维普资讯 http://www.cqvip.com

计算机科学 1999Vol 26. № 7

一个不完整分类规则挖掘算法

An Algorithm for Discovery Uncomplete Classification Rule

彭伟成 虞险云 葛叶冰 葛家翔
(复旦大学计算机科学系 上海 200433)

Abstract For dealing with the incomplete or delaying information in the real time system, we present an algorithm for mining uncomplete classification rules in this paper and the algorithm is based on Rough Set theory.

Keywords Expert system, Uncomplete classification rules, Rough set, Concept hierarchical

1 引言

在专家系统中,知识获取被公认为是最大的"瓶颈"。目前,随着数据挖掘的兴起,出现了大量的算法,这些算法中利用 Rough 集方法是非常有效的,因为此类算法不需要关于数据的预先的或附加的信息,因此可用于专家系统中的知识获取,以克服知识获取的"瓶颈"。利用 Rough 集理论的挖掘算法[1]主要着重生成与数据一致的规则,且这些规则所利用的属性是最少的。但由于各种原因,进入专家系统的数据往往不完全或有错,特别在实时专家系统中,这种情况很突出。文[1]中算法生成的规则往往不能对此种情形进行处理。例如假设专家系统的已有历史数据都存储在数据库中,以表 1 的形式表示,a,b 是条件属性,d 是决定属性。每一行表示系统一次采集的数据,vote 表示相同的数据进入的次数。一般算法[1]可得到如下规则:

$$a=1\wedge b=1\rightarrow d=1 \quad a=1\wedge b=0\rightarrow d=2$$
$$a=0\wedge b=1\rightarrow d=3 \quad a=0\wedge b=0\rightarrow d=4 \qquad (1)$$

表 1

	a	b	d	vote
E1	1	1	1	26
E2	1	0	2	23
E3	0	1	3	26
E4	0	0	4	25

现在,一新记录 E5:a=1,b=2 进入数据库,专家系统的推理机制将这一记录做为事实与(1)中规则结合进行推理。由于 b 值错误,推理机制无法对 d 值做出决策。事实上,如果属性 b 被忽略,则上面的规则可约制成如下规则

$$a=1\rightarrow d=1\wedge 0.531 \quad a=1\rightarrow d=2\wedge 0.469$$
$$a=0\rightarrow d=3\wedge 0.510 \quad a=0\rightarrow d=1\wedge 0.490 \qquad (2)$$

显然,结合以上规则,专家系统首先作出推理 d=1,当利用 d=1 不能进一步推理时,则回溯 d=2。对(2)中的规则一般算法是不生成的,此类规则的生成实际上是通过逐步减少条件属性的个数,即每次只利用条件属性的一个子集来生成规则。例如,表 1 条件属性为 a,b,其属性子集有{a,b},{a},{b},Φ。一般算法只利用属性子集{a,b}来生成规则,而我们的算法还会利用其它属性子集来生成规则。上面的规则(2)就是利用属性子集{a}来生成的。最极端的情形是条件属性个数为 0,即利用属性子集 Φ 来生成规则,则专家系统的决策判断仅依赖于决定属性取值的概率分布。由于这些条件属性的所有子集构成一个格空间,每个子集称为格空间上的一个结点,我们可通过遍历这个格空间上的结点来生成所有规则,因为有些结点产生的规则包含在其它结点产生的规则集中,我们利用分明矩阵舍去这些结点,同时利用分明矩阵生成的分明函数来构造规则。为了减少生成分明矩阵时的计算量,我们应用面向属性的方法[3]来对属性值进行概括。我们称这种在数据的不完整性上得到的分类规则为不完整分类规则。

2 Rough 集基本概念[1]

定义 2.1(信息系统) 决策系统信息系统(IS)是一有序对 A=(U,A),其中 U 是非空有限对象集,A 是 U 上的属性集。对每一个属性 $a\in A$ 有 $a:U\rightarrow V_a$,其中 V_a 是属性 a 的取值范围。以下用 x_i 表示

