

信息系统的规则提取研究

何 群

(南京工业大学 电子与信息工程学院, 江苏 南京 210009)

摘 要: 指出求信息系统的最简规则是一个二维极简化的问题, 将行分辨且列组合具有最大强度的属性值定义为粒极值, 以其逻辑关联的程度逐级进行粒运算, 得最简粒(规则)解集. 从理论上统一了求解信息系统的算法并用各类实例以证明.

关键词: 粒极值; 粒计算; 粗糙集; 逻辑无关; 决策规则

中图分类号: TP391

文献标识码: A

文章编号: 1000-7180(2012)07-0026-06

Research on Extracting Rules in Information System

HE Qun

(College of Electronics and Information Engineering, Nanjing University of Technology, Nanjing 210009, China)

Abstract: Pointing out that method for solving the information system is a minimization problem of two-dimensional, attribute value that the defining on row and compositing on column intensity is maxima was defined as the limiting value of granular, logic operation of granular comput on the preferred degree of logic independence with it stage by stage, finally minimization granular (rules) sets is got. Unifies the algorithm solving for the information system on theory and the experiments show that it is efficient.

Key words: limiting value of granular; granular computing; rough set; logic independence; decision rules

1 引言

从信息表(系统)中提取规则是信息处理的关键环节. 当前基于粗糙集理论^[1]、粒计算^[2]所提出的主要算法是以属性、基本粒为基本单位进行的约简, 有以下问题: (1) 由于基本单位选择过大而约去对规则最简化有效的属性值; (2) 计算和判定的数据集非动态改变, 不能准确及时地提出已分辨出的规则并消除该规则所及信息对后续提取规则的影响; (3) 多以概率等模糊算法进行先优处理为前提, 使得最后在处理不具有决定性的非优属性时可选的域扩大, 不易获取最佳规则^[3].

针对上述问题, 本文结合粗糙集和粒计算两种理论^[4]: (1) 将属性值定为基本单位. 以粗糙集的不可分辨关系求出属性值的行向分辨强度值、以粒计算的可组合关系求出属性值的列向组合强度值, 用

这两种强度值判定出当前必选的属性值(粒极值), 再以粒极值运算确定极值粒(规则); (2) 在提取规则后, 将其所约简行集的数据删除且重计组合强度, 以去除这些行集对后续确定粒极值的影响. 在新的数据集中再计算对应的粒极值; (3) 算法是以行、列极化的方式在准确数据支持下实现的: 从全集(当前表)范围内, 先确定必选的行集, 即极值行; 在该行集中以行为单位, 确定必选的列集, 即粒极值, 该行中的各个粒极值之“与”即为规则. 算法的核心是只在当前表中确定必选且其可选范围最小的属性值组成规则, 是一个基于粒极值递进逐级提取规则的过程. 得出以下结论:

(1) 提出粒极值和逻辑相关的概念, 将属性值的行分辨和列组合统一以逻辑相关程度进行择优确定粒极值.

(2) 以极化的方式对当前表动态地行、列分级,

收稿日期: 2011-11-03; 修回日期: 2011-12-18

基金项目: 江苏省建筑业科研计划项目(JGJH2008-17)

可始终选择关联最少的极值粒首先提出,并且及时地消除其对后续处理的关联。

(3) 以准确的数据直接从信息表中获得最简规则,不涉及属性约简的中简步骤及规则可信度^[2]等概率问题,统一了各类信息表(相容、不相容;全核、有核、无核)的规则提取算法。

2 算法理论和实现

对信息表^[4] $S = (U, A_n \cup D, V, F)$ 的总体结构分析可知,要由 $\{A_n\}$ 分辨出 D 的最简式,是一个二维(行和列)求属性极值点的问题:在行方面,以粗集不可分辨关系^[3],求这个属性值 a_{ij} 相对 $a_{kj} \in A_j | (a_{ij} \neq a_{kj})$ 可分辨出 $d_i \in D | (d_i \neq d_k)$ 的分辨强度;在列方面,以粒公式^[2],求出粒 $(A_j \in \{A_n\}, v = a_{ij})$ 与其它粒 $(A_k \in \{A_n\}, v = a_{ik})$ 相对于 d_i 的可组合强度。这样,每个属性值在行向有一个分辨强度值而列向有一个组合强度值,这两个值的大小决定了该属性值的重要程度。

算法1 (求属性值的行分辨强度)根据定义1.

