

文章编号: 1001-0920(2004)10-1086-05

## 一种基于分类一致性的决策规则获取算法

代建华, 潘云鹤

(浙江大学 人工智能研究所, 浙江 杭州 310027)

**摘 要:** 提出一种基于分类一致性的规则获取算法。它是一种例化方向的方法, 即从空集开始, 以条件属性子集的分类一致性来度量属性的重要性, 逐步加入重要的属性, 当选择的属性子集能够正确分类时, 则获取到决策规则。算法中设计了一个规则约简过程, 用来简化所获得的规则, 增强规则的泛化能力。实验结果表明, 所提出的算法获得的规则更为简洁和高效。

**关键词:** 决策规则; 粗糙集; 分类; 知识发现  
**中图分类号:** TP18      **文献标识码:** A

### Algorithm for acquisition of decision rules based on classification consistency rate

DA I J ian-hua, PAN Yun-he

(Institute of Artificial Intelligence, Zhejiang University, Hangzhou 310027, China Correspondent: DA I J ian-hua, Email: jhdai@126.com)

**Abstract:** An algorithm for acquisition of decision rules, which uses the attribute importance measure based on classification consistency rate, is proposed. The algorithm uses a method by specialization, in which condition attributes are considered to be added to selected attributes set in order of significance until the selected attributes set can make classification. A procedure for reduction of decision rules is also constructed. The procedure helps to get more precise rules which have more generalizing ability. The experiment and comparison show that the algorithm provides more precise and simple decision rules.

**Key words:** decision rule; rough set; classification; knowledge discovery

## 1 引 言

分类决策是通过分析训练数据集而产生关于类别的描述, 这种类别描述通常表现为决策规则, 这些决策规则可对未知数据进行分类预测和决策。知识发现的两个主要目标是描述和预测, 从描述的角度考虑, 相比决策树等方式, 用规则描述所获得的知识更容易接受和理解。

粗糙集方法将知识看成分类, 具有很强的数据分析能力, 已广泛应用于知识发现和机器学习等领

域<sup>[1]</sup>。基于粗糙集的决策规则获取(值约简)的主要方法有: 原始的基于核值的方法, 原始的等价类匹配方法, 基于布尔推理<sup>[1,2]</sup>的最小决策算法, 缺省规则的方法<sup>[3]</sup>, 考虑覆盖度的规则方法<sup>[4]</sup>, 确定性规则和概率性规则方法<sup>[5-7]</sup>。

本文提出一种基于分类一致性的规则获取算法, 它以条件属性子集的分类一致性来度量属性的重要性, 并逐步加入重要的属性。算法中设计了一个规则约简过程, 用来简化所获得的规则, 增强规则的

收稿日期: 2003-11-03; 修回日期: 2004-01-10

基金项目: 国家 973 计划重大项目(2002CB312106); 中国博士后科学基金资助项目(20040350715); 浙江省科技计划项目(2004C31098)。

作者简介: 代建华(1977—), 男, 湖北荆州人, 博士后, 从事人工智能、粗糙集等研究; 潘云鹤(1946—), 男, 浙江杭州人, 中国工程院院士, 教授, 博士生导师, 从事人工智能、形象思维等研究。

泛化能力 通过算法的应用实例与比较发现,该算法能获得简洁而有效的规则

## 2 相关基本概念

为了后续叙述的方便,首先介绍一些粗糙集相关的基本概念

**定义 1(决策系统)** 一个决策系统(又称决策表) 定义为一个四元组  $S = (U, A, V_a, F_a)$ ,  $a \in A$ . 其中:  $U = \{x_1, x_2, \dots, x_n\}$  为对象集,即论域;  $A$  为属性集合,  $A = C \cup D$ , 且  $C \cap D = \emptyset$ ,  $C$  为条件属性集,  $D$  为决策属性集, 一般有  $D = \{d\}$ ,  $d$  为决策属性;  $V_a$  为属性  $a$  的值域;  $F_a$  为  $U \times A \rightarrow V_a$  的映射, 它为  $U$  中各对象的属性指定唯一值

