基于多最小支持度的加权关联规则挖掘算法

邹力鹍 张其善

(北京航空航天大学 电子信息工程学院, 北京 100083)

摘 要: 针对交易数据库中数据项重要性不同的现象, 提出了新的加权关联规则模型, 并基于该模型设计了一个基于多最小支持度的加权关联规则挖掘新算法, 允许用户设定多个最小支持度, 为数据项设置不同的权重, 从而解决数据项的重要性各不相同且出现频率不均匀的问题, 发现更多有趣的规则. 理论证明了相关性质, 并实验验证了算法的有效性.

关 键 词: 数据挖掘; 加权关联规则; 多支持度

中图分类号: TP 312

文献标识码: A 文章编号: 100125965(2007)0520590204

Algorithm of weighted association rules mining with multiple minimum supports

Zou Likun Zhang Qishan

(School of Electronics and Information Engineering Beijing University of Aeronautics and Astronautics, Beijing 100083, China)

Abstract: A new model of weighted association rule was presented in order to solve the problem that data item have not the same importance in datasets. Based on this model, a new algorithm of mining weighted association rules with multiple minimum supports was proposed. The algorithm allows the user to specify varied minimum supports and items weights to reflect the importance and frequency of each data item in datasets. The algorithm aims to deal with problem that items have different importance and varied frequency in transaction database and find more interesting rules which involve both frequent and rare items. The correlative properties of model and algorithm were given and the theories were proved. Finally, the algorithm was tested on the experimental data. Experiment results show that the new algorithm is effective for large databases.

Key words: data mining; weighted association rule; multiple minimum supports

数据库中关联规则的概念由 R. Agraval等[1]首先提出, 是数据挖掘研究的重要分支. 近年, 随着关联规则应用的不断普及与深入, 关联规则挖掘逐渐成为一个具有实用意义的挖掘技术.

挖掘关联规则是从已知数据中得出所有满足最小支持度和最小置信度的关联规则. 通常这一问题可以被分解为两个子问题, 即: 求出数据库中满足最小支持度的所有频繁项集和利用频繁项集生成满足最小置信度的所有关联规则. 事实上, 挖掘关联规则的整个执行过程中第一个子问题是核心问题. 当找到所有的频繁项集后, 相应的关联规则将很容易生成. 因此第一个子问题是挖掘关联规则的重点.

目前, 挖掘频繁项集的经典算法是 Apriori算法[1]和 FP_growth算法[2], 其他大部分算法都是这两种算法的变种. 经典算法有两个默认假设: 数据库中各项目的重要性相同以及数据库中各项目的分布是均匀的, 即出现频率相同或相似. 因此, 经典算法是以平等一致的方式处理数据库中的各个项目. 然而数据库的实际情况却并非如此, 各项目的重要性不尽相同且分布不均. 为了解决前一问题, 研究者提出了加权关联规则的算法[3-5], 而后一问题则通过设计多最小支持度的关联规则挖掘算法来解决[6-7]. 由于这两问题并非对立存在, 因此需要考虑多最小支持度的挖掘加权关联规则的算法. 文献[8]提出了 AMWARMS算法解决该

收稿日期: 2006207203

基金项目: 航空基金资助项目(2006ZC51032); 863资助项目(2006AA701424)

作者简介: 邹力鹍(1977-), 女, 湖南新化人, 博士生, likun_zou@163.com

问题,但该算法存在一定的不足,算法中考虑的多支持度并不是针对不同项目的出现频率,而是针对 1项集、2项集、、、k项集的多个支持度,因此不能较好地解决项目分布不均的情况. 本文提出了新的加权关联规则模型,并设计了一种允许用户根据数据库中数据项的重要性和出现频率设置不同的权重和最小支持度的关联规则挖掘算法 DWARMMS. 理论证明了其相关性质,并通过实验验证了算法的有效性.

