大型数据库中多层关联规则的元模式制导发现*

*#欧阳为民 #蔡庆生

*(安徽大学计算中心 合肥 230039)

#(中国科学技术大学计算机系 合肥 230027)

E-mail:oywm@mars.ahu.edu.cn

摘要 本文将元查询制导技术与多层关联规则发现技术结合起来,提出了发现多层关联规则的自顶向下逐层递进风格的元模式制导方法.元模式是一个预先确定待发现规则的形式的规则模板,从而可引导数据的发掘过程.

关键词 知识发现,元模式,关联规则,概念层次.

中图法分类号 TP301

在数据库中发现知识 KDD(knowledge discovery in databases)亦称为数据发掘(Data Mining),是当今国际人工智能和数据库研究的最富活力的新兴领域,其目标是在大量的数据中发现令人感兴趣的模式.[1~3]由于其强大的应用潜力以及广泛可用的,存在于各种数据库中的巨量数据,KDD 成为一个具有迫切现实需要的很有前途的研究课题.

在近年研究中,人们通过集成数据库、机器学习和统计技术已经提出了很多数据发掘方法.[4]然而,存在这样一种常见现象,即虽然 KDD 系统发现了相当多的模式或知识,但是往往或者缺乏重点,或者对用户缺乏兴趣.我们认为导致这一现象的两个重要因素是:① 未选择待研究的与任务相关的数据集,即缺乏适当的聚焦;② 未限制模式或知识的种类或形式,即缺乏适当的约束.第 1 个问题可通过引入一种指定与特定发掘任务相关的数据集的数据发掘接口来解决.例如,加拿大 Simon Fraser 大学 Han Jiawei 教授主持研制的 DBMiner[5]系统就是一个很好的范例,它用类-SQL 接口来为数据发掘查询确定与任务相关的数据集.这样,如果我们希望发现计算机系研究生的一般特点,那么可以用一个 where 子句来限定只对上述学生进行检索.第 2 个问题就不能这样直截了当地解决了.事实上,我们有多种方法来说明待发现知识的类型或形式.例如,待发现知识的类型可以说明为特征规则(Characteristic Rule)、分类规则(Classification Rule)、关联规则(Association Rule)等,也可以指定

* 本文研究得到国家自然科学基金和国家教委博士点基金资助.作者**欧阳为民**,1964 年生,博士生,副教授,主要研究领域为 KDD,机器学习,人工智能及其应用.**蔡庆生**,1938 年生,教授,博士导师,主要研究领域为机器学习,知识发现,人工智能.

本文通讯联系人:欧阳为民,合肥 230039,安徽大学计算中心

本文 1997-05-16 收到修改稿

(广义)规则或模式的析取项的个数,即指定每个广义属性的不同值的最大(期望)个数,或者指定广义关系的元组个数,而且,我们还可以对待发现知识作语法或语义上的限制.

W. Shen 等在文献[6]中提出了一个称为元查询(Metaquery)的技术,在数据发掘过程中指定待发现规则的形式.这样一个元查询不仅表达了待发现规则的期望形式,而且用作为知识发现系统的重要用户界面.在他们的元查询制导的数据发掘研究中,待发现规则被限制为单一概念级.我们认为,如果在数据发掘过程中结合概念层次关系,就可以将元查询制导技术推广到多层关联规则的发现,从而极大地扩展了元查询制导技术的效能.为此,本文将元查询制导技术与多层关联规则发现技术结合起来,以期提高知识发现系统的能力和性能,我们称之为元模式制导的多层关联规则发现技术.

本文第 1 节引入有关元模式制导的多层关联规则发现技术的基本概念.第 2 节提出单变量多层关联规则的元模式制导发现方法及相应的算法描述.第 3 节比较上节所提两种算法的性能.第 4 节指出存在的问题及进一步研究方向.

