文章编号: 1001-0920 (2015) 08-1531-06 **DOI:** 10.13195/j.kzyjc.2014.0724

基于差别信息树的 rough set 属性约简算法

蒋 瑜

(成都信息工程大学 软件工程学院，成都 610225)

摘 要: 差别矩阵为粗糙集属性约简提供了很好的思路, 但差别矩阵中存在冗余的重复和父集元素. 为了消除这些冗余元素, 提出一棵有序树: 差别信息树, 该树能消除差别矩阵中的重复元素, 同时在大多数情况下也能完全消除父集元素, 实现对差别矩阵中非空元素的压缩存储. 为了验证差别信息树的有效性, 提出一种属性约简完备算法, 并使该算法的时间复杂度降为 $O(|C||U|^2)$.

关键词: 粗糙集；差别矩阵；属性约简；差别信息树

中图分类号: TP18 **文献标志码:** A

Attribute reduction with rough set based on discernibility information tree

JIANG Yu

(College of Software Engineering，Chengdu University of Information Technology，Chengdu 610225，China. E-mail: jiangyu@cuit.edu.cn)

Abstract: Attribute reduction plays an essential role in rough set. Discernibility matrix provides a good way for attribute reduction in rough set, but there are many redundancy and pointless nonempty elements in discernibility matrix, such as duplications and supersets. In order to eliminate these redundancies and pointless elements, a discernibility information tree is proposed, which is an extended order-tree, and can be easy to fully and partly eliminate duplicates of elements and supersets of elements in discernibility matrix efficiently. In order to efficiently utilize the dicernibility information tree structure for attribute reduction, an attribute reduction complete algorithm is developed, and it's complexity is reduced to $O(|C||U|^2)$.

Keywords: rough set；discernibility matrix；attribute reduction；discernibility information tree

0 引 言

波兰学者 Pawlak[1]于 1982 年提出了粗糙集理论, 其主要研究内容之一是属性约简. 到目前为止, 基于不同的约简思想, 人们提出了许多优秀的属性约简算法[2]. 其中, Skowron 等[3]提出的差别矩阵 (DM) 因其简洁直观而受到了广大研究者的关注[3-15]. 目前, 基于差别矩阵或差别函数, 人们已提出了许多优秀的属性约简启发式算法. 然而, 这些启发式算法都有一共同的缺点, 即算法中使用差别矩阵中全部非空元素来构建属性约简. 文献 [3] 指出, 差别矩阵中所有非空元素的合取 ($\bigwedge$) 运算构成属性约简集, 由合取 ($\bigwedge$) 运算可知, 差别矩阵中大量非空重复元素和父集元素在属性约简中是没有任何作用的, 但这些元素占据了大量的存储空间, 并且增加了求解属性约简的计算复杂度. 虽然文献 [10] 提出了一种新的差别矩阵存储方法 (C-Tree), 消除了差别矩阵中重复元素的出现, 实现了差别矩阵的压缩存储, 但对于冗余的父集元素, 该方法则显得无能为力.

为了能快速消除差别矩阵中大量重复和父集元素, 实现差别矩阵中非空元素压缩存储, 本文提出一种虚拟的树形结构: 差别信息树. 该树是一棵有序树, 能使相同的差别信息映射到同一路径上, 并且在大部分情况下也能使父集差别信息映射到其子集差别信息所对应的路径上, 从而实现消除差别矩阵中的父集和重复元素. 为了验证所提出差别信息树的有效性, 本文基于差别信息树提出一种属性约简完备算法 (根据 Pawlak 的定义, 若算法求得的约简 R 满足: 1) $\mathrm{POS}_R(D) = \mathrm{POS}_C(D)$ 和 2) 对于任意 $a \in R$, $\mathrm{POS}_{R-\{a\}}(D) \neq \mathrm{POS}_C(D)$, 则称该算法为属性约简完备算法[15]; 若算法求得的 R 只满足条件 1), 则称该

收稿日期: 2014-05-10；**修回日期:** 2014-07-22.

