一种高效的最大频繁项挖掘算法*

梁宝华[1],罗振菊[2],徐英虎[3]
(1.巢湖学院 计算机科学与技术系,安徽 巢湖 238000;
2.巢湖市半汤镇街道办事处,安徽 巢湖 238000;
3.巢湖市第一人民医院 计算机网络中心,安徽 巢湖,238000)

摘 要:提出一种下三角矩阵的最大频繁项挖掘算法DTMFIM,首先将事务数据库映射到一个布尔矩阵中,并产生频繁1-项集,然后根据频繁1-项集生产频繁2-项集,且对其结果用下三角的布尔矩阵存储,最大频繁项集可通过这个下三解矩阵求得,在求解过程中不断地压缩这个下三角矩阵。实验表明,此算法实现简洁、高效,与经典的Apriori算法及部分Apriori改进算法相比,特别对大事务集、长项目集数据挖掘效果更为明显。

关键词:关联规则;最大频繁项;向量内积;包含
中图分类号:TP311.13 **文献标识码**:A

An efficient algorithm of maximal frequent itemsets

LIANG Bao Hua[1], LUO Zhen Ju[2], XU Ying Hu[3]
(1.Computer Departmen, Chaohu College, Chaohu 238000, China;
2.Bantang Neighborhood Offices of Chaohu, Chaohu 238000, China;
3.Center of Computer Network, The First Hospital of Chaohu, Chaohu 238000, China)

Abstract: The article proposes a maximal frequent itemsets mining algorithm based on down triangular matrix-DTMFIM. At first, the algorithm projects database into a Boolean matrix and get out frequent 1-itemsets. Secondly, according the frequent 1-itemsets comes into being frequent 2-itemsets and the results are storaged by the Boolean matrix. The maximal frequent itemsets can find out through this down triangular matrix and reducing it in the solving process. Experiment results indicate that the algorithm is not only simple, but also has good efficiency compared with the apriori algorithm and part of improved apriori algorithm. When you mine a large database and long Item-set, the mining effect of this way is more visible than the apriori algorithm.

Key words: association rules; maximal frequent itemsest; vector inner product; include

关联规则[1]挖掘作为数据挖掘的一个重要研究分支,主要研究从大型数据集中发现隐藏的、有趣的、属性间存在的规律。频繁项挖掘则是关联规则挖掘中重要也是第一步要完成的任务。怎样从海量原始数据和大项目集的事务数据库中快速、有效地挖掘出最大频繁项,已成为许多专家、学者所关注的焦点。

目前,能进行有效关联规则挖掘的算法有许多[2-4],但绝大多数挖掘算法都是基于Apriori算法所提出的"自底向上"的思想,其不足之处在于:(1)多次扫描数据库;(2)虽然采取了一定的剪枝策略,但产生候选集仍然较大;(3)测试候选集需要花费大量的时间。本文针对Apriori算法的不足,提出一种利用向量求内积[5-7]运算并对所求结果用下三角矩阵存储的改进方法,来提高频繁项集的挖掘效率。首先,将事务数据库映射到布尔矩阵,然后将矩阵的每列看成一个向量,利用向量求"与"运算,通过频繁1-项集生成频繁2-项集,且用下三角矩阵存储;其次,利用此下三角矩阵生成最大频繁项。研究表明,该算法不仅运算简单、只需扫描1次事

* 基金项目:安徽省级自然科学研究项目(KJ2008A35ZC)

务数据库，而且不会产生大量的候选集等优点。对大数据量大项目集的事务数据库，挖掘的效果与Apriori算法相比，效果更明显。

1 相关定义及定理

设关联规则挖掘的事务数据库记为 D，$|D|$表示整个事务数据库中事务的个数。$I=\{i_1,i_2,\cdots,i_n\}$是项集，其中的元素称为项(item)，每个交易的事务记为 T，则 T 是项的集合，且有 $T\subseteq I$，每个事务都有一个唯一的标识符，称 TID。I的任何子集 X 称为 D 中的项目集(I temset)，$|X|=k$ 称为集合 X 为 k 项目集(k-Itemset)。T 和 X 分别为 D 中的事务和项目集，如果 $X\subseteq T$，称事务 T 包含 X。

