数据库中加权关联规则的发现*

欧阳为民[1]， 郑 诚[2]， 蔡庆生[3]

[1](安徽大学 计算中心，安徽 合肥 230039)；

[2](安徽大学 计算机系，安徽 合肥 230039)；

[3](中国科学技术大学 计算机科学技术系，安徽 合肥 230027)

E-mail：oywm@mars.ahu.edu.cn

http://www.ahu.edu.cn

摘要：关联规则发现是数据库中知识发现研究中的热点课题，有着广泛的应用领域.在现有的研究中，数据库中的各个项目是按平等一致的方式加以处理的.然而，在现实世界数据库中却并非如此，不同的项目往往有着不同的重要性.为了将它们反映出来，对项目引入权值，从而提出了新的加权关联规则问题.由于项目权值的引入，频繁项目集的子集不再一定是频繁的.为此，又提出了项目的k-支持期望概念，并由此提出了加权关联规则的发现算法.

关键词：数据发掘；知识发现；加权关联规则

中图法分类号：TP311 **文献标识码**：A

近年来，数据库中的知识发现(knowledge discovery in databases，简称KDD)，也称数据发掘(data mining)，受到当今国际人工智能与数据库界的广泛重视[1].关联规则是KDD研究中的一个重要的研究课题.该问题是由R.Agrawal等人提出的，目的是要在交易数据库中发现各项目之间的关系[2,3].例如，有这样一条关联规则：黄油，牛奶⇒面包(30%和2%).其含义是购买了黄油和牛奶的顾客还将购买面包，30%和2%分别是该规则的信任度和支持度.在关联规则发现研究中最著名的算法是R.Agrawal等人提出的Apriori算法.该算法将关联规则的发现分为两步.第1步是识别所有的频繁项目集(frequent itemset)，即其支持不低于用户最低支持(minimum support)的项目集.第2步是从频繁集中构造其信任不低于用户最低信任(minimum confidence)的规则.其他大多数算法都是在该算法的基础上加以改进或扩展的，基本框架没有变化.

该算法实际上存在两大前提假设：(1) 数据库中各项目相同的性质和作用，即重要性相同；(2) 数据库中各项目的分布是均匀的.即出现频率相同或相似.也就是说，在该算法框架下，数据库中的各个项目以平等一致的方式处理.然而，在现实世界数据库中却往往并非如此.当数据库中项目分布不均匀出现频率相差较大时，就会导致最低支持设高设低都有问题的两难局面，如果设高了，所发现的关联规则将可能涉及不到出现频率较低的项目；而若设低了，就会发现太多的没有意义的甚至是虚假的关联规则，还有可能导致组合爆炸，从而降低算法效率直至不可行[1].近年来，对这一问题，国际上已有若干研究工作[4~7].

然而，对于前一个问题，目前国内外尚未有相关文献.事实上，不同的项目往往有着不同的重要

* 收稿日期：1999-12-06；修改日期：2000-01-20

基金项目：国家自然科学基金资助项目(69975001)

作者简介：欧阳为民(1964－)，男，安徽芜湖人，博士，教授，主要研究领域为知识发现，机器学习，人工智能及其应用；郑诚(1964－)，男，安徽屯溪人，副教授，主要研究领域为知识发现，机器学习，人工智能及其应用；蔡庆生(1938－)，男，重庆人，教授，博士生导师，主要研究领域为机器学习，知识发现，协调智能.

性.这几乎是现实世界数据库的内在特征.比如,项目的利润不尽相同,有的项目可能正处于促销中等等.为了反映各个项目的不同重要性,我们引入了项目权值概念,从而扩展了现有的问题模型,提出了新的所谓加权关联规则问题.为了在数据库中发现加权关联规则,我们又提出项目的 k-支持期望概念,并以此为基础提出了加权关联规则的发现算法.

当我们计算加权关联规则的支持度时,既要考虑规则中所有项目在数据库中同时出现的频率,也要考虑所有项目的加权值.

