计算机科学2000Vol 27№. 5

多周期经常性关联规则的发现

Discovery of All Frequent Cyclic Association Rules

黄益民
（西南师范大学计算机科学系　重庆 400715）

Abstract The problem of founding all frequent cyclic association rules, which are present in most but not the entire series of time units that have periodic time intervals, has been considered in this paper. Discovering these rules and their periodicity may reveal interesting information that can be used for predictions and making decisions. When the range of cycle lengths had been specified, the frequent association rules could been found by methods already existed, while this paper proposed an algorithm to reduce the unnecessary amount of work performed during the data mining process. Furthermore, we showed the effectiveness of this method through experiments.

Keywords Data mining, Association rule, Timestamped database

1 引言

近年来，现实生活中数据量在高速增长，而在数据中发现有效知识的技术却相对匮乏，因此数据挖掘这一领域成为大家注目的焦点。对事务数据库进行分析的一个十分重要内容是关联规则的发现。此问题被Rakesh Agrawal等[1]首先提出，尔后得到了广泛的研究，如文[2,4]考虑了发现关联规则的效率问题，文[3]考虑了增量发现问题，文[7]考虑了在时间序数据库中发现周期性模式等等，但以上的工作都是将数据库看成是一个整体，没有考虑时间段的问题。最近，B. Ozden等[5]研究了发现"完全的"周期性关联规则的问题，并提出了相应的解决方案。他们提出的概念叫做周期性规则，此种规则在指定时间间隔的周期性时间单元里具有很高的支持度和信度，但在整个数据库中的信度和支持度可能不高，在此基础上，文[8]提出了经常性周期关联规则的发现问题，给出了发现这种规则的多种不同方法。但在文[8]中，所有的问题都是在假定已知时间周期的情况下谈论的。

本文考虑了如何在时间周期不确定的情况下，发现所有经常性周期关联规则。与文[5]和[8]中一样，在本文中假定被处理的事务数据库是时间序的，用户指定的时间段长度将整个数据库分割成一个个的时间单元。同时，在本文中还假定进行数据挖掘时事务数据库是静态的。这样，在给定了时间单元间隔的情况下，时间单元的个数也就确定了，假如在一系列有一定周期间隔的时间单元内，一条关联规则的发生次数（即支持度和信度同时超过指定值的次数）达到用户指定的比率（最小频率信度），就称此规则是具有此周期的一条"经常性周期关联规则"。本文首先给出了问题的定义并考虑直接扩展已有的算法来解决问题。然后本文提出了一种改进方法，能减少数据挖掘的时间。最后，将给出改进方法的具体效果和结论。

2 问题定义

这里给出经常性周期关联规则的定义与文[8]一致，设 $I=\{i_1,i_2,i_3,\cdots,i_m\}$ 是一个项目(item)的集合。T 是一个事务的集合，每个 T 中的事务 S 都是一个项目子集，即 $S\subset I$。假定数据库 D 中的数据可以被划分成 n 个有时间标记的数据子集，每一个数据子集对应于某一时间单元内的事务。标记第 i 个时间时间单元为 t_i，$0<i<n+1$，t_i 相当于时间间隔$((i-1)\times t,i\times t)$，在这里 t 是时间的单位。

对每个时间单元 t_i，用 d_{ti} 标记在 t_i 内执行的事务的集合。称 d_{ti} 为时间单元 t_i 对应的数据集。因此，数据库 D 可以被表示为 d_{ti} 的序列：$D=\{d_{t1},d_{t2},d_{t3},\cdots,d_{tn}\}$，其中 d_{ti} 中的一条规则具有如此形式：$X\rightarrow Y$，$X\subset I$，$Y\subset I$，并且 $X\cap Y=\varnothing$。如果在 d_{ti} 中同时包含 X 和 Y 的事务的比率为 s，称规则 $X\rightarrow Y$ 在 d_{ti} 中的支持度为 $s\%$。如果 d_{ti} 中包含 X 的事务同时包含 Y 的比率为 c，称规则 $X\rightarrow Y$ 在 d_{ti} 中的信度为 $c\%$。

