一种基于最大频繁项目集的挖掘事务间关联规则方法*)

任永功　张琰渝
（辽宁师范大学计算机与信息技术学院　大连 116029）

摘　要　Web事务间关联规则挖掘是通过发现网页之间的关联关系来预测用户的兴趣。提出一种新的事务间关联规则挖掘方法，通过对MAFIA算法改进，得到最大频繁项目集的同时得到对应的共有用户集，通过对事务内到事务间最大频繁项目集的转换，分析不同用户之间的关系，分析用户对网站上不同网页的访问数据，直接发现不同用户之间的关联关系来预测用户的兴趣。该方法经试验证明能够更加全面的预测用户感兴趣的网页，更好地为用户提供个性化服务。

关键词　Web事务间关联规则，改进的MAFIA算法，最大频繁项目集，用户兴趣模型

Method of Mining Inter-transaction Association Rules Based on Maximum Frequent Itemsets

REN Yong-gong　ZHANG Yan-yu
(School of Computer and Information Technology, Liaoning Normal University, Dalian 116029, China)

Abstract　Mining Web inter-transaction association rules is to predict the users' interesting by finding the relationship among Web pages. We proposed a new method of mining inter-transaction association rules, that improve the Mafia for getting the maximum frequent itemsets and the relative CUI(Common User Intersection), transform the maximum frequent itemsets from intra-transaction into inter-transaction, analysed the relationship among different users and analysis the visiting information of Web pages. The association rules among different users instead of among different Web pages can be found. The experiment proved that this method provides more Web pages which could content users' interesing and serve the users more personalized.

Keywords　Web inter-transation association rules, Improved algorithm of mafia, Maximum frequent itemsets, User's interesting model

1　引言

如今的WWW网上，对于网站管理者而言，最关注的问题是如何为网站的用户提供个性化服务[1]，获得更多的浏览量。对网络用户而言，最关注的问题是如何在信息的海洋中方便快速的找到自己所需的信息。而Web使用挖掘[2]是从Web数据中发现用户使用模式以及行为习惯的数据挖掘技术，它的存在使上述的两个问题迎刃而解。关联规则是Web使用挖掘中的一个挖掘方法，分为事务内关联规则和事务间关联规则[3]。以往大部分的关联规则的挖掘是基于事务内的，并将着重点放在用户所访问的页面上，查找网页之间的关联[4]，而忽略了页面的访问者、一切服务的最终受益者和技术进步的推动者——“用户”。用户是一切网络活动的中心，所以本文采用了预测性更好、能够在不同事务之间发现关联规则的事务间关联规则来挖掘不同用户之间存在关联关系，从而能够更准确、更有预见性、更加全面的为用户提供含有他们感兴趣信息的网页。

下面举例说明Web事务间关联规则[5]和事务内关联规则的差别：

例1　一个用户浏览了网页P1，P2和P3。那么可以发现关联规则P1P2⇒P3，从而得到用户浏览了P1和P2之后，有60%的可能性会浏览P3。

例2　用户U1、U2访问了页面P1，P2，P3，用户U5访问了页面P1，P5，P6。那么可以发现关联规则U1U2⇒U5，从而得到用户U1访问页面P1和P2之后，有60%可能性会访问P5，P6。

通过对上述两个例子进行比较发现，例1非常有局限性，着重点在于某个用户浏览的网页，发现这些网页之间的关联规则。而例2不仅将着重点放在用户上，还在不同的用户之间寻找关联规则。并且对用户想要浏览的网页能够给与范围更广、准确性更高的预测。可见例2虽然更加复杂，但能够在不同用户之间发现关联规则，并合理预测用户感兴趣的网页。显然是更加方便有用的。本文从用户代表事务、用户所访问的网页为事务中的项目的数据库中，通过改进的Mafia[6]算法以及相应的从事务内到事务间的转换方法，得到事务间最大频繁项目集，从中挖掘不同用户之间的关联规则，使用这些关联规则来预测用户浏览趋势。当某个用户浏览了某个网页的同时，网站预测此用户还会对哪些网页感兴趣，并将这些网页推荐给他们，从而使用户快捷的获得他们想要的信息。