数据集 D 中包含项目集 X 的事务数目称为项目集 X 的支持数，记为 σ_x。项目集 X 的支持度记为 $\sup(X)$：

一个关联规则是形如蕴涵式 $X\Rightarrow Y$，且 $X\cap Y=\varnothing$，这里 $X\subset I$，$Y\subset I$。

关联规则 $X\Rightarrow Y$ 的置信度记作：

$\text{confidence}(X\Rightarrow Y)=\sup(X\cap Y)/\sup(X)$；

定理 1：设有 n 个布尔变量 $V_1,V_2,\cdots,V_m$，若：

$V_1,V_2,\cdots,V_m$

$V_1\times V_2=1,V_1\times V_3=1,\cdots,V_1\times V_m=1$

$V_2\times V_3=1,V_2\times V_4=1,\cdots,V_2\times V_m=1$

……

$V_{m-1}\times V_m=1$

则 $V_1\times V_2\times\cdots\times V_m=1$

证明：定理 1 的条件保证了 $V_1,V_2,\cdots,V_m$ 中所有项均为 1，所以结论很显然成立。

关联规则知识发现主要分以下 2 步：(1) 频繁项的挖掘；(2)强关联规则的挖掘。

本文主要针对最大频繁项，因为频繁项为最大频繁项的子集。

2 DTMFIM 算法

关联规则挖掘的任务是要挖掘 D 中所有的强规则。并结合行业知识得出有用的知识，但首先要完成的是频繁项的挖掘。

由于 Apriori 算法所存在的缺点，本文提出了一种基于下三角矩阵的最大频繁项挖掘算法，此算法避免了多次扫描数据库，可直接产生最大频繁项，且算法实现简单。

2.1 算法思想描述

(1)扫描事务数据库，同时计算每个项目的支持度，根据给定的频繁项条件产生频繁 1-项集，并将频繁 1-项集 set $\{\{S_1\},\{S_2\},\cdots,\{S_p\}\}$ 事务信息映射到布尔矩阵 B_array[ ][ ]中，若第 i 个事务、第 j 个属性发生了，则 B_array[i][j]=1，否则以 0 记。

(2)频繁 2-项集的产生：设由(1)得到的频繁 1-项集集合为 set$\{\{S_1\},\{S_2\},\cdots,\{S_p\}\}$，定义一个下三角矩阵 A，若 $\sup(S_i\ \&\ S_j)\geqslant\min\_\sup$，则 $A_{ij}=1$，否则 $A_{ij}=0$。A 数组采用如图 1 格式存储。这样，数组中的元素为 $\sum_{i=1}^{p-1}i=p(p-1)/2$ 个，比正常存储要节省空间 $p(p+1)/2$ 个单位。且为频繁 1-项集中的各项建立一链表 List[]，其中存放能与头节点构成频繁 2-项集的属性标识(标识号按序号从小到大排序)。

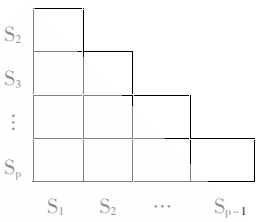

图 1 频繁 2-项集下三角矩阵

(3)扫描下三角矩阵中左边第 i 列($1<i<p-1$)，记下该列中“1”元素的行下标集合 $R=\{i_1,i_2,\cdots,i_k\}$ 且($i_1<i_2<\cdots<i_k$)。若剩余未扫描的属性列号已经在频繁项中，则终止整个算法，否则转至(4)。

