基于属性值重要性的 Rough 集值约简算法

宋旭东,朱伟红,宁 涛
(大连交通大学 软件学院,辽宁 大连 116028)

摘 要:值约简是 Rough 集理论的一个重要研究课题。很多学者对它进行了研究并提出了不同的值约简算法,但是在执行效率上还有待提高。在启发式值约简算法基础上,结合属性值的重要性,提出了一种改进的基于属性值重要性的 Rough 集值约简算法,该算法在执行效率上有很大的提高,通过实例分析验证了该算法的可行性和有效性。

关键词:Rough 集;信息表;值约简
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2007)06-0077-03

Rough Set Algorithm for Value Reduction Based on Importance of Attribute Value

SONG Xu-dong, ZHU Wei-hong, NING Tao
(Software College, Dalian Jiaotong University, Dalian 116028, China)

Abstract: The value reduction is an important research topic in rough set theory, many researchers studied it and put forward different algorithms of value reduction. But it still remains to improve in carrying out efficiency. Puts forward a kind of improved rough set algorithm for value reduction based on importance of attribute value on the basis of the heuristic algorithm for value reduction and combining the importance of attribute value. This algorithm is improved a lot in carrying out efficiency. And it has verified feasibility and validity of this algorithm through the instance analysis.

Key words: rough set; information table; value reduction

0 引 言

Rough 集理论是由波兰科学家 Pawlak 在 1982 年提出的一种处理含糊和不精确问题的新型数学工具[1]。值约简是粗糙集理论的一个重要研究课题。很多学者对它进行了研究并提出了不同的值约简算法,但目前还没有十分高效的值约简算法。文献[2]和[3]提出的启发式值约简算法实质是首先寻找出决策规则中的核属性值、冗余属性值和那些一次不能决定(即待定)是否是冗余的属性值,然后再依次考虑那些待定是否是冗余的属性值。但此算法在考虑那些待定是否是冗余的属性值上花费了大量的时间。针对上述问题,笔者提出一种改进的基于属性值重要性的 Rough 集值约简算法。

1 Rough 集的一些相关概念

下面先给出文中一些关于 Rough 集的基本概念,详细请参考文献[1~5]。

定义1 一个信息表 S 可以表示为:$S=<U,R,V,f>$,其中 U 是对象的集合,也称论域;$R=C\cup D$ 是属性集合,C 和 D 分别称为条件属性集合和决策属性集合;$V=\bigcup_{r\in R}V_r$ 是属性值集合,V_r 表示属性 $r\in R$ 的属性值范围,即属性 r 域;$f:U\times R->V$ 是一个信息函数,它指定 U 中每一个对象 X 的属性值[2]。

定义2 在信息表 S 中,对于每个属性子集 $B\subseteq R$,可以定义一个不可区分关系 $\mathrm{IND}(B)$:$\mathrm{IND}(B)=\{(x,y)\in U\times U:\forall b\in B, f(x,b)=f(y,b)\}$,显然 $\mathrm{IND}(B)$ 是一个等价关系,对象 x 在属性集 B 上的等价类 $[x]_{\mathrm{IND}(B)}$ 定义为 $[x]_{\mathrm{IND}(B)}=\{y:y\in U, y\mathrm{IND}(B)x\}$,为简便起见,在不产生混淆的情况下用 B 代替 $\mathrm{IND}(B)$[4]。

定义3 在信息表 S 中,对于 $\forall x\in U$,用 d_x 表示决策规则,即:$d_x:\mathrm{des}([x]_C)\rightarrow\mathrm{des}([x]_D)$,其中 $\mathrm{des}([x]_C)$ 表示对等价类 $[x]_C$ 的描述,即等价类 $[x]_C$ 对于各条件属性值的特定取值;$\mathrm{des}([x]_D)$ 表示对等价类 $[x]_D$ 的描述,即等价类 $[x]_D$ 对于各决策属性值的特定取值;而对于 $\forall a\in C\cup D, d_x(a)=a(x)$,

收稿日期:2006-08-24
作者简介:宋旭东(1969-),男,辽宁大连人,副教授,研究方向为数据挖掘、决策支持系统。

$a(x)$ 为个体 x 关于属性 a 的属性值，且 $d_x \mid C$ 和 $d_x \mid D$ 分别称为 d_x 的条件和决策[5]。

定义 4　如果对于每个 $y \neq x(x, y \in U)$，$P \subseteq C, d_y \mid P = d_x \mid P$，意味着 $d_y \mid D = d_x \mid D$ 则由属性 R 下的属性值就可做出正确决策，相反如果 $d_y \mid D \neq d_x \mid D$ 则称在属性 R 下决策规则产生冲突。

