单调和反单调约束条件下关联规则的挖掘算法分析*)

杜剑峰　李　宏　陈松乔　陈建二
(中南大学信息科学与工程学院　长沙410083)

摘　要　本文充分利用了Eclat算法的概念格理论和等价类划分方法，将约束条件融入基于垂直数据分布的关联规则挖掘算法中。提出了一种新的反单调和单调约束条件下关联规则的挖掘算法，分别为EclatA算法和EclatM算法。算法采用自底向上的搜索方法，在发现频繁项集的同时进行约束条件的检验。数据库的扫描次数较少，无需对候选项集进行剪枝，占用内存较小。实验证明：该算法的执行效率比已有算法有显著提高。

关键词　数据挖掘，约束性关联规则，概念格，等价类

Algorithm Analysis for Anti-monotone and Monotone Constraints Association Rules Mining

DU Jian-Feng　LI Hong　CHEN Song-Qiao　CHEN Jian-Er
(College of Information Engineering, Central South University, Changsha 410083)

Abstract　Anti-monotone and monotone constraints are integrated into the algorithm of association rules mining based on vertical data layout by utilizing the lattices theory and the decomposing method of equivalence classes sufficiently. Two new algorithms are also presented. One is based on anti-monotone constraint named EclatA. The other is based on monotone constraint named EclatM. A bottom-up search method is put forward and the constraints are checked at the process of calculating frequent itemsets. The new algorithms scan the database few times and have no need of pruning candidate itemsets. At the same time, they can be solved in memory. Results from our experiments show that the performance of the new algorithms is unmatched by any previous algorithms

Keywords　Data mining, Association rules with constraints, Lattices, Equivalence classes

1　引言

近几年来，随着关联规则的理论与实际相结合的不断发展，基于约束的关联规则挖掘算法成为了研究的热点。约束关联规则挖掘算法能够从大量的频繁项集中筛选出有用的规则，提高挖掘效率。目前，应用最广泛的约束主要包括反单调约束和单调约束。典型的基于单调和反单调约束条件下的关联规则挖掘算法是基于Apriori算法[2]的CAP算法[4]和基于FP-growth算法[3]的约束关联规则挖掘架构CFG-约束频繁模式增长[5,6](constrained frequent pattern growth)。

CAP算法将反单调约束分为2种情况：(1) 约束条件既是反单调约束，又是简洁性约束，如$\min(S)\geqslant v$；(2)约束条件是反单调约束，但不是简洁性约束，如$\mathrm{sum}(S)\leqslant v$。对于第一种情况，CAP算法仅仅是将Apriori算法中的C_1替换为C_1'。C_1指候选1-项集，$C_1'=\{e\mid e\in C_1 \& e\in \delta_p(Item)\}$，即$C_1'$指满足约束条件$C$的所有候选1-项集；对于第二种情况，令$C_k$为Apriori算法中的候选$k$-项集，约束条件为$C$，对于任一项集$S\subseteq C_k$，如果$S$不满足约束条件$C$，则将$S$从$C_k$中删除。也就是说，在扫描数据库进行支持度计数之前，就进行约束条件的检验。在CAP算法中，没有明确提出单调约束的概念。根据Apriori类算法的特点，可以通过一个后处理的步骤对Apriori算法的结果进行过滤，也可以通过定义一个弱约束C'的方法解决。CAP算法能够产生完备的满足约束条件的频繁项集，但存在Apriori算法生成的候选频繁项集数目较大和需要重复扫描数据库等缺陷。

CFG架构能够根据约束条件将数据库中的项集划分为多个前缀条件数据库，按照频繁模式增长的方法递归发现频繁项集。条件数据库具有下列性质：如果模式α在f的条件数据库$TDB|f$中是频繁的，则$\alpha\cup f$在$TDB|f$中也一定是频繁的。对于反单调约束C_a，CFG架构给出了将约束用于挖掘的几种策略：(1) 去除不满足约束的单个项目；(2) 如果模式α不满足约束，则不必产生α的条件项目集，也不必产生α的条件数据库$TDB|\alpha$；(3) 如果$\alpha\cup\beta$满足约束，则不必对条件数据库$TDB|\alpha$中的其余部分用C_a进行约束检查，此处β为$TDB|\alpha$中的频繁项目集。对于单调约束的解决算法同Apriori类算法相似，都需要产生所有的频繁项集(无论频繁项集是否满足约束条件)。但是有一点不同，如果一个频繁项集满足约束，则CFG架构只需测试以该项集作为前缀的频繁项集，而不需要对所有频繁项集进行约束测试。CFG架构不产生候选频繁项集，只需两次扫描数据库。此外，由于采用了前缀条件数据库的概念，因此大大减少了满足约束条件的频繁项集的搜索空间。但会占用大量的内存。