(4)取 R 集合中第 1 个元素 i_1，剩余元素组成的新集合为 R'，求 $List[i_1]\cap R'$，若非空，则继续取 R' 的子集，直到空为止，此时将刚才取出的属性序号作为频繁项加入频繁项集中。

(5)若 R 中元素未取完则转至(4)。

(6)若 R 中元素取完，则转到(3)，此时要删除下三角矩阵中的第 i 列，继续检索第 $i+1$ 列。

2.2 算法描述

算法涉及到的部分主要数据结构说明如下：

(1)B_array[ ][ ]为事务数据集要映射的布尔矩阵。

(2)List[i]为频繁 1-项集中第 i 个属性与其他频繁 1-项集中元素能构成频繁 2-项集的属性序号的链表。

(3)array[m][m]频繁 2-项集构成的下三角矩阵。

(4)L 为最大频繁项集。

(5)R 为临时集合，用来存放待搜索的元素。

(6)R'为在 R 中取走部分元素剩余元素的集合。

算法输入：事务数据集 DataSet 及最小支持度 MinSup。

算法输出：最大频繁项集 L。

算法步骤及部分伪代码如下：

(1)扫描数据事务集产生布尔矩阵及产生频繁 1-项集。

(2)将 List[]、array[ ][ ]、L、R、R′清空。

(3)通过频繁 1-项集中各项两两组合看是否能构成频繁 2-项集，并将信息存入 array 中且将相应信息存入 List[ ]中。

(4)$col=1$；$row=2$。

(5)while(col! =m) //m 为频繁 1-项集中元素个数。

(6) { while(row<=m)

(7) { 扫描 array[row][col]，若 array[row][col]=1，则将 row 加入R 中。

(8) row++; }

(9) while(R! =?)

{ 取 R 中的第 1 个元素 i_1，此时在 R 中删除 i_1 得到 R'，并求 $List[i_1]\cap R'$，若为空，则将{col，i_1}并入 L 中，否则再取交集中第 1 个元素 i_{11}，再求 $List[i_{11}]\cap\{List[i_1]\cap R'-i_{11}\}$。依此类推，直至 $List[i_1]\cap R'$为空，将相应最大频繁项并入 L 中。

}

col++;

row=col+1;

}

2.3 算法实现举例

例 1：设一事务数据集如表 1 所示。设最小支持度为 20%，所以得到频繁 1-项集为{a，b，c，d，e，f，g，h}，转换为如图 2 所示的布尔矩阵 B_array。

表 1 事务数据集

事务 TID	事务列表
t_1	cfgh
t_2	ab
t_3	bcdfh
t_4	cefgh
t_5	de
t_6	aefgh
t_7	bdfh
t_8	cdh
t_9	fg
t_{10}	abdeg

B_array=

0	0	1	0	0	1	1	1
1	1	0	0	0	0	0	0
0	1	1	1	0	1	0	1
0	0	1	0	1	1	1	1
0	0	0	1	1	0	0	0
1	0	0	0	1	1	1	1
0	1	0	1	0	1	0	1
0	0	1	1	0	0	0	1
0	0	0	0	0	0	1	0
1	1	0	1	1	0	1	0

图 2 布尔矩阵

由频繁 1-项集中的各项，构造频繁 2-项集且用下三角矩阵存放：为叙述方便将上述频繁 1-项集属性名用数字表示，分别为从 1~8，若(i,j)能构成频繁 2-项集，则 array[i][j]=1，否则 array[i][j]=0。

根据频繁 2-项集的下三角矩阵，可得到频繁 1-项集中的 i 项能与其他项构成频繁 2-项集，且序号大于 i 的项($1\leqslant i<8$)，各集合的包含关系如图 3 所示。

array=

2 1
3 0 0
4 0 1 1
5 1 0 0 1
6 0 1 1 1 1
7 1 0 1 0 1 1
8 0 1 1 1 1 1 1
1 2 3 4 5 6 7

