ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.11, November 2007, pp.2752−2765
DOI: 10.1360/jos182752

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

挖掘多关系关联规则*

何 军[1+], 刘红岩[2], 杜小勇[1,3]

[1](中国人民大学 计算机科学与技术系,北京 100872)

[2](清华大学 管理科学与工程系,北京 100084)

[3](教育部数据工程与知识工程重点实验室,北京 100872)

Mining of Multi-Relational Association Rules

HE Jun[1+], LIU Hong-Yan[2], DU Xiao-Yong[1,3]

[1](Department of Computer Science and Technology, Renmin University of China, Beijing 100872, China)

[2](Department of Management Science and Engineering, Tsinghua University, Beijing 100084, China)

[3](Key Laboratory of Data Engineering and Knowledge Engineering, Ministry of Education, Beijing 100872, China)

+ Corresponding author: Phn: +86-10-62514014, E-mail: hejun@ruc.edu.cn

He J, Liu HY, Du XY. Mining of multi-relational association rules. *Journal of Software*, 2007,18(11): 2752−2765. http://www.jos.org.cn/1000-9825/18/2752.htm

Abstract: Association rule mining is one of the most important and basic technique in data mining, which has been studied extensively and has a wide range of applications. However, as traditional data mining algorithms usually only focus on analyzing data organized in single table, applying these algorithms in multi-relational data environment will result in many problems. This paper summarizes these problems, proposes a framework for the mining of multi-relational association rule, and gives a definition of the mining task. After classifying the existing work into two categories, it describes the main techniques used in several typical algorithms, and it also makes comparison and analysis among them. Finally, it points out some issues unsolved and some future further research work in this area.

Key words: data mining; association rule; relational database; star schema

摘 要: 关联规则的挖掘是数据挖掘中的一项重要和基础的技术,已进行了多方面的深入研究,有着广泛的应用.传统数据挖掘算法是针对单表数据进行处理的,在应用于多关系数据挖掘时存在诸多问题.对多关系关联规则的挖掘问题进行了重新定义和总结.提出了多关系关联规则挖掘的一个框架,并对已有算法进行了分类.然后对各类代表性算法进行了描述、分析和对比,对尚存在的问题进行了分析和总结.最后,对该领域未来的研究工作提出了建议.

关键词: 数据挖掘;关联规则;关系数据库;星型模式

中图法分类号: TP311 **文献标识码**: A

计算机在信息社会的广泛应用,使得越来越多的数据积累在信息系统中.在这些数据中隐藏着大量的不为

* Supported by the National Natural Science Foundation of China under Grant Nos.70471006, 70621061, 60496325, 60573092 (国家自然科学基金)

Received 2006-11-11; Accepted 2007-03-19

人们所知的模式或知识,寻找有价值的数据模式或知识是数据挖掘研究的主要内容.数据挖掘技术经历了近一二十年的发展,取得了较大进展,在融合统计学、数据库、计算机图形学、人工智能和机器学习等学科内容的基础上,形成了一套自己的理论、技术和方法.已经提出的数据挖掘技术包括关联规则、分类、聚类、序列模式、预测、趋势分析、时间序列中的相似性搜索,等等.

然而,在现有的数据挖掘技术中所提出的数据挖掘算法基本上都是基于关系数据库中的单个表上的数据实现的,而实际应用中的数据大多数分散存储在关系数据库的多个表中,因而多关系数据挖掘引起了越来越多的研究人员的关注,并成功地应用于多种领域[1].关联规则是数据挖掘中的重要技术之一,有着广泛的应用.对于单表数据集,该方面的研究已经做了很多[2−13].从发现模式的角度,可将其分为两大类:一类是发现频繁模式(frequent pattern)[2−8];另一类是发现频繁闭合模式(frequent closed pattern)[9−13].在这些算法中,有代表性的算法涉及不同的挖掘方法:Apriori[3]是按层次不断生成候选频繁集从而发现频繁模式的典型算法;FP-growth[6]是采用树形结构不必生成候选频繁集而发现频繁项集的典型算法;CHARM(closed association rule mining)[10]是采用垂直结构而不是通常的水平结构来表示数据、发现频繁闭合模式的典型算法;CARPENTER(closed pattern discovery by transposing tables that are extremely long)[12]则是采用转置(transposition)方法发现包含大量属性的小数据集中的频繁闭合模式的典型算法.

如果将传统的关联规则挖掘算法直接应用到多关系数据挖掘中则存在诸多的问题[14,15],多关系数据挖掘正是为了解决这些问题而诞生的.目前,这一概念还没有统一的严格定义.有的学者认为,直接在一个关系数据库的多个表中寻找有意义的模式,不需要将多个表转换为单个表的技术称为多关系数据挖掘[15].持这种观点的学者强调多关系数据挖掘不应该是通过将多个表转换为单个表后对数据进行分析;也有的学者则致力于将多个关系的数据进行一定的归纳、汇总,通过创建新的属性等方法将多个关系的信息集中到一个表,然后处理.根据我们目前了解的有关多关系关联规则挖掘研究的现状,给出多关系关联规则挖掘的定义如下:

定义 1. 通过分析一个关系数据库的多个表中的数据发现存在于单个表以及多个表的属性值之间的关联规则的过程称为多关系关联规则挖掘(multi-relational association rule mining,简称 MRARM).

本文第 1 节给出相关的基本概念的定义,提出多关系关联规则挖掘研究的基本框架.第 2 节总结多关系关联规则挖掘需要解决的问题.第 3 节讨论基于 ILP(inductive logic programming)技术的多关系关联规则挖掘方法及其代表性算法.第 4 节描述和比较各种解决星型模式下多关系频繁项集以及频繁闭合项集的挖掘算法.第 5 节对已有工作进行总结,提出尚存在的问题、挑战以及未来的研究工作.

1 定义与问题描述

为了便于理解和描述已有工作和算法,我们首先给出基本概念和术语的定义.

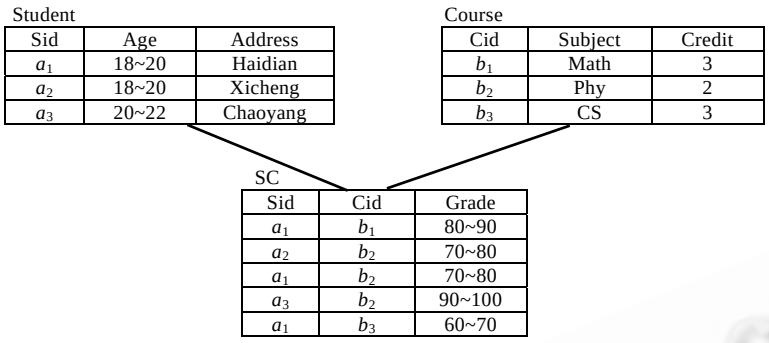
很多已有多关系关联规则挖掘的研究是基于以星型模式组织的数据库而展开的.如同多维数据模型(multi-dimensional data model)中的星型模式(star schema),假设数据挖掘的数据是以星型模式组织的,即由一个事实表(fact table,简称 FT)和多个维表(dimensional table)构成,每个维表的主键(primary key)在事实表中以外键(foreign key)的形式出现,以表达维表和事实表之间的 1 对多联系.如图 1 中所示的数据库由 3 个表构成.Student 表记录了所有学生的信息,Course 表记录了所有的课程信息,SC 表中记录了有关学生选课的信息,这个数据库可以看作是一个星型模式.其中,表 SC 是事实表,表 Student 和表 Course 分别是维表.SC 表中的外键属性 Sid 参照的是表 Student 的主键 Sid,外键属性 Cid 参照的是表 Course 的主键 Cid.

