

不完备信息系统中知识获取算法^{*})

顾沈明^{1,2} 吴伟志² 高 济³

(浙江大学软件学院 杭州 310027)¹ (浙江海洋学院信息学院 浙江舟山 316004)²

(浙江大学人工智能研究所 杭州 310027)³

摘 要 粗糙集理论是一种新的处理模糊和不确定知识的软计算工具。应用粗糙集理论,可以将隐藏在系统的知识能够以决策规则的形式表达出来。根据粗糙集上下近似的概念,决策规则能够分成确定性规则和可能性规则两种。本文将介绍从不完备信息系统中知识获取的算法,通过这些算法能够从不完备决策表中生成一种确定性的规则和两种可能性的规则,同时也介绍了不完备决策表中描述约简的算法。

关键词 粗糙集,决策规则,不完备信息系统,算法

Algorithms for Knowledge Acquisition in Incomplete Information System

GU Shen-Ming^{1,2} WU Wei-Zhi² GAO Ji³

(Software College, Zhejiang University, Hangzhou 310027)¹ (Information College, Zhejiang Ocean University, Zhoushan 316004)²

(Institute of Artificial Intelligence, Zhejiang University, Hangzhou 310027)³

Abstract Rough Set theory is emerging as a powerful tool for reasoning about data. Using Rough Set theory, knowledge hidden in Incomplete Information System may be unraveled and expressed in the form of decision rules. According to the lower and upper approximations, decision rules can be divided into certain and possible rules. Algorithms for knowledge acquisition in incomplete information systems are proposed. As the result, one type of "certain" and two types of "possible" decision rules are generated from incomplete decision tables. Algorithms for reduction of descriptors in such tables are also discussed.

Keywords Rough sets, Decision rules, Incomplete information systems, Algorithms

1 引言

20 世纪 80 年代初,波兰的 Pawlak 教授提出了粗糙集理论^[1],这是一种新的处理模糊和不确定知识的软计算工具。经过二十多年的研究和发展,粗糙集理论已经在机器学习、决策分析、智能控制、模式识别等方面取得了较为成功的应用^[2~11],并且越来越受到国际学术界的广泛关注。

粗糙集理论利用集合近似的思想,能够把隐藏在信息系统中的知识以决策规则的形式表达出来。但在实际应用中许多信息系统是不完备的,所以不完备信息系统是粗糙集理论的一个重要研究方向。近年来人们在不完备信息系统方面取得了一些研究成果^[12~15]。

传统的方法常常从决策表中提取出经过属性约简和值约简的规则,并试图找到最小约简的规则。其目的是在保持决策能力不变的前提下,删除多余的知识,以提供简洁而正确的决策规则。但是求最小约简是 NP-难问题^[2],特别在不完备信息系统中计算的复杂度将大大增加,从而使算法的效率较低,在一定程度上限制了粗糙集理论的广泛应用,因此寻求切实可行的粗糙集算法具有重要意义。本文采用描述的概念,运用描述近似的方法,研究了在不完备信息系统中知识获取的方法,并给出了具体的规则产生算法。

2 信息系统与决策表

定义 1 $S=(U,AT)$ 称为一个信息系统,其中:

(a) U 为有限非空对象集, $U=\{x_1, x_2, \dots, x_n\}$ 。

(b) AT 为有限非空属性集, $AT=\{a_1, a_2, \dots, a_m\}$ 。

(c) 任意的对象 $x \in U$, 对于任意属性 $a \in AT$, 对象 x 在属性 a 上的取值为 $a(x) \in V_a$, V_a 是属性 a 的值域。

定义 2 在 $S=(U,AT)$ 中,如果任意 $a(x)$ 是唯一确定的,则称 S 是完备的信息系统。否则称 S 是不完备的信息系统。

在不完备信息系统中,一些属性值是不确定的,比如有缺省值或知道部分值,这样属性值函数 $a(x)$ 可以定义为从 U 到 V_a 的幂集的一个集值映射。例如知道 $a(x)$ 不是 V_a 中 b, c 值,则 $a(x)$ 可表示为 $V_a - \{b, c\}$ 。

