一种使用二进制差别矩阵的属性约简方法

王治和, 杜 跃, 张小侠

(西北师范大学 数学与信息科学学院, 甘肃 兰州 730070)

[摘要] 针对区分矩阵属性约简算法中区分矩阵存在空值元素和重复元素等缺点, 提出了一种基于二进制差别矩阵的属性约简算法. 该算法不仅保证了属性约简的完整性和正确性, 同时也降低了运算所需的时间和空间.

[关键词] 区分矩阵, 二进制差别矩阵, 属性约简

[中图分类号] TP182 [文献标识码] A [文章编号] 1672-1292(2010)03-0056-04

A Method of Attributes Reduction Using the Binary Discernibility Matrix

Wang Zhihe, Du Yue, Zhang Xiaoxia

(College of Mathematics and Information Science, Northwest Normal University, Lanzhou 730070, China)

Abstract Aiming at the shortages of discernibility matrix that contains null elements and duplicate elements, a method of attributes reduction based on binary discernibility matrix was presented in this paper. It not only guarantees the integrity and accuracy of the attributes reduction results, but also cuts down the time and space of the operation.

Key words discernibility matrix, binary discernibility matrix, attributes reduction

区分矩阵(也称可辨识矩阵或分明矩阵)是由斯科龙(Skowron)教授提出的. 利用区分矩阵(discernibility matrix)来表达知识有很多优点, 特别是它能容易地计算约简和核[1,2]. 基于区分矩阵的属性约简算法虽然可以得到决策表的所有可能的属性约简结果, 但是有两个瓶颈影响了算法的性能:

(1) 处理空值元素会造成极大的空间浪费;

(2) 属性约简时化简逻辑公式的计算量很大.

针对基于区分矩阵属性约简算法的这两个缺点, 本文提出一种新的基于二进制差别矩阵的属性约简算法.

1 预备知识和相关研究工作

1.1 基本概念

定义 1 令决策表为 $IS = (U, A, V, f)$, $A = C \cup D$, $C \cap D = \varnothing$, $C = \{c_1, c_2, \cdots, c_m\}$ 为条件属性集, $D = \{d\}$ 为决策属性集, $U = \{x_1, x_2, \cdots, x_n\}$ 是论域, $c_i(x_j)$ 是样本 x_j 在属性 c_i 上的取值. C_{ij} 表示区分矩阵中第 i 行第 j 列的元素, 则区分矩阵 $\boldsymbol{C}_D$ 中的元素定义为:

$$C_{ij} = \begin{cases} \{c_k \mid c_k \in C \text{ 且 } c_k(x_i) \neq c_k(x_j)\}, & d(x_i) \neq d(x_j); \\ 0, & d(x_i) = d(x_j). \end{cases} \quad (1)$$

其中, $i, j = 1, \cdots, n$.

显然区分矩阵是一个依主对角线对称的矩阵, 在考虑区分矩阵的时候, 只需要考虑其上三角(或下三角)部分就可以了.

根据区分矩阵的定义可知, 当两个样本的决策属性取值相同时, 它们所对应的区分矩阵元素的取值为 0; 当两个样本的决策属性不同且可以通过某些条件属性的取值不同加以区分时, 它们所对应的区分矩阵

收稿日期: 2010-06-28

基金项目: 西北师范大学 2007-2010年度重点学科基金(2007C04).

通讯联系人: 王治和, 教授, 研究方向: 数据挖掘. E-mail: wangzh@nwnu.edu.cn

元素的取值为这两个样本属性值不同的条件属性集合, 即可以区分这两个样本的条件属性集合; 当两个样本发生冲突时, 即所有的条件属性取值相同而决策属性的取值不同时, 则它们所对应的区分矩阵中的元素取值为空集.

定义 **2** 区分函数 (discernibility function) 定义为: $\Delta = \wedge (\vee C_{ij})$.

1.2 约简算法

基于区分矩阵的约简算法的主要思想是: 首先利用区分矩阵导出区分函数, 然后求解区分函数的析取范式, 该范式中的每一个析取项即为系统的一个约简[3-5].

Step 1 计算决策表的区分矩阵 $\boldsymbol{C}_D$;

Step 2 对于区分矩阵中的所有取值为非空集合的元素 C_{ij} ($C_{ij} \neq 0$, $C_{ij} \neq \cong$), 建立相应的析取逻辑表达式 $L_{ij} = \bigvee_{c_i \in C_{ij}} c_i$;

Step 3 将所有的析取逻辑表达式 L_{ij} 进行合取运算, 得到一个合取范式 L, 即 $L = \bigwedge_{C_{ij} \neq 0, C_{ij} \neq \cong} L_{ij}$;

Step 4 将合取范式 L 转换为析取范式的形式, 得 $L' = \bigvee_i L_i$;

Step 5 输出属性约简结果. 析取范式中的每个合取项就对应一个属性约简的结果, 每个合取项中所包含的属性组成约简后的条件属性集合.