维普资讯 http://www.cqvip.com

对象 x 的属性 a 在 V_a 中的取值。一个决策系统也是一个信息系统 ISA=(U,A),但是 A 中的属性进一步区分为条件属性 C 和决策属性 D($A=C\cup D$,$C\cap D=\Phi$)。(以下定义都是针对决策系统而言)

一个决策系统可表示成一个二维数据表格(以下都采用这种表示方式),其中每一行代表一个对象,列代表对象的属性值。如表 2,有 3 个对象。频率、方位角,距离是条件属性,发射方式是决策属性,vote 意义见后。

表 2

	频率	方位角	距离	发射方式	vote
1	C 波段	锐角	中	方式一	9
2	C 波段	钝角	远	方式二	4
3	E 波段	锐角	远	方式三	3

定义 2.2(分类规则,最少属性分类规则) 一个分类规则是形如以下形式的规则:

$$c_1(v_1)\wedge c_2(v_2)\wedge\cdots\wedge c_m(v_m)\rightarrow d(v)\wedge p \quad (1)$$

或 $$\rightarrow d(v)\wedge p \quad (2)$$

其中 $c_i(1\leqslant i\leqslant m)$是条件属性,d 是决策属性;$c_i(v_i)$表示属性 c_i 取值 v_i;p 表示规则的正确率,p=(符合规则的对象数)/(符合规则左面属性条件的对象数),若 p=1 则省略 p。(1)式这种形式表示若一个对象 x 满足 $x_{c_i}=v_i(1\leqslant i\leqslant m)$,则这个对象的决策属性 d 值为 v 的可能性为 p。(2)式表示规则的条件属性个数为 0,决策属性 d 取值为 v 的可能性为 p。

若一个分类规则 $c_1(v_1)\wedge c_2(v_2)\wedge\cdots\wedge c_m(v_m)\rightarrow d(v)\wedge p$ 满足 $\forall c_k(1\leqslant k\leqslant m)$、$\wedge c_i(v_i)\rightarrow d(v)\wedge p1(1\leqslant i_1\leqslant m, i_1\neq k), p1\neq p$。即去掉规则的任一个属性、规则的正确率就改变。这样的规则我们称为最少属性分类规则。

定义 2.3(分明矩阵[2]) 在 $A=(U,C\cup D)$中,U 中有 n 个对象,$B\subseteq C$、分明矩阵 $M(B)=\{m(i,j)\}_{n\times n}$定义如下:

$$m(i,j)=\{a\in B\mid a(x_i)\neq a(x_j)\} \quad i,j=1,2,\cdots,n$$

即 m(i,j)是 B 中可区分对象 x_i,x_j 的所有属性的集合,显然 M(B)是对称的。对表 2 所计算出来的分明矩阵如表 3。我们以 a,b,c 分别表示频率,方位角、距离。

表 3

	1	2	3
1		b,c	a,c
2			a,b
3			

定义 2.4(分明函数[2]) 分明函数 $f(B)=\wedge\vee\delta(x_i,x_j)$、$B\subseteq C$。我们将分明矩阵 M(B)中 m(i,j)的每个元素看作布尔变量,$\vee\delta(x_i,x_j)$表示这些布尔变量的析取。和表 3 对应的分明函数 $f(B)=(b\vee c)\wedge(a\vee c)\wedge(a\vee b)$。

定义 2.5(概念层次) 为了获得有用或有意义的规则,信息系统的数据必须抽象到一定程度。图 1 就是一个表示波段概念层次(为简化起见,我们以集合的形式来表示),每一"⊂"左面的概念都是相同层次,相应右面的概念是比左面高一层次的概念。

