文章编号: 1001-0920 (2013) 04-0563-06

垂直划分二进制可分辨矩阵的属性约简

杨传健[1a], 葛 浩[1b,2b], 李龙澍[2a]

(1. 滁州学院 a. 计算机与信息工程学院，b. 机械与电子工程学院，安徽 滁州 239012；2. 安徽大学 a. 计算机科学与技术学院，b. 计算智能与信号处理教育部重点实验室，合肥 230039)

摘 要: 针对二进制可分辨矩阵属性约简方法在处理大数据集时的不足, 首先给出两种二进制可分辨矩阵属性约简的定义, 并证明这两个属性约简定义与正区域的属性约简定义是等价的; 然后, 给出对二进制可分辨矩阵按条件属性垂直划分后进行属性约简的方法; 为了进一步降低空间开销, 提出将垂直分解的二进制可分辨矩阵存于外部介质中, 在约简过程中, 仅将所需部分调入内存, 由此设计启发式属性约简算法, 其时间和空间复杂度的上界分别为 $O(|C||U|^2)$ 和 $O(|U|^2)$; 最后, 理论分析和实验结果验证了该算法的正确性和高效性.

关键词: 粗糙集；可分辨矩阵；二进制可分辨矩阵；属性约简

中图分类号: TP181 **文献标志码:** A

Attribute reduction of vertically partitioned binary discernibility matrix

YANG Chuan-jian[1a], *GE Hao*[1b,2b], *LI Long-shu*[2a]

(1a. School of Computer and Information Engineering，1b. School of Mechanical and Electronic Engineering，Chuzhou University，Chuzhou 239012，China；2a. School of Computer Science and Technology，2b. Key Laboratory of Computation Intelligence and Signal Processing of Education Ministry，Anhui University，Hefei 230039，China. Correspondent：YANG Chuan-jian，E-mail：tocjy474@126.com)

Abstract: Attribute reduction algorithms based on binary discernibility matrix are disadvantageous to the larger database sets. To overcome above shortcoming, firstly, the two definitions of attribute reduction based on binary discernibility matrix are proposed. It is proved that attribute reductions acquired from the definitions are all equivalent to the attribute reduction based on positive region. Then the method of attribute reduction is present, which is based on the vertically partitioned binary discernibility matrix. In order to decrease the express of space, the partitioned binary attribute columns are all stored on the external space. In the process of reduction, essential part is transferred into the memory merely. Based above, a heuristic attribute reduction algorithm is designed, in which upper bounds of the time and space complexity are $O(|C||U|^2)$ and $O(|U|^2)$ respectively. Finally, both of theoretical analysis and experimental results show that the algorithms are correct and efficient.

Key words: rough set；discernibility matrix；binary discernibility matrix；attribute reduction

0 引 言

粗糙集理论[1]是波兰数学家 Pawlak 教授于 1982 年提出的一种处理含糊和不精确性知识的数学工具, 它能有效地分析和处理不精确、不一致、不完备的信息, 并从海量数据中发现隐含的知识. 属性约简是粗糙集理论的核心内容之一, 常用的属性约简算法有: 基于信息熵的方法[2], 基于正区域的方法[3]和基于可分辨矩阵的方法[4-12].

Hu[5]根据 Skowron 可分辨矩阵[4]提出了一种属性约简算法, 算法的时间和空间复杂度分别为 $O(|C|^2|U|^2)$ 和 $O(|C||U|^2)$, 并不理想; 在对不一致决策表进行处理时, 该算法求得的约简与正区域算法求得的约简不一致. 支天云等[8]给出一种基于二进制可分辨矩阵的属性约简算法, 该方法可以减少一些空间开销, 但其时间复杂度为 $O(|C|^2+|U|^4)$, 空间复杂度为 $O(|C||U|^2)$, 时空效率仍不理想, 并且同样无法正确处理不一致决策表. 徐章艳等[9]提出了简化的二进制可分辨矩阵的约简算法, 该算法解决了

收稿日期: 2011-12-19；**修回日期:** 2012-03-25.

基金项目: 安徽省自然科学基金项目(090412054)；安徽省高等学校自然科学研究项目(KJ2012A212,KJ2011Z276)；安徽省高等学校优秀青年人才基金项目(2011SQRL123)；滁州学院科学研究项目(2010kj014B, 2011kj003Z).

作者简介: 杨传健(1978−), 女, 副教授, 硕士, 从事数据挖掘、粗糙集的研究；李龙澍(1956−), 男, 教授, 博士生导师, 从事不精确信息处理、智能软件等研究.