**定义 2(近似集)** 在决策系统  $S$  中,  $X \subseteq U$  为一对象集,  $R$  为一等价关系, 则在  $R$  水平下  $X$  的下近似  $R_-(X) = \{x_i \in U \mid [x_i]_R \subseteq X\}$ , 表示在关系  $R$  水平下一定能归入  $X$  的对象集

**定义 3(不可分辨关系)** 在决策系统  $S$  中, 对于一属性集  $I \subseteq A$ , 可构造对应的二元等价关系  $ND(I) = \{x, y \in U \times U \mid \forall a \in I, a(x) = a(y)\}$ , 称为由  $I$  构造的不可分辨关系

**定义 4(正域)** 在决策系统  $S$  中, 对于属性集  $P, Q \subseteq A$ , 由二者构造的不可分辨关系为  $ND(P)$ ,  $ND(Q)$ , 则  $Q$  的  $P$  正域

$$POS_{ND(P)}(ND(Q)) = \bigcup_{X \in U/ND(Q)} ND(P)_-(X).$$

**定义 5(描述子、原子公式)**  $a = v$  是属性集  $B \subseteq C \cup \{d\}$  与决策系统的值域  $V$  上的描述子, 表示属性  $a$  取值  $v$ , 也称为  $B$  和  $V$  上的原子公式

**定义 6(公式)** 1) 原子公式是公式; 2) 如果  $F$  和  $G$  是原子公式, 则  $\neg F$ ,  $(F)$ ,  $F \cup G$ ,  $F \cap G$  和  $F \rightarrow G$  都是公式; 3) 只有按上述 1) 和 2) 所组成的式子是公式

**定义 7(决策规则、前件、后件)** 公式  $F \rightarrow G$  的逻辑含义称为决策规则,  $F$  称为规则前件,  $G$  称为规则后件, 它们表达一种因果关系 其中:  $F$  所包含的原子公式中只有决策系统的条件属性,  $G$  所包含的原子公式中只有决策系统的决策属性

## 3 基于分类一致性的规则获取算法

### 3.1 算法思想

在决策系统中, 条件属性包含了决策所需要的信息, 当决策系统一致时(决策系统不一致时的规则获取将另文介绍), 根据整个条件属性  $C$  集便可在训练集作出明确决策 决策系统的每个对象可认为是一个决策规则, 但这样的规则泛化能力差且数目多,

这就需要选择部分条件属性作为规则的前件 有两种方向的方法: 泛化方向, 即逐步减少规则的前件, 使得规则前件包含的条件属性数目减少, 进而得到最后的规则; 例化方向, 即从空集或某些基本的属性开始, 逐步增加规则的前件, 使得规则前件所包含的条件属性数目增加, 进而得到所需规则

本文算法是一种例化方向的算法, 它从空集开始逐步加入条件属性, 尽量以少的属性提取隐含在决策系统中的有用模式 当属性数目不足以作决策时, 则引入新的属性 各个条件属性对于决策的重要性是不同的, 可以分类一致率来衡量属性的重要性

$$\text{对于决策系统 } L = (U, C \cup \{d\}, V, f), \text{ 考虑 } POS_{ND(P)}(ND(\{d\})), \quad (1)$$

它表明在规则前件为  $P \subseteq C$  的属性时所能正确分类的对象 考虑属性  $a \in C - P$  的重要性

$$SGF(a) = \text{card}(POS_{ND(P \cup \{a})}(ND(\{d\}))), \quad (2)$$

它表明在引入属性  $a$  后, 能正确分类的对象数目越多, 属性  $a$  对分类越重要, 即以分类一致率来衡量属性的重要性 当有多个属性同时达到最大时, 选择满足

$$\min(H(\{d\} \mid \{a\})), \quad (3)$$

其中

$$H(\{d\} \mid \{a\}) = - \sum_{i=1}^n p(X_i) \sum_{j=1}^k p(Y_j \mid X_i) \log p(Y_j \mid X_i)$$

