一种新的基于 Web 日志的挖掘用户浏览偏爱路径的方法*)

任永功[1] 付 玉[1] 张 亮[1] 吕君义[2]
(辽宁师范大学计算机与信息技术学院 大连 116029)[1] (辽河油田锦州工程技术处 凌海 121209)[2]

摘 要 提出了一种新的基于 Web 日志的挖掘用户浏览偏爱路径的方法。该方法首先在单元数组存储结构(存储矩阵)基础上建立以浏览兴趣度为基本元素的会话矩阵和路径矩阵。然后,在会话矩阵上采用两个页面向量夹角余弦作为相似用户的页面距离公式进行页面聚类,求得相似用户的相关页面集。最后,利用路径选择偏爱度在相似用户的路径矩阵上挖掘出相似用户的浏览偏爱路径。实验证明此方法是合理有效的,能够得到更准确的偏爱路径。

关键词 浏览兴趣度,路径选择偏爱度,Web 日志,矩阵,页面聚类

New Approach of Mining User's Preferred Browsing Paths

REN Yong-gong[1] FU Yu[1] ZHANG Liang[1] LU Jun-yi[2]
(School of Computer and Information Technology, Liaoning Normal University, Dalian 116029, China)[1]
(Jinzhou Engineering Technology Department of Liaohe Oilfield, Linhai 121209, China)[2]

Abstract This article proposed a new approach of mining user's preferred browsing paths based on Web logs. This approach first establishes session matrix and trace matrix by taking browsing interest as the fundamental element based on cell storage structure (storage matrix). Then carries on page clustering in the session matrix through using angle cosine in vector space between two pages, which is called the similar user's page distance formula. And we can get the similar user's relative pages set. Finally, mines the similar user's browsing preferred paths by using path choice-preference in similar user's trace matrix. Experiments prove that the approach is reasonable and effective and can discover more accurate preferred browsing paths.

Keywords Browsing interest, Path choice-preference, Web log, Matrix, Page cluster

1 引言

随着 Internet 和 Web 技术和电子商务、网络教育等基于 Web 应用的迅猛,Web 数据挖掘发展已成为人们关注的重要课题。由于 Web 是一个无集中控制、无统一结构、无完整性约束、无事务管理、无标准查询语言和数据模型、可无限扩充的一个松散的分布式信息系统,从理论上讲,对其挖掘是困难的,获取的知识是不可靠的。然而,Web 服务器的 log 日志却有完美的结构,每当用户访问 Web 站点时,所访问的页面、时间、用户 IP 等信息,在 log 日志中都有相应的记录(如表 1 所示)。分析 Web 日志,发现用户浏览路径的共同行为,从而可以"投其所好",为用户提供个性化服务,并且对站点的智能化设计具有重大意义。

目前基于 Web 日志的分析浏览路径的方法很多,取得了一定的成果。文献[4]中提出了偏爱度的概念,着重强调了访问次数在挖掘用户浏览偏爱度算法中的作用,但忽略了访问时间和访问页面速度对用户浏览偏爱度的影响。文献[5]中用树存储结构存储 Web 日志中的网页访问情况,不能准确描述有回溯的浏览路径。文献[7]仅用页面访问次数作为算法分析度量值,有时不能准确得到相似用户的频繁路径。还有部分文献没有明显的独立存储结构,在算法分析和计算时需频繁访问 Web 日志数据库文件,这势必会加重 I/O 负担,降低算法效率。分析 Web 日志,发现用户浏览路径的共同行为,从而可以"投其所好",为用户提供个性化服务,并且对站点的智能化设计具有重大意义。

表 1 Web 日志格式

域(field)	描述(description)
日期(date)	请求页面的时间、日期和时区(date, time and timezone of request)
客房端 IP(client IP)	远程主机的 IP 域 DNS 入口(remote host IP and/or DNS entry)
用户名(username)	远程登录的用户名(remote logname of the user)
字节(bytes)	发送和接收的字节(bytes transferred sent and received)
服务器(server)	服务器名称、IP 地址和端口(server name, IP address and port)
请求(request)	URL 查询和枝节(URL query and stem)
状态(status)	返回给 HTTP 状态标识(http status code returned to the client)
服务名(service name)	用户请求的服务名称(requested service name)
耗用时间(time taken)	完成浏览的时间(time taken for transaction to complete)
协议版本(protocol version)	传输用的协议版本(version of used transfer protocol)
用户代理(user agent)	服务提供者(service provider)
Cookie	标识号(cookie ID)
参照页(referrer)	本页的上一页(previous page)

