一种新的面向属性归纳中概念层次技术研究①

孙华梅[1]，郭茂祖[2]，焦　杰[2]，黄梯云[1]
(1. 哈尔滨工业大学管理学院，哈尔滨 150001；
2. 哈尔滨工业大学计算机科学与技术学院，哈尔滨 150001)

摘要：面向属性归纳的方法目前是数据挖掘主要技术之一. 在分析基本的面向属性归纳算法不足的基础上，提出了一种概念层次优化技术，包括：将基于规则的概念图转化成一棵概念树；对于不平衡的概念树，再转化成平衡的概念树；用节点集合来记录数据库中每个元组在概念层次中的泛化路径. 利用置信度、支持度和 LS 充分性因子等评价指标对学习结果进行取舍. 最后，将算法在爱尔兰教育经历数据库分析中进行了应用测试，结果显示了算法更加有效，适用性更强.

关键词：数据挖掘；概念描述；基于规则的面向属性归纳；概念层次
中图分类号：C931.6　　**文献标识码**：A　　**文章编号**：1007 - 9807(2004)01 - 0065 - 08

0　引　言

数据挖掘是从大量的数据中提取一些事前未知的、对任务有用的信息的过程[1~3]. 过去的 20 年里，在工业、商业等领域的关系数据库中收集和管理着大量的数据. 数据库的大小和数量之多，已经远远超过了人类通过现有技术所能分析和管理的范围. 这就产生了一个新的需求，即从海量的数据库中提取有用的信息，或称之为基于知识的技术. 许多大公司已经开始在某些领域采用一些数据挖掘的工具，他们使用数据仓库技术，建立一个完整的系统来支持决策管理和商业计划. 数据挖掘涉及数据库系统、统计学、机器学习等学科，主要研究从大量数据中提取或“挖掘”知识，可用于计算机科学的数据库中知识发现以及管理科学的智能决策支持系统，包括描述式数据挖掘和预测式数据挖掘. 概念描述是描述式数据挖掘的最基本形式，它以简洁汇总的形式描述给定的任务相关数据集. 概念描述由概念特征化和比较组成，概念特征化汇总并描述称作目标类的数据集，它有两种一般方法：基于数据立方体 OLAP 的方法和面向属性归纳的方法(attribute-oriented induction，AOI)[1,4~7]，二者都是基于属性或维的概化方法. 面向属性的归纳使用概念分层，通过以高层概念替换低层数据概化训练数据，目前是数据挖掘主要技术之一.

在早期的研究中，基本的面向属性的归纳方法有三种基本算法，分别是 AOI(attribute-oriented induction)，LCHR(learn characteristic rule)[8]，GDBR(generalize database relation)[9]. 其中，AOI 的算法具有更好的时间和空间复杂性[10]. AOI 算法是从关系数据库的知识发现过程中发展起来的，它可以从关系数据库中发现不同类型的知识，并且根据特征化或者区分的任务将其表达为特征化规则或者区分规则.

面向属性归纳(AOI)的一个重要特点是它可以把关系数据库中的属性值泛化成较高层次的概念，然后合并那些在泛化后变成相同的记录，压缩数据空间[1,4~6]. 结果是每一个这样泛化后得到的记录(元组)都表示了在初始关系数据库中那些被

① 收稿日期：2002 - 09 - 25；修订日期：2003 - 04 - 01.
基金项目：国家自然科学基金资助项目(70071008)；黑龙江省博士后科研启动基金资助项目；哈尔滨工业大学校基金资助项目(HIT.2002.78)；国家“863”计划资助项目(2003AA118030).
作者简介：孙华梅(1972—)，女，吉林长春人，副教授.

泛化了的元组的特征[1,4,5,7].

面向属性归纳的方法中进行概念泛化的技术称为概念层次技术[1,4,10]. 通过概念的上升和下降表示属性的泛化和特化. 早期的概念泛化的结构是用概念树或者方格表示泛化的背景知识,但是概念树和方格在表达背景知识的时候都有很大的局限性[2,4,10],后来,研究者采用更一般的概念层次图来代替[5,9,11]. 但是,无论概念树还是一般的概念层次图,都只能表示单一属性的概念层次,每个属性只能在自身的基础上由底层向高层泛化,不能结合任务的背景知识. 本文使用一种基于规则的概念层次图来表示背景知识,并且将其转化为更加易于操作和记录规则的概念树.