作者简介: 蒋瑜(1980−), 男, 副教授, 硕士, 从事粗糙集理论和数据挖掘等研究.

算法为属性约简不完备算法). 该算法在每次迭代过程中, 从右至左选取差别信息树中根节点的子节点, 同时删除包含该节点的所有路径, 从而求得决策表的一个完备约简, 并使该约简算法的时间复杂度为 $O(|C||U|^2)$.

1 基本概念

本节基于文献[1, 3]给出粗糙集和差别矩阵的相关知识介绍.

定义 1 一个决策表(或信息表或信息系统)可定义为 $S=(U,A,V,f)$. 其中: U 为对象的集合; 属性集 A 由条件属性集 C 和决策属性集 D 构成, 且 $C\bigcap D=\varnothing$; V 为属性的值域, $V=\{v_{a1},v_{a2},\cdots,v_{am},v_d\}$; f 为信息函数, $f:U\times A\to V$, $\forall a\in A,x\in U$, 有 $f(x,a)\in v_a$. 表1为一决策表. 其中: $U=\{u_1,\cdots,u_6\}, C=\{a,b,c,d,e\}, D=\{f\}$.

表 1 决策表

U	C					D
	a	b	c	d	e	f
u_1	ma	sm	ma	ma	ma	sm
u_2	ne	ma	sm	sm	sm	lo
u_3	sm	ma	sm	sm	sm	sm
u_4	sm	ma	sm	ne	sm	lo
u_5	sm	ne	ne	sm	sm	sm
u_6	ma	ne	ne	ma	sm	sm

定义 2 对于决策表 $S=(U,A,V,f)$, 令 $R\subseteq A$, 则该决策表的不可区分关系定义为: $\mathrm{ind}(R)=\{(x_i,x_j)|f(x_i,b)=f(x_j,b),\forall b\in R\bigwedge x_i,x_j\in U\}$. 显然, 不可区分关系是一个等价类, 含 x 的等价类记为 $[x]_{\mathrm{ind}(R)}$ 或 $[x]_R$. R 在 U 上导出的划分记为 $U/\mathrm{ind}(R)$ 或 U/R.

定义 3 决策表 $S=(U,A,V,f)$ 中, $P\subseteq C$, D 的 P 正区域记为 $\mathrm{POS}_P(D)$, 定义为

$$\mathrm{POS}_P(D)=\bigcup_{X\in U/D}P_{-}(X).$$

D 的 P 正区域是 U 中所有根据 $\mathrm{ind}(P)$ 的信息可以划分到 D 的等价关系中的对象集合.

定义 4 设 $|U|=n$, 则决策表 $S=(U,A,V,f)$ 的差别矩阵是一个包含 $n\times n$ 个元素的矩阵, 差别矩阵中每一元素称为差别信息(DI), 其具体定义如下:

$$a^*(x_i,x_j)=$$
$$\{a\in C|f(x_i,a)\neq f(x_j,a)\bigwedge w(x_i,y_j)=1\}.$$

对于 $x_i,y_j\in U, w(x_i,y_j)$ 满足

$$\begin{cases}1,\ x_i\in\mathrm{POS}_C(D)\bigwedge x_j\notin\mathrm{POS}_C(D);\\ 1,\ x_i\notin\mathrm{POS}_C(D)\bigwedge x_j\in\mathrm{POS}_C(D);\\ 1,\ x_i,x_j\in\mathrm{POS}_C(D)\bigwedge(x_i,x_j)\notin\mathrm{ind}(D);\\ 0,\ \text{otherwise}.\end{cases}\tag{1}$$

差别矩阵是一对角线为空的对称矩阵, 因此只考虑矩阵的上三角或下三角元素即可. 表2给出了表1所对应的差别矩阵.