for $(\forall i \in U | d = d_i) \{$

for $(\forall j \in U | d_j \neq d_i) \{$

$$S_{ij} = \sum (a_{ik} \odot a_{jk})$$

$$RD_{a_{ik}} = \sum (a_{ik} \oplus a_{jk})^{S_{ij}}$$

$$N_{Ri} = \max d\{RD\{a_{ik}\}\}$$

称 $RD(a_{ik})$ 为属性值 $a_{ik} (k=1, 2, \dots, n)$ 的行分辨强度,每个属性值对应一个,称 N_{Ri} 为 i 行中的最大 RD 域长,每行对应一个。

$RD(a_{ik})$ 反应了 i 行中的每个属性值对 d_i 的分辨强度,其高域的值越大(逻辑真值个数越多)则对应的属性值对 d_i 的分辨能力越强; N_{Ri} 反应了 i 行的行分辨强度级别。

算法2 (求属性值的列组合强度)根据定义2.

for $(\forall i \in U | d = d_i \in U) \{$

for $(\forall j \in U | d_j = d_i) \{$

$$S_{ij} = \sum (a_{ik} \odot a_{jk})$$

$$CD_{a_{ik}} = \sum (a_{ik} \odot a_{jk})^{S_{ij}}$$

$$N_{Ci} = \max d\{CD\{a_{ik}\}\}$$

称 $CD(a_{ik})$ 为属性值 $a_{ik} (k=1, 2, \dots, n)$ 的列组合强度,每个属性值对应一个,称 N_{Ci} 为 i 行中的最大 CD 域长,每行对应一个。

$CD(a_{ik})$ 反应了 i 行中的每个属性值对 d_i 的分辨时,从列方向可组合的能力,其高域的值越大(逻辑

真值个数越多)则对应的属性值对 d_i 的列向组合能力越强; N_{Ci} 反应了 i 行的列组合强度级别。

每个属性值 a_{ij} 都与其 $RD(a_{ij})$ 、 $CD(a_{ij})$ 相对应,每行中的各属性值 $\{a_{ik}\} (k=1, 2, \dots, n)$ 之间可根据两值的比较以确定其在该行中综合分辨力的强弱;而行与行之间的重要程度则是由该行的 N_{Ri} 和 N_{Ci} 所决定的。

对 S 的规则提取是在以上数据支持下动态分级在进行,前级相对后级是有关联的,或直接约简后级或对后级有引导作用。所以首先对行进行分级,即求极值行集;其次分别在每个极值行中对列进行分级,即求粒极值。

定义1 (极值行集) 设 $i (i=1, 2, \dots, m)$ 行中已求得 N_{Ri} 和 N_{Gi} 。若 $N_{Ri} > N_{Gi}$, 则 RD_i 冗余度大;若 $N_{Gi} > N_{Ri}$, 则 CD_i 冗余度大;而 $N_{Ri} = N_{Gi}$, RD_i 和 CD_i 两者没有冗余。所以 $|N_{Ri} - N_{Gi}|$ 的大小即为 i 行的重要程度(0 为最高级)。将当前表中所有最高一级对应行所组成的行集称为极值行集,记为 $\{i\}$ 。

定义2 (逻辑无关) 设 $\{L_m\} (m=1, 2, \dots, k)$ 是 n 维逻辑变量组,若从 $\{L_m\} - L_i$ 中任选若干个逻辑变量的“或”运算结果都不能完全覆盖 L_i 的全部真值,则 L_i 是逻辑无关变量,反之 L_i 为逻辑相关变量。

定义3 (极大逻辑无关组) 设一个 n 维逻辑变量组 $\{L_r\}$ 是 $\{L_m\}$ 的一个部分组。如果 $L_1, L_2, \dots, L_r \in \{L_r\}$ 逻辑无关且变量个数 r 最小

(1) 若 $m=r$, 称 $\{L_r\}$ 为 $\{L_m\}$ 极大逻辑无关组,记为 $\{L\}_=$;

(2) 若 $m > r$, 从 $\{L_m\}$ 中任选一个变量,都能被 $\{L_r\}$ 中变量的“或”运算结果所覆盖,则称 $\{L_r\}$ 为 $\{L_m\}$ 的一个准极大逻辑无关组,记为 $\{L\}_>$;

