多层次关联规则的有效挖掘算法*

程继华 施鹏飞

(上海交通大学图象处理与模式识别研究所 上海 200030)

摘要 数据挖掘(Data Mining)被认为是解决"数据爆炸"和"数据丰富,信息贫乏(Data Rich and Information Poor)"的一种有效方法.关联规则(Association Rules)是数据挖掘的重要研究内容.提出了多层次关联规则的挖掘算法——AR_SET,利用集合"或"、"与"运算求解频繁模式(Frequent Itemset),提高了挖掘的效率和速度.实验结果表明,算法AR_SET是有效的,并对AR_SET算法的几个变种进行了讨论.

关键词 数据挖掘,关联规则,概括,具体.

中图法分类号 TP311

数据挖掘(Data Mining),也称数据库中知识发现KDD(knowledge discovery in database),是从大量原始数据中挖掘出隐含的、有用的、尚未发现的信息和知识(如知识规则,限制等)[1],被认为是目前解决"数据爆炸"和"数据丰富,信息贫乏(Data Rich and Information Poor)"的一种有效方法.

关联规则(Association Rules)由R. Agrawal等人提出,是KDD研究的重要内容.给定一个事务集合(Transaction Set),每个事务由一组项目(Itemset)组成,关联规则表示为$A \Rightarrow B(sup, conf)$的形式,其中$A$,$B$是项目集合,$sup$是支持率,$conf$是可信度,表示在事务集合中,有$sup$部分事务支持项目集合$A$,支持$A$的事务中有$conf$部分也支持项目集合$B$.[2]例如,5%的顾客买"家佳牌"面包,买"家佳牌"面包的顾客8%也买"家佳牌"黄油,表示为"家佳牌"面包⇒"家佳牌"黄油(5%,8%).关联规则的应用包括商场的顾客购物分析、商品广告邮寄分析、网络故障分析等.[2~4]

关联规则只用原始数据表示,由于支持率低,难于表达普遍的数据关联关系.通常,项目是分类的(Taxonomies)[3,4],例如,饮料分为可乐类、矿泉水类等,可乐类可分为百氏可乐和可口可乐等.这样的层次关系可看成从下到上的概括或抽象(Generalization)以及从上到下的具体(Specification).关联规则中的项目用归纳的项目表示,易于理解、获得大的支持率和表达普遍的数据关联关系.

Agrawal等人提出了AIS[2],Apriori和AprioriTid[6],Cumulate和Stratify[3]等算法,Houtsma等人提出了SETM[6],Park等人提出了DHP[7]算法,挖掘关联规则.Han等人在面向属性归纳(Attribute_Oriented Induction)的基础上[8],提出了多层次关联规则的挖掘算法ML_T2L1[4]等,先求出高概括层的频繁模式(Frequent Itemset),逐渐具体化,挖掘低概括层的频繁模式,最后由频繁模式求解关联规则.为提高挖掘的效率,AprioriTid算法在事务中记录支持的模式;ML_T2L1算法也采用了类似AprioriTid算法的结构,在事务中记录支持的概括层的模式,减少了冗余匹配和对事务数据的访问.

本文提出的多层次关联规则挖掘算法AR_SET(multi_level association rules with SET operation),利用集合"或"和"与"运算求解频繁模式,避免了模式的匹配运算,提高了挖掘的效率和速度.在求候选频繁模式的同时,求出候选频繁模式的支持率,而不是先求候选频繁模式集合,再求支持率,提高了内存的使用效率.算法的实验结果表明,算法是有效的.本文还对AR_SET算法的几个变种进行了讨论.本文第1节描述了关联规则的挖掘问题,提出了挖掘算法AR_SET和几个变种;第2节对算法的性能分别进行了分析研究;第3节是结论.

1 关联规则的挖掘问题

1.1 多层次关联规则的挖掘

概括的层次关系用有向无环图(DAG)表示[3],顶点表示项目或概括的项目(即分类).若从顶点A到B之间有弧,

* 本文研究得到国家自然科学基金资助.作者程继华,1964年生,讲师,主要研究领域为数据挖掘,计算机可视化技术.施鹏飞,1939年生,教授,博导,主要研究领域为图象处理,模式识别,计算机视觉,智能技术与系统.