1　基本概念

为讨论方便,我们采用关系型数据模型,不过,只要对我们在本文所提出的方法稍加修改即可应用于其它数据模型,如扩展的关系模型、面向对象的数据模型.为有效地进行数据发掘,用户常常只对存储在大型数据库中某个与特定任务相关的数据子集感兴趣.为此,提交给知识发现系统的类-SQL 的数据发掘查询需表达两个目的:① 相关数据收集;② 知识发现.前者由对数据库进行检索的 SQL 查询负责,这可在 SQL 中指定合适的条件及所涉及的各属性,从而收集与任务相关的数据子集.这一问题不是本文重点,因此略而不述,我们这里主要讨论知识发现部分的有关问题.

例 1:假定存在如下某大学数据库的部分关系模式:

student(name,sno,status,major,gpa,birth_date,birth_place)

course(cno,title,dept)

grading(sno,cno,instructor,semester,grade)

设有这样一个数据发掘查询 Q1:在出生于安徽省的学生中,寻找与 major 相关的属性 status,gpa,birth_place 和 address 之间的关系,其形式如下所示:

```
Q1:discovery rules in the form of
    major(s:student,x) ∧ Q(s,y)→R(s,z)
    from student
    where birth_place="AnHui"
    in relevance to major,status,gpa,birth_place
```

Q1 中的元模式 major(s:student,x) ∧ Q(s,y)→R(s,z)指明了待发现的规则的形式,即待发现的规则是一个前件为两个二元谓词 major(s:student,x)和 Q(s,y),后件为一个二元谓词 R(s,z)的逻辑规则.所有 3 个谓词的第 1 个变元均为关系 student 的关键字 s,Q 和 R 是两个谓词变量,可以例示为数据发掘查询中所指出的相关属性,如 sataus,gpa,birth_place 和 address.

通过数据发掘技术,我们可从数据库中发现如下规则:

major(s,"Science") ∧ gpa(s,"Excellent")→status(s,"Graduate")　(60%)　(1)

规则(1)指出,gpa 为优秀的理科专业学生中有 60%是研究生.

如果我们可以利用有关属性的概念层次关系,这些规则还可以用较低层的概念进行表达,如规则(2),其语义十分清楚,不言自明.

major(s,"Physics")∧gpa(s,"3.8–4.0")→status(s,"Graduate") (80%) (2)

实际上,通过关系链接,我们还可以发现若干不同关系之间的联系.在数据发掘查询的元模式中可以明确指出所需的关系链接,如下述数据发掘查询 Q2:

Q2:discovery rules in the form of

major(s,x)∧P(c,y)→Q(s,S,c,C,z)

from student S,course C,grading G

where S.birth_place="Hunan"

这个查询是要发现 3 个谓词之间的关系,其中一个谓词已经例示为 major(s,x),第 2 个谓词包含关系 course 的关键字,第 3 个作为后件的谓词包含 student 和 course 这两个关系的关键字,相关数据集是出生在湖南的学生.利用概念层次关系,通过数据发掘,我们也许可以发现如下规则:

major(s,"Science")∧dept(c,"CS")→grade(s,c,"Good") (60%) (3)

major(s,"Math")∧cno(c,"CS–4–level")→grade(s,c,"A-") (45%) (4)

在上例中,数据发掘查询和所发现的规则都包含了非原始概念层的概念,即表述的概念层次高于存储在数据库中的实际数据所直接体现出的概念,例如"Science","Graduate"和"Excellent"等等.出现在查询中的高层次概念有助于收集相关数据集,而分层次组织概念有助于自顶向下逐层递进地进行数据发掘,即首先试图在最高概念层上发现知识,然后逐步降低概念层次,在较低概念层上逐层进行知识发现.

本文采用概念层次关系(Concept Hierarchy)来表达概念的多级别.当然,这一概念层次关系可以动态调整或自动生成[7],以适应更灵活的数据发掘任务.并且,为讨论方便,我们假定待发现规则是合取式的,即规则的头与体均是合取式,而谓词变量只能针对有关的数据库模式(属性)进行例示,且在元模式中指出的谓词互不相同,即应例示为不同的谓词名.作为一种记法习惯,我们规定以大写字母开头的谓词名表示一个谓词变量,它可以通过将其约束为关于数据库模式的某个具体的谓词名(以小写字母开头)的方式来被例示.例如,谓词变量 $P(x,y)$ 可以在例 1 中例示为 *student*(*x*,"*Graduate*").