在基于规则的概念图中,每一个概念都可以泛化到更高的层次,规则可以判断概念图中节点表示的属性应该经过哪一条路径泛化到什么层次中,并且基于规则的概念图要求是无循环的线性概念图. 通过将非线性的概念图转化为线性的概念树,为每个节点确定唯一的父节点,并从中得到相应的规则.

对于泛化结果的评价和表示,采用规则的支持度和可信度的衡量,并且对于得到的关系表所表示的规则,采用不确定性推理中的主观 Bayes 方法的 LS 充分性因子对学习结果进行评价,从而得出最简单的、最可信和具有最大兴趣度的特征规则[12,13]. 本文首先介绍基本的面向属性的归纳方法和一般的概念层次图,然后将其转化为基于规则的面向属性的归纳方法和用条件概念层次树表示的规则. 通过对初始关系表泛化,得到主关系表,并从中产生规则. 使用可信度和支持度阈值以及 LS 充分性因子来评价和控制规则,用爱尔兰教育经历数据库作为例子说明.

1 基本的面向属性的归纳

基本的面向属性的归纳(basic attribute-oriented induction)是从关系数据库中发现规则的一种方法. 主要的技术是使用概念泛化的方法. 在爱尔兰教育经历数据库(Irish Educational Transitions Data)中学生的关系表示为:

Student(name, sex, DVRT, educational level attained, leaving certificate, prestige score for father's occupation, type of school),即:学生(姓名,性别,测试分数,教育程度,是否结业,父亲职业的期望分数,学校类型);DVRT 表示 drumcondra verbal reasoning test score.

其中的属性,比如 sex, DVRT, restige score for father's occupation, type of school 等都可以根据给定的背景知识进行泛化. 例如属性 DVRT ∈[0, 200],将 DVRT 小于 80 的属性值泛化为 poor,将 DVRT 值在 80 和 100 之间泛化为 average,在原有的关系表中就会出现许多相同的记录,将其合并,并增加一个新的列 count,用来表示合并后的记录条数.

在基本的面向属性的归纳中,一个学习任务有三个重要的元素:和任务相关的初始数据,用概念层次描述的背景知识和学习结果的表达方式及评价.

1) 和任务相关的初始数据直接来自原始关系数据库,但是,对于海量的数据和属性要进行筛选,找出和任务相关的属性和数据,这样就得到了和任务相关的数据,称为初始关系表(initial relation).

2) 背景知识在面向属性的归纳中表示为概念层次图,简单的概念层次图是图 1 所示的概念树,其中叶子节点是初始关系表中的属性值,其父节点就是这个属性较高层的概念,这样对于每一个属性值,只有唯一的一个父节点,也就是泛化后的较高层概念.

例如:属性 DVRT

根据图 1 给定的概念层次树,产生相应的规则

{0～80} →poor

{81～100} →average

{101～120} →good

{121～200} →excellent

{poor, average} →weak

{good, excellent} →strong

{weak, strong} →any

这些规则是对属性进行泛化的规则,应用相应的规则,对相应的属性进行泛化,然后合并相同的记录数得到一个关系表,称为主关系表(prime relation).

3) 学习结果的表达. 最后得到的概化元组,也就是主关系表,是用来产生学习结果的. 通过将关系表转化为逻辑关系式来产生规则.

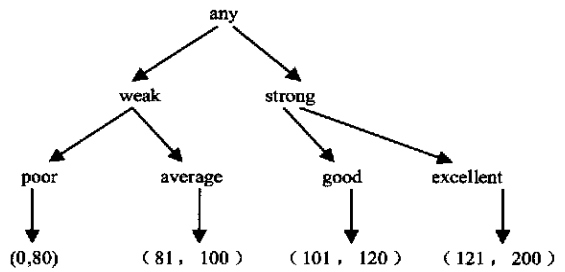

图 1　DVRT 的概念层次树

2　量化规则和 LS 充分性因子的定义以及规则的保留准则

定义 1　带有量化信息的逻辑规则称为量化规则.

结合了量化特征规则 t. 权和量化区分规则 d. 权的量化规则描述,如下形式

$$(\forall x)\ \text{target\_class} \Leftarrow \text{condition1}(x)[t:w_1, d:w_2] \vee \cdots \vee \text{condition2}(x)[t:w_3, d:w_4]$$

定义 2　规则的置信度

$$\text{confidence}(A \Rightarrow B) = P(B|A) = \frac{\text{Support-count}(A \cup B)}{\text{Support-count}(A)}$$

最小置信度阈值 min. conf,由领域专家给定,在学习结果的规则中不满足最小置信度阈值的规则将被删除.