1.3 实例分析

例 **1** 对于表 1所示的决策表系统, 首先可以得到其区分矩阵如表 2所示.

表 **1** 决策表系统

Table 1 Decision table

U	Condition Attributes				Decision Attribute
	c_1	c_2	c_3	c_4	d
1	1	1	1	2	1
2	1	2	2	2	1
3	1	1	1	1	1
4	2	2	1	2	1
5	3	1	1	3	1
6	2	2	3	2	1
7	2	2	1	3	2
8	2	1	3	2	2
9	3	1	1	2	2

表 **2** 区分矩阵

Table 2 Discernibility matrix

1	2	3	4	5	6	7	8	9
1								
2								
3								
4								
5								
6								
7	$c_1c_2c_4$	$c_1c_3c_4$	$c_1c_2c_4$	c_4	c_1c_2	c_3c_4		
8	c_1c_3	$c_1c_2c_3$	$c_1c_3c_4$	c_2c_3	$c_1c_3c_4$	c_2		
9	c_1	$c_1c_2c_3$	c_1c_4	c_1c_2	c_4	$c_1c_2c_3$		

进而可以得到 18个析取逻辑表达式: $L_{17} = c_1 \vee c_2 \vee c_4$, $L_{18} = c_1 \vee c_3$, $L_{19} = c_1$, $L_{27} = c_1 \vee c_3 \vee c_4$, $L_{28} = c_1 \vee c_2 \vee c_3$, $L_{29} = c_1 \vee c_2 \vee c_3$, $L_{37} = c_1 \vee c_2 \vee c_4$, $L_{38} = c_1 \vee c_3 \vee c_4$, $L_{39} = c_1 \vee c_4$, $L_{47} = c_4$, $L_{48} = c_2 \vee c_3$, $L_{49} = c_1 \vee c_2$, $L_{57} = c_1 \vee c_2$, $L_{58} = c_1 \vee c_3 \vee c_4$, $L_{59} = c_4$, $L_{67} = c_3 \vee c_4$, $L_{68} = c_2$, $L_{69} = c_1 \vee c_2 \vee c_3$.

将这些表达式进行合取得到合取表达式 L:

$L = L_{17} \wedge L_{18} \wedge L_{19} \wedge L_{27} \wedge L_{28} \wedge L_{29} \wedge L_{37} \wedge L_{38} \wedge L_{39} \wedge L_{47} \wedge L_{48} \wedge L_{49} \wedge L_{57} \wedge L_{58} \wedge L_{59} \wedge L_{67} \wedge L_{68} \wedge L_{69} = (c_1 \vee c_2 \vee c_4) \wedge (c_1 \vee c_3) \wedge c_1 \wedge (c_1 \vee c_3 \vee c_4) \wedge (c_1 \vee c_2 \vee c_3) \wedge (c_1 \vee c_2 \vee c_3) \wedge (c_1 \vee c_2 \vee c_4) \wedge (c_1 \vee c_3 \vee c_4) \wedge (c_1 \vee c_4) \wedge c_4 \wedge (c_2 \vee c_3) \wedge (c_1 \vee c_2) \wedge (c_1 \vee c_2) \wedge (c_1 \vee c_3 \vee c_4) \wedge c_4 \wedge (c_3 \vee c_4) \wedge c_2 \wedge (c_1 \vee c_2 \vee c_3) = c_1 \wedge c_2 \wedge c_4$. 因此, 这个决策表系统条件属性约简结果为 $\{c_1c_2c_4\}$ (如表 3所示).

表 **3** 约简结果

Table 3 Reduction results

U	Condition Attributes			Decision Attribute
	c_1	c_2	c_4	d
1	1	1	2	1
2	1	2	2	1
3	1	1	1	1
4	2	2	2	1
5	3	1	3	1
6	2	2	2	1
7	2	2	3	2
8	2	1	2	2
9	3	1	2	2

2 基于二进制差别矩阵的属性约简算法

区分矩阵中的元素取值为两个样本属性值不同的条件属性集合, 在对析取逻辑表达式 L_{ij} 进行合取运算时, 需要将这些条件属性符号转化成二进制运算, 最后再将运算结果转换成条件属性符号, 如果在建立

区分矩阵时直接采用二进制,也就是说用二进制的 0、1来区分两个样本的各条件属性值是否相等,这样计算时会减少计算时间.