为决策属性  $d$  相对于条件属性  $a$  的相对熵;  $X = \{X_1, X_2, \dots, X_n\}$ ,  $Y = \{Y_1, Y_2, \dots, Y_k\}$  为  $ND(\{a\})$  和  $ND(\{d\})$  产生的划分  $H(\{d\} \mid \{a\})$  从信息论的角度反映出条件属性  $a$  对决策的重要性

### 3.2 算法描述

**算法 1** 基于分类一致率的规则获取算法

输入: 一致的决策系统  $L = (U, C \cup \{d\}, V, f)$

输出: 规则集 Rules

Procedure R ICCR

Begin

1) 初始化阶段

$$G = L$$

$\setminus G$  为未被 Rules 规则覆盖的对象集  $\setminus$

$$\text{Rules} = \emptyset$$

$\setminus \text{Rules}$  为已获得的决策规则集  $\setminus$

$$\text{SelecA ttr} = W$$

$\setminus \text{SelecA ttr}$  为当前已考察过的条件属性集, 即已选择的属性, 其初始值为用户感兴趣的

属性集  $W$ , 可以为空 /  
 $unSelecA\ ttr = C - W$   
 / $unSelecA\ ttr$  为待引入的条件属性集 /

2) 规则获取阶段

While  $G \neq \emptyset$  do  
 Begin  
    $rule = \emptyset$   
   for  $i = 1$  to  $card(unSelecA\ ttr)$  do  
     计算  $POS_{ND}(SelecA\ ttr - \{a_i\})(ND(\{d\}))$   
   end  
   选择使得  $card(POS_{ND}(SelecA\ ttr - \{a_i\}) \times (ND(\{d\})))$  达到最大的属性  $a$   
   如果有多个属性同时达到最大  
   则选择属性  $a$  使得  $H(\{d\}|\{a\})$  最小  
    $SelecA\ ttr = SelecA\ ttr \cup \{a\}$   
    $unSelecA\ ttr = unSelecA\ ttr - \{a\}$   
   if  $POS_{ND}(SelecA\ ttr - \{a\})(ND(\{d\})) = \emptyset$   
   /此时有新的规则生成 /  
   then  
     begin  
       用属性  $SelecA\ ttr$  从对象集  
        $POS_{ND}(SelecA\ ttr - \{a\})(ND(\{d\}))$   
       导出规则  
       将化简后的规则并入  $Rules$   
        $G = G - POS_{ND}(SelecA\ ttr - \{a\})(ND(\{d\}))$   
     end  
 end

3) 规则约简阶段

for 规则集  $Rules$  中的每条规则  $rule$   
 for 规则  $rule$  中的每个描述子  
 /根据属性重要性从小到大排序 /

if 删除该描述子  $b = v$  后  
 规则与其他规则不产生冲突  
 then 删除该描述子  
 end

end

整个算法分为 3 个阶段: 第 1 阶段为初始化阶段, 为各参数指定初始值; 第 2 阶段为循环的规则获取阶段, 从决策系统获取原始的决策规则; 第 3 阶段为规则约简阶段, 对循环规则获取的原始决策规则进行约简, 得到泛化能力更强、更易于理解的规则。第 2 阶段是 3.1 节所述思想的具体体现, 当决策系统的对象未被所获决策完全覆盖时, 逐步引入重要的属性。当有多个属性同时达到最大时, 考虑它们的条件熵  $H(\{d\}|\{a\})$ , 选择其中条件熵最小的属性。若  $\{d\}$  的  $SelecA\ ttr - \{a\}$  正域不为空, 则产生新的规则并将规则覆盖的对象删除。第 3 阶段考察规则中的描述子, 看它是否可以简约。属性的考察顺序是很重要的, 本文采用属性重要性逆序, 即从最不重要的属性开始考察。

4 应用实例与比较

4.1 算法应用实例

为考察本文算法的有效性, 现用 Quinlan 提供的著名的高尔夫球数据作为决策系统, 如表 1 所示。其中: 论域  $U = \{1, 2, \dots, 14\}$ , 条件属性集  $C = \{a_1, a_2, a_3, a_4\}$ , 决策属性为  $Class$ 。

