# 基于模糊关系合成的 HITS 算法的改进

杨雯迪，周 军

(辽宁工业大学 电子与信息工程学院，辽宁 锦州 121000)

**摘 要**：从 Web 结构挖掘的角度出发，比较了基于链接结构分析的 PageRank 和 HITS 2 个经典算法，针对 HITS 单纯利用链接，忽略主题相关性问题，利用模糊关系的合成，得到页面与查询词之间的模糊隶属关系，对原有的 HITS 算法进行改进。实例验证了算法的有效性。

**关键词**：Web 结构挖掘；PageRank；HITS；模糊关系合成

**中图分类号**：TP393 **文献标识码**：A

## Improvement of HITS based on composition of fuzzy relation

YANG Wen Di, ZHOU Jun

(School of Electronic and Information Engineering, Liaoning University of Technology, Jinzhou 121000, China)

**Abstract**: From the direction of Web structure mining, compares the authoritative algorithms based on Web hyperlink structure——Pagerank and hyperlink-induced topic search. For HITS only use of a simple link, ignore the relevance of the theme. Composition of fuzzy relation is used for improving the original HITS algorithm.Examples verify the effectiveness of the algorithm.

**Key words**: Web structure mining; PageRank; hyperlink induced topic searcha; composition of fuzzy relation

## 1 Web 结构挖掘

随着社会的发展，Internet 已经成为世界上最丰富最广泛的信息来源并不断地更新、发展壮大。目前人们已经开始对网络中的资源进行挖掘，希望从中提取出有利于用户使用、学习的知识。因此，Web 挖掘成为一个影响 Internet 长期发展的关键手段。如何能既快又好地发现有用的知识，方便用户使用，是目前众多学者研究的热点问题。Web 挖掘是指从大量的 Web 文档集合中发现蕴涵的、未知的、有潜在应用价值的、非平凡的模式。它所处理的对象包括：静态网页(文字、多媒体信息等)、Web 数据库、Web 页面的内部结构、Web 结构、用户使用记录等信息。通过对这些信息的挖掘，可以得到仅通过文字检索所不能得到的信息。通常根据处理对象的不同，可以把 Web 挖掘分为内容挖掘、使用挖掘和结构挖掘 3 种类型[1]。本文从 Web 结构挖掘入手，提出了一种用于页面搜索的 HITS 的改进算法。首先找出查询词与词项间的模糊矩阵和网页与词项间的模糊矩阵；其次利用模糊关系合成思想，转化成网页与查询词的关系矩阵，计算页面权重，给出新的改进算法；最后举例说明算法的有效性。

Web 结构挖掘是 Web 挖掘三大分支之一[1]，World Wide Web 由许许多多的 Web 站点构成，而每个 Web 站点又包含许多 Web 页，Web 结构所包含的信息有[2-3]：(1) URL 字符串中的目录路径结构信息；(2) 网页内部内容可以用 HTML、XML 表示成的树形结构；(3) 网页之间的超链接结构。结构挖掘主要是从 Web 组织结构和链接关系中推导信息和知识，如哪些页面被其他页面所链接、哪些页面指向了其他页面等。其实，网络可以看作是一个巨型的有向图，$G=(V,E)$。节点 $V$ 代表 Web 页面，有向边 $E$ 代表一个链接。Web 页之间的超链接包含了许多有用的信息。当存在一个超链接 A 到 B 时，则说明网页 A 认为网页 B 的内容非常重要，且 2 个网页的内容具有相似性的主题。如果大量的链接都指向了同一个网页，认为它就是一个权威页(Authority)。如果 1 个 Web 页

* 基金项目：辽宁省教育厅科学研究基金(20060409)；
辽宁省高校优秀青年骨干教师基金

链接了许多网页，就称它是一个中心页(Hub)。

## 2 经典算法

利用超链进行挖掘的2个经典算法是PageRank算法和HITS算法，这2种方法是运用权威页和中心页的思想提出的。

### 2.1 PageRank算法

该算法是最早利用超链接信息计算页面的权威性来进行Web页挖掘，由Stanford大学的Brin和Page提出的[4]，搜索引擎Google就是利用该算法和链接文本标记、词频统计等因素相结合的方法对检索出的大量结果进行相关度排序，将权威值高的网页尽量排在前面。PageRank算法的基本思想是：忽略掉Web页面上的文本和其他内容，只考虑页面间的超链接，将Web对应成有向图，$F_i$是页面$i$指向的所有页面的集合，$B_i$是指向页面$i$的所有页面的集合。则页面$i$的等级PageRank值$PR(i)$可以通过以下2步计算得出：(1)以概率$(1-d)$随机取Web上任一页面$i$；(2)以概率$d$随机取指向当前页面$i$的页面$j$。则PageRank算法的具体迭代公式为：

