文章编号:1005－0523(2007)02－0106－04

基于 Rough 集理论的增量式数据挖掘算法

陈红丽

(华东交通大学 信息工程学院,江西 南昌 330013)

摘要:提出一种求精简规则的带支持信息的增量式算法 IDMA,该算法改进了传统挖掘算法的缺点,可充分利用已挖掘出的规则集来对新增实例进行决策规则挖掘,同经典 Rough 集挖掘算法比较,算法 IDMA 计算过程简单,而且效率较高.

关 键 词:Rough 集理论;增量式挖掘;支持度;决策规则

中图分类号:TP391　　**文献标识码**:A

1 引言

Rough 集理论是波兰科学家 Z·Pawlak 在 1982 年提出的一种处理含糊和不精确性问题的一种新型数学工具[1],其主要思想是,在保持信息系统分类不变的前提下,通过对知识约简,导出对问题的决策或分类规则.目前,Rough 集理论已被应用于机器学习、故障诊断、控制算法获取、过程控制以及关系数据库中的知识获取等各种应用领域.Z·Pawlak[2]指出机器学习中的若干问题都可以用 Rough 集理论和方法来解释、分析和处理.在文[2]中,提出了两个关于机器学习的新术语:静态学习和动态学习;前者也称非增量式学习,后者则称增量式学习.

在经典的方法中,Rough 集应用于数据挖掘一般由以下几部分构成:

(1)把数据集转化成数据表表示;

(2)根据数据表,删除多余的属性(属性的约简);

(3)删除重复的行;删除属性多余的值;

(4)求出最小约简;

(5)根据最小约简,导出逻辑规则.

Rough 集应用于数据挖掘的基本计算消耗是约简的求解,包括求属性集合的约简、求值的约简.但是,对一个信息系统而言,求出所有约简与求最小约简都是 NP－难问题,因此,常常借助于某种启发式来求近优解.

2 相关理论介绍

2.1 有关决策逻辑语言基本概念

在逻辑语言中,含义 $\theta\rightarrow\Psi$ 称为知识表达语言中的决策规则,θ 和 Ψ 分别称为决策规则的前代和后继.当 S 中决策规则为真时,说该决策规则是 S 中协调的,否则说该决策规则是 S 中不协调的.当 $\theta\rightarrow\Psi$ 为一个决策规则,且 θ 和 Ψ 分别为 P 基本公式和 Q 基本公式,P、Q 已知,则决策规则 $\theta\rightarrow\Psi$ 称为 PQ 基本决策规则,简称为 PQ 规则.

决策逻辑语言中任何有限决策规则集称为决策逻辑语言中的决策算法,而任何有限基本决策规则称为一个基本决策算法.当基本决策算法中所有的决策规则都是 PQ 决策规则时,该算法称为 PQ 决策算法,或简称 PQ 算法,记作(P, Q).当且仅当所有决策规则是协调的,则 S 中 PQ 算法是协调的,否则 PQ 算法是不协调的.

2.2 决策表最小化

一个决策表的最小化[3]包括两个步骤:第一步:属性约简算法,即从 PQ 算法得到 RQ 基本规则(RQ 算法)$R=RED(P, Q)$;第二步:值约简算法,即得到

收稿日期:2007－01－15

作者简介:陈红丽(1977－),女,满族,辽宁省绥中县人.

RQ 基本规则的所有约简,并化简得到一个最小规则集.

为了验证算法先引进一个如表 1 所示的决策表.其中,U={1,2,3,4,5,6,7},A={a,b,c,d,e}.P={a,b,c,d}和 Q={e}分别为条件属性和决策属性.经过计算得到所有 RQ 基本规则的简化如表 2 所示:

表 1　决策表

U	a	b	c	d	e
1	1	0	0	1	1
2	1	0	0	0	1
3	0	0	0	0	0
4	1	1	0	1	0
5	1	1	0	2	2
6	2	2	0	2	2
7	2	2	2	2	2

表 2　所有 RQ 基本规则的简化

U	a	b	d	e
1	1	0	x	1
1′	x	0	1	1
2	1	0	x	1
2′	1	x	0	1
3	0	x	x	0
4	x	1	1	0
5	x	x	2	2
6	2	x	x	2
6′	x	2	x	2
6″	x	x	2	2

其中属性约简 R={a,b,d}.

