

基于 MDPI 的多维关联规则算法的研究

彭 硕, 吴 昊

(湖南大学 计算机与通信学院, 湖南 长沙 410082)

摘 要: 多维关联规则是数据挖掘中的一个重要研究方向, 由此提出了一种高效的多维关联规则挖掘算法, 该方法通过引入 MDPI-tree(多维谓词索引树)结构, 有效地将数据立方体技术和频繁项集挖掘算法 FP-Growth 结合起来, 能用于挖掘维间和混合维关联规则. 最后将此算法应用于移动通信交叉销售模型, 通过实验验证算法的有效性和实用性.

关键词: 数据挖掘; 多维关联规则; 数据立方体; MDPI; FP-Growth

中图分类号: TP31

文献标识码: A

文章编号: 1000-7180(2011)01-0078-05

The Research for Multidimensional Association Rules Algorithm Based on MDPI

PENG Shuo, WU Hao

(College of Computer and Communication, Hunan University, Changsha 410082, China)

Abstract: Multidimensional data mining association rules is an important research direction. In this paper, we propose an efficient algorithm for mining multidimensional association rules, which combine data cube technique with FP-Growth efficiently by constructing a MDPI-tree, the algorithm can explore both inter-dimension and hybrid-dimension association rules. Lastly this algorithm is applied to cross-selling model of mobile communication, and we can verify the practicality and effectiveness of the algorithm by experiment.

Key words: data mining; multidimensional association; data cube; MDPI; FP-Growth

1 引言

多维关联规则挖掘主要应用于现代企业的大型事务数据中^[1]. 目前, 主要有两种途径研究多维关联规则挖掘, 一种是扩展维内关联挖掘算法(如: Apriori^[2])直接应用于多维数据库产生多维关联规则^[3], 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, 从整体上来说效率较低. 另一种是利用数据方进行多维关联规则挖掘^[4-5], 此方法具有较好的 I/O 性能, 特别是当数据立方较小, 可用多维数组有效实现, 然而对于大型数据库, 构造及维护相应的数据方是极其昂贵的.

本文设计的多维关联规则算法, 旨在将数据方表示技术和频繁项集挖掘技术结合, 引入 MDPI-tree(多维谓词索引树)结构, 有效地将数据方技术

和 FP 算法结合起来, 其 MDPI-tree 由维信息部分的 MDP-tree^[6] 和项信息部分事务构造的 FP-tree 及其连接构造而成. 此算法不仅具有良好的 I/O 性能, 还具有很好的扩展性.

2 相关理论

2.1 工作数据立方体

给定一个关联规则挖掘任务, 其内容涉及 $d_1, \dots, d_n$ 个维, 并根据用户挖掘需求确定各维的维层次, 然后从数据仓库中通过 OLAP 操作生成数据立方体. 其中每一维包含 $|d_i| + 1$ 个数值, $|d_i|$ 是第 i 维包含的互不相同的维成员个数. 前 $|d_i|$ 行各代表 d_i 中一个互不相同的维成员. 最后一行存储了一个称之为“Sum”的维成员, 记录了它所对应的维的合计值, 这种合计值极大地方便了关联规则挖掘中

支持度的计算.立方体的方格中记录的是对应维成员的频繁度量值,记为 count.这样涉及 $d_1, \dots, d_n$ 维数据的一个关联规则挖掘任务就对应一个 n 维的数据立方体 $Cube(d_1, \dots, d_n | count)$. $d_1, \dots, d_n$ 是立方体的维,是立方体的事实度量 count.

2.2 多维关联规则的问题描述

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 是事务标识; d_inf 是维信息,它是一个 m 维向量 $(d_1, d_2, \dots, d_m)$,而 d_i 是属性维 D_i 上的取值; $item_set$ 是文字(项)的集合. TDB 中项的全体记为 $I = \{i_1, i_2, \dots, i_n\}$. 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
-----	-----	------------	--------	-------

其中 d_inf 是 3 维向量,包括客户的基本信息,形式如下:

100	26...30	工程师	5k...20k	{个人电脑, word2000...}
-----	---------	-----	----------	---------------------

这是一个具体的事务,记录一个特定客户的一次购物,事务 t 在每个维 D_i 上都有取值.不失一般性,我们假设数值属性已经离散化为区间值.例如上面事务中的 age 和 income. 每个维 D_i 有一个对应的谓词 pi ,它可以是维名,例如 age.