2　问题描述

2.1　Web关联规则的基本描述

假设某个网站上包含着若干的网页的集合为$P=\{p_1, p_2, p_3, \cdots, p_m\}$，访问此网站的用户的集合为$U=\{u_1, u_2, u_3,$

*)国家自然科学基金项目(60603047)，辽宁省自然科学基金，辽宁省教育厅高等学校科研基金(2008341)，大连市优秀青年科技人才基金(2008J23JH026)。**任永功**　教授，博士，研究方向为数据挖掘、图像处理技术等；**张琰渝**　硕士研究生，研究方向为Web数据挖掘。

…,u_n},每一个 $p_i \in P$ 都有一个为了识别方便而设置的编号，记为 TID(Transaction ID)，其中 p_i 便可称为一个事务，其中所包含的各个浏览此页面的用户为项目。

则有：$p_i=\langle(u_1,w(u_1)),(u_2,w(u_2)),\cdots,(u_n,w(u_n))\rangle$。

因为对于某个特定的网站来说，网页是有限的，而访问者也是有限的，每个访问者对于网页的访问都会在网页上留下相关的信息。如上式中，$w(u_n)$是一个权值，当 $w(u_n)=1$ 时，用户 u_n访问过网页 p_i，而当 $w(u_n)=0$ 时，用户 u_n没有访问 p_i。本文就是挖掘不同用户之间的关联规则，从而将某些用户所感兴趣的网页认定为同他们有关联的其他用户也感兴趣的网页，并进行推荐。

2.2 相关定义和性质

定义 1(滑动窗口[7]) 像例 2 中所举的例子那样，活动发生在两个不同时间的事务中，在这样事务中寻找的关联规则称为事务间关联规则。如果将所有事务展开，寻找事务间关联规则，无疑对数据的处理是巨大且低效的。所以引入了滑动窗口技术[7]，滑动窗口 W 在一个事务数据库中是一段连续的区间，区间数为 w(即子窗口的个数)。如图 1，w 为 4。

本文中使用的是固定窗口大小的滑动窗口方法，滑动窗口的大小由使用者根据数据集的情况自己规定。当规定窗口大小后，窗口的两端随着新项目的加入而不断向同一个方向移动。

定义 2(使用者扩展集) 使用者扩展集(UES)即一个元事务。就是将每个滑动窗口中的各个项目扩展到一个集合当中，这样便可以打破事务的限制来进行事务内到事务间的转换。

即 $UES=\{ei(j)\mid ei\in W[j],\ 1\leqslant i\leqslant u,\ 0\leqslant j\leqslant w-1\}$。其中，$u$ 为每个事务中所包含项目的个数，w 为滑动窗口的跨度。

定义 3(共有用户集) 共有用户集(CUI)是一个包含为了挖掘用户间关联规则而找到的某个最大频繁项目集中每个项目在数据库中都出现过的事务的 TID 集合。

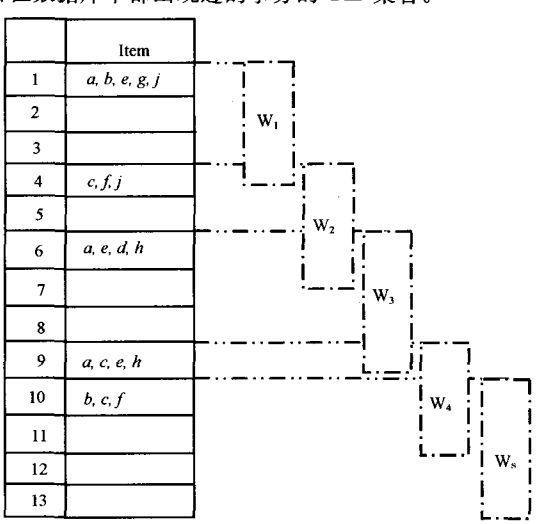

图 1 滑动窗口定义

定义 4(关联规则的支持度) 关联规则 $X\Rightarrow Y$ 的支持度定义为项集 $X\cup Y$ 的支持度(support)，记做 support($X\Rightarrow Y$)。