定义4 (粒极值) 设极值行集 $\{i\}$ 和对应的 N_{Ri} 、 N_{Gi} 和 $RD(a_{ik})$ 、 $CD(a_{ik}) | i \in \{i\}, (k=1, 2, \dots, n)$ 。以 i 行为单位,将 $RD(a_{ik})$ 的最高域(N_{Ri})非0且 $CD(a_{ik})$ 的最高域(N_{Gi})非0 称为最高级逻辑变量组(这个级别根据 N_{Ri} 和 N_{Gi} 的递减类推)。每级中 $(\{RD(a_{iu})\}, \{CD(a_{iu})\}) | u \subseteq k$ 是两个逻辑变量组。以 $\{RD(a_{iu})\}$ 为首、 $\{CD(a_{iu})\}$ 为次组成一个逻辑关联程度的判定序列,可总体分为两个部分: $\{RD(a_{iu})\}$ 是 $\{L\}_=$, 可获粒极值; $\{RD(a_{iu})\}$ 是 $\{L\}_>$, 则只可获准粒极值。具体方法如下:

(1) 若 $\{RD(a_{iu})\}$ 是 $\{L\}_=$, 则 $RD(a_{is}) | s \in u$ 所对应的 a_{is} 称为粒极值,记为 $G_1(A_s, a_{is})$ 或 G_1 ;

(2) 若 $\{RD(a_{iu})\}$ 是 $\{L\}_>$ 且唯一, 则 $RD(a_{is}) | s$

$\in v$ 所对应的 a_{is} 称为准粒极值, 记为 $g_1(A_s, a_{is})$ 或 g_1 ;

(3) 若 $\{RD(a_{iv})\}$ 是 $\{L\}_>$ 且不唯一, 而 $\{CD(a_{iv})\}$ 是 $\{L\}_=$ 、 $\{L\}_>$ 且唯一, 则 $RD(a_{is})|s \in v$ 所对应的 a_{is} 也称为 g_1 .

(4) 若 $\{RD(a_{iv})\}$ 和 $\{CD(a_{iv})\}$ 都是 $\{L\}_>$ 且都不唯一, 即本级等价, 则要按上述顺序进行下一域级的比较.

根据粒计算的定义^[2], 同一行中高一级的 G_1 对后一级的 G_1 是逻辑关联的, 前级粒极值 G_1 产生后, 用其的 RD 对 i 行内各属性值的 RD 分别进行一次强度覆盖(用符号 ∞ 示^[4]), 去除后续级中与之等价的逻辑值; 用 G_1 的 CD 对 i 行内各属性值的 CD 分别进行一次组合同取(用符号 $\wedge$ 示^[4]), 去除后续级中不能与之组合的逻辑值. 之后, 在 i 行中重复上述过程进行下一级 G_1 的确定, 顺序获取各级的 G_1 , 直至 i 行的 RD 为空. 将各级的 G_1 之“与”的运算结果 $G(\{A_v\}, \{a_{iv}\}) \wedge = \{G_1(A_s, a_{is})\}$, 称为极值粒(简记为 G); 类似地, 将 $g(\{A_v\}, \{a_{iv}\}) \wedge = \{g_1(A_s, a_{is})\}$, 称为准极值粒(简记为 g).

算法 3 (求粒极值和极值粒) 根据定义 4.

```
RG( $i, N_R, N_G, G$ )
for( $RD_i \neq \Phi; N_R --$ ) {
 $a_{is} = \text{Limit } V\{RD(a_{ik})\}$ 
 $G_1(A_s, a_{is});$ 
 $RD_i \infty = RD(a_{is}); CD_i \wedge = CD(a_{is})$ 
 $G(\{A_g\}, \{a_{ig}\}) \wedge = G_1(A_s, a_{is})\}$ 
```

最简规则即为 G , 其核心是用最少的 G_1 组合在覆盖本行 $RD(a_{ik})$ 全部值的前提下使可约简的行数最大. 下面分析当从当前 $\{i\}$ 中求出 G 之后的处理问题, 在属性值 a_{ij} 的两个参数 RD 和 CD 中, RD 具有全局性逻辑关系, 是一次计算完成且不可改变的, 而 CD 是局部性逻辑关系, 是随 G 的提取而可变化的, 即在对 $d_i \in D$ 中提取极值粒后, 则 $d_i \in D$ 域内的行数减少. 只有重新计算 CD 才会使下一级的分辨准确. 所以在 $\{i\}$ 同级环境下, 对 $d_i \in D$ 域提取全部 G 后, 要在去除 G 约简的行集后, 并对 $d_i \in D$ 域中重新计算的 CD . 而对于逻辑相关的情况, 即 $d_i \in D$ 域非空且没有 G (或已提完), 则要根据定义 3 按准极大逻辑无关组从高至低的级别进行判定, 但因其各准极值粒 g 逻辑相关, 所以在同级中对 $d_i \in D$ 域一次只提一个 g . 综上所述, 算法的思路:

(1) 按定义 1 选择当前表中最高级的极值行集 $\{i\}$; 按行的分辨强度递降分级;

(2) 按定义 4 在 $\{i\}$ 中计算出 G_1 集及 G ; 以行为单位, 按属性值的行、列强度分级递降判定得至各级 G_1 或 g_1 . 最终由各级 G_1 或 g_1 获得本行的 G 或 g .

(3) 提取所得全部 G 或只提取一个 g (无 G 时): 逻辑完全无关的 G 全提出, 而逻辑相关的 g 则只提一个.

(4) 从当前行集中去除 G (或 g) 所及行, 并对 G (或 g) 所及 $d_i \in D$ 重新计算 CD . 重复上述过程, 直至 S 表的行集为空.

输入: 原始样表 $\{U, An, D\}$;

输出: 最简规则表 $\{Ur, Anr, Dr\}$

$RD(\{U, An, D\}, \{RD\})$ // 求原始表 RD (算法 1)

for($\{U, An, D\} \neq \Phi$) {

$CD(\{U, An, D\}, \{CD\})$ // 求当前表 CD (算法 2)

for($N_R, N_G; N_R -- (N_G --)$) // 2 级循环

$\{i\} + = i | (N_R; N_G = 1, 2, \dots, m)$ // 求极值行集

$RG(\forall i \in \{i\}, N_R, N_G, G)$ // 求极值粒 (算法 3)

$\{Ur, Anr, Dr\} + = G$ // 存入一个规则

$U - = \{G\}$ // 从表中删除 G 所约简的行集

3 实例分析

3.1 决策表的综合分析

对照参考文献[4]表 2 进行综合分析.

$D=1$: 最高级行只有 $(N_R = N_G = 3) | U=4$, 所以首选行为 4, 在该行中, $N_R = 3$ 且 $N_G = 3$ (最高域) 非 0 的属性值有 2 个: $(b=1); (c=3)$, 两者的 RD 都为 1 (等价), 但是 $(c=3)$ 的 $CD=3$, 逻辑无关; 而 $(b=1)$ 的 $CD=2$, 逻辑相关. 所以 $c=3$ 为 g_1 (定义 4. (3)). 由该 g_1 的 RD 和 CD 对该行各属性值的 RD 和 CD 分别覆盖和同取后, 类上述判定 $(a=1)$ 为第 2 级的 g_1 , 至此得 $g: (c=3) \wedge (a=1)$, 约简了 $U=3, 4, 5$.

$D=2: (N_R = N_G)$ 的有 $U=9, 11, 14, 15$. 同样方法, 对 $U=15, N_R=3$ (最高域) 只有 $(c=2)$ 的 $RD (=3)$ 逻辑无关; 经其对本行内 RD 、 CD 进行覆盖、同取后, $N_R=2$ (次高域) 只有 $(e=1)$ 必选, 得 $G: (c=2) \wedge (e=1)$, 约简 $U=14, 15$; 对 $U=11$, 得 $G: (e=4) \wedge (a=3)$, 约简 $U=10, 11$; 对 $U=9$, 得 $g: (a=2) \wedge (b=3)$, 约简 $U=9, 12$.

$D=3$: 只有 $(N_R = N_G) | U=19$ (首选), 在该行中, $N_R=3$ 且 $N_G=3$ (最高域) 非 0 的属性值只有 2 个: $(a=4); (b=4)$, 两者的 RD 和 CD 前两级是等价的, 而第 3 级时 $(b=4)$ 的 $RD (=1F)$ 逻辑无关, 而 $(a=4)$ 的 $RD (=17)$ 是逻辑相关的, 得 $G: (b=4)$, 约简了 $U=17, 18, 19$.

本轮结束后,对 $D=1,3$ 域分别提出 1 个 g 和 1 个 G ,而对 $D=2$ 域提出 2 个 G 及 1 个 g . 之后重新计各域的 CD ,在后续轮中,对 $D=1$ 顺序提得: $(a=2) \wedge (b=2)$ 、 $(b=3) \wedge (c=1)$;对 $D=2$: $(b=3) \wedge (c=3)$ 、 $(a=2) \wedge (b=1)$;对 $D=3$: $(a=4) \wedge (e=3)$ 、 $(a=3) \wedge (e=2)$. 则求得本表的 11 个最简规则(21 个属性值)涉及 4 个属性 $\{a, b, c, e\}$.