$$PR(j)=(1-d)+d\sum_{i\in B_i}\frac{PR(i)}{|F_i|} \tag{1}$$

式中，$d$是0~1之间的衰减因子，$d$通常被设置为0.85。

PageRank算法的实现过程为：将网页的URL对应成唯一的整数，把每一个超链接用其整数ID存放到索引数据库中，经过预处理之后，设每个网页的初始$PR$值为1，通过以上的递归算法计算每一个网页的PageRank值，反复进行迭代，直至结果收敛。PageRank值越大，该页面权威性越高。该算法与用户查询条件无关，只是给出每一页面的等级PageRank值，作为搜索引擎结果排序的一个参考，等级越高的页面排序越靠前。

### 2.2 HITS算法

由Cornell大学的Kleinberg提出了一种更为精确的关于页面权威性的算法[5]，通常认为：好的Hub是指向许多好的权威页面；好的权威页是指由许多好的Hub所指向的页面；利用Hub/Authority方法来搜索网页。

HITS算法过程：

(1)将查询$q$提交给搜索引擎，搜索引擎返回很多页面，从中取前$n$个页面作为根集(Root set)，用$s$表示。

(2)通过向$s$中加入被$s$引用的页面和引用$s$的页面将$s$扩展成一个更大的集合$T$，作为基本集(Base set)。

(3)将网页$p$的Authority权重记为$a_p$，Hub权重记为$h_p$，为$T$所有网页赋初值均为1，通过以下迭代公式对$a_p$和$h_p$计算，直至结果收敛：

$$\text{I操作：}a_p=\sum_{\forall q:q\to p}h_q \tag{2}$$

$$\text{O操作：}h_p=\sum_{\forall q:p\to q}a_q \tag{3}$$

最后HITS算法输出一组具有较大Hub权重和具有较大Authority权重的页面。

### 2.3 HITS算法的改进

虽然PageRank和HITS算法都取得了很大的成功，但它单纯利用页面之间的链接关系，在应用过程中有时会出现主题漂移的现象。针对HITS算法的不足，提出了一种利用模糊合成关系的改进方法，如果输入一个查询词$C$，就可能会产生出与查询词$C$有一定模糊关系的一些词项$(C_1,C_2,L,C_n)$，词项与词项之间的关系为$f(C_i,C_j)\in(0,1),j\in1,2\Lambda,n$。并且从页面的角度考虑，页面$d$与词项$(C_1,C_2,L,C_n)$之间也存在一定的模糊关系$g(d_i,C_j)\in(0,1),j\in1,2\Lambda,n$。则应用模糊关系的合成可以得到页面$d$和查询词$C$之间的模糊隶属关系，体现了页面与查询词之间的主题相关性。把页面与查询词之间的模糊关系的度量应用在HITS算法中，使得HITS算法增加了与页面主题相关的度量因子，搜索出的网页避免了主题漂移的现象。

定义1：模糊合成。

$\boldsymbol{Q}\in P(U\times V),\boldsymbol{R}\in P(V\times W),\boldsymbol{S}\in(U\times W)$若$(u,w)\in\boldsymbol{S}\Leftrightarrow\exists v$，使$(u,v)\in\boldsymbol{Q},(v,w)\in\boldsymbol{R}$，则称关系$\boldsymbol{S}$是由关系$\boldsymbol{Q}$和$\boldsymbol{R}$合成的，记作$\boldsymbol{S}=\boldsymbol{Q}\circ\boldsymbol{R}$用特征函数表示为：

$$(\boldsymbol{Q}\circ\boldsymbol{R})(u,w)=\bigvee_{u\in V}(\boldsymbol{Q}(u,v)\wedge\boldsymbol{R}(v,w)) \tag{4}$$

式中，$\wedge$表示取最小，$\vee$表示取最大。

定义2：词项模糊矩阵$\boldsymbol{T}$：表示词与词之间的模糊隶属关系。

$$\boldsymbol{T}=(t_{ij})_{m\times n}\quad t_{ij}=\frac{t_i\,\mathrm{I}\,t_j}{t_i\,\mathrm{Y}\,t_j}\quad t_{ij}\in(0,1) \tag{5}$$

