点击流中事务数据模型的设计与实现*)

辛 燕 鞠时光 蔡 涛 阎星娥
(江苏大学 镇江212013)

Design and Implement of Transaction Data Model in Clickstream

XIN Yan JU Shi-Guang CAI Tao YAN Xing-E
(Jiangsu University, Zhenjiang 212013)

Abstract In this paper, we first briefly introduce the concepts of clickstream data and data warehouse, analyze two existing clickstream star schema——click star schema and session star schema in webhouse, then induce a new model——transaction star model based on them, and expressed the method of bringing out the model. Comparing with the two schemas mentioned above, its most apparent speciality is that it includes a series of meaningful page-view sequence rather than a single click. Thus, on the one hand it improves the query performance of data, on the other hand it is in favor of executing more deepen analysis——data mining, and simplifies the process of data pretreatment. At last, the paper verifies its' feasibility and validity using association rules based on the model.

Keywords Clickstreams, Data warehouse, Star schema, Transaction fact model, Association rules

点击流数据简单说就是Web服务器上一系列有序的日志记录。随着WWW应用及电子商务的高速发展，电子商务网站的Web服务器上自动收集了大量的用户访问信息记录，即所谓的Web日志。Web日志蕴涵了大量的有用信息，如客户来源、客户访问趋势、客户兴趣、网站流量等，因而记录和分析Web日志数据已逐渐成为e企业的一项重大活动。点击流数据仓库对原始的Web日志数据进行过滤、清洗并集成，以便于利用联机分析处理和数据挖掘技术对点击流数据做进一步分析，从而为企业创造巨大的信息财富。

本文主要针对点击流数据分析中的一次事务访问过程，提出对点击流数据进行事务事实建模，即为单个的事务访问建立数据分析模型。文章在分析Kimball提出的两种星型模式——点击星型模式和会话星型模式的基础上[1]，设计了事务星型模式，并给出了实现方法，最后通过关联规则挖掘在该模型上的应用来说明其在点击流数据分析中的可行性及有效性。

1 点击流数据与点击流数据仓库

顾客从进入一个电子商务站点，到离开这个站点的一个访问周期中，所浏览的页面、滞留时间、点击的链接和广告都会被顺序地记录在网站的日志文件中。这种有序的Web日志记录形成了所谓的点击流数据。而在Web服务器日志文件中，无论是普通日志格式还是扩展日志格式，都包括客户端IP、用户名/用户ID、请求时间、请求方式、请求的目标文件名、状态、参引页、代理、cookie等基本的数据域。但是在点击流数据分析中，并非每一个数据域都是有用的数据域，如请求方式、状态等数据域对分析过程并没有多大作用，因而具体分析时可将无关的数据域剔除掉。

为了对点击流数据进行分析，通常需为原始点击流数据建立数据仓库。这是因为：①点击流数据来源于Web服务器上原始的日志记录，对于一个较大的电子商务网站来说，每天将产生数以百万条计的日志记录，数据仓库本质上来说就是一种大型的数据库，因而能有效地组织和管理大量的点击流数据；②原始的点击流数据中含有一些无用的数据成分，因此在分析点击流数据之前，需要对其进行适当的预处理，数据仓库中的数据一般是经过抽取、转换、清洗与集成了的合成数据，因而在数据仓库上进行点击流数据分析可免去好多数据预处理的工作；③数据仓库中集成了大量的历史数据，而点击流数据分析中好多也都与时间有关，如点击序列模式分析等。因而，建立面向点击流的数据仓库，已成为点击流数据分析中

*)本课题的研究得到国家自然科学基金(60173064)及江苏省自然科学基金(NO. BK200204)的资助。

5 Pagurek B, Wang U, White T. Integration of mobile angents with SNMP: Why and how. Submiited to NOMS'2000, 2000
6 Martin-Flatin J-P, Bovert L, Hubaux J-P. JAMAP: a web-based management platform for IP networks. In: Proc. of the 10th IFIP/IEEE Intl. Workshop on Distributed System: Operations and Management(DSOM'99)
7 Hwang K-C, Hong J-J, Lee, K-H. A SNMP Group Polling for the Management Traffic. TENCON 99. In: Proc. of the IEEE Region 10 Conf. 1999, 2: 797～800
8 Case J, Fedor M, Schoffstall M, Davin J. A Simple Network Management Protocol (SNMP). RFC1157, 1990
9 Stallings W. SNMP, SNMPv2, SNMPv3, and RMON 1 and 2(3rd edition). Addison-Wesley Pub Co, 1998
10 Chelkhrouhou M, Labetoulle J. An Efficient Polling Layer for SNMP. Network Operations and Management Symposium, 2000. NOMS 2000. 2000 IEEE/IFIP, 2000. 477～490
11 Jacquenet C, Proust C. An introduction to IP multicast traffic engineering Universal Multiservice Networks, 2002. EDUMN 2002. 2nd European Conference on, 2002. 263～267
12 CISCO. CGMP Isn't Preventing the Flooding of Multicast Packets for Certain Group Addresses. http://www.cisco.com/warp/public/105/mcastguide9.html
13 白彩英,等. 计算机网络管理系统设计与应用. 北京：清华大学出版社，1998
14 Sprenkels R, Martin-Flatin J-P. Bulk transfer of MIB data. The Simple Times, The Quarterly Newsletter of SNMP Technology, Comment, and Events, 1999, 7(1): 1～8