因决策表不一致造成的正区域约简和二进制可分辨矩阵约简不一致的问题; 由于对决策表进行了简化, 使算法的时间和空间复杂度分别降为$\max\{O(|C|^2\times(|U'_{\rm pos}||U/C|)), O(|C||U|)\}$和$\max\{O(|C||U'_{\rm pos}||U/C|)), O(|C||U|)\}$, 但该算法并不是完备的约简, 并且在处理大数据集时空间开销仍是瓶颈. 杨萍等[10] 提出了利用二进制可分辨矩阵中属性的区分度和区分率进行属性约简, 该算法可以保证获得最小属性约简, 但区分度和区分率的计算无疑增加了算法的开销, 而且时间和空间复杂度并没有得到本质的降低.

基于二进制可分辨矩阵的属性约简算法的空间复杂度为$O(|C||U|^2)$, 虽然有些算法对决策表简化后, 构建了简化的二进制可分辨矩阵, 减少了一些空间开销, 但空间复杂度没有根本性地改进, 仍高达$O(|C||U/C|^2)$; 尤其在处理多条件属性、多样本对象的大数据集时, 基于二进制可分辨矩阵的属性约简方法仍显得力不从心. 对此, 本文首先给出两个基于二进制可分辨矩阵属性约简的定义, 并证明由这两个定义获得的约简与正区域的约简是等价的. 然后, 给出将二进制可分辨矩阵垂直划分成不同的二进制属性列后, 进行属性约简的方法; 为了进一步降低空间开销, 提出将垂直分解的二进制属性存储于外部介质中, 在属性约简过程中仅将所需运算的二进制属性列调入内存, 由此设计反向启发式属性约简算法, 算法的时间和空间复杂度分别为$O(|C||U|^2)$和$O(|U|^2)$.

1 决策表及其二进制可分辨矩阵

决策表$S=(U,A,V,f)$. 其中: U为论域, 是对象的有限集; A为属性集, $A=C\bigcup D$且$C\bigcap D=\varnothing$, C为条件属性集, D为决策属性集; $V=\bigcup\limits_{a\in A}V_a$, V_a是属性a的值域; $f:U\times A\to V$为信息函数, $\forall a\in A$, $x\in U$, 有$f(x,a)\in V_a$. 不失一般性, 也为了便于操作, 假设D仅有一个决策属性值, 其取值范围是$1\sim n$的整数.

关于粗糙集其他一些概念可参见文献[1].

本文研究建立在一致决策表的基础上, 而对于不一致决策表, 可以先按照文献[12]的方法将决策表一致化后再进行.

性质 1 决策表$S=(U,C\bigcup D,V,f)$中, 设$P\subseteq C$, $\mathrm{POS}_P(D)=\{x_i|\forall x_j\in[x_i]_P(i\neq j)$, 有$f(x_i,P)=f(x_j,P)$且$f(x_i,D)=f(x_j,D)\}$.

定义 1 决策表$S=(U,C\bigcup D,V,f)$, 其可分辨矩阵定义为$\boldsymbol{M}=\{m_{ij}\}$, m_{ij}表示矩阵中第i行第j列的元素, 即

$$m_{ij}=\begin{cases}\{a|a\in C\},\ f(x_i,a)\neq f(x_j,a)\bigwedge\\ \qquad\qquad f(x_i,D)\neq f(x_j,D);\\ \varnothing,\ \text{otherwise.}\end{cases}$$

定义 2 决策表$S=(U,C\bigcup D,V,f)$, 其二进制可分辨矩阵定义为$\mathrm{BM}=\{m((i,j),a_k)\}$, 其中$m((i,j),a_k)$表示为

$$m((i,j),a_k)=\begin{cases}1,\ f(x_i,a_k)\neq f(x_j,a_k)\bigwedge\\ \qquad f(x_i,D)\neq f(x_j,D),\ a_k\in C;\\ 0,\ \text{otherwise.}\end{cases}$$

例 1 表 1 为一个决策表$S=(U,C\bigcup D,V,f)$. 其中: 条件属性$C=\{a,b,c,d\}$, D为决策属性.

表 1 决策表 S

U	a	b	c	d	D
x_1	1	1	0	1	1
x_2	1	0	1	0	3
x_3	0	1	0	0	2
x_4	1	1	1	0	1
x_5	0	1	1	1	2
x_6	0	0	0	1	3
x_7	0	0	0	1	3

理论上整个可分辨矩阵的空间开销为$O(|C|\times|U|^2)$, 二进制可分辨矩阵的空间开销为$O(|C||U|^2/2)$. 通过对例 1 分析可以发现, 如果存储整个可分辨矩阵, 则需要$7\times7=49$个空间 (见表 2); 若存储可分辨矩阵上三角部分, 则需要 21 个空间; 而二进制可分辨矩阵仅需 16 个空间 (见表 3).