定义 3 $T=(U,AT \cup \{d\})$ 是一个决策表,其中:

(1) (U,AT) 是一个信息系统;

(2) $d \notin AT$, 是一个完备的属性,称为决策属性; AT 称为条件属性。

定义 4 在 $T=(U,AT \cup \{d\})$ 中,如果 (U,AT) 是完备的信息系统,则称 T 是完备的决策表;如果 (U,AT) 是不完备的信息系统,则称 T 是不完备的决策表。

^{*}) 本课题得到国家自然科学基金(60373078)和浙江省教育厅科研计划项目(20040538)资助。顾沈明 副教授,主要研究领域包括粗糙集、人工智能等。吴伟志 教授,博士,主要研究领域包括近似推理、粗糙集、随机集等。高 济 教授,博士生导师,主要研究领域包括人工智能、软件工程等。

例如,表1是一个不完备的决策表,其中 $U=\{x_1, x_2, \dots, x_7\}$, $AT=\{a, b\}$ 是条件属性集, d 是决策属性。 a, b 和 d 分别表示收缩压、舒张压和血压。 $V_a=V_b=V_d=\{H, N, L\}$, H, N 和 L 分别表示收缩压、舒张压和血压是高、正常和低。

表1 不完备信息决策表

U	a	b	d
x_1	$\{N\}$	$\{N\}$	N
x_2	$\{H, N\}$	$\{N\}$	H
x_3	$\{N\}$	$\{N, H\}$	N
x_4	$\{L\}$	$\{L\}$	L
x_5	$\{H\}$	$\{N, H\}$	H
x_6	$\{H, N\}$	$\{H\}$	H
x_7	$\{L, N\}$	$\{N\}$	L

3 描述

定义5 在信息系统 $S=(U, AT)$ 中, 任意 $a \in A, A \subseteq AT$, 任意 $v \in V_a$, 属性-值 (a, v) 称为 A 的基本元。所有的 A 的基本元或它们的“ $\wedge$ ”连接称为 A 描述。

设 t 是一个 A 描述, 则 t 中涉及的属性集记为 $A(t)$; t 中基本元的个数, 称为 t 的长度, 记为 $L(t)$; 如果基本元 (a, v) 在 t 中出现, 记为 $(a, v) \in t$ 。包含 A 描述 t 的对象集合称为 t 的支持集, 记为 $\|t\|$, 即 $\|t\| = \{x \in U | v \in a(x), (a, v) \in t\}$ 。

算法1 在信息系统 $S=(U, AT)$ 中, 求描述 t 的支持集 $\|t\|$ 。

- (1) 令 $\|t\| = \Phi$
- (2) 任取 $x \in U$
- (3) $B = A(t)$
- (4) 任取 $a \in B$, 设 $(a, v) \in t$
- (5) 若 $v \notin a(x)$, 则转(9)
- (6) $B = B - \{a\}$
- (7) 若 $B \neq \Phi$, 则转(4)
- (8) $\|t\| = \|t\| \cup \{x\}$
- (9) $U = U - \{x\}$
- (10) 若 $U = \Phi$, 则结束, 否则转(2)

算法1的时间复杂度是 $O(n \times \text{card}(A(t)))$, 其中 $n = \text{card}(U)$ 。若 $A(t) = AT$, 则时间复杂度是 $O(n \times m)$, 其中 $m = \text{card}(AT)$ 。

定义6 在 $S=(U, AT)$ 中, 设 t 和 s 是二个描述, 如果对任意的 $(a, v) \in t$, 都有 $(a, v) \in s$, 则称 t 粗于 s (或称 s 细于 t), 记为 $t \geq s$ (或 $s \leq t$)。如果 t 是由 s 中的基本元构成, 则称 t 真粗于 s (或称 s 真细于 t), 记为 $t > s$ (或 $s < t$)。

