DOI:10.13374/j.issn1001-053x.2006.09.020
第28卷 第9期 北京科技大学学报 Vol.28 No.9
2006年 9月 Journal of University of Science and Technology Beijing Sep. 2006

一种分布式 Web 使用模式挖掘模型及算法

张克君[1,2)] 杨炳儒[2)] 赵 耿[1)] 曲文龙[2)] 李 欣[2)]

1) 北京电子科技学院计算机科学与技术系，北京 100070 2) 北京科技大学信息工程学院，北京 100083

摘 要 给出了一种分布式 Web 日志挖掘模型 DWLMS. 根据对挖掘过程及算法进行分析，提出了一种基于 DWLMS 的局部频繁路径的更新算法 LFP 和全局频繁路径的更新算法 GFP，较好地解决了 Web 访问信息的异地存储、实时增长、分布式算法通讯量等因素给模式分析过程带来的困难. 在实验室对该方法进行了简单实现和实际日志数据的测试，结果表明了算法的有效性.

关键词 分布式数据挖掘；Web 使用模式挖掘；Web 日志挖掘；频繁路径

分类号 TP311.13

Web 使用模式挖掘，又称 Web 日志挖掘，是从 Web 的访问记录中抽取感兴趣的模式的过程[1-2]，其常用的技术有 WEB 挖掘特有的路径分析技术和数据挖掘领域的传统技术. 现有路径分析技术一般基于 Chen 等在文献[3]中的工作，Alexandros Nanopoulos 和 Yannis Manolopoulos 于 2001 年又提出了 MF，FS，SS 算法[4]. 这些算法都是用于集中式数据库系统中的挖掘算法.

目前，人们对分布式数据挖掘(DDM)的研究已取得了一些进展，并提出了一些分布式数据挖掘算法和分布式数据挖掘体系，如 PADMA[5]，CDM[6]，HDM[7]和 DDMINER[8]等. 所谓分布式数据挖掘就是使用分布式算法，从逻辑上或物理上分布的数据源中发现知识的过程.

本文对多镜像站点环境下的分布式日志挖掘系统中频繁访问路径模式挖掘进行了研究，主要讨论了站点访问日志量增加时频繁访问路径如何高效更新的问题，提出了一种基于多镜像站点的分布式 Web 使用挖掘系统模型(distributed Web log mining system，DWLMS)，以及基于 DWLMS 的局部频繁访问路径的更新算法 LFP 和全局频繁访问路径的更新算法 GFP. 该算法能够产生较少数量的频繁访问路径候选集，减小网络通信量，有效解决了 Web 访问信息的异地存储、实时增长、分布式算法通讯量等因素给模式分析过程带来的问题.

收稿日期：2005-07-20 **修回日期**：2005-09-09

基金项目：国家自然基金资助项目(No. 70431002)和北京电子科技学院重点实验室资助项目

作者简介：张克君(1972—)，男，博士；杨炳儒(1943—)，男，教授，博士生导师

1 DWLMS 模型及相关概念

1.1 DWLMS 基本思想

分布式 Web 使用挖掘系统模型 DWLMS 的基本思想(图 1)：采用基于分布式数据库基础上的全局—局部站点的数据挖掘模式，由全局和局部站点协同完成分布式全局模式挖掘任务. 全局用户控制挖掘的过程，发出挖掘指令. 局部站点接受指令，并执行任务——对增量部分的日志 Log_i^+ 进行预处理等，生成局部用户访问事务增量 d_i^+；执行局部增量挖掘算法，产生局部模式 F_i'，并保存于局部模式库中；将局部模式的支持数及局部模式 F_i' 传送给全局站点. 全局站点整合所有局部站点的局部模式 F_i'，从中找出全局模式知识 F，保存在全局模式知识库中，以供局部站点访问.

