维普资讯 http://www.cqvip.com

第3卷第3期 管理科学学报 Vol. 3 No. 3
2000年9月 JOURNAL OF MANAGEMENT SCIENCES IN CHINA Sep., 2000

查询驱动的一种挖掘特征规则的新方法①

程 岩，黄梯云
（哈尔滨工业大学管理学院，哈尔宾 150001）

摘要：智能查询是决策支持系统的一个重要内容，在数据库中挖掘特征规则是实现智能查询的一个重要手段. 目前的数据挖掘方法都是面向具体问题的，不能适应用户灵活的查询需要. 本文结合粗糙集理论，提出了一个面向查询的挖掘特征规则的新方法，利用该方法挖掘出的特征规则可以在精度上达到最优. 以该算法为核心，本文设计出一个查询特征规则的类SQL命令，使用户与系统的交互问答更加方便、灵活.

关键词：数据挖掘；决策支持系统；特征规则；粗糙集
中图分类号：TP311 **文献标识码**：A **文章编号**：1007-9807(2000)03-0052-08

0 引 言

信息系统的查询功能提供给用户的往往是大量的原始数据，但数据本身并不是信息，隐藏在数据中的规律、规则才是对决策有帮助的信息. 因此，在数据中发现这些规律、规则的数据挖掘技术是实现智能决策支持系统的一个重要手段[1].

特征规则是数据挖掘的一个重要内容，所谓特征规则就是一些描述规则，这些规则概括了数据集中不同概念的特征，如果规则中还包括定量信息，称这些规则为定量特征规则. 特征规则表示为 $Y \rightarrow X_1$ or X_2 or…or X_β，规则中 Y 是对某概念的定义，β 是指将符合该概念定义的所有数据划分为 β 类，X_i 是对概念中第 i 类数据特征的描述[2]. 如例1就是从某汽车市场交易数据库（transaction. dbf）中挖掘出的masda626汽车买主的特征规则.

例1 buy _ car ="masda626" →
sex ="male" and salary ="$30 000—$45 000" and own _ of _ house ="yes" and
married ="yes" [60%],[45%]
or sex ="male" and salary ="$60 000—$75 000" and own _ of _ house ="yes" and
married ="yes" [25%],[75%]
or sex ="female" and salary ="$45 000—$60 000" and own _ of _ house ="no" and
married ="no" [15%],[65%]

该特征规则表示如果将masda626汽车的买主划分为3类，60% 的masda626汽车被年薪在30,000—45,000美元、拥有住房的已婚男士购买，但在本汽车市场中只有45% 的这类用户购买了masda626汽车. 式中[60%],[45%]是对规则的定量描述，从这一特征描述中，决策者可以判断出：这类买主是masda626汽车的主要买主，且这类用户群还有很大的开发潜力；而规则中描述的第2,3类用户群，既不是主要用户也没有很大的开发潜力. 利用特征规则，决策者便可以作出一个正确的销售和广告宣传决策.

决策者对特征规则的查询往往是随机多变

① 收稿日期：1999-06-28；修订日期：2000-01-12.
作者简介：程 岩(1976-)，男，辽宁沈阳人，博士生.

维普资讯 http://www.cqvip.com

的,如例 1 中,决策者可能向系统提出以下一系列问题:"masda626 汽车买主的特征是什么?","masda626 汽车或 toyoto _ paseo 汽车买主的特征是什么?","美国公民中 masda626 汽车买主的特征是什么?".可见,特征规则的挖掘技术必须满足决策者灵活的查询需要.目前的数据挖掘技术往往是针对具体问题设计的,在挖掘特征规则时常常采用示例学习的手段,即数据被汇聚到两个集合(正例集和反例集)中,通过对正例集数据进行的概括过程及排除反例集数据的特殊化过程归纳出特征规则[3].但这种方法往往不适合用户灵活的查询需求,这是因为:用户对概念的定义往往不能将数据集截然划分为正例集和反例集,符合正例集描述的数据也可能出现在反例集中,因此无法进行排除反例的特殊化过程.

