Web 日志挖掘中数据预处理的研究*

赵红玲[1]，宋瀚涛[1]，牛振东[2,3]，刘桂山[1]

(1. 北京理工大学 计算机系，北京 100081；2. 北京理工大学 软件学院，北京 100081；3. 中国数字图书馆有限责任公司，北京 100081)

摘　要: 针对框架式页面存在的问题,对数据预处理过程进行了改进,在数据清洗和用户识别部分添加了页面过滤部分,同时对预处理过程中的页面过滤算法和用户识别策略也进行了改进。

关键词: 数据挖掘；Web 日志挖掘；数据预处理

中图法分类号: TP393　　**文献标识码**: A　　**文章编号**: 1001-3695(2005)06-0067-03

Data Preparation in Web Log Mining

ZHAO Hong-ling[1], SONG Han-tao[1], NIU Zhen-dong[2,3], LIU Gui-shan[1]

(1. *Dept. of Computer Science, Beijing Institute of Technology, Beijing* 100081, *China*; 2. *School of Computer Software, Beijing Institute of Technology, Beijing* 100081, *China*; 3. *China Digital Library Co. Ltd., Beijing* 100081, *China*)

Abstract: Bettered the process of data preprocess and added a page-filtering in the data preprocess. What s more, both the algorithm of the page filtering and the technique of user identification were improved.

Key words: Data Mining; Web Log; Data Preprocessing

1　引言

随着 Internet 技术的发展,网络资源也迅猛增长。如何使 Internet 用户有效快速地获得所需的资源,已成为网站设计者亟待解决的问题。解决这个问题的途径之一就是将数据挖掘技术应用到 Web 服务器日志的挖掘,即通过挖掘服务器中的日志文件,获得用户的访问模式,从而可以进一步分析和研究日志记录的规律,来改进网站的组织结构和服务。

Web 日志挖掘是从 Web 的存取模式中获取有价值的信息或模式的过程,就是对用户访问 Web 时在服务器留下的访问记录进行挖掘。

Web 的基本结构多为: 客户端 - 代理服务器 - Web 服务器。客户端记录的是单个用户访问多站点的信息。代理服务器日志记录的是多用户访问多站点的信息。Web 服务器日志则记录多用户访问单站点的信息。Web 服务器日志记录了用户访问本站点的信息。典型的服务器日志包括以下信息: IP 地址、请求时间、方法(GET 或 POST)、被请求文件的 URL、HTTP 版本号、返回码、传输字节数、引用页的 URL(指向被请求文件的页面)和代理。

Web 日志挖掘多存在下面的困难:

(1) 由于本地缓存、代理服务器和防火墙的存在,使得 Web 日志中的数据并不准确,并不能真实地反映用户的浏览行为。

(2) 用户浏览 Web 的过程可以看作是一个“异步并发”的过程。多数情况下,用户将同时浏览多个页面。如果将用户浏览过程中的页面序列按照时间特性(如访问页面的开始时间)串行排列,则无法反映用户浏览行为的异步并发模式。

(3) 当用户请求一个框架式结构的网页时,浏览器会自动下载其包含的子网页。如果在进行数据预处理时忽略这种情况,那么得到的结果将是不准确的。

针对上述问题,本文在数据预处理过程中数据清理和用户识别步骤之间添加了页面过滤步骤,同时对预处理过程中的算法进行了改进。

2　数据预处理过程中的算法改进

2.1　数据预处理过程

数据预处理是在将日志文件转换成数据库文件以后进行的,其目的是把 Web 日志转换为适合进行数据挖掘可靠的精确数据。考虑到在数据挖掘中遇到的问题,将数据预处理过程分为五个步骤: 数据清理、页面过滤、用户识别、会话识别和路径识别。图 1 为 Web 日志挖掘的预处理过程。

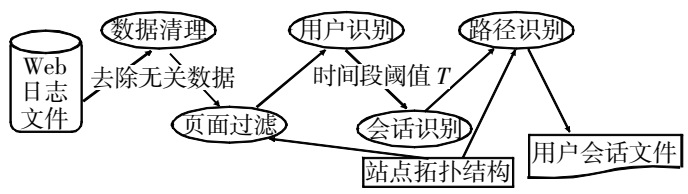

图 1　Web 日志挖掘预处理的过程次

数据清洗是指根据需求,对日志文件进行处理、删除与挖掘算法无关的数据。页面过滤是根据网页中 Frame 和 Subframe 之间的关系,将文件中对 Frame 和其 Subframe 页面的请求用 Frame 页面代替,从而删除文件中多余的 Subframe 页面。