表 2 表1所对应的差别矩阵

	u_1	u_2	u_3	u_4	u_5	u_6
u_1	$\varnothing$	$\{a,b,c,d,e\}$	$\varnothing$	$\{a,b,c,d,e\}$	$\varnothing$	$\varnothing$
u_2		$\varnothing$	$\{a\}$	$\varnothing$	$\{a,b,c\}$	$\{a,b,c,d\}$
u_3			$\varnothing$	$\{d\}$	$\varnothing$	$\varnothing$
u_4				$\varnothing$	$\{b,c,d\}$	$\{a,b,c,d\}$
u_5					$\varnothing$	$\varnothing$
u_6						$\varnothing$

定义 5 决策表的差别函数 f_A 定义为

$$f_A(a'_1,\cdots,a'_m)=\bigwedge\{\bigvee a^*(x_i,x_j)|a^*(x_i,x_j)\neq\varnothing\}.$$

差别函数的极小析取范式中的所有合取式是 C 的所有 D 约简, 简称约简.

例 1 表1所对应的差别函数为

$$(a\bigvee b\bigvee c\bigvee d\bigvee e)\bigwedge(a\bigvee b\bigvee c\bigvee d\bigvee e)\bigwedge$$
$$a\bigwedge(a\bigvee b\bigvee c)\bigwedge(a\bigvee b\bigvee c\bigvee d)\bigwedge d\bigwedge$$
$$(b\bigvee c\bigvee d)\bigwedge(a\bigvee b\bigvee c\bigvee d)=a\bigwedge d.$$

所以决策表1的约简为 $\{a,d\}$.

定义 6 在决策表 $S=(U,A,V,f)$ 中, 令 B ($B\subseteq C$) 为决策表所有约简的"交", 若 B 不为空, 则称为 C 相对 D 的核, 简称核, 记为 $\mathrm{Core}_D(C)$. 然而, 对于差别矩阵而言, 矩阵中所有只包含一条件属性的非空元素的"并"构成核. 所以根据表2, 可得 $\mathrm{Core}_D(C)$ 为 $\{a,d\}$.

2 C-Tree 相关知识

文献[10]中基于频繁模式树的思想提出了一种能压缩存储差别矩阵中非空元素的虚拟树结构(C-Tree), 该树是一颗有序树, 可使差别矩阵中多个非空元素映射到同一路径上或共享某一路径前缀, 相比于差别矩阵, 实现了差别信息的压缩存储. 图1给出了基于表1构建的一棵C-Tree.

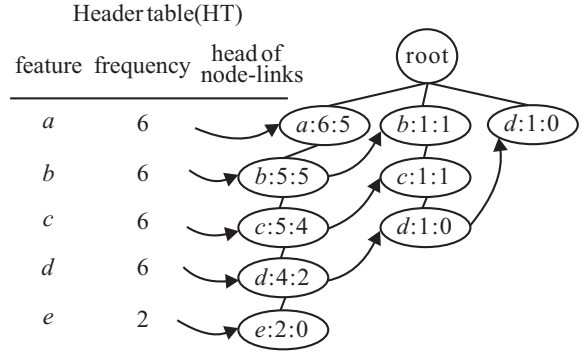

图 1 基于表1和式(1)构建的C-Tree

如图 1 所示, 虽然 C-Tree 实现了差别矩阵非空元素的压缩存储, 但是由 C-Tree 的构建过程可知, 属性集 $\{a,b,c,d,e\}$ 与其子集 $\{a,b,c,d\}$、$\{a,b,c\}$ 和 $\{a\}$ 均被映射到同一路径 $\langle a,b,c,d,e\rangle$ 上. 然而, 由差别函数可知: $a\bigwedge(a\bigvee b\bigvee c)\bigwedge(a\bigvee b\bigvee c\bigvee d)\bigwedge(a\bigvee b\bigvee c\bigvee d\bigvee e)=a$. 因此, 路径 $\langle a,b,c,d,e\rangle$ 中的节点 (b)、(c)、(d) 和 (e) 都是冗余节点, 它们的存在不仅增加了存储空间, 而且也增加了构建 C-Tree 的时间. 因此, 需要设计一种新的数据结构来存储差别矩阵中非空元素.