关联规则的支持度计算如下：

$$support(X)=\frac{count(X)}{|D|}$$

$$support(X\Rightarrow Y)=support(X\cup Y)$$

定义 5(关联规则的置信度) 关联规则 $X\Rightarrow Y$ 在 D 中的置信度(confidence)是交易中包含项集 X 的情况下也包含项集 Y 的条件概率，用 $confidence(X\Rightarrow Y)$ 表示。$confidence(X\Rightarrow Y)$计算如下：

$$confidence(X\Rightarrow Y,D)=P(Y|X)=\frac{support(X\cup Y)}{support(X)}$$

定义 6(强规则) 用户根据数据集的情况给定最小支持度和最小置信度。若 $support(X\Rightarrow Y)\geqslant$最小支持度，且 $confidence(X\Rightarrow Y)\geqslant$ 最小置信度，称关联规则 $X\Rightarrow Y$ 为强规则，否则称关联规则 $X\Rightarrow Y$ 为弱规则。关联规则挖掘的任务就是挖掘出数据库 D 中所有的强规则。

定义 7(最大频繁项目集) X 是数据库 D 中 n 个项目的集合，当 X 的支持度不小于用户给定的最小支持度，则称 X 为 D 中的频繁项目集。当 X 没有其它频繁项目集为超集时，X 则是一个最大频繁项目集。

性质：频繁项目集的任何真子集都不是最大频繁项目集。

定义 8(事务间关联规则) 一个事务间关联规则包含形如：$X\Rightarrow Y$

1. $X\in\sum, Y\in\sum$。
2. 存在 $e_i(0)\in X, 1\leqslant i\leqslant u$。
3. 存在 $e_i(j)\in Y, 1\leqslant i\leqslant u, j\neq 0$。
4. $X\cap Y\neq\varnothing$。

3 基于最大频繁项目集的事务间关联规则方法

Web 使用挖掘是应用数据挖掘的技术来从 Web 信息中发现用户的行为模式，能够更好的理解用户的需求来为用户提供个性化服务。本文通过数据预处理与分析、模式发现、关联规则生成这 3 步来完成用户间关联规则的发现。

3.1 数据预处理

从 Web 日志上得到的原始数据不能直接使用，必须经过预处理[8,9]。数据预处理对实验的准确性是非常重要的。本文去掉了原数据库中无用的数据，并将数据库由网页→用户的形式改为用户→网页，更加的符合本文直接寻找用户间关联关系的方法。

3.2 模式发现

通过改进的 Mafia 算法找到最大频繁闭项目集以及每个项目集对应 CUI，进行最大频繁项目集事务内到事务间的转换，分析转换之后的项目集在不同滑动窗口内的包含关系即可发现用户的使用模式。

步骤分别如下：

第一步：找到并以适当格式存储最大频繁项目集。

由于网络上需要处理的数据量是巨大的，所以采用处理大数据集效率比较高的 Mafia 算法来找最大频繁项目集。但是 Mafia 并不能够记录所找到的每个最大频繁集中所有项目对应的原数据库中事务号(Tid)的交集。本文改进了 Mafia，在找到最大频繁项目集的同时实现每个项目与其所出自的事务号对应，并存储每个最大频繁项目集所对应 CUI。