定义7 在 $S=(U, AT)$ 中, 任意 $A \subseteq AT$, 设 $DES(A) = \{t | t \text{ 是 } A \text{ 的描述, 且 } \|t\| \neq \Phi\}$ 。对任意的 $t \in DES(A)$, 如果 $A(T) = A$, 则 t 称为 A 的全描述。记 $FDES(A) = \{t \in DES(A) | t \text{ 是 } A \text{ 的全描述}\}$ 。

算法2 在不完备信息系统 $S=(U, AT)$ 中, 求属性集 $A \subseteq AT$ 的全描述集 $FDES(A)$ 。

- (1) 令 $FDES(A) = \Phi$
- (2) 任取 $x \in U$
- (3) $B = A, F_1 = \Phi, F_2 = \Phi$
- (4) 任取 $a \in B$, 计算 $a(x)$
- (5) 对每一个 $v \in a(x)$, 构造 $t = (a, v)$, 执行(6)~(7)。

完毕后转(8)

(6) 若 $F_2 = \Phi$, 则 $F_1 = F_1 \cup \{t\}$, 转(5)

(7) 否则, 对每一个 $s \in F_2, s = s \wedge t, F_1 = F_1 \cup \{s\}$ 完毕后转(5)

(8) $F_2 = F_1, F_1 = \Phi, B = B - \{a\}$

(9) 若 $B \neq \Phi$, 则转(4)

(10) $FDES(A) = FDES(A) \cup F_2$

(11) $U = U - \{x\}$

(12) 若 $U = \Phi$, 则结束, 否则转(2)。

算法2的时间复杂度是 $O(n \times k^m)$, 其中:

$k = \max(k_1, k_2, \dots, k_m)$,

$k_i = \max(\text{card}(a_i(x_j)), j=1, 2, \dots, n)$ 。

定义8 在不完备决策表 $T=(U, AT \cup \{d\})$ 中, 任意 $t \in DES(AT)$, 函数 $\partial(t) = \{d(y) | y \in \|t\|\}$ 称为 t 的广义决策。任意的 $(d, w), w \in \partial(t)$ 称为 t 的广义决策描述。

算法3 在不完备决策表 $T=(U, AT \cup \{d\})$ 中, 任意的 $t \in FDES(AT)$, 求 t 的广义决策 $\partial(t)$ 。

- (1) $\partial(t) = \Phi$
- (2) $SUP = \|t\|$
- (3) 任取 $x \in SUP$
- (4) $\partial(t) = \partial(t) \cup \{d(x)\}$
- (5) $SUP = SUP - \{x\}$
- (6) 若 $SUP = \Phi$, 则结束, 否则转(3)

由算法1知道, 步(2)的时间复杂度为 $O(n \times m)$, 步(3)~(6)的时间复杂度为 $O(\text{card}(\|t\|))$, 又因为 $\|t\| \subseteq U$, 即 $\text{card}(\|t\|) \leq n$, 所以算法3的时间复杂度是 $O(n \times m)$ 。

4 近似集

设 $S=(U, AT)$ 是一个不完备信息系统, 对于任意的 $a \in AT$ 和 $v \in V_a$, 如果 $v \in a(x) \cap a(y)$, 也就是 $\{x, y\} \subseteq \| (a, v) \|$, x 和 y 被称为关于属性 a 相似, 同样, 若 $A \subseteq AT, t \in FDES(A), x, y \in U$ 则 x 和 y 被称为关于 A 相似当且仅当 $\{x, y\} \subseteq \|t\|$, 这些相似关系就把有限论域 U 分成了几个子集, 这些子集类构成了 U 的一个覆盖, 记成 $U/A = \{\|t\| : t \in FDES(A)\}$, 包含 x 的描述集类记为 $A(x)$, 即 $A(x) = \{\|t\| : x \in \|t\|, t \in FDES(A)\}$ 。