多维关联规则有两种多维关联规则又分为维间关联规则和混合维关联规则.维之间的关联规则:规则的主体和头都在不同的维中,如 $age(X, "30-39") \wedge income(x, "42K...48K") \rightarrow buys(X, "computer")$,混合维关联规则:是上维之间和维之内关联规则的合成,其中包含重复的维,如: $age(X, "30-39") \wedge buys(X, "software") \rightarrow buys(X, "computer")$.

3 基于 MDPI 的多维关联规则算法

本算法在事务数据库的挖掘步骤如下:

对维信息部分构造数据立方体,将数据立方体信息压缩到 MDP-tree^[7];

通过 MDP-tree 计算频繁谓词集;

构建 MDPI-tree,对事务数据库中维信息与项信息建立关联;

计算多维频繁项集.

3.1 构建 MDP-tree

MDP-tree 与 FP-tree 具有类似的结构,描述如下:

(1) MDP-tree 包含一个标记为"root"的根结点,及其由维信息部分事务构造的基本树体.每个维事务映射为树体中的一条路径,所有事务路径的组合构成树体.事务内维的排列次序按每维属性取值由少到多排列.

(2) Header Table 由两个域组成:属性域 attr-val 和链域* link.

(3) 节点结构:包含谓词标识符、计数 count、指向父亲结点的指针* parent、指向孩子结点的指针* child 和指向同名谓词的链接* link.

(4) 每个叶子节点连接一个信息节点,存放 MDP-tree 树约束下事务标号集合和与 FP-tree 树连接的链,以及维谓词集约束下的频繁项集.

例 1 设某超市销售记录表(表 1),其维信息部分的 3-D 数据立方体如表 2 所示.

表 1 多维事务表

TID	年龄	收入	职业	商品
001	A1	S2	C1	p,l,o,n,k
002	A1	S1	C2	p,n,k
003	A2	S2	C2	p,o
004	A2	S3	C3	p,l
005	A1	S2	C3	l,o,k
...	...	...	...	...
100	A2	S3	C4	p,l,n

表 2 3-D 数据立方体

年龄	收入	职业	Count
A1	S2	C1	30
A1	S1	C2	15
A2	S2	C2	20
A2	S3	C3	10
A1	S2	C3	5
...	...	...	...
A2	S3	C4	5

MDP-tree 的构造过程如下:

首先得到数据立方体的维数为 3,所以对应的 MDP-tree 应为三层,比较各个维的不同属性值的个数(Sum 值), $Sum(年龄) = 2, Sum(收入) = 3, sum(职业) = 4$, MDP-tree 的第一层应为年龄维结点,第二层为收入维结点,第三层为职业维结点.然后得到数据立方体的第一条记录 $\{A1, S2, C1, 30\}$ 对

应的树中路径为{A1:30,S2:30,C1:30},取数据立方体的第二条记录{A1,S1,C2,15},遍历树中根节点的孩子节点,发现 A1 结点已经存在,则将树中 A1 结点的计数值变为 45;遍历 A1 的孩子节点,没有 S1 结点,则新建 S1 结点,计数值为 15,新建 S1 结点的孩子节点 C2,计数值为 15. 依此下去,直至取完数据立方体中记录,最终得到的 MDP-tree 如图 1 所示.

性质 1 MDP-tree 的完备性:给定一个数据立方体 D,它对应的 MDP-tree 包含该数据立方体的完备信息.

性质 2 MDP-tree 中叶子结点等高,并且 MDP-tree 的深度由数据立方体的维数决定.