本文通讯联系人:程继华,上海200030,上海交通大学图象处理与模式识别研究所

本文1997-07-07收到原稿,1997-11-24收到修改稿

称 A 是 B 的祖先，B 是 A 的子孙. 祖先关系具有传递性. 记 I 为项目集合(*Itemsets*)，GI 为 I 中的项目概括后的概括项目集合，即 GI 为 I 中的项目的祖先组成的集合. 从根节点到叶节点的长度称为概括的层次.

事务(Transaction)表示为{Tid，$\langle A_1,A_2,\ldots,A_p\rangle$}，其中 Tid 为事务标识号，全局唯一，项目 $A_i\in I(i=1,2,\ldots,p)$.

定义 1. 模式 P 定义为：$\{A_1,A_2,\ldots,A_k\}$，$A_i\in GI\cup I(i=1,2,\ldots,k)$，称模式 P 的长度为 k. 对于事务 $t=$ {Tid，$\langle B_1,B_2,\ldots,B_p\rangle$}，若$\forall A_i(i=1,2,\ldots,k)$，$\exists B_j(j=1,2,\ldots,p)$，$A_i=B_j$ 或 A_i 是 B_j 的祖先，称事务 t 支持模式 P，或事务 t 包含模式 P.

例 1：一个顾客购买商品 apple，bread，butter，在事务集合中表示为{101，$\langle apple,bread,butter\rangle$}，其中 101 为事务标识号，全局唯一.

模式 P 在事务数据集合 D 中的支持率(Support) $\sigma(P/D)=\frac{D\text{中包含模式}P\text{的事务个数}}{D\text{中事务总个数}}$，规则 $A\Rightarrow B$ 的可信度(Confidence) $\Psi(A\Rightarrow B/D)=\frac{\sigma(A\cap B/D)}{\sigma(A/D)}$，即事务支持 A 时支持 B 的条件概率，其中 A，B 均为模式.[2,4,5]

支持率表示在事务数据集合中模式出现的概率，可信度表示规则的前件与后件的依赖程度. 为挖掘有效的关联规则，必须定义最小支持率 $\sigma_{\min}$ 与最小可信度 $\Psi_{\min}$.[2]

定义 2. 若模式 P 的支持率 $\sigma(P/D)>\sigma_{\min}$，称 P 是频繁模式(Frequent). 若 $\Psi(A\Rightarrow B/D)>\Psi_{\min}$，并且 $\sigma(A\cap B/D)>\sigma_{\min}$，称关联规则 $A\Rightarrow B$ 为强的(Strong).[4]

挖掘关联规则即是对于最小支持率 $\sigma_{\min}$ 与最小可信度 $\Psi_{\min}$ 求解强的规则.

定义 3. 在事务数据集合 D 上，支持率大于最小支持率 $\sigma_{\min}$ 的长度为 i 的模式，称为 i 频繁模式. 全部 i 频繁模式的集合，称为 i 频繁模式集合，记为 L_i.[1] $\bigcup_i L_i$ 称为频繁模式集合，记为 L.[5]

定义 4. 长度为 i 的模式 P，去掉任意一个项目所形成的长度为 $i-1$ 的模式，称为 P 的 $i-1$ 子模式.

引理 1. i 频繁模式 P 的 $i-1$ 子模式也是频繁模式.[2,4]

定理 1. 给定长度为 i 的模式 P 和 P 的任意两个 $i-1$ 子模式 A，B，同时支持 A 和 B 的事务组成了 P 的支持事务集合.

证明：由定义 4，P 的 $i-1$ 子模式 A，B 包含了 P 的 i 个项目. 由定义 1，支持 A 和 B 的事务也支持 P. 由引理 1，支持 P 的事务也支持 A，B. 所以，P 的支持事务集合是由同时支持 A 和 B 的事务组成的. □

Agrawal 等人给出了由频繁模式集合构造关联规则的算法.[6]以下着重讨论求解频繁模式集合 L.