规则的支持度 $\text{support}(A \Rightarrow B) = P(A \cup B)$

最小支持度阈值 min. sup,由领域专家给定,在学习结果的规则中不满足最小支持度阈值的规则将被删除.

定义 3　设 E 是规则的条件, H 是规则的结果, $P(H/E)$ 为规则的置信度, $P(H)$ 是目标类的先验概率,则充分性因子 LS 定义为

$$LS = \frac{P(H|E)^{*} - (1 - P(H))}{P(H) - (1 - P(H|E))}$$

其中, $LS \in [0, +\infty]$, LS 值也由领域专家给出.

根据定义,只有当一个规则的置信度大于最小置信度阈值,并且支持度大于最小支持度阈值,并且 LS 大于 1 的规则才可以保留,否则就视为对其相关规则或对结果引起误导的规则,而从学习结果规则中删除. 最后得到的学习规则的条数要满足一个由领域专家给定的规则控制阈值,否则可通过修改最小置信度阈值或者最小支持度阈值,进一步减少规则的条数.

3　基于规则的面向属性归纳

基本的面向属性的归纳方法可以快速地从大量数据中产生特征和对比的规则,但是却不能体现关系数据库中属性之间的关联关系,为了更好的挖掘关系数据库中潜在的有用的知识,提出了基于规则的面向属性的归纳方法.

首先定义一个基本的基于规则的面向属性泛化的模型. 一个基于规则的 AOI 系统由 5 部分组成 ——DB, CH, DS, KR, Ta,其中: DB 是大型的关系数据库,就是要进行挖掘和学习的原始数据库; CH 是基于规则的概念层次,在基本的概念层次图上结合条件规则加以改进; DS 是支持概念层次的推理系统,包括 CH 以及其它的背景知识规则,本文中的 DS 就是 CH; KR 是学习结果的知识表达方式,是结合了量化特征规则和量化区分规则的量化规则描述; Ta 是一组由领域专家给定的控制阈值,包括最小置信度阈值、最小支持度阈值、期望的属性删除控制阈值、规则数控制阈值.

3.1　基于规则的概念层次

在基本的面向属性的归纳方法中,进行属性泛化的主要工具是概念层次树. 这种无条件的概念泛化,只能对某一个属性在自身属性上进行不同层次的泛化,而不能结合关系数据库中属性之间的关联关系进行条件的属性概念层次的泛化. 为此,将使用一种基于规则的概念层次图,在这个概念层次图中,一个概念可以沿不同的路径上升到指定的泛化层次.

基于规则的概念层次是指概念图中路径和规则的概念层次,也就是说概念图中的路径不能简单地从一个节点到唯一的另一个节点,而是通过条件的规则判定从一个节点经过哪一条路径到达更高层次的概念节点. 一个概念层次图可以是平衡的或者不平衡的,平衡的概念层次是指每一个叶节点的深度都相同. 对于不平衡的概念层次,可以将其转化为平衡的概念层次,可采用复制节点法完成. 在基本的面向属性的泛化中,要求概念图是非循环的、平衡的.

非循环的概念层次是指在概念图中不存在任何的环,否则将无法区分高层与底层概念,是无效

的概念层次. 图 2 示上例 DVRT 的条件概念层次为

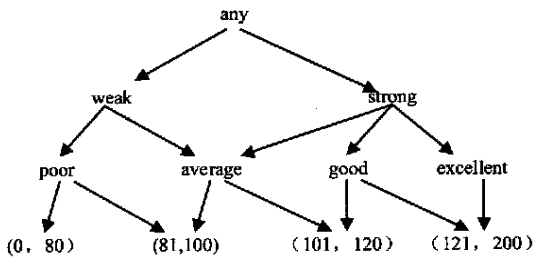

图 2 DVRT 的条件概念层次

相应的条件泛化规则为

R1 :{0～80} →poor

R2 :{81～100}{ leaving certificate = 2} →average

R3 :{81～100}{ leaving certificate = 1} →poor

R4 :{101～120}{ leaving certificate = 2} →good

R5 :{101～120}{ leaving certificate = 1} →average

R6 :{121～200}{ leaving certificate = 2} →excellent

R7 :{121～200}{ leaving certificate = 1} →good

R8 :{poor} →weak

R9 :{average} ∧{poor , junior cycle incomplete } →weak

R10 :{average} ∧{ junior cycle terminal leaver , senior cycle , 3rd level } →strong

R11 :{good} →strong

R12 :{excellent} →strong

R13 :{weak} →any ,

R14 :{strong} →any

比较图 1 中产生的规则和上述给定的条件规则,可以看出,条件泛化的概念层次更加清晰合理,并且更好地体现了关系数据库中的规则之间的关联.