3　差别信息树的设计与实现

首先给出差别信息树的定义, 然后提出一种构建差别信息树的算法.

定义 7　差别信息树是一棵有序树, 其特点是每个节点至多只有 $|C|$ 棵子树 (C 为决策表条件属性集), 差别信息树的子树也是有序树, 其次序不能任意颠倒. 其具体定义如下.

1) 差别信息树中每个节点主要由 4 部分构成: 属性名 (attName)、父亲指针 (parentPointer)、孩子指针 (childrenPointer) 和同名指针 (sameNamePointer). 其中: 属性名标识了该节点所对应的条件属性; 父亲指针指向该节点的父亲节点; 孩子指针指向该节点的孩子节点; 同名指针指向其他路径中与该节点具有相同属性名的节点.

2) 属性指针头表, 该表由两部分构成: 属性名和同名指针. 其中: 属性名标识了该表项对应的条件属性; 同名指针指向差别信息树中与该表项具有相同属性名的最左边节点.

基于以上差别信息的定义, 下面给出差别信息的构建算法.

算法 1　差别信息构建算法.

输入: 决策表 T;

输出: 决策表所对应的差别信息树.

Step 1: 创建决策表的根节点.

Step 2: 构建属性指针头表, 并按决策表中条件属性从左到右的次序赋值给属性指针头表中对应的属性名.

Step 3: 根据定义 4 中的式 (1), 计算决策表中所有对象对的差别信息. 设 B (B 是条件属性的一子集) 为差别信息集中任意一差别信息, 且 B 是一有序集合, 其中元素的属性按条件属性在决策表中从左至右的次序排列.

Step 4: 调用 createPath (B, currentR) 函数, 把所有差别信息插入根为 currentR 的差别信息树中.

Step 5: 算法结束.

CreatePath (B, currentR) 函数实现了把差别信息 B (B 为条件属性一子集合) 插入根节点为 currentR (currentR 为差别信息树中一节点) 的差别信息子树中, 其具体功能如下所示.

1) 如果 B 为空, 则转 9);

2) 选择 B 中最左边一条件属性 a, 并令 $B \Leftarrow B-\{a\}$;

3) 如果 currentR 的所有子节点中, 不存在一子节点 N 的属性名为 a, 则转 7);

4) 如果 N 是一叶子节点, 则采用不扩展路径策略, 不构建差别信息 B 中剩余属性对应的节点, 转 1);

5) 如果条件属性 a 是 B 中最后一元素, 则采用删除子树策略, 从差别信息树中删除以节点 N 为根的子树, 并保留节点 N, 转 1);

6) 令 currentR 指向其子节点 N, 转 1);

7) 创建一新节点 N', 节点 N' 作为 currentR 一子节点, 同时初始 N' 的属性名为 a, 并通过该节点的同名指针连接到具有与该节点有相同属性名的节点上, 从而构成一个同名属性节点链;

8) 令 currentR 指向其子节点 N';

9) 函数结束.

例 2　基于算法 1 和表 1, 给出差别信息树的具体构建过程.

首先, 创建差别信息树的根节点, 对于决策表 1 中所有对象对, 计算其差别信息, 并且使差别信息中条件属性的次序按决策表中条件属性从右至左的次序排列.

然后, 构建第 1 条差别信息 $\{a,b,c,d,e\}$ 所对应的路径 $\langle a,b,c,d,e\rangle$, 并插入差别信息树中. 对于第 2 条差别信息 $\{a,b,c,d,e\}$, 由于该差别信息与差别信息树中路径 $\langle a,b,c,d,e\rangle$ 所对应的差别信息相同, 第 2 条差别信息 $\{a,b,c,d,e\}$ 也映射到路径 $\langle a,b,c,d,e\rangle$ 上. 对于第 3 条差别信息 $\{a\}$, 因为该差别信息完全包含于差别信息树中路径 $\langle a,b,c,d,e\rangle$ 所对应的差别信息, 采用删除子树策略, 从路径 $\langle a,b,c,d,e\rangle$ 中删除节点 (a) 之后的所有节点, 即原始路径 $\langle a,b,c,d,e\rangle$ 被修改为新的路径 $\langle a\rangle$. 对于第 4 条差别信息 $\{a,b,c\}$, 因该差别信息完全包含了差别信息树中路径 $\langle a\rangle$ 所对应的差别信息, 故采用不扩展路径策略, 不构建新节点 (b) 和 (c), 即差别信息 $\{a,b,c\}$ 映射到差别信息树中的路径 $\langle a\rangle$ 上. 同理, 重复上面构建过程, 直至把最后一条差别信息 $\{a,b,c,d\}$ 插入差别信息树中. 图 2 给出了基于算法 1 和表 1 所构建的差别信息树.