收稿日期: 2004-04-13; 修返日期: 2004-06-21

基金项目: 霍英东教育基金会高等院校青年教师基金资助项目(91101)

会话识别就是将用户的所有的访问序列分成多个单独的用户一次访问序列,会话识别最简单的方法是时间戳方法。路径识别就是要找出用户会话中有意义的访问路径即获得用户完整的访问路径。

2.2 页面过滤算法的改进

Web 日志挖掘的目的是发现未知的用户行为模式,而 Frame 与 Subframe 之间的关系是已知事实,为此应当消除 Frame 页面对挖掘算法的影响,发现用户真正感兴趣的挖掘结果。该阶段的目的就是根据站点的拓扑结构中提取出的 Frame 与 Subframe 之间的关系,从数据清洗过程中生成的文件中,寻找 Frame 页面及其 Subframe 页面,将文件中对 Frame 和其 Subframe 页面的请求用 Frame 页面代替,从而删除文件中多余的 Subframe 页面。

前人对页面过滤算法的实现均是基于 Frame 和 Subframe 的关系表 FS, FS 是(Pid_{frame}, $Pid_{subframe}$) 对的集合, Pid_{frame} 是 Frame 页面的标识符, $Pid_{subframe}$ 是 Subframe 页面的标识符。但是基于栈的页面过滤算法并不能完全删除 Subframe 页面; 基于传递闭包的算法中需要额外的时间和空间计算和保存 FS^{+}, 为此我们提出了一种新的方法。

通过解析 Web 页面中的 Frame 标记可以得到 Frame 与 Subframe 之间的关系,我们将这种关系用树结构进行存储。图 2 是 Subframe 的页面,与其相对应的树结构如图 3 所示。

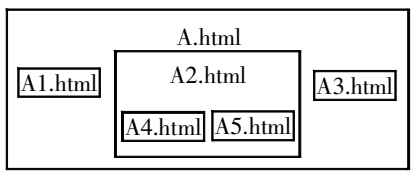

图 2 多 Subframe 的页面

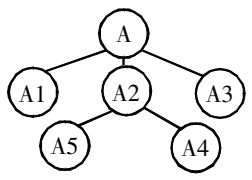

图 3 页面对应的树结构

树的存储结构如下:

```
typedef char DataType;  //应由用户定义
typedef struct{
DataType data; //节点数据
int parent;  //双亲指针,指示节点的双亲在向量中的位置
} PTreeNode;
```

图 3 中的树结构用下面的数组表示:

下标:	0	1	2	3	4	5
Data:	A	A1	A2	A3	A4	A5
Parent:	- 1	0	0	0	2	2

其中根无双亲,设 Parent 域为 - 1。

我们将这种结构成为 FS 树结构,根据 FS 树可以对数据文件进行 Frame 页面的过滤,算法如下:

```
for( 对于每条记录中访问页字段和引用字段的值)
  if( 该字段的值在 FS 树中)
    while( 该值对应的 parent 不等于 - 1)
      该条记录访问页字段的值 = 父节点的值;

for( 对于每条记录( r1, r2, …, rn) )
  if( ri = = rk)  //i > k
    册除 rk;
```

该算法在将访问页为 Subframe 的值用相应的 Frame 值代替的同时,也修改了引用页中值,即将 Subframe 的值用 Frame 值代替。算法在执行时不需要额外的空间和时间来计算 FS^{+} 的值,从而能够提高效率。

2.3 用户识别策略的改进

如果要进行用户访问模式的挖掘或对用户进行聚类分析,用户识别问题将是非常重要的。由于缓存、代理服务器和防火墙的存在,使得用户识别变得非常复杂。

前人在识别用户时采用的策略中当 IP 地址相同,用户使用的操作系统和浏览器也相同的条件下,则根据网站的拓扑结构进行识别:如果用户请求的某个页面不能从已访问的任何页面到达,则判断这是一个新的用户。如果网站的拓扑结构比较复杂,在根据拓扑结构识别页与页之间的关系时,效率会降低。为此本文对该策略进行了改进,在用户识别时,不利用网站的拓扑结构,而是根据引用页来识别,具体策略如下:

(1) 首先判断用户的 IP 地址,不同的 IP 地址代表不同的用户;

(2) 如果 IP 地址相同,则判断用户的浏览器或操作系统是否相同; 如果不同,则认为是不同的用户;

(3) 如果 IP 地址相同,用户的浏览器和操作系统也相同,则根据引用页进行判断:如果用户请求的某个页面没有请求过并且该条记录中的引用页为空,则认为这是一个新的用户。

需要说明的是:并非使用了这些规则就能准确地识别出用户。如具有相同 IP 地址的用户若在同样类型的机器上使用同种浏览器,并且请求的页面集合相同,那么很难识别。一个用户使用两种类型的浏览器,或是没有使用站点的链接结构直接输入 URL,则容易被认为是多个用户。

2.4 路径识别算法的实现

因为在页面过滤中,删除了数据文件中多余的 Subframe 页面,因此也忽略了 Subframe 页面中包含的超链接信息,所以在路径识别之前必须提升站点结构。站点提升是指将 Subframe 页面中的超链接信息添加到相应的 Frame 页面中。

路径识别就是要找出用户会话中有意义的访问路径。在路径识别之前首先要保证路径的完整性。用户在浏览网页时,通过按下浏览器上的“后退”按钮得到的页面是从本地缓冲区中得到的,在日志文件中是没有记录的,从而导致该页与用户上一次请求的页面之间没有超链接信息,所以要做的工作就是补全访问日志中没有记录的用户请求,获得完整的访问路径,这样才能正确地识别用户的有意义的访问路径。采用下面的算法进行填充:扫描某一用户的一次会话的访问路径 $P_1P_2\cdots P_iP_{i+1}\cdots P_n$,如果任何连续的两个点 P_i, P_{i+1}之间没有直接的物理链接,则说明访问者一定在 P_i 处进行了回溯,所以在 P_i 和 P_{i+1}之间插入节点 P_i,即路径为 $P_1P_2\cdots P_iP_{i-1}P_{i+1}\cdots P_n$,如果 P_{i-1}和 P_{i+1}之间还没有直接的物理链接,则在 P_{i-1}和 P_{i+1}之间插入节点 P_{i-2},此时路径为 $P_1P_2\cdots P_{i-2}P_{i-1}P_iP_{i-1}P_{i-2}P_{i+1}\cdots P_n$。重复此操作,直到该路径上的任意连续两点之间都有直接的物理链接。

