第 33 卷 第 1 期 计 算 机 工 程 2007 年 1 月
Vol.33 No.1 Computer Engineering January 2007
·软件技术与数据库· 文章编号：1000—3428(2007)01—0095—03 文献标识码：A 中图分类号：TP311

一种 Web 日志会话识别的优化方法

陈子军，王鑫昱，李 伟
(燕山大学信息学院计算机科学与工程系，秦皇岛 066004)

摘 要：会话识别是 Web 日志挖掘的关键步骤，然而很多方法所得到的会话不够精确。该文对此提出优化算法，并对最常用的 Timeout 方法识别的会话进行优化，通过实验证明会话质量得到了提高。
关键词：Web 日志挖掘；数据预处理；会话识别

Method of Web Log Sessions Reconstruction

CHEN Zijun, WANG Xinyu, LI Wei
(Department of Computer Science and Engineering, Information Institute, Yanshan University, Qinhuangdao 066004)

【Abstract】 Although sessions reconstruction is an important step in Web log mining, the sessions recognized by existing methods are not accurate. An algorithm resolving this problem is proposed. This paper improves the Timeout method with the algorithm. Finally the algorithm enhancing the quality of Web log sessions is proved by experiments.
【Key words】 Web log mining; Data preprocessing; Sessions reconstruction

1 概述

Web 日志挖掘的目的是为了发现用户的访问模式和兴趣，这些信息可以用于及时调整网站的结构和内容，从而创造更大的商业价值。

Web 日志挖掘的主要步骤有数据预处理、模式识别和模式分析，其中数据预处理是关键和首要任务。如果没有良好的预处理算法，就会降低进一步挖掘的效率和准确性。Web 日志挖掘的预处理包括数据清洗、用户识别、会话识别和路径补全等步骤。

本文主要研究会话识别的方法。目前，有两种用户访问会话的定义：

(1)一个从用户进入站点时刻起至他离开时刻止所请求的一系列链接的集合[1]；

(2)一个用户在站点中关于某一个话题所请求的一系列链接的集合[1,2]。

第 2 种定义是将第 1 种定义按照话题进行细化。为了发现用户的访问模式和兴趣，本文采用第 2 种定义。

目前，会话识别的常用方法有 3 种。最常用的是Timeout方法。在该方法中，会话的边界是在两个页面请求记录(以下简称记录)的时间间隔超过一定的阈值时确定的。经过实验和理论分析，选择 25.5min作为Timeout的时间阈值比较接近真实会话[3]。目前产业界一般选择 30min作为默认的Timeout时间阈值。

Cooley等人提出了一种事务识别会话的方法，称为序列长度法[2]。经过研究发现，用户浏览页面的模式一般是通过辅助页面到达内容页面，而且用户在内容页面停留的时间往往要比辅助页面的长。这样，如果已知内容和辅助页面的集合，在顺序读取日志记录时，一旦遇到内容页面就找到了会话的边界。但是由于在多数情况下，用户光顾一个站点不可能只对一个内容页面感兴趣，因此该方法形成的会话必然与真实会话有一定的差距。

最大向前序列的方法是根据用户访问行为划分会话，一旦用户后退浏览已经浏览过的页面时，就找到了会话的边界[4]。这种方法简便，也易于实现，但是由于所发现的会话只表达了用户的部分行为，具有很大的局限性，因此也不是很适用。

以上 3 种会话识别方法可能产生如下情况：(1)使原本在同一个会话里的记录被划分在不同的会话中；(2)使原本不在同一个会话的记录被划分在同一个会话中。当这样的不合理划分所占的比例很大时，进一步挖掘和分析所得到的结果就会受到影响，甚至是错误的。基于此，本文提出了会话识别的优化算法，并对最常用的 Timeout 方法所得到的会话进行优化，通过实验证明会话质量得到了提高。

2 基本思想

下面介绍会话识别优化方法的两种基本操作：合并和断开。

2.1 合并

会话识别可能使原本在一个会话里的记录被划分到两个不同会话里。例如，真实会话<X, ..., S, R, A, B, ... , Y>(其中 X, S, R, A, B, Y 都是记录)被划分成<X, ..., S, R, A>和<B,...>两个会话。

由于 A 和 B 在一个真实会话里，表明用户还没有转向另一个话题，或者说用户还没有离开站点。简单地说，就是在网站的拓扑结构中，从 A 到 B 有直接或者间接的链接。

