文章编号:1001 - 9081(2005)03 - 0639 - 04

一种基于决策矩阵的属性约简及规则提取算法

武志峰[1,2],吉根林[1]
(1. 南京师范大学 数学与计算机科学学院,江苏 南京 210097;
2. 石家庄经济学院 信息工程学院,河北 石家庄 050031)
(wzf_heb@sjzue.edu.cn)

摘 要:研究了 Rough 集理论中属性约简和值约简问题,扩展了决策矩阵的定义,提出了一种基于决策矩阵的完备属性约简算法,该算法利用决策属性把论域划分成多个等价类,然后利用每个等价类对应的决策矩阵计算属性约简。与区分矩阵相比,采用决策矩阵可以有效地减少存储空间,提高约简算法效率。同时,借助决策矩阵进行值约简,提出了一种新的规则提取算法,使最终得到的决策规则更加简洁。实验结果表明,本文提出的属性约简和值约简算法是正确、有效、可行的。

关键词:Rough 集;属性约简;值约简;决策矩阵;规则提取
中图分类号: TP311.13 **文献标识码**:A

Attribute reduction and rule extraction algorithms based on decision matrices

WU Zhi-feng[1,2], JI Gen-lin[1]
(1. *School of Mathematics and Computer Science, Nanjing Normal University, Nanjing Jiangsu* 210097, *China*;
2. *School of Information Engineering, Shijiazhuang University of Economics, Shijiazhuang Hebei* 050031, *China*)

Abstract: Two important issues in rough set, attribute reduction and value reduction, were discussed. The definition of extended decision matrices was presented. A novel algorithm based on extended decision matrices for attribute reduction (EDMAR) was proposed. Some equivalence classes were partitioned from the universe of objects by the decision attributes, and decision matrix for each equivalence class was created. Using the decision matrices, the attributes were reduced. Compared with algorithms based on discernibility matrices, EDMAR is of much less space complexity and time complexity. Furthermore, a new algorithm for rule extraction based on decision matrices was presented. And much more concise decision rules could be got with this method. Experimental results on the data sets in UCI machine learning repository show that the algorithms are efficient and feasible.

Key words: rough sets; attribute reduction; value reduction; decision matrices; rule extraction

0 引言

波兰数学家 Z Pawlak 于 20 世纪 80 年代初提出的 Rough 集理论是一种研究不完整、不确定知识和数据的表达、学习、归纳的理论方法。近年来,这一理论已成功应用于信息系统分析、机器学习、数据挖掘、决策支持系统、人工智能、模式识别与分类、故障检测等诸多领域[1~5]。属性约简是 Rough 集理论研究的核心内容之一。所谓属性约简指在保持信息系统的分类或决策能力不变的条件下,删除其中的冗余属性。由于属性约简是 NP-hard 问题,有效获取较优的属性约简,降低算法的时间空间复杂性,也成为 Rough 集理论研究的重点之一。目前,人们已做了许多工作,提出了多种属性约简算法。如基于正区域的属性约简算法[6,7],基于区分矩阵的属性约简算法[8,9]和基于信息熵的属性约简算法[10]等,其中 Skowron 教授提出的区分矩阵[8]方法,将信息系统中所有有关属性的区分信息都浓缩进一个矩阵中,可以很方便地得到信息系统中的属性核。但基于区分矩阵的约简算法要生成区分矩阵,时间和空间复杂度较高。文献[4]提出的区分矩阵的简化方法,虽然可以不生成区分矩阵,但却不一定能得到属性约简[11]。文献[12]在构造相对差异表的基础上,提出了属性约简的改进算法,但没有相应的值约简算法。文献[13]提出一种值约简算法,但该算法得到的规则存在属性冗余和规则冗余[14]。

本文利用决策矩阵的特性,提出一种新的属性约简和值约简算法,以提取信息表中的规则。首先扩展了决策矩阵的定义,在此基础上提出一种属性约简算法 EDMAR 和值约简算法 EDMVR。实验结果表明,算法是有效和可行的。

定义 1 一个信息系统 S 可以表示为:

$S = (U, A, V, f)$

其中,U 是对象的集合,即论域;A 是属性集合;$V = \bigcup_{a \in A} V_a$,$V_a$ 表示属性 a 的值域;$f: U \times A \rightarrow V$ 是一个信息函数,它指定 U 中每个对象 x 的属性值,即对 $x \in U$, $a \in A$,有 $f(x, a) \in V_a$。

如果属性集 A 可以分为条件属性集 C 和决策属性集 D,即 $C \cup D = A$, $C \cap D = \emptyset$,则该信息系统称为决策表或决策系统,其中 D 一般只含有一个属性。

定义 2 令决策系统为 $S = < U, A, V, f >$, $A = C \cup D$ 是属性集合,子集 $C = \{a_i \mid i = 1, \cdots, m\}$ 和 $D = \{d\}$ 分别称为条件属性集和决策属性集,$U = \{x_1, x_2, \cdots, x_n\}$ 是论域。假设 U 可被定义在 D 上的等价关系划分为 l 个概念(c_1,

收稿日期:2004 - 08 - 04 **基金项目**:国家自然科学基金资助项目(70371015)
作者简介:武志峰(1974 -),男,河北邯郸人,讲师,硕士研究生,主要研究方向:数据挖掘、知识发现; 吉根林(1964 -),男,江苏海安人,教授,主要研究方向:数据库、数据挖掘.

$c_2, \cdots, c_l$)。给定一个概念 $c \in (c_1, c_2, \cdots, c_l)$,所有属于概念 c 和不属于概念 c 的元素分别用下标 $i(i = 1, 2, \cdots, \gamma)$ 和 $j(j = 1, 2, \cdots, \rho)$ 表示。决策系统 S 中概念 c 的决策矩阵 $\boldsymbol{M}_c(S) = (\boldsymbol{M}_{ij})_{\gamma \times \rho}$,该矩阵中位置 (i, j) 元素的值为如下形式的(属性,值)对的集合:

$$\boldsymbol{M}_{ij} = \{(a, a(i)) \mid a(i) \neq a(j)\},$$
$$i = 1, 2, \cdots, \gamma, j = 1, 2, \cdots, \rho$$

1 属性约简算法 EDMAR

求所有的约简或相对约简已经被证明是 NP 完全问题。已提出的许多算法都是不完备的[7,10],不能保证一定能得到约简。我们在修改决策矩阵的基础上,提出一种基于扩展决策矩阵的完备约简算法。

1.1 扩展决策矩阵

由定义 2 可知,决策矩阵中每个元素都是属性和属性值对,这在求解属性值约简时是必要的。但当利用它求解属性约简时,属性的具体取值是无关紧要的,只需知道属性值是否相等即可。因此,我们对决策矩阵定义做一些修改,将决策矩阵的每个元素构造成:

$$\boldsymbol{M}_{ij} = \delta_1 \delta_2 \cdots \delta_m$$

其中, $\delta_k = \begin{cases} 0, & a(i) = a(j) \\ 1, & a(i) \neq a(j) \end{cases}$

$$i = 1, 2, \cdots, \gamma; j = 1, 2, \cdots, \rho; k = 1, 2, \cdots, m$$

修改后的决策矩阵实际上是把区分矩阵按决策属性的取值进行了划分,决策属性相同的对象之间不再相互比较。通常决策表中的记录(对象)个数远远大于决策属性取值个数,即 $|U| \gg |U/d|$。因此,采用决策矩阵可以减少算法所用的空间。例如对表 1 所示的决策表,其区分矩阵 $\boldsymbol{C}_D$ 和扩展决策矩阵 $\boldsymbol{M}_d(S)$ 分别为:

表 1 决策表

a	b	c	d
1	2	2	1
1	0	2	1
2	1	0	2
2	1	2	3
1	2	0	3

$$\boldsymbol{C}_D = \begin{bmatrix} 0 & 0 & 111 & 110 & 001 \\ & 0 & 111 & 110 & 011 \\ & & 0 & 001 & 110 \\ & & & 0 & 0 \\ & & & & 0 \end{bmatrix}$$

$$\boldsymbol{M}_{d=1}(S) = \begin{bmatrix} 111 & 110 & 001 \\ 111 & 110 & 011 \end{bmatrix}$$

$$\boldsymbol{M}_{d=2}(S) = [001 \quad 110]$$

另外,可以看到对求解属性约简,在计算每个决策类对应的决策矩阵时,不属于该决策类的对象数会不断减少,即扩展决策矩阵的规模会不断减小。如在计算表 1 的扩展决策矩阵 $\boldsymbol{M}_{d=2}(S)$ 时,就无需再把决策属性值 $d = 1$ 的对象作为不属于属性值 $d = 2$ 的概念处理。因为它们之间的差异已体现在扩展决策矩阵 $\boldsymbol{M}_{d=1}(S)$ 中。利用这一特性,可以按决策属性值包含的对象数决定构造每个决策类对应扩展决策矩阵的顺序,以保证最后一个扩展决策矩阵中包含最少的元素。在求解属性约简时,就可以从后向前考查每个扩展决策矩阵,以提高求解效率。

1.2 EDMAR 算法思想

首先从扩展决策矩阵中找出只有 1 位为 1 的所有元素,这些位对应的属性就是核属性,将其加入 R,删除所有扩展决策矩阵中该为 1 的元素。然后按照顺序从剩余的最小扩展决策矩阵中选择区分度最大的位(即在该扩展决策矩阵中该位为 1 的元素个数最多)对应的属性加入 R,删除所有扩展决策矩阵中该为 1 的元素,直到 R 成为一个约简。

算法描述如下:

输入 决策表 $T = < U, A, V, f >$,U 是论域,$A = C \cup D$,C 是条件属性集合,D 是决策属性

输出 决策表 T 的一个最小约简 R

1) 按决策属性值包含的对象数构造每个决策类对应的扩展决策矩阵。

2) 依次扫描所有扩展决策矩阵中的元素,若某个元素只有 1 位为 1,则该位对应的属性为核属性,加入到属性集 R 中,同时删除所有扩展决策矩阵中该位为 1 的元素。

```
for each m_c(t)
  for each m_cij
    if is( m_ij b[k] = 1; R = R ∪ k; flag = 1
if flag
  for each m_c(t)
    for each m_cij
      if contain( b ) delete m_cij
```

3) 若扩展决策矩阵中还有元素未被删除,则取规模最小的扩展决策矩阵中的元素,计算该元素中为 1 的位在整个扩展决策矩阵中的个数,取个数最多的位对应的属性加入 R,同时删除所有扩展决策矩阵中该位为 1 的元素。

```
select m_c(t)
for each m_cij
{ sum = 0;
  if δ_k ==1
    for each m_c(t)
      for each m_cij
        if δ_k ==1 sum ++;
  if max < sum max = sum; p = k;
}
R = R ∪ p;
for each m_c(t)
  for each m_cij
    if δ_k ==1 delete m_cij
```

4) 重复 3) 直到所有扩展决策矩阵中的元素全被删除为止,输出 R 即为属性约简。

该算法以相对核为起点,从每个决策类对应的扩展决策矩阵中选择区分度最大的属性加入 R,并把它从所有扩展决策矩阵中删除,最后得到的 R 中每个属性都是不可省的。故 R 一定是约简,所以该算法对于计算条件属性集的某个约简来说是完备的。

1.3 算法举例

现以文献[13]中气象预报领域的实际决策系统为例来说明算法,如表 2 所示。其中,a_1, a_2, a_3, a_4 为条件属性,*Class* 为决策属性。

利用算法 EDMAR 对表 2 进行属性约简。因为决策属性只有两个取值,故只需构造一个扩展决策矩阵 $\boldsymbol{EM}_{d=N}$。

$$\boldsymbol{EM}_{d=N} = \begin{bmatrix} 1000 & 1100 & 1100 & 1111 & 0110 & 1110 & 0111 & 1101 & 1010 \\ 1001 & 1101 & 1111 & 1110 & 0111 & 1111 & 0110 & 1100 & 1011 \\ 1111 & 0111 & 0001 & 1000 & 1001 & 0111 & 1100 & 1110 & 1101 \\ 1100 & 1000 & 1110 & 1111 & 0110 & 1010 & 0011 & 1001 & 1110 \\ 1101 & 0001 & 0111 & 1110 & 1111 & 0011 & 1010 & 1000 & 1111 \end{bmatrix}$$

