

# 基于 SQL 的多值多层关联规则挖掘

黄勇<sup>1</sup>, 刘锋<sup>2</sup>

(1. 安徽科技学院 计算机系, 安徽 凤阳 233100;

2. 安徽大学 计算机学院, 安徽 合肥 230039)

**摘要:**在分析研究关系数据库上关联规则挖掘现有方法的基础上,提出了一种基于结构化查询语言 SQL 的多值多层关联规则挖掘新方法。采用了一种新的根据概念分层的编码方法对多值属性进行离散化,然后利用 SQL 的查询语句,结合多值属性的编码,实现了关系数据库上的多层关联规则挖掘。实验表明,该算法具有快速、有效、易开发等优点。

**关键词:**关系数据库;多层关联规则;SQL 语言

**中图分类号:**TP311

**文献标识码:**A

**文章编号:**1673-629X(2008)06-0101-03

## SQL - Based Multi - Value Multilayer Mining Association Rules Excavation

HUANG Yong<sup>1</sup>, LIU Feng<sup>2</sup>

(1. Department of Computer Science, Anhui Science and Technology University, Fengyang 233100, China;

2. Department of Computer Science, Anhui University, Hefei 230039, China)

**Abstract:** On the basis of the analysis of the present algorithms about data mining of association rules in relational database, proposes one new method of multilayered association rules excavation based on the structured inquiry language SQL. First introduce a new coding method of concept lamination for multi-valued attributes discrete, then use SQL the inquiry sentence, combining multi-valued attribute coding, to achieve multilayer mining association rules in relational database. Experiments show that this algorithm has many advantages, such as quickness, effectiveness, ease to develop, etc.

**Key words:** relational database; multi-level association rule; SQL language

### 0 引言

关联规则是数据挖掘中一种重要的模式,它首先由 R. Agrawal<sup>[1]</sup>等人于 1993 年提出,现已成为数据挖掘领域中一个非常重要的研究课题,其在货篮计划、商品广告邮寄分析、仓储规划、网络故障分析中得到了较为广泛的应用。起初关联规则挖掘仅限于事务数据库,近年来,由于关系数据库技术的迅速发展和广泛应用,大量的生产、管理和科研信息都存储在关系数据库中,积极研究关系数据库中关联规则挖掘的有效技术具有极为广阔的发展前景<sup>[2]</sup>。

目前在关系数据库中挖掘关联规则的方法主要有两大类:一类是先将关系数据库转换为事务数据库,再

利用事务数据库中布尔型关联挖掘算法进行挖掘<sup>[3,4]</sup>;二是利用结构化查询语言 SQL 与关联规则挖掘算法结合起来,利用 SQL 语言进行关联规则的挖掘<sup>[5~7]</sup>。这两种方法中,第一种方法以较为成熟的布尔型关联挖掘算法为基础,有较好的理论基础,但数据在转换过程中会造成存储空间和挖掘时间的消耗,对于关系数据库中的多值型数据,转换成布尔型数据时,将占用更多的存储空间,因为一个多值属性转换成布尔型数据后,就相当于几个属性,数据库会变得更大。对于第二种方法,注重了与 SQL 的结合,是一种较好的解决方案,但目前与 SQL 语言结合的算法研究主要集中在对 Apriori 算法的关键步骤运用 SQL 语言进行实现和有限扩展<sup>[5,6]</sup>,从而影响了算法的效率;有些算法(见文献[7]),利用 SQL 语言直接进行频繁项集的挖掘,提高了挖掘效率,但没有考虑到关系数据库中数据的多值型特点及多层关联规则的挖掘,以致影响其实用性。文中在继承 Apriori 算法思想的基础上,根据一种概念分层的方法对关系数据库中多值型数据进行

收稿日期:2007-09-04

基金项目:安徽省重点自然科学研究资助项目(KJ2007A043);安徽科技学院稳定人才资助项目(ZRC2007138)