3.2 将非循环的概念图转化成概念树

对于一个不平衡的概念树,为了便于归纳算法的实现,需要将它转化为平衡的概念树,本文采用复制顶节点的方法平衡概念树. 算法的原理是,从顶层开始,对于树中的每一个节点,如果一个节点的子树深度不相等,则复制该节点,并将复制节点插入深度小的子树树根,直到所有节点的子树都平衡. 算法描述如下:

depth(x) 表示 x 节点在概念树中的深度,height(CH) 表示概念树 CH 的深度.

```
FOR depth = 2 TO (height(CH)) DO
  FOR CH 中的每个节点 u DO
    IF 有 E(v1 ,u) 和 E(v2 ,u) 并且 depth(v1) > depth(v2) THEN
      复制节点 u1 = u;
      删除路径 E(v2 ,u);
      建立新路径 E(u1 ,u) 和 E(v2 ,u1);
    ENDIF
  ENDFOR
ENDFOR
```

时间复杂度 O(np) ,其中:n 是概念树中的节点个数;p 是概念树的层次深度.

给定一个非循环的基于规则的概念图,采用复制条件节点的方法将其转换成概念树. 算法的原理是:从最底层的叶节点开始,对于每一个有一个以上父节点的节点进行复制,并分别为每个父节点连接一个节点. 算法如下:

```
FOR depth = 2 TO (height(CH)) do
  For CH 中每个节点 u DO
    IF 有 E(u ,v1) 和 E(u ,v2) THEN
      复制节点 u1 = u;
      删除 E(u ,v2);
      建立新路径 E(u1 ,v2);
      将 u 的子树复制到节点 u1;
    ENDIF
  ENDFOR
ENDFOR
```

时间复杂度 O(np) ,其中 ,n、p 同上.

得到这个概念树后,可以方便地应用条件规则进行属性的概化. 在基于规则的面向属性的泛化过程中,为了记录每一个规则在其概念树中的节点位置,还要为转化后的概念树进行节点的修改,将规则记录其中,其方法是:先为每一非根节点建立一个属性集,初始集合是节点本身,然后遍历概念树. 对于每一个路径上的每一条边,如果边上的父节点属性集中包含除本身之外的属性,则先将这些属性复制到子节点的属性集中;如果这条边对应的规则中还包括了除了该节点现有属性值之外的属性,则将其插入到该节点的属性集中. 这样,在泛化的过程中,可以将每个规则和路径记录在相应层次的节点集合中. 算法如下:

```
FOR 从根节点到叶节点的每一条路径 DO
  FOR 路径上的每一条边 E(u ,v) DO
    IF 节点 v 的属性集合中包含了除本身之外的属性 P1 , … ,Pn THEN
      将 P1 , … ,Pn 复制到属性 u 的属性集中;
    ENDIF
    IF 表示该边的规则的前件中包括除了 u 之外的其他属性 THEN
      将所有这些属性加入到节点 u 的属性集合中;
    ENDIF
  ENDFOR
  FOR 叶节点链表中的每一个属性 Pi DO
    IF Pi 不是初始关系表中的属性 then
```

将属性 P_i 根据它的概念层次向下特化到初始关系值;

用这个值代替 P_i;

ENDIF

ENDFOR

ENDFOR

时间复杂度 $O(n^2)$,其中 ,n 是节点个数.

通过转换 ,原来的基于规则的概念层次图变成图 3 所示的概念树.

