一种挖掘多维关联规则的有效算法*)

An Efficient Algorithm for Mining Multidimensional Association Rules

范 明 牛常勇 朱 琰

(郑州大学计算机科学系 郑州 450052)

Abstract In this paper, we propose a new algorithm to discover multidimensional association rules (MAR). Since the number of potential MARs tend to be extremely large, mining MARs poses more challenges on efficient processing than mining intra-dimensional association rules. We tackle this problem by constructing a FDPI-tree (frequent dimensional predicate sets index tree) to combine data cube technique with Apriori method efficiently. Compared with previous approaches which only find the inter-dimension associations, the algorithm we presented here explores both inter-dimension and hybrid-dimension associations simultaneously.

Keywords Data mining, Multidimensional association rules, Data cube

1. 引言

挖掘大型事务数据库中的关联规则是数据挖掘研究的重要课题之一[1]。由于关联规则在商务分析与决策、相关分析、分类等方面具有广泛应用，自提出以来一直受到广泛重视。一维关联规则的挖掘已有不少有效算法(如文[2,3,4]，综述参见文[1])。多维关联规则不仅考虑项集间的关联，而且考虑项集的维约束。这使得挖掘出的规则更具实用性，同时也增加了规则挖掘的难度。

基于规则模板的挖掘[5]，使用元规则限定挖掘的关联规则形式，降低了挖掘难度，但也使得其应用受到一定限制。采用类 Apriori 算法的方法通过求频繁谓词集得到多维关联规则[6]，具有很好的可扩展性，能够处理大量数据，但其处理维谓词的 I/O 开销较大。利用数据立方进行多维关联规则挖掘的算法[7]具有较好的 I/O 性能，特别是当数据立方较小，可以用多维数组有效实现、或者使用元规则对项集加以限制时，其 I/O 性能特别好，但是，由于数据立方不能有效地处理集合值，当事务数据库非常大时，其性能明显下降。

本文将数据立方技术和 Apriori 算法的思想有机地结合在一起，提出一种多维关联规则挖掘的新算法。本文的算法采用数据立方计算频繁维谓词集，而使用与 Apriori 算法类似的方法求多维频繁项集。此外，我们还引入了一种频繁维谓词集索引树结构，以加快多维频繁项集计算速度。这样，本文的算法不仅有较好的 I/O 效率，还具有很好的可扩展性。

2. 问题描述

假定事务数据库 TDB 由事务组成。事务包括三部分，形如(Tid, d_inf, item_set)；其中，Tid 是事务的唯一标识；d_inf 是事务的维信息，它是一个 m 维向量 $(d_1,\cdots,d_m)$，而 d_i 在维(属性) D_i ($1\leqslant i\leqslant m$) 上取值；item_set 是文字(项)的集合。TID 中项的全体记作 $I=\{i_1,i_2,\cdots,i_n\}$。这样，item_set 是 I 的子集。例如，对于购物篮分析，事务可以具有如下形式：

Tid	age	occupation	income	items
		d_inf		item_set

其中，维信息 d_inf 是 3 维向量，包含顾客个人信息。而

T100	26…30	工程师	5k…20k	{个人电脑, word 2000, …}

是一个具体的事务，记录一个特定顾客的一次购物。

事务 t 在每个维 D_i 上都取单个值。下失一般性，我们假定数值属性已离散化为区间值。例如，上面事务中的 age 和 income。每个维 D_i 有一个对应的谓词 p_i，它可以是维名；例如，age。事务 t 使得 $p_i(X,v_i)$ 为真，如果 X=t. Tid 时 $t.D_i=v_i$。item_set 也对应一个谓词，记作 q。对于购物篮分析，q 通常为 buys。事务 t 使得 q(X, items) 为真，如果 X=t. Tid 时 items⊆t . item_set。

*)本文的工作得到河南省自然科学基金和郑州大学科学研究基金的部分资助。

多维关联规则具有如下形式：

$$p_1(X,v_1)\wedge\cdots\wedge p_k\wedge q(X,items1)\Rightarrow q(X,items2)$$
$$[\text{support}=s\%,\text{confidence}=c\%] \quad (1)$$
$$p_1(X,v_1)\wedge\cdots\wedge p_k(X,v_k)\Rightarrow q(X,items)$$
$$[\text{support}=s\%,\text{confidence}=c\%] \quad (2)$$