对于分布式系统，尽量减少传输量是至关重要的，提出的局部或全局挖掘算法将致力于减少传输无效信息的冗余量. 另外，如果系统在传输信息时采用广播的方式，则系统需要 $O(n^2)$的通信量，其中 n 为系统中站点的个数，而 DWLMS 采用全局—局部站点模式，因此系统在传输相同信息时只需要 $O(n)$的通信量.

1.2 相关概念与理论

定义 1 用户会话. 设 $V=\{v_1, v_2, \cdots, v_m\}$ 是网站上网页的集合，一个会话 S 是同一用户在一次浏览过程中连续请求的页面序列，它代表了用户对服务器的一次有效访问，记为 $S=\langle p_1, p_2, \cdots, p_n\rangle$，$p_i \in V$. 每一个会话具有惟一会话标识 sessionID.

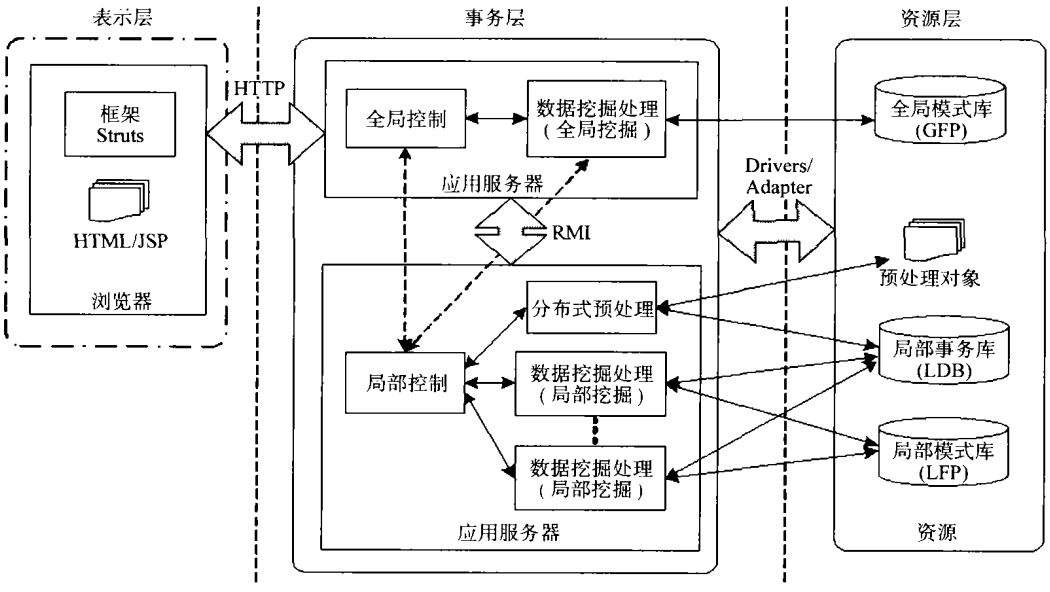

图 1　分布式 Web 日志挖掘系统体系结构(DWLMS)

Fig·1　Architecture of a distributed Web log mining system

定义 2　访问路径. 用户会话中由连续页面构成的子页面序列,可以表示为 $P=\langle p_1, p_2, \cdots, p_k\rangle$, $p_i \in V$. 一个 k—路径是长度为 k 的访问页面序列.

定义 3　最大向前访问路径. 用户会话中从起点开始的一次最大访问路径,简称 MFP.

定义 4　频繁访问路径. 最大向前路径 MFP 中满足一定支持度的子路径序列(包含频繁访问路径的 MFP 的个数占全部 MFP 数量的比例叫支持度). 局部事务数据库中生成的频繁访问路径集称为局部频繁访问路径集;全局事务数据库中生成的频繁访问路径集称为全局频繁访问路径集,记为 FP.

定义 5　事务数据库. 设分布式系统中位于 N 个站点的事务数据库分别为 $\{D_1, D_2, \cdots, D_N\}$, $D_B=D_1\cup D_2\cup\cdots\cup D_N$,则 $D_i(1\leqslant i\leqslant N)$ 称为局部事务数据库, D_B 称为全局事务数据库.

