Web行为下的正向关联规则挖掘研究

汤亚玲,秦 峰

(安徽工业大学 计算机学院,安徽 马鞍山 243002)

摘 要:Web关联规则挖掘研究用户对Web站点上不同页面之间的访问规律,为智能Web站点的个性化服务提供知识依据。文中讨论在Web使用挖掘中如何实现关联规则挖掘与访问序列相结合,挖掘切实有效的关联规则;具体阐述如何构造最大向前路径,并将关联规则与最大向前路径匹配、过虑的过程。试验证明得到的关联规则可作为智能Web站点的有效知识依据。

关键词:Web;关联规则;最大向前路径;个性化推荐

中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2007)08-0040-03

Research of Forward Association Rules Mining Under Web Behaviour

TANG Ya-ling, QIN Feng

(School of Computer, Anhui University of Technology, Maanshan 243002, China)

Abstract: The purporse of mining Web association rules is to research the visiting rules among different pages on Web site, which supports personal service of intelligence Web site. Discuss how to mine effective associations rules by integrating association rules mining and visiting sequence, and how to construct maximal forward path, with which the association rules will be filtered and matched in detail. The experiment results show that the association rules can do well support for intelligence of Web site.

Key words: Web; association rules; maximal forward path; personalization recommendation

0 引 言

挖掘关联规则[1]的任务在于发现大量数据中项集之间有用的相关联系,它是数据挖掘中的一个重要研究课题,关联规则是一个形如 $A => B$ 的蕴含式,是一种知识表现形式。其中,A 和 B 都是项目(Item)的集合,分别称为规则的前件和后件(其满足最小支持度和最小置信度)。但具体应用于Web使用挖掘中有其特殊的表现形式,事实上,Web关联规则(Web Association Rules,简称WAR)是一种知识的表现形式,与一阶逻辑的产生式大体相同,WAR是考察用户的客观访问规律所获取的知识[1,2],同时用户对Web站点的访问过程是与URL访问序列、访问时间有关系,如果在挖掘WAR时忽略这种关系,那么挖掘出的关联规则就仅仅是URL之间的一种关联关系,而割裂用户的实际访问规律,因此,将通常意义上的关联规则挖掘与实际访问序列相结合,考察关联规则的条件与结论及其内部项的时序关系。

严格意义上,我们需要的是基于用户访问序列的关联规则,而割裂了关联规则与用户访问序列的关联规则来作为用户特征的知识库很可能在实际的个性化Web服务系统中缺乏实际意义。因此,关联规则与用户访问序列相结合将更能准确地描述用户的行为特征。

1 挖掘最大向前引用序列

由于浏览器缓存等因素的存在,挖掘出的反向关联[2,3]规则带来的便利是方便用户的检索,直观上是方便了用户的需要,但是浏览器一般都内置缓存,反向关联规则并没有多大实际意义,因此,需要的是正相关联规则。如前所述,Web用户在访问感兴趣的信息时,倾向于通过链接来漫游,例如,用户为了到达当前主题的一个兄弟(兄弟主题的意思是处于同等逻辑位置)主题,总是利用“backward”后退至父主题(源主题),再向前作出选择,而不是打开一个新的根级的URL从头开始,因此在用户日志中的某些结点,被重复访问并非因其内容相关,而是其结构、位置特殊。为了从原始日志库中抽取有意义的用户访问模式,必须消除反向关联

收稿日期:2006-10-11

基金项目:安徽省教育厅自然科学基金资助项目(2006KJ062B)

作者简介:汤亚玲(1974-),男,安徽庐江人,讲师,硕士,研究方向为智能化信息处理及网络数据库系统;秦 峰,教授,研究方向为人工智能、数据挖掘及计算机网络。

的影响,因为反向关联旨在方便用户访问,而非满足用户的检索需求。这里采用寻找最大向前引用算法(Maximal Forward Reference,简称 MFR)[2,4]的思想与 Web 的超链结构特点相结合,用以挖掘用户访问模式。最大向前引用[2]指用户在浏览过程中,回退之前请求的最后一个页面。则最大向前路径(Maximal Forward Path,简称 MFP)指在用会话中由最大向前引用所分割的用户请求序列。例如:一个用户会话中请求的页面顺序是 A－B－A－C－D－C,则其最大向前引用的页面为 A 和 C,对应的最大向前路径为 A－B 和 A－C－D。这种方法是挖掘频繁遍历路径中最常被采用的算法,可应用 Web 行为挖掘[2,5]。因此,挖掘正向关联规则的过程由以下几步完成:

1)用 MFR 算法寻找所有最大向前路径。

2)由最大向前路径生成事务库挖掘产生关联规则。

3)过滤得到基于访问序列的正向关联规则。

下面首先介绍 MFR 算法的设计思想。

设:Web 访问序列为:(s_1,d_1),(s_2,d_2),(s_3,d_3),…,(s_n,d_n),其中 s_i 为引用页,d_i 为请求页。

MFR 算法分五步完成,其流程如下:

(1) 初始化:循环变量 $i = 1$;存储最大向前路径的字符串变量 Y = null,标志变量 $F = 1$($F = 1$ 表示处理向前路径)。

(2) 读取访问序列(s_i,d_i),$A = s_i$,$B = d_i$

如果 A 为空,则

如果 Y 不为空,则将 Y 写入路径库中;

$Y = B$;

转(5);

(3) 如果 B 与 Y 中某一个引用相同(假设是第 J 个引用),则

如果 $F = 1$,则

将 Y 写入路径库中;

释放 Y 中从第 J 个开始的引用;

$F = 0$;转(5)

(4) 否则,将 B 添加到 Y 中

如果 $F = 0$,令 $F = 1$

(5)$i = i + 1$,如果序列非空,则转(2);否则将 Y 写入路径库中,算法结束。

图 1 是一个简化的 Web 站点拓扑结构图,考虑浏览器缓存使得 Web 日志中没有回退路径的记录,在经过路径补充后[2,6],某用户访问 URL 的序列是 RADEDAFGHGIRCJ(单个字母表示一个 URL,R 是主页根 URL),转换成对应的 Web 访问序列是:(null,R),(R,A),(A,D),(D,E),(E,D),(D,A),(A,F),(F,G),(G,H),(H,G),(G,I),(I,R),(R,C),(C,J)。寻找最大向前路径得到的结果是:{R,A,D,E;R,A,F,G,H;R,A, F,G,I;R,C,J}。

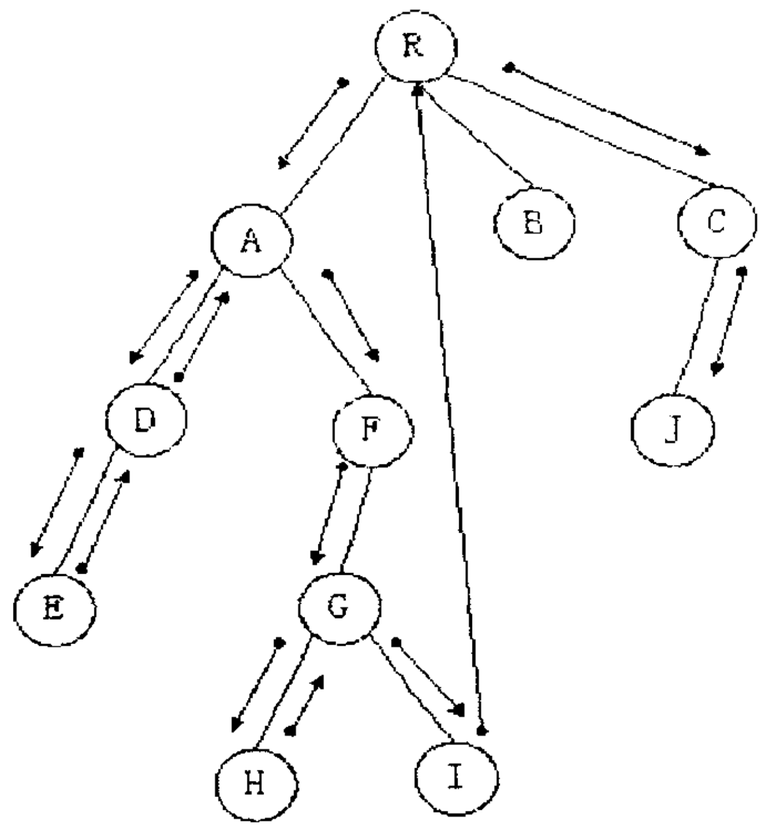

图 1　页面访问序列图

当用户访问一个曾经访问过的 URL 时,称出现了反向关联。反向关联的发生意味着一个正向关联路径的结束,并产生最大向前关联路径,然后回溯到该前向关联路径的起点,再继续寻找其他的前向关联路径;另外,源结点(即无父结点的结点)的出现也意味着反向关联路径的结束及新路径的开始。