基于上述事实，在对会话进行优化时，如果遇到会话边

基金项目：燕山大学博士基金资助项目
作者简介：陈子军(1971－)，男，博士、副教授，主研方向：数据挖掘，数据集成与交换；王鑫昱、李 伟，硕士生
收稿日期：2006-01-05 **E-mail：**zjchen@ysu.edu.cn

界 A 和 B(形式为<..., S, R, A>、<B,...>)，则分两种情况考虑将 A、B 合并到一个会话中。

情况 1：如果模式 A→B 是该用户经常访问的模式(即频繁访问模式)，则直接将 A、B 合并，否则进行情况 2 的判断。频繁访问模式的挖掘很受关注，当前时间复杂度比较好的算法是文献[5]中提出的用 suffix tree 挖掘最大频繁序列 MFS，时间复杂度是 O(nlogn)。本文利用该挖掘方法对由会话识别形成的会话进行处理，形成 MFS 文件，从而判断两个相邻会话的边界是否为频繁访问模式。

情况 2：利用网站的拓扑结构判断 A 或 A 之前的 L 条记录能否链接到 B(称为能否寻迹到 B，在算法中用函数 Trace (A→B, L)表示)，如果找到上述记录，则将 A、B 合并，否则默认原来的划分。至于 L 的取值本文将在实验分析部分给出说明。

2.2 断开

会话识别也可能使原本不在同一个会话的请求被划分在同一个会话中。例如，有两个真实会话<X, ..., A>、<B,...,Y>(其中 X,A,B,Y 都是记录) 被划分成<X, ..., A, B,...>一个会话。

A、B 虽然是一个用户在服务器上记录的两条连续记录，但是由于不在同一个真实会话里，表明用户已经转向另一话题。该用户可能是通过一定量的后退，也可能是直接输入网址等方法到达记录 B 的页面。

基于上述事实，在优化时，对于会话内部记录 A 和 B(形式为<..., A, B,...>)，如果通过 A 或者 A 前面的 L 条记录不可以寻迹到 B，则将 A、B 断开。

另外，如果用户由一个话题转向其它话题，一般要经过较长的时间段α，即 A 到 B 经历的时间是α。因此不必对会话内部的每两个相邻记录都进行判断，只需要特别关注那些时间间隔相对较大的记录。本文α的取值为当前会话的平均访问时间，随着记录的读入动态计算。

综上所述，通过合并和断开两种操作对会话进行优化，必然能够使所得到的会话更接近真实会话。

3 数据结构和算法

下面给出会话识别的优化算法。如果相邻的两个会话形式为<...,A>、<B,...>，这时就要判断 A、B 是否需要合并在同一个会话中。如果满足下面两个条件之一即可合并：(1)模式 A→B 是频繁的，也就是说该模式在 MFS 中可以找到；(2)可以通过记录 A 或者 A 前面的记录寻迹到记录 B。

如果一个会话形式为<..., A, B,...>，这时就要判断 A、B 是否需要划分在两个不同的会话里。如果同时满足下面两个条件即可断开：(1)记录 A、B 的访问时间差大于动态时间阈值α；(2)不可以通过记录 A 或者 A 前面的记录寻迹到 B。

3.1 数据结构

本文采用的数据结构如图 1 所示。各个域的含义如下：url 是页面请求记录的 URL 链接；accesstime 是访问时间；boundary 是整型域，表示该记录在会话中的位置信息，是会话的头则值为 1，是会话的尾则值为 2，否则值为 0。

