第 33 卷 第 8 期 计 算 机 工 程 2007 年 4 月
Vol.33 No.8 Computer Engineering April 2007
·博士论文· 文章编号：1000—3428(2007)08—0014—03 文献标识码：A 中图分类号：TP311

# 一种基于 Close 模式发现用户频繁访问路径的方法

陈 敏，苗夺谦
(同济大学计算机科学与工程系，上海 200092)

**摘 要：**Web 日志挖掘的一个主要任务是获得用户的浏览模式，这对 Web 站点的改进和为用户提供个性化服务提供了非常有价值的潜在信息。该文在分析用户访问模式的特点后，提出了 Close 模式的概念，基于此概念提出了一种挖掘用户频繁访问模式的 Close 算法。该算法利用频繁访问模式的封闭特性，挖掘出既是频繁的又是封闭的访问模式，在一定程度上减少了下一阶段“寻找最大频繁访问模式”的工作量。用实际数据对算法的性能进行了验证和分析。

**关键词：**Web 挖掘；频繁访问模式；访问模式的顺序子集；Close 模式

# Method for Discovering Users' Frequent Access Patterns Based on Close Patterns

CHEN Min, MIAO Duoqian
(Department of Computer Science and Engineering, Tongji University, Shanghai 200092)

**【Abstract】** One primary task of Web log mining is to discover and identify users' access patterns, which provides very valuable potential information for the improvement of Web sites and the users' personalized service. The paper proposes a concept of Close patterns after analyzing the characteristic of users' access patterns. A Close algorithm for discovering users' frequent access patterns is proposed based on this concept. The Close algorithm discovers frequent and Close patterns, which relieves the next phrase workload of finding maximal frequent access patterns to some extent, by making use of the Close property of frequent access patterns. The algorithm performance is tested and analyzed by actual datum.

**【Key words】** Web mining; Frequent access patterns; Sequential subsets of access patterns; Close patterns

World Wide Web 作为收集、传播与交换信息的平台，已经对社会的各个方面产生了深远的影响。当用户在网上冲浪时，Web 服务器以 Web 日志的形式记录了用户在网上的行为。随着 WWW 以惊人的速度迅速发展，Web 站点服务器每天产生了大量的日志，分析这些日志数据能够发现隐藏的用户访问模式规则，具有十分重要的意义。

Web日志挖掘[1]又称Web使用信息的挖掘，是指通过研究Web服务器的日志文件，抽取有用的信息和知识，以提高Web服务的质量。挖掘Web访问模式即是从Web日志中发现用户的浏览模式，研究这些模式进行关联规则分析时，不仅能够从使用者的角度改进站点的设计，如在 2 个高度相关的页面之间提供链接，而且能够在电子商务应用中制定更好的市场方案，如根据客户的爱好分发不同的广告等。

因此，挖掘用户的浏览模式就成为 Web 日志挖掘的主要任务之一。本文在 Web 日志挖掘的频繁模式发现阶段，提出了一种基于 Close 模式的挖掘算法。此算法利用文中所提出的访问模式的封闭(Close)特性对 Web 日志进行挖掘，它在每次循环中提前派生出一部分长度大于本次循环的频繁模式，最终挖掘出既是频繁的又是封闭的访问模式(频繁 Close 模式)，为下一阶段“在众多的频繁模式中找出对分析阶段更有意义的最大模式”提供了准备，在一定程度上减少了其工作量，这是相比其它一般挖掘算法的优越之处。

## 1 相关问题的描述

Web日志挖掘过程一般分为 3 个阶段[2]：数据预处理，模式发现和模式分析。在预处理阶段结束后，模式发现结果的好坏直接影响到模式分析的效果。在关联规则分析中，模式发现阶段的核心任务就是发现用户的浏览模式，所以此时模式发现阶段也称为挖掘算法实施阶段。Chen等[4]提出了基于类Apriori的改进算法FS和SS，所谓类Apriori算法即是对数据挖掘中发现频繁项集的Apriori算法[3]进行改进而得到的，这是由于Web使用的挖掘对象——访问模式是一种序列模式，不同于数据挖掘中的无序数据项。Chen等首先引入了最大前向引用(Maximal Forward Reference，MFP)的概念，即把用户会话分割为更小的事务[5]，且每个事务中不包含后退的用户的访问路径，然后把挖掘用户的访问模式过程分为以下 2 个阶段：

