【**Abstract**】The paper improves Apriori algorithm based on Matrix. At the same time, improves the algorithm of finding association rules. It can reduce the times of accessing database to enhance the efficiency of this algorithm. The examples show that the algorithm is an effective method of association rulers mining.

关联规则挖掘是发现大量数据中项集之间有趣的关联或相关关系。随着大量数据不停被地收集和存储，许多业界人士对于从数据库中挖掘关联规则越来越感兴趣。1994 年，Rakesh AgrawalRama和Krishnan Skrikant首先提出了Apriori[1]算法，它是一种最有影响的挖掘布尔关联规则频繁项集的算法。

挖掘关联规则的对象是含有大量事务的数据库，如何设计一种高效的算法，提高计算效率，降低扫描数据库的次数，是研究关联规则的主要课题。本文提出的基于矩阵的 Apriori 算法改进是将 Apriori 算法的剪枝与矩阵联系起来，这种方法只需扫描一次数据库，从而大大提高了算法的效率。在生成关联规则中，利用了概率论的基本性质也大大减少了计算量。

1 基本概念

设$I=\{i_1,i_2,\ldots,i_n\}$是项的集合，任务相关的数据D是数据库事务的集合，其中每个事务T是项的集合，使得$T\subseteq I$。每个事务有一个标识符，称作TID。

定义 1 每个项I_j的向量定义为

$D_j=\begin{pmatrix}t_{1j}\\t_{2j}\\\vdots\\t_{nj}\end{pmatrix}$，其中，$t_{ij}=\begin{cases}0, I_j\notin T_i\\1, I_j\in T_i\end{cases}$，$I_j$的支持度计数为

$$\text{support\_count}(I_j)=\sum_{i=1}^{n}t_{ij}$$

定义 2 2-项集$\{I_i,I_j\}$的向量定义为

$$D_{ij}=D_i\Lambda D_j=\begin{pmatrix}d_{1i}\Lambda d_{1j}\\d_{2i}\Lambda d_{2j}\\\vdots\\d_{ni}\Lambda d_{nj}\end{pmatrix}$$

其中，“ Λ ”是“逻辑与”运算符。这样，2-项集$\{I_i,I_j\}$的支持度计数为

$$\text{support\_count}\{I_i,I_j\}=\sum_{k=1}^{n}(d_{ki}\Lambda d_{kj})$$

定义 3 k-项集$\{I_1,I_2,\ldots,I_k\}$的向量定义为

$$D_{12\ldots k}=D_1\Lambda D_2\Lambda\cdots\Lambda D_k=(D_1\Lambda D_2\Lambda\ldots\Lambda D_{k-1})\Lambda D_k$$

这样，k-项集$\{I_1,I_2,\ldots,I_k\}$的支持度计数为

$$\text{support\_count}\{I_1,I_2,\ldots,I_k\}=\sum_{q=1}^{n}\{(d_{q1}\Lambda d_{q2}\Lambda\ldots\Lambda d_{q(k-1)})\Lambda d_{qk}\}$$

定义 4 项集 I 的矩阵记为

$$\boldsymbol{D}=(D_1,D_2,\ldots,D_n)=\begin{pmatrix}d_{11}&d_{12}\ldots d_{1n}\\d_{21}&d_{22}\ldots d_{2n}\\\vdots&\vdots\quad\vdots\\d_{p1}&d_{p2}\ldots d_{pn}\end{pmatrix}$$

2 频繁项集的生成

本文所提出的算法是基于 Apriori 算法的，其中利用了 Apriori 的性质：频繁项集的所有非空子集都必须也是频繁的。具体步骤如下：

(1)根据定义 1 生成频繁 1-项集，并保存各频繁 1-项集的计数。

(2)利用定义 2 由频繁 1-项集生成频繁 2-项集，并保存各频繁 2-项集的计数。

(3)记号$l_i[j]$表示l_i的第j项，执行连接$L_{k-1}\ \bowtie\ L_1$，其中L_{k-1}是可连接的，如果它们的前(k-2)项相同，则是L_{k-1}的元素l_1和l_2是可连接的，若

$$(l_1[1]=l_2[1])\Lambda(l_1[2]=l_2[2])\Lambda\cdots\Lambda(l_1[k-2]=l_2[k-2])\Lambda(l_1[k-1]<l_2[k-1])$$

则连接后产生的 k-项候选集是$l_1[1]l_1[2]\cdots l_1[k-1]l_2[k-1]$。接

基金项目：现代通信国家重点实验室基金资助项目(5143603ZDS0601，51436010103JB0601)

作者简介：李　超(1979－)，女，硕士生，主研方向：应用数学，数据挖掘；余昭平，教授

着对连接产生的候选项集进行剪枝，具体方法是：判断 2-项集 $l_1[k-1]l_2[k-1]$是否频繁。若不频繁，则直接将其从候选项中删除；反之，由 Apriori 性质：k-项集的所有(k-1)-项集都必须是频繁的，则判断此 k-项集的(k-2)个包含 $l_1[k-1]l_2[k-1]$ 的(k-1)-项子集是否全部频繁；若频繁，则将其纳如频繁 k-项集的候选集，若不频繁，就将其删除。