*)国家自然科学基金项目(60603047);辽宁省教育厅高等学校科研基金(2008341);辽宁省自然科学基金;大连市优秀青年科技人才基金(2008J23JH026)。任永功 教授,博士,CCF 高级会员,研究方向为数据挖掘、图像处理技术等;付 玉 硕士研究生,研究方向为 Web 数据挖掘;张 亮 硕士研究生,研究方向为 Web 数据挖掘;吕君义 工程师,硕士,研究方向为石油地质开发、油田井下作业等。

为了解决上述问题，本文首先提出存储矩阵的概念，该矩阵采用单元数组作为存储结构，仅存储算法必需的日志项，去掉了大量冗余的 Web 日志信息，引用浏览兴趣度作为该存储矩阵的一列元素，直观、真实地记录了用户对网页的感兴趣情况。存储矩阵的建立使得整个算法只需访问一次数据库，避免了在建立会话矩阵和路径矩阵时再度频繁访问数据库，提高了算法效率，节省了算法的时间。存储矩阵又为建立会话矩阵和路径矩阵提供数据依据。其次，本文在文献[7]的研究工作基础上，不仅考虑用户浏览次数，同时考虑页面浏览时间和浏览速度等因素，提出一种新的相似用户的页面聚类算法，以用户兴趣度为基本处理元素，采用页面向量夹角余弦页面距离公式作为相似用户的页面距离的公式。经实验证明，该页面距离公式得到的相关页面集更精确，相似度更高。最后，本文给出路径选择偏爱度的概念，此偏爱度充分考虑了用户浏览网页的回溯问题，由此得到的浏览偏爱路径更具真实性，更能说明相似用户在网页浏览上的真正喜好。

2 相关概念

定义 1(浏览兴趣度，记作 P)

设定 P_j 表示用户在页面 j 上的浏览兴趣度，$Count_{ij}$ 表示用户从页面 i 进入页面 j 的浏览次数，$Time_{ij}$ 表示用户从页面 i 进入页面 j 的浏览时间(单位：秒)，Sbs_{ij} 表示从页面 i 进入页面 j 所接收到的字节数(单位：MB)，则用户浏览兴趣度公式为：

$$P_j=\frac{Count_{ij} * Time_{ij}}{Sbs_{ij}} \tag{1}$$

用户浏览网站时对某一页面感兴趣程度通常由页面的浏览时间、浏览次数以及网速等几方面的因素决定。定义中的浏览时间在 Web 日志中指页面的耗用时间，页面的浏览时间越长说明用户对该页面越感兴趣，而浏览时间又与浏览速度有关，页面的浏览速度在 Web 日志中则对应页面的接收字节数(见表 1)。速度越快，接收字节数越多。这个浏览兴趣度的定义更能全面地反应出用户对页面的关注程度。

用户浏览兴趣度是本文算法中最基本的细胞元素，算法中无论是建立矩阵模型还是挖掘算法都以用户浏览兴趣度为基准展开分析和计算。

定义 2(相似用户的页面距离，记作 Pd)

设从矩阵抽取两行 i,j 行，行向量分别记为 X,Y,X_i,Y_j 分别是两行中对应的元素值，则相似用户的页面距离公式为：

$$Pd=\frac{\sum\limits_{i=1,j=i}^{n}(X_i * Y_j)}{\sqrt{\sum\limits_{i=1}^{n}(X_i)^2 * \sum\limits_{j=1}^{n}(Y_j)^2}} \tag{2}$$

