Software Engineering and Applications 软件工程与应用, 2013, 2, 125-130
http://dx.doi.org/10.12677/sea.2013.26022 Published Online December 2013 (http://www.hanspub.org/journal/sea.html)

Application of Apriori Algorithm in Finding User's Webpage Browsing Mode*

Lin Wei[1#], Jianyi Liu[2], Cong Wang[2]

[1]Key Laboratory of Trustworthy Distributed Computing and Service (BUPT), School of Computer Science, Beijing University of Posts and Telecommunications, Beijing
[2]School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing
Email: #weilin_nl@163.com, liujy@bupt.edu.cn, wangc@bupt.edu.cn

Received: Nov. 3rd, 2013; revised: Nov. 26th, 2013; accepted: Dec. 5th, 2013

Abstract: The log file of web server which recorded a large number of user's visiting webpage information, and how to analyze these data and discover the user's webpage browsing mode such as the webpages which users' interested in browsing and the best page composition so as to provide a good decision support for merchants has become increasingly important. In this paper, Apriori algorithm was used to mine the log data of recording use's accessing information for finding the regular pattern of user's browsing the webpage. Firstly, this paper made data preprocessing to the log data for extracting one session access record of user. Secondly, the Apriori algorithm was used to mine these record data, considering the feature of these data, the paper made litter improvement for the algorithm at the matching of k-candidate set and the transaction. The experimental results showed that the performance of the improved algorithm in handling a large amount of data has a good improvement. Finally, this paper analysed the rules by excavating, and through these rules, some browsing modes were found, which provided decision supports for merchants.

Keywords: Web Logs; Apriori Algorithm; Data Mining; Session Identification; k-Candidate Set

Apriori 算法在发现用户网页浏览模式上的应用*

魏　林[1#]，刘建毅[2]，王　枞[2]

[1]北京邮电大学计算机学院，可信分布式计算与服务教育部重点实验室，北京
[2]北京邮电大学软件学院，北京
Email: #weilin_nl@163.com, liujy@bupt.edu.cn, wangc@bupt.edu.cn

收稿日期：2013 年 11 月 3 日；修回日期：2013 年 11 月 26 日；录用日期：2013 年 12 月 5 日

摘　要：web 服务器的日志文件记录了大量的用户网页访问信息，如何分析这些数据并从中发现用户的网页浏览模式比如用户感兴趣的页面、最佳的页面组合等从而为商家提供良好的决策支持变得越来越重要。本文用数据挖掘技术中的 Apriori 算法对记录用户页面访问信息的日志数据进行挖掘从而得到用户浏览网页的模式。本文首先对日志数据进行了预处理，从中提取了用户的一次会话中的页面访问记录，然后用 Apriori 算法对这些访问记录数据进行挖掘，同时针对这些待挖掘数据上的特点对挖掘算法 Apriori 在 k-项候选集与事务的匹配上进行了改进，实验结果表明改进后的算法在处理数据量很大的数据时性能较传统算法有很好的提高。最后本文对挖掘后产生的规则进行了分析，发现了用户对本网站的一些网页的浏览模式，这些浏览模式为商家提供良好的决策支持。

关键词：Web 日志；Apriori 算法；Web 日志挖掘；会话识别；k-项候选集

*资助信息：本项目是由市委、市政府资助，特表示感谢。
#通讯作者。

1. 引言

目前 web 流量已成为 Internet 上信息传输的主流，全国的网页数和网页字节总数呈暴增的趋势。其中大量的访问信息都被记录在了 web 服务器的日志文件中，因此如何分析这些数据并从中提取想要的结果从而为商家提供良好的决策支持变得越来越重要。而目前的大型数据挖掘系统比如 Intelligent Miner，SPSS，DBMiner[1]等，功能布局相对不合理，当实现某些行业的某些特定目的的数据挖掘时并没有表现出特有的性质。

