第 33 卷 第 5 期 计 算 机 工 程 2007 年 3 月
Vol.33 No.5 Computer Engineering March 2007
·博士论文· 文章编号：1000—3428(2007)05—0029—03 文献标识码：A 中图分类号：TP311

具有上界约束的关联规则算法研究

宫 雨
(中国石油大学(北京)工商管理学院，北京 102249)

摘 要：针对具有上界约束的关联规则问题进行了研究，给出了上界约束的定义，分析了满足上界约束频繁集的性质，并给出了相关的证明。在此基础上，提出了基于 FP-Tree 的上界约束算法，采用了预先测试的方法，降低了测试项集的成本，提高了计算效率，实验结果证明该算法具有较高的效率。

关键词：数据挖掘；关联规则；约束

Research on Association Rules with Upper Bound Constraints

GONG Yu
(School of Business Administration, China University of Petroleum (Beijing), Beijing 102249)

【Abstract】 This paper discusses algorithms for association rules with upper bound constraints. The definition of upper bound constraints is presented, property of frequent item sets which satisfy upper bound constraints is analyzed and proved. And an algorithm of association rules with upper bound constraint is presented, which is based on FP-Tree. In this algorithm, the method of test in advance is adopted, which can reduce the cost of item set test, boost the performance. Experimental test show the algorithm is quite efficient.

【Key words】 Data mining; Association rules; Constraints

关联规则发现是数据挖掘中重要的研究领域，受到了研究者的广泛关注，有广阔的应用前景。Bose和Mahapatra对 63 家使用数据挖掘技术的企业进行了调查[1]。接近 20%的被调查企业使用了关联规则发现技术，主要应用于市场营销领域，具体包括零售分析、购物篮分析、市场细分等方面。这些应用都取得比较好的效果，对企业的相关决策起到了辅助作用。

但经典关联规则算法往往会发现大量的规则，而其中的许多规则是用户不感兴趣的或不符合用户的期望，这导致了对约束关联规则问题的研究。约束关联规则允许用户指定挖掘的约束，体现用户的领域知识，使发现的关联规则更加有效、更符合用户的要求。

关联规则的约束从性质上可以分为单调约束、反单调约束、简洁性约束、可转变约束、不可转变约束。从涉及的变量个数上可以分为单变量约束(如 $Sum(I.price)>100$)和双变量约束（如 $Sum(I.price)\times I.Count>100$）。目前约束关联规则问题的研究大多集中在单变量约束问题上[1~3]，针对双变量约束研究较少，已有的解决方法主要是将双变量类简洁性约束转变成单变量约束[1]。但是这类方法不能解决无法转变为单变量约束的双变量约束问题（如 $Sum(I.price)\times I.Count>100$）。

本文针对双变量约束问题中的上界约束问题，给出了问题的描述，分析了该问题的性质，提出了针对该问题的算法，并通过实验证明了该算法的有效性。

1 具有上界约束关联规则问题描述

设D为事务数据库，$I=\{i_1,i_2,...,i_n\}$为数据库中的全部项集，$P=\{p_1,p_2,...,p_n\}$（$p_j\geq 0$）表示对应项集I的某个变量值的集合，其中p_j表示对应项i_j的变量的值 (如商品j对应的成本或费用)。有k-项集$X(X\subseteq I)$在事务数据库D中的计数为$Count(X)$，记项集X对应的变量值之和与其计数的乘积为$P(X)=(\sum_{i_j\in X}p_j)\times Count(X)$。

具有上界约束的关联规则问题描述为：设项集上界约束为R_0，那么所要讨论的关联规则问题的形式为“$X\Rightarrow Y$”，满足最小支持度 *minsup* 以及最小置信度 *minconf*，且规则对应的$P(X\cup Y)<R_0$。

对于具有上界约束的关联规则问题有如下特点：

(1)上界约束是一种双变量约束，由项集变量值之和与项集计数的乘积构成。

(2)上界约束是混合型约束，它包含了对项集以及其计数的约束。

根据双变量约束的类简洁性定义[5]可知，上界约束不存在类简洁性。因此，现有的双变量约束算法不能解决具有上界约束的关联规则问题。对于具有上界约束的关联规则问题不能应用已有的约束关联规则算法，需要寻找新的算法来求解问题。若要寻找新的算法，则需要明确问题的定义以及求解的目标分析如下：