为了便于挖掘任务的实现,假设事实表中仅含有外键属性,如还有其他属性,则可以将它们分离出来构成另一个维度表.按照此方法,图 1 中的 3 个表可以转化为如图 2 所示的星型模式,其中,原属于事实表的属性 Grade 被分离出来单独组成另一个维表 Score.

更一般地,一个数据库可以看作是星型数据模式的一个扩展,即由多个星型模式组合而成,如图 3 所示.

定义 2(连接表). 将一个数据库中的所有表进行连接构成一个泛关系表,称为连接表(join table).

表 1(见第 2.1 节)所示的泛关系表就是图 1 中的 3 个表的连接表.



Student

Sid	Age	Address
a_1	18~20	Haidian
a_2	18~20	Xicheng
a_3	20~22	Chaoyang

Course

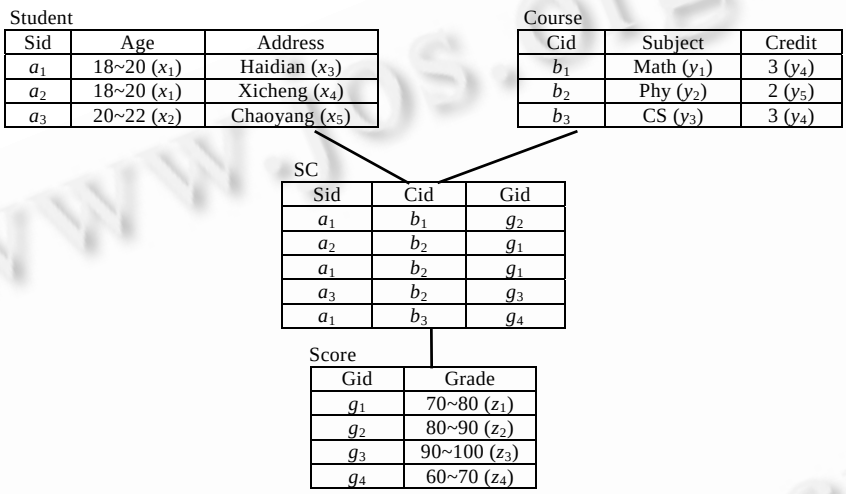
Cid	Subject	Credit
b_1	Math	3
b_2	Phy	2
b_3	CS	3

SC

Sid	Cid	Grade
a_1	b_1	80~90
a_2	b_2	70~80
a_1	b_2	70~80
a_3	b_2	90~100
a_1	b_3	60~70

Fig.1 Star schema database

图 1 星型模式数据库



Student

Sid	Age	Address
a_1	18~20 (x_1)	Haidian (x_3)
a_2	18~20 (x_1)	Xicheng (x_4)
a_3	20~22 (x_2)	Chaoyang (x_5)

Course

Cid	Subject	Credit
b_1	Math (y_1)	3 (y_4)
b_2	Phy (y_2)	2 (y_5)
b_3	CS (y_3)	3 (y_4)

SC

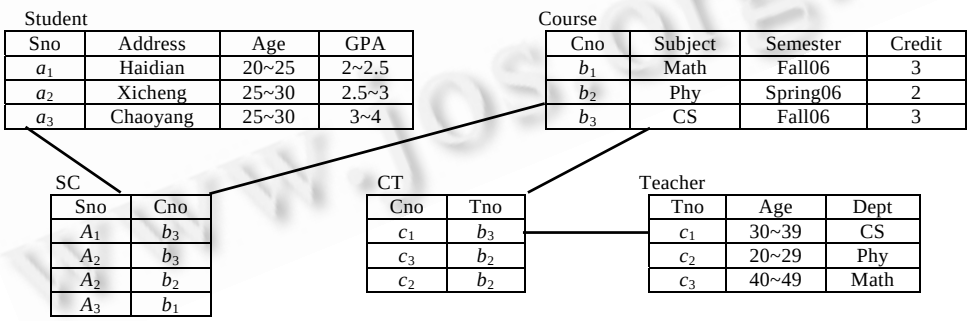
Sid	Cid	Gid
a_1	b_1	g_2
a_2	b_2	g_1
a_1	b_2	g_1
a_3	b_2	g_3
a_1	b_3	g_4

Score

Gid	Grade
g_1	70~80 (z_1)
g_2	80~90 (z_2)
g_3	90~100 (z_3)
g_4	60~70 (z_4)

Fig.2 Star schema

图 2 星型模式



Student

Sno	Address	Age	GPA
a_1	Haidian	20~25	2~2.5
a_2	Xicheng	25~30	2.5~3
a_3	Chaoyang	25~30	3~4

Course

Cno	Subject	Semester	Credit
b_1	Math	Fall06	3
b_2	Phy	Spring06	2
b_3	CS	Fall06	3

SC

Sno	Cno
A_1	b_3
A_2	b_3
A_2	b_2
A_3	b_1

CT

Cno	Tno
c_1	b_3
c_3	b_2
c_2	b_2

Teacher

Tno	Age	Dept
c_1	30~39	CS
c_2	20~29	Phy
c_3	40~49	Math

Fig.3 ER model based database

图 3 基于 ER 模型的数据库

定义 3(项、项集). 包含于任一个表的非主键和外键属性的每个不同取值称为一个项(item),记为一个属性和取值对:属性=取值,或用一个唯一的标识符表示.由多个项构成的集合称为项集(itemset).项集中的项若全部来自于一个表,则称项集为单表项集(single table itemset);若涉及的表个数大于 1,则称为跨表项集(cross table itemset).

在图 2 中,表Student中项的一个例子为Age=18~20,也可以将所有项用标识符表示,如x_1代表Age=18~20,x_2

代表Age=20~22,x_3代表Address=Haidian,x_4代表Address=Xicheng,x_5代表Address=Chaoyang.其他表可按同样方法处理,如图 2 所示,各个表中分量旁括号中的内容为其标识符.

定义 4(支持度、置信度). 一个项集 X 在一个表中出现的次数除以该表的总行数称为该项集相对于该表的支持度,记为 sup(X).相应地,一个规则 $X\rightarrow Y$(其中,X 和 Y 均为项集)的置信度定义为 XY(即 $X\cup Y$)的支持度除以 X 的支持度,记为 $conf(X\rightarrow Y)$.

注意,不同的算法对于支持度的定义不尽相同,有些算法将支持度定义为相对于连接表的支持度,有些则定义为相对于所在表的支持度.在下面介绍具体算法时还将进一步加以介绍.

定义 5(单表频繁项集、跨表频繁项集). 由用户给定一个支持度的最小阈值(记为 minsup),所有支持度不小于该阈值的项集称为频繁项集(frequent itemset).若项集为单表项集,则称为单表频繁项集;同理,若频繁项集为跨表项集,则称为跨表频繁项集.

定义 6(频繁闭合项集). 给定最小支持度 minsup,对于一个频繁的项集 X,如果不存在一个项集 Y,$Y\supset X$,且 Y 与 X 的支持度相同,则 X 称为频繁闭合项集(frequent closed itemset).

定义 7(单表关联规则、跨表关联规则). 由用户给定一个置信度的最小阈值(记为 min$conf$)和支持度的最小阈值 minsup,对于频繁项集 X 和 Y,若其对应规则 $X\rightarrow Y$ 的置信度不小于最小阈值 min$conf$,则称该规则为关联规则.若项集 XY 为单表项集,则称规则为单表关联规则;同理,若项集 XY 为跨表项集,则称规则为跨表关联规则.

多关系关联规则挖掘任务:给定支持度和置信度的最小阈值 minsup 和 min$conf$,多关系关联规则的挖掘就是要发现存在于一个数据库的多个表中的所有满足 minsup 和 min$conf$ 的单表和跨表关联规则.

