的词频信息。

#### (3) 超链权值计算模块

通过将查询关键字与分词词库中的信息进行比对, 按照上文论述的公式计算每个超链的权值, 并生成邻接矩阵 $A$ 。本文采用的基本集中的页面个数为400个, 因此矩阵 $A$ 的维度为 $400 \times 400$ 。为了减小系统开销, 本文采用Hash表的方式存储矩阵, 对矩阵 $A$ 和 $A'$ 按行建索引, 计算 $A'A$ 和 $AA'$ , 计算出的矩阵同样以索引的方式存储。

#### (4) 页面权值计算模块

将生成的矩阵 $A'A$ 和 $AA'$ 代入, 将向量 $x$ 和向量 $y$ 中的每一项赋初值为1, 根据式(3)进行迭代, 计算出每个页面的authority权值和hub权值, 并从中选取20个具有最大authority权值的页面和20个具有最大hub权值的页面作为结果输出。实验表明, 只需要很小的迭代次数 $k$ 即可满足精度的要求, 本文取 $k$ 值为5。

采用这种系统架构, 可以从任意页面集中计算出具有最大authority权值和hub权值的页面。

## 5 结束语

Web是一个巨大的、分布广泛的、蕴含着巨量信息资源的信息服务中心, 它包含丰富和动态的超链信息, 为数据挖掘提供了资源。页面作者使用超链的方式暗示了不同页面主题之间的相关性, 通过对超链结构的分析, 能够挖掘出网络中隐藏的语义关系。传统的搜索引擎往往是基于文本检索技术, 这样对于一些自描述性比较差的页面就无法奏效, 通过对超链结构的分析可以改进这一缺点。

本文在分析网络结构的基础上, 介绍了一种已形成的描述网络超链拓扑结构的算法模型HITS, 并提出了针对其弱点的改进方案, 以及实现这一方案的系统结构。对于HITS算法而言, 还有其它的方式可以进一步改进它的精度。比如, Web中的页面通常不是关于某一主题的, 不仅是hub页面, 一些authority页面中也存在这样的情况。对一个页面计算一个权值, 则一些在页面中不占主导地位的主题内容就会被掩盖, 如果能够将页面按照其主题内容划分为若干个部分分别计算权值, 理论上应该能收到更好的效果。在今后的工作中, 可以考虑根据这些思想进一步改进算法。

### 参考文献

- 1 Kosala R, Blockeel H. Web Mining Research: A Survey. ACM SIGKDD, 2000-07

编号;

- (2)支持多用户、多版本的协同测试;
- (3)支持版本功能性能清单维护功能,实现根据条件触发系统所有满足条件记录状态的自动跃迁和邮件通知功能;
- (4)支持对自主开发的自动测试工具的测试过程和结果管理。

#### 4.2.1 系统定制

根据管理通信设备系统软件测试信息的需求,需要增加的系统字段包括涉及子系统(主相关)、涉及子系统(相关)、测试覆盖、测试工具、测试环境、系统状态、故障描述、采取措施、严重程度和用户感受等,这可以通过TestDirector提供的Customize Project Entities开放接口功能实现。

#### 4.2.2 支持版本功能性能清单维护功能的实现

版本功能性能清单的维护,主要目的是在对测试的有效管理基础上,实现系统软件功能性能清单、测试用例、故障信息和采取措施、相关用户对这些故障的反馈之间的关联,从而达到方便地对不同版本的系统功能性能清单进行维护,同时也需要提供一定的安全性,确保用户不会看到机密的信息。在实现上可以将测试需求同产品功能性能清单相统一,从而减小维护的工作量,系统状态转换功能如图5所示。

- (1)等待测试部、开发部人员操作:当版本负责人触发了功能性能维护清单之后,系统应自动将状态转换到等待测试部开发部归口人员操作的状态;
- (2)等待版本负责人确认:功能性能清单相应条目的数据填写完毕后提交,系统应自动将状态转换到等待版本负责人确认并关闭相应条目的状态;
- (3)确认并关闭相应条目系统状态转换到完成;
- (4)整个过程邮件的自动通知功能。