其中，行向量 X,Y 元素总个数都为 n。

文献[7]中将 Humming 公式作为相似用户的页面距离公式虽便于理解，但有时存在一定问题：1) Humming 公式当要计算的距离矩阵维数高且是稀疏矩阵时，效率不高；2)没有考虑处理元素的数值大小在产生页面距离的差异。因此，本文采用基于向量间的夹角余弦来定义页面距离，解决了 Humming 距离的不足，全面考虑了 Web 日志中浏览总次数、浏览时间、浏览速度等对页面距离产生的影响，避免产生页面距离的差异，提高了算法的准确性。

相似用户的页面距离公式是本文一个子算法——页面聚类算法的重要分析依据，通过它计算页面间的页面距离，聚类它们的相似程度，从而可以得出更合理、更准确的相似用户的相关页面集。

定义 3(路径选择偏爱度，记作 Pc)

设 U 是网站中所有 URL 的集合，W 是所有浏览子路径的集合，如果存在 $w\subset W$，对于 $\forall x\in w$(x 是 $\forall u\in U$ 组成的浏览页面序列，称其中每 j 个浏览页面为第 j 位)，它们的前 m 位都相同，而 $m+1$ 位有 n 种不同的浏览页面，则称在 m 位上有 n 种不同的选择，其中第 $k(k=1,2,\cdots,n)$ 种选择的路径选择偏爱度(Pc)可定义为

$$Pc_k=P_j/(\sum_{j=1}^{n}P_j/n) \tag{3}$$

其中，P_j 表示第 j 种选择的浏览兴趣度。

文献[4]中以页面的浏览次数作为基本元素的选择偏爱度的定义有些片面，本文充分考虑各种影响页面浏览的因素，把浏览兴趣度作为公式中的原始元素，这样可以更准确地描述用户的浏览兴趣及由浏览兴趣所产生的浏览路径的选择倾向。路径选择偏爱度的提出，还很好地解决了带有回溯的浏览路径问题。

路径选择偏爱度是本文中一个子算法——2-项浏览偏爱子路径集生成算法的主要判断依据。在路径选择偏爱度满足一定条件时才会产生 2-项浏览偏爱子路径，进而合并子路径产生最后的偏爱路径集。

3 改进的用户浏览偏爱路径挖掘方法

3.1 算法基本思想

Step1 建立存储矩阵(记作 *MEM*)。首先调用一次存放在 access 中的预处理后的 Web 日志数据库文件，将其数据经过计算和整理后，存入一个 $n*4$ 的单元数组(即存储矩阵 *MEM*)中。存储矩阵可视性好，直观、明了地反映出算法所用的信息。*MEM* 是整个算法的数据基础。存储矩阵形式如下：

$$MEM_{n*4}=\begin{vmatrix} URL_1 & User_1 & P_1 & preURL_1 \\ URL_2 & User_2 & P_2 & preURL_2 \\ \vdots & \vdots & \vdots & \vdots \\ URL_n & User_n & P_n & preURL_n \end{vmatrix}$$

其中，矩阵维数 n 并不是数据库文件中记录总个数，而是整理合并后的记录总个数；URL 表示当前访问页；User 表示访问用户；P 表示页面浏览兴趣度；preURL 表示引用页。

Step2 建立会话矩阵(记作 *SM*)和路径矩阵(记作 *TRM*)。在 *MEM* 基础上，以 URL 为行，User 为列，页面浏览兴趣度 P 为基本元素值，建立 *SM*；以 preURL 为行，URL 为列，页面浏览兴趣度 P 为基本元素值，建立 *TRM*。这样建立起来的矩阵较文献[7]中仅以浏览次数为基准建立的矩阵更全面，准确性和有效性高。*SM* 和 *TRM* 的形式如下(矩阵中的其它说明见文献[7])：

$$\text{会话矩阵 } SM_{m*n}=\begin{vmatrix} s_{11} & \cdots & s_{1n} \\ \vdots & \ddots & \vdots \\ s_{m1} & \cdots & s_{mn} \end{vmatrix}$$

$$\text{路径矩阵 } TRM_{m*m}=\begin{vmatrix} t_{00} & t_{01} & \cdots & t_{0m} \\ t_{10} & t_{11} & \cdots & t_{1m} \\ \vdots & \vdots & \cdots & \vdots \\ t_{m0} & t_{m1} & \cdots & t_{mm} \end{vmatrix}$$