本文结合粗糙集理论和面向属性的归纳学习理论来解决上述问题.面向属性的归纳算法可以只通过对正例集的概括过程发现规则,不需要排除反例的特殊化过程[4,5],因此能够满足用户灵活的查询需要.但该算法中没有测定规则精确度的指标,这就使挖掘出的特征规则在精确度上不是最优的,也就是说不能保证规则对概念的特征进行最有效的描述.所谓概括精确度就是特征规则尽量多的概括正例数据,同时尽量少的包含反例数据,从而描述出一个概念不同于其它概念的特征,它是评价特征规则质量的一个重要标准.

本文提出了一个测量数据集粗糙度的方法,以此作为评价规则精确度的标准,并在此基础上,提出了一个挖掘特征规则的新算法,该算法可以使挖掘出的规则对概念的特征进行最有效的描述.以该算法为核心,本文设计出一个查询特征规则的类 SQL 命令,使用户与系统的交互问答更加方便、灵活.

1　面向属性的归纳学习的基本原理

1.1　数据集

关系数据库中存储的是结构化的数据,如果我们将数据库中的每一记录看作一个数据实体,其字段可视为描述该实体的特征变量或属性,每一数据实体便是每一属性取值的逻辑合取范式.数据实体的属性之间往往存在决定和被决定的关系,令 C 是非空决定属性的集合,称为条件属性集;D 为非空被决定属性的集合,称为决策属性集,且要求 $C \cap D = \varnothing$.概念是通过决策属性的取值定义的,而概念的特征是通过条件属性的取值描述的.例如例 1 中,buy _ car = "masda626" 定义了一个概念:"masda626 汽车买主",该概念通过 sex, salary 等条件属性的取值来进行特征描述.我们将数据库中所有符合概念定义的数据记录汇聚到一个数据集中,便形成概念的正例集,记为 POS.例 1 中所有 buy _ car = "masda626" 的数据记录便构成了概念"masda626 的买主"的正例集,所有其它数据构成概念的反例集.面向属性的归纳算法就是通过对正例集数据的概括挖掘出概念的特征规则.

1.2　属性概念树

在数据库中,许多属性都可以进行数据归类,以形成概念汇聚点.各属性值和概念依据抽象程度不同可构成一个层次结构,通常称为概念树.概念树上将各个层次的概念按从一般到特殊的顺序排列,最一般的概念是没有具体特性的概念,用 Any 表示;最特殊的概念(叶节点)对应数据库中具体的属性值;而处于概念树层次结构中间的概念是对该属性值归纳过程中产生的更宏观的(更广义的)概念.图 1 是反映数据库中"地理位置"属性的概念树.

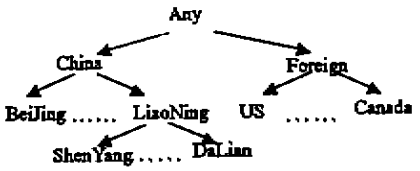

图 1　属性"地理位置"的概念树

属性概念树的形成有三种方式:自动生成、半自动生成和人工定义.在数值型属性的概念树上,每一个节点表示一个数值区间,可以根据数据的分布特征采用统计方法自动生成数值型属性的概念树[6].例如,某班的数学成绩(score)大部分集中在 60 — 85 分之间,可以首先将 score 的数据划分为:0 — 60,60 — 85,85 — 100 三个区间,再对 60 — 85 间的数据进一步详细划分,直到形成一个完整的属性概念树.某些字符型属性可以利用数据库的结构自动形成概念树,例如数据库的结构为(country,province,city…),这种结构便体现了

维普资讯 http://www.cqvip.com

属性city的概念层次，通过计算可以很容易形成city的概念树；如果经过统计分析发现属性country的值大多数为"china"，这时可以人为定义一个概念层次"foreign"，然后再通过算法自动形成city的概念树，这种方式便是半自动生成方式[7,8]. 有些属性只能由专家根据应用环境的要求进行人工定义.

属性概念树在数据挖掘过程中具有巨大的应用价值，但使用者需要很大的努力才能够生成属性概念树，不过这种工作只需做一次便可以满足以后所有数据挖掘的需要，这是因为属性概念树作为一种背景知识具有稳定性，因此用户的这种努力是值得的.