如何从这些访问信息中发现用户的浏览模式比如用户感兴趣的页面、最佳的页面组合等从而吸引更多的用户来访问和了解相关网站，涉及到用户浏览模式的分析，其中 Web 日志挖掘可以用来发现用户的浏览模式，进而可用于改进 Web 服务器的设计以方便用户使用以及增加个性化服务等。目前用于 Web 日志挖掘的算法很多，文献[2]用 PrefixSpan 算法来对 Web 日志进行分析，挖掘出用户访问的频繁序列模式，并以此构造测试负载。文献[3]用 k-means 算法对所提取的 IP 和域名的时间行为特征矢量进行聚类从而找到真正体现绝大多数用户网络访问需求的域名集合。文献[4]通过利用粗糙集理论对网站日志数据进行分析从而进行入侵检测。本文以某测评中心网络的日志分析为例，用 Apriori 算法[5]对网站日志进行挖掘并分析以得到用户对页面的浏览行为，发现他们所感兴趣的页面组合，据此来改善 Web 的信息服务，从而有效地提高网站的效益。在实现过程中首先对日志数据进行了预处理以提取用户的一次访问页面记录以作为实验挖掘数据，然后针对待挖掘数据的特点对 Apriori 算法做了一些改进，对改进的算法的时间复杂度较之前的算法进行了分析与比较，并结合实验结果对改进算法的性能进行了验证。最后对挖掘的结果进行了分析发现了用户浏览网页的一些模式。这些浏览模式为商家提供良好的决策支持。

2. 日志数据的预处理

2.1. 日志原理分析

Web 日志记录了网站服务器接收、处理请求以及运行错误等各种原始信息，如图 1 所示。通过对日志进行统计、分析和综合，不仅能有效地掌握服务器地

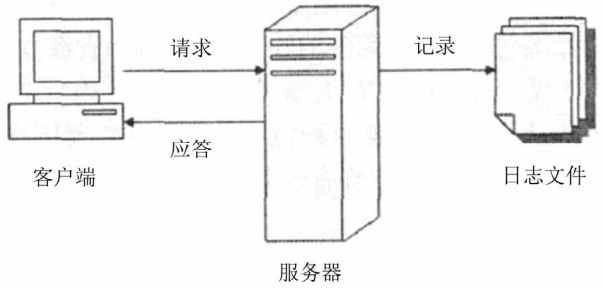

Figure 1. The web service model
图 1. Web 服务模式

运行状况，发现和排除错误原因，而且还可以了解用户的访问特性和分布等规律，从而可以更好地加强系统功能。其中，Web 服务模式主要有 3 个步骤：

1) 服务请求：用户通过浏览器向 Web 服务器发送请求(如 Get)，根据 HTTP 协议，请求包含了用户端的众多基本信息，如 IP 地址、浏览器类型、目标 URL 等等。

2) 服务响应：Web 服务器接收到请求后，按照用户的要求运行相应的功能，并将信息返回给用户。如果出现错误，将返回错误代码。

3) 追加日志：服务器将对用户访问过程中的相关信息以追加的方式保存到日志文件中。

2.2. 日志数据字段的提取

对原始数据进行提取[6]并分析从中得到我们想要的数据是挖掘前的一个必要环节。本文所用实验数据是某网站服务器中页面访问的日志文件数据。以下是本文处理的日志数据格式：

218.161.64.101—[22/Aug/2011:09:51:46+0800]“GET/?p=eHTTP/1.1”2006326“http://www.kehou.com”“Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)”

这行日志数据可以解释为：来自“http://www.kehou.com”的访客，使用 IE6.0 浏览器，应用 HTTP/1.1 协议，在北京时间(+0800：北京时间与格林威治标准时间的差)22/Aug/2011:09:51:46，访问(GET)了 218.161.64.101 主机域名下面的/?p=e 页面，访问成功(返回了 200)，得到 6326 字节数据。本文利用 java 程序提取其中的访问时间字段和访问的 url 字段，并写入到 excel 表中。

2.3. 获取用户一次会话页面访问记录

会话识别[7]就是把用户的日志记录以会话为单位

进行划分。采用超时机制来区分用户的不同会话。设定一个时间阈值 T(通常设为 600 s)，如果用户连续两次请求之间的时间间隔超过 T，则认为用户开始了新的会话。本文设定一个起始时间变量 beginVistorTime 和一个当前用户访问的时间 currentVistorTime 外加一个当前访问 url，如果 currentVistorTime 减去 beginVistorTime 的值小于 600 s，认为是同一个用户正在会话，并记录访问的 url。若 currentVistorTime 减去 beginVistorTime 的值大于 600 s，则认为当前会话结束，用户开始了一个新的会话，并将 beginVistorTime 的值设为 current VistorTime，currentVistorTime 设置为当前获取的时间字段。