定理 1　设分布式挖掘系统中站点 i 上的局部频繁访问路径集为 $FP_i'=1,2,\cdots, N$,将所有站点上的 FP_i' 合并得到集合 FP′,则 FP′一定是全局频繁访问路径集 FP 的超集.

证明　要证 FP′是全局频繁访问路径集 FP 的超集,只要证 $\forall X\in FP$,则 $X\in FP'$.

设最小支持度为 minsup,每个站点的事务记录数为 n_i, $i=1,2,\cdots, N$, X 在第 i 个站点的支持数为 $S_i(X)$,因为 $X\in FP$,所以 X 的全局支持数

$$S(X)=\sum_{i=1}^{N} S_i(X)\geqslant\left[\sum_{i=1}^{n} n_i\right]\times \text{minsup}.$$

假设 $X\notin FP'$,则每个站点上 X 的支持数 $S_i(X)<n_i\times$ minsup,从而得到 $X\in$ FP,

$$S(X)=\sum_{i=1}^{N} S_i(X)<\left[\sum_{i=1}^{N} n_i\right]\times \text{minsup},$$

这与 $X\in$FP 矛盾. 证毕

定理 1 表明,所有局部站点提取的局部频繁访问路径的总和一定包含全局频繁访问路径集. 也就是说,若 X 是全局频繁的,则一定存在某个站点使得 X 在该站点是局部频繁的. 反之,如果 X 是局部频繁的,那 X 不一定是全局频繁的.

定义 6　重频繁路径. 若 X 既是某站点的局部频繁访问路径,又是全局频繁路径,则称 X 为该站点的重频繁路径. 显然,某站点的重频繁路径包含于该站点的局部频繁路径集合中.

性质 1　若 X 是某站点的重频繁路径,则 X 的所有子集也是该站点的重频繁路径.

根据定理 1 和性质 1 可得性质 2.

性质 2　若 X 是全局频繁 k—路径,则存在一个站点 $P_i(i=1,2,\cdots, N)$使得 X 以及它的所有 $(k-1)$路径子集在站点 P_i 为重频繁路径.

2　频繁路径挖掘过程算法

2.1　局部挖掘事务库生成

Web 使用挖掘的数据预处理过程如图 2 所示,包括数据净化、用户识别、会话识别、路径补充和事务识别等步骤[1-2]. 局部日志预处理的主要职责就是对本次该服务器日志增加数据 Log_i^+ 进行处理,产生对应的新增加的事务集合 d_i^+. 其中

事务识别环节是重要的,事务识别是对用户会话进行语义分组. 通常采用 Chen 等人提出的最大向前引用路径(简称为 MFP)[3]来定义事务. 事务的具体意义是:用户为获得一项有意义的信息所点击的页面序列,也就是用户会话中的每一次前进浏览的第一页到回退的前一页组成的路径. 获得有意义的事务,是保证 Web 使用挖掘结果可靠性的关键. 为此,本文提出了一种事务识别算法如下.

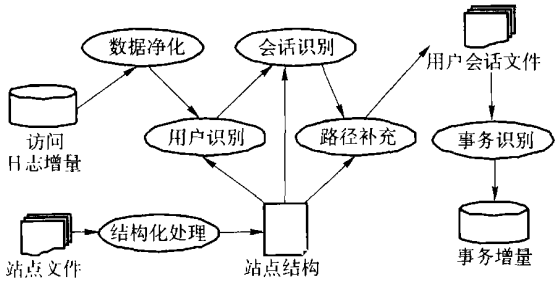

图 2 局部 Web 日志挖掘预处理的具体过程

Fig. 2 Preprocessing of local Web log mining

设$\langle x_1,\cdots,x_m\rangle$表示一个用户会话,设$\langle y_1,\cdots,y_m\rangle$代表一个含有潜在 MFP,flag 表示当前用户浏览路径方向(flag=TRUE 表示向前,flag=FALSE 表示后退),事务库用 d_i^+ 表示,网站的拓扑结构用有向图 G 表示. 则用户会话文件生成 MFP 事务库的算法为:

算法 1 基于站点图结构的事务识别算法 MFP

输入:用户会话文件

输出:MFP 事务库 d_i^+

(1) for each Session{//扫描会话文件

(2) $y_1=x_1$; $j=2$; $i=2$;flag=TRUE;

(3) for ($i\leqslant m$; $i++$){

(4) if {$x_i=y_k$, $1\leqslant k<j$){

(5) if (flag=TRUE) then {

(6) 将$\langle y_1,\cdots,y_{j-1}\rangle$作为 MFP 输出到事务库 d_i^+ 中;

(7) $j=k+1$; flag=FALSE;}

(8) else{$j=k+1$;}

(9) }else{

(10) if$\langle y_{j-1}, x_i\rangle$为 G 的有向边 then{//经过链接到达

(11) $y_j=x_i$; $j=j+1$;flag=TRUE;

(12) }else{//该页由用户直接输入 url 访问的

(13) 将$\langle y_1,\cdots,y_{j-1}\rangle$作为 MFP 输出到事务库 d_i^+ 中;

(14) $y_1=x_i$; $j=2$; flag=TRUE;}}}

(15) if (flag=TRUE) then 将$\langle y_1,\cdots,y_{j-1}\rangle$作为 MFP 输出到事务库 d_i^+ 中;

(16) }

局部日志数据预处理的结果是局部事务库中的增量集 d_i^+.

2.2 局部频繁访问路径更新算法

为便于表述,定义如下符号,如表 1 所示.

表 1 频繁访问路径挖掘相关符号的定义

Table 1 Symbols definition about frequent access paths mining

事务数据库	路径 X 的支持数	频繁路径集	重频繁路径集
d_i^+	$\sup_i^+(X)$		
D_i	$\sup_i(X)$	$L_k(i)$	$H_k(i)$
$D_i'=D_i\cup d_i^+$	$\sup_i'(X)$	$L_k'(i)$	$H_k'(i)$

其中,D_i' 为第 i 站点更新后的事务数据库,显然,$D_i'=D_i\cup d_i^+$,$D_i=D_i'-d_i^+$;$\sup_i(X)$为 D_i 中路径 X 的支持数;$L(i)$为 D_i 中的频繁访问路径集;$L_k(i)$为 $L(i)$中的频繁 k—路径集;$H(i)$为 D_i 中的重频繁路径集;$H_k(i)$为 $H(i)$中的重频繁k—路径集;$\sup_i'(X)$为更新后事务数据库 D_i' 中路径 X 的支持数;$L'(i)$为 D_i' 中的频繁路径集;$L_k'(i)$为 $L'(i)$中的频繁 k—路径集;$H'(i)$为 D_i' 中的重频繁路径集;$H_k'(i)$为 $H'(i)$中的重频繁 k—路径集;$\sup_i^+(X)$为 d_i^+ 中路径 X 的支持数.

根据支持数的定义可得如下性质 3.

性质 3 $\sup_i'(X)=\sup_i(X)+\sup_i^+(X)$.

全局频繁路径集的获取,是建立在各局部站点局部频繁路径集更新整合基础上的. 在前次获取频繁路径集的挖掘过程中,已经得到 $L(i)$和$\sup_i(X)$,$\forall X\in L(i)$. 因此,局部频繁路径集更新任务就是,给定 D_i, D_i', d_i^+, $L(i)$和$\sup_i(X)$,对于 $\forall X\in L'(i)$,如何高效计算 $L'(i)$和$\sup_i'(X)$.