Step3 在会话矩阵 *SM* 基础上，通过页面聚类算法(Cluspages 算法)计算出相似用户的相关页面集 *Simpages*。首先根据 1-项页面集偏爱阈值，剪枝掉小于此阈值的页面，

减少冗余，得到 1-项浏览偏爱页面集 *Simpage*。在这里，本文给出的由平均用户兴趣度值作为 1-项浏览偏爱页面集的判定条件，比文献[7]中的总浏览次数作为判断条件更精确。平均兴趣度值能反映普遍水平，具有一般性，而总浏览次数容易产生极端情况。其次在 *Simpage* 下，由 *Pd* 统计出相似用户的相关页面集 *Simpages*。

Step4　结合 *Simpages* 和 *TRM*，通过页面转换算法（Pgtomtx 算法）生成 *Simpages* 各自对应的路径矩阵集 *Frematrx*，同时生成标记 *Simpages* 中各页面于 *TRM* 的对应位置的数据集合 *Tagsets*。

Step5　由 *Frematrx* 和 *Tagsets* 通过 2-项浏览偏爱子路径集生成算法 Conpath2 算法，生成 2-项浏览偏爱子路径集 *Preferpath2*。

Step6　通过合并算法 Conpaths 算法合并 *Preferpath2*，直至生成 *k*-项浏览偏爱子路径集，从而得到所有的浏览偏爱路径集 *Preferpathsets*。

所得的 *Preferpathsets* 为所求。

3.2　主要算法

算法 1　页面聚类算法

```
算法名称:Cluspages()
//相似用户的相关页面的生成算法。
输入:用户会话矩阵 SM,平均兴趣度阈值 Ps,相似用户页面距离阈值 Ds
输出:相似用户的相关页面集合 simpages
//由用户访问页面的会话矩阵,得到相似用户群体的相关联页面。
算法描述:
//根据每个网页页面的平均的用户兴趣值,计算出频繁页面集。即把兴趣低的网页先剪枝掉。
for l=1:len //len 为 SM 行数
  { if 每行平均用户兴趣度值>Ps
    tag←行号;
    //tag 放置频繁页面集对应矩阵位置
      matrix ←SM(l,:);
  }
  //在 matrix 中计算页面距离 Pd,并由页面距离,进一步筛选得到相似用户的相关页面集合。
  for i=1:m //m 是 matrix 的总行数
    { userset1←第 i 行向量 X;
      //userset1 是参与计算的第一个页面
      取出 i 在 tag 中的位置;
      for j=i+1:m
      { serset2←第 j 个行向量 Y;
        // userset2 是用于计算第二个页面
        取出 j 在 tag 中的位置;
        求出 Pd;
        if Pd<= Ds
          simpage ←〈i,j〉对应的页面;
      }
    }
    // 整理所得的页面集合,要求其集合中页面路径长度均大于等于 2。长度为 1,没有意义。
for l1=1:simpage 长度
  {path ←simpage{l1};
    if path 的长度>=2
      tmpage← page;
      simpages←tmppage 中去掉重复项的结果;
//simpages 为所求。
}
```

算法 2　2-项浏览偏爱子路径集生成算法

```
算法名称:Conpath2()
//2-项浏览偏爱子路径集生成算法
输入:相似用户的相关页面矩阵集 frematrixs,相关页面在路径矩阵中的存储位置 tagsets,路径选择偏爱度阈值 Pi
输出:2-项浏览偏爱子路径集合 prepath2sets
算法描述:
prepath2sets 置空
//分别统计每个相关页面集的 2-项浏览偏爱子路径集。
for k=1:frematrixs 长度
  { matrix←frematrixs{k};
    tag←tagsets{k};
    for i=1:m //m 是 matrix 的行数
      { for j=1:n //n 是 matrix 的列数
      //判断每个非零元素对应的路径是否是偏爱路径
        { 计算每个非零元素的路径选择偏爱度 Pc;
          //将满足条件的路径〈i,j〉加入到 prepath2 中。
          if Pc>=Pi
          //根据 tag 的值,将〈i,j〉对应的浏览路径放入 prepath2 中。
          prepath2←〈i,j〉;
        }
      }
    //prepath2sets 为所求的 2-项浏览偏爱子路径集合。
prepath2sets← prepath2;
}
```