1　基本概念

考察交易数据库 T,其项目的集合为 I= {i_1, i_2, , , i_n},每一笔交易都是 I的一个子集,并赋予一个交易标识符 TID. 关联规则形如 X] Y,其中 X< I, Y< I,且 XH Y= Á. 关联规则在交易数据库 T中成立,具有支持度 s和置信度 c. 支持度 s为 XG Y在交易数据库包含的概率. 置信度 c为在某交易中包含 X的前提下同时也包含 Y的概率.

对于给定项目集合 I= {i_1, i_2, , , i_n},为表征项目的重要性,为每一个项目 i_j赋以权值 w_j,其中 0[w_j[1, j= {1, 2, , , n}. 当数据项具有权重后,数据项所属的项集也具有了相应的权重.

定义 1　数据项集 X的权重记为 $W_I(X)$

$$W_I(X) = \frac{1}{|X|}\sum_{i_j \in X} w_j$$

定义 2　交易 t的权重记为 $W_T(t)$

$$W_T(t) = \frac{1}{|t|}\sum_{i_j \in t} w_j$$

定义 3　关联规则 X] Y的加权支持记为 W_{sup}

$$W_{sup}(X \cup Y) = \frac{\sum_{t \in T \wedge X \cup Y < t} W_T(t)}{\sum_{t \in T} W_T(t)}$$

定义 4　对于数据项集 I= {i_1, i_2, , , i_n},其中任一数据项均被赋予它自身所需要满足最小支持度,称之为项最小支持度,记为 $S_{min}(i_j)$.

定义 5　对于数据项集 X< I(|X|> 1),称项集 X所包含的各个项最小支持度的最小值 min(, , $S_{min}(i_j)$, ,) (i_jI X)为项集最小支持度,记为 $S_{min}(X)$.

定义 6　某 k2项集 X被称为频繁项集,如果其加权支持不低于项集 X最小支持度,即

$$W_{sup}(X) \geq S_{min}(X)$$

定理 1　若 X< I, Y< I,且 X< Y,则必有 $W_{sup}(X) \geq W_{sup}(Y)$.

证明　设包含 X, Y的交易数据集合为 |T_x|、|T_y|,由于 X< Y,因而对于 P t∈ T_y,必然有 t∈ T_x,因此 $\sum_{t \in T \wedge X < t} W_T(t) \geq \sum_{t \in T \wedge Y < t} W_T(t)$,又因为 P t∈ T有 $|W_T(t)| > 0$,易得

$$\frac{\sum_{t \in T \wedge X < t} W_T(t)}{\sum_{t \in T} W_T(t)} \geq \frac{\sum_{t \in T \wedge Y < t} W_T(t)}{\sum_{t \in T} W_T(t)}$$

即 $W_{sup}(X) \geq W_{sup}(Y)$.　　证毕

显然这一性质类似 Apriori的递推性质,这对设计相关算法是非常有利的,因为 Apriori的递推性质可以极大地减少搜索空间. 表 1是项目权重和多支持度,表 2是包含 5种商品的交易数据库.

表 1　项目的权重和最小支持度

项名	利润	权重	项最小支持度
苹果	100	0. 1	0. 6
桔子	300	0. 3	0. 65
香蕉	400	0. 4	0. 1
牛奶	800	0. 7	0. 75
可口可乐	900	0. 9	0. 8

表 2　交易数据库

交易标识符	项集	交易权重
1	A, O, M, C	0. 5
2	A, M, C	0. 566 7
3	O, M, C	0. 633 3
4	A, O, M, C	0. 5
5	A, B, C	0. 466 7
6	O, M, C	0. 633 3
7	O, B, M, C	0. 575

如 W_{sup} ({B, C}) = (0. 466 7+ 0. 575) / 3. 875= 0. 26, 0. 26> min(S_{min}(B), S_{min}(C)) = min(0. 1, 0. 8) = 0. 1,所以 {B, C}是频繁项集. 同理, {O, M} {O, C} {O, M, C}也都是频繁项集. 但在引入多支持度后,经典的关联规则挖掘算法 Apriori的递推性质就不再成立了. 例如, W_{sup} ({A, C}) = 0. 524 7< min(0. 65, 0. 8), W_{sup} ({A, B, C}) = 0. 12> min(0. 65, 0. 1, 0. 8). 即:频繁项集 {A, M, C}的子集 {A, M}可以是不频繁的.