Zaki提出的Eclat算法[7]采用了垂直数据库结构、概念格划分和自底向上的频繁集搜索方法。利用基于前缀的等价关系将概念格划分为较小的子概念格。与Apriori算法相比，Eclat算法具有以下优点：(1) 无需复杂的Hash数据结构。(2) 在生成候选项集后，不需再进行剪枝。(3) 可以在内存中独立处理每一个子网格，扫描数据库的次数较少，挖掘效率较

*)国家杰出青年自然科学基金(69928201)资助。杜剑峰　工程师，硕士生，从事数据挖掘研究。李　宏　副教授，从事KDD、分布式处理技术和信号处理研究。陈松乔　教授，博士生导师，从事决策支持和软件工程研究。陈建二　长江学者，从事计算机算法复杂性研究。

高。与 FP-growth 算法相比，Eclat 算法占用的内存比较小。

本文在 Eclat 算法的基础上，提出了一种新的单调和反单调约束条件下关联规则的挖掘算法。算法综合了 CAP 算法和 CFG 架构的特点，避免了 Apriori 算法和 FP-growth 算法的缺陷，是一种十分有效的解决基于单调和反单调约束条件下的关联规则挖掘算法。

2 相关概念与描述

2.1 约束条件分析

目前，基于约束条件的关联规则挖掘所指的通常都是规则约束。规则约束可以分为5类：反单调约束、单调约束、简洁性约束、可转变的约束和不可转变的约束。其中应用最广泛的约束是反单调约束和单调约束。

定义1(反单调约束) 给定一个约束条件 C_a，如果一个项集 S 不满足 C_a，S 的任何超集也不可能满足 C_a，则称约束条件 C_a 是反单调的。

定义2(单调性约束) 给定一个约束条件 C_m，如果一个项集 S 满足 C_m，S 的任何超集也满足 C_m，则称约束条件 C_m 是单调的。

利用反单调约束和单调约束可以减少挖掘的时间和空间，提高挖掘效率。如著名的 Apriori 性质(频繁项集的任何非空子集也必然是频繁的)就是反单调的。这个性质用于 Apriori 算法的每次迭代，可以大大减少候选项集的个数，压缩关联规则的搜索空间。

其它几种约束的概念请参见文[5,6]。

2.2 Eclat 算法介绍[7]

Eclat 算法采用了垂直数据库结构。所谓垂直的 tid 列表数据库是指数据库中的每一条记录由一个项目及其所出现过的所有事务记录的列表构成。这样使得任何一个频繁集的支持度都可以通过一些 tid-list 的交集运算求得。表1的事务数据库是本文的示例数据库，表2是它的 tid-list 形式。

表1 示例数据库

Transaction	Items
1	A,C,T,W
2	C,D,W
3	A,C,T,W
4	A,C,D,W
5	A,C,D,T,W
6	C,D,T

表2 示例数据库的 tid-list 形式

Items	Transaction
A	1,3,4,5
C	1,2,3,4,5,6
D	2,4,5,6
T	1,3,5,6
W	1,2,3,4,5

Eclat 算法在概念格理论的基础上，利用基于前缀的等价关系将搜索空间(概念格)划分为较小的子空间(子概念格)。采用了自底向上(对于等价类)的频繁集搜索方法，将等价类分解为较小的等价类，通过枚举每一个等价类中所有原子项的交集来逐层递归求出所有的等价类。每个等价类都由一些原子项组成。相关的概念及定理请参见文[7]。

如对于表1的实例数据库，令 $P(I)$ 是它的完全概念格，I 是全部项的集合。$P(I)$ 被划分为[A]，[C]，[D]，[T]，[W]5个等价类(子概念格)，并且是有序的。Eclat 算法采用基于前缀的分类，即[X]等价类表示所有前缀为 X 的项集构成的子概念格。如[C]={C,CD,CT,CW,CDT,CDW,CTW,CDTW}。图1描述了等价类[C]的实际挖掘过程和其中用到的原子。