4　示例分析

设定英文大写字母 A,B,C……表示浏览的网页（URL，preURL），U1,U2,U3……表示浏览网页的用户（User）。

1）设某一个网站的 Web 日志经过预处理后形成如下的存储矩阵 *MEM*（部分）：

'A'	'U1'	[8.3]	'/'
'C'	'U1'	[3.2]	'A'
'E'	'U1'	[4]	'C'
'F'	'U1'	[28]	'E'
'E'	'U1'	14.2]	'A'
'C'	'U2'	[37.1]	'B'
'D'	'U2'	[25.2]	'C'

2）建立会话矩阵 *SM* 和路径矩阵 *TRM*

a）由 *MEM* 建立的会话矩阵 *SM*：

8.3	0	6.5	16.7	0.4	9.9	3.6	33.1	75.8
24.5	60.3	37.7	44.3	10.7	55.0	23.5	31.4	0
3.2	37.1	1.3	0	12.5	0	14.1	28.8	0
0	25.2	11.7	21.2	16.0	9.2	24.5	0	0
18.2	12.3	0	37.3	0	0	0	0	59.0
40.2	0	9.8	17.0	3.3	38.1	0	45.2	42.3

b）由 *MEM* 建立的路径矩阵 *TRM*：

0	154.3	104.3	0	2.6	0	41.0
0	0	54.9	24.1	9.2	82.1	0
18.0	0	0	72.9	21.2	1.3	31.9
0	0	17.5	0	74.8	4.0	22.6
14.0	0	0	0	0	35.4	16.7
14.6	0	0	0	0	0	83.7
33.9	0	110.7	0	0	0	0

3)通过 Cluspages 算法统计出相似用户的相关页面集 *Frepages*。

计算各页面的平均兴趣度，A:17.1，B:31.9，C=10.8，D:12.0，E:14.1，F:21.8。由于页面的平均兴趣度阈值 *Ps*=10，ABCDEF 都满足条件(大于等于 10)。首先得到 1-项浏览偏爱页面集为{A,B,C,D,E,F}，再由余弦页面距离公式计算频繁页面集中各页面距离如下(相似用户页面距离阈值 *Ds* 为 0.6)

Pd(AB)=0.3，*Pd*(AC)=0.2，*Pd*(AD)=0.2，*Pd*(AE)=0.8，*Pd*(AF)=0.8

Pd(BC)=0.7，*Pd*(BD)=0.8，*Pd*(BE)=0.3，*Pd*(BF)=0.6

……

Pd(EF)=0.6

由算法 Cluspages，小于等于 0.6 的页面两两相关。于是，得到相关页面集为{{ABCD},{BEF},{CDEF},{DEF},{EF}}，去掉重复项，得到三组相似用户的相关页面集 *simpages*:{{ABCD},{BEF},{CDEF}}。

4)由 Pgtomtx 算法得到相关页面的路径矩阵集 *Frematrx* 及相关页面在原始路径矩阵中的对应位置集 *Tagsets*。

由相关页面集合和路径矩阵得到新的相似用户的相关页面的路径矩阵集 *Frematrx* 如下：

$$\begin{vmatrix} 0 & 54.9 & 24.1 & 9.2 \\ 0 & 0 & 72.9 & 21.2 \\ 0 & 17.5 & 0 & 74.8 \\ 0 & 0 & 0 & 0 \end{vmatrix} \leftrightarrow (\{ABCD\})$$

$$\begin{vmatrix} 0 & 1.3 & 31.9 \\ 0 & 0 & 83.7 \\ 110.7 & 0 & 0 \end{vmatrix} \leftrightarrow (\{BEF\})$$

$$\begin{vmatrix} 0 & 74.8 & 4.0 & 22.6 \\ 0 & 0 & 35.4 & 16.7 \\ 0 & 0 & 0 & 83.7 \\ 0 & 0 & 0 & 0 \end{vmatrix} \leftrightarrow (\{CDEF\})$$