性质 3 叶子结点决定整条路径的频繁性:如果 MDP-tree 的叶子结点是频繁谓词,则该叶子结点到根结点之间路径对应的谓词集是频繁的.

3.2 计算频繁谓词集

根据 MDP-tree 的性质,自底向上遍历 Header Table(即先考察谓词名为 C4 的节点),通过 * link 指针考察 MDP-tree 中每一个同名谓词节点的支持数,判断是否满足最小支持数,若不满足则进行剪枝处理.直至到达 Header Table 的顶部,算法才结束,为了便于描述频繁谓词集的产生过程,以图 1 所示 MDP-tree 为例,设最小支持数为 10.

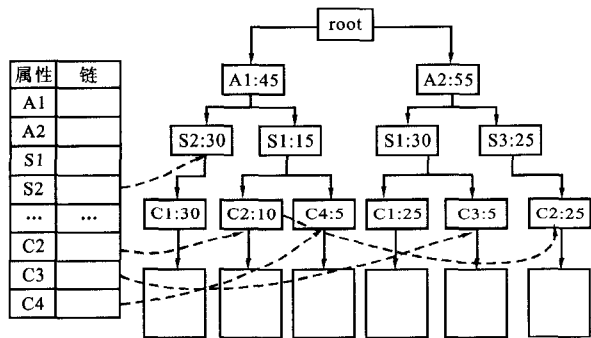


图 1 MDP-tree 的构建图

通过该方法将非频繁谓词集从 MDP-tree 中剪枝掉,其剪枝后的 MDP-tree 如图 2 所示.从图中可得知,从根节点到信息节点之间的每一条路径,对应一个频繁谓词集,其支持度为该路径上谓词的最小支持度.如{A1,S2,C1,30}、{A1,S1,C2,10}和{A2,S1,C1,25}等.

3.3 构建 MDPI-tree

(1) MDPI-tree 结构

MDPI-tree 由维信息部分的 MDP-tree 和项信息部分事务构造的 FP-tree 及其连接构造而成.

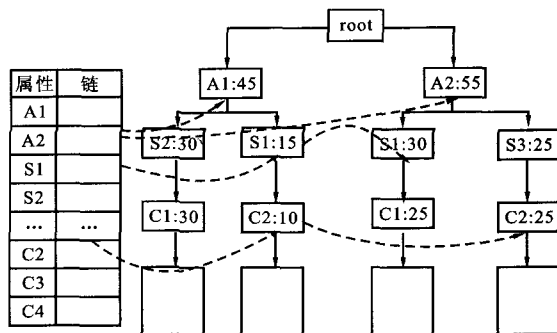


图 2 频繁谓词集

由于 MDP-tree 树中包含了事务数据库频繁维谓词集,而项信息部分通过维信息的约束,映射到数据结构 FP-tree 的一条路径和 MDP-tree 树建立对应关系.注:构建项信息的 FP-tree 结构在原先的基础上为每个叶子节点增加了一个信息节点,主要用来存储频繁项集在事务数据库中所对应的事务编号.

(2) MDPI-tree 的构建过程

① 第一遍扫描事务数据库,对维信息部分按照本节所介绍的方法构建 MDP-tree,并过滤掉非频繁谓词集;对项信息部分,按出现频度由高到低的次序构造 FP 树结构头表 Lh(包含三个域:属性域 att-val、计数域 count 和链域 side link),并过滤掉非频繁项.

② 第二遍扫描事务数据库,每一个事务将项信息部分映射到 FP-tree 的一条路径上,并将相应事务信息(tid)存入信息节点中.

③ 检查 MDP-tree 与 FP-tree 信息节点中的事务信息,如果属于同一事务,则将 MDP-tree 树中叶子的信息节点与 FP-tree 中叶子的信息节点建立关联.否则如果 FP-tree 中的某个项集不满足维约束,则剪枝.

以表 1 所示的多维事务数据库为例,图 3 中给出 MDPI-tree 的构造.