显然,当多个 g_1 等价时,要根据其涉及已确定的属性列(由前轮各级的 G 所选定)进行选择,这样的处理往往可使最终的全部规则所及属性个数集中而达到最少的目的,实质上是一个动态属性约简的过程. 例如 $U=20$,首先确定的是 $G_1(e=2)$,用 G_1 的 RD 和 CD 覆盖和同取后, a, b, c, d 强度都为 $(RD=0.0.0.D; CD=0.0.0.0)$,是 4 个等价的 g_1 ,此时,要选已被前轮确定的属性列,如选 b (或 c)列而非 d 列. 按此方法,本例最终的全部规则都不涉及属性 d 列,即约简了 d .

算法的基本思路是始终选择关联最少 G (或 g) 先提取,再去除其对后续提 G 的影响,是一个动态极化数据集、递级求解的过程. 这就是为什么目前很多算法只在最初的原始数据上计算、判定很难得到最优结论的原因之一. 以下是对不同类型 S 的实例分析.

3.2 全核属性决策表分析

逻辑化简是粗糙集属性值约简的一个特例^[4].

设 $f(abcd) = \sum m(0, 2, 3, 4, 5, 7, 8, 9, A, D, E, F)$ 其卡诺图如表 1, () 中的值示最小项序号,首轮求得的各属性值的强度表,见参文^[4]表 3(左),对 $f=1$,所有 m 的 $N_R = N_G = 3$,即所有行都是极值行. 在表 3(左)中标示的 $\square$ 和 $_$ 均为 G_1 . 以 G_1 进行最高级覆盖、同取后, $m(0, 2, 5, 7, 8, A, D, F)$ 和 $m(3, 4, 9, E)$ 都只有同级别的 g_1 . 可任选一个,若选 $m0$ 提得 g (规则):

$$r_1: ((b=0) \wedge (d=0) \rightarrow (f=1))_{\{0,2,8,A\}}$$

对余下行集重计 CD ,见参文^[4]表 3(右). 以 G_1 进行最高级覆盖、同取后,全部 m 都是 g_1 ,即在 RD 级等价,需要比较 CD 级,因 $m(5, 7, D, F)$ 各自对应的 3 个属性值的 CD 都非 0 且互为逻辑相关;而 $m(3, 4, 9, E)$ 各自对应的 2 个属性值的 CD 中 1 个为 0.0.0(逻辑相关)、另一个则为 1.0.0(逻辑无关). 所以分别选 $m3, m4, m9, mE$. 得规则:

$$r_2: ((a=0) \wedge (c=1) \wedge (d=1) \rightarrow (f=1))_{\{3,7\}};$$

$$r_3: ((a=0) \wedge (b=1) \wedge (c=0) \rightarrow (f=1))_{\{4,5\}};$$

$$r_4: ((a=1) \wedge (c=0) \wedge (d=1) \rightarrow (f=1))_{\{9,D\}};$$

$$r_5: ((a=1) \wedge (b=1) \wedge (c=1) \rightarrow (f=1))_{\{E,F\}}.$$

至此, $f(a, b, c, d)=1$ 的最简式全部提出. 若用卡诺图的化简准则对 $f=1$ 化简,则会多一个冗余的逻辑式 $r_0: ((b=1) \wedge (d=1) \rightarrow (f=1))_{\{5,7,D,F\}}$,其原因是没有及时去除已提取 g 的 CD 对后续组合的影响.

对等价的 g_1 ,选择是任意的,若首选为 $m3$ (见表 1)且先化简 $m3$ 和 $m7$. 重计 CD 后,由于 $m2$ 与 $m3$ 的 CD 已无关系,所以只有 $m2$ 为必选,由它化简 $m2, m0, m8$ 和 mA . 重计 CD 后则必选 $m4, m9$ 和 mE ,所得的规则个数和逻辑值个数同上,即与上述首选 $m0$ 的情况等价. 显然,首选任一个 m ,都可以得到等价结果,这里起决定因素的是 CD 的动态改变.