1.3 面向属性的归纳学习算法

某概念的正例集POS中所有数据记录的条件属性值是对该概念特征的详细描述，而决策者需要的是对概念特征的概括描述. 基于概念树的知识发现方法是一种归纳方法，它其实是一个元组合并的处理过程. 其基本思想是：首先，一个属性的较具体的值被该属性的概念树中的父概念所替代；然后，对表中出现的相同元组进行合并(merge identical tuples)，构成更宏观的元组(称为宏元组)，并计算宏元组所覆盖的元组数目；如果数据库记录生成的宏元组数目仍然很大，那么用这个属性的概念树中更一般的父概念去替代，或根据另一个属性进行概念树的提升操作，最终生成覆盖面更广、数量更少的宏元组；最后，将归纳所得的最后结果转换成逻辑规则.

利用面向属性的归纳算法挖掘特征规则是针对每一个概念进行挖掘的，通过对概念正例集POS的概括归纳出概念的特征规则. 对正例集POS的概括程度可以通过规定一个阈值β来进行控制，β的含义是将符合概念定义的数据划分为β类，算法的任务就是从数据中挖掘出每一类数据的特征描述X_i，β的大小是用户根据决策需要确定的，在例1中，如果用户规定$\beta=3$，其含义是将"masda626汽车买主"划分为3类，并概括出每一类用户的特征. 面向属性的归纳算法如下所示：

Algorithm：**DBLEARN**

Input A relational databases S，A concept tree H，threshold β，concept definition X

Output Characteristic rules learned from database **S**

1 Collect the concept _ relevant _ data from **S** into POS /{即形成正例集}

2 Attribute _ oriented induction in positive _ set **POS**

3 Simplification of the generalized _ positive _ set **POS**

4 Transform **POS** into a logical rule

该算法的核心是步骤2，下面是步骤2的算法：**Algorithm _ 1**

Algorithm _ 1(Attribute _ oriented induction in positive _ sample _ set)

Input A positive _ sample _ set **POS**，A concept tree **H** ，threshold β

Output generalized _ positive _ sample _ set **POS**

Begin for each attribute C_i in POS **do**

While number _ of _ distinct _ values _ in _ C_i > β **do**

Begin

if no higher level concept in the H for C_i

then remove C_i

else substitute for the values for C_i's by its corresponding minimal generalized concept

merge identical tuples

End

While number _ of _ tuples _ in _ generalized _ POS > β **do**

begin

Selectively generalize some attributes

merge Identical Tuples

End

End (Attribute _ oriented induction in positive _ set)

2 数据集粗糙度与面向属性的最优归纳学习算法

2.1 数据集的粗糙度

令$DB=\langle U,C,D,V,f\rangle$表示一个信息系统，其中$U=\{u_1,u_2,\cdots,u_N\}$是数据记录的集合，$C$为条件属性集，$D$为决策属性集，且要求$C\cap D=$

维普资讯 http://www.cqvip.com

$\varnothing$;令 $A = C \cup D$;V 表示所有属性可取值的集合;f:$U \times A \to V$ 是一个信息函数,该函数为某个数据记录的某个属性赋予一个特定的值,例如 $f(u_i, a) = \lambda$,表示记录 u_i 在属性 a 上的取值为 λ.

数据记录间往往存在着不分明关系,令 $B \subseteq A$,属性集 B 上的不分明关系 $R(B)$ 定义为:$R(B) = \{(u_i, u_j) \in U^2 | \forall\ a \in B,\ f(u_i, a) = f(u_j, a)\}$. 不分明关系是集合 U 上的等价关系,它将集合 U 划分为一些等价类. 令 $u_i \in U$,记录 u_i 在属性集 B 上的等价类定义为:$R_B(u_i) = \{u_j \in U | \forall\ a \in B,\ f(u_i, a) = f(u_j, a)\}$. 集合 U 上的等价类又称为基本集,基本集 $R_B(u_i)$ 中任一个数据记录在属性集 B 上所取的所有属性值称为该基本集的描述[9-11]. 例如(sex ="male") ∧ (married ="yes") 是一个描述,集合 U 中所有符合该描述的数据记录构成一个基本集,记为[(sex ="male") ∧ (married ="yes")]. 区分两种基本集:特征集 $R_C(u_i)$ 和概念 $R_D(u_i)$.

