DOI:10.13657/j.cnki.gxkxyxb.2013.01.002

广西科学院学报
Journal of Guangxi Academy of Sciences

2013,29(1):4～6,10
Vol. 29, No. 1 February 2013

一种基于粗糙集理论的值约简算法*

A Complete Value Reduction Algorithm based on Rough Set Theory

樊艳英[1,2],徐章艳[1],张 伟[1],张自敏[3],陈冠萍[2]
FAN Yan-ying[1,2],XU Zhang-yan[1],ZHANG Wei[1],ZHANG Zi-min[3],CHEN Guan-ping[2]

(1.广西师范大学计算机科学与信息工程学院,广西桂林 541004;2.贺州学院计算机系,广西贺州 542800;3.贺州学院教育技术中心,广西贺州 542800)
(1. College of Computer Science and Information Engineering, Guangxi Normal University, Guilin, Guangxi, 541004, China; 2. Computer Engineering and Technology, Hezhou University, Hezhou, Guangxi, 542800, China; 3. Education and Technology Center, Hezhou University, Hezhou, Guangxi, 542800, China)

摘要:基于粗糙集理论,提出一种无需建立差别矩阵,无需计算分明函数的值约简算法,阐述该算法的设计思想和具体步骤,并用具体算例证明此算法可行,而且获取的规则是完备无冗余的。

关键词:粗糙集 值约简 规则提取 完备值约简
中图法分类号:TP31 **文献标识码**:A **文章编号**:1002-7378(2013)01-0004-03

Abstract: A value reduction algorithm without both structuring decision matrix and calculating decision function is proposed based on rough set theory, for which the design ideas and concrete steps were expounded. And specific cases were used to illustrate its feasibility. The result of this algorithm is complete and redundant.

Key words: rough sets, value reduction, rule extraction, complete value reduction

粗糙集理论能够很好地处理不完备、不确定、不精确的含糊信息,通过近似集概念来描述和表达系统中的含糊性和不确定性,在数据分析、机器学习、知识发现、知识获取、决策分析和过控制等领域都得到了广泛的应用[1]。属性约简和规则提取是粗糙集应用研究方面的两个核心问题[2]。目前已有较多的学者对粗糙集的属性约简和值约简算法进行了研究,得到了较多的成果.顾军华等[3]提出一种新的求解属性值约简算法。罗来鹏等[2]提出了基于矩阵的最简决策规则获取方法。徐凤生[4]提出一种属性与值约简及规则提取算法。鄂旭等[5]提出一种基于可辨识向量的规则提取算法。林晓斌等[6]提出的一种基于扩展差别矩阵的规则获取方法等.然而在分析大量现有的值约简算法后发现,一般的值约简算法获取的规则不够完备,另外这些方法对规则属性的删除比较随性。事实上规则处理的顺序不同,得到的规则也会不同,而基于差别矩阵的算法,构造差别矩阵的过程即耗时又会大量增加内存储空间。文献[4]提出的属性与值约简及规则提取算法,虽然不需要构造差别矩阵,但是要计算分明函数,而且当属性值比$U \times R \to V$较多时,从分明函数中求最小析取的计算过程同样比较复杂。本文基于粗糙集理论提出一种无需建立差别矩阵,无需计算分明函数就能获取最优的、完备的、无冗余的值约简算法,并通过具体的算例证明该算法正确可行。

收稿日期:2012-12-25
修回日期:2013-01-10
作者简介:樊艳英(1983-),女,讲师,主要从事人工智能、粗糙集、数据挖掘研究。
*国家自然科学基金项目(NO. 60963008),广西自然基金项目(2011GXNSFA018163)资助。

1 算法的设计思路

假设经过条件约简后的规则表为 S,U 为 S 的论域,Y 为决策属性,对规则表 S 进行值约简,则先对 U/Y 进行划分,即根据决策值先对 U 进行等价类划分。设 $P_i=\mathrm{red}-[X_j]A_j$,即 P_i 为 U/Y 集合中除掉 X_j 所隶属于的 A_j 等价类后的规则集合;再按传统的值约简算法扫描 S 中的每条规则。设计思路如下:

(1)设 red 为 S 中规则 dr(j) 的条件属性,试着去掉 red 中的任一个属性,若生成的新规则与 P_i 中的规则冲突,表示该属性不能去掉,若 dr(j) 中的任意条件属性都不能去掉,则 dr(j) 本身即为最简规则,则把它放入 K 中(K 为存放最终的规则集的表)并结束该规则的生成,进入 S 的下一条规则值约简;否则,把新规则放入表 M 中,但是此时该属性并没有真正的从 dr(j) 删除。当 dr(j) 中的每个条件属性都重复(1),则此时获得的约简规则都放入 M 中,再判断 M 中的每条规则能否再进一步约简。

(2)若能去掉 M 中每一条规则 dr(s) 的任意条件属性,而不与 P_i 中的规则冲突,并且 T 中还不存在该规则,则把新规则放入 T 中,对 M 中每条规则都重复(2),则当 M 中约简后的规则都放入 T 中,而且 T 不为空,则令 M=T,把 T 置为空,重复(2)。若 T 为空,说明 M 中的任意规则,都不能进一步化简,说明 M 本身就是规则 dr(j) 的完备最简规则,即结束 dr(j) 的值约简。对 M 中的每条规则,若该规则在 K 中还不存在,则放入 K 中,若存在,则放弃掉。若对 S 中的每条规则都重复以上的操作,则 K 中最终保存的规则就是 S 的所有规则的最终值约简结果,所得的 K 即为 S 的完备无冗余的最简值约简规则集。

2 算法的设计步骤

输入:经过属性约简后的规则集;

输出:经过值约简后的规则集。

步骤 1 设 red$=U/Y=\{A_1,A_2,\cdots,A_i\}$ //Y 为决策属性值,即对 S 根据决策属性值进行划分;

步骤 2 for(each dr(j) $\in S$) //dr(j) 为 S 中的任意一条规则,

{令 $P_i=\mathrm{red}-[X_j]A_j$ //$[X_j]A_j$ 为包含元素 X_j 的 A_j 等价类,P_i 为 U/Y 集合中除掉 X_j 所隶属于 A_j 的等价类后的规则集合。

令 flag=0, $K=\varnothing$,$M=\varnothing$,$T=\varnothing$//K 用来保存 S 中每条规则最终的约简规则集};

步骤 3 令 SU 是 dr(j) 的条件属性集,For(each $a_i\in SU$) //a_i 为 SK 中的任一属性,{若 dr(j) 试着去掉 a_i 后,生成的新规则 dr(k) 与 P_i 中其他规则不冲突,则令 flag = 1,若 M 中还不存在 dr(k),则把 dr(k) 放入 M 中,若冲突或者 M 中已有 dr(k),则不放入 M 中,但原规则的属性不删除,原规则也不从 S 中删除};

步骤 4 如果 flag = 1,转步骤 5,若 flag = 0,则把 dr(j) 放入 K 表中,转步骤 2// 说明 dr(j) 本身就是最简的,直接把它放入 K 中即可;

步骤 5 令 flag = 0, $T=\varnothing$,For(each dr(s) $\in M$)//dr(s) 为 M 中的任一规则,{令 SK 是 dr(s) 规则的条件属性集,For(each $a_j\in SK$)//a_j 为 SK 中的任一属性,

{

若规则 dr(s) 去掉 a_j 后,生成的新规则 dr(k)' 与 P_i 中的规则冲突,则不放入 T 中,若规则不与 P_i 中其他规则冲突,则令 flag = 1,把 dr(k)' 与 T 中规则比较, M 还不存在 dr(k)',则把 dr(k)' 放入 T 中,如 T 中已有此规则,则放弃}

};

步骤 6 若 flag = 0, $T=\varnothing$,则把 M 放入 K 中,转步骤 2(进入 S 中的下一条规则值约简过程),若 flag = 1,令 $M=T$, $T=\varnothing$,转步骤 5,

{

对 S 中的任一规则都重复以上操作后,若 K 有重复行,将其删除,输出 K,则最终 K 保存的即为 S 经过值约简后的完备的最简规则}。

3 算例分析

现以决策表 1(称 S 表)为例来说明新算法的可行性.