表 2 可分辨矩阵 $\boldsymbol{M}$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_1	$\varnothing$	bcd	ad	$\varnothing$	ac	ab	ab
x_2		$\varnothing$	abc	b	abd	$\varnothing$	$\varnothing$
x_3			$\varnothing$	ac	$\varnothing$	bd	bd
x_4				$\varnothing$	ad	$abcd$	$abcd$
x_5					$\varnothing$	bc	bc
x_6						$\varnothing$	$\varnothing$
x_7							$\varnothing$

表 3 二进制可分辨矩阵 BM

(i,j)	$abcd$	(i,j)	$abcd$
(1, 2)	0111	(3, 4)	1010
(1, 3)	1001	(3, 6)	0101
(1, 5)	1010	(3, 7)	0101
(1, 6)	1100	(4, 5)	1001
(1, 7)	1100	(4, 6)	1111
(2, 3)	1110	(4, 7)	1111
(2, 4)	0100	(5, 6)	0110
(2, 5)	1101	(5, 7)	0110

当数据集中存在较多的重复对象和不一致对象时, 二进制可分辨矩阵的空间优势相对于一般的可分辨矩阵将更加明显. 但是在处理多条件属性的大数据集时, 二进制可分辨矩阵往往会因空间开销过大而引起内存溢出错误.

2 两种二进制可分辨矩阵属性约简的等价性

定义 3　设 Red 为决策表 S 基于正区域的约简集, 决策表对于 $\forall R \subseteq C$, $R \in$ Red 满足以下两个条件[1]:

1) $\mathrm{POS}_C(D) = \mathrm{POS}_R(D)$;

2) $\forall a \in R$, 有 $\mathrm{POS}_R(D) \neq \mathrm{POS}_{R-\{a\}}(D)$.

定义 4　决策表 S 的二进制可分辨矩阵为 BM, 令 $\exists R \subseteq C$, 若 $\exists a_k \in R$, 有 $m((i,j),a_k) = 1$, 则称 $m((i,j),R) \neq 0$; 若 $\forall a_k \in R$, 有 $m((i,j),a_k) = 0$, 则称 $m((i,j),R) = 0$.

定义 5　设 RedBM 为决策表基于二进制可分辨矩阵 BM 的约简集, 对于 $\forall R \subseteq C$, $R \in$ RedBM 满足以下两个条件:

1) $\forall 0 \neq m((i,j),C) \in$ BM, 有 $m((i,j),R) \neq 0$;

2) $\forall a \in R, m((i,j),R) \neq 0$, 有 $m((i,j),R-\{a\}) = 0$.

定理 1　RedBM 是决策表 S 基于二进制可分辨矩阵的约简集, Red 为 S 基于正区域的约简集, 对于 $R \subseteq C$, $R \in$ RedBM, 有 $R \in$ Red.

证明　首先证明 $\forall R \in$ RedBM, 有 $\mathrm{POS}_C(D) = \mathrm{POS}_R(D)$. 采用反证法证明. 假设 $\exists R \in$ RedBM, 有 $\mathrm{POS}_C(D) \neq \mathrm{POS}_R(D)$. 由性质 1, $\exists x_i \in U$, 有 $x_i \in \mathrm{POS}_C(D)$ 但 $x_i \notin \mathrm{POS}_R(D)$, 则 $\exists x_j \in U (i \neq j)$, 有 $f(x_i,C) \neq f(x_j,C) \bigwedge f(x_i,D) \neq f(x_j,D)$ 和 $f(x_i,R) = f(x_j,R) \bigwedge f(x_i,D) \neq f(x_j,D)$. 由定义 4 知 $m((i,j),C) \neq 0$ 但 $m((i,j),R) = 0$, 此与定义 5 的条件 1) 矛盾, 故 $\mathrm{POS}_C(D) = \mathrm{POS}_R(D)$.

然后证明 $\forall R \in$ RedBM, $\forall a \in R$, 有 $\mathrm{POS}_C(D) \neq \mathrm{POS}_{R-\{a\}}(D)$. 采用反证法证明. 假设 $\exists R \in$ RedBM, $\exists a \in R$, 有 $\mathrm{POS}_R(D) = \mathrm{POS}_{R-\{a\}}(D)$. 由于 $R \in$ RedBM, 根据定义 5 的条件 1), $\forall 0 \neq m((i,j),C) \in$ BM, 有 $m((i,j),R) \neq 0$, 即 $f(x_i,R) \neq f(x_j,R) \bigwedge f(x_i,D) \neq f(x_j,D)$. 由于 x_i, x_j 的任意性, 有 $x_i \in \mathrm{POS}_R(D)$. 因为 $\mathrm{POS}_R(D) = \mathrm{POS}_{R-\{a\}}(D)$, 所以 $x_i \in \mathrm{POS}_{R-\{a\}}(D)$. 由性质 1, 有 $f(x_i,R-\{a\}) \neq f(x_j,R-\{a\}) \bigwedge f(x_i,D) \neq f(x_j,D)$. 由定义 4, 进而有 $m((i,j),R-\{a\}) \neq 0$, 这与定义 5 的条件 2) 矛盾, 故 $\mathrm{POS}_R(D) \neq \mathrm{POS}_{R-\{a\}}(D)$. □

定理 2　Red 为决策表 S 基于正区域的约简集, RedBM 为 S 基于二进制可分辨矩阵的约简集, 对于 $R \in C$, $R \in$ Red, 有 $R \in$ RedBM.