定义 1　如果确定基本集 $R_B(u_i)$ 的属性集 B 满足:$B = C$,即基本集是条件属性集上的等价类,称这一类基本集为特征集;如果 $B = D$,即基本集是决策属性集上的等价类,称该基本集为概念. 为描述方便,我们也用 $[X]$ 表示一个特征集 $R_C(u_i)$,其中 X 是特征集 $[X]$ 的描述;用 $[Y]$ 表示一个概念 $R_D(u_i)$,其中 Y 是概念 $[Y]$ 的描述.

一个特征集 $[X_i]$ 中的数据可能分别属于不同的概念,称这样的数据集为粗糙集. 粗糙集不能根据概念描述将数据集截然划分为互不相交的正例集和反例集,只能划分为近似集. 所谓概念 $[Y]$ 的下近似集 $PE(Y)$ 是指所有完全属于概念 $[Y]$ 的特征集的并,记为

$$PE(Y) = \bigcup \{[X_i] \subseteq U : [X_i] \subseteq [Y]\}$$

概念 $[Y]$ 的上近似集是所有与概念 $[Y]$ 的交不为空的特征集的并,记为

$$UPE(Y) = \bigcup \{[X_i] \subseteq U : [X_i] \cap [Y] \neq \varnothing\}$$

概念 $[Y]$ 的边界区域是指那些属于 $UPE(Y)$ 但又不属于 $PE(Y)$ 的基本集的并,记为

$$BIND(Y) = UPE(Y) - PE(Y)$$

边界区域中的数据越多,说明数据集越粗糙,那么由该数据集概括出的特征规则必然精度越低. 如果数据集中概念 $[Y]$ 的 $BIND(Y) = \varnothing$,那么符合该概念的特征描述的数据中必然不会包括属于其它概念的数据,则该概念的特征规则的精度就为 100% . 令 δ 为概念 $[Y]$ 的上近似集 $UPE(Y)$ 的粗糙度,$0 \leqslant \delta \leqslant 1$,$\gamma$ 表示概念 $[Y]$ 的特征规则的平均精度,则 $\gamma = 1 - \delta$. 文献[12]给出了上近似集 $UPE(Y)$ 的粗糙度的计算公式,如公式(1)所示

$$\delta = \frac{card(BIND(Y))}{card(UPE(Y))} \tag{1}$$

公式中运算符 $card()$ 是求集合中元素的个数. 公式(1)没有区分边界区域 $BIND(Y)$ 内不同特征集对数据集粗糙度的影响. 事实上同样属于边界区域 $BIND(Y)$ 的两个特征集对数据集粗糙度的影响往往是不一样的,如果 $[X_i]$ 中 50% 的数据属于概念 Y,而 $[X_j]$ 中 99% 的数据属于概念 Y,则 $[X_j]$ 比 $[X_i]$ 使数据集更粗糙. 为了衡量不同的特征集对数据集粗糙度的影响,需要测量特征集属于边界区域的程度(或称特征集的粗糙度). 特征集属于边界区域的程度越高,说明特征集的粗糙度越大,则它对数据集粗糙度的影响越大. 如果数据集的所有特征集粗糙度都很大,则数据集的粗糙度必然很大. 边界区域 $BIND(Y)$ 中特征集 $[X_i]$ 对数据集粗糙度的贡献率 δ_i 可以用公式(2)计算

$$\begin{aligned}\delta_i &= 1 - \frac{card([X_i] \cap [Y])}{card([X_i])} \\ &= \frac{card([X_i]) - card([X_i] \cap [Y])}{card([X_i])}\end{aligned} \tag{2}$$

公式中 $[X_i] \subseteq UPE(Y)$

该公式表明,如果特征集 $[X_i]$ 中属于概念 $[Y]$ 的数据所占比例越接近 100%,特征集越精确. 设 $UPE(Y)$ 中共有 n 个特征集,令 $\omega_i = \frac{card([X_i])}{card(UPE(Y))}$ 表示 $[X_i]$ 在 $UPE(Y)$ 所占权重. 概念 $[Y]$ 的上近似集的粗糙度可更精确地用公式(3)计算.