2　挖掘 Web 关联规则

得到最大向前路径序列后,应用具体的关联规则挖掘算法对基于最大向前引用路径的事务集合进行挖掘得到基于访问序列的 Web 关联规则。

所得到的 WAR 的形式是:

$$<\mathrm{url}_1,\mathrm{url}_2,\cdots,\mathrm{url}_m>=><\mathrm{url}_{m+k},\mathrm{url}_{m+k+1},\cdots,\mathrm{url}_{m+m}> \quad (1)$$

式(1) 中的 url_i 表示用户在访问 Web 站点过程中访问到的第 i 个 url。

2.1　关联规则挖掘

由于 Web 关联规则挖掘的特殊性,如:Web 页面众多,页面更新情况,以及用户的访问数据的阶段性积累,需要选定一种切实有效的增量挖掘算法来实现关联规则的挖掘过程,解决阶段性关联规则的产生以及关联规则的增量更新[7],而对于频繁项集的挖掘,近年来的研究成果很多,如经典的 Aprioir 算法及其改进算法,DHP(Direct Hashing and Pruning)算法,FP_Growth 算法[1]等等。

增量挖掘算法,大体上有统计学的方法和结合数理逻辑[4,7,8]的方法,增量挖掘要考虑的两种情形是:

①在支持度阈值 minsup 置信度阈值 minconf 保持不变的情形下,增量的事务库 db 添加到原事务库中

DB 中求 db∪DB 的关联规则;

②给定的事务库,当支持度或置信度发生变化时如何重新生成所需的关联规则。

对于情形①FUP(Fast Update Algorithm)算法[5]较好解决了这个问题,而情形②,计算起来要复杂一些,文中假设在支持度不变的情形下挖掘。

在增量挖掘算法中,有一个很有用的定理,可以在增量挖掘中大大减少计算量。

定理 1:如一个给定事务集是频繁的,则它在原事务库是频繁的或者在增量事务库中是频繁的。

此定理的证明是很简单的,此处限于篇幅,不给出它的证明。利用定理 1,在进行增量更新时,可以将那些在 DB 和 db 中都不是频繁的事务集排除在外,可以避免不必要的计算量。

下面给出本课题选取增量挖掘算法的伪语言描述,算法首先计算增量数据库中频繁项集 Ldb,假设 DB 中的频繁项集已求得,为 LDB,设:

$L1 = L\text{DB} \cap L\text{db}$

$L2 = L\text{DB} - L1$

$L3 = L\text{db} - L1$

其中:| DB |,| db | 分别代表 DB 数据库中事务集的数量和 db 数据库中事务集的数量,T 代表事务,Minsup 是最小支持度阈值。

增量挖掘算法伪语言描述:

① 更新任意 $C \in L1$,C 的支持度

$L1.C.\text{support} = (L\text{DB}.C.\text{count} + L\text{db}.C.\text{count})/(|\text{DB}|+|\text{db}|)$

② 更新 $L2$ 频繁项的支持度,并保留满足最小支持度阈值的频繁项。

For all $T \in$ db

For all Itemset $C \in L2$

If $C \subseteq T$ then ++ $L2.C.$count

For all $C \in L2$

If $L2.C.\text{count} < \text{minsup} * (|\text{DB}|+|\text{db}|)$ then delete C from $L2$

③ For all $T \in$ DB

For all Itemset $C \in L3$

If $C \subseteq T$ then ++ $L3.C.$count

For all $C \in L3$

If $L3.C.\text{count} < \text{minsup} * (|\text{DB}|+|\text{db}|)$ then delete C from $L3$

④ $L\text{DB} \cup \text{db} = L1 \cup L2 \cup L3$

最后在 LDB ∪ db 得到增量后数据库 DB ∪ db 的频繁项集。

下一步的工作是产生关联规则,对于任何事务集合中的任一频繁项集 LS 满足最小支持度,可产生 L 的所有非空子集,对于 LS 中的每一个非空子集 S,如满足:

$$\text{support\_count}(LS)/\text{support\_count}(S) \geqslant \text{min\_conf}$$