一个简单的办法是忽略权值较小的项目.但是,有的规则在与高权值项目相关的同时,很可能也与低权值项目相关.例如,在促销商品 A 时,我们可能发现商品 A 销售受到商品 B 的影响,即有规则 $B\Rightarrow A$,而商品 B 最初由于我们不感兴趣而被赋予较低的权值.如果我们因权值较低而忽略了商品 B,那么规则 $B\Rightarrow A$ 就不可能发现.因此,该方法在这种情况下是不可取的.

另一种方法是直接采纳现有的关联规则发现算法,如 Apriori 算法.这些算法均基于所谓的向下闭包性质,即频繁项目集的任一子集必是频繁的.然而,在加权关联规则模型中,该性质不再成立.因此,Apriori 系列算法不能直接加以应用.为此,我们提出项目的 k-支持期望概念,并以此为基础提出了加权关联规则的发现算法.

本文第 1 节给出加权关联规则模型.第 2 节描述加权关联规则发现算法.第 3 节是算法的性能测试.第 4 节对加权关联规则模型作简单讨论.最后是结论与进一步的研究方向.

1 加权关联规则模型

与文献[3]类似,我们考察交易数据库 D,其项目的集合为 $I=\{i_1,i_2,\ldots,i_n\}$,每一笔交易都是 I 的一个子集,并赋以一个交易标识符 TID.

定义 1. 关联规则形如 $X\Rightarrow Y$,其中 $X\subset I$,$Y\subset I$,且 $X\cap Y=\varnothing$.

定义 2. 关联规则形如 $X\Rightarrow Y$ 的支持度为 $X\cup Y$ 在交易数据库包含的概率.

定义 3. 关联规则形如 $X\Rightarrow Y$ 的信任度为在某交易中包含 X 的前提下同时也包含 Y 的概率.

换一种更加通俗的说法就是,关联规则形如 $X\Rightarrow Y$ 的支持度为数据库中包含 $X\cup Y$ 的交易数与总交易数之比;关联规则形如 $X\Rightarrow Y$ 的信任度为数据库中包含 $X\cup Y$ 的交易数与包含 X 的交易数之比.

给定项目集合 $I=\{i_1,i_2,\ldots,i_n\}$,为表征项目的重要性,我们为每一个项目 i_j 赋以权值 w_j,其中 $0\leqslant w_j\leqslant 1$,$j=\{1,2,\ldots,n\}$.

仿照定义 2,我们可以为加权关联规则定义加权支持.

定义 4. 关联规则形如 $X\Rightarrow Y$ 的加权支持(weighted support)为

$$\left(\sum_{i_j\in X\cup Y}w_j\right)(\text{support}(X\cup Y)).$$

定义 5. 某 k-项目集被称为频繁项目集,如果其加权支持不低于最低加权支持阈值 w min sup,即

$$\left(\sum_{i_j\in X\cup Y}w_j\right)(\text{support}(X\cup Y))\geqslant w\ \text{min sup}.$$

定义 6. 关联规则 $X\Rightarrow Y$ 是令人感兴趣的,如果 $X\cup Y$ 是频繁项目集,并且其信任度不低于最低信任阈值 min conf.

例 1:设有如表 1 和表 2 所示的数据库.表 1 表示各项目的信息,如条形码、商品名、利润、权值等等;表 2 是交易数据库,对每一笔交易都有一个交易表示符 TID 以及所购各商品的条形码.为简

单起见,条形码用自然数来表示.

Table 1 Items databases

表 1 商品数据库

Bar code①	Item name②	Total profit③	Weight④
1	Apple	100	0.1
2	Orange	300	0.3
3	Banana	400	0.4
4	Milk	800	0.8
5	Coca-Cola	900	0.9

①条形码,②商品名,③总利润,④权值.

Table 2 Transaction databases

表 2 交易数据库

TID①	Itemset②
1	1,2,4,5
2	1,4,5
3	2,4,5
4	1,2,4,5
5	1,3,5
6	2,4,5
7	2,3,4,5

①交易标识符,②项目集.

假定交易数据库中共涉及 5 种商品、7 笔交易. 如果 w min sup 为 0.4,那么{2,5}就是频繁项目集,因为

$$(0.3+0.9)\times\frac{5}{7}=0.86>0.4.$$

同理,{4,5},{2,4,5}也是频繁的.