证明只需分 2 步: 1) $\forall R \in$ Red, 有 $\forall 0 \neq m((i,j),C) \in$ BM, 有 $m((i,j),R) \neq 0$; 2) $\forall R \in$ Red, $\forall a \in R$, $\exists m((i,j),R) \in$ BM, 有 $m((i,j),R-\{a\}) = 0$. 类似于定理 1, 采用反证法即可得证.

定理 3　基于正区域的属性约简定义与基于二进制可分辨矩阵 BM 的属性约简定义是等价的.

由定理 1 和定理 2 即可得证.

定义 6　设决策表 S 的二进制可分辨矩阵为 BM, 改进的二进制可分辨矩阵为 MB $= \{m_{ij}^a\}$, 其中 $m_{ij}^a = m((i,j),a)(a \in C)$; 对于 $R \subseteq C$, 有

$$m_{ij}^a = \sum_{k=1}^{|R|} m((i,j),a_k).$$

定义 7　在决策表 S 中, 设 $a \in C$, 属性 a 在改进二进制可分辨矩阵中的频率记为 $f(a)$, $f(a)$ 满足

$$f(a) = \sum m_{ij}^a,\ 1 \leqslant i,j \leqslant |U|.$$

定义 8　设 RedMB 为决策表 S 基于改进二进制可分辨矩阵 MB 的约简集, 对于 $\forall R \subseteq C$, $R \in$ RedMB 满足以下两个条件:

1) $\forall 0 \neq m_{ij}^C \in$ MB, 有 $m_{ij}^R \neq 0$;

2) $\forall a \in R, \exists m_{ij}^R \neq 0$, 有 $m_{ij}^{R-\{a\}} = 0$.

定理 4　定义 5 的约简定义与定义 8 的约简定义是等价的.

定理 5　基于正区域的属性约简定义与改进的二进制可分辨矩阵 MB 的属性约简定义是等价的.

3 基于垂直划分二进制可分辨矩阵的属性约简

性质 2　对决策表 $S = (U, C \bigcup D, V, f)$ 按决策属性值进行划分, 可分成 $|U/D|$ 个部分 (其中 $|U/D|$ 表示 U/D 中等价类的数目), 即 $\{Y_1, Y_2, \cdots, Y_{|U/D|}\}$ 共 $|U/D|$ 个类, 则可分辨矩阵上三角部分非空元素个数为

$$N = \sum_{j=1}^{|U/D|-1} |Y_j| \cdot \Big(\sum_{i=j+1}^{|U/D|} |Y_i| \Big) < |U|^2/2.$$

显然, N 也是二进制可分辨矩阵中元素的个数.

例如, 例 1 的决策表 S 按决策值划分后, 获得决策表 S' (见表 4), 其对应的可分辨矩阵为 M' (见表 5). 其中: $Y_1 = \{x_1, x_4\}, Y_2 = \{x_3, x_7\}, Y_3 = \{x_2, x_6, x_7\}$. 于是 M' 上三角部分非空元素个数和对应的二进制可分辨矩阵中元素个数均为 $2 \times (2+3) + 2 \times (3) = 16$ 个.

表 4　决策表 S'

Y_i	U	a	b	c	d	D
Y_1	x_1	1	1	0	1	1
	x_4	1	1	1	1	1
Y_2	x_3	0	1	0	0	2
	x_5	0	1	1	1	2
Y_3	x_2	1	0	1	0	3
	x_6	0	0	0	1	3
	x_7	0	0	0	1	3

表 5 可分辨矩阵 M'

Y_i	U	x_1	x_2	x_3	x_4	x_5	x_6	x_7
Y_1	x_1	$\varnothing$	$\varnothing$	ad	ac	bcd	ab	ab
	x_4		$\varnothing$	ac	ad	b	$abcd$	$abbcd$
Y_2	x_3			$\varnothing$	$\varnothing$	abc	bd	bd
	x_5				$\varnothing$	abd	bc	bc
Y_3	x_2					$\varnothing$	$\varnothing$	$\varnothing$
	x_6						$\varnothing$	$\varnothing$
	x_7							$\varnothing$

将决策表 S 按 U/D 划分成不同的决策类 $Y_i(1 \leqslant i \leqslant |U/D|)$ 后, 二进制可分辨矩阵的计算只需在非同类之间来完成, 并且所需空间大小在计算前便可预先知道(即需占用 N 个空间).

