# 一种基于序列数的 Web 使用挖掘算法①

方　刚　(重庆三峡学院 数学与计算机科学学院 重庆 404000)

**摘　要：** 针对 Web 服务器日志中会话模式的页面属性为布尔量的特点，提出一种基于序列数的 Web 使用挖掘算法。该算法将用户会话模式转换成二进制数，然后用数字递增方式搜索候选频繁项；算法通过序列数的维来计算支持数，实现一次扫描用户会话模式，有效地提高了 Web 使用挖掘的效率。实验表明其效率比现有算法更快速而有效。

**关键词：** Web 使用挖掘；会话模式；Web 服务器日志；递增搜索；序列数

## Web Usage Mining Algorithm Based on Sequence Number

FANG Gang(College of Math and Computer Science, Chongqing Three Gorges University, Chongqing 404000, China)

**Abstract:** Aiming to the character that page attribute of session pattern in Web server log is Boolean quantity, an algorithm of Web usage mining based on sequence number is presented. The algorithm turns session pattern of users into binary, and then uses the way of number ascending to search candidate frequent itemsets. The algorithm computes support by sequence number dimension in order to scan once session pattern of users, and then the efficiency of Web usage mining is efficient improved. The experiment indicates that the efficiency is faster and more efficient than presented algorithms.

**Keywords:** Web usage mining; session pattern; Web server log; ascending search; sequence number

根据 Web 挖掘数据来源的不同，可将其分为三类：Web 内容挖掘、Web 结构挖掘和 Web 使用挖掘(也称用户访问模式挖掘) [1]。Web 内容挖掘是指对 Web 页面内容及后台交易数据库进行挖掘，从 Web 文档内容中获取有用知识的过程；Web 结构挖掘是对 Web 页面之间的结构进行挖掘，从 WWW 上的组织结构和链接关系中推导知识；然而若要想了解网络用户的浏览行为，研究 Web 使用挖掘就非常关键；用户访问模式可以从多个层次检测和挖掘到，针对某个用户在一次会话中产生的系列单击到跨越了很久的用户群浏览模式，长期以来收集的信息可以挖掘形成一个特征文件，其可表示当前用户的浏览行为，这些特性文件可用于指导网站管理和提供个性化服务。然而现有的 Web 使用挖掘算法效率并不高；于是针对 Web 服务器日志产生的会话模式数据具有布尔属性的特点，本文提出一种基于序列数的 Web 使用挖掘算法 WUMBSN(An algorithm of Web usage mining based on sequence number)，其有效地提高了 Web 挖掘的算法效率。

## 1　扩展Web服务器日志的相关知识

HTTP 点击流是指每当响应一条请求时，Web 服务器就会产生相应的日志记录[2]，其为 Web 数据库提供数据源；点击流通过分析能获取用户在访问网站时的浏览情况等信息。Web 服务器日志的原始标准是通用日志格式，其包括七个数据元素。在扩展的通用日志格式(ECLF)标准中，增加了两个额外元素。表 1 记录了 ECLF 的元素[3]。

为了记录更详尽的用户浏览行为，在 ECLF 中再添加域 time-taken，用其描述浏览页面所需的时间，

① 基金项目:重庆庆教委科技项目(KJ091108)
收稿时间:2010-04-10;收到修改稿时间:2010-05-22

以秒为单位。忽略页面加载时间，用 time-taken 描述用户访问某个页面时的逗留时间。

为了便于算法研究，在此只选取几个人们感兴趣的域进行数据处理，如表 2 所示。

表 1　ECLF 的域描述

| 域 | 描述 |
|---|---|
| remotehost | 远程主机域或 IP 地址 |
| t | 用户登录名 |
| user001 | 服务器授权用户名 |
| authuser | 响应相应请求的日期和时间 |
| date | 请求方法( Get 、Head、Post 等) |
| request | HTTP 状态码 |
| status | 传输字节数 |
| bytes | 用户此刻点击进入当前页面的 URL |
| referrer | 用户使用的操作系统和浏览器 |
| agent | |

表 2　数据结构表

| 扩展域 | 描述 |
|---|---|
| remotehost | 远程主机域或 IP 地址 |
| user001 | 用户登录名 |
| date | 响应相应请求的日期和时间 |
| status | HTTP 状态码 |
| referrer | 用户此刻点击进入当前页面的 URL |
| time-taken | 完成事务所需的时间,以秒为单位 |