表 1 参文献[5]的卡诺图

$cd \backslash ab$	00	01	11	10
00	1/(0)	1/(4)	0/(C)	1/(8)
01	0/(1)	1/(5)	1/(D)	1/(9)
11	1/(3)	1/(7)	1/(F)	0/(B)
10	1/(2)	0/(6)	1/(E)	1/(A)

3.3 部分核属性的决策表分析

文献[5]中决策表 1 如表 2(左). 按算法首轮求得的各属性值的行、列综合强度表,见表 2(右).

表 2 属性值与首轮粒极值对应表

U	a	b	c	d	e
1	0 $\square$ 1 0 1 0 $\square$	$\square$ 0.2.0 0	1.3.0 0	1.1.0 1	0.2.0 0
2	1 0 $\square$ 0 $\square$ 1 $\square$	0.0.1.1 0	0.0.0.0 0	$\square$ 1.0.1.1 1	0.0.1.1 0
3	1 0 0 0 0 2	0.0.1.1 0.0	0.0.1.0 1.0	0.1.0.1 0.0	0.0.1.1 1.0
4	0 $\square$ 1 1 $\square$ 2	0.0.1.1 0.0	$\square$ 0.1.0.0 1.0	0.1.0.1 0.0	0.0.1.1 1.0
5	0 0 1 1 1 3	0.0.1.1 0.0.0	0.0.2.0 1.0.0	0.0.3.1 1.0.0	0.0.1.1 1.0.0
6	1 0 $\square$ 0 $\square$ 3	0.0.1.0 0.0.0	0.0.0.0 1.0.0	$\square$ 1.1.0.0 1.0.0	0.0.1.0 1.0.0

因为 $(Nr6; Nc6)=4:3$ 是级别最高,所以首选 $U=6$. 因 $RD(c=1) | U=6$ 逻辑无关,所以 $G_1: (c=1)$,用 G_1 的 $RD(c=1), CD(c=1)$ 对 RD_6, CD_6 进

和行强度覆盖、同取后,只有 $RD(e=1)=(0,0,1,0)$ 、 $CD(e=1)=(1,0,0)$ 最强,即为第 2 个 $G_1(e=1)$. 再用 $G_1(e=1)$ 对 RD_6 和 CD_6 进和行强度覆盖、同取后, $RD_6=0$. 最终得 G (规则):

$$r_1: ((c=1) \wedge (e=1) \rightarrow (g=3))_{(5,6)}$$

r_1 约简了 $U=5,6$ 两行,在表 2(左)中($U=6$)用 $\square$ 表示. 同理,共涉及 $\{b, c, e\}$ 3 个属性(其中 c 和 e 是核属性)直接得到了表 1 的 4 条规则(共 7 个属性值). 这个结论相对于文献[5] 中选 $\{a, c, e\}$ 所得属性约简及规则(表 2)更精准.

3.4 无核属性的决策表分析

文献[6]中的表 1 是一个无核属性的决策表. 首轮求得各属性值的分辨、组合强度见表 3(上:属性值;中: RD ;下: CD).

表 3 首轮属性值与粒极值对应表

U	$a1$	$a2$	$a3$	$a4$	$a5$	$a6$	d
1	3	$\square$ 2	1	2	$\square$ 1	0	0
	0.7.1.0	$\square$ 1.C.0.0	0.4.1.0	1.8.0.0	$\square$ 0.3.1.0	0.B.1.0	
	0.0.0.0	$\square$ 1.0.0.0	1.0.0.0	0.0.0.0	$\square$ 1.0.0.0	1.0.0.0	
2	4	2	1	1	1	0	0
	1.0.B.0	1.1.8.0	0.1.4.0	0.1.F.0	0.0.7.0	0.0.F.0	
	0.0.0.0	1.0.0.0	1.0.0.0	0.0.0.0	1.0.0.0	1.0.0.0	
3	3	$\square$ 3	1	3	1	1	1
	0.7.3	$\square$ 1.7.3	0.0.3	1.7.3	0.6.1	1.1.2	
	1.0.0.0	$\square$ 1.0.0.0	1.0.0.0	0.0.0.0	1.0.0.0	0.0.0.0	
4	3	3	1	1	1	0	1
	1.0.1.3	3.0.1.3	0.0.1.0	2.0.1.3	0.0.0.3	0.0.0.3	
	1.0.0.0	1.0.0.0	1.0.0.0	0.0.0.0	1.0.0.0	0.0.0.0	
5	4	1	$\square$ 2	2	1	0	2
	2.1.7	3.1.7	$\square$ 3.1.7	1.1.1	0.0.6	0.0.7	
	1.0.0	0.0.0	$\square$ 1.0.0	1.0.0	0.0.0	0.0.0	
6	4	2	2	2	2	1	2
	3.0.2.1	0.0.0.1	3.0.3.1	0.0.1.1	0.0.3.1	0.0.3.0	
	0.0.0	1.0.0	0.0.0	1.0.0	1.0.0	0.0.0	
7	2	2	$\square$ 1	2	2	1	3
	1.1.3.3	0.0.1.3	$\square$ 1.0.0.1	0.0.3.2	$\square$ 0.1.3.3	0.1.2.3	
	0.0.0.0	1.0.0.0	$\square$ 1.0.0.0	1.0.0.0	$\square$ 1.0.0.0	1.0.0.0	
8	1	2	1	2	2	1	3
	1.1.3.3	0.0.1.3	1.0.0.3	0.0.3.2	0.1.3.3	0.1.2.3	
	0.0.0.0	1.0.0.0	1.0.0.0	1.0.0.0	1.0.0.0	1.0.0.0	