式中，$t_i\,\mathrm{I}\,t_j$表示同时有出现词$i$与词$j$的页面个数，$t_i\,\mathrm{Y}\,t_j$表示有词$i$或词$j$出现的页面个数。

定义3：传递闭包$K^*$：反映出元素间可见的和隐藏的关系，$K$满足自反性、对称性和传递性，按定义1进行合成有：

$$K^2=K\circ K=\bigvee_{l=1}^{n}(k_{il}\wedge k_{lj}),i\geqslant1,j\leqslant n \tag{6}$$

如果存在$K^P=K^{P+1}=K^{P+2}$，则$K^*=K^P$。

定义4：页面词频矩阵：表示页面$i$与词$j$之间的模糊隶属关系。

$$\boldsymbol{D}=(d_{ij})_{m\times n},d_{ij}=\frac{N_j}{\sum_{j=1}^{n}N_j},d_{ij}\in(0,1) \tag{7}$$

式中，$N_j$表示词$j$在页面$i$中出现的次数。

算法过程：

(1)将查询$q$提交给搜索引擎，搜索引擎返回很多页面，从中取前$n$个页面作为根集(Root set)，用$s$表示。

(2)通过向 $s$ 中加入被 $s$ 引用的页面和引用 $s$ 的页面将 $s$ 扩展成一个更大的集合 $T$,作为基本集(Base set)。

(3)找出与查询词有一定模糊关系的词项,词项可以从用户轮廓文件(user profiles)中得到[6],根据定义 2 得出词项模糊矩阵 $T=(t_{ij})_{m\times m}$,$m$ 为词项数目。

(4)根据定义 3 得出词项传递闭包 $T^*$,取出查询词所对应的列向量,得到查询词与词项的关系矩阵 $T'=(t'_{ij})_{m\times k}$,$m$ 为词项数目,$k$ 为查询词数目。

(5)按定义 4 得到词项与页面间的页面词频矩阵 $D=(d_{ij})_{n\times m}$,$n$ 为页面数目,$m$ 为词项数目。

(6)由(3)、(4)、(5)步得到模糊页面查询词合成关系矩阵 $D^*=DoT'=(d_{ij}^*)_{n\times k}$,$n$ 为页面数目,$k$ 为查询词数目。

(7)把权值 $\omega$ 加入原始 HITS 算法中,先将网页 $p$ 的 Authority 权重记为 $a_p$,hub 权重记为 $h_p$,并赋初值均为 1,再通过以下迭代公式对 $a_p$ 和 $h_p$ 计算,直至结果收敛。

$$\omega_p=\sum_{i=1}^{n} d_{pi}^* \tag{9}$$

$$\text{I 操作:}\ a_p^{(t+1)}=\omega_p\sum_{\forall q:q\to p} h_q^{(t)} \tag{10}$$

$$\text{O 操作:}\ h_p^{(t+1)}=\omega_p\sum_{\forall q:p\to q} a_q^{(t+1)} \tag{11}$$

每次迭代后对 $a_p$ 和 $h_p$ 进行规范化处理以保证不变性。

$$a_p=\frac{a_p}{\sqrt{\sum_{p\in T}(a_p^2)}},\quad h_p=\frac{h_p}{\sqrt{\sum_{p\in T}(h_p^2)}} \tag{12}$$

通过以上(3)、(4)、(5)、(6)、(7)步的改进,在单纯的页面结构链接关系的基础上,加入了页面与查询词项上的关联,使得算法避免了主题漂移的现象,最后的结果更能令人满意。