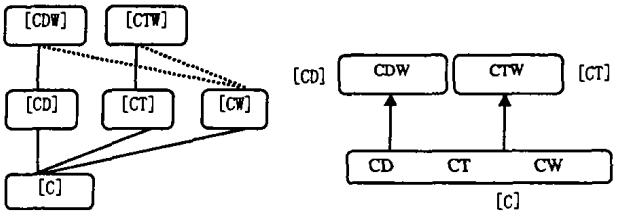

图1(a)等价类[C]的实际挖掘过程 图1(b)等价类[C]的原子

3 基于 Eclat 算法的单调反单调约束关联规则挖掘算法实现

引理1[7] 由等价关系 θ_k 生成的等价类 $[X]_{\theta_k}$ 是 $P(I)$ 的1个子概念格。

定理1 Eclat 算法中的概念格和等价类划分方法可以确保利用自底向上的搜索方法遍历所有满足约束条件的频繁项集。

证明：由引理1可得出，对于数据垂直分布，可以采用概念格和等价类划分生成完备的频繁项集 L。令所有满足约束条件的频繁项集的集合为 S，根据约束的定义可知：$S \subseteq L$。因此，可以将约束整合到等价类的自底向上的搜索过程中，在发现频繁项集的同时进行约束条件的检验。

3.1 EclatA：反单调约束关联规则挖掘算法

例1 示例数据库如表1所示，设支持度阈值为3，约束条件 $C = \mathrm{sum}(S) \leqslant 180$ 为反单调约束，各项的值分别为：$\langle A(50), C(150), D(10), T(20), W(30)\rangle$。图2描述了实际的挖掘过程。图中带圈的项集为满足反单调约束的频繁项集，其它为非频繁项集。

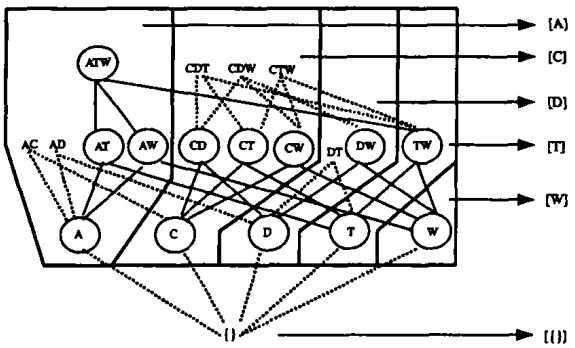

图2 反单调约束示例的实际挖掘过程

由例1可以看出：根据反单调约束和 Eclat 算法的特点，只要在自底向上的搜索算法中验证支持度计数时，同时对约束条件进行判断，就可以挖掘出所有满足反单调约束条件的频繁项集。详细的算法描述如下：

算法1 EclatA 算法

```
Bottom_Up(S);
For all atoms Ai∈S do
  Ti=∅;
  For all atoms Aj∈S, with j>i do
    R=Ai∪Aj;
    L(R)=L(Ai)∩L(Aj);
    If σ(R)≥min_sup and C(R)==true then
      Ti=Ti∪{R}; F|R|=F|R|∪{R};
```

```
  End
End
For all Ti≠θ do Bottom_Up(Ti);
```

其中，S 为等价类；A_i 为原子；$\sigma(R)$为项集 R 的支持度计数；若 $C(R)==$true，则表明项集 R 满足约束条件 C，若 $C(R)==$false，则表明项集 R 不满足约束条件 C；$F_{|R|}$为输出的频繁项集。

3.2 EclatM：单调约束关联规则挖掘算法

例2 示例数据库如表1所示，设支持度阈值为3，约束条件 $C=\text{sum}(S)\geqslant 100$为单调约束，各项的值分别为：$\langle A(50), C(150), D(10), T(20), W(30)\rangle$。图3描述了实际的挖掘过程。图中带圈的项集为满足单调约束的频繁项集，带方框的项集为不满足单调约束的频繁项集，其它为非频繁项集。