作者简介:黄勇(1974-),男,安徽肥东人,硕士,讲师,研究方向为数据库与数据挖掘;刘锋,博士,教授,研究方向为遗传算法、数据挖掘。

离散化处理,并利用 SQL 的聚集函数及分组语句在不产生候选项集的情况下直接实现了频繁项集的挖掘,并支持多层关联规则的挖掘。

## 1 Apriori 算法

Apriori 算法<sup>[8]</sup>是一种最有影响的挖掘布尔关联规则频繁项集的算法。该算法使用一种称作逐层搜索的迭代方法, $k$ -项集用于探索 $(k+1)$ -项集。先找出频繁 1-项集记  $L_1$ ,  $L_1$  用于找  $L_2$ ,  $L_2$  用于找  $L_3$ , 依此,直到不能找到  $L_k$ 。找每个  $L_k$  需要一次数据库扫描。

算法主要有三个主要步骤:连接、剪枝、扫描。在连接步中,为找  $L_k$ ,通过  $L_{k-1}$  与自己连接产生候选  $k$ -项集记  $C_k$ 。两个  $L_{k-1}$  是可连接的,如果它们的前 $(k-2)$ 项相同。 $C_k$  是  $L_k$  的超集,它的成员可以是也可以不是频繁的,因此在剪枝步中要根据 Apriori 性质,即,频繁项集的所有非空子集都必须也是频繁的,剪去候选集中不可能是频繁项集的选项。在扫描步中,扫描数据库,发现候选集  $C_k$  中的  $L_k$ 。算法中有两点影响其性能:一是可能需要产生大量候选集,二是它需要多次扫描数据,通过模式匹配检查一个很大的候选集合。

## 2 基于 SQL 的多值多层关联规则挖掘算法

### 2.1 SQL 语言实现的优势

SQL (Structured Query Language), 即结构化查询语言,是一个通用的、功能极强的关系数据库的标准语言。基于 SQL 的挖掘算法的优越性体现在以下几点<sup>[7,9]</sup>:

首先,被挖掘的数据源为关系型数据库,若利用非 SQL 算法去实现,其数据的访问必须以间接的方式进行,那么将被挖掘的数据转换成算法所需要的格式必然消耗大量的时间和巨大的存储空间,一旦数据源库的数据发生变化,那么这些转换后的数据必须重新准备。

其次,使用基于 SQL 算法可充分利用数据库本身所提供的查询和执行引擎,也不必担心随着数据增加而出现的问题,仅仅需要关心算法本身。

更为重要的是,利用 SQL 的聚集函数  $\text{count}()$  及分组语句  $\text{group by}$  在不产生候选项集的情况下可直接实现频繁项集的挖掘。例:若要统计人事数据库 (RSH.DBF) 中学历字段 (XL) 中是否有频繁 1-项集,则可以用“ $\text{select xl, count(*) from rsh group by xl having count(*)} \geq \text{min\_sup into table p1}$ ”语句完成。若学历字段 (XL) 中有频繁 1-项集,则相应项存于表 P1 中。设人事数据库中学历字段及职称 (ZHC) 字段中皆

有频繁 1-项集,则频繁 2-项集的挖掘则可直接在字段级别上实现,其语句是“ $\text{select xl, zhc, count(*) from rsh group by xl, zhc having count(*)} \geq \text{min\_sup into table p2}$ ”,频繁 2-项集的挖掘不再是首先生成候选 2-项集,再逐一判断候选 2-项集是否满足最小支持度,而是在更高的级别上(字段上)进行,扫描数据库的过程被隐含在 SELECT 查询操作中,SELECT 语句在进行频繁项集判断时采用了优化的扫描查询策略,这种策略往往比用户固定的扫描方法具有更高的效率。

### 2.2 基于 SQL 的多值多层关联规则挖掘算法

#### 2.2.1 多值属性的离散化

在关系数据库中存在大量的多值型字段,若直接使用 SELECT 语句,在一定的最小支持度约束下,挖掘不出有实际意义的频繁项集,在挖掘前必须对数据库中的多值型字段作离散化处理。