## 3 实例

为了更好地理解改进算法的计算过程,下面给出实例加以说明,搜索查询词 Java,网页链接结构如图 1 所示。

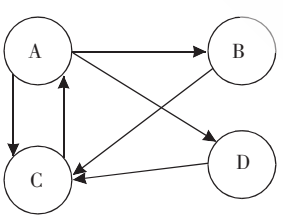


图 1 网页链接结构

采用未改进的 HITS 算法,直接运用链接,经多次迭代计算后,Authority 权重排序 C>B=D>A,Hub 权重排序为 A>B=D>C。

下面运用改进后的算法,得到相关模糊词项[6](Java, Book, Computer, Internet, Network, Corba, Software, Unix, Family, Newspaper)构成词项模糊矩阵 $T=(t_{ij})_{m\times m}$,$m=10$。

$$T=\begin{array}{c}\\ java\\ book\\ computer\\ int\ ernet\\ network\\ corba\\ software\\ unix\\ family\\ newspaper\end{array}\begin{bmatrix} 1.0 & 0.9 & 0.3 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0\\ 0.9 & 1.0 & 0.0 & 0.2 & 0.2 & 0.9 & 0.0 & 0.0 & 0.0 & 0.0\\ 0.3 & 0.0 & 1.0 & 0.5 & 0.5 & 0.8 & 0.3 & 0.9 & 0.0 & 0.0\\ 0.0 & 0.2 & 0.5 & 1.0 & 0.2 & 0.0 & 0.0 & 0.0 & 0.7 & 0.0\\ 0.0 & 0.2 & 0.5 & 0.2 & 1.0 & 0.4 & 0.3 & 0.8 & 0.0 & 0.0\\ 0.0 & 0.9 & 0.8 & 0.0 & 0.4 & 1.0 & 0.0 & 0.5 & 0.6 & 0.0\\ 0.0 & 0.0 & 0.3 & 0.0 & 0.3 & 0.0 & 1.0 & 0.1 & 0.2 & 0.1\\ 0.0 & 0.0 & 0.9 & 0.0 & 0.8 & 0.5 & 0.1 & 1.0 & 0.0 & 0.0\\ 0.0 & 0.0 & 0.0 & 0.7 & 0.0 & 0.6 & 0.2 & 0.0 & 1.0 & 0.0\\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.1 & 0.0 & 0.0 & 1.0 \end{bmatrix}$$

求出传递闭包 $T^*$:

$$T^*=\begin{array}{c}\\ java\\ book\\ computer\\ int\ ernet\\ network\\ corba\\ software\\ unix\\ family\\ newspaper\end{array}\begin{bmatrix} 1.0 & 0.9 & 0.8 & 0.6 & 0.8 & 0.9 & 0.3 & 0.8 & 0.6 & 0.1\\ 0.9 & 1.0 & 0.8 & 0.6 & 0.8 & 0.9 & 0.3 & 0.8 & 0.6 & 0.1\\ 0.8 & 0.8 & 1.0 & 0.6 & 0.8 & 0.8 & 0.3 & 0.9 & 0.6 & 0.1\\ 0.6 & 0.6 & 0.6 & 1.0 & 0.6 & 0.6 & 0.3 & 0.6 & 0.7 & 0.1\\ 0.8 & 0.8 & 0.8 & 0.6 & 1.0 & 0.8 & 0.3 & 0.8 & 0.6 & 0.1\\ 0.9 & 0.9 & 0.8 & 0.6 & 0.8 & 1.0 & 0.3 & 0.8 & 0.6 & 0.1\\ 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 1.0 & 0.3 & 0.3 & 0.1\\ 0.8 & 0.8 & 0.9 & 0.6 & 0.8 & 0.8 & 0.3 & 1.0 & 0.6 & 0.1\\ 0.6 & 0.6 & 0.6 & 0.7 & 0.6 & 0.6 & 0.3 & 0.6 & 1.0 & 0.1\\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 1.0 \end{bmatrix}$$

取出查询词 JAVA 与词项(JAVA, Book, Computer, Internet, Network, Corba, Software, Unix, Family, Newspaper)对应的列向量得到关系矩阵 $T'=(t'_{ij})_{m\times k}$,$k=1$,$m=10$。

$T'=[1.0\ \ 0.9\ \ 0.8\ \ 0.6\ \ 0.8\ \ 0.9\ \ 0.3\ \ 0.8\ \ 0.6\ \ 0.1]^{\mathrm{T}}$

网页 A、B、C、D 计算页面词频矩阵 $D=(d_{ij})_{n\times m}$,$n=4$。

$$D=\begin{array}{c}\\ A\\ B\\ C\\ D\end{array}\begin{bmatrix} 0.0 & 0.0 & 0.2 & 0.0 & 0.0 & 0.4 & 0.4 & 0.0 & 0.0 & 0.0\\ 0.3 & 0.0 & 0.5 & 0.2 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0\\ 0.0 & 0.0 & 0.2 & 0.5 & 0.0 & 0.1 & 0.1 & 0.0 & 0.0 & 0.0\\ 0.3 & 0.1 & 0.2 & 0.0 & 0.0 & 0.2 & 0.2 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