规则获取阶段开始, 客户没有特别感兴趣的属性集, 即  $SelecA\ ttr = W = \emptyset$ , 考察各条件属性和决策属性的分类, 并按式(2) 计算正域

$$POS_{ND}(\{a_1\})ND(\{Class\}) = \{3, 7, 12, 13\},$$

表 1 高尔夫球数据集

$U$	Outlook ( $a_1$ )	Temperature ( $a_2$ )	Humidity ( $a_3$ )	Windy ( $a_4$ )	Class
1	sunny	hot	high	false	Don't play
2	sunny	hot	high	true	Don't play
3	overcast	hot	high	false	Play
4	rain	mild	high	false	Play
5	rain	cool	normal	false	Play
6	rain	cool	normal	true	Don't play
7	overcast	cool	normal	true	Play
8	sunny	mild	high	false	Don't play
9	sunny	cool	normal	false	Play
10	rain	mild	normal	false	Play
11	sunny	mild	normal	true	Play
12	overcast	mild	high	true	Play
13	overcast	hot	normal	false	Play
14	rain	mild	high	true	Don't play

表 2 选择属性  $a_1$  得到的决策系统

$U$	Outlook ( $a_1$ )	Temperature ( $a_2$ )	Hum idity ( $a_3$ )	W indy ( $a_4$ )	Class
1	sunny	hot	high	false	Don t play
2	sunny	hot	high	true	Don t play
4	rain	m ild	high	false	Play
5	rain	cool	nom al	false	Play
6	rain	cool	nom al	true	Don t play
8	sunny	m ild	high	false	Don t play
9	sunny	cool	nom al	false	Play
10	rain	m ild	nom al	false	Play
11	sunny	m ild	nom al	true	Play
14	rain	m ild	high	true	Don t play

表 3 选择属性  $a_1$  和  $a_3$  得到的决策系统

$U$	Outlook ( $a_1$ )	Temperature ( $a_2$ )	Hum idity ( $a_3$ )	W indy ( $a_4$ )	Class
4	rain	m ild	high	false	Play
5	rain	cool	nom al	false	Play
6	rain	cool	nom al	true	Don t play
10	rain	m ild	nom al	false	Play
14	rain	m ild	high	true	Don t play

$$PO S_{ND((a_2))} ND(\{Class\}) = \emptyset,$$

$$PO S_{ND((a_3))} ND(\{Class\}) = \emptyset,$$

$$PO S_{ND((a_4))} ND(\{Class\}) = \emptyset.$$

选择属性  $a_1$ , 得到规则

$$\text{“Outlook}(a_1) = \text{overcast”} \Rightarrow$$

$$\text{“Class} = \text{Play”}\{3, 7, 12, 13\}.$$

删除对象 3, 7, 12 和 13 后, 得到的决策系统如表 2 所示

此时,  $SelecA\ ttr = \{a_1\}$ ,  $unSelecA\ ttr = \{a_2, a_3, a_4\}$ . 考察条件属性集  $\{a_1, a_2\}$ ,  $\{a_1, a_3\}$ ,  $\{a_1, a_4\}$  和决策属性  $Class$  的分类, 并按式(2) 计算正域

$$PO S_{ND((a_1, a_2))} ND(\{Class\}) = \{1, 2, 9\},$$

$$PO S_{ND((a_1, a_3))} ND(\{Class\}) = \{1, 2, 8, 9, 11\},$$

$$PO S_{ND((a_1, a_4))} ND(\{Class\}) = \{4, 5, 6, 10, 14\}.$$

此时

$$\text{card}(PO S_{ND((a_1, a_3))} ND(\{Class\})) = 5,$$

$$\text{card}(PO S_{ND((a_1, a_4))} ND(\{Class\})) = 5,$$

同时取最大值, 需要计算条件熵

$$H(D|\{a_3\}) = 0.5004, H(D|\{a_4\}) = 0.6068$$

显然,  $H(D|\{a_3\}) < H(D|\{a_4\})$ , 因此选择属性  $a_3$ , 得到规则

$$\text{“Outlook}(a_1) = \text{sunny Hum idity}(a_3) = \text{high”} \Rightarrow$$