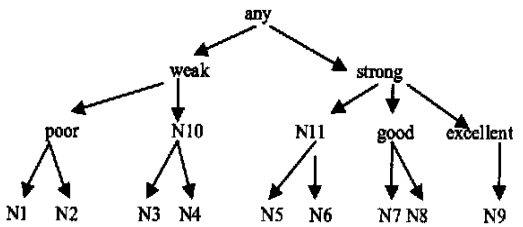

图 3　转换后的基于规则的概念树

其中:每个节点的属性集合为:

N1 :DVRT < = 80

N2 :{81～100} ∧{ leaving certificate = 1}

N3 :{81～100} ∧{ leaving certificate = 2} ∧{poor ,junior cycle incomplete }

N4 :{101～120} ∧{ leaving certificate = 1} ∧{poor ,junior cycle incomplete }

N5 :{81～100} ∧{ leaving certificate = 2} ∧{junior cycle terminal leaver , senior cycle , 3rd level }

N6 :{101～120} ∧{ leaving certificate = 1} ∧{junior cycle terminal leaver , senior cycle , 3rd level }

N7 :{120～200} ∧{ leaving certificate = 1}

N8 :{101～120} ∧{ leaving certificate = 2}

N9 :{120～200} ∧{ leaving certificate = 2}

N10 :{average} ∧{poor ,junior cycle incomplete }

N11 :{average} ∧{ junior cycle terminal leaver , senior cycle , 3rd level }

显而易见 ,N3 是 N5 的复制节点 ,其自身表示的规则是相同的 ,但是两个节点的属性组内容却不相同 ,所以相应的泛化路径不同. 这样 ,由于每个属性的属性组内容是唯一的 ,所以可以记录每一条应用在这个路径上所有节点的规则 ,同时记录了每个属性的泛化路径 ,可以很容易地进行泛化 ,而当一个属性被发现是过度泛化的时候 ,还可以很容易地找到路径回溯 ,恢复进行泛化前的属性节点.

如果在泛化过程中不能记录泛化的层次 ,也就是关系表中的值在概念树中的节点层次 ,还可以将非叶节点的属性组的每一个属性根据其相应的概念层次特化到叶节点层 ,将其内容加入到相关的属性概念树的叶节点的属性组中 ,这样也可以对每一条规则都在叶节点层次找到并记录它的相应泛化路径. 本文使用的是记录泛化层次 ,也就是相应每个属性的概念树的层次方法来记录每个属性的泛化路径.

4　实例分析

采用爱尔兰教育经历数据库来测试结果 ,任务是从中学习已拿到毕业证和未拿到毕业证的学生的特征 ,并进行量化比较.

原始表中有 7 个属性 ,分别是:

①姓名

②性别 1 = 男 , 2 = 女 ,

③DVRT:学生的测试成绩.

④educational level attained 达到的教育程度:

1 = Primary terminal leaver 小学毕业

2 = junior cycle incomplete: vocational school 职业学校一年级未毕业

3 = junior cycle incomplete: secondary school 中学一年级未毕业

4 = junior cycle terminal leaver: vocational school 职业学校一年级毕业

5 = junior cycle terminal leaver: secondary school 中学一年级毕业

6 = senior cycle incomplete: vocational school 职业学校二年级未毕业

7 = senior cycle incomplete: secondary school 中学二年级未毕业

8 = senior cycle terminal leaver: vocational school 职业学校二年级毕业

9 = senior cycle terminal leaver: secondary school 中学二年级毕业

10 = 3rd levels incomplete 三年级未毕业

11 = 3rd levels complete 三年级毕业

⑤Leaving Certificate 毕业证. 1 未取得毕业证; 2 取得毕业证.

⑥Prestige score for father 's occupation 父亲职业的评分

⑦学校类型:

1 = secondary: 中学

2 = vocational :职业学校

9 = primary terminal leaver. 小学

基于规则的 AOI:

输入:

DB: 爱尔兰教育经历数据库

CH,DS: 如下

属性 name 是数据库中的键值,没有相应的概念层次.

属性 sex 的概念层次见图 4.

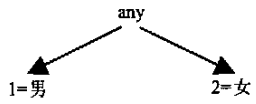

图 4 性别的概念层次

属性 DVRT 的概念层次见图 3.

属性教育程度 (educational level attained) 的概念层次见图 5.

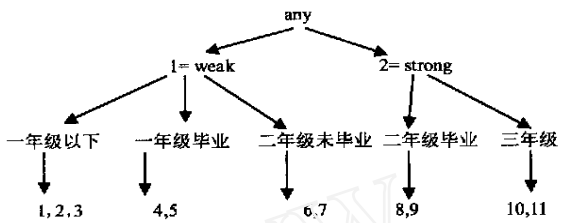

图 5 教育程度的概念层次

属性结业证明书没有概念层次.