Tagsets 为{{2,3,4,5},{3,6,7},{4,5,6,7}}

5)统计各页面集中的 2-项浏览偏爱子路径集 *Preferpath2*。

给定用户路径选择偏爱阈值 *Pi*=1。

对于页面集{ABCD}，第一行中(AB)

Pc=54.9/(88.2/3)=1.9>1，由 *Simpages* 和 *Tagsets*，将路径〈AB〉加入 2-项浏览偏爱子路径集 *Prepath2* 中；(AC)：*Pc*=24.1/(88.2/3))0.8<1，将〈AC〉剪枝掉；(AD)：*Pc*=9.2/(88.2/3))=0.3<1，将〈AD〉剪枝掉。第二行中(BC)：*Pc*=72.9/(94.1/2))=1.5>1，将〈BC〉加入 2-项浏览偏爱子路径集 *Prepath2* 中。……依此类推，得到 2-项浏览偏爱子路径集 *Prepath2*:{AB,BC,CD}。

对于页面集{BEF}，第一行中(BE)：*Pc*=1.3/(33.2/2)=0.1<1，将〈BE〉剪枝掉；(BF)：*Pc*=31.9/(33.1/2)=1.9>1，将〈BF〉加入 2-项浏览偏爱子路径集 *Prepath2* 中。依次计算第二行和第三行，可以得到 2-项浏览偏爱子路径集 *Prepath2*:{BF,EF,FB}。

……

于是得到所有的 2-项浏览偏爱子路径集 *Preferpath2*:{{AB,BC,CD},{BF,EF,FB},{CD,DE,EF}}。

6)合并浏览偏爱子路径集，得到所有浏览偏爱路径集 *Preferpathsets*。

对于页面集{ABCD}的 2-项浏览偏爱子路径集 *Prepath2*:{AB,BC}，〈AB〉与〈BC〉合并生成〈ABC〉，〈AB〉与〈CD〉不能合并，〈BC〉与〈CD〉合并生成〈BCD〉，从而得到 3-项浏览子路径集 *Prepath3* 为{ABC,BCD}。对于 3-项浏览偏爱子路径集中的〈ABC〉与〈BCD〉，合并生成〈ABCD〉，从而得到 4-项浏览偏爱子路径集 *Prepath4* 为{ABCD}，合并结束。

同理，对于页面集{BEF}的 2-项浏览偏爱子路径集 *Prepath2*:{BF,EF,FB}，合并后的 3-项浏览偏爱子路径集 *Prepath3* 为{BFB,EFB}，不能生成 4-项浏览偏爱子路径集，合并结束。

……

最后得到所有的浏览偏爱路径集为 *Preferpathsets*:{{ABCD},{BFB,EFB},{CDEF}}。

5 算法分析与比较

本文用 CPU：Intel Pentium processor 1.7GHz，内存：1.21GB的笔记本在 Windows XP 平台上用 Matlab 语言实现了本文中的算法以及文献[7]中的页面聚类算法和 2-项浏览偏爱子路径挖掘算法。由实验结果，分析比较如下：

1)文献[7]中并不能直观、清晰地查看算法所用的日志信息，并且在生成用户矩阵和路径矩阵时需要频繁调用数据库文件。而本文采用单元数组作为数据存储结构，减少了数据冗余，为算法提供了清晰的数据平台，使整个算法易懂、直观。

2)在页面聚类相关页面集的算法和路径挖掘算法中，文献[7]与文献[4]只考虑了访问次数，没有把时间和浏览速度考虑在用户兴趣度内，结果不准确。而本文不但综合考虑浏览次数、浏览时间以及浏览速度等因素，采用浏览兴趣度作为算法分析依据，并将用于比较的阈值细化到有效小数位数为 1 的数值，使准确性更高。

3)与文献[6]相比，同样以浏览兴趣度为出发点，本文不仅能得到频繁路径，而且更合理地表征出不同相似客房群体的偏爱路径，使得结果更具实用价值。

显而易见，数据库的大小对本文的结论不产生过多的影响，故表 2 和表 3 以 100 条记录为例，实现了本文的主要算法——页面聚类算法和 2-项浏览偏爱子路径集挖掘算法同文献[7]的比较。