{1MHz..25MHz}⊂A1 波段,{26MHz..50MHz}⊂A2 波段,…,{226MHz..250MHz}⊂A10 波段
{A1 波段,A2 波段, ,A10 波段}⊂A 波段
{251MHz..275MHz}⊂B1 波段,{276MHz..300MHz}⊂B2 波段,…,{476MHz..500MHz}⊂B10 波段
{B1 波段,B2 波段, ,B10 波段}⊂B 波段
{501MHz..550MHz}⊂C1 波段,{551MHz..600MHz}⊂C2 波段,…,{951MHz..1000MHz}⊂C10 波段
{C1 波段,C2 波段,..C10 波段}⊂C 波段
…
{60001MHz..64000MHz}⊂M1 波段,{64001MHz..68000MHz}⊂M2 波段,…,{96001MHz..100000MHz}⊂M10 波段
{M1 波段,M2 波段, ,M10 波段}⊂M 波段
{A 波段,B 波段,C 波段,..M 波段}⊂任意(波段)

图 1 概念层次

3 算法

专家系统中的历史数据被存储在数据库中,可将其看做信息系统的一种形式,每一元组看为一对象。这一节,给出的算法用于从数据库中挖掘分类规则。算法大体上分三步。首先,将面向属性的概念层次树上升技巧用于数据的分析,以便移去不相关的属性和将相关属性的数据值抽象到一定程度,设置相关的阈值;接着利用 Rough 集的分明矩阵来生成属性格空间要遍历的结点;最后遍历格空间,生成相应规则。

3.1 面向属性的概念树方法[3]

表 4 为某实时武器系统中的数据,其条件属性为圈数、频率、方位角、距离、决策属性为导弹发射方式。

维普资讯 http://www.cqvip.com

表 4

圈数	频率(MHz)	方位角(°C)	距离(kM)	发射方式
1	551	30	876	方式 1
2	800	25	1100	方式 1
3	988	47	1589	方式 1
4	700	78	876	方式 1
5	689	24	1197	方式 1
6	800	67	798	方式 1
7	900	150	1056	方式 3
8	490	112	1098	方式 3
9	796	116	1187	方式 3
10	745	47	1123	方式 1
11	2570	34	794	方式 2
12	2800	56	623	方式 2
13	545	67	980	方式 1
14	505	21	1002	方式 1
15	2990	156	1098	方式 3
16	2990	45	750	方式 2

(0 800)⊂近,(801-1200)⊂中,(1201 1600)⊂远,|近,中,远|⊂任意(距离)
(1 89)⊂锐角,(91 179)⊂钝角,|锐角,直角,钝角,)⊂任意(方位角)

图 2

图 1 和图 2 是表 4 的属性概念层次树。下面用表 4 来说明算法的过程。为了生成所有不完整分类规则,我们采取的策略步骤如下:

策略 1 面向属性的概括 在一信息系统 ISA=(U,C∪D)中,设集合 $U_a=\{a(x)|x\in U\}$表示 U 中对象在属性 a 下不同取值的集合,$|U_a|$表示集合中元素的个数。一般而言,U_a 是一个非常大的集合。为了减少生成规则时的计算量和使规则具有一定的代表性,我们必须利用概念层次树对 U_a 进行抽象。表 4 的第一个属性 P# 是键属性,即每一个对象在此属性下的取值都是不同的,如果在概念层次树上没有这个属性的高层次上的概念,那么这个属性下的取值就不能简化概括,我们就移去此属性,所以我们移去属性圈数。反之若属性在概念层次树上有高层次的概念,那么就将所有对象的属性值以高层次的概念值取代。例如,将"551"(频率)替代成"C2 波段","789"(距离)替代成"近"。

策略 2 增加属性 vote 作为策略 1 的执行结果,必然会有许多不同的对象由于属性值完全相同而合并成一个对象。为了反应这一变化,引入属性 vote 来记录最初的表中不同对象被概括成现行表中相同对象的个数。利用 vote 可定义规则的正确率 P,用于规则的取舍,P=(符合规则的 vote 值)/(符合规则左面属性条件的 vote 值)。