使用该方法可以记录用户一次会话的访问 url 记录，本文使用的 url 有七种类型，分别为 url_1、url_2、url_3、url_4、url_5、url_6、url_7 对应挖掘的七种属性。若用户在一次会话中访问了某个 url，则该 url 对应的访问值为 1，否则为 0。比如用户在一次会话中访问了 url_1、url_3、url_4，则对应的 url 的属性访问值为表 1 所示。

本文使用上述方法记录每一次用户会话中各个 url 属性的访问值，并作为待挖掘的实验数据。从而来发现用户浏览这些页面之间的关系。

3. 基于改进的 Apriori 算法的数据挖掘

数据挖掘模块中要实现关联规则的找寻，即找到相应的频繁集，数据挖掘引擎采用关联规则中的经典算法 Apriori 算法。本文针对待挖掘数据的特点对 Apirori 算法进行了改进。对改进后算法与原始算法的时间复杂度作了对比和分析，分析表明改进的算法在一定程度上减少了匹配开销，并在第四部分结合实验结果对此进行了验证。

3.1. 传统 Apriori 算法

Apriori 算法是一种寻找频繁项集的基本算法，其基本原理是使用一种称作逐层搜索的迭代方法，即用 k_项集去探索(k+1)_项集。Apriori 算法使用频繁项集性质的先验知识，首先找出频繁 1_项集的集合，该集合记作 L_1。L_1 用于找出频繁 2_项集的集合 L_2，而 L_2 用于找出 L_3，如此下去，直到不能找到频繁 k_项集。将 Apriori 性质用于算法中，由频繁项目集 L_{k-1} 生成候选项目集 C_k 的过程分两步进行：

Table 1. Session access record for Url of user
表 1. 用户的一次会话访问 Url 记录数据

URL_1	URL_2	URL_3	URL_4	URL_5	URL_6	URL_7
1	0	1	1	0	0	0

- 连接：为了找出 L_k，通过 L_{k-1} 与自己连接产生候选 k_项集的集合。该候选项集的集合记作 C_k。设 l_1 和 l_2 是 L_{k-1} 中的项集，记号 $l_i[j]$表示 l_i 的第 j 项。执行连接 $L_{k-1}\infty L_{k-1}$，其中 L_{k-1} 的元素是可连接的，如果它们的前 k-2 个项相同，那么连接 l_1 和 l_2 产生的结果项集为 $l_1[1]l_1[2]\cdots\cdots l_1[k-1]l_2[k-1]$。
- 剪枝步：扫描数据库，确定 C_k 中的每个候选项集的计数, 并与用户指定的最小支持数比较，大于最小支持数的项目集并入 k 项频繁集 L_k 中。然而，当 C_k 中的候选项集的数目很多时，涉及的计算量就很大。为了提高效率，可用 Apriori 性质压缩 C_k，即任何非频繁的-项集都不可能是频繁(k-1)-项集的子集，可得：如果一个候选 k-项集的(k-1)-子集不在 L_{k-1} 中，则该候选项集也不可能是频繁项集，便可将它从 C_k 中删除，这种子集测试可使用所有频繁项集的散列树快速完成。

输入：事务数据库 D，相关参数(项目数，事务数，最小支持度)。

输出：D 中频繁项集和对应的支持度

算法伪代码如下所示:

```
L1=hashtreeroot;//找出一项集(D)
for(k = 1;Lk-1≠Φ;k++)
{
transatrahashtree(root);//T 扫描事务数据库
checkcounter(root);
//较检 htn.counter>=minsup*总事务数，剪枝
Ck=checkhashtree(root);//生成候选项目集
checkcountedall(root);
//确认 itemset 已经被 count
}
printhashtree(root);
//输出 hashtree 上的所有频繁项集合项集的支持度
checkhashtree(root)
{
gensuperset(itemset);
//为项集生成超集,实现连接
```

```
gensubset(itemset);//生成子集
circlefound(root);
//根据 Apriori 性质,剪枝
}
```

3.2. Apriori 算法的不足

在用上述方法之后我们得到了可能成为频繁集的 k-项候选集 C_k，通过扫描数据库中的每个事务计算出候选项目集 C_k 中的每一个候选项集的支持数 counter。如果该事务包含该候选项集，则相应的 counter 加 1。假设匹配的时间为 m，数据库中的事务个数为 N，则每次计算候选项集的支持数的时间为复杂度 O(m*N)，可见该计算开销还是很大的，若能减少不必要的匹配，从而减少匹配的时间，就能减少一定计算开销。因此，针对上述问题以及结合本文数据特点，本文提出了一种基于数据特征的改进方法。