属性父亲职评分的概念层次见图 6.

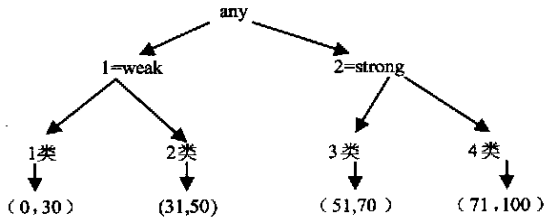

图 6 父亲职业评分的概念层次

属性学校类型的概念层次见图 7.

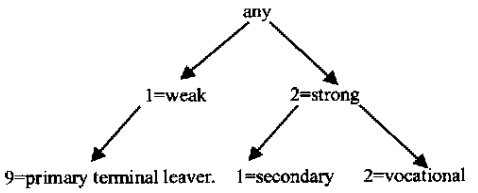

图 7 学校类型的概念层次

KR: 量化规则的表示.

Ta: min_sup = 0.1, min_conf = 0.5, LS > 1, 属性泛化控制阈值是 2, 规则控制阈值为 4;

算法步骤:

步骤 1 从原始数据表中产生和任务相关的数据表见表 1.

表 1 原始关系表

姓名	性别	DVRT	教育程度	毕业证	父亲职业期望成绩	学校类型
NAME1	1	113	3	1	28	1
NAME2	1	101	1	1	28	9
NAME3	1	110	9	2	69	1
NAME4	1	121	5	1	57	1
NAME6	1	82	4	1	18	2
NAME7	1	85	4	1	28	2
NAME8	1	84	1	1	28	9
NAME9	1	98	2	1	43	2
NAME10	1	92	4	1	33	2
NAME11	1	90	1	1	18	9
NAME12	1	88	4	1	28	2
NAME13	1	84	4	1	18	2
NAME14	1	114	11	2	18	1
NAME15	1	70	4	1	57	2
NAME16	1	109	9	2	40	1
…	…	…	…	…	…	…

搜索整个数据表,根据给定的属性删除阈值,这里是 2,可知属性姓名有超过阈值个数的不同属性值,并且没有相应的概念泛化层次,则将其删除. 其它的属性均保留,生成初始关系表如表 2 所示.

表 2　初始关系表

性别	DVRT	教育程度	毕业证	父亲职业评分	学校类型
1	113	3	1	28	1
1	101	1	1	28	9
1	110	9	2	69	1
1	121	5	1	57	1
1	82	4	1	18	2
1	85	4	1	28	2
1	84	1	1	28	9
1	98	2	1	43	2
1	92	4	1	33	2
1	90	1	1	18	9
1	88	4	1	28	2
1	84	4	1	18	2
1	114	11	2	18	1
…	…	…	…	…	…

步骤 2　对属性进行泛化,得到主关系表

对属性进行重复泛化,合并相同的元组,添加一列 count,计算每条记录的个数. 然后,判断每个属性中不同值的个数是否满足属性泛化控制阈值,即每个属性中不同值的个数 ≤2,满足则停止泛化,否则继续泛化. 最后得到的主关系表如表 3 示.

表 3　主关系表

DVRT	教育程度	毕业证	父亲职业评分	学校类型	总数
1	1	1	1	1	30
1	1	1	2	1	5
1	1	1	1	2	192
1	1	1	2	2	34
1	2	1	1	2	6
1	2	1	2	2	2
1	1	2	1	1	1
1	1	2	1	2	6
1	2	2	1	2	86
1	2	2	2	2	35
2	2	2	1	2	53
2	2	2	2	2	35

步骤 3　对规则进行评价和简化

为每条记录计算相应的 sup, conf, LS,删除不满足 min_sup 和 min_conf 的规则,删除 LS ≤1 的记录,最后得到的规则关系表(表 4).