为了求解 D_i' 中的频繁路径集 $L'(i)$,给出局部频繁路径集更新算法 LFP,其方法是:用类似于 Apriori 算法的迭代方法,从频繁 k—路径候选集 $C_k(i)$中求出所有频繁 k—路径集 $L_k'(i)$. 首先需要用频繁 k—1 项集生成长度为 k 的候选集,记为$CA_k(i)=\text{candidate-gen}(L_{k-1}'(i))$. 但是,在分布式环境中,$L_{k-1}'(i)$中有些路径集不是全局频繁的,由 $L_{k-1}'(i)$产生的 $C_k(i)$集合必定比较大,这是不可取的. 这里给出 k—路径候选集的生成算

法，令 $CH_k(i)$=candidateFP-gen($L'_{k-1}(i)$)，函数 candidateFP-gen 的算法如下.

算法 2　局部候选路径生成算法 candidateFP-gen

输入：频繁 k-路径集 L_k

输出：频繁($k+1$)-路径候选集

(1) for each$\langle y_1,\cdots,y_k\rangle \in L_k$｛

(2) N=｛G 中全部以 y_k 为起点的有向边的终点的集合｝;

(3) for each $v\in N$

(4) if $v\neq y_i(1\leqslant i\leqslant k)$ and$\langle y_2,\cdots,y_k,v\rangle \in L_k$｛

(5) C=$\langle y_1,\cdots,y_k,v\rangle$;

(6) S=｛路径 C 所有长度为 k 的子路径｝;

(7) for each $s\in S$

(8) if $s\in L_k$ 把 C 加入到 C_{k+1};｝

(9) ｝

(10) ｝

因为 $H'_{k-1}(i)\subseteq L'_{k-1}(i)$，所以 $CH_k(i)\subseteq CA_k(i)$. 根据性质 2，事务数据库更新后全局频繁 k-路径集

$$FP'_k\subseteq \bigcup_{i=1}^{N} CH_k(i)=\bigcup_{i=1}^{N} \text{candidate-gen}(H'_{k-1}(i)).$$

显然通过重频繁路径集 $H'_{k-1}(i)$ 通过 candidateFP-gen 函数产生一个数量较小的频繁项目候选集 $C_k(i)$，这也是提出重频繁路径集概念的原因.

令 $C_k(i)$=candidateFP-gen($H'_{k-1}(i)$). 将 $C_k(i)$划分为两个部分：$P_k(i)=C_k(i)\cap L_k(i)$，$Q_k(i)=C_k(i)-P_k(i)$. 其中 $P_k(i)$是 D_i 中的频繁 k 项集；$Q_k(i)$是 D_i 中的非频繁项目集，即 $Q_k(i)\notin L_l(i)$.

$\forall X\in P_k(i)$，利用性质 3 可知：$\sup'_i(X)=\sup_i(X)+\sup_i^+(X)$，其中$\sup_i^+(X)$通过扫描 d_i^+ 求得. 如果$\sup'_i(X)\geqslant|D'_i|\times$minsup，则 $X\in L'_k(i)$.

$\forall X\in Q_k(i)$，$\sup_i(X)$未知，但$\sup_i(X)<|D_i|\times$minsup. 通过下面的定理 2，可将 $Q_k(i)$中的某些非频繁路径删掉.

定理 2　如果 $X\notin L(i)$，并且$\sup_i^+(X)\leqslant|d_i^+|\times$minsup，则 $X\notin L'(i)$.

证明　$X\notin L(i)$，则$\sup_i(X)<|D_i|\times$minsup. 因此，

$$\sup'_i(X)=\sup_i(X)+\sup_i^+(X)<|D_i|\times\text{minsup}+|d_i^+|\times\text{minsup}=(|D_i|+|d_i^+|)\times\text{minsup}=|D'_i|\times\text{minsup},$$

从而，由 $L'(i)$的定义可知 $X\notin L'(i)$. 证毕

为了将 $Q_k(i)$中的某些非频繁路径删去，$\forall X\in Q_k(i)$，扫描 d_i^+，计算$\sup_i^+(X)$. 根据定理 2，若$\sup_i^+(X)\leqslant|d_i^+|\times$minsup，则将 X 从 $Q_k(i)$中删掉. 对于 $Q_k(i)$中的剩余项 X，计算 $\sup'_i(X)=\sup_i(X)+\sup_i^+(X)$. 若$\sup'_i(X)\geqslant|D'_i|\times$minsup，则将 X 加入 $L'(k)$中.