定义 1 若有项集 X 满足 $P(X)<R_0$，称 X 为满足上界约束项集。

根据定义 1，要使 $P(X)<R_0$，即需 $(\sum_{i_j\in X}p_j)\times Count(X)<R_0$，即

$$Count(X)<R_0/(\sum_{i_j\in X}p_j) \quad (1)$$

不等式（1）表明为了满足上界约束R_0，项集的最小支持计数与项集的变量值之和有关。如果变量值之和大，则项集的支持数则需要相应地减少，变量值之和小，那么它的支持

作者简介：宫 雨(1976－)，男，博士、讲师，主研方向：数据挖掘，管理信息系统

收稿日期：2006-03-09 **E-mail：**gongyu_arm@126.com

数则可以稍大。

定义 2 若有频繁集 X 满足 $P(X)<R_0$，称 X 为满足上界约束的频繁集。

由于 X 为频繁集，因此它的计数满足 $Count(X)\geq|D|\times Minsup$，同时由于 X 为满足上界约束的项集，因此有 $Count(X)<R_0/(\sum_{i_j\in X}p_j)$，即 X 的计数满足

$$R_0/(\sum_{i_j\in X}p_j)>Count(X)\geq|D|\times Minsup \quad (2)$$

通过上述转换后，“具有上界约束的关联规则发现问题”转化成为寻找满足不等式(2)的项集。

2 具有上界约束关联规则问题的性质

对具有上界约束的关联规则问题，最直接的求解方法是：首先寻找满足最小支持度的频繁集的集合；然后在此基础上生成满足变量值约束的频繁集的集合，这种方法可以称为直接算法。

但直接算法没有充分利用问题的特性，需要寻找新的算法。由于关联规则中FP-Growth算法[6]的效率较高，因此本文选择其作为上界约束的关联规则算法的基础。具体分析 如下：

根据文献[6]，FP-Growth 算法中存在如下的一个性质：若一个 FP-Tree 中有且仅有一个分枝 P，那么 FP-Tree 中频繁集的全集可以由 P 中的项组合而成，频繁集的计数由该集合中项的最小计数决定。

而根据上述的性质，可以进一步推导得出如下的性质：

性质 1 设频繁 1-项集I_m对应的FP-条件树有且仅有一个分枝，该分枝为

$$<I_{m1}:C_{m1},I_{m2}:C_{m2},...,I_{m(n-1)}:C_{m(n-1)},I_{mn}:C_{mn}>$$

其中 $I_{mi}(i=1,2,...,n)$ 为项，而对应的 $C_{mi}(i=1,2,...,n)$ 为项 $I_{mi}(i=1,2,...,n)$ 在该条件树上的计数，如果 $C_{m1}\geq C_{m2}\geq C_{m(n-1)}\geq C_{mn}$，那么必然有由该条件树生成的频繁集 $X(X\supseteq\{I_m,I_{mn}\})$的计数为 C_{mn}。

证明 由 FP-Growth 算法生成频繁集的过程以及性质可知，当条件树有且仅有一个分枝时，该条件树所生成频繁集的计数必然等于该项集中计数为最小的项，由于 $C_{m1}\geq C_{m2}\geq C_{m(n-1)}\geq C_{mn}$，因此 $X\supseteq\{I_m,I_{mn}\}$ 的计数必然为 $Count(X)=C_{mn}$。同理频繁 n-项集 X 对应的 FP-条件树有且仅有一个分枝，那么性质 1 也成立。

例如：有 I_1 对应的FP条件树的唯一一个分枝为 $<I_2:3,I_4:2,I_3:2>$，那么必然有 $\{I_1,I_3\}$ 的计数为 2，$\{I_1,I_4,I_3\}$、$\{I_1,I_2,I_3\}$、$\{I_1,I_2,I_4,I_3\}$ 的计数也为 2。

条件 1 设频繁 1-项集I_m对应的FP-条件树有且仅有一个分枝，该分枝为

$$<I_{m1}:C_{m1},I_{m2}:C_{m2},...,I_{m(n-1)}:C_{m(n-1)},I_{mn}:C_{mn}>$$

其中 $I_{mi}(i=1,2,...,n)$ 为项，而对应的 $C_{mi}(i=1,2,...,n)$ 为项 $I_{mi}(i=1,2,...,n)$ 在该条件树上的计数。计数满足不等式 $C_{m1}\geq C_{m2}\geq C_{m(n-1)}\geq C_{mn}$，计数相同的节点按照项的变量值的降序排列。