关联规则的挖掘通常分为两个步骤:第 1 步发现所有的频繁项集,第 2 步发现这些频繁项集之间满足最小置信度的关联规则.由于发现频繁项集的时间复杂度远远大于第 2 步关联规则的发现,因而,通常我们只讨论第 1 步的挖掘方法.在多关系的环境下,寻找满足最小支持度阈值的单表频繁项集和跨表频繁项集的过程,称为多关系频繁项集挖掘.

2 多关系关联规则挖掘的主要问题

将传统的单表关联规则挖掘算法进行扩展,用于发现多表关联规则时会遇到若干问题,我们将其总结为 3 类:性能问题、统计偏斜问题和信息丢失问题.下面分别加以说明.

2.1 性能问题

将传统单表关联规则挖掘算法用于发现多表关联规则的最直接的方法是将一个数据库的所有表进行连接,构造一个连接表或泛关系表.以如图 1 所示的有关学生-课程的数据库为例,当将 Student,Course,SC 表通过连接操作生成一个泛关系表 S-C-SC(见表 1)后,可以在此表上运用传统的数据挖掘算法进行挖掘.

Table 1 Universal table S-C-SC

表 1 泛关系表 S-C-SC

Sid	Cid	Grade	Age	Address	Subject	Credit
a_1	b_1	80~90	18~20	Haidian	Math	3
a_2	b_2	70~80	18~20	Xicheng	Phy	2
a_1	b_2	70~80	18~20	Haidian	Phy	2
a_3	b_2	90~100	20~22	Chaoyang	Phy	2
a_1	b_3	70~80	18~20	Haidian	CS	3

然而,在数据库的各种数据操作中,连接是最费时的操作之一[16],因此,这种方法会引起如下问题:

- 将多个关系进行连接操作,当涉及的表很大时,连接操作可能根本无法进行,因为连接操作的执行代价昂贵,很费时;
- 多表连接的结果所对应的单一关系表,数据冗余现象非常严重,规模可能非常庞大,在这样的表上运行数据挖掘算法将非常费时.

大多数多关系关联规则算法并不采用这种简单的处理方法,但即使是这样,为了挖掘跨表关联规则,也需要

将多个表的属性进行关联,如何避免连接操作从而提高算法的性能是一个需要解决的问题.

2.2 统计偏斜问题

多表关联规则算法在发现存在于多个表的关联规则的过程中还会产生另一个问题:统计偏斜问题(statistical skew)[14].为了将位于多个表的项进行关联,若将这些表的数据进行连接,生成一个连接表,则将改变数据在原有表中的分布特性,偏离其原有统计特性.下面仍以如图 1 所示的数据库作为例子进行说明.

假设最小支持度设为 50%,最小置信度设为 60%,则在表 1 给出的连接表中可以发现如下关联规则:

Rule: Age=18~20→Address=Haidian

容易看出,该关联规则在 S-C-SC 表中的支持度和置信度分别是

Support(Age=18~20→Address=Haidian)=60%(3/5),

Confidence(Age=18~20→Address=Haidian)=75%(3/4).

由于该关联规则中所涉及的属性均来自 Student 表,而仅利用 Student 表计算这一关联规则的支持度和置信度分别是

Support(Age=18~20→Address=Haidian)=33.3%(1/3),

Confidence(Age=18~20→Address=Haidian)=50%(1/2).

显然,这应该是所发现的关联规则正确的支持度和置信度,而通过 S-C-SC 表计算的支持度和置信度存在问题.其原因是,原存于一个表的数据经过连接操作之后,有些信息被放大了,而有些则被缩小了,偏离了原来的数据分布状况,产生了统计上的偏斜.

因此,为了使发现的关联规则最大程度地符合数据所反映的语义,计算正确的支持度和置信度,需要将统计偏斜问题作为需要解决的问题之一.

2.3 信息丢失问题

将多关系转换为单关系的另外一种方法是创建一些新的属性,将来自其他表的信息通过汇总和聚集集成在一个关系中,从而将多关系数据库转化为单一关系[17],在ILP领域,这种方法称为命题化(prepositionalization).例如,对于图 1 中的 3 个表,可以通过该方法生成如表 2 所示的单个表Student1.

Table 2 Summarizing multi-relations to single relation Student1

表 2 将多个表汇总为单个表: Student1

Sid	Age	Address	NoOfCourses	MaxGrade	SumCredit
a_1	18~20	Haidian	3	80~90	8
a_2	18~20	Xicheng	1	70~80	2
a_3	20~22	Chaoyang	1	90~100	2

然而,这种方法容易导致信息的丢失,例如,每个学生所修的每门课程的信息就不存在了.因此,原本存在于这些数据之间的关联规则也就无法发现.另外,由于找出恰当的新属性并不容易,这需要掌握大量的对域值的理解,而这本身就属于数据挖掘的任务,因而该方法实现起来并不容易.

因此,多关系关联规则的挖掘如果以挖掘存在于多个表的所有满足支持度和置信度的关联规则为目的,则应避免此类问题的发生.

已有的挖掘多关系关联规则的研究工作可以按照不同的分类标准进行不同的分类.在本文中,我们从解决上述问题的侧重面角度进行分类,可将已有工作分为两大类:一类是主要解决统计偏斜问题;另一类是解决性能问题.信息丢失问题在这两类工作中都没有出现.

解决统计偏斜问题的主要工作采用基于 ILP 的方法.解决统计偏斜问题的主要方法是,通过引入关键原子(在第 3 节中将介绍其概念)的方法,一个频繁模式对应一个查询(具体内容见下节),在查询结果中,按照关键原子的取值进行支持度的计算.因此,即使目标表中的一行元组经过连接操作之后与其他表的多行进行了连接,但其关键原子的取值只有一个,支持度的值不会因此而改变,从而避免了信息的放大和缩小.

解决性能问题的工作主要针对星型模式下的多关系进行挖掘,主要方法包括:避免连接结果的物化、避免

连接操作的执行以及采用快速的单表关联规则挖掘算法等.这些方法在一定程度上改善了多表连接的计算代价,同时,通过一些树型数据结构避免数据表的多次扫描,提高了算法的性能.

3 基于 ILP 技术的多关系关联规则挖掘方法

ILP技术是最早用于将传统数据挖掘算法扩展到多关系情况下的技术之一. ILP注重发现表达为逻辑程序的模式,最初,ILP通过示例完成一些自动的程序综合工作,现在其研究范围已扩展到数据挖掘的各项工作中.对于关联规则的挖掘,采用基于ILP的技术挖掘多表关联规则的典型算法有WARMR[18,19]和算法FARMER (finding interesting association rule groups by enumeration of rows)[20].这类算法以逻辑原子的方式表达项集,通过Prolog查询来计算项集的支持度,借鉴典型单表关联规则挖掘算法,通过分层叠代的方法发现存在于多表的关联规则.这种方法在实现挖掘多表关联规则的同时可以避免统计偏斜问题的发生.下面我们首先介绍与ILP有关的一些基本概念和术语,然后介绍这两种典型算法.

3.1 基本概念和定义

ILP领域的研究人员利用ILP技术将传统的单表关联规则挖掘算法扩展到了多表情况下.在提出的此类算法中,项集(模式,pattern)被定义为如下形式的查询(query)[18,21,22]:

$$?-A_1,A_2,\ldots,A_n$$

其中,每个A_i是一个逻辑原子(logical atom),是一个以变量或常数作为参数的谓词(predicate).