表 2　页面聚类算法比较表

	页面聚类算法	
	I	II
Ds=0.2	{/}	{{ACD},{CE},{DF}}
Ds=0.4	{/}	{{ABCD},{BE},{CEF},{DEF}}
Ds=0.6	{/}	{{ABCD},{BEF},{CDEF}}
Ds=0.7	{/}	{{ABCD},{BCEF}{CDEF}}
Ds=1	{{ABF},{EF}}	{ABCDEF}
Ds=3	{{ABCDF},{EF}}	{ABCDEF}

表 2 列举出当页面聚类算法初始条件相同 (1-项偏爱页面集都为{ABCDEF})时不同的页面距离阈值下得到的不同的相关页面集。I 为文献[7]的实验结果，II 为本文的实验结果。通过表 2 可以看出，算法 I 得到的结果比较粗略，II 算法因为求得的是向量夹角余弦值，故 *Ds* 在 0～1 之间时得到的结果比对应的算法 I 中 *Ds* 在 1～3 时更精准。

表 3　浏览路挖掘算法比较

	2-项浏览偏爱子路径挖掘算法	
	I	II
Pi=1	{AB,AC,AE,BC,CD,DF,EF,FB}	{AB,AE,BC,BF,CD,DE,EF,FB}
Pi=1.2	{AB,AE,BC,CD}	{AB,AE,BC,CD,DE}
Pi=2	{BC,CD}	{AE,BC,CD}
Pi=3	{/}	{/}

表 3 列举出浏览偏爱子路径挖掘算法在当初始条件相同(相关页面集都为{ABCDEF})时不同的浏览选择偏爱阈值下得到的不同浏览偏爱路径集。I 为文献[7]的实验结果,II 为本文的实验结果。通过表 3 可以看出,以本文中改进的浏览兴趣度为基本元素的算法 II 得到的偏爱路径比算法 I 准确。

由实验结果分析,本文算法在有效性和准确性上有一定的优势,可扩展性良好。

结束语　本文提出了一种改进的基于 Web 日志的用户浏览偏爱路径的挖掘方法。本文主要是在以单元数组的存储结构为基础建立的两个矩阵模型上,挖掘了不同的相似用户群体的相关页面集的浏览偏爱路径。方法只需访问一次数据库文件,减少了 I/O 负担。本文相对于其他算法,在有效性和准确性方面具有一定优越性,能准确、充分地表现不同用户群体在浏览路径上的偏爱倾向,可扩展性好。笔者在下一阶段将在浏览兴趣度及算法的处理对象上做进一步的研究。

参 考 文 献

[1] Hua Jiawei, Kamber M. Data Mining Concepts and Techniques [M]. China Machine Press, 2001

[2] Anand S S, Patrick A R, Hughes J G. A Data Ming Methodology. Cross Sales Knowledge Based System Journal, 1998, 10(7): 449-461

[3] Pierrako S D, Paliouras G. Web Usage Mining as a Tool for Personalization: A survey[J]. Kluwer Academic Publishers, 2003: 311-372

[4] XING Dong-shan, SHEN Jun-yi, SONG Qin-bao. Discovering Preferred Browsing Paths from Web Logs[J]. Chinese Journal of Computers, 2003, 11(26): 1518-1523

[5] NING Xiao-hong, YU Sen-sen. Study on s-Tree Algorithm for Personalized Recommendation [J]. Computer Science, 2007, 34(4): 217-221

[6] Zhang Hai - yu , Liu Xiao - xia . A New Way to Discover User Browsing Mode[J]. Computer Applications and Software, 2007, 24(2): 143-150

[7] Du Jia-qiang, Han Qi-rui, Wang Ke, et al. A Fast Algorithm for Mining User Frequent Paths from Web Logs[J]. Computer Engineering and Applications, 2005, 22: 164-167

[8] Tia Chang-peng. Base on the Analysising and Researching Web-sever Log of Web Qos[J]. Computer Science, 2007, 34(6): 78-80

[9] Mao Guo-jun, Duan Li-juan. Data Mining Principles and Algorithms[M]. Tsinghua University Press, 2005