则:$S \Rightarrow (LS - S)$

其中,support_ count(LS) 表示事物集 D 中包含 LS 的事务数,support_ count(S) 表示事物集 D 中包含 S 的事务数,min_ conf 是最小置信度阈值。

2.2 产生正向关联规则

现在得到的关联规则是通过挖掘 Web 访问序列中的最大向前路径所得到,但由于关联规则的挖掘过程是首先产生频繁项集,然后通过分析条件概率得到关联规则,从而破坏了关联规则前件与后件之间应有访问序列关系,可以通过简单序列匹配方法将关联规则与已有的访问序列集合中的序列进行匹配,选出符合访问顺序的关联规则。产生正向关联规则的过滤方法简述如下:

```
Rules_filter(WARS, Rs, SeqS)
  {For each WAR_i in WARS do
  {String S_WAR_i = append(A, B); //g 规则连接成串
  If((∃Seq_j in SeqS)and(Issubstring(Seq_i, S_WAR_i))then Add(Rs, WAR_i);
  }
  }
```

其中:Issubstring(S, T) 判定 T 串是否是 S 串的子串),如果是返回真值,否则返回假值。

WAR_i 形式为 $< \text{url}_1, \text{url}_2, \cdots, \text{url}_m > => < \text{url}_{m+k}, \text{url}_{m+k+1}, \cdots, \text{url}_{m+m} >$,简写为 $A => B$,WARS 为关联规则集合,Seq$S$ 为最大向前路径的集合,Rs 为得到的正向关联规则集合。

3 应用 Web 关联规则

智能 Web 站点[6]的实现功能包含分析当前用户的访问序列,根据当前已经访问的 URL 序列推荐给用户可能感兴趣的 URL,这一功能由推荐引擎来实现,根据当前的用户会话产生实时的推荐集。

用户当前访问序列 Seq 可以表示为:$\text{Seq} = \{U, s_1, s_2, \cdots, s_n\}$,其中 U 表示用户,s_i 表示页面,总体使用特征 C 可以表示为:$C = \{A; R; I\}$,其中 A 表示用户所在聚类的特征,R 表示正向关联规则,I 表示用户兴趣。作为其中关联规则引擎的一个简单的例子,如图 2 所示。聚类引擎主要从当前会话中取出用户信

(下转第 47 页)

了一个运行时库打包工具 Runtime Packager,利用此工具可以很方便地把 PowerBuilder 运行时需要的文件打包,然后和应用程序一起发行。

3.2 安装数据库接口

当应用程序需要访问数据库时,在为用户安装应用程序的同时还必须为其安装好数据库接口文件。安装数据库接口文件包括两方面的内容:第一,安装 PowerBuilder 提供的专用数据库接口或 ODBC 驱动程序(根据应用程序要访问的数据库而定);第二,安装数据库厂商提供的数据库驱动程序。表 4 列出了访问大型数据库所需的 PowerBuilder 专用接口文件,这些文件应该安装在应用程序所在的目录或系统的搜索路径中。

表 4　Windows 下各数据库使用的专用接口

数据库管理系统	PowerBuilder 专用接口文件
Informix 9	Pbin990.dll
Oracle 8.X	Pbo8490.dll
Powersoft ODBC 接口	Pbodb90.dll,Pbodb90.ini
SQL Server 2000	Pbmss90.dll
Adaptive Server Enterprise(11.X,12.X)	Pbsyc90.dll

3.3 配置 ODBC 数据源

如果应用程序使用了 ODBC 数据源,在为用户安装应用程序的同时还必须为其安装和配置 PowerBuilder ODBC 驱动程序 Pbodb90.dll 和 Pbodb90.ini,这两个文件应该安装在应用程序所在的目录或系统的搜索路径中。另外,还需要修改 ODBC 初始化文件 odbcinst.ini 和 odbc.ini,这两个文件通常在 Windows 目录下,如果用户机器上没有这两个文件,可从开发环境中复制[1]。

4 结束语

文中就基于 Windows 平台的 PowerBuilder 9.0 应用程序编译发布的关键技术进行了分析,并给出了四种编译成可执行文件的打包模型。详细介绍了在 PowerBuilder 9.0 中创建工程的步骤、编译格式的选择、可执行文件的建立、资源文件的使用、动态库的优点、应用程序的发行环境等,为脱离开发环境运行 PowerBuilder 应用程序打下了坚实的基础。

参考文献:

[1] 樊金生,张翠肖,沙　金,等. PowerBuilder9.0 实用教程[M].北京:科学出版社,2004.