2 加权关联规则的发现

由于加权关联规则本身所固有的特性,我们需要有新的发现算法. 在关联规则发现研究中,以文献[3]提出的 Apriori 算法最为典型,其他算法虽有一些不同的变化或扩展,但其基本思想是一致的,即均基于同一个结论:频繁项目集的任一子集必是频繁的. 然而,在加权关联规则这一问题模型下,为了处理体现项目重要性的权值,提出了项目集的加权支持概念,频繁项目集的含义由此便发生了变化,因而频繁项目集的子集就未必是频繁的了. 例如,在例 1 中,{2,4,5}是频繁项目集,但其子集{2,4}却不是频繁的.

2.1 k-支持期望

给定一个交易数据库,其交易总数设为 T. 对任一 k-项目集 X,其支持数(support count)为交易数据库中包含 X 的交易的个数,记为 $SC(X)$. 如果某 k-项目集 X 是频繁的,那么其支持数 $SC(X)$应满足下式:

$$SC(X)\geqslant\frac{w\ \min\ \sup\times T}{\sum_{i_j\in X}w_j}.$$

令 I 为所有项目的集合. 假定 Y 为一个 q-项目集,$q<k$. 在剩余项目集合$(I-Y)$中,记前$(k-q)$个权值最大的项目为 $i_{r_1},i_{r_2},\ldots,i_{r_{k-q}}$,那么包含项目集 Y 的任一 k-项目集的最大可能值为

$$W(Y,k)=\sum_{i_j\in Y}w_j-\sum_{j=1}^{k-q}w_{r_j}.$$

其中,第1个和式为q-项目集Y中各项目的权值之和,第2个和式为剩余的前$(k-q)$个最大权值之和.

结合上述两式,我们可以推知:如果包含Y的k-项目集是频繁的,那么其最低支持数应为

$$B(Y,k)=\left\lceil\frac{w\ \min\ \sup\times T}{W(Y,k)}\right\rceil.$$

我们称该$B(Y,k)$为Y的k-支持期望.考虑到$B(Y,k)$应取整数,为了保证包含Y的k-项目集有可能是频繁的,我们向上取整,而不是向下取整.否则,会由于$SC(Y,k)$的值过低而不足以使k-项目集成为频繁项目集.

例2:仍然沿用表1和表2.$\{2,4\}$的3-支持期望为

$$\left\lceil\frac{0.4\times 7}{(0.3+0.8)+0.9}\right\rceil=2.$$

这就是说,如果项目集$\{2,4\}$是某频繁3-项目集的子集,那么该频繁3-项目集的支持数应不低于2.

本文提出的加权关联规则发现算法就是建立在所有可能的k-项目集的支持期望的基础之上的.

2.2 加权关联规则发现算法

我们在Apriori算法的基础上,结合加权关联规则问题模型的特性,提出了发现加权关联规则的算法.为方便起见,我们引入一些记号,见表3.

Table 3 Symbol across table

表3 代号对照表

Symbol①	Meanings②
D	Transaction databases③
W	Set of item weights④
L_k	Set of frequent k-itemsets⑤
C_k	Set of k-itemsets which maybe k-subsets of frequent j-itemsets for $j\leqslant k$⑥
$SC(X)$	Support count of itemset X⑦
w min sup	Weighted support threshold⑧
min conf	Minimum confidence threshold⑨

①代号,②含义,③交易数据库,④项目权值的集合,⑤频繁k-项目集,⑥频繁j-项目集的可能的频繁k-项目子集的集合,⑦项目集X的支持数,⑧最低加权支持阈值,⑨最低信任阈值.

下面,我们首先给出算法描述,然后给出一个实例以作进一步的说明.

Algorithm. Discovery of Weighted Association Rules DWAR.

Input: (1) A transaction databases D, in which each item i_j has its weight w_j;
(2) Two threshold values w min sup and min conf.