以如图 4 所示的数据库为例,这是一个由表 Kids、表 Likes、表 Has 和表 Prefers4 个关系所构成的数据库.这 4 个关系用谓词的形式可表达为:Kids(KID),Likes(KID,object),Has(KID,object),Prefers(KID,object,to).对于该数据库,一个查询的例子为

?−Kids(KID),Likes(KID,*X*),Has(KID,*Y*) (查询 1)

该查询可以转换为一个如下的 SQL 查询:

```
SELECT Kids.KID, Likes.object, Has.object
FROM Kids, Likes, Has
WHERE Kids.KID=Likes.KID AND Likes.KID=Has.KID
```

Table kids

KID
Joni
Elliot

Table likes

KID	Object
Joni	Icecream
Joni	Dolphin
Elliot	Piglet
Elliot	Gnu
Elliot	Lion

Table Has

KID	Object
Joni	Icecream
Joni	Piglet
Elliot	Icecream

Table prefers

KID	Object	To
Joni	Icecream	Dolphin
Joni	Pudding	Raisins
Joni	Giraffe	Gnu
Elliot	Lion	Icecream
Elliot	Piglet	Dolphin

Fig.4 Database with four relations

图 4 由 4 个关系组成的数据库

显然,从数据库的角度,该查询可返回 7 行,包括(Joni,icecream,icecream),(Joni,icecream,piglet),(Joni,dolphin,icecream),(Joni,dolphin,piglet),(Elliot,Piglet,icecream),(Elliot,gnu,icecream),(Elliot,Lion,icecream).从 ILP 的角度来看,查询1的结果是由替换θ(substitution θ)构成的,一个替换θ是由一组数据库中的常数构成的,其中每个常数是对查询中一个原子中的变量的实例化,即将变量用常数替换.例如,(Joni,icecream,piglet)可看作一个替换,其中,Joni 替换变量 KID,icecream 替换 *X*,而 piglet 替换 *Y*.

为了定义一个查询的支持度(或称为 frequency,频率),对于关联规则的挖掘任务,需要指定另外一个参数:关键原子(key atom).该原子需要在每个查询中都出现,它决定了支持度的计算方法.例如对于查询 1 来说,如果指定 Kids(KID)是关键原子,则查询结果中 KID 的不同取值的个数就是此查询的绝对支持度.因此,尽管查询的结果中有 7 种情况存在,但只涉及两个孩子的情况,故绝对支持度为 2.相对支持度需要把绝对支持度除以关键原子中的变量用表中的常数替换后的不同替换情况的个数,在本例中只有两个孩子存在,也为 2.因此,相对支持度

为 100%.

一个查询,如果其支持度符合用户定义的最小支持度则称为一个频繁查询.对于两个查询Q_1=?–l_1,…,l_m和Q_2=?–l_1,…,l_m,l_{m+1},…,l_n,如果对于Q_1的每个替换,用常数替换后的每个原子均出现在Q_2的某个替换对应的替换后的原子中,则称$Q_1\theta$-包含Q_2,即Q_1是比Q_2更一般的查询,Q_2是比Q_1更具体的查询.例如,若Q_1=?–Kids(KID),Likes(KID,*X*),Has(KID,*Y*),而Q_2=?–Kids(KID),Likes(KID,*X*),Has(KID,*Y*),Prefer(KID,*X*,*Y*),则$Q_1\theta$-包含Q_2,也称Q_2是Q_1的特殊化(specialization).

对于两个频繁的查询Q_1=?–l_1,…,l_m和Q_2=?–l_1,…,l_m,l_{m+1},…,l_n,满足$Q_1\theta$-包含Q_2的关系,如果查询Q_2的支持度除以查询Q_1的支持度大于等于最小置信度,则称$Q_1\rightarrow Q_2$为一个关系关联规则(relational association rule),又称为查询扩展(query extension),可简写为l_1,…,$l_m\rightarrow l_{m+1}$,…,l_n.以图 4 中的数据库为例,关系关联规则的一个例 子为

Kids(KID),Likes(KID,*X*),Has(KID,*Y*)→Prefers(KID,*X*,*Y*) (查询 2)

其中,Q_1=?–Kids(KID),Likes(KID,*X*),Has(KID,*Y*),Q_2=?–Kids(KID),Likes(KID,*X*),Has(KID,*Y*),Prefer(KID,*X*,*Y*),满足Q_2的一个替换为:Kids(Elliot),Likes(Elliot,lion),Has(Elliot,icecream),Prefers(Elliot,lion,icecream).满足Q_1的替换如前所述为 2 个Kid:Joni和Elliot.因此,该规则的绝对支持度为 1,置信度为 50%.

3.2 算法WARMR

WARMR[18,19]是一个典型的利用ILP技术实现多关系关联规则挖掘的算法.它将著名的单表关联规则挖掘算法Apriori算法[2]升级用于多关系情况下关联规则的发现.Apriori从空项集开始,在每一层寻找项的个数等于层数的那些频繁项集,大频繁项集可以通过向一个频繁项集中加入项而生成,即,第l+1 层的候选项集是通过向在第l层获得的频繁项集中添加项而获得的.Apriori算法的有效性是基于“一个频繁项集的所有子集一定是频繁的”这一事实,只有通过这一检验的候选项集才能通过扫描数据库确定它们是否为频繁项集.

WARMR算法的核心思想与Apriori算法类似,也采用层次搜索的方法,在给定数据库*D*中查找频繁Prolog查询.它首先按照层次叠代的方法产生候选查询,而后计算候选查询的支持度,从中确定频繁项集.在初始情况下,层次为 1 时,查询由关键原子构成.从中发现,频繁的查询之后,在以后的各层次,利用θ-包含的特殊化语义,将每一个l层的频繁查询Q_l追加一个原子,从而构成一个新的候选查询Q_{l+1},使得$Q_l\theta$-包含Q_{l+1}.在对候选查询进行支持度计算时,每个候选查询提交Prolog查询系统执行,查询结果中不同的关键原子的替换个数作为该查询的支持度.该算法的主要步骤如图 5 所示.

```
Algorithm. WARMR.
Inputs: Database D, Minimal support minsup, Key atom key.
Outputs: Frequent query set F.
Steps:
1.  Initialize: level d=1, Frequent Query Set F=∅;Infrequent query set I=∅;
2.  Generate candidate queries of length 1: Q1={?–key};
3.  while Qd is not empty
4.      Calculate the frequency(support) of each query in Qd;
5.      Move each query with frequency <=minsup to set I.
6.      F=F∪Qd
7.      Generate candidate queries of length d+1 Qd+1;
8.      d=d+1;
9.  return F;
```

Fig.5 Major steps of algorithm WARMR

图 5 算法 WARMR 的主要步骤

在算法 Apriori 中,一个频繁项集的所有子集一定是频繁的;在 WARMR 中,一个频繁项集对应的 prolog 查询的所有子查询并非都是频繁的.假设如下查询?–person(*X*),parent(*X*,*Y*),hasPet(*Y*,cat)是频繁的,则不允许?–person(*X*),hasPet(*Y*,cat)这样的子查询,因为它违反了 hasPet 谓词的第 1 个参数必须较早地出现在查询中这一约束,这会导致在剪裁生成的候选频繁查询时使问题复杂化.因而,WARMR 保持了一个不频繁查询列表,并检验在列表中生成的候选项集是否被某个查询所包含.

WARMR是在多关系情况下对Apriori算法[3]的升级.主要的不同是确定查询(即项集)的支持度和候选查询的生成方法.Apriori可看作是WARMR的一个特例,即只解决单谓词无变量情况下的频繁项集的发现.

WARMR 算法的优点是利用原子表达项集,表达力更强.另外,由于设计了一个额外的参数——关键原子,使得支持度的计算不会出现统计偏斜的问题.由此可以看到,凡是采用 ILP 技术的多关系关联规测挖掘算法,由于都要通过引进关键原子来定义查询的支持度,这使得统计偏斜问题在这类算法中得到自然的解决.