(1)最大前向引用中发现大引用序列(即支持度不小于用户指定的最小支持度阈值的频繁访问模式)；

(2)从大引用序列中找出最大引用序列(即不被其它任何大引用序列所包含的大引用序列)。

这 2 个阶段中，相比而言第 2 阶段比较容易，因此第 1 阶段就成为解决的关键。FS 和 SS 正是为解决第 1 阶段的问题而提出的算法。然而不难想象，通常发现的频繁模式数目很多，从庞大的数据量中找出最大引用序列，工作量也是不

**基金项目：**国家自然科学基金资助项目(60175016，60475019)；山西省高校高科技研究开发项目(20051277)
**作者简介：**陈 敏(1980－)，女，博士生，主研方向：数据挖掘，人工智能，粗糙集，语义 Web；苗夺谦，教授、博导
**收稿日期：**2006-04-25 **E-mail：**minchen.tj@163.com

容忽视的。本文在关注于第 1 阶段解决的同时也充分考虑到第 2 阶段问题的解决，提出了挖掘用户频繁访问模式的 Close 算法。

## 2 Web 挖掘中的 Close 算法

本文在分析用户访问模式特点的基础上，提出了封闭访问模式(Close 模式)的概念，然后基于此概念提出一种挖掘用户频繁访问模式的 Close 算法。此算法利用频繁访问模式的封闭特性，挖掘出既是频繁的又是封闭的访问模式，大大减少了第 2 阶段“寻找最大引用序列”的工作量。

### 2.1 相关定义和定理

由于用户访问模式是建立在用户访问路径的基础上，是一种序列模式。为了说明问题方便，假设当前的事务数据库为 T(MFP 的集合)，其中的每个事务 t(t∈T)即是一个最大前向引用，作出以下定义：

**定义 1**(访问模式的顺序子集) 已知访问模式X=<$X_1,X_2,\ldots,X_m$>和Y=<$Y_1,Y_2,\ldots,Y_n$>，如果它们满足以下 2 个条件：(1)M≤n；(2)存在m个连续的整数$1\le i_1<\ldots<i_m\le n$，使得$X_1=Y_{i_1},\ldots,X_m=Y_{i_m}$，则称Y顺序包含X，或X是Y的顺序子集，简记为$X\subseteq^s Y$，其中$\subseteq^s$右上角的s是代表Sequence (序列)的意思。若$X\neq Y$，则称X是Y的顺序真子集，简记为$X\subset^s Y$。

**定义 2**(访问模式的顺序中心子集) 已知访问模式 X 和 Y，且 X 是 Y 的顺序子集(即$X\subseteq^s Y$)。若存在一个访问模式 Z，满足条件：$X\subseteq^s Z$ 且 $Z\subseteq^s Y$，则称 Y 以 X 为中心顺序包含 Z，或称 Z 是 Y 以 X 为中心的顺序子集，简记为 $Z\subseteq^s_X Y$。若 $X\subset^s Z$ 且 $Z\subset^s Y$，即 $Z\neq X$ 且 $Z\neq Y$，则称 Z 是 Y 以 X 为中心的顺序真子集，简记为 $Z\subset^s_X Y$。

**定义 3**(访问模式的顺序中心交集) 已知访问模式 X、Y 和 Z，且 $Z\subseteq^s X$，$Z\subseteq^s Y$。若存在一个访问模式 P，满足 $P\subseteq^s_Z X$ 且 $P\subseteq^s_Z Y$，则称 P 是 X 和 Y 以 Z 为中心的顺序子集，简记为：$P=X\cap^s_Z Y$。

**定义 4**(Close模式) 已知访问模式X和事务集S(X)，且有$S(X)=\{t_i \mid X\subseteq^s t_i 且 t_i\in T,\ 1\le i\le \|T\|\}$(其中||T||表示T中所包含的事务的数目)。假设$S(X)=\{t_1,t_2,\ldots,t_m\}$，则由定义 3 可以得到访问模式clo(X)：

$$clo(X)=\cap^s_X S(X)=t_1\cap^s_X t_2\cap^s_X\ldots\cap^s_X t_m$$

称 clo(X)是 X 的封闭访问模式，即 Close 模式。

由 Close 模式的定义，不难得出如下性质：

(1)$S(X)=S(clo(X)) \Leftrightarrow clo(X)=clo(clo(X))$

(2)$X\subseteq^s clo(X)$

(3)$X\subset^s Y \Rightarrow S(X)=S(Y)\cup S(X-Y)$，其中$S(X-Y)=\{t_i \mid X\subseteq^s t_i 且 Y\not\subset^s t_i,\ t_i\in T,\ 1\le i\le\|T\|\}$