Output: Weighted Association Rules.

```
Begin
(1) size=Scan (D);
(2) L=∅;
(3) for (i=1; i≤size; i++) {
(4)     C_i=∅;L_i=∅;
(5)     }
(6) for each transaction do
(7)   (SC,C_1)=Count(D,W);
(8) for (k=2; k≤size; k++) {
```

```
(9)  C_k=Join (C_{k-1});
(10)   C_k=Prune (C_k);
(11)   (C_k,L_k)=Check (C_k,D);
(12) L=L∪L_k;
(13) }
(14) Rules Set=Rules Gen(L);
End.
```

说明:

(1) Scan(D):该子程序交易数据库 D 为处理对象,发现其中频繁项目集的最大可能长度,并返回该数值.

(2) Count(D,W):该子程序累计 1-项目集的支持数,计算每个 1-项目集的 k-支持期望.然后收集其支持数不低于 k 支持期望的 1-项目集,形成 C_1.

(3) Join(C_{k-1}):根据 C_{k-1} 生成 C_k 的链接方法与文献[4]的 Apriori_Gen 相同.例如,如果 C_3 中有{1,2,3}和{1,2,4},那么将链接生成{1,2,3,4}.

(4) Prune(C_k):项目集的修剪方法如下:

(a) C_k 中候选项目集的子集不在 C_{k-1} 中;

(b) 估计候选 k-项目集 X 的支持数($SC(X)$)的上界,它是 C_{k-1} 中 k 个不同的($k-1$)-项目子集中的最低支持数.根据已计算出的所有项目集的 k-支持期望,如果对支持数 $SC(X)$估计出的上界表明项目集 X 在后继遍历中不可能成为任何频繁项目集的子集,那么该项目集 X 就可以被修剪掉.

(5) Check(C_k,D):该检查子程序遍历交易数据库 D,更新 C_k 中所有候选项目集的支持计数.通过类似修剪步骤的方法,删除那些不满足所有可能频繁项目集支持期望的候选项目集.剩余的候选项目集均保存在 C_k 中.然后,再检查各项目集的加权支持,从中挑选出频繁 k-项目集 L_k.

(6) Rules_Gen(L):与文献[4]相同,根据 L 中的频繁项目集生成符合最低信任阈值的关联规则.

该加权关联规则发现算法的框架与 Apriori 算法相似,但在部分具体细节上有着明显的不同.首先,虽然也是按照项目集大小以递增方式生成频繁项目集,但是,由于在加权关联规则问题模型下,频繁项目集的子集未必是频繁的,我们就不能像 Apriori 算法那样仅由($k-1$)-频繁项目集简单地生成候选 k-项目集.为此,我们设法另行寻找在后继遍历中有可能生成频繁 k-项目集的 j-项目集($k\leqslant j$).为了从数据库中提取这种 k-项目集,我们利用 j-支持期望.j-支持期望是根据所有候选 k-项目集计算得到的.j 介于 k 和频繁项目集的最大可能长度之间.如果某 k-项目集的支持数低于所有的 j-支持期望,我们就可以断定在后继遍历中它不可能是任何频繁项目集的子集,因而可以删除.所有可能成为频繁项目集的子集的 k-项目集构成候选集 C_k.

例 3:利用表 2 和表 3,我们举例说明如何从交易数据库中生成频繁项目集.假定加权最低支持阈值 w min sup 为 1.

(1) 在搜索处理过程中,该算法将仅收集每个交易的长度,取其最大长度作为频繁项目集的最大可能长度.在本例中,该值为 4.

(2) 在第 1 次遍历中,$k=1$,k 为项目集的长度.在计算项目集支持计数时要对交易数据库作一次遍历.该阶段将计算出 1-项目集的支持数.对每个 1-项目集,根据项目的支持数和权值,我们为所有可能的后继遍历计算支持期望.在本例中,1-项目集{1,2,3,4,5}中各项目的支持数分别为{4,5,2,6,7}.计算包含{1,2,3,4,5}中各项目的 k-项目集的支持期望为

$$B(\{1\},2)=\text{upper}(1\times 7/(0.1+0.9))-7,$$
$$B(\{1\},3)=\text{upper}(1\times 7/(0.1+(0.8+0.9)))=4,$$
$$B(\{1\},4)=\text{upper}(1\times 7/(0.4+(0.8+0.9)))-4.$$

其他项目的k-支持期望的计算类似进行.