## 2　基于序列数的Web使用挖掘算法

### 2.1 预处理挖掘数据

对 Web 服务器日志进行数据预处理[4]，设用户在每个页面的停留时间为 t，设定时间阈值 $t_1$ 和 $t_2$，如果 $t \in [t_1, t_2]$，则表示用户对当前页面有兴趣，否则就没有兴趣。用数字表示 remotehost(r-host)，用字母表示 referrer，设会话中访问的页面是{a，b，c，d，e，f，g}。针对每个页面来说，用户对其或者感兴趣或者无兴趣，其属于布尔型，故用户的会话模式(即用户的访问页面)可用二进制形式表示，最后得到挖掘所使用的数据，其转换过程如表 3 所示。

表 3　日志数据预处理过程

| r-host | referrer | *a* | *b* | *c* | *d* | *e* | *f* | *g* |
|---|---|---|---|---|---|---|---|---|
| 1 | *a*, *b*, *c*, *d*, *g* | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 2 | *a*, *b*, *d*, *f* | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 3 | *b*, *c*, *f*, *g* | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 4 | *b*, *d*, *e*, *g* | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 5 | *a*, *c*, *d*, *e* | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 分段转换获得子项 1 | | 25 | 30 | 21 | 27 | 3 | 12 | 22 |
| 6 | *b*, *d*, *e*, *f* | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 7 | *a*, *b*, *c*, *d*, *e*, *f* | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 8 | *a*, *b*, *c*, *f*, *g* | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 9 | *b*, *c*, *e*, *f*, *g* | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 10 | *b*, *c*, *d*, *f*, *g* | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 分段转换获得子项 2 | | 12 | 31 | 15 | 25 | 26 | 31 | 7 |
| 序列数 | *a*-{25, 12}, *b*-{30, 31}, *c*-{21, 15}, *d*-{27, 25} *e*-{3, 26}, *f*-{12, 31}, *g*-{22, 7} | | | | | | | |

### 2.2 相关定义及性质

定义 1. 序列数(Sequence Number)，记为 SN，是一组有序数。序列数中的每个整数称为子项。

例，若序列数 SN={33，19，39，46，75，62，31}，75 是其中的一个子项。

定义 2. 数字维(Number Dimension)，记为 ND，是个整数；它等于其二进制数中含“1”的个数。

定义 3. 序列数的维(Sequence Number Dimension)，记为 SND，是个整数；其值为子项数字维的总和。

举例，若SN={24,39,45}，则SND(SN)=SND(24) + SND(39)+SND(45)=2+4+4=10。

性质 1 二进制 a、b 对应的会话模式分别为 $T_a$ 和 $T_b$，$T_a \subseteq T_b$ 的充要条件是 $a \wedge b = a$。

性质 2 二进制 a、b 对应的会话模式分别为 $T_a$ 和 $T_b$，若 $a \wedge b = a$：

① 若 $T_b$ 为频繁项，则其子集 Ta 也为频繁项。

② 若 $T_a$ 为非频繁项，则其超集 Tb 为非频繁项。

### 2.3 算法产生候选项的方式及修剪策略

产生会话模式候选项的方法：通过数字的自动递增来控制会话页面的不同组合方式，即产生候选频繁会话模式。

具体实现过程：首先根据会话页面总数获得一个整数区间，这个区间的最大值为 $2^m-1$(m 为会话页面的总数)，最小值为 1；然后从这个整数区间的最低点开始，按数字递增方式产生一个数值，其二进制形式对应着一个会话页面的组合，即这个页面组合可以成为候选频繁会话模式；最后产生完区间所有数字，也就产生了访问页面的