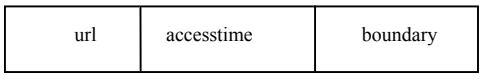

图 1 数据结构

3.2 算法

下面给出本文的算法，对同一个用户的访问会话进行优化，描述如下。

Algorithm reSessionizer

输入：会话识别形成的会话文件 sessions_file、网站拓扑结构文件和最大频繁访问序列文件 MFS

输出：优化后的会话文件

```
begin
(1)打开所有输入文件;
(2) readItem(sessions_file, R1);
(3) α＝0;
(4) while   (sessions_file 不为空)
(5)     readItem(sessions_file, R2);
(6)     if (α = = 0)   //判断是否开始一个新的会话
(7)         α = R2. accesstime – R1. accesstime;
(8)     if (R1.boundary = =2 and R2.boundary = = 1)
(9)       if (R1.url->R2.url in MFS) or (Trace(R1.url →R2. url, L))
(10)            R1.boundary = R2.boundary = 0;//合并
(11)       else     α＝0; //开始一个新的会话
(12)     else if ( R2.accesstime – R1.accesstime >= α)
and (!Trace (R1.url→R2.url,L)
(13)            R1.boundary=2; R2.boundary=1; //断开
(14)            α＝0;
(15)     if (α!=0)
(16)         α = (R2.accesstime – R1.accesstime+α) /2;
(17)     R1 = R2;   //end while
(18) 关闭所有文件
end.
```

算法首先打开会话识别形成的会话文件(sessions_file)、网站拓扑结构文件和挖掘得到的最大频繁访问序列文件(MFS 文件)，读取第 1 条记录，此时设置会话内部的平均访问时间α为 0。第 5 行顺序读取该用户会话的其它记录。第 6、7 行是在开始一个新的会话时设置平均访问时间α的初值。第 8-11 行进行合并操作，第 8 行判断目前的两条记录是否是两个相邻会话的边界，第 9 行判断它们是否符合合并条件，第 10 行更改 boundary 域值使记录 R1、R2 合并，第 11 行不符合合并条件，将平均访问时间α置为 0，表明将要开始优化新的会话。第 12-14 行是进行断开操作，第 12 行判断目前的两条记录 R1、R2 是否符合断开条件，第 13 行更改 boundary 域值使 R1、R2 断开，则第 14 行将平均访问时间α置为 0。第 15 行判断α的值，第 16 行α的值不为 0，表明没有开始新的会话，仍需要动态计算当前会话的平均访问时间α。第 17 行将记录 R2 赋值给 R1，返回到第 4 行，形成循环。

下面解释算法中涉及的各个函数的功能：

readItem(sessions_file, R)从会话文件 sessions_file 中读取记录，将其存储到 R 中。

Trace(R1.url→R2.url, L)在网站的拓扑结构中，寻找 R1 到 R2 的链接，如果没有找到，再去寻找 R1 前面的第 1 条记录到 R2 的链接，依此类推，一直寻找到 R1 前面的第 L-1(L 是大于等于 1 的整数)条记录。如果找到链接，则函数返回值为真，否则返回值为假。

4 实验分析

本节将给出会话识别优化算法的实验结果并加以分析。实验数据采用最常用的 Timeout 方法所得到的会话。日志文件来源于燕山大学信息科学与工程学院，记录的是 2005 年 6 月份访问该部门网站的信息。由于要人工识别真实会话，数

据量和工作量都很大，因此只选取一天的记录进行分析。根据分析判断，6 月 25 日的日志比较接近整个月份的平均访问水平线，具有典型性，因此本文对该天的日志进行分析。

经过人工识别，6 月 25 日一天内所产生的实际真实会话有 1 191 个。Timeout 方法发现的会话有 854 个，其中发现真实会话有 763，占实际真实会话的比例是 64.064%。

算法 reSessionizer 发现会话的结果将在下表中给出。根据算法中函数 Trace (R1. url→ R2. url, L)参数 L 的值的不同，发现会话的结果也有所不同。

表 1 中，M 表示实际真实会话的个数(M=1 191)，A 表示发现的会话个数，B 表示发现的真实会话个数。

表 1 reSessionizer 发现会话结果

	A	B	B/A	B/M
Timeout	854	763	89.344%	64.064%
reSessionizer, L＝1	975	873	89.538%	73.300%
reSessionizer, L＝2	866	782	90.300%	65.604%
reSessionizer, L＝3	839	760	90.584%	63.812%

从上表的实验数据可以看出，当 L 取值为 1 时，算法 reSessionizer 识别出的真实会话数目最多，此时断开的会话比较多，得到的会话数目增加较大；当 L 取 2 时断开的会话数目较少，会话数目增加不大；当 L 取 3 时，发现的真实会话占发现会话比例最大，但是此时算法合并的会话多而断开的会话少，造成会话的绝对数目最少。可见 L 取值太大反而会影响效果。参数 L 可以根据经验数据(如网站的拓扑结构和繁忙程度等)和需要适当选取。由实验数据可以看出，只要选取合理的参数，算法 reSessionizer 就能够有效地提高会话识别的质量。