算法 1 记录 CUI 的 Mafia 算法

输入：要找关联规则的数据库

输出：最大频繁项目集、每个最大频繁闭项集对应的 CUI

```
///原始的 mafia 算法
input the database
//创建 DFS 空间和候选项目树
Simple(Current node C, MFI)
{……
```

```
  }
//三种剪枝策略
//第一种 Parent Equivalence Pruning
PEP(Current node C, MFI)
{ ……
  }
//第二种 HUTMFI Superset Pruning
HUTMFI(Current node C, MFI)
{……
}
//第三种 FHUT - Frequent Head-Union-Tail
FHUT(node C, MFI, Boolean Is HUT)
{……
}
Use Mafia to find the maxmum frequent itemsets
///生成每一个最大频繁项目集的 CUI
///获取原数据库的每一个项目的对应的 TID 并记录
read from the orignal database file;
for(getline form original database)
{
    stroe each(transnum1,Tidset1) in table1;
}
end for
///创建 CUI
read from the file included maxmum frequent itemsets info. ;
for(getline from 最大频繁项目集)
{
    split every number in every 最大频繁项目集;
    find (transnum2, Tidset2) in table1;
    stroe(transnum2, Tidset2);
    //取交集
    intersection(transnum2) named CUI;
      //(Tidset2,CUI)既为每一个最大频繁项目集以及对应的 CUI
    store(Tidset2,CUI)
}
end for
```

由改进的 Mafia 算法得到的 CUI，是通过本文提出的方法找关联规则的直接依据。通过找不同最大频繁项目集对应的 CUI 之间的包含关系，从而确定关联规则。

第二步：完成事务内到事务间的最大频繁集的转换。

本文将 CUI 中的值对应到原数据库中并使用滑动窗口中进行处理，完成事务内到事务间的最大频繁集的转换，寻找第一个子窗口与其后面的子窗口所对应的 CUI 的包含关系，从而发现用户使用模式。

具体方法是：

1. 利用算法 1，存储最大频繁项目集以及 CUI。

2. 每个滑动窗口都是从原数据库中非空的事务开始，令 t=原数据库中的事务(TID)，即 CUI 中的每个值；n = 滑动窗口的子窗口号。则有性质：如果某个项目 t 出现在第 n 个子窗口中，那么 t 必定是在以项目 $t-n+1$ 开头的窗口中。如果 $t-n+1$的值满足以下筛选方法的任意一项，则将 $t-n+1$ 对应的滑动窗口从最大频繁项目集中删除。

筛选方法：

1. 如果所得到的 Tidset 中存在小于等于 0、内容为空或者大于最大 Tid，则筛选掉。

2. 如果所得到的 Tidset 中存在包含关系，则将被包含者筛选掉。

筛选之后，按照子窗口排列各项目集，就得到事务间最大频繁项目集。

算法 2 将事务内最大频繁项目集转换成事务间最大频繁项目集，发现用户使用模式

输入：最大频繁项目集和每个最大频繁项目集对应的 CUI

输出：根据 CUI 中的信息生成事务间的最大频繁项目集，并发现用户使用模式

```
read from the database;
for(1≤n≤w);
for(Tidset, Transaction num2)
{
  New Transaction num = each one in Transaction num2-n+1;
  Add New_Transaction num to New Intratransaction Maxmum Frequent Itemsets;
}
///数据筛选
for(each New Transaction)
  //筛选方法一
  if(New Transaction num≤0 ‖ New Transaction num =∅‖ New Transaction num > Max Tidset number )
  delete New Transaction num;
  //筛选方法二
  if(New Tidset 1⊂New Tidset 2)
  delete New Tidset 1;
end for
```

事务间最大频繁项目集包含着用户的使用模式，只有先发现事务间最大频繁项目集，才能发现事务间关联规则。

3.3 事务间关联规则生成

在第二步中生成了事务间的最大频繁项目集，确定最小置信度，寻找 sub_window[1]后面的 sub_window 与 sub_window[1]对应的 CUI 存在的包含与被包含的关系[4](详情见表 3)，如果包含与被包含的项目的比例不小于最小置信度，就可以确立一个关联规则。如果不满足，则进行剪枝[10]，如此直到最后一个 sub_window，便能找到所有事务间关联规则。例如：表 3 中 sub_window [2]与 sub_window [1]的第一项 CUI 的比例有 67%，满足最小置信度，那么将 sub_window [1]的第一项置于“⇒”的左侧，sub_window [2]置于“⇒”的右侧，生成一个关联规则。