WARMR 算法的主要缺点是计算复杂度高,使得算法效率不高.这主要是因为 WARMR 极大地依赖于θ-包含计算,而θ-包含的计算实际上是一个 NP 完全问题.另外,由于关键原子参数的指定,WARMR 每次实际上只发现完全的关联规则集合中的一个子集,即与关键原子有关的关联规则.要发现所有存在于各个表中的关联规则,需要变换关键原子,重复执行该算法.这也是采用 ILP 技术的多关系关联规则挖掘算法中都存在的缺点.

3.3 算法FARMER

为了对WARMR算法进行改进,改善其运行效率,算法FARMER[20]被提出来,用于快速发现存在于多表中的频繁项集.与WARMR相同,FARMER仍然沿用一阶逻辑的表达方式来定义项集(即查询)及其支持度.为了提高算法的运行效率,降低计算复杂度,该算法采用以下策略:

- 采用一种复杂的数据结构——Trie树来存储和操纵查询.Trie树中从根节点到叶子节点的每条路径都对应一个查询.利用该树进行查询的生成和支持度的计算,比采用Hash树效率要高[23].
- 生成新的查询时不采用θ-包含计算,因为该计算非常耗时.代替θ-包含计算的方法是,为每个模式声明(mode declaration,用于说明谓词在查询中的使用方式)创建一个多维矩阵,以便快速判断一个原子是否可以追加到当前的查询中.FARMER 将此类原子分为 3 个类别,借助该矩阵和 Trie 树生成查询.经证明,在满足特定条件(限定类型的语言偏置)的情况下,利用该方法生成的查询与 WARMR 相同.

实验结果表明,FARMER 在运行效率方面与 WARMR 相比有较大的改善.

3.4 算法WARMR与FARMER的比较和分析

WARMR和FARMER两种算法是基于ILP技术发现多关系关联规则的代表,在实际应用中都得到了应 用[24−26].两者的共同点在于都是利用ILP中的一阶谓词逻辑来定义项集,将项集对应为查询.这种表达方式拓展了传统的项集的表达能力,能够发现比较复杂的模式.两者的主要区别在于以下几点:

- WARMR 借鉴传统的 Apriori 算法的实现方法而实现,而 FARMER 采用改进了的 Apriori 算法来实现,通过 Trie 树数据结构的使用,加快项集的计数和生成过程.
- WARMR 采用θ-包含计算的方法来进行项集的生成和剪裁,可以剪裁掉较多的候选项集.FARMER 则摒弃了θ-包含计算,通过限定语言偏置的类型,实现发现与 WARMR 同样的项集.
- WARMR 与 FARMER 相比,在语言偏置方面更灵活,但执行效率较低,FARMER 对于语言偏置有所限制,但效率相对更高.另外,由于 FARMER 需要在内存保存多个矩阵,因此,耗用的内存也相应要高.

4 星型模式下的多关系关联规则挖掘算法

在已有工作中,多数多关系关联规则挖掘算法是为了解决性能问题而提出来的,这类工作都是针对星型模式下的多关系数据进行挖掘算法的研究,这种结构与实际应用中多数采用的实体联系模型模拟的数据库相比,具有结构简单的特点.对于这种结构的数据库,为了挖掘满足条件的频繁模式以及频繁闭合模式,已有工作主要在性能方面进行考虑.提高性能的方法主要集中于连接操作的避免以及数据扫描次数的降低.与基于ILP技术的多关系关联规则挖掘方法不同,这些工作没有解决统计偏斜问题.这类工作的典型算法包括:JSApriori算法[27]、masl算法、masb算法[28]以及MultiClose算法[29]等.下面分别加以介绍.

4.1 算法JSApriori

当数据库中的表以星形模式组织在一起时,文献[27]提出一种采用非 ILP 方法的算法(为便于描述,我们称其为 JSApriori),该算法分为两个阶段:在第 1 阶段,通过分析事实(在文献[27]中称为联系表,relationship table)表的外键到维表(即原文中的主表,primary table)主键之间的联系,算法首先在每个单独的表上运用修改的 Apriori 算法找寻频繁项集;在第 2 阶段,将属于不同表的频繁项集进行合并,寻找合并后仍频繁的项集.该算法可以得出与将 Apriori 算法运行在连接的表上相同的结果,而且具有更好的性能.

算法的具体步骤如下:

(1) 在每个单独的维表上寻找频繁项集

假设星型模式由n个维表$T_1,T_2,\ldots,T_n$和一个事实表T_{1n}构成.例如,在如图 1 所示的 3 个表中,Student和Course可看作维表,SC可看作事实表.每个维表$T_t(id_t,a_{t1},a_{t2},\ldots,a_{tmt})$的主键是$id_t$,事实表为$T_{1n}(id_1,id_2,\ldots,id_n)$,其中的每个$id_t$是事实表的外键,是相应的$T_t$维表的主键.算法要发现连接表$T=T_{1n}\bowtie T_1\bowtie T_2\bowtie\ldots\bowtie T_n$中的频繁项集.

先通过事实表T_{1n}将每个外键id_t在表中出现的不同值的次数统计出来,用一个向量将外键id_t的每个取值存储起来,向量中元素的个数即是id_t取不同值的数目.由于每个外键对应一个维表T_t,因而一个向量就为一个维表存储了其主键的每个取值在事实表中出现的次数.

接下来,计算维表中的频繁项集.项集是否频繁是以该项集在事实表中出现的次数来统计的,这是对经典的Apriori算法的改进.利用上一步在向量中记录的外键取值数,就可以将维表中在将来的连接表中的频繁项集找出来.如此找出的有n个频繁项集,每一个频繁项集合l_t中的频繁项集的长度可为 1~m_t.

(2) 利用事实表发现跨维表的频繁项集

用n维的记数数组记录来自n(取值为 2~n)个维表的候选项集,其中第t维元素对应于表T_t的频繁项集集合l_t中的项集或空项集.

根据事实表和n个维表,可逐行生成这些表的连接表T.每生成这样的一行连接元组,马上处理它,处理完之后不保留该行,接着生成和处理下一行.不实际地物化连接表T的每一行,是该算法的主要特点.针对生成的T表的每一行r,首先按照每个维表T_t的属性作投影$\pi_{Tt}(r)$,得到相当于维表T_t的一行,找出其中包含的属于频繁项集集合l_t中的所有项集(包括一个空项集),构成集合i_t.这样对所有维表的属性进行投影处理之后,就可以得到n个集合:$i_1,i_2,\ldots,i_n$.然后通过将这些集合中的项集进行组合,即$I_t=i_1\times i_2\times\ldots\times i_n$,得到所有候选项集$I_t$.对于每个这样的候选项集在$n$维计数数组的相应位置记录下来(若已经存在,则将其计数累加 1),接着处理下一行.将T的所有行处理完毕之后,则在n维数组中记录了所有候选项集的支持度.据此,可以确定哪些项集是频繁的.

该算法先后采用两步完成不同的处理工作,连接表 T 只被计算和处理 1 次,因此,不必存储和完整地生成它,这是该算法比直接在连接表上运行 Apriori 算法效率有所提高的主要原因.

4.2 算法masl与masb

Masl与masb算法[28]是为了改进文献[27]中的算法而提出来的.文献[27]中提出的算法需要将各个表进行连接但不需要物化该连接结果,它是对生成的每一个连接行进行处理.由于缺乏剪裁机制,因此,内存和计算代价都很高.Masl与masb算法的特点是不需要进行自然连接操作,因而在一定程度上提高了算法的性能.两种算法的区别在于,在实现tid-list时,前者采用linked list而后者采用位图矩阵和计数数组的数据结构.它们的主要步骤相同.