定理 1 有I_m对应的FP-条件树满足条件 1，若有 $(k+2)$-项集 $\{I_m\}\cup\{I_{m1},I_{m2},...,I_{mk}\}\cup\{I_{mq}\}$（其中 $k<q$）不满足上界约束，那么 $(k+2)$ - 项集 $\{I_m\}\cup\{I_{m1},I_{m2},...,I_{mk}\}\cup\{I_{mi}\}$（其中 $k<i<q$，且 $C_{mq}=C_{mi}$）也不满足上界约束。

证明 若 $\{I_m\}\cup\{I_{m1},I_{m2},...,I_{mk}\}\cup\{I_{mq}\}$不满足上界约束，即 $C_{mq}\times(\sum_{j=1}^{k}p_{mj}+p_m+p_{mq})\geq R_0$。

由于 $k<i<q$，由条件 1 可知 $p_{mq}\leq p_{mi}$，且 $C_{mq}=C_{mi}$，因此有

$$C_{mq}\times(\sum_{j=1}^{k}p_{mj}+p_m+p_{mi})\geq C_{mq}\times(\sum_{j=1}^{k}p_{mj}+p_m+p_{mq})\geq R_0 \quad (3)$$

即

$$C_{mi}\times(\sum_{j=1}^{k}p_{mj}+p_m+p_{mi})\geq R_0 \quad (4)$$

也即 $\{I_m\}\cup\{I_{m1},I_{m2},...,I_{mk}\}\cup\{I_{mi}\}$（其中 $k<i<q$，且 $C_{mq}=C_{mi}$）不满足上界约束，定理 1 得证。同理若频繁 n-项集 X 对应的 FP-条件树有且仅有一个分枝，那么定理 1 也成立。

例如：设上界约束R_0=10，各项的变量值为p_1=2，p_2=3，p_3=1，p_4=4，I_1 对应的FP-条件树的唯一一个分枝为 $<I_2:3,I_4:2,I_3:2>$，若 $\{I_1,I_2,I_3\}$ 不满足上界约束，那么 $\{I_1,I_2,I_4\}$也不满足上界约束。

定理 2 有I_m对应的FP-条件树满足条件 1，若有 $(k+2)$-项集 $\{I_m\}\cup\{I_{m1},I_{m2},...,I_{mk}\}\cup\{I_{mq}\}$（其中 $k<q$）不满足上界约束，那么项集

$$\{I_m\}\cup\{I_{m1},I_{m2},...,I_{mk}\}\cup\{I_{m(k+1)}\}....\cup\{I_{m(q-1)}\}\cup\{I_{mq}\}$$

也不满足上界约束。

证明 若 $\{I_m\}\cup\{I_{m1},I_{m2},...,I_{mk}\}\cup\{I_{mq}\}$不满足上界约束，即 $C_{mq}\times(\sum_{j=1}^{k}p_{mj}+p_m+p_{mq})\geq R_0$。

由于 $p_{mj}\geq 0$，必然 $\{I_m\}\cup\{I_{m1},I_{m2},...,I_{mk}\}\cup\{I_{mq}\}$的变量值之和小于 $\{I_m\}\cup\{I_{m1},I_{m2},...,I_{mk}\}\cup\{I_{m(k+1)}\}....\cup\{I_{m(q-1)}\}\cup\{I_{mq}\}$ 变量值之和。

由条件 1 和性质 1 知 $\{I_m\}\cup\{I_{m1},I_{m2},...,I_{mk}\}\cup\{I_{mq}\}$ 与项集 $\{I_m\}\cup\{I_{m1},I_{m2},...,I_{mk}\}\cup\{I_{m(k+1)}\}....\cup\{I_{m(q-1)}\}\cup\{I_{mq}\}$ 的计数为 C_{mq}，那么有

$$\{I_m\}\cup\{I_{m1},I_{m2},...,I_{mk}\}\cup\{I_{m(k+1)}\}....\cup\{I_{m(q-1)}\}\cup\{I_{mq}\}$$