由扩展决策矩阵 $\boldsymbol{EM}_{d=N}$ 可知,元素 m_{11},和 m_{33} 中只有 1 位为 1,因此它们对应的属性为核属性,即 $\{a_1, a_4\}$ 为核属性。将矩阵中包含核属性的元素删除,得到如下矩阵:

$$EM_{d=N}=\begin{bmatrix}\cdots & \cdots & \cdots & \cdots & 0110 & \cdots & \cdots & \cdots & \cdots\\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0110 & \cdots & \cdots\\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\ \cdots & \cdots & \cdots & \cdots & 0110 & \cdots & \cdots & \cdots & \cdots\\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\end{bmatrix}$$

因为在整个决策矩阵中 a_2 为 1 的个数和 a_3 为 1 的个数都是 3,我们取 a_2 加入 R,然后将矩阵中包含 a_2 的元素删除。此时矩阵中所有元素均已被删除,因此得到属性约简集 $\{a_1,a_2,a_4\}$。当然如选择 a_3 加入 R,则得到属性约简集 $\{a_1,a_3,a_4\}$。

表 2　一个实际的决策系统

U	Condition Attribute(C)				Decision Attribute(d)
	Outlook(a_1)	Temperature(a_2)	Humidity(a_3)	Windy(a_4)	*Class*
1	Sunny	Hot	High	False	N
2	Sunny	Hot	High	True	N
3	Overcast	Hot	High	False	P
4	Rain	Mild	High	False	P
5	Rain	Cool	Normal	False	P
6	Rain	Cool	Normal	True	N
7	Overcast	Cool	Normal	True	P
8	Sunny	Mild	High	False	N
9	Sunny	Cool	Normal	False	P
10	Rain	Mild	Normal	False	P
11	Sunny	Mild	Normal	True	P
12	Overcast	Mild	High	True	P
13	Overcast	Hot	Normal	False	P
14	Rain	Mild	High	True	N

2　值约简算法 EDMVR

Rough 集理论具有从决策表中抽取决策规则的能力,事实上,从 Rough 集理论中抽取决策规则的过程就是对决策表进行值约简的过程。属性约简后,可以利用该属性集对应的决策表,进行值约简,以提取决策规则。文献[14]指出利用文献[13]中提出的值约简算法得到的规则,但仍存在属性冗余和规则冗余,并给出了反例。本文利用扩展决策矩阵,提出一种新的值约简算法,使得到的规则更进一步简化。

2.1　算法思想

首先对每个决策类构造对应的扩展决策矩阵,与属性约简不同的是扩展决策矩阵不会减小,因为每个决策类都有其对应的决策规则。利用与属性约简相似的方法,从扩展决策矩阵中找出每条记录中只有 1 位为 1 的所有元素,这些位对应的属性值就是该条规则的核值,将其加入 S,删除该记录中该为 1 的元素。然后按照顺序从该记录剩余的元素中选择区分度最大的位(即在该记录中该位为 1 的元素个数最多)对应的属性值加入 S,删除该记录中该位为 1 的元素,直到记录中所有元素均已删除,S 即为一条规则。

算法描述如下:

输入　决策表 $T=<U,A,V,f>$,U 为论域,$A=R\cup D$,R 是约简后的属性集合,D 是决策属性

输出　决策表 T 的规则集 S

1) 按决策属性值构造每个决策类对应的扩展决策矩阵。

2) 对每个扩展决策矩阵中的每一行作如下处理:

① 扫描该行所有元素,若某个元素只有 1 位为 1,则该位对应的属性值为核值,加入到集 S 中,同时删除该行中该位为 1 的所有元素。

② 若该行还有元素未被删除,计算所有未删除元素中为 1 的位的个数,取个数最多的位对应的属性值加入 S,同时删除该行中所有该位为 1 的元素。

③ 重复 ② 直到该行所有元素均已删除。

④ 输出 S 中的属性值对,即为一条规则。

⑤ 删除该扩展决策矩阵中包含该规则的所有元素。

2.2　算法举例

仍以气象预报领域的实际决策系统[13]为例来提取规则,取 $\{a_1,a_2,a_4\}$ 为属性约简集,利用算法 EDMVR 对其进行属性值约简。

首先,构造扩展决策矩阵 EM 为:

$$EM_{d=N}=\begin{bmatrix}100 & 110 & 110 & 111 & 010 & 110 & 011 & 111 & 100\\ 101 & 111 & 111 & 110 & 011 & 111 & 010 & 110 & 101\\ 111 & 011 & 001 & 100 & 101 & 011 & 110 & 110 & 111\\ 110 & 100 & 110 & 111 & 010 & 100 & 001 & 101 & 110\\ 111 & 001 & 011 & 110 & 111 & 001 & 100 & 100 & 111\end{bmatrix}$$

$$EM_{d=P}=\begin{bmatrix}100 & 101 & 111 & 110 & 111\\ 110 & 111 & 011 & 100 & 001\\ 110 & 111 & 001 & 110 & 011\\ 111 & 110 & 100 & 111 & 110\\ 010 & 011 & 101 & 010 & 111\\ 110 & 111 & 011 & 100 & 001\\ 011 & 010 & 110 & 001 & 100\\ 111 & 110 & 110 & 101 & 100\\ 100 & 101 & 111 & 110 & 111\end{bmatrix}$$

然后逐行检查每个决策矩阵,生成决策规则。如检查决策矩阵 $EM_{d=N}$ 的第一行,可知在 m_{11},m_{15} 和 m_{19} 中只有一位为 1,则记下其对应的属性值对,即 a_1 = Sunny, a_2 = Hot,同时删除该行中 a_1 或 a_3 为 1 的其他元素。此时该行所有元素都已被删除,所以得到决策规则为:

1) (a_1, Sunny) and (a_2, Hot)$\Rightarrow$*Class* = *N*

删除 $EM_{d=N}$ 中所有包含该规则的元素,得到如下决策矩阵:

$$EM_{d=N}=\begin{bmatrix}\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\ 101 & \cdots & \cdots & \cdots & 011 & \cdots & 010 & \cdots & 101\\ 111 & 011 & 001 & 100 & 101 & 011 & 110 & 110 & 111\\ 110 & 100 & 110 & 111 & 010 & 100 & 001 & 101 & 110\\ 111 & 001 & 011 & 110 & 111 & 001 & 100 & 100 & 111\end{bmatrix}$$

类似地,可得到其余决策规则如下:

2) (a_1, Rain) and (a_4, True)⇒*Class* = *N*

3) (a_1, Sunny) and (a_2, Mild) and (a_4, False)⇒*Class* = *N*

4) (a_1, Overcast)⇒*Class* = *P*

5) (a_1, Rain) and (a_4, False)⇒*Class* = *P*

6) (a_2, Cool) and (a_4, False)⇒*Class* = *P*

7) (a_1, Sunny) and (a_2, Mild) and (a_4, True)⇒*Class* = *N*

该规则集与文献[13]中得到的规则集一样都是7条规则,其中前6条规则与文献[13]中的一致,而第7条规则与文献[13]中的不同。经分析文献[13]中第7条规则是错误的,会导致记录11和记录14冲突。

3 实验结果

为验证本文提出的EDMAR算法的有效性,采用UCI机器学习数据库中的数据集进行测试。对该数据集中的11个决策表分别采用文献[9]中的算法与EDMAR算法进行属性约简,结果如表3所示。从表3可知,EDMAR算法一定能得到某个约简,同时在效率上优于传统基于区分矩阵的约简算法。实验结果表明EDMAR算法是正确、有效,可行的。对于大数据集的约简在一定程度上简化了计算,提高了运算速度。