假设有 2 个维度表:A,B 和一个事实表 FT.算法的主要步骤如下:

(1) 初始化

对每个维度表扫描一次,收集每个 1-项集对应的行号列表,即tid-list.对事实表FT扫描一次,统计每个维度表的行号tid在FT中出现的次数以及对于每个维度表的行号,如A表中的a_i与B表中的行号共同出现在FT表中的次数,记为$B_key(a_i)$.例如,对于图 6 中的两个维度表A,B和一个事实表FT,$B\_key(a_1)=\{b_2(1),b_3(1),b_5(2)\}$.

(2) 预处理

读取维度表,并将其进行转置转换为 VTV(vertical tid-vector)格式的表,记录每个属性的每个取值出现在哪些行中,各行用行号 tid 表示,并用一个数组记录每个维度表中每个主键在事实表中出现的次数.如图 6 所示.

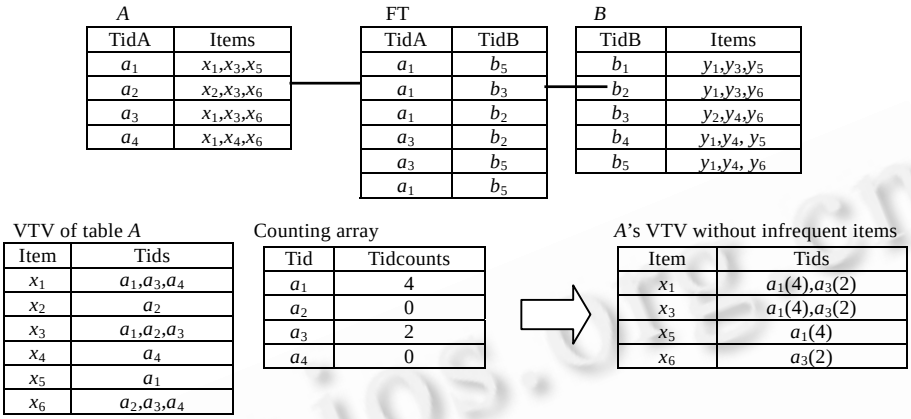


Fig.6 VTV format table and counting array of table *A*

图 6 维度表 *A* 表的 VTV 格式及其计数数组

(3) 局部挖掘

对每个维度表进行局部挖掘.这一步可以采用任何单表频繁项集挖掘算法,如 FP-growth,对其进行稍加修改即可.修改主要是为了计算单表中的项集在事实表中出现的次数,需要用到在初始化步骤中统计的值(每个维度表的行号 tid 在 FT 中出现的次数).该步的时间消耗比起下一步来并不显著.

(4) 全局挖掘

此步骤又细分为 3 个步骤来完成.

第 1 步:扫描事实表.

扫描事实表 FT,并利用前缀树将 FT 中各个行号的信息进行存储.如果维度表有 4 个:*A*,*B*,*C* 和 *D*,则事实表(图 7 中显示在左侧的表)可用图 7 右侧所示的前缀树表示.

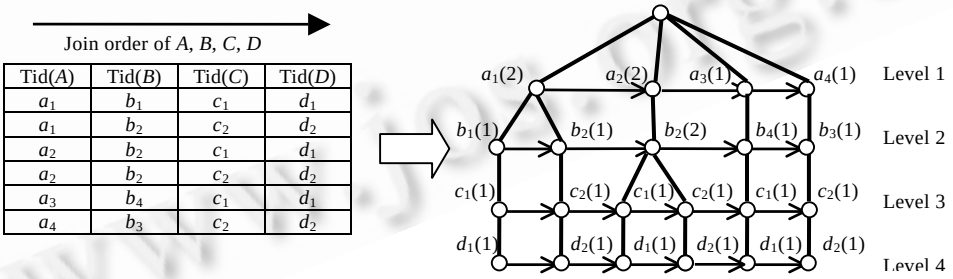


Fig.7 Fact table FT and its prefix tree

图 7 事实表 FT 及其前缀树

对维度表设定一个处理顺序,假设 4 个维度表的处理顺序为 *A*,*B*,*C*,*D*,则按如下 2 个步骤继续处理.

第 2 步:挖掘 2 项项集.

生成 2 项项集,其中一个项来自 *A* 表,另一个来自 *B* 表.计算其在 FT 表中出现的频率,主要依据图 7 中的前缀树所存储的行号信息进行计算,检验是否满足最小支持度要求.

第 3 步:为表 *A* 和表 *B* 挖掘其余项集.

由 *k* 项集生成候选 *k*+1 项集(*k*=2,3,…),两个 *k* 项集若前 *k*–1 项相同,最后一项不同,则生成一个 *k*+1 项集,方

法类似于 Apriori_gen,具体可参考文献[30].然后计算每个新生成的项集的频率,找出其中的频繁项集.此步骤重复执行,直至生成的 k+1 项集集合为空.

第 2 步和第 3 步又称为维度表 A 和表 B 的绑定(binding)过程.第 2 步和第 3 步完成之后,发现的频繁项集来自 A,B 和 AB.把这些频繁项集看作是从一个单一的表 AB 中发现,然后重复第 2 步和第 3 步绑定表 AB 和 C 以挖掘来自表 AB 和表 C 的所有项集,这些项集包含的项部分来自表 AB,另一部分来自表 C.接下来,再重复第 2 步和第 3 步绑定 ABC 和 D,以挖掘来自表 ABC 和表 D 的所有项集,这样,就可以得到来自这个数据库的所有的频繁项集.

实验表明,该方法比将所有表连接之后使用人们非常熟悉的一些高效的单表算法,如基于 fp-tree 的算法 fp-growth,性能要好.

4.3 算法MultiClose

MultiClose 算法与 masl 与 masb 算法相似,不同之处在于,前者发现的是频繁闭合项集而不是频繁项集.如同单表中的频繁闭合项集与频繁项集的区别和事实,在多关系情况下,频繁闭合项集是频繁项集的一种无损压缩,通常可以大幅度减少所发现的项集的个数.

MultiClose 算法的主要过程与 masl 与 masb 算法相似,分为如下的 3 个主要步骤:

(1) 预处理

扫描事实表,统计每个 tid 出现的次数,构建前缀树以记录事实表信息.然后扫描维表,将其转化为垂直格式表示,即每个属性取值对应一个 tid 列表.

(2) 局部挖掘

为每个维表采用CHARM算法[10]单独挖掘满足条件的单表频繁闭合项集.为了处理多关系情况,将CHARM进行修改,使其可以计算单表项集在连接表中的支持度.另外,为了加快项集是否频繁的检验,采用一个两层哈希表结果树,该数据结构优于CHARM中原来采用的方法.

(3) 全局挖掘

利用局部挖掘中得到的频繁闭合项集以及前缀树进行跨表频繁闭合项集的挖掘.经证明,一个跨表频繁项集 XY 如果是闭合的,其中 X 和 Y 分别是一个单表频繁项集,则 X 和 Y 分别是所在关系的闭合项集.全局挖掘的主要步骤与 masl 与 masb 算法类似,在此不再赘述.

实验结果表明,MultiClose 在运行时间上优于 CHARM.

4.4 算法比较

文献[27]提出在只包含一个星型模式组织的多表数据库情况下,如何将经典的 Apriori 算法进行修改以挖掘存在于多个表之间的关联规则,所提出的 JSApriori 算法的主要特点是避免将多个表的连接结果保存后再进行挖掘.由于连接结果通常较大,因此,这在一定程度上提高了算法的效率.但是,由于处理过程中无法进行候选项集的剪裁,n 维计数数组需要同时记录任意大小的候选项集,因此,需要大量内存.另外,连接表虽然不需要保存,但每一行都需要生成和处理,这也是比较耗费时间的.

为了进一步提高运行效率,文献[28]中提出的 masl 与 masb 两种算法通过单独处理维表和事实表发现所有单表和跨表的频繁项集,避免了将多个表进行连接,有效地提高了挖掘效率.