$$\begin{aligned}\delta &= \sum_{i=1}^{m} (\omega_i \times \delta_i) \\ &= \sum_{i=1}^{m} \frac{card([X_i]) - card([X_i] \cap [Y])}{card(UPE(Y))}\end{aligned} \tag{3}$$

公式中 $[X_i] \subseteq UPE(Y)$

2.2　面向属性的归纳学习的最优算法

性质 1　属性概念爬升使数据集的粗糙度向大的方向变化.

这是因为属性概念爬升后,原来不属于同一

维普资讯 http://www.cqvip.com

个特征集的数据可能合并为同一个特征集，这就使特征集包含属于其它概念的数据的机会大大增加，从而提高了数据集的粗糙度.

在面向属性的归纳中，概念爬升的路径是很多的，算法 DBLEARN 的一个主要缺陷是：没有对概念爬升的路径确定一个标准，这就使特征规则的挖掘中存在以下问题：

(1) 如果由用户来确定对哪些属性进行概念爬升，那么需要用户具有很高的程序设计知识，但大多数用户并不是软件专家.

(2) 如果由算法随机挑选一个属性进行概念爬升，挖掘出的特征规则的精度就不是最优的. 例如在例 1 中，如果在数据集中删除属性 sex 不改变数据集的粗糙度，那么意味着概念特征与性别无关，概念爬升时首先就应该将 sex 概念爬升到 any，但当参数 β 大于 DBLEARN 不可能将 sex 从数据集中删除.

为了使特征规则精确度达到最优，本文设计出一个新的算法 DBLEARN _ 2，该算法对 DBLEARN 进行了如下改进：进行概念爬升时不是任选一个属性，而是选择使数据集粗糙度的变化幅度最小的属性进行爬升. 这样就可以保证概念爬升的路径是最佳的，即经过算法 DBLEARN _ 2 归纳出的特征规则可最大限度地排除反例集中的数据，从而使特征规则更能反映出一个概念不同于其它概念的特征.

算法 DBLEARN _ 2 的一个难点是计算数据集的粗糙度，对此采用如下策略：首先令临时表 S_3 等于数据集 S，删除 S_3 中所有决策属性，然后对 S_3 和正例集 POS 进行同步概括. 在概括的"merge identical tuples"阶段，用若干计数器 $vote_i$ 分别累计出 POS 和 S_3 中每一类(即属于同一等价类的)记录合并的数量，这样就计算出了每个等价类的数量. POS 的计数器(记为：p_vote_i)累计出的是每个等价类与概念的交的数量，即 $p\_vote_i = card([X_i] \cap [Y])$；而 S_3 中的计数器(记为 s_vote_i)累计出的是每个等价类的总数，即 $s\_vote_i = card([X_i])$. 根据计数器的值计算出粗糙度就是一个非常简单的问题了.

Algorithm: DBLEARN _ 2 (Attribute _ oriented induction)

Input data set **S**, concept tree **H** , Threshold β, concept definition **Y**

Output Characteristic rules learned from database **S**

begin

Collect the concept _ relevant _ data from S into POS(Y) {即形成概念[Y]的正例集}

merge identical tuples of POS(Y) and accumulate the votes

$S_1 \leftarrow POS(Y)$, $S_2 \leftarrow POS(Y)$ /{S_1, S_2 是对正例集 POS(Y) 的临时备份}

$S_3 \leftarrow S$ /{S_3 是对数据集 S 的临时备份}

Remove all decision attributes from S_3

merge identical tuples of S_3 and accumulate the votes

While number _ of _ tuples _ in _ generalized _ **POS** $>$ **β do**

Begin

$\delta_1 \leftarrow 1$ {首先令粗糙度指标为最大值，以便于下面计算}

for each condition attribute C_i in S_2 **do** {通过该循环，找出一个使数据集粗糙度变化最小的条件属性进行概念爬升}