其中，$k\leqslant m$；support$=s\%$和confidence$=c\%$表示规则的支持度和置信度分别为$s\%$和$c\%$。规则的支持度反映该规则涵盖了多大比例的事务，它是使规则两端均为真的事务在全体事务中所占的百分比。规则的置信度反映规则的可信程度，它是使规则右端为真的事务在使得规则左端为真的事务中所占的百分比。(1)式包含重复谓词，称作混合维关联规则，而(2)式不包含重复谓词，称作维间关联规则。下面是一个多维关联规则的例子：

$$age(X,\text{"21}\cdots\text{25"})\wedge occupation(X,\text{"}student\text{"})\Rightarrow buys(X,\text{"}laptop\ computer\text{"})$$
$$[\text{support}=4\%,\text{confidence}=60\%]$$

该规则表明4%的事务是年龄21…25的学生购买便携计算机，并且年龄21…25的学生的购物活动60%涉及购买便携计算机。

为使规则提供有用的知识，规则的支持度和置信度应当分别满足预先给定的最小支持度(记作min_sup)和最小置信度阈值(记作min_conf)。这种规则称作强规则。而满足最小支持度阈值的谓词集(项集)称作频繁谓词集(频繁项集)。我们也使用(最小)支持度计数，它等于(最小)支持度乘以事务总数。最小支持度计数记作min_sup_count。

本文旨在给出一种有效的算法，挖掘所有形如(1)和(2)式的强规则。

3. 算法描述

本文的算法首先采用数据立方计算频繁维谓词集，并构造频繁维谓词集的索引树(FDPI树)。然后，使用与Apriori算法类似的方法求多维频繁项集。最后，由多维频繁项集生成多维关联规则。算法的步骤如下：

1. 计算频繁维谓词集。
2. 由频繁维谓词集构造频繁维谓词集索引树。
3. 用修改后的Apriori算法求多维频繁项集。
4. 由多维频繁项集生成多维关联规则。

由多维频繁项集生成多维关联规则的方法与已有的算法相同，下面，我们详细地讨论步骤1，2和3。

3.1 计算频繁维谓词集

首先，构造维基本数据立方。数据立方的维取自任务相关事务数据库的m个维，其度量为count。设维D_i($i=1,\cdots,m$)有n_i个不同值，$v_{i1},\cdots,v_{in_i}$。立方单元$(v_{1i_1},\cdots,v_{mi_m})$存放的度量值为在维$D_i$上取值$v_{ii}$($i=1,\cdots,m$)的事务个数。

然后，使用数据立方计算技术，计算该数据立方的所有$m-D$，$(m-1)-D$，…，1－D立方体，注意：$m-D$立方体就是基本数据立方。

最后，由立方体中度量值大于最小支持度计数阈值min_sup_count的立方单元构造频繁谓词集的集合L^D：考虑一个$k-D$($1\leqslant k\leqslant m$)数据立方，如果其立方单元$(v_{1i_1},\cdots,v_{ki_k})$的度量大于min_sup_count，则我们有$\{p_{i_1}(X,v_{1i_1}),\cdots,p_{i_k}(X,v_{ki_k})\}\in L^D$。

容易明白以下引理成立：

引理1　L^D包含并且仅包含所有的频繁维谓词集。

3.2 构造频繁维谓词集索引树

频繁维谓词集索引树(FDPI树)T是一棵m层树，其第i层对应于维D_i。T的结点是变长的，若结点N包含k个值，则它有k个指针，分别与k个值相关联。T的一条由根到结点N的路径用与指针关联的值序列表示，T按如下方法构造：