1.2 关联规则挖掘算法 AR_SET

在 AR_SET 算法中，模式由模式包含的项目集合和模式的支持事务集合组成，模式 P 表示为$\langle Pattern,Support\rangle$，*Pattern* 为模式包含的项目集合，项目按字典序排列，记为 $P.Pattern$；*Support* 为模式的支持事务集合，由事务标识号 *Tid* 组成，按升序排列，记为 $P.Support$($P.Support$ 中的事务数目记为 $|P.Support|$)，模式根据 *Hash* 函数确定其在 L_i 中的位置[7]，以实现快速查找.

记 $|D|$ 为事务数据集合 D 的事务数目，$|I|$ 为项目集合中项目的数目，$|L_i|$ 为 L_i 中的模式数目，$L_i[m]$ 为 L_i 中的第 m 个 i 频繁模式.

AR_SET 算法：

```
Input：事务数据集合 D，最小支持率 σmin
Output：频繁模式集合 L
Method：
    L1={(A,S)|A∈I∪GI，S 为支持 A 的事务集合，|A.Support|/|D|>σmin}
    k=2
    While Lk-1<>{} Do {
        Lk={}
        For n=1 to |Lk-1|-1 Step 1
            For m=n+1 to |Lk-1| Step 1
                If (|Lk-1[m].Pattern∪Lk-1[n].Pattern|= k) and
                  (Lk-1[m].Pattern 中项目与 Lk-1[n].Pattern 中项目无子孙关系)
                Then {P.Pattern=Lk-1[m].Pattern∪Lk-1[n].Pattern
                      If (P.Pattern∉Lk)
                      Then {P.Support=Lk-1[m].Support∩Lk-1[n].Support
                            If |P.Support|/|D|>σmin Then Lk=Lk∪{P}
```

$k=k+1$

}

$L=\bigcup_k L_k$

算法说明:根据引理1和定理1,若模式$P\in L_k$,必存在P的两个$k-1$子模式$M\in L_{k-1}$,$N\in L_{k-1}$,并且P的支持事务集合是M和N的支持事务集合的并集.算法AR_SET根据L_{k-1}中的模式,构造k频繁模式集合L_k:在L_{k-1}中找出只有一个项目不同的模式M和N,作为长度为k的模式$P=M\cup N$的两个$k-1$子模式;如果L_{k-1}中不存在只有一个项目不同的两个模式M和N,由引理1和定理1知,不存在k频繁模式.算法AR_SET在无k频繁模式时,终止.

算法AR_SET首先访问事务数据集合,计算出1频繁模式集合L_1,然后循环执行:如果L_{k-1}中两个$k-1$频繁模式$L_{k-1}[m]$和$L_{k-1}[n]$只有一个项目不同,并且无子孙关系,求出长度为k的模式$P.Pattern=L_{k-1}[m].Pattern\cup L_{k-1}[n].Pattern$,然后计算模式$P$的支持$P.Support=L_{k-1}[m].Support\cap L_{k-1}[n].Support$,如果$P$的支持率大于$\sigma_{\min}$,则$P$为$k$频繁模式,将$P$放入$L_k$中.最后求出频繁模式集合$L=\bigcup_k L_k$.

模式中记录了模式的支持事务集合,在求模式的支持时,只需利用集合的"或"和"与"运算,由于模式的支持事务集合以事务的Tid的升序排列,求集合的"或"和"与"运算的时间是两个集合中元素(事务)的个数与两个整数(事务的标识号Tid)进行比较的时间的乘积:$c*(|L_{k-1}[m].Support|+|L_{k-1}[n].Support|)$($c$为两个整数比较的时间).比常用模式的支持判别(模式的匹配)效率高;另外,AR_SET算法在求候选频繁模式的同时,求出模式的支持,而不是先求候选频繁模式集合,再求候选频繁模式的支持,提高了内存的使用效率.

选择模式对支持事务的记录、频繁模式在L_k中的排序、循环求解频繁模式的步长等的不同处理方法,可提出AR_SET算法的以下几个变种.

1.3 算法AR_SET_K

由于在循环计数k较小时,模式的支持事务集合很大,保持模式的支持事务集合占用较多的存储空间.为节约存储空间,对AR_SET算法作如下修改:当循环变量k大于某个常量C(或当L_k中模式的支持事务集合的元素数目小于某个值)时,模式中才记录支持事务集合,用集合运算求解模式的支持;而当k不大于C(或L_k中模式的支持事务集合中的元素数目大于某个值)时,采用Apriori等算法利用模式的匹配方法求解模式的支持.修改后的算法为AR_SET_K.