3.3. 基于数据特点的 Apriori 算法的改进

由于本文中的数据库中的事务中项的值只有 1 和 0 两种(1 表示访问了该项对应的页面，0 表示没有访问该项对应的页面)，而且本文研究的是用户访问了的页面之间的关系，比如用户访问了哪些页面之后会访问另一些页面。所以产生规则的 k-项频繁集应该是由表示访问了 k 个页面的事务组成的，而产生 k-项频繁集的 k-项候选集应与那些表示访问了 k 个页面的事务(项的值为 1 的个数至少为 k)进行匹配，从而来计算该候选集的支持数 counter。由于事务中的项的值为 1 或 0，所以可以求出该事务项的值的代数和 P，若该代数和小于 k，则表示该事务中访问的页面个数小于 k，说明该事务不能与该 k-项候选集匹配，这样就节省了不必要的匹配开销。

基于这种特点本文在 k-项候选项集与数据集中的事务进行匹配时提出了一种先验方法，即在计算某 k-项候选集的支持度计数扫描数据库事务时，可以求出该事务项的值的代数和 P，若 P 小于 k，跳过该事务往下处理其他事务，之后当再次寻找(k+1)-项集时，就不需要再扫描该事务了。

该改进方法从某种程度上减少了不必要的匹配，而数据集中的事务是以数组的方式被加载进内存并进行匹配的，因为数组的访问和求和时间是很快的，假设数组求和的时间为 n，其中 n 应该远小于事务与候选集的匹配时间 m，假设某 k-项候选集在与数据集的事务进行匹配计数时，事务中的项的值的代数和 P 小于 k 的个数为 Z，总的事务数为 N，则该方法的计数所用时间为 T1 = Z*n + (N-Z)*m，而传统 Apriori 算法的计数时间为 T2 = N*m，那么改进后的算法节省时间为 Z(m-n)。从节省的时间中看出 Z 越大时节省时间越大，说明该方法在对大数据进行处理时将会有良好的表现。

本文算法的实现过程是在 weka[8]的基础上对其源代码中计算某一事务是否与 k-项候选集匹配过程做了改进，改进后的代码如下：

```
public boolean containedBy(Instance instance,int k)
//k 表示该事务是与 k-项候选集进行匹配
  {
if(sum(instance) < k) return false;
//求该项事务的代数和，若小于 k 则不匹配
for (int i = 0; i < instance.numAttributes(); i++)
{
if (m_items[i] > -1) // m_items 表示 k-项候选集
{
   if (instance.isMissing(i) ||(m_treatZeroAsMissing
&& (int)instance.value(i) == 0))
                return false;
      if (m_items[i] != (int)instance.value(i))
                return false;
}
}
}
```

代码的第一行是利用本文提出的改进方法添加的用来计算该项事务的代数和，若该代数和小于 k，那么与 k-项候选集不匹配，则跳过后面的匹配过程，从而减少了不必要的开销。

4. 实验结果与分析

4.1. 算法性能比较

本文在实验中使用了三天的日志数据，利用两种算法分别对其进行数据挖掘，其中三天的日志数据分别为第 1 天(380 条事务记录)、前 2 天(810 条事务日志)和 3 天(1280 条事务日志)。另外，实验中记录了用这三天的数据进行挖掘时两种算法挖掘时间并进行

了比较，如图 2 所示。

从图 2 可以看出，随着数据集中事务数的增加，优化后的算法在速度上体现的优势也逐渐增大，这跟理论的分析结果是一致的。当数据量越大时计算 k-项候选集的支持度节省的时间越大，从而体现了该方法对处理数据量很大的数据集时有很好的表现。

4.2. 规则获取结果

规则获取模块将数据挖掘引擎对日志数据进行挖掘时得到的频繁项集及其支持数转化成规则，产生强关联规则，以供网站设计者改善 Web 的信息服务时使用。置信度根据公式(1)计算而得，其中条件概率用项集支持度表示，公式中的 support_count(A∪B)是包含项集 A∪B 的事务数，support_count(A)是包含项集 A 的事务数，confidence(A=>B)表示 A 发生的情况下 B 也发生的概率，即 P(B|A)。

$$P(B|A) = \text{support\_count}(A \cup B) / (\text{support\_count}(A)) \quad (1)$$