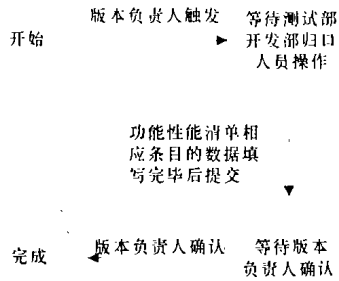


图5 系统状态跃迁

开发的外挂功能性能清单维护程序界面如图6所示。

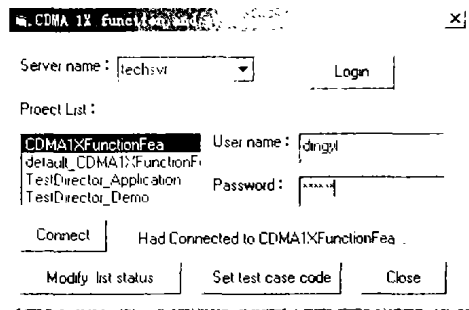


图6 功能性能清单维护外挂程序界面

#### 4.2.3 与自己开发的自动测试工具关联实现

在测试通信设备系统软件时,使用的测试方法通常有两种:手工测试和自动测试。TestDirector直接支持Winrunner自动测试工具。但测试通信设备系统软件时,通常使用很多自己开发的自动测试工具。所以需要解决对自主开发的自动测试工具的管理和关联问题。基于TestDirector提供的TestType方法可以实现关联自动测试工具,使用Visual Basic6.0实现了与自己开发测试工具的自动关联。

### 5 结论语

目前已经实现了通信设备系统软件测试信息管理系统,并已经投入到CDMA系列产品系统软件和网管系统软件测试过程的实际应用中,通过使用该系统,有效地将测试需求,测试计划,测试用例和测试故障等信息管理起来,在测试需求,测试计划和测试用例的复用和提高测试效率方面取得了明显成效。由于时间关系,目前系统还只是能对通信设备系统软件测试信息进行管理,由于在研究体系结构时考虑了硬件测试管理的情况,下一步将扩大测试管理的范畴,将硬件部分也纳入整个测试信息管理体系。从而形成一个更完善的通信设备系统测试信息管理系统。

#### 参考文献

- 1 Boehm B W. Software Engineering Economics. Prentice-Hall, Englewood Cliffs, NJ, 1981
- 2 TestDirector®.Open Test Architecture Guide 7.2.Mercury Interactive Corporation,2001
- 3 Bedell, Paul. 无线通信速成教程. 北京:人民邮电出版社,1999

(上接第42页)

- 2 Mizuuchi Y, Tajima K. Finding Context Paths for Web Pages. In Proc. of ACM Hypertext, 1999-02: 13-22
- 3 HTML4.01 Specification. W3C Recommendation, 1999-12-24
- 4 Chakrabarti S, Dom B E, Gibson D, et al. Mining the Link Structure of the World Wide Web. IEEE Computer, 1999,32(8)
- 5 Henzinger M R. Hyperlink Analysis for the Web. IEEE Internet Computing, 2001,(1):45-50
- 6 Brin S, Page L. The Anatomy of a Large-scale Hypertextual Web Search Engine. In Proc. of WWW8, Brieman(Australia), 1998-04: 107-117
- 7 Page L, Brin S, Motwani R, et al. The PageRank Citation Ranking: Bringing Order to the Web. Stanford Digital Libraries Working Paper, 1998
- 8 Kleinberg J. Authoritative Sources in a Hyperlinked Environment.

- Proc. of 9th ACM SIAM Symposium on Discrete Algorithms. Also Appeared as IBM Research Report RJ 10076, 1997-05
- 9 Kleinberg J, Lawrence S. The Structure of the Web. Science, 2001, 294: 1849-1850
- 10 Flake G W, Lawrence S, Giles C L. Efficient Identification of Web communities. In Proc. of the Sixth International Conference on Knowledge Discovery and Data Mining(ACM SIGKDD-2000), 2004:150-160
- 11 Bharat K, Henzinger M R. Improved Algorithms for Topic Distillation in a Hyperlinked Environment. In Proceedings of the ACM-SIGIR, 1998
- 12 Chakrabarti S, Dom B, Gibson D, et al. Automatic Resource List Compilation by Analyzing Hyperlink Structure and Associated Text. In Proc. of the 7th Int. World Wide Web Conference, 1998-05