策略 3 阈值的设置 有时对一个属性 a 而言,为了将$|U_a|$减少到一定程度,必须沿着 a 的概念层次树上爬几次。为了控制这个过程,有必要对$|U_a|$设置一个阈值,只要$|U_a|$达到这个阈值,沿相应概念层次树的进一步概括就不必进行,相反,如没达到这个阈值,则进一步概括必须进行。例如在这个例子中,阈值设置成了 2,则对属性频率而言,必须进行两个抽象概括(551-C2 波段—C 波段)。通过上述策略,对表 4 可简化到表 2。

3.2 格空间的形成和规则的生成

在规则生成过程中,实际每一次都是利用信息系统的条件属性的一个子集,生成的规则依赖这个属性子集。对一个信息系统,设有 n 个条件属性,则有 2^n 个属性子集,例如考虑表 2 所示的信息系统的条件属性集,其不同的属性子集有 8 个,其组成的格空间如图 3 左边所示。如果信息系统条件属性数量很多话,则利用这些属性子集表生成规则显然是一个不可行的方法。事实上大量属性子集产生的都是在推理效果上等同的规则,从分明矩阵的构成可以看出,因为分明矩阵中的每个元素 m(i,j)一般是条件属性的一个很小子集,这个属性子集在生成最少属性分类规则时是必不可少的[2]。而我们每次只有去掉这些必不可少的子集后,在剩下的属性子集上生成的规则才不都包含在其它属性子集生成的规则集中。表 2 的分明矩阵如表 3 所示。由其分明矩阵得到的条件属性子集的集合为 $C^*=\{\{a,b\},\{b,c\},\{c,a\}\}$。我们首先从条件属性全集{a,b,c}这个格空间上最大的结点出发,去掉 C^* 中第一个子集{a,b},得到结点{c},由于结点{c}中没有 C^* 中的子集,于是去掉{c}本身,得到结点 Φ。类似再从{a,b,c}出发,依次去掉 C^* 的其它元素得到{b},{a}结点。最后形成的要遍历的条件属性格空间如图 3 右边所示。每一个"■"表示要遍历的结点,实线表示遍历路径。

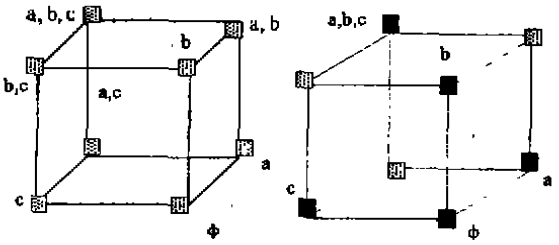

图 3

要遍历的格空间形成后,依赖于每个结点对应的分明矩阵,得到相应的分明函数,我们可将分明函数化成 MDNF[5]形式,再利用布尔代数吸收定律来

维普资讯 http://www.cqvip.com

简化成最简形式[7]，然后结合属性值即可得到分类规则，且规则是最少属性分类规则[2]。对图3右边的第一个遍历结点{a,b,c}，由表3的分明矩阵得到的分明函数为后得到的最小属性分类规则如下：

(1)c(中)→d(1),a(C波段)∧b(锐角)→d(1),b(锐角)→d(2),a(C波段)∧c(远)→d(2),a(E波段)→d(3),b(锐角)∧c(远)→d(3)
c()∧a()→d(3)

(2)a(C波段)→d(1)∧0.692,a(C波段)→d(2)∧0.308,a(E波段)→d(3)