3 实例分析

对表 1 中已经经过清洗的 Web 日志记录进行页面过滤,用户识别、会话识别、路径识别。网站的拓扑结构如图 4 所示。

对表 1 中的日志记录进行页面过滤,得到的结果如表 2 所示。

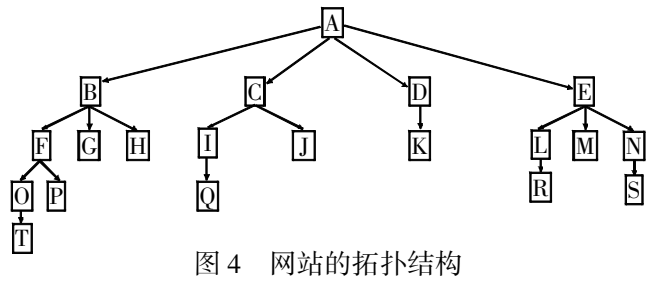

图 4　网站的拓扑结构

表 1　已经清洗的 Web 日志记录

ID	IP 地址	时间	URL	引用页	代理
1	202.107.231.145	05/Nov/2003:09:46:36 8000	A.html		Mozilla/4.0(Win+98)
2	202.107.231.145	05/Nov/2003:09:46:36 8000	A1.html		Mozilla/4.0(Win+98)
3	202.107.231.145	05/Nov/2003:09:46:36 8000	A2.html		Mozilla/4.0(Win+98)
4	202.107.231.145	05/Nov/2003:09:46:36 8000	A3.html		Mozilla/4.0(Win+98)
5	202.107.231.145	05/Nov/2003:09:46:36 8000	A4.html		Mozilla/4.0(Win+98)
6	202.107.231.145	05/Nov/2003:09:46:36 8000	A5.html		Mozilla/4.0(Win+98)
7	202.107.231.145	05/Nov/2003:09:46:39 8000	B.html	A4.html	Mozilla/4.0(Win+98)
8	202.107.231.145	05/Nov/2003:09:46:40 8000	L.html		Mozilla/4.0(Win+98)
9	202.107.231.145	05/Nov/2003:09:46:42 8000	F.html	B.html	Mozilla/4.0(Win+98)
10	202.107.231.145	05/Nov/2003:09:46:48 8000	A.html		Mozilla/4.0(Win+NT)
11	202.107.231.145	05/Nov/2003:09:46:48 8000	A1.html		Mozilla/4.0(Win+NT)
12	202.107.231.145	05/Nov/2003:09:46:488000	A2.html		Mozilla/4.0(Win+NT)
13	202.107.231.145	05/Nov/2003:09:46:48 8000	A3.html		Mozilla/4.0(Win+NT)
14	202.107.231.145	05/Nov/2003:09:46:48 8000	A4.html		Mozilla/4.0(Win+NT)
15	202.107.231.145	05/Nov/2003:09:46:48 8000	A5.html		Mozilla/4.0(Win+NT)
16	202.107.231.145	05/Nov/2003:09:46:51 8000	B.html	A4.html	Mozilla/4.0(Win+NT)
17	202.107.231.145	05/Nov/2003:09:46:55 8000	R.html	L.html	Mozilla/4.0(Win+98)
18	202.107.231.145	05/Nov/2003:09:46:57 8000	C.html	A5.html	Mozilla/4.0(Win+NT)
19	202.107.231.145	05/Nov/2003:09:47:07 8000	O.html	F.html	Mozilla/4.0(Win+98)
20	202.107.231.145	05/Nov/2003:09:47:15 8000	J.html	C.html	Mozilla/4.0(Win+NT)
21	202.107.231.145	05/Nov/2003:17:33:23 8000	G.html	B.html	Mozilla/4.0(Win+98)
22	202.107.231.145	05/Nov/2003:17:33:31 8000	A.html		Mozilla/4.0(Win+98)
23	202.107.231.145	05/Nov/2003:17:33:31 8000	A1.html		Mozilla/4.0(Win+98)
24	202.107.231.145	05/Nov/2003:17:33:31 8000	A2.html		Mozilla/4.0(Win+98)
25	202.107.231.145	05/Nov/2003:17:33:31 8000	A3.html		Mozilla/4.0(Win+98)
26	202.107.231.145	05/Nov/2003:17:33:31 8000	A4.html		Mozilla/4.0(Win+98)
27	202.107.231.145	05/Nov/2003:17:33:31 8000	A5.html		Mozilla/4.0(Win+98)
28	202.107.231.145	05/Nov/2003:17:33:35 8000	D.html	A2.html	Mozilla/4.0(Win+98)

表 2　经过页面过滤的 Web 日志记录

ID	IP 地址	时间	URL	引用页	代理
1	202.107.231.145	05/Nov/2003:09:46:36 8000	A.html		Mozilla/4.0(Win+98)
2	202.107.231.145	05/Nov/2003:09:46:39 8000	B.html	A.html	Mozilla/4.0(Win+98)
3	202.107.231.145	05/Nov/2003:09:46:40 8000	L.html		Mozilla/4.0(Win+98)
4	202.107.231.145	05/Nov/2003:09:46:42 8000	F.html	B.html	Mozilla/4.0(Win+98)
5	202.107.231.145	05/Nov/2003:09:46:48 8000	A.html		Mozilla/4.0(Win+NT)
6	202.107.231.145	05/Nov/2003:09:46:51 8000	B.html	A.html	Mozilla/4.0(Win+NT)
7	202.107.231.145	05/Nov/2003:09:46:55 8000	R.html	L.html	Mozilla/4.0(Win+98)
8	202.107.231.145	05/Nov/2003:09:46:57 8000	C.html	A.html	Mozilla/4.0(Win+NT)
9	202.107.231.145	05/Nov/2003:09:47:07 8000	O.html	F.html	Mozilla/4.0(Win+98)
10	202.107.231.145	05/Nov/2003:09:47:15 8000	J.html	C.html	Mozilla/4.0(Win+NT)
11	202.107.231.145	05/Nov/2003:17:33:23 8000	G.html	B.html	Mozilla/4.0(Win+98)
12	202.107.231.145	05/Nov/2003:17:33:31 8000	A.html		Mozilla/4.0(Win+98)
13	202.107.231.145	05/Nov/2003:17:33:35 8000	D.html	A.html	Mozilla/4.0(Win+98)