基于上述分析, 可以将决策表 S 按 U/D 分类后, 创建二进制可分辨矩阵; 然后对二进制可分辨矩阵的每个属性列进行垂直划分, 分解成 $|C|$ 个二进制属性列, 分别采用一维数组来存储, 则每个数组中的元素个数为 N, 即 $C_i[1,2,\cdots,N](1 \leqslant i \leqslant |C|)$. 定义一个累加和数组 $\mathrm{Sum}[1,2,\cdots,N]$, 且 $\mathrm{Sum}[k]$ 应满足

$$\mathrm{Sum}[k] = \sum_{i=1}^{|C|} C_i[k],\ 1 \leqslant k \leqslant N.$$

对于表 4 的决策表 S', 采用上述数据结构构建二进制可分辨矩阵, 其描述如表 6 所示.

表 6 二进制属性列 C_i 和数组 Sum

N	C_1 a	C_2 b	C_3 c	C_4 d	Sum	N	C_1 a	C_2 b	C_3 c	C_4 d	Sum
1	1	0	0	1	2	9	1	1	1	1	4
2	1	0	1	0	2	10	1	1	1	1	4
3	0	1	1	1	3	11	1	1	1	0	3
4	1	1	0	0	2	12	0	1	0	1	2
5	1	1	0	0	2	13	0	1	0	1	2
6	1	0	1	0	2	14	1	1	0	1	3
7	1	0	0	1	2	15	0	1	1	0	2
8	0	1	0	0	1	16	0	1	1	0	2

3.1 垂直划分二进制可分辨矩阵的属性约简算法

根据上述研究, 下面给出基于垂直划分二进制可分辨矩阵的启发式属性约简算法. 算法思路是: 初始 R 为 $\varnothing$, 每次从 $C-R$ 中选取一个具有最小初始频率的属性 C_i, 执行 $C \leftarrow C-\{C_i\}$, 并将 $C_i[1\ \cdots\ N]$ 从 $\mathrm{Sum}[1\ \cdots\ N]$ 中减去. 如果 Sum 中出现新的 0 值元素, 则对 $\forall k$ 满足 $C_i[k]=1(1 \leqslant k \leqslant N)$ 的 $\mathrm{Sum}[k]$ 清 0, 并将属性 a 加入约简 R 中; 如此反复, 直到 C 为 $\varnothing$, R 即为属性约简, 通常为最小属性约简. 算法可描述如下.

算法 1 基于垂直划分二进制可分辨矩阵的启发式属性约简算法.

输入: 决策表 $S=(U, C\bigcup D, V, f)$, C 为条件属性集, D 为决策属性集;

输出: 属性约简 R.

Step 1: $R=\varnothing$, 按决策属性分类 S, 得 $\{Y_1, Y_2, \cdots, Y_{|U/D|}\}$, 并计算得到 N;

Step 2: for $i=1$ to $|C|$ Do

Step 2.1: {创建 $C_i[1\ \cdots\ N]$, 统计 $f(a_i)$;

Step 2.2: 执行 $\mathrm{Sum}[1\ \cdots\ N] \leftarrow \mathrm{Sum}[1\ \cdots\ N] + C_i[1\ \cdots\ N]$; }

Step 3: while $(C \neq \varnothing)$

Step 3.1: {从 $C-R$ 中选择具有最小初始 $f(a_i)$ 值的属性, 设为 a;

Step 3.2: if $(\mathrm{Sum}[1\ \cdots\ N] \leftarrow \mathrm{Sum}[1\ \cdots\ N] - a[1\ \cdots\ N]$ 中产生新的 0 值元素) then { $R \leftarrow R\bigcup\{a\}$; 对于 $\forall k$ 有 $a[k]=1(1 \leqslant k \leqslant N)$, 执行 $\mathrm{Sum}[k] \leftarrow 0$;}

Step 3.3: $C \leftarrow C-\{a\}$; }

Step 4: Output R.

在算法 1 中, 凡涉及等价类划分算法均采用基数排序法来实现, 则 Step 1 的时间开销为 $O(|U|)$, Step 2 的时间开销为 $2O(|C|N)$. 因 Step 3.2 的时间开销为 $2O(N)$, Step 3 的循环次数为 $|C|$, 故 Step 3 的时间开销为 $2O(|C|N)$. 因此, 算法1总的时间开销为 $O(|U|)+2O(|C|N)+2O(|C|N)=O(|U|)+4O(|C|N)$, 时间复杂度为 $O(|C|N)<O(|C||U|^2)$. 算法 1 的空间开销主要有两个方面: $|C|$ 个二进制条件属性数组 C_i 和一个属性累加和数组 Sum, 故空间开销为 $O((|C|+1)N)$, 其空间复杂度为 $O(|C|N)$, 最大上界为 $O(|C||U|^2)$.