文献[29]提出的 MultiClose 算法挖掘频繁闭合项集.该算法利用与 masl 与 masb 两种算法类似的方法进行处理,不同之处在于发现时闭合项集,而不是频繁项集,闭合项集是一种频繁项集的压缩.MultiClose 算法利用项集的闭合特性,通过单独处理维表和事实表发现其中的频繁闭合项集,避免了表之间的连接操作.

这些算法的共同点在于,它们主要致力于避免连接操作而提高算法的性能,没有考虑可能出现的统计偏斜问题.因而对于单表项集,其支持度的计算是相对于连接表而言的.

5 总结与未来研究展望

鉴于在实际应用中数据事实上都是存放在数据库的多个表中,因而多关系数据挖掘近年来成为一个研究热点,其中,多关系关联规则的研究也引起了很多人的关注,提出了各种算法.本文定义了多关系关联规则挖掘算法研究的一个总体框架,总结了这些算法需要解决的 3 大问题,并按照解决这些问题的侧重面不同,将已有工作分为两大类:解决统计偏斜问题的基于 ILP 技术的研究工作和重点解决性能问题的针对星型模式的数据库的工作.就这两类工作,我们介绍了各种典型代表算法的主要步骤和方法.由于多关系数据库相对于单表数据集而言,数据环境更加复杂,需要解决的问题还有很多,特别是在以下方面还需要研究人员做进一步深入的研究:

- 处理结构复杂的数据库

尽管很多已有工作都是针对性能的提高而展开的,但是,它们绝大多数以星型模式数据库为研究对象.在这种情况下,只需考虑单个维表的处理问题,以及各维表经过事实表连接后的连接表的处理问题.在实际应用中,更具有普遍意义的是以实体联系模型对应的数据库,远比星型模式复杂得多.虽然有些研究人员提出的方法原则上可以经过一定的修改推广到更一般的数据库,但是对于复杂的数据库结构,不仅存在项集的组合爆炸问题,还存在表之间组合的爆炸问题以及属性间的组合问题,这就对算法的性能提出了进一步的挑战.在这方面,可以借鉴一些其他方法,如多关系分类算法CrossMine[31]中所提出的虚拟连接方法以及传统数据挖掘中的高维度频繁模式的发现方法[12,32]等,以便降低问题的难度,提高系统的性能,使得多关系关联规则的挖掘真正成为可能.

- 优化关联规则的发现和输出

在一个大型数据库中,表之间的连接路径通常很多、很长,为了发现可能存在于所有连接路径中的规则,需要花费很多时间.但是通常连接路径越长,表之间的语义关系可能就越弱,属性值之间存在关联的可能性也越小.因此,如何利用表之间的语义联系以及应用背景的领域知识(domain knowledge),尽快发现有意义的关联规则,是需要研究人员认真研究的问题.该问题的解决不仅可以输出让用户更感兴趣的规则,而且也可以同时缩减算法的搜索空间,提高算法的运行效率.为了使得多关系关联规则算法能够利用领域知识,可以考虑使算法具有使用元数据或自描述数据的能力.

- 避免统计偏斜的新方法

基于 ILP 技术的关联规则挖掘算法可以解决统计偏斜问题,但 ILP 方法存在许多缺陷,例如,它主要面向演绎数据库,不能直接处理关系数据库,而关系数据库在实际应用中使用更为广泛;再者,由于它生成候选查询的方式不同于传统的候选项集的生成方法,计算复杂度更高,算法性能受到影响.另外,它解决统计偏斜问题的方法依赖于一个新增的参数——关键原子,该参数难以被普通用户理解.因此,如何借鉴 ILP 方法在不损失系统性能的前提下解决统计偏斜问题需要进一步的研究.最后,利用 ILP 技术发现的模式通常不易被用户理解,限制了此类算法在实际中的使用.因而,研究基于数据库技术的避免多表连接过程中的统计偏斜问题的方法是很有必要的.

- 研究多关系关联规则的新型实现技术

多关系数据挖掘面向的是存储于多个表中的数据,然而,对于关系数据库中的数据,基于ILP技术的关联规则挖掘算法需要将其转换为ILP可以处理的用谓词表达的事实;而对于其他算法,则将各个表看作数据集合,或者仅将表看作存放数据的容器,没有充分利用数据库系统对数据的处理能力.因此,挖掘算法与数据库之间的耦合度非常低,很少使用数据库现成的技术,如查询处理和优化等.提高挖掘系统与数据库系统的耦合度有许多优点[33].例如,可以降低挖掘算法的实现复杂度、提供算法运行效率、改善挖掘系统使用的易用性、进一步利用数据库方面的投资,等等.在多关系数据挖掘的其他技术中,例如分类技术,已有这方面的工作[34].另外,在传统数据挖掘领域,很多算法,如关联规则、序列模式以及分类等类型的算法,如何利用数据库技术实现的研究已有很多.因而,借鉴这些工作的方法研究与数据库紧密结合的多关系关联规则算法的实现技术具有重要的实际意义.

- 拓展多关系关联规则的应用

传统的关联规则有各种各样的应用.例如,利用关联规则创建分类器、利用频繁模式辅助聚类等等.在多关系的数据环境下,同样的应用需求仍然存在,如何将多关系关联规则算法进行扩展以便满足不同的应用需求是

很有意义的研究工作,可以发现多关系关联规则更多的用武之地.例如在传统数据挖掘算法中,基于关联规则的分类算法具有能够发现更多有意义分类规则的特点,因此,常常可以取得更高的分类准确度.同样,在多关系数据环境下,基于关联的分类模型的构造方法非常值得研究.另外,在单表数据集情况下,关联规则的应用领域非常广泛,例如多媒体数据.在多关系关联规则的应用领域也值得进一步探究.

总之,随着数据挖掘技术的发展,多关系数据挖掘逐渐成为数据挖掘的一个新的研究分支,鉴于它与实际应用数据环境的贴合性,其研究越来越受到国际上众多研究者的关注.关联规则技术是数据挖掘中的一种重要的基础算法,尽管研究多关系关联规则的挖掘的工作已有不少,但各有其优缺点,如何将其优点进行融合,同时解决各种问题,对研究者提出了挑战,值得我们进一步深入研究.

References:

[1] Domingos P. Prospects and challenges for multi-relational data mining. ACM SIGKDD Explorations Newsletter, 2003,5(1):80−83.

[2] Agrawal R, Imilienski T, Swami A. Mining association rules between sets of items in large datasets. In: Buneman P, Jajodia S, eds. Proc. of the 1996 ACM SIGMOD Int'l Conf. on Management of Data. New York: ACM Press, 1993. 207−216.

[3] Agrawal R, Srikant R. Fast algorithm for mining association rules. In: Bocca JB, Jarke M, Zaniolo Z, eds. Proc. of the 20th Int'l Conf. on Very Large Data Bases. Burlington: Morgan Kaufmann Publishers, 1994. 487−499.

[4] Mannila H, Toivonen H, Verkamo AI. Efficient algorithms for discovering association rules. In: Fayyad UM, Uthurusamy R, eds. Proc. of the 1994 AAAI Workshop Knowledge Discovery in Databases (KDD'94). AAAI Press, 1994. 181−192.

[5] Savasere A, Omiecinski E, Navathe S. An efficient algorithm for mining association rules in large databases. In: Dayal U, Gray PMD, Nishio S, eds. Proc. of the 21st Int'l Conf. on VLDB. Burlington: Morgan Kaufmann Publishers, 1995. 432−443.

[6] Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation. In: Chen WD, Naughton JF, Bernstein PA, eds. Proc. of the 2000 ACM SIGMOD Int'l Conf. on Management of Data. New York: ACM Press, 2000. 1−12.