文献<sup>[5]</sup>中利用语言场理论给出了一种多值属性的划分方法,该方法采用了一种类似二进制的编码方法,通过数据字典可以建立二进制编码和其语义的对应关系。由于采用二进制,降低了存储空间,但该二进制的编码方法过于简单,不能适应多层关联规则的挖掘。然而,对于关系数据库来说,一些项或属性所隐含的概念是有层次的。例如,对于销售事务库中的商品“羽绒服”,而对于一个决策者来说,可能关心它的更高层次概念,如“冬季服装”。在实际应用中发现,往往强关联规则不出现在概念的低层,在较高的层次才有可能有出现。基于以上分析,采用了一种新的多值属性的划分方法,该方法中一方面对多值属性采用二进制编码,以降低存储空间,另一方面其编码方法能够支持多层关联规则的挖掘。编码方法如下:首先对多值属性进行概念分层,得出概念层次图。职称的概念分层图如图 1 所示。

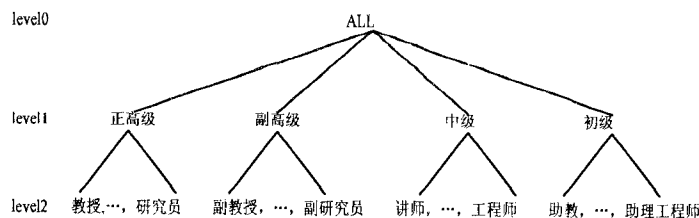


图 1 职称的概念分层

首先根据 level1 进行编码,因 level1 为四类,故编码为二位二进制,即 00 表示正高级,01 表示副高级,以此类推,中级为 10,初级为 11,然后再根据 level2 进行编码,在 level2 层次上找出分类最多的项,然后进行编码,其它 level2 层次上的分类项按最多项的长度进行编码,以便于后续挖掘算法的实现。设正高级在 level2 水平上有可分成四类:教授、……、研究员,则教

授职称在 level2 层次上的编码为 00,二者合起来,则教授职称的编码为 0000,研究员的编码为 0011。助教的编码为 1100,其中编码中的 11 说明在 level1 概念层上,该字段值为初级。这种编码方法为多层关联规则挖掘提供了支持,在实际挖掘中,若在 level2 层次上没有强关联规则,可以缩短该挖掘字段的长度,即舍去 level2 层次上的编码,比如教授和研究员原来的编码为 0000、0011,舍去 level2 层次上的编码 00、11 后,教授和研究员的编码值皆为 00,即在 level1 层次上两者属于一类,即正高级。

对于量化属性,首先在一个小的区间进行离散化,离散化后的区间值就相当于一个分类值,然后再采用以上方法进行编码。如数据库中年龄字段,首先可以在某一概念层 level $i$  上细分成  $[20 \sim 25]$ 、 $[26 \sim 30]$ 、 $[31 \sim 35]$  三个区间,然后,在概念层 level $i - 1$  上把三个区间归为一类,即青年。

### 2.2.2 算法描述

根据上面多值属性的划分方法对数据库中的多值型字段进行数据转换,以形成挖掘库,在实际中,可以在源库中增加相应的挖掘字段而不必生成新的数据库,这一方面可以节约存储空间,另一方面也没有破坏源库中的数据。

算法描述<sup>[7,8]</sup>:

输入:挖掘库(RDB)、最小支持度阈值(min-sup)、最小可信度阈值(min-conf)  $k = 1$

$L_k = \text{generate-frequent-attribute}(C_1)$  //产生 RDB 中频繁 1 字段集,  $C_1$  为库中 1 组合字段集

Do while  $L_k \neq \emptyset$

{  $C_{k+1} = \text{generate-attribute-combinations}(L_k)$  //根据频繁字段集  $L_k$  产生候选字段集

$L_{k+1} = \text{generate-frequent-attribute}(C_{k+1})$  //产生 RDB 中频繁  $k + 1$  字段集

$k = k + 1$  }

AR = generate-association-rules( $L_k$ .dbf) //产生 RDB 中所有强关联规则,其中  $L_k$ .dbf 为频繁字段集所对应的频繁项集

在以上算法中,generate-frequent-attribute( $C_{k+1}$ )操作产生两个结果:一是频繁字段集,存于数组中;二是频繁字段集对应的频繁项集,存于  $L_k$ .dbf 中,  $L_k$ .dbf 库生成后仅在产生强关联规则时使用,不参与后续频繁项集的挖掘,如上所述,基于 SQL 的频繁项集的挖掘是基于字段级进行的,不需要根据频繁  $k$ -项集产生候选频繁  $k + 1$ -项集,再扫描数据库判断每一候选集是否为频繁项集,扫描数据库的过程被隐含在 SELECT 查询操作中。产生  $C_{k+1}$  操作是根据  $L_k$  的频繁字段集,利用 Apriori 算法“连接”与“剪枝”操作生成字段级上的候选项集。由于字段级上的组合数少,可将