证明：

(1)根据定义 4 显然可证。

(2)根据定义 3 和定义 4 显然可证。

(3)对于任意的 t∈T，若 $Y\subseteq^s t$，由条件 $X\subseteq^s Y$ 可知，必然有 $X\subseteq^s t$；但是对于任意的 t∈T，若 $X\subseteq^s t$，则由条件 $X\subseteq^s Y$ 不能确定 t 是否顺序包含 Y。所以顺序包含 X 的事务集可分为 2 部分的并集：一部分是顺序包含 Y 的事务集；另一部分是顺序包含 X，但是不顺序包含 Y 的事务集，即 S (X) = S(Y) ∪ S(X-Y)。

基于以上的定义和性质，得到如下 2 个定理和 2 个推论：

**定理 1** 一个访问模式 X 的支持度与它的封闭访问模式 clo(X)的支持度是相等的，即 support(X)=support(clo(X))。

证明 访问模式 X 的支持度是$\text{support}(X)=\frac{\|S(X)\|}{\|T\|}$(其中 T 是事务数据库，||S(X)||和||T||分别表示 S(X)包含的事务数和 T 包含的事务数)，封闭访问模式 clo(X)的支持度：$\text{support}(clo(X))=\frac{\|S(clo(X))\|}{\|T\|}$(||S(clo(X))||表示 S(clo(X))包含的事务数)，由性质(1)可知：S(X)=S(clo(X))，因此||S(X)||=||S(clo(X))||，support(X)=support(clo(X))。

**推论 1** 若访问模式 X 是频繁的，其 clo(X)也必是频繁的。此时，称 clo(X)是 X 的频繁 Close 模式。

证明 由定理 1 显然可证。

**定理 2** 已知访问模式 X 和它的 Close 模式 clo(X)，若存在访问模式 Y，满足 $Y\subset^s_X clo(X)$，则 Y 和 X 的支持度相等，即 support(X)=support(Y)。

证明 根据定义 2 可知，$Y\subset^s_X Clo(X)$等价于 $X\subset^s Y$ 且 $Y\subset^s clo(X)$。因为 $X\subset^s Y$，由性质(3)可得

$$S(X)=S(Y)\cup S(X-Y) \tag{1}$$

又因为 $Y\subset^s Clo(X)$，同理可得

$$S(Y)=S(clo(X))\cup S(Y-clo(X)) \tag{2}$$

由性质(1)可知 S(X)=S(Clo(X))，代入式(2)得

$$S(Y)=S(X)\cup S(Y-clo(X)) \tag{3}$$

把式(3)带入式(1)得

$$S(X)=S(X)\cup S(X-Y)\cup S(Y-clo(X))$$

$$\Leftrightarrow\ S(X-Y)\cup S(Y-clo(X))\subseteq S(X)$$

$$\Leftrightarrow\ S(X-Y)\subseteq S(X)\ 且$$

$$S(Y-clo(X))\ \subseteq S(X) \tag{4}$$

根据式(2)可知 $S(Y-clo(X))\cap S(clo(X))=\phi$，因为 S(X)=S(clo(X))，所以 $S(Y-clo(X))\cap S(X)=\phi$，又因为式(4)中 $S(Y-clo(X))\subseteq S(X)$，所以必然有：