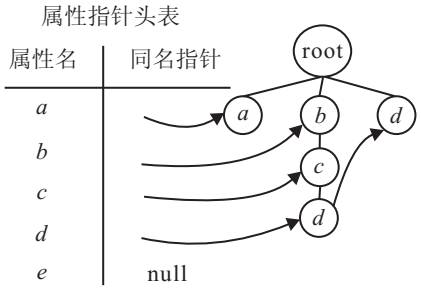

图 2 基于表 1 和式 (1) 构建的差别信息树

由以上差别信息树的构建过程可知, 在构建差别信息树过程中采用了不扩展路径策略和删除子树策略, 保证了属性集 $\{a,b,c\}$ 与其子集 $\{a,b,c\}$、$\{a,b\}$ 和 $\{a\}$ 映射到同一路径 $\langle a\rangle$ 上, 从而有效地减少了构建差别信息树的时空复杂度, 进一步实现了对差别信息的压缩存储.

3.1 差别信息树的特性

由差别信息树的构建过程可以得出如下差别信息树的特性.

定理 1 差别信息树包含了能够获得属性约简所需要的全部差别信息.

证明 假设 DS 是差别信息树中所有路径对应的差别信息的集合, 由差别信息树的构建过程可知, $\mathrm{DS}\subseteq\mathrm{DM}$ (DM 为式 (1) 所求得的差别矩阵), 并且对于任意一差别信息 DI, 如果 $\mathrm{DI}\in\mathrm{DM}-\mathrm{DS}$, 则在 DS 中一定存在一差别信息 DI′, 使得 $\mathrm{DI}'\subseteq\mathrm{DI}$ 成立. 由差别函数可知, DI 与 DI′ 进行合取运算, 其结果为 DI′. 所以差别信息树包含了能够获得属性约简所需要的全部差别信息. □

定理 2 差别信息树中所有只有一个节点的路径所对应差别信息的"并"构成了决策表的 $\mathrm{Core}_D(C)$.

证明 由差别信息树的构建过程可知, 假设差别信息树中某一节点的属性名标识的是条件属性 a, 并且差别信息树中存在某一路径只包含该节点, 则一定存在一差别信息 $\{a\}$ 与该路径对应. 在差别矩阵中, 若 $a\in C$ 且 $\{a\}$ 是差别矩阵中一非空元素, 则称 a 为 C 中相对 D 必要的. C 的所有必要属性的集合称为 C 相对 D 的核. □

根据定理 2, 结合图 2 可知, 决策表 1 的核为 $\{a,d\}$.

定理 3 设 R 是差别信息树中根节点所有子节点所代表的条件属性集合, 则 $\mathrm{POS}_R(D)=\mathrm{POS}_C(D)$ 成立.

证明 由定理 1 可知, 差别信息树包含了能够获得属性约简所需要的全部差别信息. 假设 DS 是差别信息树中所有路径对应的差别信息的集合, 则 $\forall\mathrm{DI}\in\mathrm{DS}$, 一定有 $R\bigcap\mathrm{DI}\neq\varnothing$, 从而进一步证明了 $\mathrm{POS}_R(D)=\mathrm{POS}_C(D)$ 成立. □

基于定理 3, 只要删除 R 中所有不必要的条件属性, 就可以求得决策表的一个完备约简. 由图 2 可知, R 为 $\{a,b,d\}$, 因为 $\mathrm{POS}_{R-\{b\}}(D)=\mathrm{POS}_C(D)$, b 是 R 中不必要的条件属性, 而 a 和 d 是 R 中必要的条件属性, 所以 $\{a,d\}$ 是决策表 1 的一个完备约简.