3.2 改进的属性约简算法

对算法 1 分析可以发现, 所有的 C_i 数组并不是同时参加运算, 某一时刻只有一个 C_i 数组参与运算. 因此, 可以将所有的 C_i 数组存放到外存中, 当需要哪个 C_i 数组参与运算时再将其调入内存, 这样可以始终保持内存中仅有一个 C_i 数组和一个 Sum 数组, 从而大大节省了算法空间开销.

为此, 对算法 1 进行改进, 具体算法描述如下.

算法 2 改进的属性约简算法.

输入: 决策表 $S=(U, C\bigcup D, V, f)$, C 为条件属性集, D 为决策属性集;

输出: 属性约简 R.

Step1: $R=\varnothing$, 按决策属性分类 S, 得 $\{Y_1, Y_2, \cdots, Y_{|U/D|}\}$, 并计算得到 N;

Step 2: for $i=1$ to $|C|$ Do

Step 2.1: {创建 $C_i[1\ \cdots\ N]$, 统计 $f(a_i)$;

Step 2.2: 执行 $\mathrm{Sum}[1\ \cdots\ N] \leftarrow \mathrm{Sum}[1\ \cdots N] + C_i[1\ \cdots\ N]$;

Step 2.3: 将 $C_i[1\ \cdots\ N]$ 存于外存中; }

Step 3: while $(C \neq \varnothing)$

Step 3.1: {从 $C-R$ 中选择具有最小初始 $f(a_i)$ 值

的属性, 设为 a;

Step 3.2: 将 $a[1\ \cdots\ N]$ 从外存调入内存;

Step 3.3: if ($\mathrm{Sum}[1\ \cdots\ N] \leftarrow \mathrm{Sum} - a[1\ \cdots\ N]$ 中产生新的 0 值元素) then $\{R \leftarrow R \bigcup a$; 对于 $\forall k$ 有 $a[k] = 1 (1 \leqslant k \leqslant N)$, 执行 $\mathrm{Sum}[k] \leftarrow 0$; $\}$

Step 3.4: $C \leftarrow C - \{a\}$; $\}$

Step 4: Output R.

如果忽略内外存交互的时间开销, 算法 2 与算法 1 具有相同的时间开销和时间复杂度. 算法 2 的空间开销仅为一个 C_i 数组和一个 Sum 数组所占用的空间, 故空间开销为 $O(N) + O(N) = 2O(N)$, 空间复杂度为 $O(N)$, 最大上限为 $O(|U|^2)$.

3.3　算法复杂度比较

3.3.1　时间复杂度比较

算法 1 和算法 2 的时间复杂度一样, 均为 $O(|C| \times |U|^2)$, 小于文献 [8] 约简算法的时间复杂度 $O(|C|^2 \times |U|^4)$, 也小于文献 [9] 约简算法的时间复杂度 $\max\{O(|C|^2(|U'_{\mathrm{pos}}||U/C|)), O(|C||U|)\}$.

3.3.2　空间复杂度比较

空间复杂度方面, 算法 1 和文献 [8] 均为 $O(|C| \times |U|^2)$, 文献 [9] 基于简化二进制可分辨矩阵约简算法的空间复杂度为 $\max\{O(|C|(|U'_{\mathrm{pos}}||U/C|)), O(|C| \times |U|)\}$, 而算法 2 仅为 $O(|U|^2)$. 可见算法 2 的空间复杂度低于算法 1、文献 [8] 和文献 [9], 并且算法 2 占用的空间不受条件属性个数的影响, 因此, 在处理条件属性 C 较多的大数据集时, 算法 2 的空间效率优势尤其明显.