(上接第 157 页)

算机只得采取"以数量换质量"的策略,即用多个明确的、单义的概念(及其关系)表达一个人脑中的概念(及其关系)。由此导致了概念系统规模的急剧增大,这不仅涉及工作量,更关键的是如此多的概念(及其关系)由谁来选取与甄别?只有领域专家,这在实际操作上有诸多困难。

(2)持续性维护。概念系统一定是开放的。人脑中的概念是通过后天习得和人际交互的传授(最典型的是教育),概念的更新与概念系统的丰富则是由其主动积极的思维活动来实现。相比之下,当今的计算机只有"人操作的外部输入"这一种途径,因此必须不断地由人对概念(及其关系)进行添加、修改与更新。

(3)分布式及其集成应用。人类知识的巨量与进化决定了概念系统只能是分布式的。多个概念系统存在的优势是兼顾不同领域、并行开发与维护、使用时的按需取舍等。然而,分布式带来的直接问题是如何相互调用,其中异质性(heterogeneity)是最大阻碍。本体集成(ontology integrating)应运而生,包括本体映射(ontology mapping)、本体合并(ontology merging)、本体调整(ontology aligning)和本体连接(ontology articulating),成为目前 Ontology 研究中的主要分支。

结束语　语义 Web 所说的"语义"就是组成计算机中的概念系统的那些概念。如此界定只是一种现实可行的选择,所做的两项简化为"搁置语境"和"以概念近似观念"。

语义 Web 由 Web 扩展而来,两者同为信息集合。前者中的信息采用概念标记符标记,供机器交互理解,后者中的信息仅用格式标记符标记,由人直接阅读。扩展的目的就是要从"由人直接阅读信息内容"进步到"先由机器阅读并进行推理,再将结果按需提供给人使用"。扩展的措施为(1)信息采用概念标记;(2)计算机内置概念系统。

概念标记就是用概念标记符(即概念系统中的概念)对将要交由计算机处理的信息进行标记。也就是用 XML 对信息做"语义化处理"。目前的难点主要有:信息内容的切分、概念标记符的选用、标记过程的自动化等。

概念系统就是 Ontology,是由概念、概念的属性、概念之间相互关系组成的层次网络。它的主要功用有二:读出信息的直接语义、进行概念推理。基于概念系统通达性的概念推理才是语义 Web 的最突出的特征。应用概念推理计算机能够读出给定信息的多种语义。构建概念系统目前需要解决的问题主要有:规模庞大、持续性维护、分布式及其集成应用等。

参 考 文 献

[1] Berners-Lee T, Hendler J, Lassila O. The semantic Web[J]. Scientific American, 2001, 284(5): 34-43

[2] Jacobs I, Walsh N. Architecture of the World Wide Web[R]. W3C: Recommendation. http://www.w3.org/ TR/2004/REC-Webarch-20041215/

[3] Uschold M. Where are the semantics in the semantic Web[EB/OL]. http://www.starlab.vub.ac.be/WhereAre Semantics-AI-Mag-FinalSubmittedVersion2.pdf

[4] Antoniou G, Harmelen F. The semantic Web primer[M]. London: The MIT Press, 2004: 17-18

[5] 张志毅,张庆云. 词汇语义学(修订本)[M]. 北京:商务印书馆, 2005: 1-3

[6] 李幼蒸. 理论符号学导论[M]. 北京:社会科学文献出版社, 1999: 128-133, 292-296

[7] 陈嘉映. 语言哲学[M]. 北京:北京大学出版社, 2003: 44-57

[8] Frege G. 弗雷格哲学论著选辑[M]. 王路,译. 北京:商务印书馆, 2006: 95-119

[9] 王寅. 语义理论与语言教学[M]. 上海:上海外语教育出版社, 2001: 34-38

[10] 罗旋. 基于复句领域本体的语义标注方法研究[D]. 硕士论文. 武汉:华中师范大学, 2006

[11] Gruber T R. A translation approach to portable ontology specifications, KSL92-71[R]. San Francisco: Knowledge Systems Laboratory of Stanford University, 1993