定义9 在不完备信息系统 $S=(U, AT)$ 中, 设 $X \subseteq U$ 且 $A \subseteq AT$, 则

$\underline{A}(X) = \{\|t\| : \|t\| \subseteq X, t \in FDES(A)\}$ 称为 X 关于 A 的下近似,

$\bar{A}(X) = \{\|t\| : \|t\| \cap X \neq \Phi, t \in FDES(A)\}$ 称为 X 关于 A 的上近似,

$\bar{A}(X) - \underline{A}(X)$ 称为 X 关于 A 的边界, 记为 $BN_A(X)$ 。

5 约简

5.1 描述的约简

定义10 在不完备信息系统 $S=(U, AT)$ 中, $A \subseteq AT, t \in FDES(A)$, 存在 $t' \in DES(AT)$, 满足 $t' \geq t, \|t'\| = \|t\|$, 而且 $\forall t'' > t', \|t''\| \neq \|t\|$, 则 t' 称描述 t 的约简。

算法4 在信息系统 $S=(U, AT)$ 中, 求属性集 $A \subseteq AT$ 且 $t \in FDES(A)$, 求 t 的约简。

- (1) $B = A$
- (2) 任取 $a \in B$, 设 $(a, v) \in t$
- (3) 构造 s , 使得 $s \wedge (a, v) = t$
- (4) 若 $\|s\| = \|t\|$, 则 $t = s$

(5) $B = B - \{a\}$

(6) 若 $B = \Phi$, 则结束, 否则转(2)

由算法 1 知道, 步(4)的时间复杂度为 $O(n \times \text{card}(A))$, 所以算法 4 的时间复杂度是 $O(n \times (\text{card}(A))^2)$ 。若 $A = AT$, 则时间复杂度是 $O(n \times m^2)$ 。

5.2 描述的广义约简

定义 11 在不完备决策表 $T = (U, AT \cup \{d\})$ 中, $t \in FDES(A)$, 存在 $t' \in DES(AT)$, 满足 $t' \geq t$, $\partial(t') = \partial(t)$, 而且 $\forall t'' > t', \partial(t'') \neq \partial(t)$, 则 t' 称描述 t 的广义约简。

算法 5 在不完备决策表 $T = (U, AT \cup \{d\})$ 中, 任意的 $t \in FDES(AT)$, 求 t 的广义约简。

(1) $B = AT$

(2) 任取 $a \in B$, 设 $(a, v) \in t$

(3) 构造 s , 使得 $s \wedge (a, v) = t$

(4) 若 $\partial(s) = \partial(t)$, 则 $t = s$

(5) $B = B - \{a\}$

(6) 若 $B = \Phi$, 则结束, 否则转(2)

由算法 3 知道, 步(4)的时间复杂度为 $O(n \times m)$, 所以算法 5 的时间复杂度是 $O(n \times m)^2$ 。

5.3 描述的 u 约简

定义 12 在不完备决策表 $T = (U, AT \cup \{d\})$ 中, $t \in FDES(A)$, 存在 $t' \in DES(AT)$, 满足 $t' \geq t$, $u_{t'}(w) = u_t(w)$, $\forall w \in V_d$, 而且 $\forall t'' > t', \exists w \in V_d, u_{t''}(w) \neq u_{t'}(w)$, 则 t' 称描述 t 的 u 约简, 其中:

$$u_t(w) = \frac{\text{card}(\{t' \mid t' \geq t, u_{t'}(w) = u_t(w)\})}{\text{card}(\{t' \mid t' \geq t\})}, w \in V_d$$