3.4 频繁多维项集挖掘

在多维事务数据库中,每个事务的项集被事务中的维值约束.这样频繁 K 项集也要受到维值约束,此外,只有那些被频繁维谓词集约束的频繁 K 项集才能与频繁谓词集一起形成多维频繁项集,因此只需要考虑频繁谓词集代表的维值约束,由于 MDP-Tree 中的信息节点与频繁谓词集是 1-1 对应的,可以用信息节点存放对应的频繁维谓词集下的频繁项集,设 MDP-tree 有 n 个信息节点 $b_1, b_2, \dots, b_n$,其多维频繁项集的挖掘过程 procedure MDPI-growth()如下:

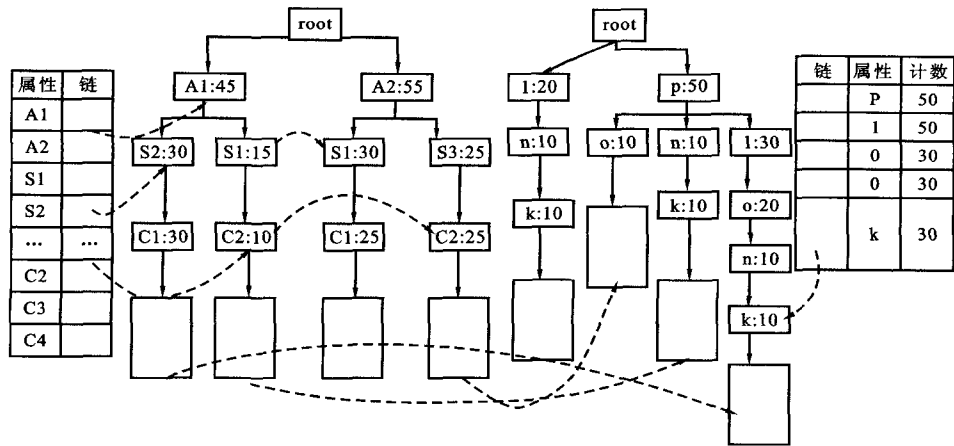


图 3 多维谓词索引树

Step1:取出第 i 个信息节点 b_i ,其对应的频繁维谓词集为 a ,根据信息节点与 FP 树的链接域,得到相应的 FP 子树($tree_a$);

Step2:采用 $FP\text{-}Growth(tree_a, a)$ 算法挖掘出与频繁维谓词集 a 相对应的频繁项集,并将频繁项集存入 b_i 中.

Step3: $i++$;如果 $i < n$,转到 step 1,否则转到 step4;

Step4:最后将信息节点中的频繁项集与该信息节点对应的频繁谓词集合并,得到多维频繁项集,算法结束;

其中 $FP\text{-}Growth$ 算法比较经典.

4 算法应用与性能分析

4.1 多维关联规则在交叉销售中的应用

本实验数据来源于某移动通信公司 CRM 数据库,数据取自 2008 年 6 月份的用户业务订购记录,维度包括[品牌],[ARPU 值],[地区],其项集为用户所订购的业务,依据该数据采用 Microsoft 的 Analysis Service 构建数据立方体.其前台开发平台采用 Visual C++ 6.0,所选取的最小支持度为 0.15,最小置信度为 0.4.部分挖掘结果表 3 所示.

表 3 多维关联规则结果表

规则号	规则内容	支持度	置信度
1	[全球通]+[ARPU 值高]→[手机视频]	0.2	0.46
2	[青年]+[动感地带]+[ARPU 值低]→[飞信]	0.55	0.7
3	[全球通]+[ARPU 值高]→[手机报]	0.33	0.52
4	[动感地带]+[彩铃]→[全曲下载]	0.25	0.6
5	[ARPU 值高]+[手机报]→[手机游戏]	0.16	0.44

规则说明:

(1) 从规则 2 可以得出,动感地带学生用户且 ARPU 值比较低的情况下,使用飞信业务的可能性很大(置信度达到 70%).

(2) 从规则 3 可以得出,全球通用用户在 ARPU 值非常高的情况下,对手机报业务的需求较大(其置信度达到 52%).