4 实验结果

本文使用了 msnbc990928 数据集。此数据集描述了一天之内综合门户网站 msnbc. com 上 17 个代表不同内容分类的子页面的访问情况，分别为“frontpage”，“news”，“ tech”，“local”，“opinion”，“on-air”，“misc”，“weather”，“ health”，“living”，“business”，“sports”，“summary”，“bbs”，“travel”，“msn-news”和“msn-sports”。

此数据库的格式如表 1，而对于本文而言，需要将原来的格式进行转换，转换之后的格式为如表 2。

表 1 转换前的格式

Users	Webpage
a	frontpage, frontpage
b	news
c	tech,news, news, ….
d	opinion
e	frontpage
…	…

表 2 转换后的格式

Webpage	Users
frontpage	a, e, g, k, …
news	b, c, …
tech	c, j, …
local	d, j, …
opinion	f, h, i, j, …
…	…

按照表 2 的格式来找最大频繁闭项目集。设滑动窗口的窗口大小为 4，最小支持度为 40%，使用改进的 Mafia 算法在得到事务内的最大频繁项目集的同时还得到了每一个集合中的每一项在原数据库中所在事务号的交集(对应算法 1)。再进行第二步，将找到的最大频繁项目集进行事务间的转换(对应算法 2)，得到的结果如表 3。

表 3 使用滑动窗口转换到的事务间最大频繁项目集

No.	Users(sub_window)	CUI
1	e, d, i, h, c, l, f, n, m[1]	2 3 4 5 7 14
2	a, l, f, m[1]	3 4 5 6 9 10
3	g, j, d, o, h, c, l, n, m[1]	1 2 3 4 5 7
4	k, d, c, l, n, m[1]	1 2 3 4 13 14
5	d, b, c, l, n, m[1]	1 2 4 5 7 13
6	a, l, f, m[2]	2 3 4 5 8 9
7	a, l, f, m[3]	1 2 3 4 7 8
8	p, l, f, n, m[4]	1 2 3 4 7 10

前面已经设最小支持度为 40%，此处设最小置信度为 60%。对于用户来说，存在一个容错的问题。即用户允许网站向他们推荐的网页中存在并不是他们兴趣所在的网页，但是要有一个限度。这个限度就是置信度。例如：第 6 行的交集有 67%是包含在第 1 行中的，那么就有 $e, d, i, h, c, l, f, n, m[1] \Rightarrow a, l, f, m[2]$ 置信度为 67%。

根据上述的寻找关联规则的方法，可以得到关联规则：

$e, d, i, h, c, l, f, n, m[1] \Rightarrow a, l, f, m[2]$，置信度为 67%。

$g, j, d, o, h, c, l, n, m[1] \Rightarrow a, l, f, m[3]$，置信度为 83%。

$k, d, c, l, n, m[1] \Rightarrow p, l, f, n, m[4]$，置信度为 67%。

需要注意的是生成前两个关联规则右边都是 a, l, f, m，其中第二个关联规则的置信度 83%，要大于第一个关联规则的置信度 67%。所以舍掉第一个。则最终生成的关联规则为：

$g, j, d, o, h, c, l, n, m[1] \Rightarrow a, l, f, m[3]$，置信度为 83%。

$k, d, c, l, n, m[1] \Rightarrow p, l, f, n, m[4]$，置信度为 67%。

从此可以看出，如果用户 l 看了网页 3，那么就可以推断网页 1、2、4、5、7、13、14 中有 l 想了解的信息，并推荐给用户 l。

结束语 本文提出了一种新的基于 Web 事务间关联规则的挖掘方法。以直接找用户之间关联规则的思想为基础，首先提出了用改进的 Mafia 算法找到最大频繁项目集以及对应的 CUI，然后以 CUI 为依据来对项目集进行由事务内到事务间的转换，相比从直接生成事务间项目集的找事务间关联规则的方法要简单且高效，能够推荐给用户更多包含他们感兴趣信息的网页，并随着网站数据的不断增加，仍然继承了 Mafia 算法对大数据集处理效率高的优点。