$$S(Y-clo(X))=\phi \tag{5}$$

把式(5)带入式(3)得

$$S(X)=S(Y)$$

根据访问模式的支持度可知 support(Y)= support(X) ⇔ S(Y)=S(X)，即得证。

**推论 2** 已知频繁访问模式 X 和它的 Close 模式 clo(X)，若存在访问模式 Y，满足 $Y\subset^s_X clo(X)$，则 Y 也是频繁的。

证明 由定理 2 显然可证。

### 2.2 Close 算法

本节给出了使用 Close 算法发现最大频繁访问模式(即所有的频繁 Close 模式)的详细过程。在算法的每次循环中，利用频繁 Close 模式的特性，可以提前派生出一部分长度大于本次循环的频繁访问模式，这样在下次循环时能够减少扫描数据库的次数。算法结束时，不仅挖掘出所有的频繁访问模式，而且也识别出其中的频繁 Close 模式。实际上，既是封闭(Close)的又是频繁的访问模式的数目通常是远远小于频繁访问模式的数目。因此，在“发现用户访问模式”的第 2 阶段，通过扫描数目相对少得多的频繁 Close 模式，发现最终的大引用序列将会花费很少的时间。

为了提高算法的效率，采用文献[6]中的 DHP 算法的思想利用哈希表来生成候选访问模式，并且把每次循环时产生

的访问模式集 Access Patterns 和循环后产生的频繁访问模式集 Frequent Access Patterns 分别分成 3 个域(见表 1)。

**表 1 $AP_K$(Access Patterns)和$FAP_K$(Frequent Access Patterns)**

| 集合 | 域 | 属性 |
|---|---|---|
| $AP_k$ | C<br>Close<br>support | 长度为 k 的候选访问模式<br>C 的 Close 模式：Close=clo(C)<br>访问模式的支持度计数:<br>support=count(Close)=count(C) |
| $FAP_k$ | L<br>Close<br>support | 长度为 k 的频繁访问模式<br>L 的 Close 模式：Close=clo(L)<br>访问模式的支持度计数:<br>support=count(Close)=count(L) |

**Close 算法**

输入 数据库 T，最小支持度 min_sup。

输出 频繁 Close 模式集及其支持度。

Step 1 产生 1－阶候选访问模式集，相应的Close模式和支持度，即生成{$AP_1$.C, $AP_1$.Close, $AP_1$.support}；

Step 2 根据最小支持度min_sup，把$AP_1$中支持度大于等于 min_sup 的项放入$FAP_1$集合中，得{$FAP_1$.L, $FAP_1$.Close, $FAP_1$.support}；然后派生出长度大于 1 的频繁访问模式，即{$FAP_j$.L, $FAP_j$.Close, $FAP_j$.support}(j>1)；

Step 3 k←2；

Step 4 若$FAP_{k-1}$.L=$\phi$，则终止；否则，转Step 5；

Step 5 由哈希表产生k－阶候选访问模式$AP_k$.C，并且$AP_k$.Close←$\phi$，$AP_k$.support←0；

Step 6 首先检查$AP_k$.C中的候选模式是否已经存在于$FAP_k$.L中，若存在，表示它已是频繁的，从$AP_k$中删除此模式对应的所有项；然后，扫描数据库T，计算k－阶访问模式相应的Close模式和支持度，即{$AP_k$.C, $AP_k$.Close, $AP_k$.support}；

Step 7 根据最小支持度min_sup，把$AP_k$中支持度大于等于min_sup的项放入$FAP_k$中，得到{$FAP_k$.L, $FAP_k$.Close, $FAP_k$.support}；然后派生出长度大于k的频繁访问模式，即{$FAP_j$.L, $FAPj_k$.Close, $FAP_j$.support}(j>k)；

Step 8 k ← k+1，转 Step 4；

Step 9 Answer ← $\bigcup_{j=1}^{j=k-1}$ {$FAP_k$.Close,$FAP_k$.support}。

算法的Step 1 和Step 6 中生成k－阶候选访问模式对应的Close模式的规则如下：若一个候选模式(设为p)是第 1 次生成，其Close模式为顺序包含它的事务t；若候选模式不是第 1 次生成，根据访问模式的顺序中心交集的定义(定义 3)求此时的Close模式，即p.Close←p.Close $\cap_p^s$ t。Step 2 和Step 7 中的派生规则如下：假设存在p∈$FAP_k$.L，且||p.Close||≥ ||p||+1，则得到集合$Q_0$←{q|q $\subseteq_p^s$ p.Close and q≠p}，对于任意的q∈$Q_0$，有$FAP_{||q||}$← $FAP_{||q||}$ ∪ {q, p.Close, p.support}(其中||q||表示q包含的页面数)。$Q_0$就是满足p.Close以p为中心的且不等于p的顺序子集的集合。因为根据推论 1 和推论 2 可知：第k次循环，若p∈$FAP_k$.L，当||p.Close||≥||p||+1 时，p.Close所包含的每个以p为中心的且不等于p的顺序子集均是一个频繁访问模式，且其长度大于k。最后，算法挖掘出所有的频繁Close模式(Step 9)。