定义 1. 元模式是一个如下形式的规则模板:

$$P_1 \wedge P_2 \wedge \ldots \wedge P_m \rightarrow Q_1 \wedge Q_2 \wedge \ldots \wedge Q_n \quad (5)$$

其中 $P_i(i=1,2,\ldots,m)$,$P_j(j=1,2,\ldots,n)$或者是例示谓词或者是谓词变量.

定义 2. 某规则 R 与某元模式 R_m 一致,当且仅当 R 可以与 R_m 合一(Unified).

例如规则(1)与元模式 *major*(*s*;*student*,*x*)∧*Q*(*s*,*y*)→*R*(*s*,*z*)就是一致的.

定义 3. 模式 P 是一个谓词 P_i 或一组谓词 $P_i,\ldots,P_j$ 的合取式 $P_i \wedge \ldots \wedge P_j$,其中谓词 $P_i,\ldots,P_j$ 是根据数据库模式(属性)例示的谓词.

定义 4. 模式 P 在集合 S 中的支持度 *support*(P/S)为 S 中包含模式 P 的元组数与 S 中的总元组数之比;规则 $p \rightarrow q$ 在集合 S 中的信任度 *confidence*($p \rightarrow q/S$)为 *support*($p \wedge q/S$)与 *support*(p/S)之比.

为发现相对常见的模式(Frequent Pattern)和合理信任度的规则,用户或有关专家一般

会确定两个阈值:最低支持度 minsup(minimum support)和最低信任度 minconf(minimum confidence).注意,为在发现多层关联规则,可在不同概念层设定不同的最低支持度 minsup[L]和最低信任度 minconf[L],其中 L 代表概念的层次.

定义 5. 模式 P 在集合 S 的第 L 层是常见的(Frequent),如果 P 的支持度不低于相应层的最低支持度 minsup[L];规则 p→q 在集合 S 的第 L 层是高的(High),如果它的信任度不低于相应层的最低度 minconf[L].

定义 6. 规则 p→q/S 是强的(Strong),如果 p∧q/S 和 p→q/S 在集合 S 的相应层分别是常见的和高的,而且 p 和 q 中每个谓词的先辈(相应的高层谓词)在相应层均为常见谓词.

引理 1. 如果规则 p→q/S 是强的,那么在相应概念层 L 有如下结论:

(1) support(p∧q/S)≥minsup[L],support(p/S)≥minsup[L],support(q/S)≥minsup[L];

(2) confidence(p→q/S)≥minconf[L].

这就意味着,如果要检查某个较低概念层的模式,那么其先辈模式(相应的高层概念模式)必须是常见的.而我们的数据发掘是根据概念层次关系自顶向下逐层递进的,按照上述定义,恰好可以避免由非常见模式构成无意义的组合,从而意味着一个过滤过程.例如,在与 major(s,x)相关的数据集中,如果"(x=)Science"是常见模式,那么就检查其低层模式,如"Physics";否则,就不必检查其任何低层模式.

基于例 1 提出的两种数据发掘查询,多层关联规则的元模式制导发现技术可分为两类:① 发现单变量多层关联规则;② 发现多变量多层关联规则.前者是为了发现形如(1)和(2)的规则,即其中所有谓词只能含有一个相同的变量;而后者则可以发现形如(3)和(4)的规则,即某些谓词可以含有 1 个以上的变量,且往往涉及多个关系之间的链接.本文仅讨论前者,对后者将另文专述.

2　单变量多层关联规则的元模式制导发现

本节讨论单变量多层关联规则的元模式制导发现方法.单变量多层关联规则可在不同概念层表达同一数据关系的若干特性(属性)之间的联系.

定义 7. 单变量元模式形如:

$$P_1(t;rel,x_1)\wedge P_2(t;rel,x_2)\wedge\ldots\wedge P_m(t;rel,x_m)\rightarrow Q_1(t;rel,y_1)\wedge Q_2(t;rel,y_2)\wedge\ldots\wedge Q_m(t;rel,y_m) \quad (6)$$

其中 $P_i(i=1,2,\ldots,m)$,$P_j(j=1,2,\ldots,n)$或者是例示谓词或者是谓词变量,公共变量 t 为关系 *rel* 的关键字.