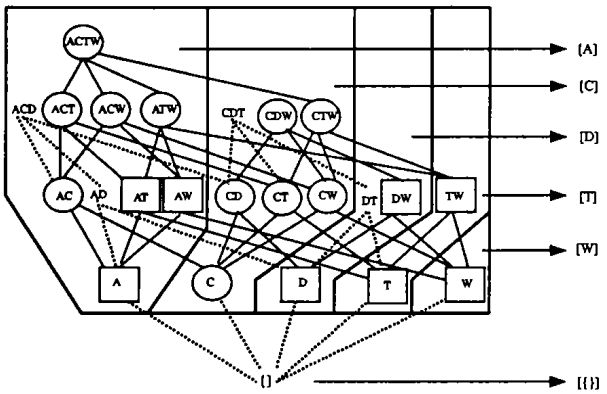

图3 单调约束示例的实际挖掘过程

由例2可以看出，频繁1-项集可以分为两部分。一部分是满足约束条件的项集($\{C\}$)，另一部分是不满足约束条件的项集($\{A\}\{D\}\{T\}\{W\}$)。根据单调约束的特点，如果项集$\{C\}$满足约束条件，则所有以$\{C\}$为前缀的项集均满足该约束条件，因此对于等价类$[C]$，按照 Eclat 算法的自底向上的搜索算法，就可以挖掘出所有满足单调约束条件的频繁项集，而不需要再进行约束条件的判断。对于等价类$[A]$、$[D]$、$[T]$、$[W]$，还需要在 Eclat 算法的自底而上的搜索算法中进行约束条件的判断，但是只要某个原子 X 满足约束条件，则等价类$[X]$就不必再进行约束条件的判断了。例如图3中原子项$\{AC\}$满足约束条件，则等价类$[AC]$中的频繁项集($\{ACT\}\{ACW\}\{ACTW\}$)均不必再进行约束条件的判断。

算法2 EclatM 算法

```
For all atoms Ai∈S do //对最初的等价类进行约束条件的判断，并设置判断标志。
  If C(Ai)==true then Ai.check_flag=1;
  Else Ai.check_flag=0;
End
Call Bottom_Up_M(S);
//具体的搜索方法
Function Bottom_Up_M(S):
For all atoms Ai∈S do
  Ti=θ;
  If Ai.check_flag==1 then //对满足约束条件的等价类不必再进行约束条件的判断
    For all atoms Aj∈S,with j>i do
      R=Ai∪Aj;
      L(R)=L(Ai)∩L(Aj);
      If σ(R)≥min_sup then
        Aj.check_flag=1;Ti=Ti∪{R}; F|R|=F|R|∪{R};
      End
    Else //对不满足约束条件的等价类，还需再进行约束条件的判断
      For all atoms Aj∈S,with j>i do
      R=Ai∪Aj;
      L(R)=L(Ai)∩L(Aj);
      If σ(R)≥min_sup and C(Aj)==true then
        Aj.check_flag=1;Ti=Ti∪{R}; F|R|=F|R|∪{R};
      Else if σ(R)≥min_sup and C(Aj)==false then
        Aj.check_flag=0;Ti=Ti∪{R};
    End
  End
  For all Ti≠θ do Call Bottom_Up_M(Ti);
```

4 实验结果

为了验证算法的性能，用 VC++6.0对 EclatA 算法和 CAP 算法，以及 EclatM 算法和 AprioriM 算法(通过一个后处理的步骤对 Apriori 算法的结果按约束条件进行过滤，这种方法命名为 AprioriM 算法)进行了对比测试。操作系统为 Windows2000 Server，CPU 为 P4 1.8GHz，内存为512MB。实验数据集采用文[2]的合成数据方法。用 DB 表示合成的交易数据库；T 表示平均事务长度；I 表示最大的潜在频繁项集的平均长度；L 表示最大的潜在频繁项集的数量；N 表示项目的数量；D 表示事务数量。实验结果分别如图4和图5所示。