对表 2 中的日志记录进行用户识别, 发现有三个用户。其浏览路径分别为 A-B-F-O-G-A-B, A-B-C-J, L-R。接下来对日志记录进行会话识别, 一般情况下, 时间段设置为 30min, 得到的结果为 A-B-F-O-G, A-B-C-J, L-R。最后一步就是对站点结构的提升和路径的识别, 应用算法得到的结果为 A-B-F-O-F-B-G, A-D, A-B-A-C-J, L-R。

上述实例通过预处理过程的各个步骤得到的结果能够反映实际用户的访问模式。通过在数据预处理过程中增加页面过滤过程, 增加了数据的准确性, 页面过滤算法和用户识别策略的改进提高了效率。该算法将应用到数图工程中, 提高网站的个性化服务。

4　结束语

Web 日志数据的准确性是 Web 日志挖掘的一个重要前提和基础。只有准确的数据才能正确地反映用户的访问意图, 才能保证分析的正确性。本文所做的工作就是针对网站中框架式网页存在造成的日志信息的不准确性问题, 对 Web 日志数据的预处理过程进行了改进, 添加了页面过滤步骤, 并且对预处理过程中的相关算法也进行了改进。目前, 如何提高与改进 Web 日志挖掘预处理技术, 确保数据的正确性已经成为重要的课题和研究方向。

参考文献:

[1] 陆丽娜, 杨怡玲, 管旭东, 等. Web 日志挖掘中的数据预处理的研究[J]. 计算机工程, 2000, 26(4): 66-67, 72.

[2] Kamdaf T, Joshi A. On Creating Adaptive Web Servers Using Web Log Mining[EB/OL]. http://citeseer.nj.nec.com/kamdar00creating.html, 2002.

[3] Nanopoulos A, Katsaros D, Manolopoulos Y. Effective Prediction of Web-user Accesses: A Data Mining Approach[EB/OL]. http://citeseer.nj.nec.com/nanopoulos01effective.html, 2001.

[4] Bartolini G, Redpath R. Web Usage Mining and Discovery of Association Rules from HTTP Servers Logs[EB/OL]. http://www.prato.linux.it/2gbartolini/pdf/wum.pdf, 2001.

[5] [加] Han J, Kamber M. 数据挖掘概念与技术[M]. 北京: 机械工业出版社, 2001.

[6] Pei J, Han J, Mortazavi-asl B, *et al*. Mining Access Patterns Efficiently from Web Logs[C]. Proc. of the 4th Pacific-Asia Conf. on Knowledge Discovery and Data Mining, 2000. 396-407.

[7] 陈宝树, 党齐民. Web 数据挖掘中的数据预处理[J]. 计算机工程, 2002, 28(3): 125-127.

[8] Bamshad Mobasher, Robert Cooley, Jaideep Srivastava. Automatic Personalization Based on Web Usage Mining[EB/OL]. http://maya.cs.depaul.edu/~mobasher/personalization/, 2002.

[9] 徐宝文, 张卫丰. 数据挖掘技术在 Web 预取中的应用研究[J]. 计算机学报, 2001, 24(4): 430-436.

作者简介:

赵红玲, 女, 硕士研究生, 主要研究领域为 Web 挖掘; 宋瀚涛, 男, 教授, 博士生导师, 主要研究领域为人工智能、多媒体技术; 牛振东, 男, 软件学院院长, 中国数字图书馆有限责任公司技术总监, 美国卡内基梅隆大学兼职研究员, 香港大学客座教授, 博士, 主要研究领域为数字图书馆; 刘桂山, 男, 教授, 硕士生导师, 主要研究领域为人工智能、多媒体技术。