根据(1)式，关联规则可以通过如下方法产生：对于每个频繁项集 l，产生 l 的所有的非空子集。对于 l 的每个非空子集 s，如果 P((l-s)|s)≥min_conf，则输出规则 s =>(l-s)。其中，min_conf 是最小置信度值，l-s 表示 l 中排除 s 的项构成的项集。

对该测评中心的日志进行数据挖掘后得到的频繁项集为 1 = {url_2,url_4,url_7}，它的非空子集有{url_2,url_4}，{url_2,url_7}，{url_4,url_7}，{url_2}，{url_4}，{url_7}。结合散类表中提供的支持数，可以得到的关联规则以及它们的置信度如下：

I url_2∧url_4→url_7, confidence=265/300=88.3%

II url_2∧url_7→url_4, confidence=265/285=93.0%

III url_4∧url_7→url_2, confidence=265/272=97.4%

IV url_2→url_4∧url_7, confidence=265/331=79.6%

V url_4→url_2∧url_7, confidence=265/333=79.5%

VI url_7→url_2∧url_4, confidence=265/371=71.4%

由于规则是由频繁项集产生的，每个规则都自动的满足最小支持度。现在它们的置信度如上所示，再综合 url_1~url_7 所代表的网页进行分析，如式 I 说明了同时访问 2，4 和 7 网页的用户占全部用户的 20%以上，访问 2 和 4 网页的用户有 86.3%的可能会访问 7 网页。

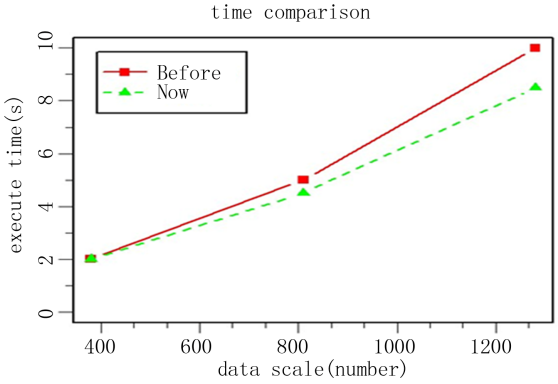

Figure 2. Performance Comparison between two methods
图 2. 改进前后算法的性能比较

这些产生的规则说明了用户对该网站的浏览模式，网页设计者可以根据这些规则来开发用户喜欢的网页，从而为商家带来更好的价值。

5. 结论

本文首先对网站日志数据进行了预处理，之后针对处理后的数据的特点对 Apriori 算法的候选集与事务的匹配过程提出了先验判断方法，并从时间复杂度方面分析了该方法减少不必要的匹配开销，同时用实验数据对该方法进行了验证。实验结果表明当日志数据很大时，该方法有着良好的表现。最后文章对由频繁集产生的规则进行了分析并从中找出了用户浏览页面的一些规则，为商家提供良好的商业决策提供了指导。

6. 致谢

本文的课题是市委、市政府重点工作及区县政府应急项目(基于大数据管理与智能分析的政务网站信息安全监控管理系统研发与应用)的一个日志分析部分，该项目受到各个单位的资助，特表示感谢。

参考文献 (References)

[1] 朱扬勇, 周欣, 施伯乐 (2000) 规则型数据采掘工具集 AMINER. *高技术通讯*, **3**, 19-22.

[2] 朱靖君, 吴海燕, 高国柱等 (2010) 一种基于日志分析的 Web 负我测试方法. *计算机工程*, **23**, 25-27.

[3] 季成, 李晓东, 袁坚等 (2010) 基于 k-means 算法的 DNS 查询模式分析. *清华大学学报: 自然科学版*, **4**, 601-604.

[4] 杨文兵 (2010) 基于 Rough 集理论的入侵检测方法研究. 硕士论文, 南昌大学, 南昌.

[5] 许晓东, 李柯, 朱士瑞 (2010) Web 使用挖掘中 Apriori 算法的改进研究. *计算机工程与设计*, **3**, 539-541.

[6] 李燕, 冯博琴, 鲁晓锋 (2009) Web 日志挖掘中的数据预处理技术. *计算机工程*, **22**, 44-46.
[7] 周爱武, 程博, 李孙长等 (2010) Web 日志挖掘中的会话识别方法. *计算机工程与设计*, **5**, 936-938.
[8] Hall, M., Frank, E., Holmes, G., et al. (2009) The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter*, **11**, 10-18.