不满足上界约束，定理 2 得证。同理若频繁 n-项集 X 对应的 FP-条件树有且仅有一个分枝，那么定理 2 也成立。

例如：设上界约束R_0=10，各项的变量值为p_1=2，p_2=3，p_3=1，p_4=4，I_1 对应的FP-条件树的唯一一个分枝为 $<I_2:3,I_4:2,I_3:2>$，若 $\{I_1,I_2,I_3\}$ 不满足上界约束，那么 $\{I_1,I_2,I_4,I_3\}$也不满足上界约束。

3 具有上界约束关联规则的算法

由上述的定理和性质，可以得出基于 FP-Tree 的具有上界约束问题的算法 UBC-FP（Upper Bound Constraints with FP-Tree），其步骤为：

(1)扫描数据库一次，得到频繁 1-项集，将 1-项频繁集按照计数降序排列。

(2)再次扫描数据库，生成 FP-Tree，调用 FP-Growth 算法，根据 FP-Growth 算法生成频繁集的过程如下面的步骤。

(3)对所有有且仅有一个分枝的 FP-条件树，首先对各个条件树中项按照计数的降序排列，然后对各个条件树中计数相同的节点按照变量值的降序排列，最后进行如下计算：

1)若 $\{I_m,I_{mn}\}$不满足上界约束，那么进行如下计算：

①根据定理 2，若 $\{I_m,I_{mn}\}$不满足上界约束，那么包含 $\{I_m,I_{mn}\}$的 n 项集（$n>2$）也不满足上界约束。

②根据定理 1，计数也为C_{mn}的节点与 $\{I_m\}$构成的 2-项集也不满足上界约束。

③从计数大于C_{mn}的节点与$\{I_m\}$构成 2-项集开始测试。

2)若$\{I_m, I_{mn}\}$满足上界约束，将其加入到L_R中，并进行如下计算：

①从包含$\{I_m, I_{mn}\}$长度为 3 的项集开始测试，首先从除了I_m、I_{mn}两项之外变量值最小的 1 项与$\{I_m, I_{mn}\}$合并，形成 3-项集，测试其是否满足上界约束，若不满足则停止在包含$\{I_m, I_{mn}\}$项集上的计算，若满足则其加入到L_R中。然后，计算变量值次小的前 1 项，若不满足则停止在包含$\{I_m, I_{mn}\}$项集上的计算，若满足则其加入到L_R中。依次计算直到测试完变量值最大 1 项为止。

②从包含$\{I_m, I_{mn}\}$且长度为l=4、5、…、n-1 的项集开始测试，首先从除了I_m、I_{mn}两项之外变量值最小的l-2 项与$\{I_m, I_{mn}\}$合并，形成l-项集，测试其是否满足上界约束，若不满足则停止在包含$\{I_m, I_{mn}\}$且长度为l的项集计算，若满足则加入到L_R中。然后，计算变量值次小的前l-2 项，若不满足则停止在包含$\{I_m, I_{mn}\}$项集上的计算，若满足则其加入到L_R中。依次计算直到测试完变量值最大l-2 项为止。

3)完成 1）与 2）计算后，I_{mn}可以从条件树中删除，因为所有包含I_{mn}的项集都已经测试过。

4)从$\{I_m, I_{m(n-1)}\}$、$\{I_m, I_{m(n-2)}\}\ldots\{I_m, I_{m1}\}$的项集重复上述 1）和 2)、3）相同的测试，完成对所有有且仅有一个分枝的 FP-条件树的计算。

(4)对有多条分枝的FP-条件树，调用FP-Growth算法的第(2)步。若没有相应的条件FP-Tree，那么构造生成频繁集，然后根据上界约束进行计算，将满足约束的频繁集加入到L_R中，若有相应的条件FP-Tree，那么递归调用FP-Growth算法，直到只有一条FP-条件树或无法构造FP-条件树为止。

本文基于 FP-Tree 的具有上界约束算法有如下特点：

(1)由于算法使用了 FP-Tree，因此整个计算过程只需要扫描数据库两次。

(2)对有且仅有一个分枝的 FP-条件树采用了预先测试的方法，避免生成所有频繁集后测试的方法，提高了计算效率。

(3)对有多个分枝的 FP-条件树递归调用 FP-Growth 算法。

4 实验结果及分析

下面通过实验来分析分组多支持度关联算法与原有算法之间的性能比较。实验的运行环境是 CPU 为 P4 1.6GHz，内存为 512MB，操作系统为 Windows 2003，用 Java 编写了两种算法。直接算法采用 FP-Growth 算法，首先计算出频繁集，然后根据上界约束计算满足约束的频繁集。