Begin

if no higher level concept in the H for C_i **then**

begin

remove C_i from S_2

remove C_i from S_3

end

else

begin

substitute for the values of S_2 for C_i's by its corresponding minimal generalized concept

substitute for the values of S_3 for C_i's by its corresponding minimal generalized concept

merge identical tuples of S_2 and accumulate the votes

merge identical tuples of S_3 and accumulate the votes

end

compute rough-degree δ_2 of UPE(Y)

if $\delta_2 < \delta_1$ then $S_1 \leftarrow S_2$, $\delta_1 \leftarrow \delta_2$

$S_2 \leftarrow POS(Y)$

End

维普资讯 http://www.cqvip.com

POS(Y)←S_1

End

Simplification of generalized _ POS

Transform generalized _ POS into a logical rule

End (DBLEARN _ 2)

2.3 算法时间复杂度分析

假设数据集 S 中数据记录的数量为 N,且描述概念特征的条件属性有 m 个(即数据集 S 中有 N 条记录,m 个条件属性),每个条件属性的概念树有 H 个层次,每个条件属性 X_i 可取的不同的值的数量为 V_i(即该属性概念树共有 V_i 个叶节点),不同属性间各种取值的最大组合的数量为 $K = V_1 \times V_2 \times \cdots \times V_m$. 在最坏情况下的算法时间复杂度分析如下:

对一个属性进行概括的计算主要包括两部分:(1)对该属性的数据概念爬升一个层次,合并属性值对应相同的记录. 用高层概念数据替换低层概念数据的计算时间为 N,将属性值对应相等的记录合并为一条记录(merge identical tuples)并累计计数器的数量的计算时间为 $N\log N$. (2)计算概括后的概念的上近似集的粗糙度:计算粗糙度主要根据概括后的 POS(Y) 和 S_3 中的计数器 *vote* 的数值进行计算. 设每次概括后 POS(Y) 中的记录数为 k_1,每次概括后 S_3 中的记录数为 k_2,最坏情况下,$k_1 = k_2 = V_1 \times V_2 \times \cdots \times V_m = K$,根据概括后的 POS($Y$) 和 S_3 中的计数器 *vote* 的数值计算粗糙度需要进行 K^2 次运算.

因为要同时对 POS(Y) 和 S_3 进行概括,所以对一个属性进行概括并算出粗糙度的计算时间为:$2N + 2N\log N + K^2$. 而算法 DBLEARN _ 2 对某个属性进行概括前,要分别对 m 个条件属性进行概括,并选择一个使数据集粗糙度变化最小的属性进行概念爬升,因此,最终确定一个属性并进行概念爬升的计算时间为:$m(2N + 2N\log N + K^2)$,因为概念树的最大层数为 H,所以对单个属性进行概括的时间最多为 $H \times m \times (2N + 2N\log N + K^2)$,对 m 个属性进行概括的计算时间为 $m \times H \times m \times (N + N\log N + K^2)$. 因为数据集 S 中 H , m 远远小于 N,因此算法时间复杂度为 $O(N\log N + K^2)$.

2.4 定量特征规则

如果对特征规则进行定量测定,便得到了定量特征规则. 定量特征规则的表示形式为 $Y \rightarrow X_1[m\%],[n\%]$or…. 规则中 $[m\%]$ 表示概念 $[Y]$ 中符合某特征描述 X_i 的数据占 $[Y]$ 中数据的比例,即 $m\% = card([Xi] \cap [Y]) \div card([Y_i])$;$n\%$ 表示概念 $[Y]$ 中符合某特征描述 X_i 的数据占 $[X_i]$ 中数据的比例,即 $n\% = card([X_i] \cap [Y]) \div card([Y])$.

以上两个数量值,只需用算法中计数器 *votes* 的数值就可以计算出来,因此不需要额外的计算花销.