综上所述，算法 LFP 在站点 i 的第 k 次循环的操作描述如下.

算法 3　局部频繁路径集更新算法 LFP

输入：局部事务集合

输出：局部频繁访问路径集、局部频繁访问路径支持数

(1) if $k=1$ then $C_1(i)=V_i$ else

(2) $C_k(i)$=candidateFP-gen($H'_{k-1}(i)$);

(3) if $C_k(i)=\varnothing$ then break;

(4) 将 $C_k(i)$划分为两部分：$P_k(i)$和 $Q_k(i)$;

(5) for each $X\in P_k(i)\cup Q_k(i)$，扫描 d_i^+，计算$\sup_i^+(X)$;

(6) for each $X\in P_k(i)$，计算 $\sup'_i(X)=\sup_i(X)+\sup_i^+(X)$;

(7) for each $X\in Q_k(i)$

(8) if $\sup_i^+(X)\leqslant|d_i^+|\times$minsup then delete X from $Q_k(i)$;

(9) for each $X\in Q_k(i)$

(10) 扫描 D_i，计算$\sup'_i(X)=\sup_i(X)+\sup_i^+(X)$;

(11) for each $X\in P_k(i)\cup Q_k(i)$

(12) if $\sup'_i(X)\geqslant|D'_i|\times$minsup then 将 X 加入 $L'_k(i)$.

2.3　全局频繁路径集产生算法

全局频繁路径集的更新是建立在局部频繁路径集更新基础上的，即要找出全局频繁 k-路径集，必须求解每个局部站点的频繁 k-路径集. 前面已经论述了局部频繁路径集的更新算法 LFP，本部分给出基于 DWLMS 系统的全局频繁路径集更新算法 GFP. 算法 GFP 采用迭代方法求解事务数据库更新后的全局频繁 k-路径集，$k=1,2,\cdots,N$. 各站点第 k 次循环迭代的步骤描述如下.

算法 4 全局频繁路径集更新算法 GFP

(1) 各局部站点根据局部频繁路径集更新算法 LFP 的第 k 次循环迭代的过程，计算事务数据库更新后的局部频繁路径候选集 $C_k(i)$及其支持数，并将其发送给全局站点；保存局部频繁路径集 $L'_k(i)$及其局部支持数；

(2) 全局站点对所有局部频繁路径候选集 $C_k(i)$进行合并，计算其全局支持数，扫描计算 $\sup'(X)\geqslant|D'|\times\text{minsup}$，得到全局频繁 k—路径集 FP_k，并将 FP_k 发送到各局部站点；

(3) 各局部站点将收到全局发来的 FP_k 与本站点的局部频繁路径候选集 $L'_k(i)$进行比较得出本站点的重频繁路径集 $H'_k(i)$，以便利用 LFP 算法计算事务数据库更新后的局部频繁$(k+1)$—路径集；

(4) $k=k+1$，转第(1)步.