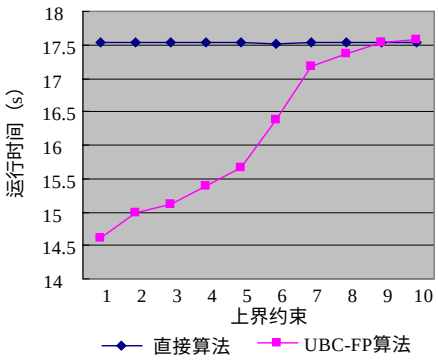


图 1 上界约束变化时 UBC-FP 与直接算法对比

数据源为合成数据，使用数据生成器生成，数据的分布方式为水平分布，数据文件的大小为 4.6MB，包含了 10 万条事务，100 属性，平均事务长度为 7，其中最大的事务长度为 15。当最小支持度为 10%固定不变，上界约束变化时，实验结果如图 1 所示。

从图 1 中可以看出在最好情况下(上界约束$R_0 = 0$)，所有的频繁集都不满足上界约束的情况。UBC-FP 算法的效率比直接算法的效率要高。效率提高的主要原因是由于当所有的频繁集都不满足上界约束时，测试单条路径中满足上界约束的频繁集的工作要少得多，因此提高了算法的效率。在最坏情况下(即所有的频繁集都满足上界约束)，UP-FP 算法与直接算法效率相当，两种算法都需要对所有的频繁集进行测试。

当上界约束为固定不变，最小支持度变化时，实验结果如图 2 所示。

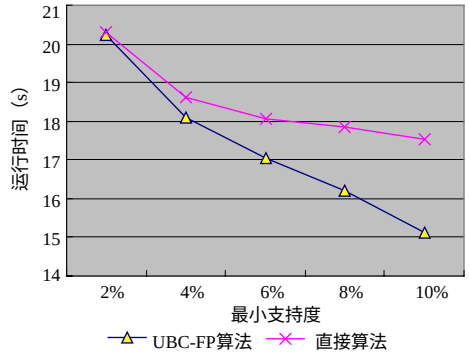


图 2 最小支持度变化时 UBC-FP 与直接算法对比

图 2 中数据显示，随着最小支持度逐渐升高，两种算法的计算效率的差别增大。效率差异增大的原因是：(1)当最小支持度比较小时，很多频繁集满足上界约束，因而从单条路径生成满足约束的频繁集时需测试较多的次数；(2)随着最小支持度升高，不满足上界约束的频繁集增多，在单条路径上测试的次数逐渐降低。因此，提高了算法的效率，特别是最小支持度为 10%时，所有的频繁集都不满足上界约束，在单条路径的需测试次数更少。上述的实验结果表明 UBC-FP 比直接算法更加有效。

5 结束语

本文讨论了具有双变量上界约束的关联规则问题，分析了问题的性质，给出了相关的性质、定理以及证明，提出了该问题的算法，并通过实验分析了算法的高效性。

参考文献

1 Bose I, Mahapatra R. Business Data Mining: A Machine Learning Perspective[J]. Information & Management, 2001, 39(3): 211-225.

2 Srikantm R, Vu Q, Agrawal R. Mining Association Rules with Item Constraints[C]//Proceedings of the International Conference on Knowledge Discovery and Data Mining, Menlo Park, USA. 1997: 67-73.

3 Ng R, Lakshmanan L, Han J, et al. Exploratory Mining and Pruning Optimizations of Constrained Associations Rules[J]. ACM SIGMOD Record, 1998, 27(2): 13-24

4 卢炎生, 杨　芬, 赵　栋. 带单调约束的关联规则挖掘[J]. 计算机工程, 2004, 30(15): 78-79

5 Lakshmanan L, Ng R, Han J, et al. Optimization of Constrained Frequent Set Queries With 2-variable Constraints[J]. ACM SIGMOD Record, 1999, 28(2): 157-168

6 Han J, Pei J, Yun Y. Mining Frequent Patterns without Candidate Generation[J]. ACM SIGMOD Record, 2000, 29(2): 1-12.