2.3　动态学习

众所周知,学习过程中经常会有新的例子加进来,那么新例子用已有的知识是否就可以判断其归属呢? 利用先前得到的知识来对新例子进行分类被认为是一种动态学习[3].在文[3]中,Pawlak 认为加入一个新的例子有 3 种情况:

(1)新例子与实际知识相同(指新例子已经出现过);

(2)新例子与实际知识矛盾(新例子的前代在已有例子中出现,但他们的后继不同;)

(3)新例子完全是新的情况(新例子不属于已知的任何决策类).

3　一种带支持信息的增量式挖掘算法

首先,我们来分析一下规则出现重叠的几种情况.

1) 规则 r1:$\theta_1 \rightarrow \Psi_1$ 与 r2:$\theta_2 \rightarrow \Psi_1$ 的后件相同,但 $\theta_1 \subset \theta_2$,即支持 r2 的元组数比支持 r1 的元组数多.

2) 规则 r1:$\theta_1 \rightarrow \Psi_1$ 与 r2:$\theta_2 \rightarrow \Psi_1$ 的后件相同,且 $\theta_1 = \theta_2$,即支持 r2 的元组数和支持 r1 的元组数一样多,但与 θ_1 相关联的属性条数比与 θ_2 相关联的属性条数少.

根据 Occam 原理:一个概念能用更简单的形式表达,就不用更复杂的形式表达.另外,一个覆盖更大的规则应该更有效;当所有元组都被覆盖时,挖掘不必再进行下去;已经覆盖的元组不必再参与挖掘.因此,我们将以根据新增实例与已有决策规则三种关系以及规则重叠的情况,利用 IDMA 算法从数据表中挖掘出描述简洁且具有高支持度的规则构成的无冗余的规则集.

IDMA 算法主要步骤如下:

1) 比较新增实例与原有规则的关系,如果相同则保持原有规则集不变,挖掘结束;如果矛盾则进行步骤 2);如果是完全新则进行步骤 3).

2) 与相同后件的规则进行比较,如果前件关联的属性数目一样(即同样简单)时,选取覆盖元组数更多的规则.

3) 对全部元组,先挖掘具有更简单描述的规则,再进行步骤 2)的操作,直至所有元组都被规则覆盖为止;对已确定的规则,删去已经被该规则覆盖的元组.

IDMA 算法如下:

```
Mimed-RuleSets={已挖掘出的规则集} //从原挖掘系统中获得
RuleSets={Null→Null,Null}
//规则集中每个元素记录了前件,后件及规则覆盖的元组 3 个方面的信息
If newX∈ Mimed-RuleSets then
//如果新增例子属于原有规则集,输出规则并退出
   Output(Mimed-RuleSets)
   Exit
Else
   C-AttSets={x1,x2,…,xn}
//假设有 n 个条件属性,一个决策属性 d
   Elements={1,2,…,m}
//假设数据表共有 m 个元组
   D-Class=U/IND(d)
//首先计算出论域关于决策属性的等价类
```

```
While Elements≠{}
  For i= 1 to n
//属性数由少到多顺序进行挖掘
    RS-Temp={Null→Null,Null}
  //供临拾存放前件含 i 个属的规则集
    For Each subset X of C-AttSets that
    contains i elements
    For Each element Y of U/IND(X)
    If Exists Z∈D-Class such that Y⊆Z
    且 RuleSets 和 RS-Temp 的覆盖中没有
    包含 Y 的项 Then
      RS-Temp= RS-Temp+ {Des(Y)→
      Des(Z),Y}
    EndIf
  Next Y
 Next X
CheekTemp()
//检查 RS-Temp 中是否有冗余的规则
RuleSets=RuleSets+ RS-Temp
//加入由 i 个属性的描述构成的前件所
形成的规则之规则集
RemoveElements(RS-Temp)
//删除已被规则覆盖的元组
If IsNull(Elements)Then Exit
  While
    Next i
    End While
endIf
Output(RuleSets)
```

4 实例分析

以表1作为决策表,该决策表属性约简为 $R=RED(P,Q)=\{a,b,d\}$.先用IDMA算法对决策表1进行规则挖掘,得出决策规则集(1)如下:

$\{a_1b_0\rightarrow e_1,(1,2)\}$

$\{a_0\rightarrow e_0,(3)\}$

$\{b_1d_1\rightarrow e-0,(4)\}$

$\{d_2\rightarrow e_2,(5)\}$

新增加的例子 x:$a_0b_1c_2d_1\rightarrow e_1$ 下面分别用传统算法和 *IDMA* 算法来求增加 x 后的决策规则集.