2　算　法

不难发现,导致频繁项集的子集必频繁这一性质不成立是由于新项目的加入,而新项目的项最小支持度降低所导致的. 在多最小支持度策略下,通过对数据项按照项最小支持度进行升序排序后,得到图 1所示的子集搜索空间. 在该搜索策略下,不难得到频繁项集的子集必频繁这一性质是满足的,因此可以沿用 Apriori算法中生成候选集的思想.

定理 2　当 X< Y(X< I, Y< I),且 X与 Y具

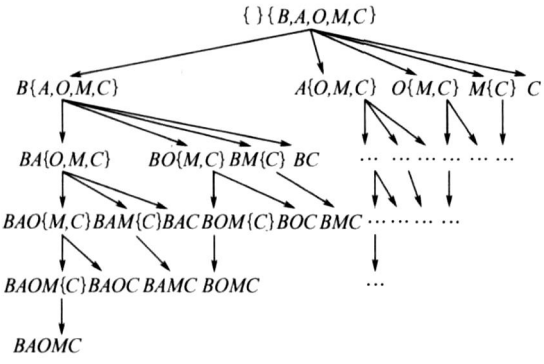


图 1 子集搜索树

有相同的前缀时, 若 X不频繁, 则 Y必不频繁.

证明 设 X= {$X_1, X_2, \cdots, X_m$}, 其中 $S_{min}(X_1)$ [$S_{min}(X_2)$ [… [$S_{min}(X_m)$; Y= {$Y_1, Y_2, \cdots, Y_n$}, $S_{min}(Y_1)$ [$S_{min}(Y_2)$ [… [$S_{min}(Y_n)$, m< n. X与 Y具有相同的前缀, 即 X与 Y的前 k(n\ k\ 1)项相同, 即 $X_1 = Y_1, \cdots, X_k = Y_k$. X不频繁, 有 $W_{sup}(X) < S_{min}(X)$. 根据定义 5和题设, 易得 $S_{min}(X) = S_{min}(X_1) = S_{min}(Y) = S_{min}(Y_1)$, 所以可得 $W_{sup}(X) < S_{min}(Y_1)$. 由定理 1知 $W_{sup}(Y) < W_{sup}(X)$, 因而 $W_{sup}(Y) < S_{min}(Y) = S_{min}(Y_1)$, 所以 Y不频繁. 证毕

类似的, 当 X< I(X< I, Y< I), 且 X与 Y具有相同的项集最小支持度, 若 X不频繁, 则 Y必不频繁. 由此可得相应剪枝性质, 当由 L_{k-1}生成 C_k时, 若某一 k-1项候选集的 k- 1项子集与该 k-1项候选集具有相同首元素或 k-1项候选集的前 2项具有相同的项最小支持度且 k- 1项子集不频繁 (不在 L_{k-1}中)时, 则该 k-1项候选集不可能频繁, 可删除.

算法伪码如下所示. 输入: 事务数据库 D; 各项目的权重和最小支持度数组 MIS; 输出: 加权关联规则.

```
¹ M = sort(I, MIS);
º C1 = init-pass(M, T);
» L1 = {c|c∈ C1, Wsup(c) \ MIS(c)};
¼ L = Á;
½ for (i= 2; i< size; i++) {
¾ Ci = Á; Li = Á;
¿ }
À for(k= 2; Ck-1 X Á; k++) {
Á if(k= 2)
Â C2 = level-2-candidate-gen(C1);
Ã else
Ä Ck = candidate_gen(Lk-1);
Å for each transaction t∈ T do
Æ for each candidate c∈ Ck do
Ç if(c∈ t)
È update Wsup(c)
É Lk = {c|c∈ Ck, Wsup(c) \ MIS(c)};
Ê } L= L∪ Lk;
  }
```

对算法有 4点说明.

1) M= sort(I, MIS): 根据项最小支持度数组 MIS对项集 I进行升序排序, 以形成图 1的搜索空间.