3 实验结果与分析

利用 4 台 PC 构建采用 DWLMS 模型的分布式 Web 使用挖掘系统，节点配置为 PⅣ1.8G，256 MB，运行平台 Windows 2000 Server. 其中 3 台微机存储分布式事务数据库，并执行局部挖掘任务；另一台微机执行全局挖掘算法任务. 分布式软件平台及 LFP/GFP 更新算法采用 J2EE 技术实现. 数据源为北京科技大学网站(www.ustb.edu.cn) 2 周的访问日志，日志数据量总共是 145 MB. 将其按时间分段并分布于 3 台微机中，进行分布式用户频繁访问路径挖掘算法 LFP/GFP 的运行结果和性能测试. 通过多次实验，结果表明频繁路径更新算法 LFP/GFP 的正确、有效性. 例如，当，全局 4—频繁访问路径及其支持数、局部 4—频繁路径及其支持数的实验结果如表 2. 分析 FP_41 路径可以看出，目前访问报考研究生的相关信息的频率是较高的，这样的知识是有意义的.

为了测试算法 LFP/GFP 的性能，用 Java 语言实现了 LFP/GFP 算法和现有的分布式挖掘算法 DMA 算法[9]. 在不同支持度和不同数据库更新情况下，将 GFP 算法同 DMA 算法执行时间进行比较，结果如图 3 所示. 从中可以看出，在各种情况下，GFP 算法求解频繁路径集所需执行时间比重新运行 DMA 算法求解频繁路径集所需执行时间少得多，进而说明本文提出的分布式系统中

表 2 DWLMS 系统中局部频繁路径和全局频繁路径的实验结果($k=4$, Support$=0.25$)

Table 2 Results of local frequent paths and global frequent paths algorithms test

局部站点 Site—1		局部站点 Site—2		局部站点 Site—3		全局站点 Site—G	
频繁 4—路径	支持数	频繁 4—路径	支持数	频繁 4—路径	支持数	频繁 4—路径	支持数
FP_41	2665	FP_41	2108	FP_41	2386	FP_41	7159
FP_42	2218	FP_42	1866	FP_43	1905		
FP_43	2139						

注：FP_41→/ustbcn/index.asp→/yjsy/index.htm→/yjsy/index3.htm→/yjsy/zsss2005ml.htm；

FP_42→/ustbcn/index.asp→/yjsy/index.htm→/yjsy/index5.htm→/yjsy/aj.htm；

FP_43→/ustbcn/index.asp→/lib.ustb.edu.cn/index.htm→/libcd/netdb.htm→/libcd/testdb.htm.

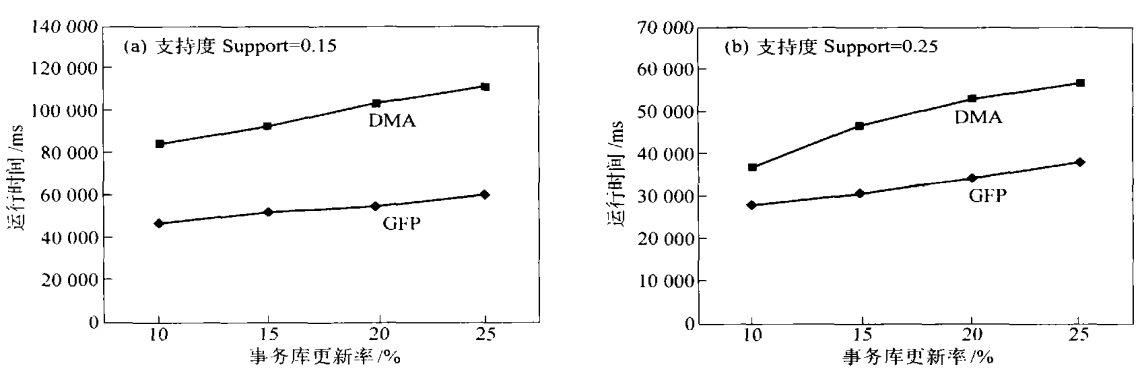

图 3 GFP 算法与 DMA 算法执行效率比较

Fig.3 Comparison of efficiency between GFP algorithm with DMA algorithm

频繁路径集的更新算法 LFP/GFP 算法具有较高的效率. 从理论上讲,局部更新算法 LFP 在求解频繁路径集 $L'_k(i)$时,利用了前次挖掘过程中产生的频繁路径集 $L_k(i)$和支持数$\sup_i(X)$及全局第 $k-1$ 次循环中产生的重频繁路径集 $H'_{k-1}(i)$. 显然,GFP 的优点是:(1)减少了数据库扫描的次数;(2)减少了 k 次叠代产生的频繁路径候选集 $C_k(i)$的大小,也就减少了频繁路径候选集生成的执行时间;(3)由于频繁路径候选集 $C_k(i)$数量的减小,从而减少了局部站点向全局站点传输数据量. 因此 GFP 算法有较高的执行效率.