3 特征规则查询命令

3.1 特征规则查询命令的语法

为了满足用户灵活的查询需要,我们提供给用户一种类似于 SQL 的特征规则查询命令:CHARACTER _ RULE _ SQL,该命令可以使用户方便地对决策概念,数据集的限制条件,规则的数量阈值等进行定义,使数据挖掘完全按用户的查询需要进行. CHARACTER _ RULE _ SQL 的语法如下所示:

MINE CHARACTER RULE FOR 〈concept definition〉
FROM〈table〉 [,〈table〉…]
IN RELATION TO 〈attribute _ name〉[,〈attribute _ name〉…]
[**WHERE** 〈condition 〉
[**AND /OR** 〈condition〉…]]
USING 〈β〉
[**QUANTITATIVE**]

因为在执行本命令时要调用 SQL 语言形成数据集 S,所以本命令的参数定义形式与 SQL 语言的形式保持一致. 查询语言中[ ]内为可选项,各个参数的含义是:

FOR 〈concept definition 〉:该子句用决策属性值的逻辑表达式定义一个概念,因为数据可能来自于不同的数据库,且定义一个概念的决策属性可能存在于不同的数据库中,因此数据来源多于一个数据库时决策属性前要注明该属性是哪个数据库中的.

FROM 〈table〉:这里的 **FROM** 子句是用来指示数据挖掘时数据是从哪些数据库(table) 获取

维普资讯 http://www.cqvip.com

的.因为源数据可能来自于不同的数据库,因此这里允许填写多个数据库文件的名称.

IN RELATION TO 〈attribute _ name〉:该子句用来规定用哪些条件属性来描述概念的特征.同样,如果这些属性属于不同的数据库文件,属性前要注明该属性所属的文件名.

WHERE 〈condition〉:该子句是可选项,用于设定数据集的过滤条件.例如在例1的数据挖掘任务中,如果决策者只关心近3年的业务数据,那么通过该子句的设定就可以将近3年的数据从源数据集中过滤出来.

USING 〈β〉:该子句用于确定将符合概念描述的数据划分的类别的数量.

[**QUANTITATIVE**]:该参数为可选项.如果命令中没有该参数,表示要挖掘的是定性特征规则;如果有该参数,表示要挖掘的是定量特征规则.

根据以上说明,例1的特征查询命令可定义为:"**MINE CHARACTER RULE FOR** buy _ car = "masdas626" **FROM** transaction.dbf **IN RELATION TO** sex , salary , own _ house , married **USING** 3 **QUANTITATIVE**".该查询命令可理解为:从transaction.dbf数据库中发现"masda626汽车买主"这一概念的定量特征规则,规则需要将masda626汽车买主划分为3类,并从性别、年薪、是否有住房、婚否等方面描述每一类用户的特征.

3.2 特征规则的发现步骤

以算法DBLEARN _ 2为核心,在查询语言CHARACTER _ RULE _ SQL的驱动下,用户定义的各种概念便可以被挖掘出来.挖掘特征规则主要有3个步骤:

(1)用查询语言CHARACTER _ RULE _ SQL对查询任务进行定义.

(2)根据用户对查询任务的定义,调用结构化数据查询语言SQL将数据挖掘所需要的数据从各个源数据库中过滤并汇总到一个数据集S中.例如有特征查询命令为:"**MINE CHARACTER RULE FOR** buy _ car = "masdas626" **FROM** transaction.dbf **IN RELATION TO** sex , salary , own _ house , married **WHERE** buy _ date〉1995/12/31" **USING** 3.运行该命令时,首先要运行一个SQL命令:"**SELECT** buy _ car, sex , salary , own _ house , married **FROM** transaction.dbf **INTO S WHERE** buy _ date〉1995/12/31",将数据汇聚到数据集S中.

(3)调用算法DBLEARN _ 2归纳出概念的特征规则.

4 实例分析

了解各种商品房用户的特征对指导房地产开发具有重要的决策参考价值.我们对沈阳市近三年购买商品房的用户进行了抽样调查,样本量为2000人,并将调查数据存储在关系数据库estate _ information中,调查内容如表1所示.其中建筑类型有普通住宅、高层住宅、公寓等类型,房型有一室一厅、二室一厅等类型.年龄,家庭人口,家庭月收人等属性的概念树通过统计方法自动生成,其它属性的概念树由专家进行定义.