[7] Zaki MJ, Parthasarathy S, Ogihara M, Li W. New algorithms for fast discovery of association rules. In: Heckerman D, Mannila H, Pregibon D, Uthurusamy R, eds. Proc. of the 3rd Int'l Conf. on Knowledge Discovery and Data Mining. AAAI Press, 1997. 283−286.

[8] Grahne G, Zhu J. Efficiently using prefix-trees in mining frequent itemsets. In: Goethals B, Zaki MJ, eds. Proc. of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI 2003), CEUR Workshop Proc., Vol-90. 2003. http://CEUR-WS.org/Vol-90/

[9] Pasquier N, Bastide Y, Taouil R, Lakhal L. Discovering frequent closed itemsets for association rules. In: Beeri C, Buneman P, eds. Proc. of the 7th Int'l Conf. Database Theory (ICDT'99). LNCS 1540, Berlin: Springer-Verlag, 1999. 398−416.

[10] Zaki MJ, Hsiao CJ. CHARM: An efficient algorithm for closed association rule mining. Technical Report, TR99-10, Department of Computer Science, Rensselaer Polytechnic Institute, 1999.

[11] Pei J, Han J, Mao R. CLOSET: An efficient logarithm for mining frequent closed itemsets. In: Gunopulos D, Rastogi R, eds. Proc. of the 2000 ACM SIGMOD Int'l Workshop Data Mining and Knowledge Discovery (DMKD 2000). 2000. 11−20.

[12] Pan F, Cong G, Tung AKH, Yang J, Zaki MJ. CARPENTER: Finding closed patterns in long biological datasets. In: Getoor L, Senator TE, Domingos P, Faloutsos C, eds. Proc. of the 9th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM Press, 2003. 637−642.

[13] Liu HY, Han J, Xin D, Shao Z. Top-Down mining of interesting patterns from very high dimensional data. In: Liu L, Reuter A, Whang KY, Zhang JJ, eds. Proc. of the 22nd IEEE Int'l Conf. on Data Engineering (ICDE 2006). Washington: IEEE Computer Society, 2006. 114−119.

[14] Getoor L. Multi-Relational data mining using probabilistic relational models: Research summary. In: Knobbe AJ, van der Wallen DMG, eds. Proc. of the 1st Workshop in Multi-Relational Data Mining. 2001. 6−17. http://mrdm.dantec.nl/procmrdm.pdf.

[15] Dzeroski S, LAvrac N. Relational Data Mining. New York: Springer-Verlag, 2001.

[16] Raghu R, Gehrke J. Database Management Systems. 2nd ed., New York: McGraw-Hill, 2000. 97−99.

[17] Kramer S, Lavrac N, Flach P. Propositonalization approaches to relational data mining. In: Dzeroski S, Lavrac N, eds. Relational Data Mining. New York: Springer-Verlag, 2001. 262−291.

[18] Dehaspe L, de Raedt L. Mining association rules in multiple relations. In: Dzeroski S, Lavrac N, eds. Proc. of the 7th Int'l Workshop on Inductive Logic Programming. LNAI 1297, Berlin: Springer-Verlag, 1997. 125−132.

[19] Dehape L. Frequent pattern discovery in first-order logic [Ph.D. Thesis]. Belgium: Katholieke Universiteit Leuven, 1998.

[20] Nijssen S, Kok J. Faster association rules for multiple relations. In: Nebel B, ed. Proc. of the 17th Int'l Joint Conf. on Artificial Intelligence (IJCAI 2001), Vol.2. 2001. 891−896.

[21] Dehaspe L, Toivonen H. Discovery of frequent datalog patterns. Data Mining and Knowledge Discovery, 1999,3(1):7−36.

[22] Dehaspe L, Toivonen H. Discovery of relational association rules. In: Dzeroski S, Lavrac N, eds. Relational Data Mining. New York: Springer-Verlag, 2001. 189−208.

[23] Pijls W, Bioch JC. Mining frequent itemsets in memory-resident databases. In: Postma E, Gyssens M, eds. Proc. of the 11th Belgium-Netherlands Artificial Intelligence Conf. (BNAIC'99). Berlin: Springer-Verlag, 1999. 75−82.

[24] Dehaspe L, Toivonen H, King RD. Finding frequent substructures in chemical compounds. In: Agrawal R, Stolorz P, Piatetsky-Shapiro G, eds. Proc. of the 4th Int'l Conf. on Knowledge Discovery and Data Mining. Menlo Park: AAAI Press, 1998. 30−36.

[25] King RD, Karwath A, Clare A, Dehaspe L. Genome scale prediction of protein functional class from sequence using data mining. In: Simoff SJ, Zaïane OR, eds. Proc. of the 6th Int'l Conf. on Knowledge Discovery and Data Mining (SIGKDD). New York: ACM Press, 2000. 384−389.

[26] Malerba D, Esposito F, Lisi FA, Appice A. Mining spatial association rules in census data. Research in Official Statistics, 2002, 5(1):19−44.

[27] Jensen VC, Soparkar N. Frequent itemset counting across multiple tables. In: Terano T, Liu H, Chen ALP, eds. Proc. of the 4th Pacific-Asia Conf. of Knowledge Discovery and Data Mining, Current Issues and New Applications. LNCS 1805, Berlin: Springer-Verlag, 2000. 49−61.

[28] Ng EKK, Fu AW, Wang K. Mining association rules from stars. In: Kumar V, Tsumoto S, eds. Proc. of the 2002 IEEE Int'l Conf. on Data Mining (ICDM 2002). Los Alamitos: IEEE Computer Society, 2002. 322−329.

[29] Xu L, Xie K. A novel algorithm for frequent itemset mining in data warehouses. Journal of Zhejiang University, 2006,7(2): 216−224.

[30] Shenoy P, Haritsa JR, Sudarshan S. Turbo-Charging vertical mining of large databases. In: Chen WD, Naughton JF, Bernstein PA, eds. Proc. of the 2000 ACM SIGMOD Int'l Conf. on Management of Data. New York: ACM Press, 2000. 22−33.

[31] Yin X, Han J, Yang J, Yu PS. CrossMine: Efficient classification across multiple database relations. In: Ozsoyoglu M, Zdonik S, eds, Proc. of the 2004 Int'l Conf. on Data Engineering (ICDE2004). Los Alamitos: IEEE Computer Society, 2004. 399−411.

[32] Liu HY, Han J, Xin D, Shao Z. Mining interesting patterns from very high dimensional data: A top-down row enumeration approach. In: Ghosh J, Lambert D, Skillicorn DB, Srivastava J, eds. Proc. of the 6th SIAM Int'l Conf. on Data Mining. Philadelphia: Society for Industrial Mathematics, 2006. 280−292.

[33] Sarawagi S, Thomas S, Agrawal R. Integrating association rule mining with relational database systems: Alternatives and implications. In: Haas LM, Tiwary A, eds. Proc. of the ACM SIGMOD Conf. on Management of Data. New York: ACM Press, 1998. 343−354.

[34] Ceci M, Appice A, Malerba D. Mr-SBC: A multi-relational naive Bayes classifier. In: Lavrac N, Gamberger D, Todorovski L, Blockeel H, eds. Proc. of the Knowledge Discovery in Databases PKDD 2003. LNAI 2838, Berlin: Springer-Verlag, 2003. 95−106.

何军(1962－),男,北京人,博士生,副教授,主要研究领域为数据库,数据挖掘,信息检索,计算机网络.

杜小勇(1963－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为智能信息检索,高性能数据库,知识工程.

刘红岩(1968－),女,博士,副教授,CCF 高级会员,主要研究领域为数据库,数据仓库,数据挖掘,神经网络.