的一个基本而重要的环节。

在数据仓库中，最流行的数据模型是多维数据模型[2]，该模型将数据看成数据立方体(data cube)的形式，允许以多维对数据建模和观察。多维数据模型通常是由包含主题的事实表和多个包含事实的非正规化的维度表所组成[3]。事实表中包括事实的名称或度量(事实是数值度量的)，以及每个相关维表的关键字；维度表由一组属性组成，每一个维度表通过一个关键字(维度关键字)直接与事实表关联。多维数据模型可以以星型模式、雪花模式或事实星座模式存在[2]。对于点击流数据仓库，常用的设计模式为星型模式，该模式包含一个事实表和一组相应的维度表，维度表围绕事实中心表显示在射线上，看上去像星星爆发，所以称之为星型模式图。使用星型模式构建数据仓库的主要优点是查询分析的执行效率比较高。

点击流数据仓库中，已有的两种星型模式为点击星型模式和会话星型模式[1]。

点击星型模式是对一次点击事实的建模，它由点击事实表和相应的维度表组成。事实表记录了站点上一次独立的点击，它包括每个相关维度表的关键字：URL_key、Session_key、TimeOfDay_key、Date_key，以及自身的度量值：Number_in_session、Click_seconds。Number_in_session 表明该点击在特定用户会话中的序列号，Click_seconds 说明此次点击的浏览时间。点击事实表有4个与之对应的维表：URL 维表、Session 维表、Date 维表、TimeOfDay 维表，每个维表都有自己的属性集。URL 维表跟踪此次点击该用户所请求的页面，Session 维表跟踪此次点击所属的会话，TimeOfDay 维表和 Date 维表分别跟踪此次点击发生的时间和日期。点击星型模式图如图1所示。

应用点击星型模式的主要优点是，点击事实表中的数据如实地反映了 Web 日志记录，它与日志记录处于相同的粒度，聚集时不会有信息的丢失。在此基础上我们可对每一次独立的点击数据进行分析。该模式的主要花费是，在分析用户会话时，为获得所需的点击序列，需要在单个点击之间执行复杂的多重自连接查询，而这种基于庞大的事实表的多重自连接查询往往具有较差的查询性能。

会话星型模式可看作是对点击星型模式的补充，其建模对象是一个完整的用户会话。会话星型模式中，事实表主要由相关维表的关键字：Date_key、TimeOfDay_key、User_key 所组成；维表主要是 Date 维表、TimeOfDay 维表、User 维表，Start_page 维表、End_page 维表，分别用来跟踪会话发生的日期、时间、会话的用户、会话的起始页以及会话终止页。

会话星型模式的最大优点就是容易跟踪网站的浏览者。然而，由于在建立会话星型模式的转换过程中丢失了有关单次点击的信息，因此在该模式上我们并不能得到此次会话过程中一次独立的点击访问信息。