候选字段级存于数组中。generate-frequent-attribute( $C_{k+1}$ )与 generate-attribute-combinations( $L_k$ )的算法如下:

generate-frequent-attribute( $C_{k+1}$ )

$i = 1$

do while  $i \leq n$  //设  $C_{k+1}$  存于 array( $n, k + 1$ ) 二维数组中,  $n$  为候选频繁字段的个数

{select &array( $i, 1$ ), &array( $i, 2$ ), ..., &array( $i, k + 1$ ),

count(\*) group by &array( $i, 1$ ), &array( $i, 2$ ), ..., &array( $i, k + 1$ ) having count(\*)  $\geq$  min-sup into table  $L_k$

$i = i + 1$  }

generate-attribute-combinations( $L_k$ )

for each attribute  $l_1 \in L_k$

for each attribute  $l_2 \in L_k$

if ( $l_1[1] = l_2[1] \wedge l_1[2] = l_2[2] \wedge \dots \wedge l_1[k - 2] = l_2[k - 2] \wedge l_1[k - 1] < l_2[k - 2]$ ) then {

$c = l_1 \text{ } \bowtie \text{ } l_2$ ; //join step

if has-infrequent-subset( $c, L_k$ ) then

delete  $c$ ;

else add  $c$  to  $C_{k+1}$  }

return  $C_{k+1}$

Procedure has-infrequent-subset( $c, L_k$ ) //use priori knowledge

for each ( $k$ ) - subset  $s$  of  $c$

if  $s \notin L_k$  then

return TRUE

else return FALSE

## 3 实验结果

为了验证算法的有效性及性能,选用了两个库结构相同的实验数据库 experiment1.dbf 与 experiment2.dbf,其中 experiment1.dbf 有记录 5000 条,experiment2.dbf 有记录 15000 条,库中有多值属性 6 个,其中,分类属性 4 个,量化属性 2 个,利用概念分层的编码方法把实验数据库转换为挖掘数据库。算法所采用的编程语言为 Delphi 6.0,数据库系统是微软的 SQL-sever 2000, C/S 结构,服务器端的 CPU 为 P4 2.66G,内存 512M,客户端的 CPU 为 P4 1.5G,内存 128M。

实验结果表明,采用文中所述的编码方法及基于 SQL 的挖掘算法可以在不同概念层次上实现频繁项集的挖掘,从而生成相应的关联规则,从时间上看,当最小支持度为 0.2,对数据库 experiment1.dbf 中频繁项集的挖掘时间为 3.26s,对数据库 experiment2.dbf 中频繁项集的挖掘时间为 4.83s,这说明该算法对数据库中记录的增加不太敏感,此原因主要是算法的关键步骤采用了 SQL 语句。

(下转第 178 页)

### 3.4 数据库服务

该系统采用 Microsoft SQL Server 2000 对数据库进行设计<sup>[3]</sup>、维护,并使用 ADO 技术操作数据库<sup>[4]</sup>。SQL Server 是一个全面的、集成的、端到端的数据解决方案,具有使用方便、可伸缩性好、与相关软件集成程度高等优点,可跨越多平台使用。ADO 是 ActiveX 数据对象(ActiveX Data Object),是 Microsoft 开发数据库应用程序的面向对象的新接口。ADO 访问数据库是通过访问 OLE DB 数据提供程序来进行的,提供了一种对 OLE DB 数据提供程序的简单高层访问接口。

本系统中,采用了为实现可视化的数据源设置方法,实现数据访问的透明性,将通过数据连接文件(\*.UDL)来创建 ADO 连接。UDL 文件,其重要作用是创建和管理计算机和 OLE DB 数据存储之间的连接<sup>[5]</sup>。ADO 连接对象可以很方便地使用 UDL 文件来动态连接数据源,这样一来无论数据源如何变化,在软件中都可以用统一的方法编程。当数据源改变时,只要打开相应的 UDL 文件即可可视化地设置数据源,无需更改软件。这样一来,又降低了因采用 C/S 架构而带来的系统维护、升级成本。