通过数据发掘,在所发现的规则中每个谓词变量均被例示为具体的谓词名,而这些谓词名均与关系 *rel* 的属性名相对应;公共变量 t 仍然存在,表示关系 *rel* 的关键字,谓词中的其它变量将例示为高层或原始概念层常量,即有关谓词在相应层的属性值.例如,例 1 中的 $major(s;student,x)\wedge Q(s,y)\rightarrow R(s,z)$就是一个单变量元模式,在其所发现的规则(1)中,公共变量 s 仍然保留,用于标识关系 *student* 的任一元组,每个谓词的其它变量均例示为常量,如 Science,Excellent 和 Graduate.

为发现单变量多层关联规则,在文献[6]的基础上提出两种方法:常见谓词增长技术和直接 p-谓词生成技术.

2.1 常见谓词增长技术

发现多层关联规则的关键在于寻找不同概念层上的常见谓词.基于文献[6]的多层关联规则发现算法,我们提出了单变量多层关联规则的元模式制导发现方法.下面,我们首先给出一个例子来说明该方法,然后再给出相应的算法描述.

例2:按照例1中的数据发掘查询Q1来发现可能的单变量多层关联规则.

第1,收集相关数据集,执行数据发掘查询中的SQL查询,选择出生在中国的学生,然后对相关属性集major,gpa,sataus,gpa,birth_place作投影运算,形成初始数据关系R_0,其片段如表1所示.

表1

major	gpa	status	birth_place
CS	3.85	Senior	AnHui
...	...	...	...

表2

major	count
Science	550
...	...
gpa	count
Excellent	450
...	...
birth_place	count
AnHui	550
...	...

表3

major	gpa	count
Science	Excellent	89
...	...	...
major	status	count
Science	underg	6000
...	...	...
major	birth_place	count
Science	AnHui	4000
...	...	...

表4

major	gpa	status	count
Science	Excellent	underg	526
...	...	...	...

表5

Rule	Support	Confidence
...	...	...
major(s,"Science")∧gpa(s,"Excellent")→status(s,"Graduate")	60%	88%
...	...	...
major(s,"CS")∧gpa(s,"3.8_3.9")→status(s,"Sennior")	15%	38%
...	...	...

第2,推导第1概念层的常见谓词表.考虑到数据库中的元组总数是一定的,所以我们直接以支持代替支持度,设该层的最低支持minsup[1]=350.考察数据发掘查询Q1,我们知道元模式指出待发现规则的前件有m=2个谓词,后件有n=1个谓词,共p=3(m+n=3)个谓词,其中一个是例示谓词.于是,我们只需如下导出常见3-谓词表即可:

(1) 扫描初始数据关系R_0,对元查询中指出的相关属性major,gpa,sataus,birth_place生成相应的常见1-谓词表Fre[1,1],如表2所示.

(2) 以Fre[1,1]为条件,过滤初始数据关系R_0,即将其中不与Fre[1,1]中任何一项匹配的元组删除,从而得到一个过滤了的数据关系R_1.

(3) 推导常见2-谓词表Fre[1,2].方法是,首先仿照文献[8]中的prior_gen候选生成算法,根据Fre[1,1]构造候选常见2-谓词表.注意,元模式指出的例示谓词必须包含在候选常见2-谓词中,在本例中,就是要以关于major的常见1-谓词分别与关于gpa,sataus,birth_place的常见1-谓词进行组合,构成候选常见2-谓词表C[1,2].然后,扫描数据关系R_1,计算各候选2-谓词的支持,选取不低于最低支持的2-谓词构成常见2-谓词表Fre[1,2].

(4) 推导常见3-谓词表Fre[1,3].方法是,先根据Fre[1,2]构造候选常见3-谓词表,即组合Fre[1,2]中的任何两个含有相同1-谓词的2-谓词,形成候选常见3-谓词表C[1,3].然后再扫描数据关系R_1,计算各候选3-谓词的支持,选取不低于最低支持的3-谓词构成常见

3-谓词表 Fre[1,3].

第 3,反复上述过程,推导第 2,3 概念层的常见谓词表,相应的表格这里略去.