得到模糊页面查询词合成关系矩阵:

$$D^*=DoT'=(d_{ij}^*)_{n\times k}=\begin{bmatrix}0.4\\0.5\\0.5\\0.3\end{bmatrix}$$

权值 $\omega_p=\sum_{i=1}^{n} d_{pi}^*$,因为就 1 个查询词 Java,$n=1$,所以 $\omega_A=0.4$,$\omega_B=0.5$,$\omega_C=0.5$,$\omega_D=0.3$。

第 1 次 $a_A$、$a_B$、$a_c$、$a_D$、$h_A$、$h_B$、$h_C$、$h_D$ 均为 1

第 2 次 $a_A=\omega_A\sum_{\forall q:q\to A} h_q=h_c\times\omega_A=1\times0.4=0.4$

$$a_B=\omega_B\sum_{\forall q:q\to B} h_q=h_A\times\omega_B=1\times0.5=0.5$$

$$a_C=\omega_C\sum_{\forall q:q\to C} h_q=(h_A+h_B+h_D)\times\omega_C=3\times0.5=1.5$$

$$a_D=\omega_D\sum_{\forall q:q\to D} h_q=h_A\times\omega_D=1\times0.3=0.3$$

$$h_A=\omega_A\sum_{\forall q:A\to q}a_q=a_B\times\omega_A+a_C\times\omega_A+a_D\times\omega_A=$$
$$0.4\times0.4+1.5\times0.4+0.3\times0.4=0.88$$

$$h_B=\omega_B\sum_{\forall q:B\to q}a_q=a_C\times\omega_B=1.5\times0.5=0.75$$

$$h_C=\omega_C\sum_{\forall q:C\to q}a_q=a_A\times\omega_C=0.4\times0.5=0.2$$

$$h_D=\omega_D\sum_{\forall q:D\to q}a_q=a_C\times\omega_D=1.5\times0.3=0.45$$

经过多次迭代后，每个页面的数值逼近一个定值，Authority 权重排序 C>B>D>A，Hub 权重排序为 A>B>D>C。

通过多次实验对比，可以得出：(1)原算法求出 Authority 权重、Hub 权重较高的网页，由于页面与查询词相关权重很小，改进后求出的 Authority 权重和 Hub 权重可能变得较低，说明网页虽然重要，但不是用户所要查询的内容网页；(2)原算法求出 Authority 权重、Hub 权重较低的网页，由于页面与查询词相关权重很大，改进后求出的 Authority 权重和 Hub 权重可能变得较高，说明主题相关，网页是用户所需求的。这样就克服了主题漂移的现象。

本文对原有的 HITS 算法进行改进，把查询词与 Web 页内容和链接相结合，改善了原有算法中主题漂移的现象。网络中的链接可以展示出它们链接目标的许多东西。网络不是单一的链接，可能的研究方向有很多，在 WWW 上网页内部的超链用 HTML、XML 如何表示成树形结构，文档是如何表示成 URL 中的目录路径结构，站点之间怎样通过超链同其他相关联的站点或页面相链接。通过查看一个单独站点的网页的链接情况及相互间链接的情况来学习其内部结构等等。如何很好地利用 Web 结构对互联网上信息进行知识获取将是未来研究的重点。

参考文献

[1] 吉根林，孙志挥.Web 挖掘技术研究 [J]. 计算机工程，2002，28(10).

[2] HAN J W，KAMBER M.数据挖掘概念与技术[M].范明，孟小峰，译.北京：机械工业出版社，2001.

[3] 杨炳辉，李岩，陈新中.Web 结构挖掘[J].计算机工程，2003，29(20).

[4] 戚美华，黄德才，郑月峰.具有时间反馈的 PageRank 改进算法[J].浙江工业大学学报，2003，33(3)：273-275.

[5] KLEINBERG J. Authoritative sources in a hyperlinked environment. Journal of the ACM，2006，46：604-32.

[6] KYUNG J K，SUNG B C. Personalized mining of web documents using link structures and fuzzy concept network [J]. Applied Soft Computing 2007：398-410.

[7] BANG J J，GUTIN G D. Theory，algorithms and applications. Heidelberg；Springer；2003.

(收稿日期：2009-03-06)