$(Nr: Nc)=1$ 最高级别有 $U=1,2,4,5,7,8$. 以 $U=5$ 为例, $RD(a3=2)$ 和 $RD(a2=1)$ 等价,但 $CD(a3=2)$ 逻辑无关,即 $g_1(a3=2)$,用 g_1 的 $RD(a3=2)$ 、 $CD(a3=2)$ 对 RD_5 和 CD_5 进和行强度覆盖、同取后, $RD_5=0$. 最终得 g (规则):

$$r_1: ((a3=2) \rightarrow (d=2))_{(5,6)}$$

r_1 约简了 $U=5,6$ 两行,在表 3 中用 $\square$ ($U=5$) 表示. 同理,共涉及 3 个属性 $\{a2, a3, a5\}$ 直接得到表 3 的 4 条规则(共 6 个属性值). 由于剩余属性 $\{a1, a4, a6\}$ 也可对 d 进行完整的分辨, S 中没有哪一个属性是必选的,所以该表是无核属性的. 但是,只有 $\{a2, a3, a5\}$ 可使涉及属性最少且规则最简. 如文献[6]中提及的 2 条确定性决策规则可用表 3 中 $((a2=2) \wedge (a5=1) \rightarrow (d=0))_{(1,2)}$ 的一条规则所取代.

3.5 不相容决策表分析

文献[7]中的表 1 是一个不相容的决策表($U=7,8$ 相同,去除 $U=7$),其中 $U=3$ 与 $U=4, U5$ 与 $U6, U8$ 与 $U9$ 不相容. 由逻辑关系式的定义,每行是一个逻辑最小项. 如 $U=3$ 与 $U=4$ 两项的逻辑关系式为: $(c1=1) \wedge (c2=0) \wedge (c3=1) \wedge (c4=1) = ((d=0) \vee (d=1))$. 左式是两个相同的逻辑式相“或”后为一个,而右式可根据不相容部分用对应值替代,比如上右式可记为 $((d=0) \vee (d=1)) = (d=01)$,这样就将不相容表替代为相容表. 对文献[7]中的表 1 替代为表 4(左)进行规则提取,其首轮求得各属性值的行、列强度,见表 4(右).

从 $U=34$ 求得: $(c1=1) \rightarrow (d=01))_{(34,56,89)} \rightarrow ((d=0) \vee (d=1))_{(3,4,5,6,8,9)}$; 从 $U=1$ 求得: $(c3=2) \rightarrow (d=0))_{(1)}$; 从 $U=2$ 求得: $(c2=2) \rightarrow (d=1))_{(2)}$. 表 4(左)中所示. 在已给条件下所得规则最简,涉及属性 $\{c1, c2, c3\}$. 若只考虑属性数量最少,可将表 4 中 $U=2$ 用式 $((c1=0) \wedge (c3=1) \rightarrow (d=1))_{(2)}$ 代替,这里增加了一个属性值但所涉及属性减至 $\{c1, c3\}$,与文献[7]中算法 4 的结论一至. 实际上,若只考虑属性数量最少,则文献[7]的 3 个属性 $\{c1, c2, c3\}$ 中任取 2 个属性的组合都是该 S 的约简. 但这只是以增加规则属性值的个数达到涉及属性个数减少的等价. 显然,属性约简算法往往只注重涉及属性个数而没考虑所得规则的简洁. 本算法是在规则最简的前提下尽量使属性个数最少(如 3.1).