对于给定最小信度阈值 min_conf%和最小支持度阈值 min_sup%，当 d_{ti} 中包含 X 的事务不少于 min_sup%时，称 d_{ti} 中的一个项目集 X 是"大"的；如

果规则 $X\rightarrow Y$ 在 d_u 中的支持度和信度不小于 min_sup% 和 min_conf%，称 $X\rightarrow Y$ 是 d_u 中的一条关联规则。

用数对 (l,o) 来表示一个时间周期，l 是时间间隔的长度，o 是偏移量，即循环发生的第一个时间单元，$0\leqslant o<l$。对于给定的时间周期 (l,o)，如果对于时间单元 t_i，有 $i \bmod l=o$，则称时间单元 t_i 在周期 (l,o) 中，其对应的数据子集 d_u 也被称为是在周期 (l,o) 中的。将所有在周期 (l,o) 中的数据子集记为 $D(l,o)$，于是有：

$$D(l,o)=\{d_{tk\cdot l+o}\mid k\in Z,0\leqslant k<n/l\}$$

本文中，将记周期 (l,o) 中的第 k 个时间单元为 $d^k_{(l,o)}$。

例1 如果时间间隔是一星期，并且已收集16个星期的事务数据（$n=16$），每个星期分别用 $t_0,t_1,t_2,t_3,\cdots,t_{15}$ 表示，每个星期对应的事务数据集分别为 $d_{t0},d_{t1},\cdots,d_{t15}$，则数据集 $d_{t2},d_{t6},d_{t10},d_{t14}$ 在周期 $(4,2)$ 中，而 d_{t1},d_{t7},d_{t13} 在周期 $(6,1)$ 中。一条规则 $X\rightarrow Y$ 在周期 (l,o) 中的频率计数和频率信度分别被定义为：

频率计数 $(X\rightarrow Y)=|\{k\mid 0\leqslant k<n \bmod l$，并且 $X\rightarrow Y$ 是 $d^k_{(l,o)}$ 中的一条关联规则$\}|$

频率信度 $(X\rightarrow Y)=$频率计数$(X\rightarrow Y)/|\{i\mid 0\leqslant i<n$ 并且 $i \bmod l=o\}|$

例2 如果 $n=16$ 并且 r 是 $d_{t0},d_{t2},d_{t5},d_{t7},d_{t11},d_{t12}$ 中的关联规则，则它的频率计数在周期 $(4,0)$ 中是4（$d_{t0},d_{t2},d_{t7},d_{t11}$）。而它的频率信度在周期 $(4,0)$ 中是 $4/5=0.8$。

如果 l_i 能够整除 l_j 并且 $o_j=o_j \bmod l_i$，我们称一个周期 (l_i,o_i) 是另一个周期 (l_j,o_j) 的倍数。如果一条关联规则在周期 (l,o) 中的频率信度不小于一个指定的**阈值**——最小频率信度（min_freq_conf），则称此关联规则在周期 (l,o) 中是经常性的。如文[8]中指出的那样，为了使发现的规则有意义，最小频率信度阈值应该是一个较大的值。本文研究的问题就是在给定周期范围的情况下，找出所有经常的关联规则。

发现经常性周期关联规则的输入可被描述为：

·时间序事务数据库 D 及时间单位 t。

·指定的周期长度或由两个整型数据 l_min 和 l_max 指定的周期长度范围。

·三个指定的阈值 min_sup，min_conf，min_freq_conf。min_sup 和 min_conf 用于判定一条关联规则是否在 d_u 中，而 min_freq_conf 用于判断一条关联规则在指定的周期中是否是经常性的。

3 发现经常性周期关联规则的方法

在文[8]中提出了在指定的周期内发现经常性关联规则的方法，这里只简要介绍其结果，然后讨论周期在一定范围内变化时经常性周期关联规则的发现方法。

3.1 指定周期时发现经常性关联规则

在给定周期 (l,o) 时，发现经常性关联规则的最直接方法是：使用某种已有的算法，如文[2]，[4]，在每个时间单元中发现所有的关联规则，然后用模式匹配技术或统计方法发现经常性规则。在文[8]中使用经常性剪枝（frequency_pruning）技术并提出了多种经常性关联规则发现方法。其中最为有效的是以下两种：