性质 1 若 $\mathrm{Core}_D(C)$ 是决策表的相对核, 假设从差别信息树中删除包含 $\mathrm{Core}_D(C)$ 中任意元素的所有路径, 设 R 是新差别信息树中根节点所有子节点所代表的条件属性集合, 则 R 中最右边的条件属性一定是一必要的条件属性.

3.2 差别信息树复杂度分析

对于一给定的决策表, 假设其拥有 $|U|$ 个对象和 $|C|$ 个条件属性, 则差别矩阵中最多具有 $|U|^2$ 个非空差别信息. 假设差别矩阵中实际非空差别信息为 N (一般情况下 N 远远小于 $|U|^2$), 由差别信息树的构建过程可知, 差别信息树中最多包含 N 条路径, 而每条路径中最多包含 $|C|$ 个节点, 因此, 差别信息树中最多包含 $|C|N$ 个节点. 又由于在差别信息树中许多路径共享前缀, 差别信息树中节点数都远远小于 $|C|N$. 综上所述, 在最坏情况下, 差别信息树的空间复杂度为 $O(|C||U|^2)$.

另外, 在差别信息树的构建过程中, 由于算法最多迭代 $|U|^2$ 次, 并且在每次迭代过程中最多比较并插入 $|C|$ 个节点, 删除 N_i 节点, 差别信息树的时间复杂度为 $|C||U|^2+(N_1+\cdots+N_{|U|^2})$. 因为差别信息树中最多包含 $|C||U|^2$ 个节点, 所以 $N_1+\cdots+N_{|U|^2}$ 的值最多为 $|C||U|^2$, 从而可得差别信息树的时间复杂度为 $O(|C||U|^2)$.

4 基于差别信息树的属性约简算法

为了验证本文所提出差别信息树的有效性, 本节基于差别信息树提出一个属性约简完备算法. 该算法基于定理 3 和性质 1, 在每次迭代过程中从右至左从差别信息树中选择必要属性, 同时删除包含必要属性的路径. 算法具体描述如下.

算法 2 采用自右向左的策略获得属性约简.

输入: 差别信息树;

输出: 决策表的一个完备约简.

Step 1: 创建一空集 R.

Step 2: 获取差别信息树中只有一个节点的路径, 假设 R' 是这些节点对应属性组合的集合. 如果 $R'\neq\varnothing$, 则从差别信息树中删除包含有 R' 中元素的路径,

并令 $R \Leftarrow R'$.

Step 3: 若差别信息树只包含根节点, 则转 5).

Step 4: 在差别信息树中, 选择根节点的最右边的一子节点, 假设该节点对应的属性为 a, 则令 $R \Leftarrow R \bigcup \{a\}$, 并从差别信息树中删除包含该节点的路径, 转 3).

Step 5: 输入 R, 算法结束.

例 3　基于图 2, 算法 2 的求解过程如下: 获得图 2 所示差别信息树中只含一个节点的路径 (a) 和 (d), 此时令 $R \Leftarrow \{a, d\}$. 同时从差别信息树中删除包含节点 (a) 或 (d) 的路径. 这时, 图 2 所对应的差别信息树中只包含根节点, 算法结束, 输出 R. 因而 $\{a, d\}$ 即为所求约简.

4.1　算法时间复杂度分析

基于 3.2 节的分析, 差别信息树中最多拥有 $|C| \times |U|^2$ 个节点, 由算法 2 的求解过程可知, 该算法最多迭代 $|C|$ 次. 假设在每次迭代过程中删除 N_i 节点, 则在 $|C|$ 次迭代过程中该算法删除的节点数最多为 $N_1 + \cdots + N_{|C|} = |C||U|^2$. 所以算法 2 的时间复杂度为 $O(|C||U|^2)$.