T的根结点由维D_1上的频繁谓词构造。若$\{p_1(X,v)\}\in L^D$，则值v在根结点中。此外，一个特殊的值any在根结点中。

假定已经构造了T的第i($1\leqslant i<m$)层结点。T的第$i+1$层结点由维D_{i+1}上的频繁谓词构造。设N是第i层上的结点，它有k个值$v_1,\cdots,v_k$，则在第$i+1$层创建k个结点$N_1,\cdots,N_k$，作为N的子女，分别用与$v_1,\cdots,v_k$相关联的指针指向它们。考虑N_1，设由根到N_1的路径为$(w_1,\cdots,w_k)$；其中，$w_k=v_1$。又设该路径的值序列中有k'个值不等于any。如果存在形如$\{\cdots,p_j(X,v)\}$的包含$k'+1$个元素的频繁维谓词集$s\in L^D$，使得只要$w_j\neq any$，就有$p_j(X,w_j)\in s$，则值v在结点N_1中。此外，一个特殊的值any在结点N_1中。结点$N_2,\cdots,N_k$用类似的方法产生。

如果由根到叶子结点N的路径值序列为(any，any，…，any)，则删除结点N中的值any。最后，与T的叶子结点中的值相关联的每个指针都指向一个桶。

例1　假定我们有频繁维谓词集的集合

$L^D=\{\{p_1(X,A1)\},\{p_1(X,A2)\},\{p_2(X,B1)\},\{p_2(X,B2)\},\{p_3(X,C1)\},\{p_3(X,C2)\},$
$\{p_1(X,A1),p_2(X,B1)\},\{p_1(X,A2),p_2(X,B2)\},\{p_1(X,A2),p_3(X,C2)\},\{p_2(X,B2),p_3(X,C2)\},$
$\{p_1(X,A2),p_2(X,B2),p_3(X,C2)\}\}$

对应的FDPI树如图1所示。由根到桶的11条路径分别对应L^D中的11个频繁维谓词集。

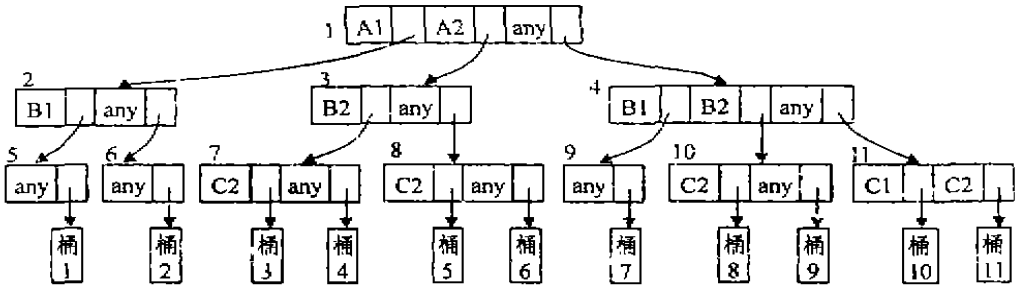

图 1　一棵包含 3 个维谓词的频繁维谓词集索引树

为了更清楚地看出由根到桶的路径与 L^D 中的频繁维谓词集的 1-1 对应关系，我们可以用如下方法扩充频繁维谓词集：对于频繁维谓词集 s，如果 $p_i(X,any)$ 不在 s 中出现，则将 $p_i(X,any)$ 添加到 s 中。$p_i(X,any)$ 意指 X 代表的事务在 D_i 上取任意值，$p_i(X,any)$ 均为真。这相当于不考虑维 D_i 上的取值，因此这种处理不改变频繁维谓词集的语义。例如，例 1 中的 $\{p_1(X,A1)\}$ 被扩充为 $\{p_1(X,A1),\ p_2(X,any),p_3(X,any)\}$，它对应于由根到桶 2 的路径。一般地，我们有如下引理：