$$\text{“Class} = \text{Don t play”}\{1, 2, 8\},$$

$$\text{“Outlook}(a_1) = \text{sunny Hum idity}(a_3) = \text{nom} \Rightarrow$$

$$\text{“Class} = \text{Play”}\{9, 11\}.$$

删除已被规则覆盖的对象 1, 2, 8, 9 和 11, 得到的决策系统如表 3 所示

此时,  $SelecA\ ttr = \{a_1, a_3\}$ ,  $unSelecA\ ttr = \{a_2, a_4\}$ . 考察条件属性集  $\{a_1, a_2, a_3\}$ ,  $\{a_1, a_3, a_4\}$  和决策属性  $Class$  的分类, 并按式(2) 计算正域

$$PO S_{ND((a_1, a_2, a_3))} ND(\{Class\}) = \{10\},$$

$$PO S_{ND((a_1, a_3, a_4))} ND(\{Class\}) = \{4, 5, 6, 10, 14\}.$$

取属性  $a_4$ , 得到如下规则:

$$\text{“Outlook}(a_1) = \text{rain Hum idity}(a_3) = \text{high W indy}(a_4) = \text{false”} \Rightarrow$$

$$\text{“Class} = \text{Play”}\{4\},$$

$$\text{“Outlook}(a_1) = \text{rain Hum idity}(a_3) = \text{nom W indy}(a_4) = \text{false”} \Rightarrow$$

$$\text{“Class} = \text{Play”}\{5, 10\},$$

$$\text{“Outlook}(a_1) = \text{rain Hum idity}(a_3) = \text{nom W indy}(a_4) = \text{true”} \Rightarrow$$

$$\text{“Class} = \text{Don t play”}\{6\},$$

$$\text{“Outlook}(a_1) = \text{rain Hum idity}(a_3) = \text{high W indy}(a_4) = \text{true”} \Rightarrow$$

$$\text{“Class} = \text{Don t play”}\{14\}.$$

此时所有的对象已被覆盖, 得到了 7 条基本规则 经算法第 3 阶段化简后, 得到的规则集为

$$\text{“Outlook}(a_1) = \text{overcast”} \Rightarrow$$

$$\text{“Class} = \text{Play”}\{3, 7, 12, 13\},$$

$$\text{“Outlook}(a_1) =$$

sunny Humidity( $a_3$ ) = high"  $\Rightarrow$   
 "Class = Don t play" {1, 2, 8},  
 "Outlook( $a_1$ ) =  
 sunny Humidity( $a_3$ ) = nom"  $\Rightarrow$   
 "Class = Play" {9, 11},  
 "Outlook( $a_1$ ) = rain Windy( $a_4$ ) = false"  $\Rightarrow$   
 "Class = Play" {4, 5, 10},  
 "Outlook( $a_1$ ) = rain Windy( $a_4$ ) = true"  $\Rightarrow$   
 "Class = Don t play" {6, 14}.

规则集中共有5条规则,规则总长为9.

## 4.2 与其他几种算法的比较

### 4.2.1 与RITD算法的比较

文献[8]提出的RITD算法采用熵测度来度量决策系统中条件属性和决策属性的相关性,通过逐步删除最不相关属性并从相容的对象中提取规则.此方法抗噪声能力强,但规则前件较复杂,有冗余且可能有些规则不一致.对于表1所示的决策系统,RITD算法得到的规则集为

"Outlook( $a_1$ ) =  
 sunny Humidity( $a_3$ ) = high"  $\Rightarrow$   
 "Class = Don t play",  
 "Outlook( $a_1$ ) =  
 overcast Humidity( $a_3$ ) = high"  $\Rightarrow$   
 "Class = Don t play",  
 "Humidity( $a_3$ ) = nom"  $\Rightarrow$  "Class = Play",  
 "Humidity( $a_3$ ) =  
 nom Windy( $a_4$ ) = false"  $\Rightarrow$   
 "Class = Play",  
 "Outlook( $a_1$ ) = rain Humidity( $a_3$ ) =  
 high Windy( $a_4$ ) = false"  $\Rightarrow$   
 "Class = Play",  
 "Outlook( $a_1$ ) = rain Humidity( $a_3$ ) =  
 nom Windy( $a_4$ ) = true"  $\Rightarrow$   
 "Class = Don t play",  
 "Outlook( $a_1$ ) = rain Humidity( $a_3$ ) =  
 high Windy( $a_4$ ) = true"  $\Rightarrow$   
 "Class = Don t play".