5 总结

本文提出的 Web 日志会话识别的优化方法，通过合并和断开两种操作，使得所发现的会话更加接近真实会话。会话识别的优化方法，是在会话识别方法的基础上线性扫描会话文件，提高了会话的质量。

今后的工作可以向两个方向努力，首先，将会话的定义标准化，提出真正有利于进一步挖掘的会话定义，对于研究高性能和高效率的算法至关重要。其次，在预处理时更多地利用已知的数据(如拓扑结构等)和经验数据(如 MFS 等)。通过这些努力提高挖掘的效率和质量，从而获得更大的商业效益。

参考文献

1 Facca F M, Lanzi P L. Mining Interesting Knowledge From Weblogs: a Survey[J]. Data and Knowledge Engineering, 2005, 53(3): 225-241.

2 Cooley R, Mobasher B, Srivastava J. Data Preparation for Mining World Wide Web Browsing Patterns[J]. Journal of Knowledge and Information Systems, 1999, 1(1): 5-32.

3 Catledge L, Pitkow J. Characterizing Browsing Strategies in the World_Wide_Web[J]. Computer Networks and ISDN Systems, 1995, 27(6): 1065-1073.

4 Chen M S, Park J S, Yu P S. Efficient Data Mining for Path Traversal Patterns[J]. IEEE Transactions on Knowledge and Data Engineering, 1998, 10(2): 209-221.

5 Xiao Yongqiao, Dunham M H. Efficient Mining of Traversal Patterns[J]. Data and Knowledge Engineering, 2001, 39(2): 191-214.

（上接第 94 页）

可以看出这个进程会产生死锁，从 MWB 显示的死锁路径看到当进程从通道 1 接收消息来执行匹配进程时就会产生死锁。反馈到 BPEL4WS 中，reply 和 receive 活动之间有一个 l 链接，限制了 reply 活动必须等到 receive 执行完后才能执行，但是在结构化活动 sequence 中必须先执行 reply 活动才能执行 receive 活动激活出站链接，这样就形成了一个死锁。

上面列举的只是 BPEL4WS 的一个简单片断，对于简单的流程可能不能充分体现这种方法的优势，但是当 BPEL4WS 文件结构复杂的时候，流程有无死锁就不是我们能够简单判断的了，通过映射到π-演算然后再利用 MWB 进行验证将会是一个很有效的方法。而且根据上面的算法编程生成能够直接用于 MWB 的π-演算表达式，再利用 MWB 进行死锁、互模拟等性质的验证，这样从 BPEL4WS 的建模到验证就成为一个自动化的过程。为其在正式被实施前得到形式上的模拟与验证提供了简便可行的方法。

3 总结

本文描述了BPEL4WS到π-演算的自动映射方法，并把映射结果利用到MWB进行死锁验证。在研究的过程中发现一些问题：pi演算中没有关于进程的顺序表达式，因此需要用并发来表示顺序。但是模型检测是基于对系统状态空间的穷举搜索，对于并发系统，其状态的数目往往随并发分量的增加呈指数增长。因此当一个系统的并发分量较多时，直接对状态空间进行搜索在实际上是不可行的，这就是状态爆炸问题[5]。因此通过并发来表示顺序并不是一个很好的办法，在以后的工作中如何解决这个问题，基于BPEL4WS是一门特殊的语言，能否针对BPEL4WS对pi演算进行适当的扩充，加上顺序表达式，值得进一步研究和探索。

参考文献

1 Koshkina M, Breugel F. Verification of Business Processes for Web Services[R]. Department of Computer Science, York University, Technical Report: CS-2003-11, 2003-10.

2 Victor B. The Mobility Workbench User's Guide Polyadic(Version 3.122)[Z]. 1995-10. http://www.it.uu.se/profundis/mwb-dist/guide-3.122. ps.gz

3 杜旭涛. π-演算交互式验证工具的研究与实现[D]. 国防科技技术大学研究生院, 2002-01.

4 傅建明, 韩光鹏, 朱福喜. 两种死锁分析的逻辑方法[J]. 武汉大学学报(自然科学版), 1999, 45(3).

5 林惠民, 张文辉. 模型检测: 理论、方法与应用[J]. 电子学报, 2002, 30 (12A).