方法1

1）首先找到 (l,o) 里前 $|D(l,o)|\times$min_freq_conf+1个数据子集中的所有关联规则，将这些规则及它们出现的次数存入一个经常性规则的候选集。

2）对于剩下的数据子集，检查候选集中的规则是否是在这些数据子集中，并进行经常性剪枝。

方法2

1）在 (l,o) 的前 $|d(l,o)|\times$min_freq_conf+1个数据子集中发现所有的"大"项目集，并将它们及它们的计数放入长度为 l 的经常性"大"项目集的候选集；

2）对于剩下的每一个数据子集，检查候选集中的项目集在它中间是否是"大"的，如果是，则此项目集在候选集中的计数增1；在一个数据子集扫描完成后，对候选集进行经常性剪枝。

3）在得到了 (l,o) 中所有长度为 $k-1$ 的经常性"大"项目集后，根据 Apriori 算法[6]产生所有长度为 k 的经常性"大"项目集的候选集。

4）对于 (l,o) 的前 $|d(l,o)|\times$min_freq_conf+1个数据子集，检查候选集中的项目集是否是"大"的，并对"大"的次数计数。然后，删除候选集中所有计数为0的项目集。

5）对于剩下的每一个数据子集，检查候选集中的项目集在它中间是否是"大"的，如果是，则此项目集在候选集中的计数增1；在一个数据子集扫描完成后，对候选集进行经常性剪枝。

6）重复步骤3到步骤5，直到某个经常性"大"项目集为空。

7）用得到的 (l,o) 中所有经常性"大"项目集来产生每个数据子集中的关联规则，然后使用统计方法并结合经常性剪枝来获得所有的经常性关联规则。

文[8]中的实验表明，方法1和2在运行时间上都远好于直接方法。在经常性"大"项目集的数目较少时，方法2优于方法1。

3.2 多周期情况下发现经常性规则

大多数情况下，人们喜欢发现具有某种自然周期的规律，因此发现指定周期的经常性关联规则在大多数情况下都是适用的。但是，有时规则的周期是不确定

的。因此，找一种高效的算法，能在只给出周期取值范围的情况下，找出所有的经常性周期规则是十分重要的。很明显，在已知周期取值范围的情况下，要求出所有的经常性周期关联规则的简单办法是对指定范围内的每一个周期都使用单周期的经常性周期关联规则发现算法。

方法3

对于给定的 min_sup，min_conf 和 min_freq_conf，用方法1或2，对每一个在范围 [l_min, l_max] 内的周期，在数据库 D 中发现具有此周期的经常性关联规则。

方法3为发现经常性周期关联规则提供了一条途径。但对每一个周期，都需要多次扫描数据库 D。实验表明，采用此方法所用的时间与周期的数目成正比。文[5]中，B. Ozden 等对多周期的情况进行过研究，但他们处理的是"完全性"的周期性关联规则，其结果不能照搬到条件较弱的"经常性"周期性关联规则中来。实际上，通过使用倍数周期中产生的经常性周期关联规则，可以减少算法的运行时间，为说明此方法，首先给出两个性质。

性质1 如果 a 是周期 (l_i, o_j) 中的经常性关联规则，而 l_i 是 l_j 的倍数，即 l_j 能够整除 l_i，则 a 至少在一个以下的周期中是经常性关联规则：

$(l_i, o_j), (l_j, o_j+l_j), (l_i, o_j+2\times l_j), \cdots, (l_i, o_j+m\times l_j)$，

其中 $m=l_i/l_j-1$

证明：由于是使用的一个静态的数据库 D 来产生经常性周期关联规则，在用户指定了时间间隔后，时间单元的划分是确定的。同时，由于对不同的周期，使用的是同样的 min_sup，min_conf 和 min_freq_conf，在 l_i 是 l_j 的倍数时，有：$D_{l_j,o_j}=\bigcup_{k=0}^{m} D_{l_i,o_j+k\times l_j}$ 根据抽屉原理，如果一条规则在任意的周期 $(l_i, o_j+k\times l_j)$ 中都不是经常性的，则它在 (l_j, o_j) 中也不可能是经常性的。