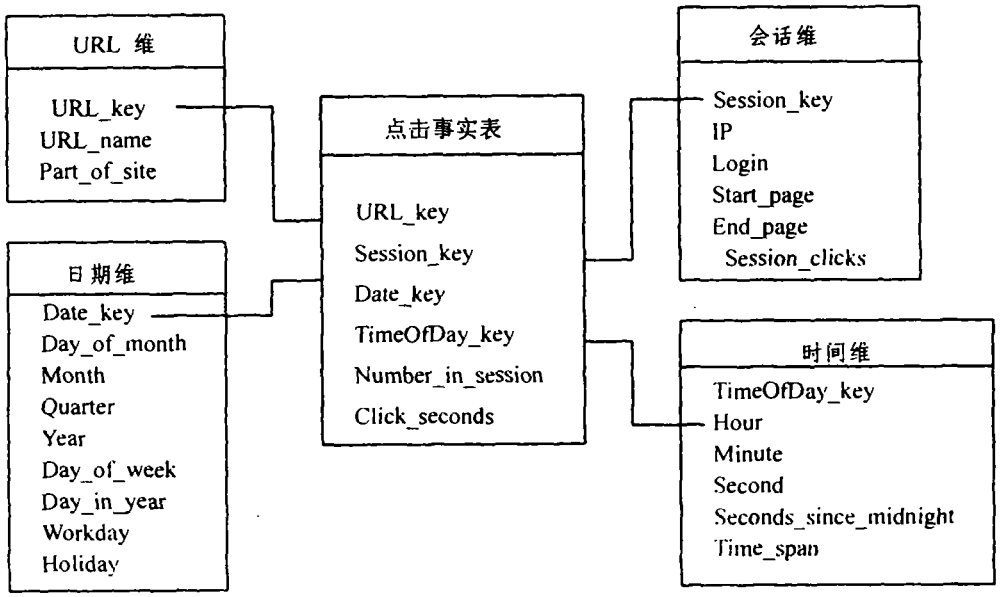

图1 点击星型模式图

从以上分析可以看出，已有的点击星型模式和会话星型模式对于分析特定会话中的一次事务访问的点击序列存在着一定的局限性，因而本文在此基础上提出建立面向事务分析的事务星型模式的思想。

2 事务星型模式设计

在分析点击流数据分析时，对特定会话中一次事务访问的点击序列进行分析，有助于理解不同用户的访问习惯和偏好，了解站点上各页面的一般访问路径、频繁遍历路径以及页面之间的关联性。

点击流数据仓库中，事务数据采用星型模式对其进行描述和表示。在点击流数据分析中，事务是指一次用户会话中页面访问序列的子序列。我们可用图2来简单表示点击、会话、事务三者之间的关系。图中，A、B、F、O、G 分别代表单次点击所请求的页面，A－B－F－O－F－B－G 表示一次完整会话过程中的页面请求序列，A－B－F－O、F－B－G 分别表示此次会话过程中两次事务访问的页面请求序列。

一次用户会话						
事务1				事务2		
Click1 (A)	Cilck2 (B)	Click3 (F)	Click4 (O)	Click5 (F)	Click6 (B)	Click7 (G)

图2 点击、会话、事务之间的关系

在点击流数据仓库中，建立事务星型模式，有利于在包含了用户页面访问序列(即用户浏览路径)的事务数据立方体上

挖掘一些隐含的知识，如频繁遍历路径。本文讨论的事务星型模式是在进行了用户识别之后的点击星型模式基础上建立的。所谓用户识别后的点击星型模式，就是在原来的点击星型模式中增加一个用户维，对每一个点击进行用户标识，用户维表包含属性集{User_key，User_IP，User_ID}。本文所设计的事务星型模式中，主要包括一个事务事实表(Transaction_fact table)和四个相关的维表：TimeOfDay 维表、Date 维表、URL_sequence 维表以及 User 维表。事务事实表包含了每一个相关维表的关键字以及它自身的一个度量值：事务经历的时间(Transaction_seconds)，维表分别用于跟踪事务访问的时间、日期、访问序列以及执行事务的用户。

对于事务星型模式，我们可用基于 SQL 的数据挖掘查询语言 DMQL[2]来定义。数据仓库中多维数据集存在的模式可以使用两种原语来定义：一种是立方体定义，一种是维定义，其语法形式分别如下所示[2]：

define cube 〈 cube_name 〉 [〈 dimension_list 〉]：〈measure_list〉；

define dimension 〈dimension_name〉 as (〈attribute_or_subdimension_list〉)

事务星型模式的 DMQL 定义如下：

define cube Transaction_star [TimeOfDay，Date，URL_sequence，User]：
Transaction_seconds = sum (Click_seconds)

define dimension TimeOfDay as （TimeOfDay_key，Hour，Minute，Second)

define dimension Date as （Date_key，Month，Quarter，Year，Day_of_week，Workday，holiday)

define dimension URL_sequence as （URL_sequence_key，URL_sequence)

define dimension User as （User_key，User_IP，User_ID)