## 4 结束语

如何更好地将信息化技术应用到制造业中,已成为制造业发展进程上的一大课题。从如何实现订单化

生产管理角度出发,提出了一个面向具体产品的系统解决方案,详细介绍了系统的作用和工作模式、系统的工作原理、系统架构,并对系统的各个组成部分功能模块设置、主要设计要素进行了着重介绍,最终完成了系统的设计和开发。

简而言之,系统是从如何保证生产过程中产品信息流和物流正确、畅通的流转而进行设计和开发的。信息流和物流是制造业企业中最最重要的两个管理内容,企业管理水平的高低最主要还是取决于对这两个方面的控制和管理。从这个意义上来说,还是任重而道远的。

### 参考文献:

- [1] 余伟萍. 计算机管理信息系统开发与应用[M]. 成都:电子科技大学出版社, 1998.
- [2] 刘晓燕, 张云生, Schwarz J J, 等. 基于 C/S 关系的实时系统构件交互规约[J]. 计算机工程与应用, 2007, 43(17): 104-107.
- [3] 杨正洪, 郑齐健. SQL Server7 关系数据库系统管理与开发指南[M]. 北京:机械出版社, 2003.
- [4] 吴 涵. 基于 VC++ 的研究生信息管理系统的设计与实现[J]. 计算机技术与发展, 2006, 16(12):184-186.
- [5] 马宏琳, 阎 磊. 基于 UDL 文件动态建立 ADO 连接方法研究[J]. 电脑知识与技术, 2006, 20(7):25-29.

(上接第 103 页)

## 4 结束语

通过分析和实验证实:利用概念分层的编码方法及 SQL 语言实现多值多层关联规则的挖掘算法是快速有效的,并且算法实现简单,开发容易,适用范围广。

### 参考文献:

- [1] Agrawal R. Mining association rules between sets of items in larger databases[C]//In: Proc of ACM SIGMOD Int'l Conf Management of Data. Washington, DC: [s. n.], 1993: 207-216.
- [2] Srikant R, Agrawal R. Mining Quantitative Association Rules in Large Relational Tables[C]//Proc. 1996 ACM SIGMOD Int'l Conf. Very Large DataBase. Montreal, Canada: [s. n.], 1996: 1-12.

(上接第 173 页)

- [1] 感改进[J]. 系统仿真学报, 2001, 13(9):220-223.
- [2] MultiGen Paradigm Inc. Vega 3.7 Option Guide[M]. Dallas: MultiGen Paradigm Inc., 2001.
- [3] MultiGen Paradigm Inc. Vega 3.7 Programmer's Guide[M].

- [3] 李 虹, 蔡之华. 关联规则在医疗数据分析中的应用[J]. 微机发展, 2003, 13(6):94-97.
- [4] 王选文, 丁 夷, 范九伦. 关联规则挖掘在人事系统中的应用[J]. 西安邮电学院学报, 2001, 6(1):21-23.
- [5] 杨炳儒, 孙海洪, 熊范纶. 利用标准 SQL 查询挖掘多值型关联规则及其评价[J]. 计算机研究与发展, 2002, 39(3): 307-312.
- [6] 周剑雄, 王明哲. 基于关联规则的数据挖掘技术的快速算法[J]. 计算机工程, 2003, 29(12):48-49.
- [7] 王 芳, 王万森. 关系数据库中关联规则挖掘的一种高效算法[J]. 微机发展, 2004, 14(9):20-22.
- [8] Han J, Kamber M. Data Mining: Concepts and Techniques [M]. Beijing: Higher Education Press, 2001.
- [9] 夏克强, 李石君. 基于 SQL-92 进行频繁集发现[J]. 计算机工程与应用, 2004(9):177-179.

Dallas: MultiGen Paradigm Inc. 2001.

- [4] 唐 凯, 康凤举, 褚彦军. Vega 中云的仿真方法[J]. 系统仿真学报, 2005, 17(9):2051-2053.
- [5] Gardner G Y. Visual Simulation of Clouds[J]. Computer Graphics, 1985, 19(3):279-303.