引理 2　由 FDPI 树的根到桶的每条路径对应 L^D 中的一个频繁维谓词集，反之亦然。换言之，桶与 L^D 中的频繁维谓词集是 1-1 对应的。

3.3　求多维频繁项集

在多维事务数据库中，每个事务中的项集被事务中的维值约束。这样，频繁 k-项集也要受维值的约束。此外，只有那些被 L^D 中的频繁维谓词集约束的频繁 k-项集才能与频繁维谓词集一起形成多维频繁项集，因此我们只需要考虑 L^D 中频繁维谓词集代表的维值约束。由于 FDPI 树中的桶与 L^D 中的频繁维谓词集是 1-1 对应的，我们可以用桶存放对应的频繁维谓词集约束下的候选项集和频繁项集，而用 FDPI 树结构由给定的事务确定应对哪些桶中的候选项集计数。设频繁维谓词集索引树中有 n 个桶 $b_1,\cdots,b_n$，求多维频繁项集采用如下修改的 Apriori 算法：

```
for (i=1;i≤n,i++)
  // 产生每个桶的频繁1-项集 b_i.L_1
find frequent_1_items_set b_i.L_1;
k=1;
while(exists b_i.L_k≠∅){
  k++;
  for(i=1; i≤n;i++)
    // 对于每个桶，产生候选 k-项集 b_i.C_k
    if (b_i.L_{k-1}≠∅)
      b_i.C_k=aprori_gen(b_i.L_{k-1},min_sup_count);
  for each transaction t∈TDB{
      //扫描 TDB，进行计数
    C_t =subset(t.item_set);
      // 得到 t.items 的子集
    b_set=get_buckets(t.d_inf);
      //得到需进行计数处理的桶
    for each b_i∈b_set
      //对相关的桶进行计数
      for each candidate c∈b_i.C_k
        if(c∈C_t)c.count++;
    }
    for(i=1;i≤n;i++)
      // 对于每个桶，产生频繁 k-项集 b_i.L_k
      b_i.L_k={c∈b_i.C_k|c.count≥min_sup_count};
  }
for (i=1,i≤n;i++)
  // 得到每个桶所有频繁项集 b_i.L
  b_i.L=∪_{j=1}^{k} b_i.L_j
```

其中，aprori_gen 和 subset 过程与 Apriori 算法相同，限于篇幅，本文从略。

get_buckets 过程由事务的维值约束确定对应的桶。对于给定事务的维值序列 $(v_1,\cdots,v_m)$，可以通过检索 FDPI 树实现。需要注意的是：当到达第 i $(i=1,\cdots,m)$ 层结点 N 时，如果 v_i 不在 N 中，则沿 any 对应的指针向下搜索；否则需要同时沿 v_i 和 any 对应的指针向下搜索。这样，搜索的结果是一个桶集合，而不是单个桶。

例2　考虑例1的 FDPI 树，设事务的维值向量为 (A2,B1,C1)。在根结点，我们找到 A2，沿 A2和 any 对应的指针向下搜索，分别到达结点3和4。在结点3，我们找不到 B1，沿 any 对应的指针向下搜索，到达结点8。结点8不包含 C1，沿 any 对应的指针向下，得到桶6。再考虑结点4，它包含 B1，我们需要沿 B1和 any 对应的指针同时向下搜索，到达结点9和11。由它们向下，分别得到桶7和桶10。最终，get_buckets 将返回{桶6，桶7，桶10}。

get_buckets 过程的实现是简单、琐碎的，限于篇幅，本文从略。

注意，产生每个桶的频繁1-项集 $b_i.L$ 写成循环是为了便于理解。事实上，它可以通过一次扫描事务数据库完成。

最后，需要指出的是：为得到多维频繁项集，我们还需要求桶中的每个频繁项集与该桶对应的频繁维谓词集的并。这可以在产生多维关联规则时完成，而不需要附加的操作。

对上述讨论形式化，容易证明：

引理3　修改的 Apriori 算法可以正确地产生所有多维频繁项集。

4. 性能分析

本文算法的正确性是引理3的简单推论。下面我们分析算法的性能，对于大型事务数据库的多维关联规则挖掘算法，其性能主要由 I/O 效率决定。

算法的第1步为计算频繁维谓词集构造数据立方，只需要扫描一次事务数据库。当维数不太大，并且每维的不同值个数不太多时，数据立方可以用内存中的多维数组实现。实践中，频繁维谓词集的数量并不大。这样，算法的第2步由频繁维谓词集构造 FDPI 树可以在内存实现，而不需要附加的 I/O 开销。算法的第3步使用修改的 Apriori 算法求所有的多维频繁项集。如果频繁项集的最大长度为 k，则最多需要 k+1 次扫描事务数据库。这样，总共需要扫描 k+2 次事务数据库。如果扫描一次事务数据库需要 n 次 I/O 操作，则共需要 (k+2)n 次 I/O。

