一种 WEB 日志挖掘的数据预处理方法①

符　翔　金　瓯　(中南大学 信息科学与工程学院 湖南 长沙 410083)

摘　要： Web 日志是目前 Web 数据挖掘的重要研究方向。数据预处理是 Web 日志挖掘中的关键技术。详细的介绍了 Web 日志挖掘的预处理过程。数据预处理包括数据清理、识别用户、识别会话和框架页面清理、路径补充。用户识别后，框架页面降低了数据挖掘的效率，可以通过过滤框架页面大幅度减少产生的无效页面数。

关键词： Web 日志挖掘;数据预处理;框架页面;过滤;会话识别

Data Preprocessing Method for Web Usage Mining

FU Xiang, JIN Ou

(School of Information Science and Engineering,Central South University, Changsha 410083, China)

Abstract： Web log mining is an important research direction about web mining. Data preprocessing is a key technology in web log mining. The article describes the preprocessing of mining logs in detail. Data preprocessing includes data clean, identifying user, recognizing session, cleaning up the frame of the page and supplementing path. After the user identification, the frame of the page reduces the efficiency of data mining. The number of invalid page can be significantly reduced through filtering the frame of page.

Keywords： web log mining; data preprocess; frame page; filter; recogniton sessi

1　引言

当前已经有很多科研工作者和研究机构发现网站日志数据具有很大的利用价值，希望通过对 web 日志的研究来进一步改善网站设计，使用户在更短的响应时间内找到他所需要的资源，增进用户体验，了解用户的兴趣和真正动机等。Web 访问日志挖掘过程一般分为三个步骤：(1)数据预处理：对原始的 web 日志文件的内容，记录，结构等转化为模式分析所能识别的有用的数据。(2)模式发现：利用包括统计学，模式识别，机器识别和神经网络等相关的算法对数据进行再处理从而生成模式。(3)模式分析：模式分析是从模型发现阶段发现的规则或模型过滤掉其中没有价值的部分，将有价值的模式提取出来。数据预处理是 web 日志挖掘的关键技术，其主要任务是从 web 日志文件中有效地识别用户访问会话。数据预处理的结果作为挖掘算法的输入直接影响日志挖掘的质量。一个 web 服务器是重要的数据来源，因为它明确记录了所有访问此网站的客户的浏览动作。它记录了多个用户对一个站点的访问信息。Web 使用记录的数据除了服务器的日志记录外，还包括浏览器端日志代理服务器日志、代理服务器日志、注册用户信息、登录信息、用户会话信息、交易信息、Cookie 中的信息、用户查询信息、鼠标点击等所有用户与网站之间可能进行的交互过程。这些日志文件可以按照不同的格式保存，日志记录的格式主要分为两种:通用日志格式 CLF(Common Log Format)和扩展型日志格式 ECLF(Extended Command Log Format)[1]，典型的日志记录形式如下:

192．110．0．17[29 / Jul / 2002：00：35：33 一 0500] “GET / Survey / history．htmHttp / 1．1”

① 基于项目:国家科技攻关计划(2003ba104c)
收稿时间:2009-11-22;收到修改稿时间:2010-01-05

20011631“http：／／www. djb. edu. cn／"“Mozilla／4. 0(compatible；MSIE5. 5；WindowsNT5 . O)”

典型的 Web 服务器日志包括以下信息：用户的 IP 地址、时间戳、方法(如 GET、POST)、被请求文件的 URL、超文本传输协议(HTTP)的版本号、返回码(请求的状态、成功或错误码)、传输字节数、代理(用户使用的浏览器和操作系统的类型)，有些扩展日志还包括参考页的 URL(用户从该页发出当前文件的请求)。

2 数据处理系统

Web 日志挖掘主要是对用户信息的分析，所以首先要从 Web 日志中识别出用户会话作为信息分析的基础，这样做的目的是把 Web 日志转化为适合进行数据挖掘的可靠的精确的数据。整个处理过程主要包括以下几个阶段：数据清洗、会话用户、识别用户会话、过滤框架页面和路径补充。