从设计的事务星型模式可以看出，该模式与点击星型模式和会话星型模式的最显著的区别在于，该模式中用一系列有意义的页面组合序列代替了单个的点击序列。这种替代在点击流数据分析中的优点主要表现在以下两方面：①提高了数据仓库的查询性能。如为查找出某一用户一次会话过程中的访问点击序列，我们就不必再对用户的每个点击事实表进行复杂的多重连接查询，而只需在事务星型模式上执行一次简单的查询操作即可，从而加快了查询执行的效率。②有利于进行更深层次的数据挖掘分析。对于点击流数据的分析，我们经常需要做的事情便是如何通过用户的点击序列分析来得出其特有的会话或事务访问模式，而这些分析又往往都是建立在一定的会话或事务访问序列基础上的。也就是说，在分析之前，必须对用户的单个点击进行聚合从而形成有意义的页面组合。而我们设计的事务星型模式正符合了这种要求，因而直接在事务星型模式上进行挖掘，可免去很多的数据预处理工作。

3 事务星型模式的实现

在数据仓库中建立事务星型模式，主要包括以下两个步骤：①创建事务数据库；②在事务数据库上定义事务星型模式。其中关键步骤是创建事务数据库。而得到事务数据库的关键是进行事务识别[4]，即从用户会话的有序点击序列中识别出其中的每一次事务点击序列。事务识别的主要步骤是事务分割，就是将一次会话过程中的页面序列分割成一组组的事务点击序列。根据组成事务的页面是辅助页(引导用户访问内容页的一系列页面)还是内容页(用户感兴趣或最想访问的页面)，可以将事务定义成两种类型的事务：由辅助页和内容页组成的混合型事务，以及仅包含内容页的内容事务。事务识别的分割方法通常有3种：引用长度分割法、最大向前路径法以及时间窗口法[4]。

3.1 创建事务数据库

本文讨论的事务识别是在记录了特定用户的一系列有序点击序列的关系数据库基础上进行的，事先并没有对每个用户进行会话识别。因而为生成事务数据库，本文设计的事务识别方法同时考虑了事务识别和会话识别这样两个过程。事务识别方法主要参照最大向前路径法来设计。最大向前路径 MFP(Maximum Forward Path)法是以 Chen 等人提出的最大向前引用工作[5]为基础的，MFP 指的是一次用户会话中从第一页到开始回退的前一页之间的所有页面组成页面集合。如对图2所示的一次用户会话，请求的页面序列为 A－B－F－O－F－B－G，对应的 MFP 为 A－B－F－O，F－B－G。与引用长度分割法中利用内容页和辅助页之间的时间分割点来划分事务相比，最大向前路径法采用 MFP 来划分事务。因而，利用该方法对于图2所示的一次用户会话进行事务分割，所得的两种类型的事务分别为①混合型事务：A－B－F－O，F－B－G；②内容型事务：O－G。会话识别的方法比较简单，主要是设定一个所允许的最长会话时间值。

以下将给出生成事务数据库的基本实现算法。该算法的主要输入是这样一个三元对〈User_ID，Referer_page，Referer_time〉，User_ID 代表用户编号，Referer_page 代表引用的页面，Referer_time 代表引用页面的请求时间，实际使用时是一系列已经按页面请求时间排好序的三元对序列。假设 Time_out 为一表示所允许的最大会话时间常量，MFP 用于存储最大向前路径，tag 为一个方向标志，当 tag＝1时表示向前引用，tag＝0时表示向后引用，Transaction_D 表示生成的事务数据库。该算法的基本步骤如下：

```
step1：// 初始化工作。
      令 i = 1，MFP = null，tag = 1；
      start_time = referrer_timei；// 记录一次会话的起始时间
step2：// 一个新的用户点击序列开始。
      Current_page = Referer_pagei；
      If User_Idi changed then
       Begin
           把 MFP 中记录的最大向前路径输出到 Transaction_D 数据库；
           start_time = Referer_timei；// 重新设定一次会话的起始时间
           MFP = Current_page；
           转到 step5
       end
step3：// 判断最大向前路径中是否包含当前引用页面
      if Current_page 蕴含于 MFP then
       begin
         if tag = 1 then
           begin
             把 MFP 中记录的最大向前路径输出到 Transaction_D 数据库；
             start_time = Referer_timei；
           end
         去除 MFP 中 Current_page 之后的路径；
         tag = 0；
         转到 step5；
       end
step4：// 处理向前引用的情况
      if Referer_timei - start_time <= time_out then
       begin
         MFP = MFP + ；+ Current_page；
         If tag = 0 then tag = 1；
       End
```

```
    Else
     begin
       把 MFP 中记录的最大向前路径输出到 Transaction_D 数
    据库;
       start_time = Referer_time;
       MFP = null;
     end
step5:// 处理未处理完的记录
    i = i + 1;
    if User_IDi <> Null then
      转到 step2
      else
        把 MFP 中记录的最大向前路径输出到 Transaction_D 数
    据库。
```