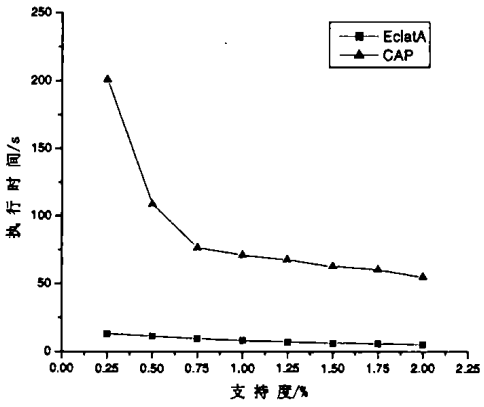

图4 (a)基于最小支持度的扩展性能

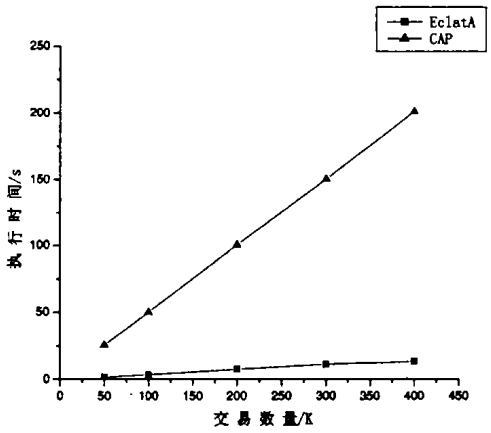

图4 (b)基于交易数量的扩展性能

图4(a)描述了 EclatA 算法和 CAP 算法基于最小支持度的扩展性能，使用的数据集为 T10I8D400K，$L=2000$，$N=1000$。用8个不同的最小支持度阈值(0.25%，0.5%，0.75%，1%，1.25%，1.5%，1.75%，2%)进行测试。约束条件为：$C=\text{sum}(S.price)\leqslant 500$。图4(b)描述了上述两种算法基于交易数量的扩展性能，最小支持度阈值为0.25%，使用了5个不同的数据集(T10I8D50K，T10I8D100K，T10I8D200K，T10I8D300K，T10I8D400K)进行测试，$L=2000$，$N=1000$。约束条件为：$C=\text{sum}(S.price)\leqslant 500$。

图5(a)描述了 EclatM 算法和 AprioriM 算法基于最小支持度的扩展性能，使用的数据集和最小支持度阈值与图4(a)

(下转第166页)

时间的随机影响，每个测试项均重复10，并取其中最小值作为有效结果。通过以上的实验数据，我们可以看到，混合的填充算法在处理较为复杂的数据量大的文件时是有优势的。

123456890
重庆大学

123456890
重庆大学

图2

该算法成功地运用到气动或激光等雕刻制版系统中，因为弓字形的填充轨迹更符合雕刻机雕刀行走的原理，因此在实际的运用中大受欢迎。

将该算法应用到气动雕刻机制版上效果更为明显。

图2的下半部分即为混合算法的填充效果在屏幕上画出来的效果，与雕刻出来的实际图形效果一致。

表1 不同扫描算法该图的完成时间(从计算到完成制版的时间)

算法名称	传统扫描算法	活性边表算法	混合算法
时间(毫秒)	3277	3081	2789

小结 结果表明，本文的混合填充算法利用对多边形数据进行取整，以及合理的数据结构，新的填充算法，可以在处理复杂的多边形填充问题上改进时间效率，改进填充出来的真实度。在激光制版系统、图形刻绘系统中经过实际检测有良好的效果。

参考文献

1 孙家广，杨长贵. 计算机图形学(新版). 清华大学出版社，1995
2 羊四清，李思昆. 改进的扫描线多边形填充算法的研究. 数学理论及应用，1999(6)
3 甘泉. 通用扫描线多边形填充算法. 计算机工程与应用，2000(2)

(上接第144页)

中使用的相同。约束条件为：$C=\mathrm{sum}(S.price)\geqslant 500$。图5(b)描述了上述两种算法基于交易数量的扩展性能，最小支持度阈值为0.5%，使用的数据集与图4(b)中使用的相同。约束条件为：$C=\mathrm{sum}(S.price)\geqslant 500$。