第 4,根据元模式及相应概念层的最低信任,从 Fre[1,3],Fre[2,3]和 Fre[3,3]中生成相应层次的关联规则,如表 5 所示.

基于该例描述的常见谓词生长技术,我们提出如下单变量多层关联规则的元模式制导发现算法:

```
Algorithm 1: 元模式制导的多层关联规则发现算法;
Input: ① DB,某关系数据库;② H,某概念层次;③ minsup[L],第 L 层的最低支持度,minconf[L],第 L 层的最低信
       任度;④ metaR,形如式(6)的元模式,其中谓词数为 p;
Output: 形如元模式的多层关联规则;
Begin
  (1) Execute the SQL query specified by the data mining query. This step generates the initial data relation R0;
  (2) for (L=1; L<max_level; L++) {
  (3)   if (L==1) {
  (4)     Fre[1,1]=get_frequent_predicate(R0,L);
  (5)     R1=get_filtered_table(R0,Fre[1,1]);
  (6)     }
  (7)   else Fre[L,1] = get_frequent_predicate(R1,L);
  (8)   for (K=2; K<=p; K++) {
  (9)     C[L,K]=get_candidate_set(Fre[L,K-1]);
  (10)    for each candidate c∈C[L,K] do
  (11)      for each tuple t in R1 do
  (12)        if (t matchs c ) c.support++;
  (13)    Fre[L,K]= {c|c∈C[L,K],c.support≥minsup[L]};
  (14)    }
  (15)  Discovered_Rule_set=get_rules(Fre[L,p],metaR,minconf[L]);
  (16)  }
End.
```

2.2 直接 p-谓词生成技术

上节所述算法是文献[1]多层关联规则发现算法的一种在元模式制导条件下的自然扩展.与普通的多层关联规则发现相比,元模式制导的多层关联规则发现有一个重要特点,即在元模式中预先确定了待发现规则的形式,包括逻辑关系、前后件谓词的个数、例示谓词的个数等等.将这一启发式信息运用到上节所述的算法中,从而得到一个我们称之为直接 p-谓词生成技术的发现算法,即在算法 1 的第(8)步不是从推导常见 2-谓词表开始,一步一步地推导出常见 p-谓词表,而是取消第(8)步的循环,直接从常见 1-谓词表生成候选常见 p-谓词表,然后扫描数据关系 R1,计算各候选 p-谓词的支持,选取不低于相应概念层的最低支持的候选 p-谓词,构成常见 p-谓词表.接着,根据元模式及相应概念层的最低信任,从常见 p-谓词表中生成关联规则.本算法 2 只需在算法 1 的基础上稍加修改即可,这里不再赘述.

3 两种算法的性能比较

我们用 Visual Foxpro 在内存为 16M 的 586 微机上实现了上述两种算法,采用合成数据进行算法测试的.测试数据库含有 5 个属性,每个属性有 100 个原始值.我们将这些值组织成一个有 4 个层次的概念层次关系,即每个属性的 100 个属性值分为 20 组,每组 5 个值,第 1 层有 1 个结点,第 2 层有 5 个结点,第 2 层的每个结点在第 3 层均有 20 个结点,第 3 层的每个结点在第 4 层均有 5 片叶子,即原始属性值.我们将第 1 层作为虚拟层处理,不参与

算法过程.我们所采用的元模式形如 $P(t,x)\wedge Q(t,y)\rightarrow R(t,z)$,而各概念层的最低信任均为50%.

我们首先测试算法的扩放性.我们将测试数据库的元组数从1 000开始,逐次递增到10 000.第2、3和4层的最低支持分别为5%,2%和0.5%.两算法的扩放性能数据曲线如图1所示.可见两个算法的扩放性均较好,不过随数据库规模的增大,算法1的计算量增长更为平缓一些.