其中第3条规则是不一致的,它与表1所示决策系统的第6个对象矛盾.此规则集共有7条规则,规则总长为19,比本文算法的规则数目和规则总长度都大.

### 4.2.2 与LEM2算法的比较

LEM2是LERS中所采用的算法<sup>[7]</sup>.对于表1所

示的决策系统,该算法所得到的结果规则集为

"Outlook( $a_1$ ) =  
 sunny Humidity( $a_3$ ) = high"  $\Rightarrow$   
 "Class = Don t play" {1, 2, 8},  
 "Outlook = rain( $a_1$ ) Windy( $a_4$ ) = true"  $\Rightarrow$   
 "Class = Don t play" {6, 14},  
 "Humidity( $a_3$ ) =  
 nom Windy( $a_4$ ) = false"  $\Rightarrow$   
 "Class = Play" {5, 9, 10, 13},  
 "Outlook( $a_1$ ) =  
 overcast Humidity( $a_3$ ) = high"  $\Rightarrow$   
 "Class = Play" {3, 12},  
 "Outlook( $a_1$ ) =  
 rain Windy( $a_4$ ) = false"  $\Rightarrow$   
 "Class = Play" {4, 5, 10},  
 "Outlook( $a_1$ ) =  
 overcast Humidity( $a_3$ ) = nom  $\Rightarrow$   
 "Class = Play" {7, 13},  
 "Outlook( $a_1$ ) = sunny Temperature( $a_2$ ) =  
 mild Windy( $a_4$ ) = true"  $\Rightarrow$   
 "Class = Play" {11}.

规则集中共有7条规则,规则总长为15,比本文算法的规则数目和规则总长度都大.

## 5 结 语

本文提出一种基于分类一致性的规则获取算法,它以条件属性子集的分类一致性来度量属性的重要性.算法中设计了一个规则约简过程,用来简化获得的规则,增强规则的泛化能力.本文算法具有如下特点:1)不需在规则获取阶段前进行属性约简的过程,挑选重要的属性子集在规则获取算法中自动进行;2)可以方便地让用户指定自己感兴趣的属性集,有较强的互动性;3)指定规则集中的规则自然地按强度由大到小排列,便于分类决策.实验结果表明,所获得的规则简洁,规则集的规模小,具有较好的可理解性和较强的泛化能力.

### 参考文献(References):

- [1] Pawlak Z. *Rough Sets: Theoretical Aspects of Reasoning about Data* [M]. Boston: Kluwer Academic Publishers, 1991.
- [2] Skowron A. Extracting laws from decision tables: A rough set approach [J]. *Computational Intelligence*, 1995, 11(2): 371-388.

(下转第1096页)

率搜索到一组操作参数,将当前工况优化到优、良或中的综合工况状态;在综合工况为“中”的条件下,优化控制算法的有效率达到96.8%。可以看出,本文提出的FCM-SCGA算法在实际生产过程中具有较高的优化效率

系统运行部分结果如图4所示,其中曲线为对应的实际工况,散点为综合指标参数优化结果

## 7 结 论

本文以铅锌烧结过程的建模与优化控制为背景,采用神经网络、模糊控制、专家系统、混沌优化、遗传算法、模糊C均值聚类搜索多种智能控制技术的集成,实现了烧结过程状态和综合指标优化控制。工业实际运行效果表明,预测模型具有较高的精度,优化控制减小了烧结过程状态波动,提高了烧结机的结块率和烧结块的质量,达到高产、低耗、优质的目标。智能集成优化控制技术在铅锌烧结过程中的成功运用,为智能控制技术的工业化、实用化提供了一个范例。