2.1 数据清理

数据清理是指根据需求，对日志文件进行处理，包括删除无关紧要的数据，合并某些记录，对用户请求页面时发生错误的记录进行适当的处理等等。当用户请求一个网页时，与这个网页有关的图片、音频等信息会自动下载，并记录在日志文件中；而如果我们挖掘的目的是用户访问模式，这些信息对我们来说显然用处不大(除非图片、音频等是用户显示请求的，即用户所需要的内容正是这些图片和音频等文件)，所以可以把日志中文件的后缀为 gif、jpg、jpeg 等的记录删除。但是，当挖掘的目的是为了进行网络流量分析或为页面缓冲与预取提供依据时，他们又变成了重要的信息来源，所以在删除这些记录的时候一定要把相关信息记录下来[2]。我们选择将其中的“发送字节数”和“接收字节数”这两个域的内容记录下来。此外，后缀名为 cgi、js 的脚本文件因对后面的分析处理不造成任何影响，所以应该删除。这个过程一定要根据正在分析的站点类型进行，如对一主要包含图形文档的站点，日志中的 GIF、JPEG 等文件可能是用户针对的请求。这时候就不能够随意的把图形文件删除。

2.2 用户识别

用户就是一个独立的个体，它通过一个浏览器访问一个或多个 w 出站点。但由于本地 cache、代理服务器、防火墙的存在，使得用户识别比较困难。如：通过代理服务器上网的用户在日志文件中的 IP 地址相同；由于防火墙的存在，在服务器日志中多个用户的 IP 地址都是相同的[3]。为此，目前的用户识别都是采用以下三条启发式原则：

1)如果用户的 IP 地址不同则认为是不同的用户。

2)如果 IP 地址相同，但浏览器软件或操作系统不同(用户代理域)，则认为是不同的用户。

3)如果 IP 地址相同，用户使用的操作系统和浏览器软件也相同，那么根据网站的拓扑结构对用户进行识别，如果用户请求的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。

2.3 会话识别

用户会话(User Session)S 是一个二元组<Uid，RS>[4]。其中 Uid 是用户标识。RS 是用户在一段时间内请求的 Web 页面。如果 RS 包含用户请求的页面的 URL 和请求的时间。则用户会话 S 表示为公式(1)所示的元组：

S=<Uid，{(urll，timel)，...，(urlk，timed}> (1)

日志文件中不同用户访问的页面当然属于不同的会话。当某个用户的页面请求在时间上跨度比较大时，就有可能是该用户多次访问同一个网站，我们可以将用户的访问记录分成多个会话来处理。最简单的方法就是设置一个 timeout 值，如果用户访问页面的时间差超过了这个值，则认为用户开始了一个新的会话。由于代理服务器和客户端的缓存，web 服务器日志并没有完整地记录用户的所有请求，因此 web 日志挖掘不能完全依赖于服务器日志，所以进行用户会话识别是比较困难的一项任务。web 日志数据预处理技术就是将原始的日志文件结合站点的结构和 web 页面的内容，经过一系列的数据处理转化为用户会话，主要包括数据净化、用户识别、会话识别、框架页面过滤和路径补充 5 个步骤。如下图 1 所示。

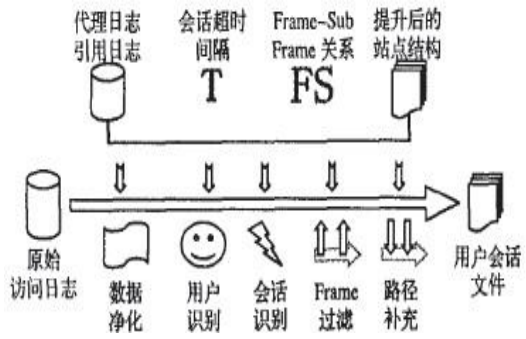

图 1 多 Web 日志处理的一般步骤图

3　框架页面过滤

HTML 规范通过“Frame”标记支持多窗口页面，每个窗口里装载的页面对应一个 URL，需要说明的是：Subframe 页面同时又可以是包含子窗口的 Frame 页面。如下图 2 所示。