参 考 文 献

[1] Ting I H, Kimble C, Kudenko D. Applying Web Usage Mining Techniques to Discover Potential Browsing Problems of Users// Advanced Learning Technologies, 2007. ICALT 2007. Seventh IEEE International Conference. 2007:929 - 930

[2] Baeza - Yates R, Hurtado C, Mendoza M. et al. Modeling user search behavior // Comput. Sci. Dept., Chile Univ., Chile, IEEE Web Congress, 2005. LA-Web 2005. Third Latin American

[3] Tung A K H, Lu Hongjun, Han Jiawei, et al. Efficient Mining of Intertransaction Association Rules. IEEE Transactions on Knowledge An Data Engineering, 2003, 15(1)

[4] Chen Jian, Yin Jian, Tung A K H, et al. Discovering Web Usage Patterns By Minig Cross-transaction Association Rules // Proceedings of the third international conference on machine learning and cybernetics. Shanghai, 2004:26-29

[5] Yang Wanzhong, Li Yuefeng, Xu Yue. Granule Based Inter - transaction Association Rule Mining. 2007 IEEE, DOI 10. 1109/ICTAI. 2007, 143

[6] Burdick D, Calimlim M, Gehrke J. MAFIA: A Maximal Frequent Itemset Algorithm for Transactional Databases. 1063-6382/01 2001 IEEE

[7] Buzikashvili N. Sliding Window Technique for the Web Log Analysis. WWW 2007. May, Banff, Alberta, Canada. ACM 978-1 —59593-654-7/07/0005

[8] Tanasa D, Trousse B. Advanced Data Preprocessing for Intersites Web Usage Mining. Intelligent Systems, IEEE 2004, 19 (2):59-65

[9] 张波，巫莉莉，周敏. 基于 Web 使用挖掘的用户行为分析. 计算机科学，2006，33(8)

[10] Agrawal R, Srikant R. Fast algorithms for Minging Association Rules // Proceedings of the 20th VLDB Conference. Santiago, Chile, 1994

(上接第 94 页)

建了一个仿真的视频点播系统，验证了两种部署算法的优劣，但本文的仿真平台只针对一天内的用户点播行为，下一步可继续研究长时间内用户点播行为的仿真。

参 考 文 献

[1] Tangy W, Fuz Y, Cherkasovay L. Amin Vahdatz; Long-term Streaming Media Server Workload Analysis and Modeling[J]// Proceedings of the 13th International Workshop on Network and Operating Systems Support for Digit. June 2003

[2] Cherkasova L, Gupta M. Analysis of enterprise media server workloads: access patterns, locality, content evolution, and rates of change [J]. IEEE/ACM Transactions on Networking (TON), 2004, 12(5)

[3] Vilas M, Paneda X G, Garcia R, et al. User behaviour analysis of a video-ondemandservice with a wide variety of subjects and-lengths[J]//EUROMICRO. IEEE Computer Society, 2005

[4] Yu Hongliang, Zheng Dongdong, Zhao B Y, et al. Understanding user behavior in large-scale video-on-demand systems[J]//ACM SIGOPS Operating Systems Review Proceedings of the 2006 EuroSys Conference EuroSys '06. April 2006

[5] Johnsen F T, HafsØe T, Griwodz C. Analysis of Server Workload and Client Interactionin a News-on-Demand Streaming System[J]//IEEE ISM. San Diego, CA, USA, December 2006

[6] Frank J T, Trude H, Carsten G, et al. Workload Characterization for News-on-Demand Streaming Services [J] // Performance, Computing, and Communications Conference, IPCCC 2007. IEEE Internationa, April 2007