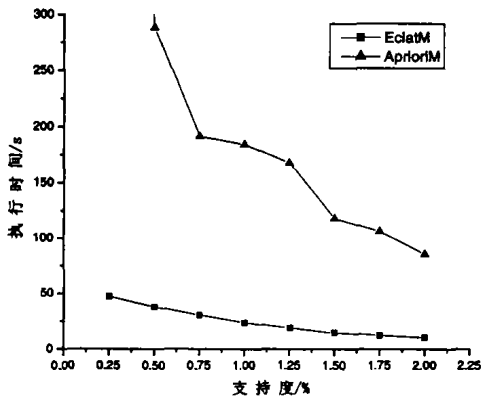

图5 (a)基于最小支持度的扩展性能

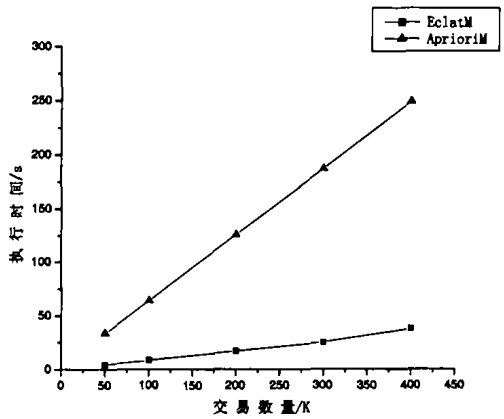

图5 (b)基于交易数量的扩展性能

从上述实验结果可以看出：

(1) EclatA 算法(EclatM 算法)基于最小支持度的扩展性能要优于 CAP 算法(AprioriM 算法)。主要体现在两个方面：一个是 EclatA 算法(EclatM 算法)的执行时间要远远少于 CAP 算法(AprioriM 算法)，另一个是当支持度阈值比较小时，CAP 算法(AprioriM 算法)的执行时间曲线的变化程度要比 EclatA 算法(EclatM 算法)剧烈。

(2) EclatA 算法(EclatM 算法)基于交易数量的扩展性能要优于 CAP 算法(AprioriM 算法)。EclatA 算法(EclatM 算法)和 CAP 算法(AprioriM 算法)均随事务数量的增长而线性增长，但是后两种算法的增长速度要远快于前两种算法。

总之，EclatA 算法(EclatM 算法)具有良好的扩展性，这是由于 Eclat 算法具有 Apriori 算法无法比拟的优越性能。此外，EclatM 算法是在挖掘过程中就进行了约束条件的检测，而 AprioriM 算法是在挖掘出所有频繁项集后再用约束条件来剪裁，这也是反映算法性能的一个重要因素。

结语 本文在 Eclat 算法的基础上，提出了一种新的单调和反单调约束条件下关联规则的挖掘算法。算法采用了垂直数据库结构，利用基于前缀的等价关系将概念格划分为较小的子概念格，在自底向上的频繁集搜索过程中，同时进行约束条件的检验。进一步的工作是如何将这种挖掘算法扩展到多种约束条件下，如简洁性约束和可转变约束。此外，如何在分布式环境下，利用概念格理论和等价类划分方法进行约束关联规则的挖掘也待进一步研究。

参考文献

1 Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases. In: Proc. of the ACM-SIGMOD Conf. on Management of Data, 1993. 207～216
2 Agrawal R, Srikant R. Fast algorithms for mining association rules[A]. In: Proc. of the 20th Intl. Conf. on Very Large Databases[C], Santiago, Chile, 1994. 487～499
3 Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation[A]. In: Proc. of the ACM-SIGMOD Conf. on Management of Data[C], Dallas, ACM Press, 2000. 1～12
4 Ng R T, Lakshmanan L V S, Han J, Pan A. Exploratory mining and pruning optimizations of constrained association rules[A]. In: Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'98)[C], Seattle, Washington, 1998. 13～24
5 Pei J, Han J. Can we push more constrained into frequent pattern mining? [A]. In: Proc. 2000 Int. Conf. Knowledge Discovery and Data Mining(KDD'00)[C], Boston, MA, 2000. 350～354
6 Pei J, Han J, Lakshmanan L V S. Mining frequent pattern with convertible constrains [A]. In: Proc. 2001 Int. Conf. Data Engineering(ICDE'01) [C], Heidelberg, Germany, 2001. 332～433
7 Zaki M J. Scalable algorithms for association mining [J]. IEEE Trans. Knowledge and Data Engineering, 2000, 12(3): 372～390
8 高飞，谢维信. 发现含有第一类项目约束的频繁集的快速算法[J]. 计算机研究与发展，2001，38(11)：295～301