相比之下，采用类 Apriori 算法的方法求频繁谓词集至少需要扫描 k+m+1 次事务数据库，从而需要 (k+m+1)n 次 I/O。利用数据立方进行多维关联规则挖掘的算法只需要扫描一次事务数据库，以构造数据立方。但是，由于包含项信息的数据立方比仅包含维信息的数据立方大得多，它不能用内存数组实现。这样，它构造和使用数据立方的 I/O 开销也不容忽略。并且由于数据立方的稀疏性，其存储开销也是非常大的。

结束语 本文，我们提出了一种挖掘多维关联规则的新算法。本文的算法将数据立方技术和 Apriori 算法的思想有机地结合在一起，既利用了数据立方能够有效处理多维数值度量的优点，又保持了 Apriori 算法的可规模性。这使得本文的算法具有较好的整体性能。Apriori 算法的各种变形与改进都不难，稍加修改用来求多维频繁项集。本文的算法不仅能够挖掘维间关联规则，而且能够挖掘混合维关联规则。

数据立方有利于多个抽象层上的挖掘。如何将本文的算法思想用于多个抽象层的多维关联规则挖掘，如何将数据立方技术与其它关联规则挖掘方法（如文[4]的方法）结合，需要进一步深入研究。我们正在探讨这些问题，希望能够在不久的将来报告我们的结果。

参 考 文 献

1 Han J, Kamber M. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2000. 225～277

2 Agrawal R, Srikant R. Fast algorithms for mining association rules. In: Proc. 1994 Int'l Conf. on Very Large Data Bases, Sept. 1994. 487～499

3 Park J S, Chen M S, Yu P S. An effective hash-based algorithm for mining association rules. In: Proc. 1995 ACM-SIGMOD Int'l Conf. on Management of Data, 1995. 175～186

4 Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation. In: Proc. 2000 ACM-SIGMOD Intl. Conf. on Management of Data, May 2000. 1～12

5 Fu Y, Han J. Meta-rule-guided mining of association rules in relational databases. In: Proc. 1st Intl. Workshop Integration of Knowledge Discovery with Deductive and Object-Oriented Databases, Dec. 1995. 39～46

6 Srikant R, Agrawal R. Mining quantitative association rules in large relational tables. In: Proc. 1996 ACM-SIGMOD Int'l Conf. on Management of Data, June 1996. 1～12

7 Kamber M, Han J, Chiang J Y. Metarule-guided mining of multidimension association rules using data cubes. In: Proc. 1997 Conf. Knowledge Discovery and Data Mining, Aug. 1997. 207～210

（上接第64页）

们实现了一个 PDA 与 PC 远程文件传输系统。该编程接口还成功地应用于"基于 PDA 和 PC 远程串行通信和神经网络的农业病虫害诊断专家系统"中。实验结果说明，本文提出的协议具有通信效率高、通用性好和可靠性高的特点。

随着 Internet/Intranet 的迅猛发展和智能信息化技术的进一步发展，基于计算机网络的专家系统越来越多。例如，在农业领域方面，北京市农林科学院和国防科技大学已经研究开发出了网络化、构件化农业专家系统开发平台 PAID。我们今后的工作，主要是在目前已经实现的 PDA 与 PC 间远程串行通信的基础上，研究 PDA 和 Internet/Intranet 的接口技术，使 PDA 能远程访问 PC 服务器上的专家系统和实现动态信息更新，以充分利用网络信息资源。

参 考 文 献

1 [美]Joe Campell. 串行通信 C 程序员指南（第二版）. 北京：清华大学出版社，1995

2 陈坚，孙志月. MODEM 通信编程技术. 陕西：西安电子科技大学出版社，1998

3 赵春江，杨宝祝，等. 网络化、构件化农业专家系统开发平台（PAID）的研究与应用. 见：第四届中国计算机智能接口与智能应用学术论文集. 北京：电子工业出版社，1999. 459～465