下面通过一个简单的例子来说明利用该算法产生事务数据库的处理过程。假设{A、B、F、O、F、B、G、A、D}为某一用户的点击序列片段，其中 Time_out 设为30分钟，表1为该算法进行处理的详细过程。

3.2 创建事务星型模式

创建了事务数据库之后，我们就可以在事务数据库上按照预先设计好的事务星型模式来实际地构建相应的结构模式。这个过程比较简单，主要是利用 SQLServer2000 analysis service 中的相关功能来实现。

4 基于事务星型模式的关联规则应用

事务星型模式中，URL_sequence 维表中记录了经事务识别后的事务访问序列。根据事务定义的两种类型，我们可以分别对这两种类型的事务进行关联规则挖掘。对于由内容页和辅助页组成的混合型事务，通过挖掘我们可得到某一给定页面的一般访问路径，而对内容型事务进行挖掘则可得到站点上内容页之间的关联。

表1 事务数据库生成算法

循环	User_ID	Referer_page	Referer_time	Start_time	MFP	tag	Tranaction_D
1	102	A	15:04:41	15:04:41	A	1	—
2	102	B	15:05:34	15:04:41	A; B	1	—
3	102	F	15:06:22	15:04:41	A; B; F	1	—
4	102	O	15:10:02	15:04:41	A;B;F;O	1	—
5	102	F	15:11:03	15:11:03	A;B;F	1→0	A;B;F;O
6	102	B	15:11:45	15:11:03	A;B	0	—
7	102	G	15:12:23	15:11:03	A;B;G	0→1	A;B;G
8	102	A	17:05:22	17:05:22	A	1	—
9	102	D	17:06:03	17:05:22	A;D	1	A; D

以下给出关联规则在内容型事务上应用的简单例子。

表2是事务星型模式中 URL_sequence 维表的数据片断，其中的 URL 序列为由内容页组成的事务序列。

表2 事务星型模式的关联规则挖掘实例

URL_sequence_key	URL_sequence
1	ACD
2	ACDBE
3	BCD
4	CBF
5	ECFD
6	CEDB

用关联规则对上述事务序列数据片断进行挖掘。设定最小支持度为0.86，利用 Apriori 算法进行挖掘得到的最大项集是{C,D}。此关联规则模式表示，至少有86%的用户在一次访问中同时访问了C、D两个页面。假设C和D分别表示的是电子商务网站中两种不同的产品介绍页面，这就说明了这两种产品之间存在着较大的关联性。利用这样的关联规则，可以帮助实现电子商务网站进行产品智能促销的功能，一方面可以减少用户访问的浏览时间，另一方面也利于企业进一步挖掘潜在客户。

从以上关联规则挖掘的简单例子，我们看出在进行点击流数据分析时，建立事务星型模式是必须的，而且也是行之有效的。

结束语 本文针对点击流数据分析中特定的数据分析问题——事务序列分析，提出了建立事务星型模式的思想，并对其中的关键步骤——如何创建事务数据库进行了详细阐述。事务星型模式中包含了一次事务访问的全部点击序列。因而在此数据模型上，我们不但可以方便而快速地执行 OLAP 分析及查询操作，而且有利于利用数据挖掘技术直接对点击事务数据做更深层次的分析，从而减少了 Web 日志挖掘中原始日志记录的部分预处理工作。此外，事务数据模型中的事务序列还可以进一步与数据仓库中的其他信息，如交易记录、客户基本信息等相结合，从而完成特定的多维事务序列模式挖掘[6]任务。

参考文献

1 Kimball R. The Data Webhouse Toolkit. Wiley, 2000

2 Han Jiawei. Micheline Kamber 著，范明，孟小峰 等译. 数据挖掘概念与技术. 机械工业出版社，2001

3 罗运模，等著. SQL server 2000 数据仓库设计和使用指南. 人民邮电出版社，2001

4 Cooley R, Srivastava J. Data Preparation for Mining World Wide Web Browsing Patterns. Journal of Knowledge and Information Systems, 1999, 1(1)

5 Chen M S, Park J S, Yu Y. Data mining for path traversal patterns in a Web environment. In: Proc. of the 16th Intl. Conf. on Distributed Computing Systems, 1996. 385～392

6 Pinto H, Han Jiwaei, Pei Jian, Wang Ke. Mutil-dimensional Sequential Pattern Mining. M. Sc. thesis, Computing Science, Simon Fraser University, Apr. 2001