另外,对无决策属性的 S 进行属性约简是本算法的子集,只对行分辨强度作判定,参见文献[4] 中 3.4.

表4 首轮属性值与粒极值对应表

U	c1	c2	c3	c4	d	c1	c2	c3	c4
1	0	1	2	1	0	3.1	4.1	7.1	0.0
						0	0	0	0
2	0	2	1	1	1	3.1	7.1	4.1	0.0
						0	0	0	0
34	1	0	1	1	01	1.1	1.1	0.1	0.0
						1.1.0	0.0.0	1.0.0	1.1.0
56	1	1	1	1	01	3.0	1.0	2.0	0.0
						3.0.0	2.0.0	1.0.0	3.0.0
89	1	1	0	1	01	1.1	0.1	1.1	0.0
						1.1.0	1.0.0	0.0.0	1.1.0

4 结束语

在文献[4]的基础上,增加了确定极值行集的算法、将以大值(二进制数中为1的个数最多)判定粒极值修改为以逻辑关联判定:

(1) 以属性值的RD和CD的域长首先按级定出极值行集,使得先处理最关键的极值行,在极值行中提取规则后,与之相关的非极值行都被约简,减少了运算复杂度。

(2) 根据解线性方程原理,提出逻辑相关的概念.将以判定二进制为1个数的多少修改为以逻辑变量相关程度来确定粒极值,使计算原理系统化从

而形成完整、准确的判定序列。

(3) 将相容表和不相容表统一处理,即单因多果的情况替代为单因单果情况处理后再展开为单因多果的规则.统一各类信息表的规则提取算法。

(4) 对大量各类信息表运算结果(本文只列出部分)都证明了所提规则最简而并非是一个NP-Hard问题^[3],算法具有较高的理论和实用价值。

参考文献

- [1] 钱锋. 结合模糊集理论的粗糙集属性约简算法[J]. 计算机应用研究, 2007, 24(11): 93-95.
- [2] 卓丽霞. 粒计算规则生成模型[J]. 重庆工学院学报自然科学, 2008, 22(9): 149-153.
- [3] 刘丰年. 基于分明矩阵的序贯属性约简算法[J]. 微电子学与计算机, 2009, 26(7): 95-97.
- [4] 何群. 基于G逻辑的最简规则提取算法[J]. 微电子学与计算机, 2010, 27(10): 91-95.
- [5] 富震. 基于自适应遗传约简算法的雷达故障诊断[J]. 控制与决策, 2010, 24(11): 93-95.
- [6] 李莉. 一种否定高信度规则的属性查找算法[J]. 计算机工程, 2008, 34(6): 65-69.
- [7] 刘海涛. 不相容决策表中一种新的属性约简算法[J]. 计算机工程与应用, 2011, 47(2): 124-126.

作者简介

何群 男, (1955-), 硕士, 高工, 研究生导师. 研究方向为数据挖掘、粗糙集、粒计算。

(上接第25页)

参考文献:

- [1] Pawlak Z, Skowron A. Rudiments of rough sets[J]. Information Science, 2007, 117(1): 3-37.
- [2] 岳晓冬, 苗夺谦, 钟才明. 基于粗糙性度量的彩色图像分割方法[J]. 自动化学报, 2010, 36(6): 807-816.
- [3] 胡清华, 于达仁, 谢宗霞. 基于邻域粒化和粗糙逼近的数值属性约简[J]. 软件学报, 2008, 19(3): 640-649.
- [4] 崔巍, 徐章艳. 一种新的信息系统启发式属性约简算法[J]. 微电子学与计算机, 2010, 27(9): 80-85.
- [5] Hu X H, Cercone N. Learning in relational databases: A rough set approach[J]. Computational Intelligence, 1995, 11(2): 323-337.
- [6] 杨明. 一种基于改进差别矩阵的核增量式更新算法[J]. 计算机学报, 2006, 29(3): 407-413.
- [7] 葛浩, 杨传健, 李龙澍. 基于可分辨矩阵的快速求核算法[J]. 计算机工程与设计, 2009, 30(5): 1201-1203.
- [8] 程玉胜, 张佑生, 胡学钢. 两类决策系统中规则获取的联合决策矩阵算法[J]. 系统工程理论与实践, 2008, 28(6): 137-142.

作者简介:

胡秦斌 男, (1974-), 硕士, 讲师. 研究方向为数据挖掘、计算智能。