2.1　算法描述及实现

Input　决策表

Output　决策表的约简属性

Step 1　求出二进制差别矩阵

定义 **3**　二进制差别矩阵中的每个元素定义为[3-5]:

$$b_S = \beta_1\beta_2\cdots\beta_m,\ \beta_p = \begin{cases} 0, & x(i) = x(j) \cap d(i) \neq d(j) \\ 1, & x(i) \neq x(j) \cap d(i) \neq d(j) \end{cases} \quad (2)$$

其中, $i = 1, 2, \cdots, n$; $j = 1, 2, \cdots, n$; $p = 1, 2, \cdots, m$.

```
for( i = 0; i < n; i++ )   //n为决策表系统中样本的个数
  for(j = 0; j < n; j++ )   //n为决策表系统中样本的个数
    for(p = 0; p < m; p++ )   //m为决策表系统中条件属性的个数; for运算用于求出二进制差别矩阵中的元素
      if(x(i,p) = x(j,p))  βp = 1;
      else βp = 0;
    bS = β1β2…βm;
    if( bS ∈ B ) add bS to B;   //二维数组 B用于存放二进制差别矩阵中的非重复元素
```

Step 2　求决策表系统的相对属性核

考察区分矩阵,不难发现如果矩阵中存在一个元素,其取值为包含单属性元素的集合,则表明该属性是区分这个矩阵元素所对应的两个样本所必须的属性,也是唯一能够区分这两个样本的属性. 区分矩阵中的这些元素所包含的属性组成的属性集合其实就是该决策表系统的相对属性核[1]. 针对本文提出的二进制差别矩阵,其中包含只有一位为 1的元素. 这些元素为 1的位对应的属性所组成的集合即为决策表系统的相对属性核.

Step 3　求出约简属性集

```
While(B ≠ ∅)
  {max = 0;
    for( j = 0; j < m; j++ )   //m为数组中列的个数
      { sum = 0;
        for( i = 0; i < w; i++ )   //w为数组中行的个数
          if(B[i][j] == 1) sum = sum + 1;
        if (max < sum)
          { max = sum; p = j; R = R ∪ j }
        }
        for( i = 0; i < w; i++ )
          { if(B[i][p] == 1) delete B[i];   //删除第 i行元素
            w = w - 1;
          }
        }
```

删除二进制差别矩阵中核属性位为 1的行后,如果矩阵非空,则计算剩余行各位为 1的个数,选取个数最多(区分度最大)的位所对应的属性添加到属性集中,并删除差别矩阵中该位为 1的行. 重复这个操作直至差别矩阵为空为止,即可得约简属性集 R.

2.2　算法演示

现以表 1的决策表系统来说明算法. 根据二进制差别矩阵的定义,决策表(表 1)的二进制差别矩阵如表 4所示.

在这个二进制差别矩阵中,第三、第九和第五十行中只有一位元素为 1,根据上节算法分析可知条件属性 $c_1c_2c_4$是核值属性. 删除 $c_1c_2c_4$位为 1的行,此时差别矩阵为空. 至此,属性约简结束. 决策表的条件属性可约简为 $\{c_1c_2c_4\}$,这与区分函数的约简结果是一样的. 与区分矩阵相比采用二进制决策矩阵可以有效

地减少存储空间, 尤其对于得到稀疏区分矩阵的决策表系统来说存储空间的减少会更加明显.

表 4 二进制差别矩阵

Table 4 Binary discernibility matrix

B	1	2	3	4	5	6	7	8	9	10	11	12
$c_1c_2c_3c_4$	1101	1011	0001	1100	0011	1010	1110	0110	0100	1000	1001	1100

2.3 算法性能分析

2.3.1 算法的正确性

根据上一节对基于二进制差别矩阵的属性约简算法的详细描述和分析, 可以认为本文算法是对原有基于区分矩阵的约简算法进行认真消化吸收与扩展的基础上提出的, 因此算法的正确性是有理论保证的. 此外对同一决策表系统进行约简, 所得到的约简结果是一致的. 结合这两方面的分析可以说明本文算法得到决策表的正确约简结果.

2.3.2 影响算法效率的因素

根据前面的介绍和上节中对基于二进制差别矩阵的约简算法的具体描述, 可以分析出约简过程中影响算法效率的重要因素如下:

与通常的基于区分矩阵的属性约简算法相比, 本文算法的 Step 1可以减少矩阵的存储空间, 即只存储区分矩阵中非 0且非空的不重复元素. 再者约简时采用二进制运算大大提高约简算法效率, 所以本文算法是在空间复杂度和时间复杂度两方面得到了改进.