表 4　规则关系表

DVRT	教育程度	毕业证	父亲职业评分	学校类型	总数	支持度	可信度	充分性因子
1	1	1	1	2	192	0.395 9	0.969 697	25.695 19
1	2	2	1	2	86	0.177 3	0.934 782 6	17.850 31
2	2	2	1	2	53	0.109 3	1	∞

步骤 4　规则为

得到毕业证的学生 →

{(dvrt = 1) ∧(教育程度 = 2) ∧ (父亲职业评分 = 1) ∧ (学校类型 = 2)} [sup: 0.177 3, conf: 0.934 782 6, ls = 17.850 31] ∨

{(dvrt = 2) ∧ (教育程度 = 2) ∧ (父亲职业评分 = 1) ∧(学校类型 = 2)} [sup: 0.109 3, conf :1]

未得到毕业证的学生 →

{(dvrt = 1) ∧ (教育程度 = 1) ∧ (父亲职业评分 = 1) ∧(学校类型 = 2)} [sup: 0.395 9, conf: 0.969 697, ls = 25.695 19]

5　结论

基于规则的 AOI 算法解决了原有 AOI 算法中不能结合条件规则的背景知识的缺点,通过基于规则的概念图表示属性的条件泛化过程,并且针对基于规则的概念层次进行泛化和记录泛化路径的问题,提出了三个解决算法,分别将基于规则的概念层次转化为概念树,将不平衡的概念树转化为平衡的概念树和记录概念树中属性泛化路径. 并且 AOI 算法的复杂度不增加,保证了在面向属性归纳算法中 AOI 算法的优势.

参 考 文 献:

[1]Cheng D, Hwang H Y, Han J W. Efficient rule-based attribute-oriented induction for data mining[J]. Journal of Intelligent Information Systems, 2000, 15: 175—200.

[2]Michalski R S, Kaufman K A. Data mining and knowledge discovery: A review of issues and a multistrategy approach[A]. In: Michalski R S, Bratko I, Kubat M, eds. Machine Learning and Data Mining: Methods and Applications[M]. London: John Wiley & Sons, 1998.

[3]黄梯云. 智能决策支持系统[M]. 北京:电子工业出版社,2001.

[4]Cai Y, Cercone N, Han J. An Attribute-oriented approach for learning classification rules from relational databases[A]. Los Angeles, CA: Proceedings of 6th Int Conf on Data Engineering(ICDE'90), 1990.

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

[6]Han J, Fu Y. Exploration of the power of attribute-oriented induction in data mining[A]. In: Fayyad U, Shapiro G P, Smyth P, *et al*, ed. Advances in Knowledge Discover and Data Mining[M]. AAAI/MIT Press, 1996. 399—421.

[7]Kamber M, Winstone L, Gong W, *et al*. Generalization and decision tree induction: Efficient classification in data mining[A]. Proceeding of 1997 Int Workshop on Research Issues in Data Engineering (RIDE'97)[C]. Birmingham, England: 1997.

[8]Cai Y, Cercone N, Han J. Learning characteristic rules from relational databases[A]. Proceedings of International Symposium of Computational Intelligence'89[C]. Milano, Italy: 1989.

[9]Carter C L, Hamilton H J. A fast, on-line generalization algorithm for knowledge discovery[J]. Applied Math Letters, 1995, 8(2): 5—11.

[10]Carter C L, Hamilton H J. Performance Evaluation of Attribute-Oriented Algorithms for Knowledge Discovery from Database[R]. Dept. of Computer Science, University of Regina, SK, Canada: 1995. 486—489.

[11]Sun J P, Bi W Y. Directed Acyclic Concept Graph Based Attribute Oriented Induction[R]. Graduate School of Computer and Information Sciences, Nova Southeastern University, Fort Lauderdale, FL: 2001. 32—45.

[12]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.

[13]Zhou X, Sha C F, Zhu Y Y, *et al*. Interest measure-another threshold in association rules[J]. Journal of Computer Research and Development, 2000, 37(5): 627—633.

New concept hierarchy optimization method in attribute-oriented induction

SUN Hua-mei[1], *GUO Mao-zu*[2], *JIAO Jie*[2], *HUANG Ti-yun*[1]

1. School of Management, Harbin Institute of Technology, Harbin 150001, China;

2. School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China

Abstract: Attribute-oriented induction (AOI) approach is becoming one of the main techniques in data mining. In this paper, the basic AOI algorithm is described and its shortage is given. Then a new concept hierarchy optimization method is presented. The rule-based concept graph is covered into concept tree; the unbalanced tree; is then covered into a balanced tree if concept tree is an unbalanced one; the induction path of each node in the concept tree is recorded as node aggregate. The learning results are evaluated with support, confidence and LS sufficient factor. The algorithm is applied to Irish Educational Transitions Data, and results show that it is more effective and more suitable.

Key words: data mining; concept description; rule-based attribute-oriented induction (AOI); concept hierarchy