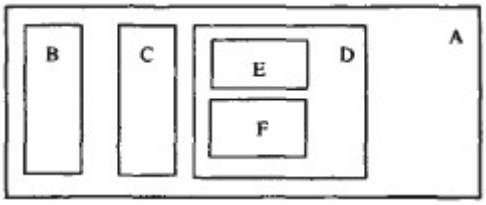

图 2　多框架网页结构图

当用户访问的 URL 对应的是一个 Frame 页面时，浏览器通过解释执行页面源程序自动向 Web 服务器请求该 Frame 页面中包含的所有 Subframe 页面，这一个过程可以重复进行，直到所有的 Subframe 页面被请求。我们应该根据网站的拓扑结构，路径补充之前进行框架的过滤。我们能得到 Frame 和 Subframe 的关系表 FS[5]。FS 表是(id_frame，id_subframe)对的集合，id_frame 是 Frame 页面的标识符，id_subframe 是 Subframe 页面的标识符。图 2 中的集合对为：{(A，B)，(A，C)，(A，D)，(D，F)，(D，E)}。根据此表对会话文件进行框架页面过滤的算法如下所示：

```
for(i=0;i<n;i++)//处理每个 frame
{
 String framecode=FrameList.get(i);
 delete(framecode,1);//去掉最后一个字符
 position=framecode.startWith("src");//子页面开始位置
if(position>0)
{
      framecode=framecode.trim(positon+4);
      if framecode.subString(1,2)='=' then
      delete(framecode.substring(1,2)//删除=号
      if(framecode[1]="")
      {
           delete(framecode,1,1)//去掉双引号
      }
      if(framecode.length()>0)//记录 frame 的父网页和子网页
      {
      setlength(FramePages,high(FramePagee)-low(framePages)+2);
      FramePages[high(FramePages)]. ParentPage=ExtractFileName(Filenames. String(i));

FremePages[high(FraraePages)].ChildPage=framecode
      }
}
}
```

4　路径补充

访问日志中是否有重要的请求没有被记录是识别会话中不得不面对的一个问题。要解决这个问题，我们就需要利用路径补充来找到这些没有被记录的信息。在客户端存在着缓存，用户很可能在浏览网页时，在当前的页面和上一次的请求页面没有超文本链接，而是通过按下浏览器上的”后退”之类的按钮得到的页面，而这个页面是从本地缓冲区中得到的[6]。在这种情况下，日志文件是没有用户的这次记录的。可以根据网站的拓扑结构，把用户的访问路径填充完整。检查引用信息确定当前请求来自哪一页，如果在用户的历史访问记录上有多个页面都包含与当前请求页面的链接，则将请求时间最接近的作为当前请求的来源，如果引用信息不完整，则可以利用站点的拓扑结构来代替。这样的话，路径补充就能完成将这些遗漏的请求补充到用户会话中。

5　实验结果分析

数据预处理技术已经在 WEB 服务器上进行了试验。运行的平台是 windows2000 Server。数据来自于 web 服务器上 10Mb 的服务器日志，记录了 2009-07-12 12:10:12 到 2009-07-14 11:22:10 时间段里的日志记录。数据中包含 430 个 html 页面，识别除了 1090 个用户会话，下表 1 列出了试验的结果数据。试验明示了一定性能的改进。

表 1 一般处理技术与改进技术的结果比较

方法	阀值	FG4	FG5	FG6	FG7
一般方法	62	54	35#	7#	0
	38	70	44	14	0
改进方法	25	33#	30*	0	0
	21	42	42	3*	0

阀值指包含频繁访问页组的最小用户会话个数，FGi|是长度为 i 的频繁访问页组的数目，#：表示用户感兴趣的频繁访问路径，*：表示用户不感兴趣的频繁访问路径。

6 结论

Web 挖掘的数据预处理的结果会直接影响到 web 挖掘的最终效果，本文对 web 挖掘中数据预处理阶段进行了研究。利用框架页面过滤的算法，用于删除用户会话中出现的 Subframe 页面，基本消除 Frame 页面对挖掘结果的影响。我们还应该根据 web 日志不同的特点，采用不同的数据预处理过程以提高数据预处理的质量从而提高 web 挖掘的质量。

参考文献

1 熊忠阳,周亚峰.web 访问挖掘的预处理技术的研究.计算机技术与发展, 2007,17(8):101－103.

2 费爱国,王新辉.一种基于 Web 日志文件的信息挖掘方法.计算机应用, 2004,24(6):57－59.

3 Baglioni M,Ferrara U,Romei A,et al. Preprocessing and mining Weblog data for Web personalization. Proc. of 8th National conf of the Italian Associa- tion for Artificial Intelligence. 2003.

4 刘立军,周军,梅红岩.Web 使用挖掘的数据处理.计算机科学, 2007,34(5):200－204.

5 Han JW, Kamber M. Data Mining.Beijing:Higher Education Press,2000.

6 邢东山,沈钧毅,宋擒豹.从 Web 日志中挖掘用户浏览偏爱路径.计算机学报, 2003,26(11):1518－1523.