4.2　算法完备性证明

根据 Pawlak 的定义, 从差别矩阵的角度考虑, 若属性集 $R \subseteq C$ 是给定决策表的完备约简, 则 R 应满足: 1) 对于任意 $B \in \mathrm{DM}$, $B \bigcap R \neq \varnothing$; 2) 对于任意 $a \in R$, 存在 $B \in \mathrm{DM}$, 使得 $B \bigcap (R - \{a\}) = \varnothing$. 根据本文定理 1, 差别信息树包含了能够获得属性约简所需要的全部差别信息, 假设 DS 是差别信息树中所有路径对应的差别信息的集合, 则从差别信息树角度考虑, 这两个条件等价于: 1) 对于任意 $B \in \mathrm{DS}$, $B \bigcap R \neq \varnothing$; 2) 对于任意 $a \in R$, 存在 $B \in \mathrm{DS}$, 使得 $B \bigcap (R - \{a\}) = \varnothing$. 所以, 证明算法 2 是完备的只需证明算法输出的 R 满足以上两条件即可.

由算法 2 的求解过程可知, 对于任意 $B \in \mathrm{DS}$, $B \bigcap R \neq \varnothing$ 成立. 将算法的 Step 2 中获得的决策表的 $\mathrm{Core}_D(C)$ 作为属性约简的一部分, 并从差别信息树中删除包含核元素的所有路径. 假设 $R' \Leftarrow R - \mathrm{Core}_D(C)$. 如果 r 是 R' 中最右边一元素, 则在当前的差别信息树中, 根节点最右一子节点所对应的属性一定是 r, 并且以该节点为根的子树中一定不包含 $R' - \{r\}$ 中任意属性所对应的节点, 从而可知 $r \in R$, 存在 $B \in \mathrm{DS}$, 使得 $B \bigcap (R - \{r\}) = \varnothing$. 同理可证, R' 中其他元素也满足该条件. 综上可得: 1) 对于任意 $B \in \mathrm{DS}$, $B \bigcap R \neq \varnothing$; 2) 对于任意 $a \in R$, 存在 $B \in \mathrm{DS}$, 使得 $B \bigcap (R - \{a\}) = \varnothing$. 所以算法 2 求得的约简是一完备约简.

5　实验结果及分析

为了验证本文所提出的差别信息树的有效性, 本节选用 UCI 机器学习数据库中的数据在 Pentium dual-core 3.2 GHz (2 GB 内存, Microsoft Windows 7 操作系统) 上进行实验, 给出了 C-Tree 与差别信息树在存储差别信息时的时空复杂度对比结果, 如表 3 所示.

表 3　基于 C-Tree 和差别信息树的实验结果

数据库名称	条件属性数	对象数	树中节点数		树的构建时间 /s	
			C-Tree	差别信息树	C-Tree	差别信息树
Lenses	4	24	16	5	0.0	0.0
Letter	15	20 000	32 528	2 075	361.016	307.133
Voting	16	435	16 746	582	0.125	0.124
Tic-tac-toe	9	958	510	45	0.406	0.327
Poker hand	10	25 010	1 024	84	273.047	241.769

由表 3 数据可知: 1) 差别信息树中节点数远远小于 $|C||U|^2$; 2) 差别信息树的时空复杂度都小于 C-Tree 的时空复杂度. 并且从树中节点数来看, 基于 Voting 数据库, C-Tree 中节点数是差别信息树中节点数的 28 倍多; 基于其他数据库, 节点数都达到了十几倍; 最少的是基于 Lenses 数据库, 其节点数也达到了 3 倍多. 所以由表 3 可知在存储差别信息时, 差别信息树比 C-Tree 具有更小的时空复杂度.