表 1 决策表

U	a	b	c	d	Y
X_1	1	1	0	2	1
X_2	1	0	0	1	1
X_3	1	0	0	0	1
X_4	1	2	1	2	0
X_5	1	2	2	2	2
X_6	2	2	2	2	2

算法执行:

$U/Y=\{\{X_1,X_2,X_3\},\{X_4\},\{X_5,X_6\}\}$。先提取 S 的第一条规则 X_1 进行分析:$P_i=\{X_4,X_5,$

$X_6\}$，X_1，$a_1b_1c_0d_2 \rightarrow Y_1$，用新算法对此规则进行值约简。

进入步骤2：

分别试着去掉 X_1 的每个条件属性 a_1，b_1，c_0，d_2，则有以下结果：

规则 dr1：$b_1c_0d_2 \rightarrow Y_1$ 与 P_i 不冲突，则 flag=1，放入 M 中；

规则 dr2：$a_1c_0d_2 \rightarrow Y_1$ 与 P_i 不冲突，则 flag=1，放入 M 中；

规则 dr3：$a_1b_1d_2 \rightarrow Y_1$ 与 P_i 不冲突，则 flag=1，放入 M 中；

规则 dr4：$a_1b_1c_0 \rightarrow Y_1$ 与 P_i 不冲突，则 flag=1，放入 M 中。

进入步骤3：在步骤3中，由于 flag=1，转步骤4：令 flag=0，$T=\varnothing$。对 M 中的每条规则即 dr1，dr2，dr3，dr4，分别执行以下操作：

(i)dr1 的每个条件属性 b_1，c_0，d_2 分别试着去掉，有如下结果：

规则 dx1：$c_0d_2 \rightarrow Y_1$ 与 P_i 不冲突，则 flag=1，把 dx1 放入 T 中；

规则 dx2：$b_1d_2 \rightarrow Y_1$ 与 P_i 冲突，把 dx2 弃掉；

规则 dx3：$b_1c_0 \rightarrow Y_1$ 与 P_i 不冲突，则 flag=1，把 dx3 放入 T 中。分别对 dr2，dr3，dr4 重复(i)，有以下结果。

由规则 dr2：$a_1c_0d_2 \rightarrow Y_1$ 产生新规则：

：$c_0d_2 \rightarrow Y_1$ 与 P_i 不冲突，但 T 已存在，需修改规则，弃掉；

$a_1d_2 \rightarrow Y_1$ 与 P_i 冲突，也弃掉；

$a_1c_0 \rightarrow Y_1$ 与 P_i 不冲突，且 T 中不存在该规则，则放入 T 中。

由规则 dx3：$a_1b_1d_2 \rightarrow Y_1$ 产生的新规则：

$b_1d_2 \rightarrow Y_1$ 与 P_i 不冲突，flag=1，且 T 中不存在该规则，放入 T 中；

$a_1d_2 \rightarrow Y_1$ 与 P_i 冲突，弃掉；

$a_1b_1 \rightarrow Y_1$ 与 P_i 不冲突，flag=1，且 T 中不存在该规则，放入 T 中。

由规则 dr4：$a_1b_2c_0 \rightarrow Y_1$ 产生的新规则：

：$b_1c_0 \rightarrow Y_1$ 与 P_i 不冲突，但 T 已经存在，弃掉；

：$a_1c_0 \rightarrow Y_1$ 与 P_i 不冲突，但 T 已经存在，弃掉；

：$a_1b_1 \rightarrow Y_1$ 与 P_i 不冲突，但 T 已经存在，弃掉。

执行完步骤4后，因为 flag=1，$T \neq \varnothing$，所以令 $M=T$，$T=\varnothing$，flag=0，再重复步骤4的下一轮，M 的每条规则产生的新规则集放入 T 中，执行的结果如下：

flag=1，此时 T 中包含新规则：

：$c_0 \rightarrow Y_1$，

：$b_1 \rightarrow Y_1$。

因此令 $M=T$，$T=\varnothing$，flag=0，重复步骤4。

很显然，继续下一轮可得 flag=0，$T=\varnothing$，所以 S 表的 X_1 生成结束。对 M 中每条规则，如果 K 中尚不存在该规则，则放入 K 中，如果已经存在，则弃掉。进入 S 的下一条规则值约简，重复以上操作，即可以得到最简规则。这说明新算法是可行的。

再利用新算法对文献[7]的表3进行值约简，所得结果如下。首先进行 U/D 的划分：

$U/D=\{(X_1,X_2),(X_3,X_4),(X_5,X_6)\}$。

规则 X_1 的值约简结果为：$a_1c_0 \rightarrow d_1$，$a_1b_1 \rightarrow d_1$；

规则 X_2 的值约简为：$a_1c_0 \rightarrow d_1$(因规则 X_1 已经有此规则所以不放入 K 表中)；$a_1b_0 \rightarrow d_1$；