表1 商品房用户调查内容

决策属性	条件属性
建筑类型	户主年龄
房型	户主受教育程度
每平米单价	户主职业
地点	户主婚姻状况
	家庭人口
	家庭月收入
	买房主要原因
	买房资金来源

为了说明我们的算法在概括精度上的改进,我们分别用DBLEARN _ 2与DBLEARN两种算法对"普通住宅且每平米单价在1000 — 3000元之间的商品房买主"的概念特征进行了查询,表2是用两种方法挖掘出的规则在精度上的对比.

表2 实验结果

规则概括度阈值 β	概括精度$(1-\delta)$	
	DBLEARN _ 2	DBLEARN
20	83.4%	80.1%
16	81.1%	76.1%
13	79.2%	74.5%
10	75.9%	67.2%
8	72.3%	63.2%

5 结论

本文提出了一个新的面向属性的归纳学习算

维普资讯 http://www.cqvip.com

法,该算法在保证规则解释能力的前提下使精度达到最优,并以此形成了一个面向查询的特征规则挖掘工具:CHARACTER _ RULE _ SQL,由于该查询工具具有灵活的查询定义机制及快速的应答能力,因此在决策支持系统中具有广阔的应用前景.

参 考 文 献:

[1] Chen Ming-Syan, Han Jiawei, Yu Philip S. Data mining: an overview from a database perspective[J]. IEEE Transactions on Knowledge and Data Engineering, 1996,8(6): 866-883

[2] Han Jiawei, Cai Yandong, Cercone Nick. Data-driven discovery of quantitative rule in relation databases[J]. IEEE Transactions on Knowledge and Data Engineering, 1993,5(1): 29-40

[3] 洪家荣. 归纳学习[M]. 北京: 科学出版社,1997,1-42

[4] Cai Y, Cercone N, Han J. Attribute oriented induction in relational databases[M]. In Piatetsky Shapiro G, Frawley W J ed. Knowledge Discovery in Databases. AAAI Press, Menlo Park Calif, 1991,213-228

[5] Carter Colin L, Hamilton Howard J. Efficient attribute-oriented generalization for knowledge discovery from large databases[J]. IEEE Transactions on Knowledge and Data Engineering, 1998 ,10(2): 193-208

[6] 刘胜军,杨学兵,彩庆生.关系数据库中概念层次自动提取算法研究[J].计算机应用研究,2000,16(12):15-17

[7] Cheng Y, Fu K S. Conceptual clustering in knowledge organization[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,1985,9(5):592-598

[8] Michalski R, Stepp R. Automated construction of classifications: conceptual clustering versus numerical taxonomy [J]. IEEE transactions on Pattern Analysis and Machine Intelligence,1983,5(4):396-409

[9] Hu X, Cercone N, Han J. Learning in relational databases: a rough set approach[J]. Computational Intelligent, 1994,11(2):323-338

[10] Ziarko Wojcicch. Variable precision rough set model[J]. Computer and System Science,1993,46(1):39-59

[11] 刘 真, 黄兆华, 姚立文[J]. ROUGH 集理论:现状与前景. 计算机科学, 1997,24(1):1-5

[12] Pawlak Z. Rough set[J]. International Journal of Information and Computer Science,1982,11(5): 341-356

A query-driven method of mining characteristic rules

CHENG Yan , HUANG Ti-yun

Management School, Harbin Institute of Technology, Harbin 150001, China

Abstract: What the query sub-system of a information system provide is large amount of original data, but data is not information, the regularity and rule hiding in data is the information which is useful for decision-making. So the data mining technology, which can discover these regularities and rules from data automatically, is an important method of realizing intelligent decision support system. Mining characteristic rules in relational databases is an important content of data mining. Many of current methods of data mining can discover the characteristic rules related to a specific concept, but they are not effective for flexible query. This problem can be solved by using attribute-oriented induce algorithm, but this algorithm can not guarantee the optimum precision of rule. In this paper, by integrating the theory of rough sets, a method of measuring the rough-degree of data set is presented, and a new attribute-oriented induce algorithm of mining characteristic rule is developed, which is not limited by the condition of query. The characteristic rule mined by this algorithm can describe the characteristics of a concept effectively. On the basis of this new algorithm, a SQL-like language of querying characteristic rule is designed, which can make the interaction between user and system more convenient.

Key words: data mining; decision support system; characteristic rules; rough set