2) init-pass(M, T): 首次扫描数据库, 对 M中各项进行计数, 并生成初始的候选集 C_1, 生成步骤如下: 按照 M的次序找到第一个加权支持数大于其项最小支持度的数据项 i, 加入 C_1, 对于其他数据项 j, 如果 j的加权支持数大于 i的项最小支持度, 则 j也加入 C_1, 具体过程如下.

```
¹ for each item i in M do
º if(Wsup(i) \ MIS(i))
» { C1 = {i}; first= i; break; }
¼ for each item j behind first in M do
½ if(Wsup(j) \ MIS(first))
¾ Add(C1, j);
```

3) level-2-candidate-gen(C_1): 生成 2-项集. 由于可能存在不是频繁 1-项集的项目可以出现成为频繁 2-项集的候选集, 即 C_2是由 C_1而非 L_1导出的, 因此, 2-项集候选集合的生成需要特别的处理, 过程如下.

```
¹ for each item c in C1 do
º if (Wsup(c) \ MIS(c))
» for each item h in C1 that is after c do
¼ if(Wsup(h) \ MIS(c))
½ insert (C2, < c, h> );
¾ return C2;
```

4) candidate_gen(L_{k-1}): 生成候选集. 该过程包含两步骤: 连接步和剪枝步. 连接步和 Apriori算法的一致, 剪枝步则是依据定理 2 剪枝步 can-didate-prune(C_k)过程如下.

```
¹ for each itemset c in Ck
º for each (k- 1)-subsets of c do
» if( c[1] ∈ s) or (MIS(c[1]) =
MIS(c[2]))
¼ if s not in Lk-1
½ delete c from Ck
```

由于整个挖掘过程是在见图 1所示的搜索空间进行的, 因此剪枝过程中步骤 º 的剪枝条件会准确将非频繁项集删除, 保留可能频繁的候选者.

由于算法是基于 Apriori算法思想设计而成的, 因此算法的复杂性与 Apriori算法相似. 算法

可按已有的基于 Apriori的增量算法 (如 FUP算法)的思想进行扩展, 使之适用于增量挖掘.

3　实验分析

试验环境是 Intel 1.5 GHz的 CPU, 256MB内存, 操作系统为 Windows2000, 用 Visual C++ 6.0编写程序. 实验数据是模拟的数据[1]. 项目权值分配如图 2所示.

根据数据项实际的出现频率设置最小支持度, 用文献 [6]的公式: 当 Bf(i) > L_S 时, $S_{min}(i)$ = Bf(i), 否则 $S_{min}(i) = L_S$, 其中 f(i)是数据项 i支持数的百分比表现形式, B则是介于 [0, 1]之间的一个用来表示数据项出现频率的参数, 此外 L_S 也是设置 $S_{min}(i)$的参数, 是用户所设定的一个最低支持度, 当 B= 0时, 所有数据项的最小支持度均相同, 当 B= 1时, f(i)为数据项的最小支持度.

当 B= 0时, 算法就是一个普通的单支持度的加权关联规则算法, 如图 3所示是随交易数据库的量增加时, 发现频繁项集所需要的时间变化, 可以看出, 当 B= 0时, L_S 越小, 数据量越大, 所消耗的时间也越多.

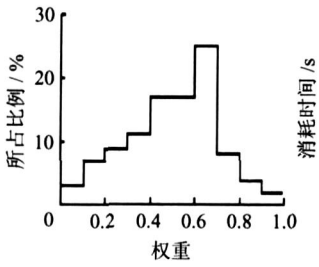


图 2　权重分配

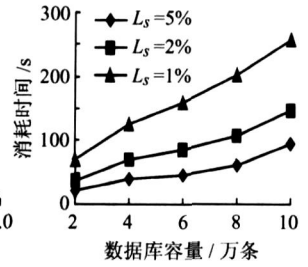


图 3　$\beta = 0$ 时算法的执行时间

当 B介于 [0, 1]时, 算法即为基于多最小支持度的加权关联规则挖掘算法. 如图 4所示是当 L_S = 1%随交易数据库的量增加时, 发现频繁项集所需要的时间变化, 当 B越小, 数据量越大时, 所消耗的时间也越多. 类似的, 当 B越小, 事务包含的数据项越多时, 所消耗的时间也越多. 如图 5所示是当数据库容量和事务包含的数据项相同的情况下, 当 L_S, B不同时, 发现频繁项集所需要的时间变化, 当 B, L_S 越小时, 所消耗的时间也越多.