定义 5　若删除某条决策规则 d_x 中的条件属性 a，该条决策规则将和其他决策规则产生冲突，则称该属性 a 的属性值 $d_x(a)$ 为关键值。即 $P \subseteq C, d_y \mid P - \{a\} = d_x \mid P - \{a\}$ 时，$d_y \mid D \neq d_x \mid D$，则称属性 a 的属性值 $d_x(a)$ 为关键值，记为 $d_x^k(a)$，对于 $\forall d_x^k(a) = R$，则称 R 为决策规则 d_x 的值核。

定义 6　在信息表 S 中，决策规则关于条件属性集合 C 的一致程度 $\mu(d_x, C)$ 定义为：

$$\mu(d_x, C) = \frac{\mid [x]_C \cap [x]_D \mid}{\mid [x]_C \mid}$$，显然 $0 < \mu(d_x, C) \leqslant 1$，且 $\mu(d_x, C) = 1$ 时 d_x 是一致的，否则是不一致的[5]。

定义 7　在决策规则 d_x 中，设 $a \in V_C - R(R \subseteq V_C)$，其中 V_C, R 分别为决策规则 d_x 的所有条件属性值集合和值核，对属性值 a 的重要性 $\mathrm{SIG}(d_x, a, R)$ 定义为：$\mathrm{SIG}(d_x, a, R) = \mu(d_x, \{a\}) - \mu(d_x, R)$，其中，若 $R = \varnothing$，则令 $\mu(d_x, R) = 1$。

2　算法描述

文中主要研究属性值约简，是在文献[2]，[3]等的启发式值约简算法基础上，结合属性值重要性而提出的一种改进启发式值约简算法，下面给出算法的详细描述。

算法：改进的基于属性值重要性 Rough 集的值约简算法(RAVI)

输入：信息表 S

输出：S 的值约简 S'

步骤 1　对信息表 S 中的每条决策规则条件属性进行逐列考察，如果删除该属性列后：

1) 若产生冲突决策规则，则保留冲突决策规则的原该属性值，该值表示不可约简；

2) 若没产生冲突并且含有重复决策规则，则将重复决策规则的该属性值标为“ * ”，该值表示可以约简；

3) 若没产生冲突并且不含有重复决策规则，将该属性值标为“?”，该值表示待定是否可以约简。

信息表中没被标记“ * ”或“?”的属性值即为值核。

步骤 2　删除可能产生的重复决策规则。若某个决策规则的所有条件属性均被标记，则将标有“?”的属性项修改为原属性值。

步骤 3　依次考察每条决策规则中的标记“?”的属性值。

(1)如果只有一个“?”时，转到(3)；如果一个决策规则中有多个“?”时，依据定义 7 算出每个标记“?”的属性值的重要性。

(2)选取该条决策规则中属性值重要性最大的为“?”的属性值。

(3)若仅由未被标记的属性值即可判断出决策，则转到(4)，否则，转到(5)。

(4)将该属性值改为“ * ”，并且如果该决策规则有多个“?”时，把比该属性值重要性小的带“?”标记的属性值都改为“ * ”。

(5)将该属性值改为原属性值。并且再转到(2)。

步骤 3　删除所有条件属性均被标为“ * ”的决策规则及可能产生的重复决策规则。

步骤 4　有两个决策规则仅有一个条件属性值不同，且其中一个决策规则该属性被标为“ * ”。

①对于不同属性值被标为“ * ”的决策规则，如果可由未被标记的属性值判断出决策，则删除不同属性值不是标为“ * ”的决策规则。

②否则，删除不同属性值被标为“ * ”的决策规则。

3　实例分析

现举一例，如表 1 所示，它是经过属性约简后的一个信息表，其中 a, b, c 为条件属性，d 为决策属性[3]。

表 1　示例信息表

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

对表 1 采用 RAVI 进行值约简，以第 1 条决策规则为例，若删除属性 a，由于 $b = 0, c = 0$ 时和第 2 条决策规则产生冲突，所以保留原该属性值；若删除属性 b，由于没产生冲突并且不含有重复决策规则，因此属性 b 的值标记为“?”；同理属性 c 的值也标记为“?”。对于第 2 条决策规则中的属性 c，如果删除属性 c，由于 $a = 1, b = 0$ 时和第 3 条决策规则重复，所以第 2 条决策规则的属性 c 的值标记为“ * ”。逐条决策规则进行处理，得到表 2。