算法 6 在不完备决策表 $T = (U, AT \cup \{d\})$ 中, 求 $t \in FDES(AT)$, 求 t 的 u 约简。

(1) $B = AT$

(2) 任取 $a \in B$, 设 $(a, v) \in t$

(3) 构造 s , 使得 $s \wedge (a, v) = t$

(4) $D = V_d$

(5) 任取 $w \in D$

(6) 若 $u_s(w) \neq u_t(w)$, 则转(10)

(7) $D = D - \{w\}$

(8) 若 $D \neq \Phi$, 则转(5)

(9) $t = s$

(10) $B = B - \{a\}$

(11) 若 $B = \Phi$, 则结束, 否则转(2)

步(6)计算 $u_s(w)$ 时主要计算 $\|t\|$ 和 $\|(d, w)\|$, 所以由算法 1 知道, 步(6)的时间复杂度为 $O(n \times m)$, 又 d 是完备的, 所以算法 3 的时间复杂度是 $O(n \times m^2 \times h)$, 其中 $h = \text{card}(V_d)$ 。

6 决策规则

隐藏在不完备决策表 $T = (U, AT \cup \{d\})$ 中的知识, 能够以决策规则: $t \rightarrow s$ 的形式来表达, 其中 $t = \bigwedge (a, v), a \in A \subseteq AT$ 且 $s = \bigvee (d, w), w \in V_d$, t 和 s 分别称为规则的条件部分和决策部分。

6.1 决策规则的类型

定义 13 在不完备决策表 $T = (U, AT \cup \{d\})$ 中, 决策规则 $t \rightarrow s$, 其中 $t = \bigwedge (a, v), s = \bigvee (d, w)$, 如果 $\|t\| \neq \Phi$, 且 $\|t\| \subseteq \|s\|$, 则称决策规则 $t \rightarrow s$ 是真的, 记作 $t \Rightarrow s$ 。

定义 14 在不完备决策表 $T = (U, AT \cup \{d\})$ 中, $t = \bigwedge$

$(a, v), s = \bigvee (d, w)$,

(1) 如果决策规则 $t \rightarrow s$ 是真的, 则称决策规则 $t \Rightarrow s$ 是确定性规则;

(2) 如果决策规则 t 是 T 中的一个广义约简描述, 则称决策规则 $t \rightarrow s$ 是广义决策规则;

(3) 如果决策规则 t 是 T 中的一个 u 约简描述, 则称决策规则 $t \rightarrow s$ 是 u 决策规则。

6.2 规则产生算法

算法 7 在决策表是指 $S = (U, AT \cup \{d\})$ 中, 求确定性规则集 $CerRule$, 并产生 $uFDES$, 为求 u 决策规则和广义规则做准备。

(1) 求 $FDES(AT)$

(2) 对每一个 $w \in V_d$, 求 $\|(d, w)\|$

(3) $uFDES = FDES(AT), CerRule = \Phi$

(4) 任取 $t \in FDES(AT)$

(5) 若存在一个 $w \in V_d$, 使得 $\|t\| \subseteq \|(d, w)\|$, 则转(6), 否则转(14)

(6) $B = AT$

(7) 任取 $a \in B$, 设 $(a, v) \in t$

(8) 构造 s , 使得 $s \wedge (a, v) = t$

(9) 若 $\|s\| \subseteq \|(d, w)\|$, 则 $t = s$

(10) $B = B - \{a\}$

(11) 若 $B \neq \Phi$, 则转(7)

(12) $CerRule = CerRule \cup \{t \Rightarrow (d, w)\}$

(13) 在 $uFDES$ 中删除包含 t 的所有描述

(14) 在 $FDES(AT)$ 中删除包含 t 的所有描述

(15) 若 $FDES(AT) = \Phi$, 则结束, 否则转(4)。

由算法 2 知道步(1)的时间复杂度是 $O(n \times k^m)$ 。由算法 1 知道步(2)的时间复杂度是 $O(n \times m \times h)$ 。步(4)~(15)对每一个描述的处理与算法 4 类似, 所以步(4)~(15)的时间复杂度为 $O(n \times m^2 \times l^2)$, 其中 $l = \text{card}(FDES(AT))$ 所以算法 7 的时间复杂度是 $O(n \times f)$, 其中 $f = \max(k^m, m \times h, m^2 \times l^2)$ 。