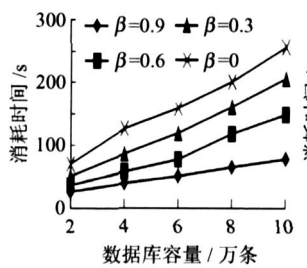


图 4　$L_S = 1\%$ 时数据库容量变化算法的执行时间

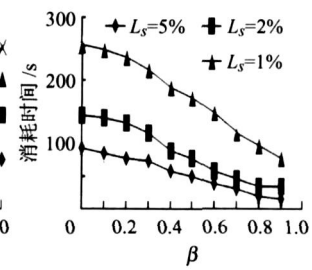


图 5　β 变化时算法的执行时间

4　结束语

本文提出了一个新的加权关联规则模型, 并基于 Apriori算法提出了一个基于多最小支持度的加权关联规则的挖掘算法, 通过为每个数据项设置不同的权重和最小支持度, 从而解决交易数据库中数据项的重要性各不相同, 且实际出现频率分布不均的问题. 多最小支持度可以避免由单支持度引起难以发现出现频率较低的关联规则或出现冗余规则的问题, 而权重的设置使算法倾向于挖掘和重要数据项相关的关联规则. 在文中给出了相应的定理、证明. 在合成数据上的实验验证该算法是有效的.

参考文献 (References)

[1] Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases[C] // Proceedings of the 1993 ACM SIGMOD. Washington: ACM SIGMOD, 1993: 207-216

[2] Jiawei Han, Jian Pei, Yiwen Yin. Mining frequent patterns without candidate generation[C] // Proceedings of the 19th ACM SIGMOD. Dallas, TX, USA: ACM SIGMOD, 2000: 1-12

[3] 欧阳为民, 郑诚, 蔡庆生. 数据库中加权关联规则的发展[J]. 软件学报, 2001, 12(4): 612-619
Ouyang Weimin, Zheng Cheng, Cai Qingsheng. Discovery of weighted association rules in databases[J]. Journal of Software, 2001, 12(4): 612-619(in Chinese)

[4] Cai C H, Fu A da W C, Cheng C H, et al. Ming association rules with weighted items[C] // IEEE In Database Engineering and Applications Symposium. Cardiff: ACM SIGMOD, 1998: 68-77

[5] Feng Tao, Fionn Murtagh, Mohsen Farid. Weighted association rule mining using weighted support and significance framework[C] // Proceedings of the 2003 ACM SIGKDD. Washington, DC, USA: ACM SIGMOD, 2003: 661-666

[6] Bing Liu, Hsu W, Ma Y. Mining association rules with multiple minimum supports[C] // Proceedings of the 1999 ACM SIGKDD. San Diego, CA: ACM SIGMOD, 1999: 337-351

[7] 王振宇, 白石磊, 熊范纶. 多最小支持度策略的关联规则挖掘方法[J]. 小微型计算机系统, 2002, 23(8): 971-973
Wang Zhenyu, Bai Shilei, Xiong Fanlun. An approach of mining association rules with multiple minimum supports stratagem[J]. Mini Microsystem, 2002, 23(8): 971-973(in Chinese)

[8] 段军, 戴居丰. 基于多支持度的挖掘加权关联规则算法[J]. 天津大学学报, 2006, 39(1): 114-118
Duan Jun, Dai Jufeng. Algorithm of mining weighted association rules based on multiple supports[J]. Journal of Tianjin University, 2006, 39(1): 114-118(in Chinese)

(责任编辑: 娄　嘉)