(3) 从规则 4 可以得出,动感地带客户在使用彩铃的情况下,订购全曲下载业务的可能性较大,由此说明彩铃与全曲下载业务具有一定相关性.

实验结果表明,采用基于 MDPI 的多维关联规则算法所挖掘出来的规则不但能反映业务与业务之间的关联,也能体现业务与客户之间的关联,具有一定实用性.

4.2 算法性能分析

对于大型事务数据库的多维关联规则挖掘来说,其性能主要由 I/O 的效率决定.

算法的第 1,2 步是构建数据立方体并计算频繁维谓词集:完成本步骤只需要扫描事务数据库一次就可以构建数据立方体,在计算频繁维谓词集的时候,只需对 MDP-tree 进行操作,无需频繁地扫描数据立方体,因此不需要附加任何 I/O 开销.第 3,4 步是通过 MDPI 树将频繁维谓词集与频繁项集建立关联,并通过经典的 FP 算法挖掘出维约束下的频繁项集:只需要扫描事务数据库一次,将项信息映射到 FP 树中,并且不需要产生候选项集.

相比之下,使用类 Apriori 算法求频繁维谓词集需要频繁的扫描数据库,I/O 开销较大.而用数据方进行多维关联规则挖掘时,只需要扫描一次数据库,以构建数据立方体.但当项信息较大时,它不能用内存数组实现,这样构造和使用数据方的 I/O 开销仍然很大.并且由于数据方的稀疏性,其存储开销

也是非常大的。

5 结束语

本文算法将数据立方技术和 FP 算法有机地结合在一起,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 FP 算法的可扩展性.使用该算法能够挖掘维间关联规则,而且能够挖掘混合维关联规则.下一步的工作是对本文的算法进行改进,使其用于多个抽象层的多维关联规则挖掘,还可以将数据立方技术与其他关联规则挖掘结合。

参考文献:

- [1] Han Jiawei, Kamber M. Data mining concepts and techniques[M]. 北京:机械工业出版社,2001.
- [2] 芦洁,刘志镜.挖掘关联规则中对 Apriori 算法的一个改进[J].微电子学与计算机,2006,23(2):10-12.
- [3] 黄勇.关系数据库中多维关联规则挖掘的新算法[J].计算机应用与软件,2007,24(10):60-61.
- [4] 李立羽,施鹏飞.OLAP 关联规则挖掘[J].计算机工程与应用,2002,38(3):128-130.
- [5] 颜石专,李战怀.基于数据仓库和 OLAP 的商务决策系统[J].微电子学与计算机,2006,23(2):64-67.
- [6] 党运峰.多维关联规则的研究与应用[D].广州:中山大学,2008.
- [7] 赵孝敏,何松华,李贤鹏.一种改进的 FP-Growth 算法及其在业务关联中的应用[J].计算机工程,2008(9):28-30.
- [8] 王青,王融清,鲍虎军,等.散乱数据点的增量快速曲面重建算法[J].软件学报,2000,11(9):1221-1227.
- [9] 林志浩,姜晓峰,王鹏远.一种改进的区域扩张三角网格面逆向重构算法[J].小型微型计算机系统,2006,27(8):1570-1573.
- [10] 田庆国,葛宝臻,郁道银,等.一种基于投影的散乱数据表面增量重建算法[J].中国图象图形学报,2006,11(10):1409-1413.
- [11] 熊邦书,何明一,俞华.三维散乱数据的 k 个最近邻域快速搜索算法[J].计算机辅助设计与图形学学报,2004,16(7):909-917.

作者简介:

彭 硕 女,(1985-),硕士.研究方向为数据仓库与数据挖掘.

吴 昊 男,(1970-),博士后,副教授.研究方向为人工智能与数据挖掘.

作者简介:

倪小军 男,硕士研究生.研究方向为计算机图形学、数字几何处理.

姜晓峰 男,副教授,硕士生导师.研究方向为计算机图形学、虚拟现实、数字几何处理.

葛 亮 男,硕士研究生.研究方向为计算机图形学、虚拟现实.