算法 8 在决策表是指 $S = (U, AT \cup \{d\})$ 中, 已知 $uFDES$, 求 u 决策规则集 $uDecrule$ 。

(1) $uDecRule = \Phi$

(2) $uF = uFDES$

(3) 任取 $t \in uF$

(4) 对 t 进行 u 约简, 得到 t_1

(5) $D = \partial(t_1)$

(6) 任取 $w \in D$

(7) $uDecRule = uDecRule \cup \{t_1 \rightarrow (d, w)\}$

(8) $D = D - \{w\}$

(9) 若 $D \neq \Phi$, 则转(6)

(10) $uF = uF - \{t\}$

(11) 若 $uF = \Phi$, 则结束, 否则转(3)。

由算法 6 知道步(4)的时间复杂度是 $O(n \times m^2 \times h)$ 。由算法 3 知道步(5)的时间复杂度是 $O(n \times m)$ 。又 $\text{card}(\partial(t_1)) \leq h$, 步(6)~(9)的时间复杂度为 $O(h)$, 所以算法 8 的时间复杂度是 $O(n \times m^2 \times h \times \text{card}(uFDES))$ 。

算法 9 对已经求得的最优可能 u 决策规则集 $uDecRule$, 组合产生真规则集 $TRule$ 。

(1) 令 $TRule = \Phi$

(2) 在 $uDecRule$ 中, 对所有的规则按条件部分进行分类,

得到 $uClassSet$ 。

(3) 任取 $uClass \in uClassSet$

(4) $s = \Phi$

(5) 任取 $r \in uClass$, 设 r 为 $t \rightarrow s_1$

(6) 若 $s = \Phi$, 则 $s = s_1$, 否则 $s = s \vee s_1$

(7) $uClass = uClass - \{r\}$

(8) 若 $uClass \neq \Phi$, 则转(5)

(9) 若 $\|t\| \subseteq \|s\|$ 则 $TRule = TRule \cup \{t \rightarrow s\}$

(10) $uClassSet = uClassSet - \{uClass\}$

(11) 若 $uClassSet = \Phi$, 则结束, 否则转(3)。

算法 9 的时间复杂度是 $O((card(uDecRule))^2)$ 。

算法 10 在决策表是指 $S = (U, AT \cup \{d\})$ 中, 已知 $uFDES$, 求广义规则集 $gDecRule$ 。

(1) $gDecRule = \Phi$

(2) $gF = uFDES$

(3) 任取 $t \in gF$

(4) 对 t 进行广义约简, 得到 t_1

(5) 若 $t_1 = t$, 则转(11)

(6) $D = \partial(t_1)$

(7) 任取 $w \in D$

(8) $gDecRule = gDecRule \cup \{t_1 \rightarrow (d, w)\}$

(9) $D = D - \{w\}$

(10) 若 $D \neq \Phi$, 则转(7)

(11) $gF = gF - \{t\}$

(12) 若 $gF = \Phi$, 则结束, 否则转(3)。

由算法 5 知道步(4)的时间复杂度是 $O(n \times m^2)$ 。由算法 3 知道步(6)的时间复杂度是 $O(n \times m)$ 。又 $card(\partial(t_1)) \leq card(V_d) \leq card(U)$, 步(7)~(10)的时间复杂度不大于 $O(n)$, 所以算法 10 的时间复杂度是 $O(n \times m^2 \times card(uFDES))$ 。

结论 在上述讨论中, 为了论述的方便把算法分解成多个子算法来阐述。在具体应用时可以把它们组合在一起并进行优化, 就可以进行计算。例如对表 1 所示的不完备决策表进行计算, 得到结果如下:

(1) 确定性决策规则:

$r_1: (a, H) \Rightarrow (d, H)$

$r_2: (a, L) \Rightarrow (d, L)$

(2) u 决策规则:

$r_3: (a, N) \wedge (b, N) \rightarrow (d, N)$

$r_4: (a, N) \wedge (b, N) \rightarrow (d, H)$

$r_5: (a, N) \wedge (b, N) \rightarrow (d, L)$

$r_6: (a, N) \wedge (b, H) \rightarrow (d, N)$

$r_7: (a, N) \wedge (b, H) \rightarrow (d, H)$

(3) 由上述 5 条 u 决策规则可产生 2 条真规则:

$r_8: (a, N) \wedge (b, N) \Rightarrow (d, N) \vee (d, H) \vee (d, L)$

$r_9: (a, N) \wedge (b, H) \Rightarrow (d, N) \vee (d, H)$

(4) 广义决策规则:

$r'_1: (a, n) \rightarrow (d, N)$

$r'_2: (a, N) \rightarrow (d, H)$

$r'_3: (a, N) \rightarrow (d, L)$

$r'_4: (b, H) \rightarrow (d, N)$

$r'_5: (b, H) \rightarrow (d, H)$

由于求最小约简是一个 NP-难问题, 为了提高效率, 避免复杂计算, 本文所讨论的算法不是求最优规则。为了更好地满足用户的要求, 对属性可以作预处理(如排序、赋优先级等), 则在约简时可以尽可能地保留用户所需的决策信息, 就能够得到令人满意的效果。

参考文献

- 1 Pawlak Z. Rough sets. International Journal of Computer and Information Science, 1982, 11(5): 341~356
- 2 Pawlak Z, Grzymala-Busse J, Slowinski R. Rough sets. Communications of the ACM, 1995, 38(11): 89~91
- 3 王珏, 王任, 苗夺谦, 等. 基于 Rough Set 理论的“数据浓缩”. 计算机学报, 1998, 21(5): 393~400
- 4 张文修, 吴伟志, 梁吉业, 李德玉. 粗糙集理论与方法. 北京: 科学出版社, 2001
- 5 王国胤. 粗糙集理论与知识获取. 西安: 西安交通大学出版社, 2001
- 6 刘清. Rough 集及 Rough 推理. 北京: 科学出版社, 2001
- 7 张文修, 米据生, 吴伟志. 不协调目标信息系统的知识约简. 计算机学报, 2003, 26(1): 12~18
- 8 苏健, 高济. 粗糙决策支持方法. 计算机学报, 2003, 26(6): 737~745
- 9 苗夺谦, 胡桂荣. 知识约简的一种启发式算法. 计算机研究与发展, 1999, 36(6): 681~684
- 10 Wang Jue, Miao Duo-Qian. Analysis on attribute reduction strategies of rough set. Journal of Computer Science and Technology, 1998, 13(2): 189~192
- 11 常犁云, 等. 一种基于 Rough set 理论的属性约简及规则提取方法. 软件学报, 1999, 10(11): 1206~1211
- 12 Wu Wei-Zhi, Mi Ju-Sheng, Zhang Wen-Xiu. A new rough set approach to knowledge discovery in incomplete information systems. In: IEEE proc. of the Second International Conference on Machine Learning and Cybernetics, 2003. 1713~1718
- 13 Mi Ju-Sheng, Wu Wei-Zhi, Zhang Wen-Xiu. Approaches to approximation reducts in Inconsistent Decision Tables. Lecture Notes in Artificial Intelligence, 2003, 2639: 283~286
- 14 Kryszkiewicz M. Rough set approach to incomplete information systems. Information Sciences, 1998, 112: 39~49
- 15 Kryszkiewicz M. Rules in incomplete information systems. Information Sciences, 1999, 113: 271~292