所有组合。

用性质 2 作为修剪策略，即产生数值不为递增过程中产生非频繁项的超集，才可以作为候选项；当产生一个频繁项时，需在已有频繁项中要删除其子集。

例，设有 4 个会话页面{a, b, c, d}，则得到数字区间[1，15]，max=15，min=1，其产生候选频繁项的过程为：

$C_1$=1，其对应的会话页面{d}；

$C_2$=2，其对应的会话页面{c, d}；

…

$C_{15}$=15，其对应的会话页面{a, b, c, d }；

### 2.4 计算支持数的方式

根据 2.1 节的知识和定义 1 可以求出会话模式中每个页面对应的序列数，然后通过计算其维可得会话模式候选项的支持数，其实现步骤如下：

(1) 将所有的用户会话模式按 2.1 节的方法转换成二进制形式，如表 3 的第 3 列所示。

(2) 将会话页面的二进制位所在列，按段形成二进制数，并形成整数，由定义 1 得到对应的页面序列数，如表 3 所示。

(3) 针对会话模式候选项，用其包含的页面序列数，进行“与”位运算，得到对应的候选项序列数，计算此序列数的维，其值为候选项的支持数。

例，如表 3 所示的 10 次会话模式，若一个候选数字为 56，其包含{b, c, d}；则计算这三个页面序列数的“与”值，计算其维数就得到该项的支持数为 3：

SN {b, c, d} = SN {b} & SN {c} & SN {d}

={30&21&27，31&15&25}={16，9}；

则 SND(SN{b, c, d})=3。

### 2.5 算法的挖掘步骤

设 Web 服务器日志预处理的数据中，会话页面数为 m，总的会话次数为 N，定义如下符号：

D：存放会话模式的二进制形式；

NF：存放递增产生的非频繁项；

F：存放递增产生的频繁项；

Step1：根据 2.1 节将 Web 服务器日志的会话模式的转换成二进制形式，并存入 D 中。

Step2：根据 D 中的二进制，计算出所有会话对应的页面序列数，记为 $SN_i$。

Step3：计算产生候选项的数字区间，即最大值为 2m-1(m 为会话页面数)，最小值为 1。

Step4：从数字区间的最低端开始，按递增方式产生数字 Ca，若其不为 NF 的超集，就计算其支持数，若为频繁项就存入 F 中，并删除其在 F 中的子集；否则存入 NF 中；最后递增 Ca，直到搜索完区间的所有数字。

Step5：由置信度从 F 中的频繁项，产生标准关联规则。

### 2.6 产生频繁会话模式的挖掘算法

设整数区间为[1，max]，Data [m, n](m 为用户会话的总次数，n 为会话页面数)为按 2.1 节方法转换得到的挖掘数据库，F 存放频繁会话模式的二进制的整数值。NF 存放递增产生的非频繁用户会话模式的二进制的整数值。算法如下：

```
(1) For (candidate=1; candidate ≤ max; candidate++) {
(2) If (NFcandidate) {
(3) If (Support (candidate) ≥ minsupport)
(4) Write candidate to F and delete its subset;
(5) Else
(6) Write candidate to NF ;}
(7) }
Support (int Candidate)
(1) int i = 0, j = 0, Support = 0, exit = 0;
(2) int [] Set = new int[Line];
(3) While ((i < Row) && (Exit == 0)) {
(4) IF ((Candidate & 1) == 1) {
(5) Exit = 1;
(6) For (j = 0; j < Line; j++) {
(7) Set[j] = Data [j, i] ;}
(8) }
(9) ELSE {
(10) Candidate = Candidate >> 1;
(11) i++;}
(12) }
(13) j = i;
(14) While ((j < Row) && (Candidate! = 0)) {
```

```
(15) IF ((Candidate & 1) == 1) {
(16) For (i = 0; i < Line; i++) {
(17) Set[i] = Set[i] & Data [i, j] ;}
(18) }
(19) Candidate = Candidate >> 1;
(20) j++;}
(21) For (i = 0; i < Line; i++) {
(22) Support = Support + SND (Set[i]) ;}
(23) Return Support;
```

## 3 算法的性能分析及实验比较

### 3.1 性能分析

(1) 时间复杂度分析：设总的会话次数为 N 个($N \leqslant 2^m$)，m 为会话页面总数，设算法每次进行“位”运算的计算量为 P，则计算序列数的维的量为 N×P，针对一个包含 t 个会话页面的频繁项，计算支持数的量为 (t+1) × N × P，则算法的时间复杂度记为：

$$[\sum_{t=1}^{m} C_m^t (t+1)] \times N \times P$$

(2) 空间复杂度分析：存储空间的占用量与算法和数据结构紧密相关，为了提高空间使用率，算法通过二进制的位运算，其空间复杂度为 $O(a \times N \times m)$，m 为会话页面总数，$a$ 是一个与支持度相关的参数。

### 3.2 实验比较

Web 使用挖掘算法研究的主流仍然是将传统的挖掘算法用于 Web 使用挖掘中，如 B_Apriori[5]和 B_ARDSM[6]；这些算法虽然通过二进制的位运算，提高了相应算法的效率，但这些算法每次计算支持数时均需扫描一次数据库，而提出的算法 WUMBSN 在 Web 使用挖掘过程中仅需扫描一次数据库，提高算法效率；为了体现提出算法的优越性，在此用算法 B_Apriori[5]和 B_ARDSM[6]进行比较。