根据同样的原理，我们可以得到下面的结果：

性质2 如果 (l_j, o_j) 是一个周期，而 l_i 是 l_j 的倍数且 a 在所有以下的周期中都是经常性关联规则：$(l_i, o_i), (l_i, o_i+l_i), (l_i, o_i+2\times l_i), \cdots, (l_i, o_i+m\times l_i)$，$m=l_i/l_j-1$，则 a 是周期 (l_i, o_i) 中的经常性关联规则。

性质1说明，如果我们已发现了 $(l_i, o_i), (l_i, o_i+l_i), (l_i, o_i+2\times l_i), \cdots, (l_i, o_i+m\times l_i)$ 中的所有经常性关联规则，我们可以用它们来产生周期 (l_i, o_i) 中的候选经常性关联规则集。所有周期 (l_i, o_i) 中的经常性关联规则都应在候选集中出现。而性质2说明，如果一条规则在所有周期长度为 l_i，偏移量为 $o_i, o_i+l_i, o_i+2\times l_i, \cdots, o_i+m\times l_i$ 的周期中都是经常性的，则它必定在 (l_i, o_i) 中是经常性的；但对那些只在部分周期中是经常性的规则，我们不能断定其在 (l_i, o_i) 中是否是经常性的，必须扫描一遍数据库进行确认。

根据以上的性质，经常性关联规则发现算法被描述如下：

方法4 （经常性关联规则的发现算法）：

```
for k=l_max downto l_min do begin
    查找一个最小的 m，k<m≤l_max 并且 m mod k=0
    if (m 不存在)then
      for n1=0 to k-1 do
        使用方法1或2产生所有 D_{k,n1} 中的经常性关联规则
        并保存这些规则
    else begin
        l=m/k
        for n1=0 to k-1 do begin
            设置周期(k,n1)的候选经常性规则集为空
            for n2=0 to l do
                forall 属于 D_{m,n1+k*n2} 的经常性关联规则 a
do
                    if a 已在候选经常性关联规则集中
                    then
                      将 a 在候选经常性关联规则集中的计
                      数加1
                    else
                      将 a 加到候选经常性关联规则集中，
                      并设定其计数为1
            forall 在候选经常性规则集中的规则 a do
                if (a 的计数等于l)
                  此规则是 D_{k,n1} 中的经常性规则，存储此
                  规则并将其从候选集中删除
                  扫描数据库 D，检查所有在候选经常性
                  规则集中的规则 a，若 a 是 D_{k,n1} 中的经常
                  性规则，存储此规则
        end
    end
end
```

4 性能研究

与方法3相比较，方法4并不是对每一个周期都进行多次数据库扫描，在可能的情况下，它尽量使用以前扫描所得到的结果，以减少算法的运行时间。只要周期 k 的倍数 m 存在，使用处理周期 m 时得到的结果将远快于直接使用方法1和2。这是因为在此情况下，数据库 D 只需要被扫描一遍，就可以确定所有周期 m 中的经常性周期关联规则。

在大多是情况下，方法4的效果好于方法3，但是，在极端情况下（当周期的取值范围不大，没有周期的倍数也在取值范围内时），此方法可能退化成方法3。

注意算法4中，在处理周期 k 时，使用的是 k 的最小倍数 m。这是因为虽然使用 k 的任意一个倍数都可以得到一个候选经常性关联规则集，但是，实验表明，倍数越大，候选集的大小也会急剧加大，这将使扫描一遍数据库的时间大大加长。所以，算法中尽量选取 k 的较小倍数来产生候选集。

本文所用到的测试数据集是按照文[2]中所描述的数据产生算法得到的，为了提高"大"项目集的数目，测试数据生成算法按文[4]进行了修改。为了产生经常性关联规则，还加入了一个新的参数 u 来表示数据子集的相似性。 （下转第56页）

重的问题，因为随后进来的 BPA 完全有可能偏向原先个数很少的那些相反意见，但由于证据推理的"0绝对化"，这些相反意见完全被排斥，这种情况并不符合某些工程实际的要求，故不能简单地去掉矛盾证据。如果对这些矛盾证据做适当的、细微的调整，不但能纠正这些矛盾证据的 mass 分配的偏颇，而且也不同程度地吸取了不同的意见，这很符合人类的决策原则，即"求大同，存小异"。