3 实验结果

为验证本文算法的效率和性能, 从 UCI及其学习数据库中选取 5个离散型数据库使用本文和基于区分矩阵的属性约简算法进行了实验比较, 结果如表 5 从表 5实验数据中可以看出本文算法在执行时间上优于基于区分矩阵的属性约简算法.

表 5 算法比较

Table 5 Algorithm comparison

决策表	原条件属性数	基于区分矩阵的属性约简算法		本文算法	
		约简后条件属性数	执行时间 /s	约简后条件属性数	执行时间 /s
Balance Scale	4	4	0.09	4	0.07
Tie-Tae-Toe	9	8	0.36	8	0.32
Mushroom	22	4	1.55	4	1.24
Solar	12	10	0.42	10	0.4
Voting database	16	9	0.68	9	0.52

4 结语

属性约简是 Rough集理论研究的核心内容之一, 由于属性约简的不唯一性, 使得找出一个决策表的最小约简是个 NP-hard问题[1]. 通过对区分矩阵的特性进行分析后, 可知区分矩阵中存在 0值元素、空集和重复元素, 这些元素在求解属性约简的过程中不起作用, 删除这些元素可以起到降低运算时间的效果. 因此提出一种新的基于二进制差别矩阵的属性约简算法, 实验结果表明, 本文提出的算法不仅是正确的, 而且是有效可行的.

[参考文献] (References)

[1] 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.

Wang Guoyin. Rough set theory and knowledge acquisition[M]. Xi'an: Xi'an Jiaotong University Press, 2001. (in Chinese)

(下转第 73页)

[参考文献] (References)

[1] Datta A, Franklin J, Garg D, et al. A logic of secure systems and its application to trusted computing[J]. Security and Privacy, IEEE Symposium, 2009, 30: 221-236.

[2] Munoz A, Mana A, Serrano D. The role of trusted computing in secure agent migration[C]// Research Challenges in Information Science, Third International Conference: Fez, 2009: 255-264.

[3] Glas B, Klimm A, Muller-Glaser K D, et al. Configuration measurement for FPGA-based trusted platforms[C]// Proceedings of the 2009 IEEE/IFIP International Symposium on Rapid System Prototyping. Washington DC: IEEE Computer Society, 2009: 123-129.

[4] Zhu Lu, Yu Sheng, Zhang Xing, et al. Formal compatibility model for trusted computing applications[J]. Wuhan University Journal of Natural Sciences, 2009, 14(5): 388-392.

[5] 龚敏明, 石志国. 可信计算及其安全性应用研究综述[J]. 江西师范大学学报: 自然科学版, 2009, 33(3): 348-352.
Gong Minming, Shi Zhiguo. The research survey of trusted computing and its application of security[J]. Journal of Jiangxi Normal University: Natural Science Edition, 2009, 33(3): 348-352. (in Chinese)

[6] 郑志明, 马世龙, 李未, 等. 软件可信性动力学特征以其演化复杂性[J]. 中国科学 (F辑): 信息科学, 2009, 39(9): 946-950.
Zheng Zhiming, Ma Shilong, Li Wei, et al. Kinetics characteristic and complexity of evolution of software trustworthiness[J]. Science in China(F): Information Science Edition, 2009, 39(9): 946-950. (in Chinese)

[责任编辑: 严海琳]

(上接第 59页)

[2] Skowron A. Extracting laws from decision tables: A Rough set approach[J]. Computational Intelligence, 1995, 11(47): 371-388.

[3] 王锡淮, 张腾飞, 肖健梅. 基于二进制可辨矩阵的决策规则约简算法[J]. 计算机工程与应用, 2007, 43(27): 178-180.
Wang Xihuai, Zhang Tengfei, Xiao Jianmei. Algorithm for decision rules based on binary discernibility matrix[J]. Computer Engineering and Application, 2007, 43(27): 178-180. (in Chinese)

[4] 桂现才. 简化的二进制差别矩阵属性约简算法的改进[J]. 计算机工程与设计, 2007, 28(16): 3971-3973.
Gui Xiancai. Improved algorithm for attribute reduction based on simple binary discernibility matrix[J]. Computer Engineering and Design, 2007, 28(16): 3971-3973. (in Chinese)

[5] 程京, 朱靖, 张帆. 一个基于差别矩阵的属性约简改进算法[J]. 湖南大学学报: 自然科学版, 2009, 36(4): 85-88.
Cheng Jing, Zhu Jing, Zhang Fan. An updated algorithm for attribute reduction based on discernibility matrix[J]. Journal of Hunan University: Natural Science Edition, 2009, 36(4): 85-88. (in Chinese)

[责任编辑: 严海琳]