[2] 柯建勋,张　涛. PowerBuilder8.0 进阶篇[M].北京:清华大学出版社,2002.

[3] 杨　昭. PowerBuilder9.0 对象与控件技术详解[M].北京:中国水利水电出版社,2003.

[4] 张遂芹. PowerBuilder9.0 系统开发实例[M].北京:中国水利水电出版社,2003.

[5] 吕睿烜,李　勇,温为民. PowerBuilder9.0 开发精解[M].北京:电子工业出版社,2003.

(上接第 42 页)

息,然后找到该用户所在的聚类,并把该类用户所具备的共有属性(类用户的 Web 关联规则、用户兴趣)推荐出来。

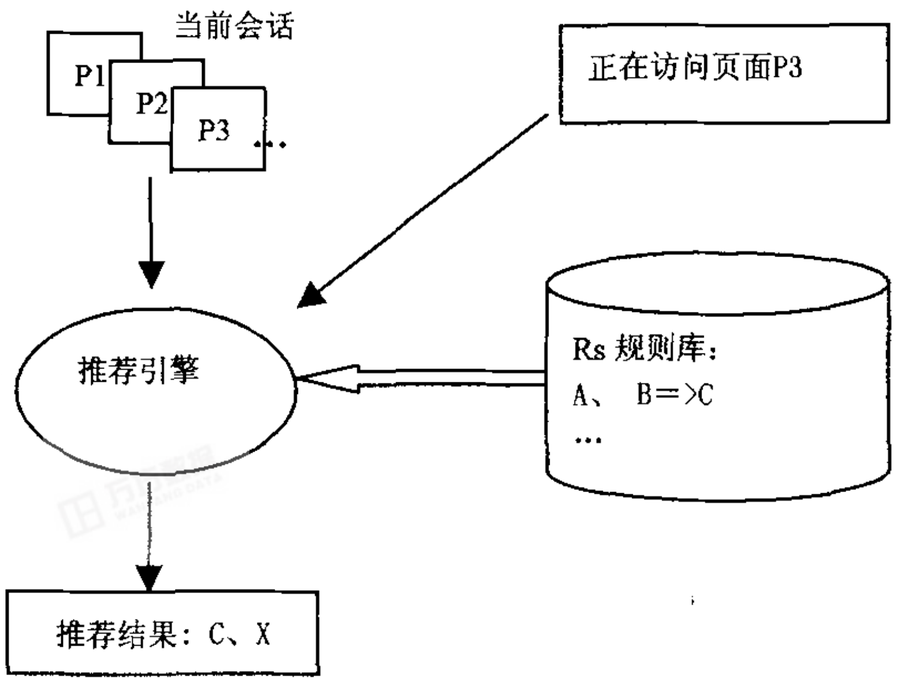

图 2　关联引擎推荐图

4 结束语

Web 关联规则作为智能站点知识库的重要组成部分,对实现用户使用的个性化具有重要的意义,文中综合考虑了关联规则挖掘过程中的用户的实际访问序列,消除反向关联规则对系统知识库的影响,提高了系统的效率和准确性。

参考文献:

[1] Han Jiawei, Kamber M. Data Mining Concepts and Techniques[M].北京:机械工业出版社,2001.

[2] Pal S K, Talwar V, Mitra P. Web Mining in Soft Computing Framework: Relevance, State of the Art and Future Directions[J]. IEEE Transactions on Neural Networks, 2002, 13(5):1163-1169.

[3] 史忠植. 知识发现[M].北京:清华大学出版社,2002.

[4] Lu Jieping, Liu Yuebo, Ni Weiwei, et al. A fast interactive sequential pattern mining algorithm[J]. Wuhan University Journal of Natural Science, 2005, 11(6):3136-3139.

[5] 冯玉才,冯剑林.关联规则的增量更新算法[J].软件学报,1998,8(4):301-306.

[6] 汤亚玲,崔志明.基于遗传算法的 Web 行为挖掘研究[J].微电子学与计算机,2006,23(8):168-170.

[7] 金　阳,左万利.有序概念格与 WWW 用户访问模式的增量挖掘[J].计算机研究与发展,2003,40(5):635-683.

[8] 朱玉全,孙志挥,赵传申.快速更新频繁项集[J].计算机研究与发展,2003,40(1):94-99.