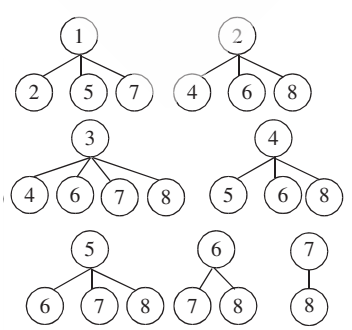

图 3 各项包含关系

找最大频繁项过程：

(1)与 1 号向量构成频繁 2-项集的有{2，5，7}，集合中第一个元素为 2，其余项为 R_1={5，7}。与 2 号向量构成频繁 2-项集的有 R_2={4，6，8}，$R_1\cap R_2=\varnothing$，所以{1，2}构成一最大频繁项集，再在剩余项 R_1={5，7}中取 5，取出后集合中只剩下 7，再查找一下在 5 号向量中恰好有 7 号向量能与其构成频繁 2-项集，这样，{1，5，7}就构成一最大频繁项。最后还乘下{7}，而{7}⊂{1，5，7}，所以与 1 号向量相关的最大频繁项就没有了。此后，删除下三角矩阵中的 1 号向量。

(2)依据同样的方式，可得到其余的最大频繁项。

{2，4，6，8}

{3，4，6，8}，{3，6，7，8}

{4，5，6，8}

{5，6，7，8}

这样，只需几步就把所有的最大频繁项集挖掘出来了。

2.4 算法分析

此算法只需扫描数据库 1 次，即把事务数据库映射到布尔矩阵中。

空间复杂度：在找最大频繁项第(1)步时需 $m\times n$ 个字节(m 为事务数，n 为属性个数)。当频繁 2-项集挖掘后，产生的下三角矩阵需空间 $\frac{n(n-1)}{2}$ 个字节($n<<m$)，此后的所有频繁项均可由此产生，此时布尔矩阵就可删除，以让出空间。在挖掘最大频繁项过程中，不断地压缩下三角矩阵。

时间需求：频繁 1-项集及频繁 2-项集挖掘时间与经典的 Apriori 算法类似，之后的最大频繁项集只需扫描下三角矩阵，无需扫描事务数据库的布尔矩阵。

本文提出了一种高效的最大频繁项 DTMFIM 挖掘算法，算法实现简单，只需扫描数据库 1 次，且利用布尔向量内积运算效率高的特点将数据库映射到布尔矩阵中。算法中利用频繁 1-项集建立频繁 2-项集且用下三角矩阵存储相应信息，然后充分利用此三角矩阵的信息生成最大频繁集，无需生成大量的候选集，不像 Apriori 算法要剪支、自连接及产生大量无用的候选集，对于大数据集、长频繁项，此算法与 Apriori 算法效果显得更为突出。

参考文献

[1] CHEN M，HAN J，YU P S. Data mining an overview from database perspective[J]. IEEE Transactions on Knowledge and Data Enginee-ring，1998(6)：866-883.

[2] AGRAWAL R，SRIKANT T. Fast algorithms for mining association rules in large database[C]. Santiago Proceedings of the 20th VLDB Conference，1994：487-499.

[3] HAY J, PEI J, YIN Y. Mining frequent patterns without candiate generation[C]. Dallas SIOMOD, 2000:1-12.

[4] PES J, HAY J, LU H, et al. H-mine: hyper-structure mining of frequent in large database [C]. San Jose Proceedings of the Int Conf on Data Mi, 2001:441-448.

[5] 刘以安,刘强,邹晓华.基于向量内积的关联规则挖掘算法研究[J].计算机工程与应用,2006,21:172-174.

[6] 胡慧蓉,王周敬.一种基于关系矩阵的关联规则快速挖掘方法[J].计算机应用,2005(7):1577-1579.

[7] 黄龙军,段隆振,章志明.一种基于上三角项集矩阵的频繁项集挖掘算法[J].计算机应用研究,2006(11):25-27.

(收稿日期:2009-03-09)