4.1 传统算法

在传统算法中,当有新增加的例子时,就重新计算.因此,首先需要重新计算 U' 的 PQ 约简,经过计算得到 $R'=\{a,b,d\}$.接下来,计算 PQ 基本规则的所有约简(值约简),得到表3.

表3 所有基本规则的简化

U	a	b	d	e
1	1	0	x	1
1′	x	0	1	1
2	1	0	x	!
2′	1	x	0	1
3	0	x	x	0
4	x	1	1	0
5	x	x	2	2
6	2	x	x	2
6′	x	2	x	2
6″	x	x	2	2

根据表3得出的一个决策规则集为:

$a_1b_0\rightarrow e_1$

$a_0b_0\rightarrow e_0$

$a_1b_1d_1\rightarrow e_0$

$d_2\rightarrow e_2$

$a_0b_1\rightarrow e_1$

4.2 增量式挖掘算法 IDMA

首先通过比较 x 与原有规则的关系,可以得出 x 与原有规则集矛盾,与规则集 1)中的规则1、2、3矛盾,利用IDMA算法对增加 x 后的决策表进行挖掘,得出决策规则集如下:

$\{a_1b_0\rightarrow e_1,(1,2)\}$

$\{a_0b_0\rightarrow e_0,(3)\}$

$\{a_1b_1d_1\rightarrow e-0,(4)\}$

$\{d_2\rightarrow e_2,(5)\}$

$\{a_0b_1\rightarrow e_1,(x)\}$

从本算例我们可以看出算法IDMA具有如下的两个特点:(1)保持了经典Rough集挖掘出的规则集的精简的特点:规则集所含规则的数目较少,每条规则所相关的属性也较少.(2)采用算法IDMA挖掘出的规则集在精简的同时保存有支持元组的信息,这为我们保证规则集的支持度提供了可靠的依据,而这正是经典Rough集方法所缺少的.

5 算法分析

对于IDMA算法和传统算法的时间复杂度的分析这里不给出详细的分析过程,只是给出结论:当 x 与原有规则集矛盾或完全新时,增量式算法IDMA优于传统算法;而当 x 与原有规则集相同时,带支持信息的增量式算法IDMA在数量级(O(1))上远远优

于传统算法时间复杂度($O(2^m n^2)$).

6　结论

为了获取最小决策规则集,当新增例子时,传统的方法通常需要对决策表中所有数据重新计算,效率欠佳.为了尽量减少重复计算量,本文从 Rough 集理论出发,首先对新例子 x 与原来的决策规则集的关系做了分析和探讨,从而提出了一种新的带支持信息的增量式挖掘算法.从理论上对新算法和传统算法在算法时间复杂度方面做了分析对比.算法 IDMA 利用 Rough 集在数据约简方面的长处,并根据实际数据挖掘的特点和充分利用属性数据量的信息直接从数据表中挖掘高支持度且描述长度小的规则集,从而避免了需先求属性集合的约简再进行规则集的约简的过程.其效率主要与属性的个数相关,当属性取不同值的数目不大时是一个高效算法.

参考文献:

[1]Pawlak Z. Rough Set[J]. International Journal of Computer and Information Sciences, 1982, 11(5): 341－356.

[2]Pawlak Z. On Learning－a Rough Set Approach[C]. Proc. Intl. Symp. On Computeation Theory and Lecture Notes in Computer Science, G. Goos, et al. (eds.), 1984, 208: 197－227.

[3]Pawlak Z. Rough Set: Theoretical Aspects and Reasoning About Data[M]. Kluwer Academic Publishers, 1991.

[4]安利平,吴育华,等.增量式获取规则的粗糙集方法[J].南开大学学报.2003,36(2),98～103.

[5]孙国梓,郁鼎文,等.基于粗糙集的全局产品结构模型研究[J].计算机学报,2005,28(3),392～401.

A Rough Set Based Incremental Data Mining Algorithm

CHEN Hong-li

(School of Information Eng., East China Jiaotong Univ., Nanchang 330013, China)

Abstract: In this paper an incremental data mining algorithm with support is presented for mining simplified rules. IDMA has improved the shortcoming of the classical method, and makes full use of mined rules to mine decision rules from new instance. IDMA algorithm is a high efficiency with simple calculation compared with classical Rough set mining algorithm.

Key words: rough set theory; incremental mining; support; decision rule