B_Apriori：是在 Apriori 算法的基础上，引入二进制形式的候选项，运用了二进制的位运算，实现快速产生候选项和计算支持数，使 B_Apriori 算法比 Apriori 效率高，其只适合挖掘短频繁项集。

B_ARDSM：是在双向挖掘算法 IDMFIA[7]的基础上，引入二进制形式的候选项，其也运用二进制的位运算来提高效率；其虽适合挖掘任何长度的频繁项，但仍是多次重复扫描数据库，制约着算法的效率提高。

现模拟挖掘数据，其用户会话模式有 4095 次，转换成二进制后，其值是 1 到 4095，会话页面总数为 m=12。

实验环境为：Intel(R) Celeron(R) M CPU 420@1.60 GHz，1.24G 的内存，操作系统为 Windows XP Professional，在 Visual C# 2005.NET 开发平台上实现算法 B_Apriori、B_ARDSM 和 WUMBSN。

B_Apriori 和 WUMBSN 算法的实验结果如图 1，B_ARDSM 和 WUMBSN 算法的实验结果如图 2。运行时间随支持度和长度变化的比较如图 3 和 4 所示。从比较结果可知，WUMBSN 算法的挖掘效率比现有算法要高效得多。


图 1 B_Apriori 和 WUMBSN 的实验结果


图 2 B_ARDSM 和 WUMBSN 的实验结果

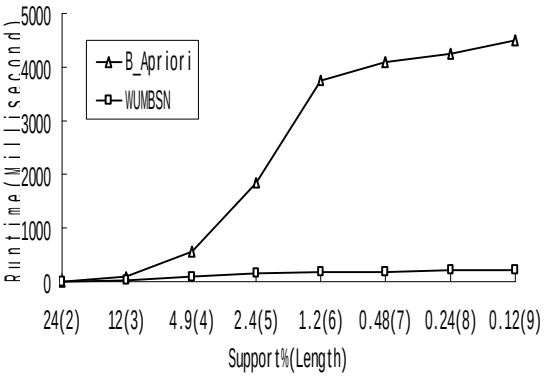


图 3　B_Apriori 和 WUMBSN 运行时间比较

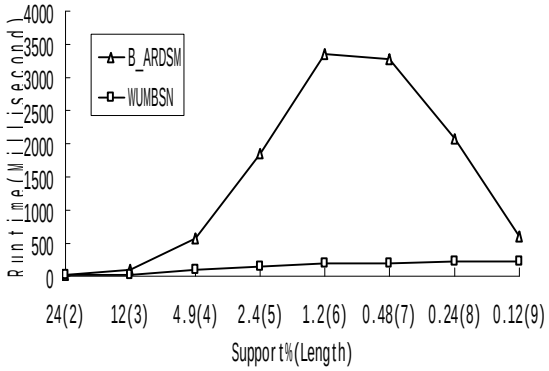


图 4　B_ARDSM 和 WUMBSN 运行时间比较

## 4　结语

提出一种基于序列数的 Web 使用挖掘算法，其将会话模式转换为二进制，算法通过数字递增的方法产生候选频繁项，通过序列数的维来计算支持数，实现一次扫描数据库。实验表明其效率比现有算法更快速而有效。

## 参考文献

1 张波,巫莉莉,周敏. 基于 Web 使用挖掘的用户行为分析.计算机科学,2006,33(8):213－215.

2 Kimball R, Merz R. Web 数据仓库构建指南.北京:清华大学出版社, 2005.

3 Sweiger M，Madsen MR，Langston J，Lombard H. 点击流数据仓库.北京:电子工业出版社, 2004.

4 Fang G, Wang JL, Ying H, Xiong J. A double algorithm of Web usage mining based on sequence number. Wenbin Hu eds. Proc. International Conference on Information Engineering and Computer Science, New York: IEEE Press, 2009:1817－1820.

5 陈耿,朱玉全,杨鹤标,关联规则挖掘中若干关键技术的研究.计算机研究与发展，2005,42(10):1785－1789.

6 Fang G, Wei ZK, Yin Q. An Algorithm of Association Rules Double Search Mining Based on Binary. Eduard Hovy eds. Proc. of 7th International Conference on Machine Learning and Cybernetics, New York: IEEE Press, 2008:184－189.

7 吉根林,杨明,宋余庆,孙志挥,最大频繁项目集的快速更新.计算机学报, 2005,28(1):128－135.