3.4　实例分析

针对例 1, 采用算法1, 有 $f(a) = 10, f(b) = 12$, $f(c) = f(d) = 8$. 其中: $f(c) = 8$ 最小, 执行 $\mathrm{Sum}[1 \cdots N] \leftarrow \mathrm{Sum}[1 \cdots N] - c[1 \cdots N]$, 无新的 0 值元素产生; 随后 $f(d) = 8$ 最小, 执行 $\mathrm{Sum}[1 \cdots N] \leftarrow \mathrm{Sum}[1 \cdots N] - d[1 \cdots N]$, 也无新的 0 值元素产生; 再后 $f(a) = 10$ 最小, 执行 $\mathrm{Sum}[1 \cdots N] \leftarrow \mathrm{Sum}[1 \cdots N] - a[1 \cdots N]$, 有新的 0 值元素产生, $\forall k$ 将有 $a[k] = 1 (1 \leqslant k \leqslant N)$ 的 $\mathrm{Sum}[k] \leftarrow 0$, 并将 a 加入约简 R 中, 有 $R = \{a\}$; 此时只有 $f(b) = 12$, 执行 $\mathrm{Sum}[1 \cdots N] \leftarrow \mathrm{Sum}[1 \cdots N] - b[1 \cdots N]$, 也有新的 0 值元素产生, $\forall k$ 将有 $a[k] = 1 (1 \leqslant k \leqslant N)$ 的 $\mathrm{Sum}[k] \leftarrow 0$, 并将 b 加入约简 R 中, 则 $R = \{a, b\}$; 至此 $C = \varnothing$, 约简结束. 获得属性约简 $R = \{a, b\}$ 且为最小约简.

4　实验比较

为了比较一般二进制可分辨矩阵约简算法以及本文算法 1 和算法 2 的性能, 在 P4 2.8 GHz, RAM 2 G, VS.NET 2005 平台的 VC++ 环境下编程, 采用 UCI 数据库中的数据集为测试数据, 设计两组实验方案 (算法 A 表示文献 [8] 的属性约简方法; Time 表示算法执行时间, 单位为 s; Space 表示算法的空间开销, 单位为兆; RL 表示最小约简的属性数, RL_A、RL_1 和 RL_2 分别表示算法A、算法 1 和算法 2 获得约简的属性数; S_2/S_1 表示算法 2 空间开销与算法 1 空间开销的比值).

1) 选取 UCI 数据库中 7 个数据集为测数据. 采用算法 A、本文算法 1 和算法 2 进行实验, 结果见表 7 (算法 2 忽略了内外存数据传输的时间开销).

表 7　3 个算法的性能比较

数据集	样本数量	属性个数	RL	算法 A			算法 1			算法 2			S_2/S_1/%
				Time	Space	RL_A	Time	Space	RL_1	Time	Space	RL_2	
Balance	625	4	4	0.17	0.94	4	0.019	0.94	4	0.019	0.58	4	61.7
Tic-Tac-Toe	958	9	8	0.92	2.36	8	0.089	2.36	8	0.089	0.72	8	30.5
Car	1 728	6	6	1.16	3.57	6	0.093	7.35	6	0.093	3.69	6	50.2
Chess	3 196	36	29	10.28	100.48	29	3.79	100.48	29	3.79	12.48	29	12.4
Mushroom	8 124	22	4	245.21	420.21	5	26.24	420.21	4	26.24	81.53	4	19.4
Nursery	12 960	8	8	196.29	673.73	8	21.27	673.73	8	21.27	281.10	8	41.7
Poker	25 010	10	7	Memory Overflow			Memory Overflow			139.2	870.71	7	–

分析表 7, 可以发现:

① 由于算法 2 忽略了内外存数据传输的开销, 算法 1 和算法 2 的时间开销相同, 且低于算法 A; 随着数据集的增大, 算法 1 和算法 2 的时间优势逐渐明显, 这是由于算法 1 和算法 2 的时间复杂度均为 $O(|C| \times |U|^2)$, 低于算法 A 的时间复杂度 $O(|C|^2 + |U|^4)$. 另外, 算法 1 和算法 2 可以获得最小属性约简, 而算法 A 并不能保证获得最小属性约简. 例如, 对数据集 Mushroom 进行约简, 算法 A 获得的并不是最小属性约简.

② 算法 A 和算法 1 具有相同的空间开销, 算法 2 的空间开销明显低于算法 A 和算法 1. 在处理 Poker 数据集时, 由于算法 A 和算法 1 占用空间过大, 导致内存溢出; 而算法 2 却可以很好地处理. 这是由于算法 2 并没有将整个二进制可分辨矩阵存储在内存中, 而是将二进制可分辨矩阵垂直分解后存放到外部介质中, 仅将参与运算的属性列调入内存, 从而大大节省了空间开销.

③ 算法2处理数据集Chess和Mushroom的空间开销仅为算法1的12.4%和19.4%, 而其余的数据集为算法1的30% ～62 %. 造成这个现象的原因是算法1需存储$|C|$个C_i和一个Sum, 而算法2仅需存储一个C_i和一个Sum, 故条件属性越多, 算法2占用的空间相对算法1便越少. 如果将数据集按条件属性C的个数递增排序, 则S_2/S_1将大致呈递减的顺序.

由上述分析可见, 算法1和算法2在时间方面优于算法A; 在空间开销方面算法2优于算法A和算法1, 尤其在处理条件属性较多的大数据集时, 算法2的空间效率优势突出.