(3)b(锐角)→d(1)∧0.750,b(钝角)→d(2),b(锐角)→d(3)∧0.250

(4)c(中)→d(1),c(远)→d(2)∧0.571,c(远)→d(3)∧0.429

(5)→d(1)∧0.563,→d(2)∧0.250,→d(1)∧0.187

(1)到(5)分别表示遍历{a,b,c},{a},{b},{c},Φ后得到的规则。以d表示发射方式，d(1),d(2),d(3)分别表示方式1，方式2，方式3。

3.3 算法的表示

下面的函数表示遍历从分明矩阵中所生成属性格空间的结点，通过遍历这些结点来生成不完整分类规则的函数。为简单起见，我们仅仅表达主要思想，通过递归来实现深度遍历格空间，没有说明如何去避免在遍历属性格空间时每个结点仅遍历一次。函数的四个参数依次为信息系统的对象集U，条件属性C，决策属性D，相应的分明矩阵Dismatrix。

```
FUNCTION UnCompleteRuleGen(U,C,D,Dismatrix)
    BEGIN
        IF Dismatrix=NULL
           /*函数ComputerDiscernibilityMatrix用来计算分明矩阵[2]*/
           Dismatrix=ComputerDiscerniblityMatrix(U,C)
        /*函数GenerateRules(U,C,D,Dismatrix)用来生成最少属性分类规则[2]*/
        GenerateRules(U,C,D,Dismatrix)
        For Dismatrix中上三角每一元素 el
            ReducedDismatrix=Dismatrix去掉所有el
        /*C-el表示从条件属性C中去掉el*/
        UncompleteRuleGen(U,C-el,D,ReducedDismatrix)
    END
```

最后，算法可形式地表示如下：

```
算法'UnCompleteRules——一个面向属性的，应用Rough集的分类规则算法
输入
1. 一个待处理的对象集U
2. 条件属性C,C={ai,1<=i<=n},决定属性D
3. 阈值T
4. 概念层次树Hi,Hi是一个在属性ai上的概念层次树,1≤i≤n
5. 规则应满足的最小正确率P
输出
    不完整分类规则
步骤
step1 对每个属性ai do{
    WHILE T<=|Ua|DO
        IF 在ai的概念层次树上没有更高的概念,就移去ai
        ELSE 替代每个对象在ai下的值为比它高一层的概念值
}
step2合并属性值全部相同的对象为一个对象,修改属性Vote的值
step3 UncompleteRuleGen(U,C,D,Dismatrix)
step4去掉规则正确率小于P的分类规则
```

讨论 此算法可在以下几个方面进一步改进：

1)以相反的顺序遍历格。遍历格空间时我们改为从下面的结点而不是从上面的结点开始，当从下至上时，越来越多的属性被考虑。于是生成规则的支持率(支持规则的对象数与信息系统中所有对象数的比)也就越来越低。因此进一步可在格上的结点设置支持率的阈值，只有当生成的规则的支持率大于这个阈值时，才允许继续在格空间上爬以生成更多的规则。使用这种方法，可以避免遍历到许多非常消耗时间的结点，这是因为分明矩阵中的属性越多，相应生成分类规则所消耗的时间也就越多。

2)使用遗传算法生成分明矩阵[4]，生成分明矩阵是算法中最耗时间之一。可利用遗传算法来生成近似的分明矩阵，其生成规则的效果比纯数学计算出来的分明矩阵稍差，但速度要快得多。

参考文献

1 Pawlak Z. Rough Sets: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publisher, 1991

2 Skowron A, Rauszer C. The Discernibility Matrices and Functions in Information Systems. In, Slowinski R, ed. Intelligent Decision Support: Handbook of Applications and Advances of Rough Theory. 1992

3 Han J, et al. Kowledge Discovery in Databases: An Attribute_ oriented Approach. In: Proc of the 18th VLDB Conf. Vancourer, B. C., Canada, 1992. 340～355

4 Wrobleski J. Finding minimal reducts using genetic algorithm: [Techical report]. Univesity of WARsaw, 1995

5 Kryszkiewicz M. Rybinski H. An Algoritm Searching for Minimal Acceptable Solutions of a Monotonic Boolean Function. Archiwum Informatyki Stosowanej Teoretycznej, 1992

6 Hu Xiaohua, Han N C J. An Attribute-Orientd Rough Set Approach for Konwledge Discovery in Datebases RSKD93. 91～96

7 Kryszkiewicz M, Rybinski H. Finding Reducts in Composed Information Systems. RSKD93. 260～273