4 结论

本文针对目前 Web 频繁访问路径算法的缺陷,对多镜像站点环境下的 Web 分布式使用挖掘系统中频繁访问路径模式挖掘进行了研究,主要讨论了当局部站点访问日志量增加时的情况下,用户频繁访问路径如何高效更新的问题. 提出了一种多镜像站点环境下的 Web 使用挖掘模型 DWLMS,以及基于 DWLMS 的局部频繁访问路径的更新算法 LFP 和全局频繁访问路径的更新算法 GFP. 实验结果表明这些算法的正确性、高效性,能够有效解决多镜像站点环境引起的 Web 访问信息的异地存储、实时增长、分布式算法通讯量等因素给频繁访问路径模式发现过程带来的困难.

参 考 文 献

[1] 韩家炜，孟小峰，王静，等. Web 挖掘研究. **计算机研究与发展**,2001,38(4):405

[2] Srivastava J, Cooley R, Deshpande M, et al. Web usage mining: discovery and application of usage patterns from Web data. **SIGKDD Explorations**, 2000, 1(2): 12

[3] Chen M S, Park J S, Yu P S. Efficient data mining for path traversal patterns. **IEEE Trans Knowl Data Eng**, 1998, 10(2): 209

[4] Nanopoulos A, Manolopoulos Y. Mining patterns from graph traversals. **Data Knowl Eng**, 2001, 37: 243

[5] Kargupta H, Hamzaoglu I, Stafford B. Scalable distributed data mining using an agent based architecture // Proc of KDD97. Menlo Park: AAAI Press, 1997: 211

[6] Kargupta H, Park B, Johnson E et al. Collective data mining from distributed vertically partitioned feature space // Workshop on Distributed Data Mining, International Conference on Knowledge Discovery and Data Mining. New York, 1998

[7] Masseglia F, Teisseire M, Poncelet P. Real time Web usage mining with a distributed navigation analysis // Proceedings of the 12th International Workshop on Research Issues in Data Engineering. San Jose, 2002: 169

[8] 吉根林,杨明,赵斌,等. 基于 DDMINER 分布式数据库系统的频繁项目集更新. **计算机学报**, 2003, 26(10): 1387

[9] Cheung D W, Ng V T, Fu A W. Efficient mining of association rules in distributed databases. **IEEE Trans Knowl Data Eng**, 1996, 8(6): 911

Construction and algorithms of distributed web usage pattern mining

ZHANG Kejun[1,2], *YANG Bingru*[2], *ZHAO Geng*[1], *QU Wenlong*[2], *LI Xin*[2]

1) Department of Computer Science and Technology, Beijing Electronic Science and Technology Institute, Beijing 100070, China

2) Information Engineering School, University of Science and Technology Beijing, Beijing 100083, China

ABSTRACT A distributed Web log mining system model (DWLMS) is presented. Based on the analysis on the procedure and algorithm of Web frequent access pattern mining, the more general incremental updating algorithms of local frequent paths (LFP) and global frequent paths (GFP) in a distributed database system based on DWLMS are proposed for discovering the frequent access paths. Some troubles produced by real time incremental distributed Web access information and more communication data are solved better by the algorithms. The method was realized simply and tested with real world Web log information in laboratory, and the results show that the algorithms are valid.

KEY WORDS distributed data mining; Web access pattern mining; Web log mining; frequent path