表3 约简算法比较

决策表	实例数	原条件属性数	文献[9]的约简算法			EDMAR算法		
			约简后条件属性数	是否约简	执行时间/s	约简后条件属性数	是否约简	执行时间/s
文献[6]中的决策表	36	5	3	否	0.00	2	是	0.00
The Monk's Problems(1)	432	6	3	是	0.05	3	是	0.02
The Monk's Problems(2)	432	6	6	是	0.06	6	是	0.03
The Monk's Problems(3)	432	6	3	是	0.05	3	是	0.02
Balance-scale	625	4	4	是	0.09	4	是	0.04
Voting Records Database	435	16	9	是	0.09	9	是	0.05
Tic-Tac_Toe Endgame Database	958	9	8	是	0.36	8	是	0.29
Solar Flare Database	1066	12	10	是	0.42	10	是	0.31
Mushroom Database(1)	1637	22	4	是	1.55	4	是	1.23
Mushroom Database(2)	2346	22	5	否	39.35	4	是	1.26
Chess End-Game	3196	36	29	是	618.19	29	是	252.67

为验证值约简算法EMDVR的有效性,以UCI数据库中的两个决策表为测试数据集,采用波兰华沙大学与挪威科技大学联合开发的Rosetta软件与EDMVR算法作对比测试。将每个决策表分为训练集和测试集两部分,训练集的记录随机选择,但仍保持各决策类在总数据中的比例,其余的记录作为测试集。实验结果如表4所示。

从表4可知,和Rosetta软件相比,用EDMVR算法得到的决策规则无论是数量上还是长度上都要简洁得多,同时对测试集的分类正确率也比Rosetta软件高。因此,EDMVR算法是有效和可行的。

表4 值约简算法结果比较

决策表	训练集/测试集	Rosetta软件			EDMVR算法		
		规则数	平均属性数	正确率(%)	规则数	平均属性数	正确率(%)
The Monk's	310/122	36	3	100	22	2.7	100
Problems	370/62	36	3	100	22	2.7	100
Voting Records	300/135	106	7	91.1	36	2.7	92.6
Database	350/85	149	8	87.1	43	2.5	90.6

参考文献:

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

[2] PAWLAK Z. Rough Sets: Theoretical Aspects of Reasoning about Data[M]. Dordrecht: Kluwer Academic Publishers, 1991.

[3] SKOWRON A. Rough Sets and Boolean Reasoning[A]. WED P. Cranular Computiong: An Emerging Paradgm[C]. New York: Physica-Verlag, 2001. 95 - 124.

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

[5] 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.

[6] 刘少辉,盛秋戬,吴斌, 等. Rough集高效算法的研究[J]. 计算机学报,2003,26(5):524 - 529.

[7] HU XH, CERCONE N. Learning in relational databases: A rough set approach [J]. International Journal of Computational Intelligence, 1995, 11(2): 323 - 338.

[8] SKOWRON A, RAUSZER C. The discernibility matrices and functions in information system[A]. Intelligent Decsion Support Handbook of Applications and Advances of the Rough Sets Theory[C]. Dordrecht: Kluwer Academic Publishers, 1992. 331 - 338.

[9] 王珏,王任,苗夺谦, 等. 基于Rough Set理论的"数据浓缩"[J]. 计算机学报,1998,21(5):393 - 399.

[10] 苗夺谦,胡桂荣. 知识约简的一种启发式算法[J]. 计算机研究与发展,1999,36(6):681 - 684.

[11] 庞彦军, 刘开第. 计算约简的差别矩阵简化算法不成立[J]. 系统工程理论与实践,2004,24(2):142 - 144.

[12] 潘丹,郑启伦. 属性约简自寻优算法[J]. 计算机研究与发展,2001,38(8): 904 - 910.

[13] 常犁云, 王国胤,吴渝. 一种基于Rough Set理论的属性约简及规则提取方法[J]. 软件学报,1999,10(11):1206 - 1211.

[14] 林嘉宜,彭宏,郑启伦. 一种新的基于粗糙集的值约简算法[J]. 计算机工程, 2003,29(4):70 - 71.