由 4.2 节可知, 基于差别信息树, 算法 2 可以获得决策表一完备约简. 然而, 基于图 1 的 C-Tree, 算法 2 所获得的约简为 $\{d\}$, 而约简 $\{d\}$ 并不是决策表的一约简, 因为表 1 所对应决策表的 $\mathrm{Core}_D(C)$ 为 $\{a, d\}$. 从而可知, 基于 C-Tree, 已有的约简策略很难获得决策表的属性约简.

6　结　论

本文提出了一个能压缩存储差别信息的数据结构: 差别信息树. 与现有的差别信息压缩存储结构 C-Tree 相比, 差别信息树具有更小的时空复杂度. 然而, 本文的差别信息树的构建只是基于条件属性的原始顺序, 没有考虑属性重要度和核在差别信息树构建

中的作用. 下一步的工作是将属性重要度和核引入差别信息树的构建中, 探讨在不同条件属性序关系下差别信息树是否能进一步对差别信息实现压缩存储.

参考文献(References)

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

[2] Thangavel K, Pethalakshmi A. Dimensionality reduction based on rough set theory: A review[J]. Applied Soft Computing, 2009, 9(1): 1-12.

[3] 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, 1991: 331-362.

[4] Yao Y Y, Zhao Y. Discernibility matrix simplification for constructing attribute reducts[J]. Information Sciences, 2009, 179(5): 867-882.

[5] Hu Q H, Xie Z X, Yu D R. Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation[J]. Pattern Recognition, 2007, 40(12): 3509-3521.

[6] Yuhua Qian, Jiye Liang, Witold Pedrycz, et al. Positive approximation: An accelerator for attribute reduction in rough set theory[J]. Artificial Intelligence, 2010, 174(9/10): 597-618.

[7] Qian Y H, Liang J Y. Combination entropy and combination granulation in rough set theory[J]. Int J of Uncertainty, Fuzziness and Knowledge-Based Systems, 2008, 16(2): 179-193.

[8] 蒋瑜, 王燮, 叶振. 基于差别矩阵的Rough集属性约简算法[J]. 系统仿真学报, 2008, 20(14): 3717-3720.
(Jiang Y, Wang X, Ye Z. Attribute reduction algorithm of rough sets based on discernibility matrix[J]. J of System Simulation, 2008, 20(14): 3717-3720.)

[9] Chen D G, Zhao S Y, Zhang L, et al. Sample pair selection for attribute reduction with rough set[J]. IEEE Trans on Knowledge and Data Engineering, 2012, 24(11): 2080-2093.

[10] Yang M, Yang P. A novel condensing tree structure for rough set feature selection[J]. Neurocomputing, 2008, 71(4): 1092-1100.

[11] 杨明, 吕静. 一种基于C-Tree的属性约简增量式更新算法[J]. 控制与决策, 2012, 27(12): 1769-1775.
(Yang M, Lü J. An incremental updating algorithm for attribute reduction based on C-Tree[J]. Control and Decision, 2012, 27(12): 1769-1775.)

[12] Neil Mac Parthaláin, Qing Shen, Richard Jensen. A distance measure approach to exploring the rough set boundary region for attribute reduction[J]. IEEE Trans on Knowledge and Data Engineering, 2010, 22(3): 305-317.

[13] Jiang Yu. Minimal attribute reduction for rough set based on attribute enumeration tree[J]. Int J of Advancements in Computing Technology, 2012, 4(19): 391-399.

[14] 蒋瑜, 王鹏, 王燮, 等. 基于差别矩阵的属性约简完备算法[J]. 计算机工程与应用, 2007, 43(19): 185-187.
(Jiang Y, Wang P, Wang X, et al. Complete algorithm for attribute reduction based on discernibility matrix[J]. Computer Engineering and Applications, 2007, 43(19): 185-187.)

[15] 王宾, 陈善本. 一种基于差别矩阵的属性约简完备算法[J]. 上海交通大学学报, 2004, 38(1): 43-46.
(Wang B, Chen S B. A complete algorithm for attribute reduction based on discernibility matrix[J]. J of Shanghai Jiaotong University, 2004, 38(1): 43-46.)

(责任编辑：李君玲)