(4)将候选 k-项集按定义 3 生成频繁 k-项集，并保存各频繁 k-项集的计数。

3 关联规则生成

关联规则产生如下：(1)对于每个频繁项集 l，产生 l 的所有非空子集；(2)对于 l 的每个非空子集 s，如果

$$\frac{\sup\_count(l)}{\sup\_count(l-s)} \geq \min\_conf \qquad (1)$$

则输出规则“ $s \Rightarrow (l-s)$ ”。其中，min_conf 是最小置信度阈值。这样，对于每个频繁 k-项集，需要考虑的规则数为

$$C_k^1 + C_k^2 + \cdots + C_k^{k-1}$$

定理 对于项集 A、B，如果 $A \subset B$，则

$$\sup\_count(A) \geq \sup\_count(B)$$

证明 因为 $A \subset B$，所以若一事务包含项集 B，则它一定包含项集 A；反之，若一事务包含 A，它不一定就包含 B，故 $\sup\_count(A) \geq \sup\_count(B)$。

由判定式(1)知，对某个频繁k-项集l_k，其支持度(分子)是不变的，分母越小，置信度越大。由定理 1，只需找到满足判定式的最小项集$(l-s)$，则所有比它大的子集肯定也满足判定式，直接得出关联规则，从而能大大减少计算量，提高算法的效率。

4 应用实例

根据改进的算法，下面通过一个具体的例子进行分析，事务数据库由表 1 给出，假定最小事务支持计数为 2。即

$$\min\_sup = 2/9 = 22\%$$

表 1 事务数据库

TID	项集
T1	I1,I2,I5
T2	I2,I4
T3	I2,I3
T4	I1,I2,I4
T5	I1,I3
T6	I2,I3
T7	I1,I3
T8	I1,I2,I3,I5
T9	I1,I2,I3

$$(1)\ D = (D_1, D_2, \ldots, D_n) = \begin{pmatrix} 1&1&0&0&1 \\ 0&1&0&1&0 \\ 0&1&1&0&0 \\ 1&1&0&1&0 \\ 1&0&1&0&0 \\ 0&1&1&0&0 \\ 1&0&1&0&0 \\ 1&1&1&0&1 \\ 1&1&1&0&0 \end{pmatrix}$$

$$\text{support\_count}(I_1) = \sum_{i=1}^{9} t_{i1} = 6 \geq 2$$

同理计算I2, I3, I4, I5 的最小支持度计数均≥2，生成频繁 1-项集L_1={I1:6,I2:7,I3:6,I4:2,I5:2}。

(2)利用L_1生成候选 2-项集{I1I2, I1I3, I1I4, I1I5, I2I3,I2I4,I2I5,I3I4,I3I5,I4I5}。

$$D_{12} = D_1 \Lambda D_2 = \begin{pmatrix} d_{11}\Lambda d_{12} \\ d_{21}\Lambda d_{22} \\ \vdots \\ d_{91}\Lambda d_{92} \end{pmatrix} = (1\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 1)^T$$

$$\text{sup\_count}\{I1I2\} = \sum_{q=1}^{9} (d_{q1}\Lambda d_{q2}) = 4$$

同理可计算sup_count{I1I4}=1， sup_count{I3I4}=0，sup_count{I3I5}=1， sup_count{I4I5}=0，其余的都≥2，故频繁 2-项集L_2={I1I2:4, I1I3:4, I1I5:2, I2I3:4, I2I4:2, I2I5:5 }。

(3)连接$L_2 \bowtie L_2$={I1I2I3, I1I2I5, I1I3I5, I2I3I4, I2I3I5, I2I4I5}产生候选 3-项集C_3，由于{I3I5, I3I4, I4I5}是非频繁的 2-项集，将{I1I3I5, I2I3I4, I2I3I5, I2I4I5}从C_3中删除。因此只需考虑{I1I2I3, I1I2I5}的支持度即可，由定义 3 知

$$D_{123} = (D_1 \Lambda D_2) \Lambda D_3$$
$$= (1\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 1)^T \Lambda (0\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 1)^T$$
$$= (0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1)^T$$

sup_count{I1I2I3}=2

同理可知 sup_count{I1I2I5}=2 ；从而频繁 2-项集L_2={I1I2I3:2, I1I2I5:2}。

(4) 连接$L_3 \bowtie L_3$={I1I2I3I5}产生候选 3-项集C_4，由于{I3I5}是非频繁的 2-项集，将{I1I2I3I5}从C_4中删除。这样 $C_4 = \Phi$，因此算法终止，找出了所有的频繁项集。

(5)令最小置信度为 70%，在{I1I2I5}的所有 1-项子集中，只有 sup_count{I1I2I5}/sup_count{I5}=2/2=1，满足最小置信度。由{I1I2I5}产生的关联规则是 $I5 \Rightarrow I1\Lambda I2$，$I5\Lambda I2 \Rightarrow I1$，$I1\Lambda I5 \Rightarrow I2$；在此方法中，频繁项集所包含的项越多，越能提高效率。

5 总结与展望

该算法与传统的 Apriori 算法相比，可以大大减少扫描数据库的次数，从而提高算法的效率；在生成关联规则中，利用了概率论的基本性质，与传统的要求频繁项集的所有非空子集，并计算置信度进行判定相比，也是大大减少了计算量。

虽然关于关联规则的研究已进行了近 10 年，但是离广泛应用还有很长的一段时间，算法本身及将算法有效的实现都有待于进一步的研究。

参考文献

1 Agrawal R, Imielinski T, Wami A S. Mining Association Rules Between Sets of Items in Large Databases[C]. Proc.of the ACM SIGMOD Conference on Management of Data,Washington, 1993-05: 207-216.

2 范　明译. 数据挖掘: 概念与技术[M]. 北京: 机械工业出版社, 2003.

3 Kleinberg J, Papadimitriou C, Raghavan P. Segmentation Problems[C]. Proceedings of the 30th Annual Symposium on Theory of Computing, 1998.

4 邵峰晶，于忠清. 数据挖掘原理与算法[M]. 北京: 中国水利水电出版社, 2003.