对于第 7 条决策规则，其所有条件属性均被标记，则将标有“?”的属性项 a 和 c 修改为原属性值，即 $a \rightarrow 2$ 和 $c \rightarrow 2$。对于第 1 条决策规则，根据定义 7，再参看表 1 算出属性值 $(b, 0)$ 和 $(c, 0)$ 的重要性，其中属性值

$(b,0)$ 的重要性如下:

$$\frac{|[1]_b \cap [1]_d|}{|[1]_b|} - \frac{|[1]_a| \cap [1]_d|}{|[1]_a|} = \frac{1}{3} - \frac{1}{1}$$

$$= -\frac{2}{3}$$

而属性值$(c,0)$ 的重要性如下:

$$\frac{|[1]_c \cap [1]_d|}{|[1]_c|} - \frac{|[1]_a| \cap [1]_d|}{|[1]_a|} = \frac{1}{2} - \frac{1}{1}$$

$= -\frac{1}{2}$。依据 RAVI 首先考虑属性值最重要的,即$(c,0)$。依据定义 4,$(a,0)$ 可以做出正确决策,所以把表 2 中第一条决策规则中的属性 c 的值改为"*",又因属性值$(b,0)$ 比属性值$(c,0)$ 小,所以第一条规则中的属性 b 的值也改为"*"。

表 2　初步值化简结果

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

逐条决策规则进行考虑,得到表 3。

表 3　进一步处理结果

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

因为决策规则 5 和 6 重复,所以删除决策规则 6。信息表中决策规则 5 和 7 除属性 a 外其余属性值对应都相等,并且决策规则 5 为"*",并且依据定义 4 和表 1,可知仅由$(c,2)$ 即可做出正确决策,所以删除决策规则 7。最后得到化简结果表 4。

表 4　最终值化简结果

U	a	b	c	d
1	0	*	*	0
2	1	0	*	1
3	*	1	1	0
4	*	*	2	2

4　结　语

利用 Rough 集理论,参考文献[2],[3]等的启发式值约简算法,结合属性值的重要性,提出了一种改进的基于属性值重要性的 Rough 集值约简算法(RAVI)。大大地提高了属性值约简算法的执行效率。

参考文献:

[1] PAWLAK Z. Rough sets[J]. International Journal of Computer and Information sciences, 1982, 11(5): 341 - 356.

[2] 王国胤. Rough 集理论与知识获取[M]. 西安:西安交通大学出版社,2001.

[3] 林杰斌,刘明德,陈　湘. 数据挖掘与 OLAP 理论与实务[M]. 北京:清华大学出版社,2003.

[4] 刘　清. Rough 集及 Rough 推理[M]. 北京:科学出版社,2001.

[5] 胡　斐,张峰筠,刘少辉. 一种基于 Rough 集的属性值约简算法[J]. 计算机工程与应用,2003(31):48 - 51.

(上接第 76 页)

采用文中的自适应免疫遗传算法的实验结果为:运行 50 次,平均解为 220.674,最优解为 217.8,搜索到最优解的次数 32 次,而搜索到最优解的平均进化代数为 30.53。

通过以上实验结果可以发现,采用自适应遗传算法搜索到最优解或满意解的概率要较大,较好地克服了遗传算法的"早熟"现象,而且,由于采用了自适应的交叉、变异概率,该算法的平均进化代数要少,从而节约了运算时间。实验还进一步发现,随着进化代数的增加,能得到更好的较优解。

4　结　论

文中将自适应免疫遗传算法应用于车辆调度问题。该算法在传统遗传算法的基础上加入了免疫思想,引入了生物免疫机制,还加入了自适应的交叉、变异算子,实验证明,该算法具有较好的全局搜索性能和较快的收敛性,能在较少的进化代数下找到较优解,能较好地避免局部最优解,可有效地解决车辆调度问题。

参考文献:

[1] 杨　弋,顾幸生. 物流配送车辆优化调度的综述[J]. 东南大学学报:自然科学版,2003,33(9):105 - 111.

[2] Dasgupta D. Artificial Immune Systems and Their Applications[M]. Berlin, Heidelberg: Springer - Verlag, 1999.

[3] 章　棘,周　泉. 基于免疫克隆算法的物流配送车辆路径优化研究[J]. 湖南大学学报:自然科学版,2004,31(5):54 - 58.

[4] 阎　庆,鲍远律. 新型模拟退火算法求解物流配送路径问题[J]. 计算机应用,2004,24(6):261 - 263.

[5] 李　菁,王宗军,蒋元涛,等. 免疫算法在车辆调度问题中的应用[J]. 运筹与管理,2003,12(6):96 - 100.

[6] 宋玉林,齐　欢. 基于自适应遗传算法的配送车辆调度聚类分析[J]. 计算机与数字工程,2004,32(2):45 - 47.