## 3 实验数据与分析

为了评价 Close 算法的性能，本文在主频为 Intel Pentium III 933MHz、256MB 内存和 Windows 2000 操作系统的 PC 机上对该算法进行了测试，并与挖掘频繁访问模式的其它算法进行了比较，所有的程序代码由 C++ Builder 6.0 编写，数据库是微软的 SQL Server7.0。采用的数据为同济大学网站从 26/Dec/2005:10:03:00 至 26/Dec/2005:23:59:00，共 60 278 条记录，经过数据清洗后有 8 663 条记录，再应用文献[4]中的 MFP 算法转换为最大前向引用路径的集合，就构成了拥有 2 571 个事务的事务数据库。

图 1 给出了基于不同的最小支持度，FS、SS 和 Close 这 3 个算法各自的运行时间。3 个算法有相似的时间特性，且 Close 算法的平均时间多于 FS 和 SS 算法，这是因为在每次循环时 Close 算法要多一步计算每个访问模式的 Close 模式。图 2 给出了扫描频繁 Close 模式(Close 算法计算所得)和频繁访问模式(FS 和 SS 算法计算所得)分别发现最大频繁访问模式的时间，图中各表示为 Close 曲线及 FS 和 SS 曲线。由于频繁 Close 模式的数目要少于频繁访问模式的数目，因此 Close 曲线上的每个时间点均在 FS 和 SS 曲线的下方，且支持度小于等于 3 时 2 个曲线的差值变大。因为随着支持度的减小，虽然 3 个算法生成的最终频繁模式都逐渐增多，但是相对于 FS 和 SS 算法生成的频繁访问模式而言，Close 算法会生成数目相对更少的频繁 Close 模式，这时扫描频繁 Close 模式发现最大频繁模式会花费更少的时间。

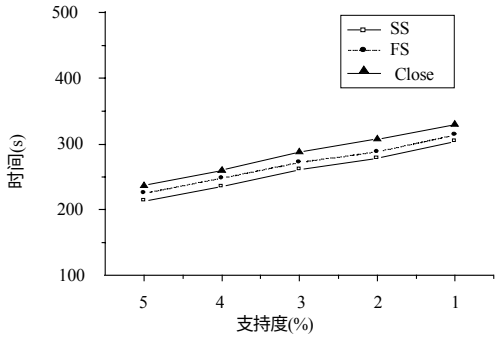


**图1 3种挖掘算法的运行时间**

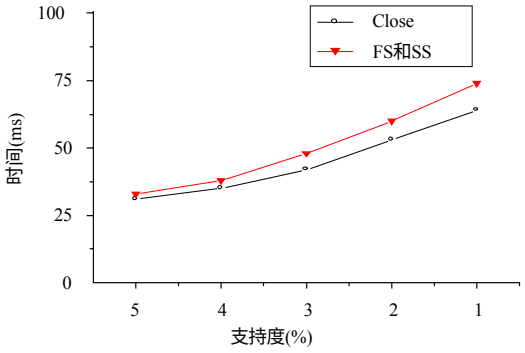


**图2 第3阶段发现最大频繁模式的时间**

综合图 1 和图 2 可知，虽然在挖掘频繁访问模式阶段中 Close 算法会花费较多的时间，但在发现最大频繁模式的最后一个阶段，Close 算法能花费较少的时间，且挖掘出的频繁访问模式数目越大(支持度越小)，就越能体现出 Close 算法的优越之处。

## 4 结论

从用户浏览网页的行为中挖掘出用户频繁访问模式在网站的设计和维护、电子商务和教育等领域都有十分重要的实用价值。本文提出的 Close 算法从用户访问路径中抽取用户的访问模式，通过对算法的分析，以及由实际数据对算法的性能评估，可见算法是有效的，尤其是在最终挖掘出“最大频繁访问模式”时的优越之处，具有一定的使用价值和意义。

(下转第 19 页)