## 参考文献(References):

- [1] 谢良贤. 烧结过程自动控制技术的发展[J]. 冶金自动化, 1994, 18(6): 7-11.  
(Xie Liangxian. Development of automatic control technology for agglomeration process[J]. *Metallurgical Industry Automation*, 1994, 18(6): 7-11.)
- [2] 王海东, 邱冠周, 黄圣生. 烧结过程控制技术的发展[J]. 矿冶工程, 1999, 19(3): 3-6.  
(Wang Haidong, Qiu Guanzhou, Huang Shengsheng. Advances in control techniques of sintering process[J]. *Mining and Metallurgical Engineering*, 1999, 19(3): 3-6.)
- [3] 周取定, 孔令坛. 铁矿石造块理论及工艺[M]. 北京: 冶金工业出版社, 1989. 76-79.
- [4] 王雅琳. 智能集成建模理论及其在有色冶炼过程优化控制中的应用研究[D]. 长沙: 中南大学, 2001.
- [5] 仲蔚, 俞金寿. 基于模糊C均值聚类的多模型软测量建模[J]. 华东理工大学学报, 2000, 26(1): 83-87.  
(Zhong Wei, Yu Jinshou. Study on sensing modeling via FCM-based multiple models[J]. *J of East China University of Science and Technology*, 2000, 26(1): 83-87.)
- [6] 周志坚, 毛宗源. 一种基于遗传算法的模糊神经网络最优控制[J]. 控制理论与应用, 2000, 17(5): 784-788.  
(Zhou Zhijian, Mao Zongyuan. The fuzzy neural networks optimal control based genetic algorithm[J]. *Control Theory and Applications*, 2000, 17(5): 784-788.)
- [7] 李兵, 蒋蔚孙. 混沌优化方法及其应用[J]. 控制理论与应用, 1997, 14(4): 613-615.  
(Li Bing, Jiang Weisun. Chaos optimization method and its application[J]. *Control Theory and Applications*, 1997, 14(4): 613-615.)
- [8] 恽为民, 席裕庚. 遗传算法的全局收敛性和计算效率分析[J]. 控制理论与应用, 1996, 13(4): 455-460.  
(Yun Wein, Xi Yugeng. The analysis of global convergence and computational efficiency for genetic algorithm[J]. *Control Theory and Applications*, 1996, 13(4): 455-460.)
- [9] 何琳, 王科俊, 李国斌, 等. 关于“遗传算法的全局收敛性和计算效率分析”一文的商榷[J]. 控制理论与应用, 2001, 18(1): 142-145.  
(He Lin, Wang Kejun, Li Guobin, et al. The discussion about the paper “The analysis of global convergence and computational efficiency for genetic algorithm”[J]. *Control Theory and Applications*, 2001, 18(1): 142-145.)
- [3] Mollestad T, Skowron A. A rough set framework for data mining of propositional default rules[A]. *Proc of Ninth Int Symp on Methodologies for Intelligent Systems*[C]. Berlin: Springer-Verlag, 1996. 448-457.
- [4] Tsumoto S. Modelling medical diagnostic rules based on rough sets[A]. *Proc of the First Int Conf on Rough Sets and Current Trends in Computing*[C]. Warsaw, 1998. 475-482.
- [5] Stefanowski J. On rough set based approaches to induction of decision rules[A]. *Rough Sets in Data Mining and Knowledge Discovery*[C]. Berlin: Physica-Verlag, 1998. 1: 500-529.
- [6] Stefanowski J. Rough set based rule induction techniques for classification problems[A]. *Sixth European Congress on Intelligent Techniques and Soft Computing*[C]. Aachen, 1998. 1: 109-113.
- [7] Grzymala-Bausse D M, Grzymala-Busse J W. The usefulness of machine learning approach to knowledge acquisition[J]. *Computational Intelligence*, 1995, 11(2): 268-279.
- [8] Wu X. Induction by attribute elimination[J]. *IEEE Trans on Knowledge and Data Engineering*, 1999, 11(5): 805-812.

(上接第1090页)