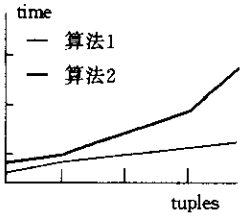

图1

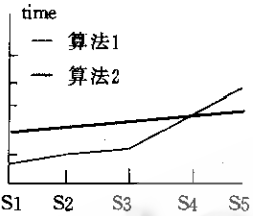

图2

接着,我们比较算法在不同最低支持下的性能变化.此时,测试数据库固定为5 000元组,分如下5次变化各层的最低支持,即S1(6%,1%,0.5%),S2(4%,1%,0.1%),S3(4%,0.5%,0.1%),S4(2%,0.5%,0.1%)和S5(2%,0.5%,0.05%).测试结果如图2所示.可以看出,当支持下降时,执行时间上升,原因是过滤条件减弱了.另外,算法1对最低支持的变化更为敏感,因为它在每次迭代中都是以最低支持为阈值来删除非常见谓词的.当最低支持较大,因而过滤条件较强时,算法1较好;反之,算法2较好.一般地,如果最低支持选择得较为合理,我们认为应首先选用算法1.

4 结论与进一步工作

本文将元查询制导技术与多层关联规则发现技术结合起来,提出了发现多层关联规则的元模式制导的自顶向下逐层递进的方法.由于元模式对系统所期望发现的规则的形式作出了限制,从而又产生了直接p-谓词生成测试技术.

本文较为详尽地讨论了单变量多层关联规则的元模式制导发现的有关问题,但未讨论多变量关联规则、交叉层次的关联规则和元模式中允许有重复谓词等问题.下一步工作就是研究这些与元模式制导的多层关联规则发现相关的问题.另外,将元模式制导发现的思想应用到序贯模式(Sequential Pattern)的发现也是一个颇有价值的研究课题.

致谢 本文工作得到了国际KDD研究知名学者加拿大Simon Fraster大学Jiawei Han教授的指导.Han教授不仅指出了研究方向,而且提供了相关的资料.特此深表感谢.

参考文献

1 欧阳为民,蔡庆生.在数据库中自动发现广义序贯模式.软件学报,1997,8(11):864～870.

2 Han J, Fu Y. Discovery of multiple-level association rules from large databases. In Proc. 21th VLDB Conf. Zurich, Swizerland, 1995.

3 Piatetsky-Shapiro G. Discovery, analysis, and presentation of strong rules. In: Piatetsky-Shapiro G, Frawlay W J eds, Knowledge Discovery in Databases, AAAI/MIT Press, 1991. 229～238.

4 Han J, Cai Y, Cercone N. Data-driven discovery of quantitative rules in relational databases. IEEE Trans. Knowledge and Data Engineering, 1993,(5):29～40.

5 Han J, Fu Y, Wang W *et al*. DBMiner: a system for mining knowledge in large relational databases. In: Proc.

1996 Int'l Conf. on Data Mining and Knowledge Discovery (KDD'96), Portland, Oregon, August 1996. 250～255.

6 Shen W, Ong K, Mitbander B *et al*. Meta-queries for data mining. In: Fayyad U M, Piatetsky-Shapiro G, Smyth P *et al* eds, Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, 1995.

7 Han J, Fu Y. Dynamic generation and refinement of concept hierachies for knowledge discovery in databases. In: AAAI'94 Workshop on Knowledge Discovery in Databases, Seattle, WA, July 1994. 157～168.

8 Agrawal R, Srikant R. Fast algorithm for mining association rules. In: Proc. 1994 Int. Conf. Very Large Data Bases, Santiago, Chile, Sept. 1994. 487～499.

META-PATTERN GUIDED DISCOVERY OF MULTIPLE-LEVEL ASSOCIATION RULE IN LARGE DATABASES

*#OU-YANG Weimin　#CAI Qingsheng

*(*Computing Center　Anhui University　Hefei* 230039)

#(*Department of Computer Science　University of Science and Technology of China　Hefei* 230027)

E-mail:oywm@mars.ahu.edu.cn

Abstract　In this paper, the authors put forward a meta-pattern guided method for discovering multiple-level association rules with a top-down and progressively deepening style by integration of pattern-guided technique and multiple-level association rules mining technique. A meta pattern is a rule template which predefine the form of the rules to be discovered, such a rule template can be used as a guidance in the data mining process.

Key words　Knowledge discovery, meta pattern, association rule, concept hierarchy.

Class number　TP301