2) 为了测试算法1和算法2空间开销随条件属性数$|C|$的增长变化情况, 对UCI的Chess数据集(3 196条记录, 36个条件属性)分别取其前5, 10, 15, 20, 25, 30和36个条件属性进行测试, 结果如表8和图1所示(其中: N为二进制可分辨矩阵中元素个数; Space 1和Space 2分别表示算法1和算法2的空间开销, 单位为M).

表8 算法1和算法2的内存开销与$|C|$和N的关系

$\|C\|$	5	10	15	20	25	30	36
N	72	3 882	85 080	253 139	618 728	1 035 951	2 548 563
Space 1	0.02	0.08	1.66	6.08	17.85	34.76	100.40
Space 2	0.02	0.05	0.48	1.35	3.23	5.36	12.48

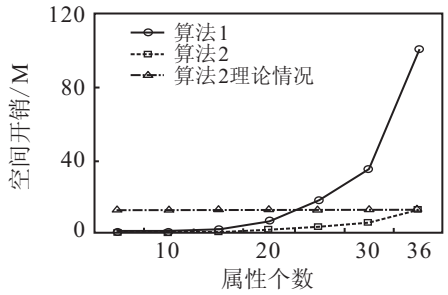

图1 两个算法内存开销与属性个数的关系

由图1可见, 随着条件属性的增加, 算法1的空间开销迅速增加, 而算法2的空间开销增长缓慢. 例如, 当属性个数由30个增加到36个时, 算法1内存开销由34.7 M增至100.4 M, 增加了65 M多; 算法2内存开销由5.36 M增至12.48 M, 仅增加了约6 M.

由前面的理论分析可知: 由于算法2仅需存储一个C_i和一个Sum, 其空间复杂度为$O(|U|^2)$. 从空间复杂度上看, 算法2的空间开销应不受条件属性个数的影响. 然而图1结果表明: 算法2随条件属性个数的增加, 占用的空间也在增加. 这是由于算法2实际的空间开销为$2O(N)$, 随着条件属性的增加, 决策表中不一致对象和相同对象个数减少, 导致二进制可分辨矩阵中元素数目N增加. 例如, 在有30个条件属性时, 二进制可分辨矩阵中元素的个数为1 035 951; 当有36个条件属性时, 二进制可分辨矩阵元素的个数增加到2 548 563, 因而C_i和Sum数组中元素的个数也由1 035 951增加到2 548 563. 元素个数增加了, 必然导致算法2的空间开销增大. 如果二进制可分辨矩阵元素的个数不随条件属性个数变化而变化(假设一直保持在36个条件属性时的元素个数), 则算法2的空间开销始终是12.48 M(即图1中算法2理论情况).

5 结 论

基于可分辨矩阵的属性约简算法, 需要将整个可分辨矩阵存储于内存中, 常因数据集较大, 可分辨矩阵需消耗很多空间, 从而造成内存溢出. 采用二进制可分辨矩阵虽然可以减少空间开销, 也常常因大数据集中含有条件属性过多而造成空间使用紧张. 为此, 本文提出对二进制可分辨矩阵按条件属性进行垂直划分, 并将所分解的二进制属性列保存到外部介质中, 在约简过程, 仅将当前需要的二进制属性列调入内存进行运算, 使内存中始终保持一个二进制属性列数组C_i和一个累加和数组Sum. 然后, 设计反向启发式属性约简算法(算法2), 算法的时间和空间复杂度上界分别为$O(|C||U|^2)$和$O(|U|^2)$. 理论分析和实验结果均表明该算法具有明显的时间和空间优势, 适用于多条件属性的大数据集.

随着网络技术的发展, 分布式技术应用越来越广泛, 本文的属性约简方法如果能够充分利用分布式处理技术, 将会大大提高其处理速度. 因此, 针对不一致信息系统, 将分布式技术应用于属性约简是下一步的研究工作.

参考文献(References)

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

[2] 王国胤, 于洪, 杨大春. 基于条件信息熵的决策表约简[J]. 计算机学报, 2002, 25(7): 759-766.
(Wang G Y, Yu H, Yang D C. Decision table reduction based on conditional information entropy[J]. Chinese J of Computers, 2002, 25(7): 759-766.)

[3] 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.
(Liu S H, Sheng Q J, Wu B, et al. Research on efficient algorithms for rough set methods[J]. Chinese J of Computers, 2003, 26(5): 524-529.)

[4] Skowron A, Rauszer C. The discernibility matrices and functions in information systems[C]. Intelligent Decision Support-handbook of Applications and Advances of the Rough Sets Theory. Dordrecht: Kluwer Academic Publisher, 1991: 331-362.

(下转第573页)