结论 通过以上的分析，可以看出本文提出的解决证据推理中一类"0绝对化"问题的方法是可行的。由于 BPA 的给定带有不同程度的主观性，因而有必要在用 Dempster 合成规则进行证据合成之前，对某些 BPA 做适当的调整。本文为了计算的简化，一开始就将所有被合成的 BPA 都近似转化成贝叶斯 mass 函数，所以最终的信度测度和似然测度是相等的。这样处理是合理的，因为文[2]证明了当人们关心的是单个假设的最终结论时，用贝叶斯近似法能够得到满意的结论。其实如果一开始不进行贝叶斯近似，并不影响本文方法的效果，当然如果那样做的话，将保持证据推理所特有的表示和处理"不确定"和"不知道"的风范，只是随着假设空间的维数和被合成的 mass 函数个数的增多，其计算量将是非常巨大的。此外，也可以将 mass 函数值中原来 $m(\Theta)=0$的项赋与很小的值（相应地减少另一个数值很大的 BPA），以达到调整 BPA 的目的，这似乎更符合人的"常识"，更体现 BPA 分配的合理性，但在本文的方法中，即使这样做了，在随后的贝叶斯近似转换中还会将这个 $m(\Theta)$值"分配"到所有的单个假设的 BPA 中，因此不如本文的方法来得直截了当。

参 考 文 献

1 Shafer G. A Mathematical Theory of Evidence. Princeton University Press. Princeton, New Jersey, 1976

2 Voorbraak F. A Computationally Efficient Approximation of Dempster-Shafer Theory. Int. J. Man-Machine Studies, 1989(30). 525～536

3 Dillard R A. Using Data Quality Measures in Decision-Making Algorithms IEEE Expert, 1992(12): 63～72

4 张文修，梁怡. 不确定性推理原理. 西安交通大学出版社，1998. 10

（上接第52页）

实验结果表明，如果周期 k 的倍数 m 存在，使用方法4求周期 k 的经常性周期关联规则的时间将是直接对周期 k 使用方法2的24%～35%。在实际运用中，周期的取值范围和数据库 D 的具体特性对方法4的效率有很大影响。从文[8]可知，当周期 k 中的经常性周期关联规则较少时，方法1和方法2的运行时间是较小的。因此，在指定周期范围的情况下，只有存在经常性关联规则的周期会花费较多的运行时间。如果在指定的周期范围中，大多数周期中都只有少量经常性关联规则，方法4的优点将更为突出。

参 考 文 献

1 Agrawal R, et al. Mining Association Rules between Sets of Items in Large Databases. In: Proc. of the 1993 ACM SIGMOD Intl. Conf. on Management of Data. Washington, DC, 1993. 207～216

2 Areaway R, Syrian R. Fast Algorithms for Mining Association Rules. In: Proc. of the 20th Intl. Conf. on VLDB. Santiago, Chile, Sep. 1994. 487～499

3 Han Jiawei, Fu Yongjian. Discovery of Multi-level Association Rules From Large Databases. In: Proc of the 21st Intl. Conf. on VLDB. Zurich, Switzerland, Sep. 1995. 430～431

4 Park J S, et al An Effective Hash-based Algorithm for Mining Association Rules. In: Proc. of the 1995 ACM SIGMOD Intl. Conf. On Management of Data. May 1995. 175～185

5 O'zden B, et al. Cyclic Association Rules. In: Proc. of the 1998 Intl. Conf. On Data Engineering (ICDE'98) Orlando. FL, Feb. 1998. 412～421

6 Agrawal R, et al. Fast Discovery of Association Rules. In: Advances in Knowledge Discovery and Data Mining. AAAI Press, 1995. 1～36

7 Han Jiawei, et al. Mining Segment-Wise Periodic Patterns in Time-Related Databases. In, Proc. of 1998 Intl. Conf. on Knowledge Discovery and Data Mining (KDD'98). New York City, NY, Aug. 1998

8 黄益民. 经常性周期关联规则的研究. 计算机科学，2000，27(4)：50～53