1.4 算法AR_SET_STEP

在AR_SET算法中,循环变量k的步长为1.算法AR_SET_STEP作如下修改:频繁模式集合的求解顺序改为:L_i计算$L_{i+1},\ldots,L_{i+i}$;再由$L_{i+1},\ldots,L_{i+i}$求出$L_{i+i+1},\ldots,L_{2*i+1}$;循环变量$k$的步长为$i$.修改后的算法一方面减少了存储访问,另一方面,在求频繁模式的过程中增加了非频繁模式的计算量,算法AR_SET_STEP在存储的访问和频繁模式计算的有效性上,作了折衷.

1.5 算法AR_SET_SORT

在AR_SET算法中,L_{k-1}中的模式根据生成的顺序排列,计算k频繁模式P的两个$k-1$频繁模式:$L_{k-1}[m]$和$L_{k-1}[n]$,它们是L_{k-1}中排列在最前面的两个P的$k-1$子模式,计算P的支持事务集合的时间为$c*(|L_{k-1}[m].Support|+|L_{k-1}[n].Support|)$.显然,$|L_{k-1}[m].Support|+|L_{k-1}[n].Support|$越小,求解$k$频繁模式的时间越少.基于这种考虑,对$L_{k-1}$中的模式按照支持事务集合的大小($|L_{k-1}[n].Support|$)作升序排列,求解$k$频繁模式时,选取支持事务集合最小的两个$k-1$子模式计算模式的支持事务集合.修改后的算法为AR_SET_SORT.

2 算法AR_SET的分析

算法AR_SET所需要的数据存储空间是存储频繁模式包含的项目和支持事务集合的存储空间,为

$$\sum_{P\in L}(|P.Support|+|P.Pattern|).$$

AR_SET算法在求候选频繁模式的同时求出模式的支持,对内存大小的要求不高.与先求候选模式集合,再求候选模式的支持的方法相比,事务集合越大,速度的提高越大.

在Pentium_133,8M内存的环境下的实验结果如图所示.图1给出了在$|I|=50$,事务平均包含的项目个数$=8$,概括层次$=4$,$\sigma_{\min}=2\%$的试验数据下,算法AR_SET的执行时间与事务数据量的关系.从图中可以看出,算法执行时间的上升远远小于数据量的上升.图2给出了在挖掘单层次关联规则(概括层次$=1$)时,事务平均包含的项目个数

Press, 1995. 420～431

5 Agrawal R, Mannila H, Srikant R *et al*. Fast discovery of association rules. In: Fayyad M, Piatetsky-Shapiro G, Smyth P eds. Advances in Knowledge Discovery and Data Mining. Menlo Park, CA: AAAI/MIT Press, 1996. 307～328

6 Houtsma M, Swami A. Set-oriented mining for association rules in relational databases. In: Yu P, Chen A eds. Proceedings of the International Conference on Data Engineering. Los Alamitos, CA: IEEE Computer Society Press, 1995. 25～33

7 Park J, Chen M, Yu P. An effective hash based algorithm for mining association rules. In: Anon ed. Proceedings of the ACM SIGMOD International Conference on Management of Data. New York, NY: ACM Press. 1995. 175～186

8 Han Jia-wei, Cai Yang-dong, Cercone N. Data-driven discovery of quantitative rules in relational databases. IEEE Transactions on Knowledge and Data Engineering, 1993,5(1):29～40

Efficient Mining Algorithm for Multiple-Level Association Rules

CHENG Ji-hua　SHI Peng-fei

(*Institute of Image Processing and Pattern Recognition　Shanghai Jiaotong University　Shanghai*　200030)

Abstract　Data mining is recognized as an efficient method for solving Data-Explosion and Data Rich and Information Poor. Mining association rules is one of the important aspects of data mining. The mining algorithm AR-SET for multiple-level association rules is given in this paper. The algorithm calculates the frequent itemsets by set operation——Union and Disjoin, and speeds up the mining. The experiments show that AR-SET is efficient. Also, some variants of AR-SET are discussed.

Key words　Data mining, association rules, generalization, specification.