项目{1}的支持为4,这意味着它可能是频繁3或4项目集的子集,于是我们应在C_1中保留{1}.而对于项目{3},由于其所有k-支持期望均大于其本身的支持数,所以{3}不可能成为后继频繁项目集的子集,从而C_1中不应保留,应予删除.类似地,可以推知,$C_1=\{\{1\},\{2\},\{4\},\{5\}\}$.

按照类似的方法可以生成所有其他候选和频繁项目集.

(3) 第2次遍历.

在链接步,该算法将生成如下可能的频繁项目集{{1,2},{1,4},{1,5},{2,4},{2,5},{4,5}}.

在修剪步,仿照前面的方法计算上述各项目集的支持期望,结果分别为{4,4,4,5,5,6}.由于这些2项目集在后继遍历中均有可能是频繁的,故均保留在C_2中.

在检查步,实际计算各2-项目集的支持数,结果分别为{2,3,4,5,5,6}.根据修剪步计算出的支持期望,{1,2}在当前和后继遍历中都不可能是频繁项目集,因为它的支持数低于其支持期望.这样,该项目集{1,2}就可以删除.

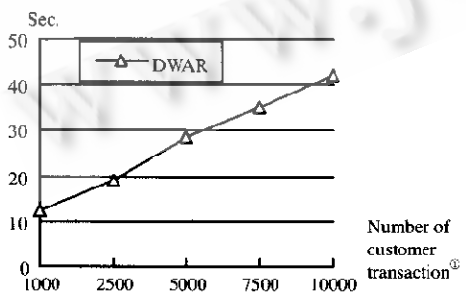
计算所有剩下的候选2-项目集{{1,4},{1,5},{2,4},{2,5},{4,5}}的加权支持,我们发现{4,5}是频繁的,因而将{4,5}加入L_2.C_2中的项目集在下一次遍历中将会再次用到.

(4) 从$L=L\bigcup L_k$中生成关联规则.方法与Apriori算法是一样的.

3 算法性能评测

我们用Visual Foxpro在内存为64M的Pentium II 400微机上实现了算法DWAR(discovery of weighted association rules),采用合成数据进行算法性能测试.合成数据的生成方法与文献[4]类似,在生成各项目的权值时按指数分布随机生成.测试数据库含有400种项目.首先测试算法的扩放性.我们将测试数据库的顾客交易数从1000开始,逐次递增到10000.最低支持设为2%.该算法的扩放性能数据曲线如图1所示.

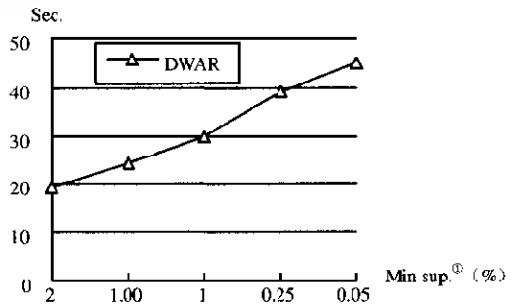
接着,我们测试算法在不同最低支持下的性能变化.此时,测试数据库固定为5 000元组,分如下5次变化最低支持,即$S_1(2\%)$,$S_2(1\%)$,$S_3(0.5\%)$,$S_4(0.25\%)$和$S_5(0.05\%)$.测试结果如图2所示.可以看出,当支持下降时,执行时间上升,原因是过滤条件减弱了.



①顾客交易数.

Fig. 1

图1



①最低支持度.

Fig. 2

图2

4 关于加权关联规则模型的讨论

在加权关联规则模型中,我们应设法在权值和支持度之间取得恰当的平衡.实际上,我们有项目权值、项目集支持度和信任度这3种参数来对加权关联规则进行评价.在本文中,我们引入项目集的加权支持概念.将加权支持定义为项目集中每个项目的权值之和与该项目集支持度的乘积.

如果将支持度与权值分开考虑,那么我们将会发现具有足够支持度和权值的项目集.然而,这样也许会遗漏某些令人感兴趣的模式.权值是项目或项目集重要性的度量.如果某项目集非常重要,比如说正处于促销过程中或利润很高,即使没有很多顾客购买,对用户来说它们仍然是令人感兴趣的.另一方面,如果某项目集从权值上考虑不是非常重要,但它却十分流行,有大量顾客购买,那么它也应是令人感兴趣的项目集.