规则 X_3 的值约简为：$a_0 \rightarrow d_0$；

规则 X_4 的值约简为：$b_1c_1 \rightarrow d_0$；

规则 X_5 的值约简为：$b_2 \rightarrow d_2$；

规则 X_6 的值约简为：$b_2 \rightarrow d_2$(因规则 X_5 已经有此规则所以不放入 K 表中)；$a_2 \rightarrow d_2$；$c_2 \rightarrow d_2$；

对以上规则进行整理，最终得到的规则如下：

规则一：$(a_1c_0) \vee (a_1b_1) \vee (a_1b_0) \rightarrow d_1$；

规则二：$(a_0) \vee (b_1c_1) \rightarrow d_0$；

规则三：$a_2 \vee b_2 \vee c_2 \rightarrow d_2$；

若按照文献[7]中的规则提取算法最终可能得到的规则约简如下(由于其值约简的顺序不同，得到的结果也会不同)：

规则一：$(a_1c_0) \rightarrow d_1$；(漏掉了 $(a_1b_1) \vee (a_1b_0) \rightarrow d_1$；)；

规则二：$(a_0) \vee (b_1c_1) \rightarrow d_0$；

规则三：$b_2 \rightarrow d_2$；(漏掉了 $a_2 \vee c_2 \rightarrow d_2$)。

由此可知，本文给出的算法获取的规则更加的完备，并且得到的规则也是无冗余的。

4 结束语

由于本文提出的算法在进行规则比较前，先进行 S/Y 的划分，使得每条约简后的规则只需要与本身决策值不等的规则进行比较，这样可以减少规则比较上的工作量，在决策表规模较大的情况下，可以

(下转第10页)

[5] 高铁杠，顾巧论. 一种大容量的图像可逆信息隐藏算法[J]. 光电子·激光，2008，19(5)：663-666.
[6] 李耿，熊志勇. 基于动态峰值的图像可逆水印算法[J]. 武汉理工大学学报，2010，32(7)：160-163.
[7] 赵彦涛，李志全，董宇青. 基于排序和直方图修改的可逆信息隐藏方法[J]. 光电子·激光，2010，21(1)：102-106.
[8] Tsai Piyu，Hu Yuchen，Yeh Hsiulien. Reversible image hiding scheme using predictive coding and histogram shifting[J]. Signal Processing，2009，89：1129-1143.
[9] Hong Wien，Chen Tungshou，Chang Yuping. A high capacity reversible data hiding scheme using orthogonal projection and prediction error modification[J]. Signal Processing，2010，90：2911-2922.
[10] 任洪娥，常春武，张健. 基于相邻两像素差值的无损数据隐藏算法[J]. 计算机工程与设计，2009，30(16)：3897-3902.

（责任编辑：尹 闯）

（上接第 6 页）

较大幅度提高工作效率. 算法的时间消耗主要在步骤 4，最坏的情况是每条规则可能需要反复执行多次，但是一般来讲，经过属性约简后的决策规则条件属性的个数不会太多，所以每条规则在这一步骤反复执行的次数一般不会太多。另外，由于本算法每条规则的任意一种情况都已考虑，所以获取的约简规则是完备的，又由于算法对每条规则都化简到最简的程度，因此，所得的约简规则也是最优的. 而且无论弃掉条件属性的顺序如何，最终得到的规则都是最简和最优的值约简结果.

参考文献：

[1] 徐章艳. 基于粗糙集的属性约简及其算法研究[D]. 北京：北京科技大学，2007：1-10.
[2] 罗来鹏，刘二根，王广超. 基于矩阵的最简决策规则获取[J]. 计算机工程，2008，34(19)：41-43.
[3] 顾军华，周艳聪，宋洁，等. 一种新的求解属性值约简算法[J]. 南开大学学报：自然科学版，2003，36(4)：38-42.
[4] 徐凤生. 一种属性与值约简及规则提取算法[J]. 计算机工程与科学，2008，30(2)：61-63.
[5] 鄂旭，邵良彬，杨芳，等. 一种基于可辨识向量的规则提取算法[J]. 辽宁工程技术大学学报，2010，29(5)：777-790.
[6] 林晓彬，叶东毅. 一种基于扩展差别矩阵的规则获取方法[J]. 计算机科学，2008，35(3)：231-233.
[7] 鄂旭，邵良衫，张毅智，等. 一种基于粗糙集理论的规则提取算法[J]. 计算机科学，2011，38(1)：232-235.

（责任编辑：尹 闯）