另外一种似乎可选的方法是发现具有足够支持度或足够权值的项目集.不过,这样一来,我们就不能有效地处理权值为0的项目了.

本文所提方法是比较合理的,不过也存在一个小问题.如果某项目集中项目的个数较多,即使每个项目的权值较低,其各权值之和却也有可能是高的.这种情形有时也存在一定的问题,这取决于具体的应用.如果以一个整体来看待项目集的总权值,那么这种情形就没什么问题.但是,如果我们认为带有较多低权值项目的规则不是令人感兴趣的,那么这种情形就有问题了.

5 结论与进一步的工作

本文根据现实世界数据库中不同的项目往往有着不同重要性的实际情况,对关联规则发现问题进行了推广,提出了加权关联规则发现问题.为反映各个项目的不同的重要性,我们对项目引入了权值.由于项目权值的引入,频繁项目集的子集不再一定是频繁的.为此,我们提出了项目的k-支持期望概念,并由此提出了加权关联规则的发现算法DWAR.实验结果表明,DWAR算法性能良好,具有较好的可扩放性.

本文目前的研究仅涉及单层次概念,没有引入多层次概念.作为进一步的研究方向,我们将研究多概念层次的加权关联规则的发现问题.另外,将对项目加权的思想应用到定量关联规则之中的发现也是一个值得研究的问题.

致谢 本文的工作得到了国际KDD研究知名学者、美国IBM Almaden Research Center的Rakesh Agrawal教授和加拿大Simon Fraster大学Han Jiawei教授的支持.他们为笔者提供了有关的研究资料,特此深表感谢.

References:

[1] Ou-Yang, Wei-min, Cai, Qing-sheng. Researches on discovery of association rules. Computer Science, 1999,3:41~44 (in Chinese).

[2] Agrawal, R., Imielinski, T., Swami, A. Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data. Washington, DC, 1993. 207~216.

[3] Agrawal, R., Srikant, R. Fast algorithm for mining association rules. In: Proceedings of the 1994 International Conference on Very Large Data Bases. Santiago, Chile, 1994. 487~499.

[4] Brin, S., Motwani, R., Ullman, J., *et al*. Dynamic itemset counting and implication rules for market basket data. In: Proceedings of the International Conference on Management of Data. 1997. 255~264.

[5] Brin, S., Motwani, R., Silverstein, C. Beyond market baskets: generalizing association rules to correlations. In:

Proceedings of the ACM SIGMOD International Conference on Management of Data. 1997. 265～276.

[6] Bing, Liu, Hsu, W., Ma, Y. Mining association rules with multiple minimum supports. In: Proceedings of the KDD'99. San Diego, CA, 1999.

[7] Aggarwal, C., Yu, P. S. A new framework for itemset generation. IBM Research Report, RC-21064.

附中文参考文献：

[1] 欧阳为民，蔡庆生. 国际关联规则发现研究述评. 计算机科学，1999，3：41～44.

Discovery of Weighted Association Rules in Databases*

OU-YANG Wei-min[1], ZHENG Cheng[2], CAI Qing-sheng[3]

[1](*Computing Center, Anhui University, Hefei* 230039, *China*);

[2](*Department of Computer Science, Anhui University, Hefei* 230039, *China*),

[3](*Department of Computer Science and Technology, University of Science and Technology of China, Hefei* 230027, *China*)

E mail: oywm@mars.ahu.edu.cn

http://www.ahu.edu.cn

Abstract: Discovery of association rules is a very hot topic in data mining research, which has been found applicable and useful in many areas. In the current researches, all the items in a databases are treated in a uniform way. However, it is not true in the real world databases, in which different items usually have different importances. In order to represent the importance of individual items, the weight value for items is introduced, and a new problem of discovery of weighted association rules is put forward. Due to the introduction of weight for items, it is not sure that any subset of a frequent itemset is also frequent. Thus, a concept of k-support bound of itemsets is set forth, and an algorithm to discover weighted association rules is proposed.

Key words: data mining; knowledge discovery; weighted association rule

* Received December 6, 1999; accepted January 20, 2000

Supported by the National Natural Science Foundation of China under Grant No. 69975001