

属性序下的快速约简算法

胡 峰 王国胤

(重庆邮电大学计算机科学与技术研究所 重庆 400065)
(西南交通大学信息科学与技术学院 成都 610031)

摘 要 将分治法的思想溶入 Rough 集算法中,在给定属性序下,提出了基于分治策略的属性约简算法.利用该算法可以计算给定属性序下的唯一约简,并能快速得到海量数据的属性约简.在一次性将决策表的所有数据调入计算机内存的情况下,算法的平均时间复杂度为 $O(|U| \times |C| \times (|C| + \log|U|))$,空间复杂度为 $O(|U| + |C|)$.仿真实验结果说明了算法的高效性.

关键词 粗集;分治;属性约简;属性序
中图法分类号 TP18

Quick Reduction Algorithm Based on Attribute Order

HU Feng WANG Guo-Yin

(Institute of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065)
(School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031)

Abstract The idea of divide and conquer is adopted in attribute reduction of rough set theory. A quick algorithm for attribute reduction of ordered attributes is proposed based on the divide and conquer method. A unique attribute reduction can be obtained with this algorithm. It is suitable for dealing with huge data reduction. If all data of a decision table could be loaded in memory one time, the average time complexity of this algorithm will be $O(|U| \times |C| \times (|C| + \log|U|))$ and its space complexity will be $O(|U| + |C|)$. Simulation experimental results show its efficiency.

Keywords rough set; divide and conquer; attribute reduction; attribute order

1 引 言

粗集(Rough Set, RS)理论^[1]由波兰学者 Pawlak 教授于 1982 年提出,由于它能有效地分析和处理不精确、不一致、不完整等各种不完备信息,并能从中揭示潜在的规律,近年来在机器学习、数据挖掘等多个领域得到了广泛应用^[2].

在基于粗集理论的知识获取研究中,许多学者已经对属性约简的算法进行了大量的研究^[3-11].文献[3]给出了较好的启发函数,使其属性约简算法的

时间复杂度为 $O(|C|^3 \times |U|^2)$.文献[4]给出了基于条件信息熵的约简算法,其时间复杂度为 $O(|C|^2 \times |U|^2)$.文献[5-6]通过快速排序的方法来计算正区域,将基于正区域的属性约简算法的时间复杂度降为 $O(|C|^2 \times |U| \times \log|U|)$.文献[7]采用了基数排序的方法计算正区域,得到了时间复杂度为 $\max\{O(|C| \times |U|), O(|C|^2 \times |U/C|)\}$ 的属性约简算法.文献[8]给出了基于 Skowron 分辨矩阵的属性约简算法.文献[3-8]中的属性约简算法都是在无属性序的条件下给出的.对于属性序的属性约简算法,文献[9-11]进行了深入的研究,在给定属性序的条件下,文献

[9] 结合 Skowron 分辨矩阵给出了时间复杂度 $O(|U|^2 \times |C|)$ 的属性约简算法. 文献[10]给出了树表示下的属性约简算法, 其时间复杂度为 $O(|U| \times |C|^2)$. 然而, 在一次性将决策表的所有数据调入内存, 且不考虑这些数据本身所占内存的情况下, 这些算法的空间复杂度至少为 $O(|U| \times |C|)$, 有的甚至达到 $O(|U|^2 \times |C|)$, 当 $|U| > 10^5$ 时, 算法所需的辅助空间会占用大量内存, 导致算法性能急剧下降, 这也是现有的约简算法不能很好地处理大数据集约简问题的主要原因. 因此, 在设计属性约简算法时, 同时考虑时间复杂度和空间复杂度是非常必要的.

属性序的研究在面向领域用户的数据挖掘中具有重要意义^[11]. 本文在给定属性序下, 将分治法的思想融入到决策表的正区域计算和属性约简过程中, 提出一种新的属性约简算法. 该算法的平均时间复杂度为 $O(|U| \times |C| \times (|C| + \log |U|))$, 空间复杂度为 $O(|U| + |C|)$. 算法得到的属性约简是给定属性序下的唯一约简, 改进了文献[9-10]中的约简算法. 实验结果表明, 在 $|U| \gg |C|$ 和 $|C| \gg |U|$ 两种情况下, 本文给出的约简算法都能快速地得到决策表的约简, 适合海量数据集的属性约简处理.

本文第 2 节介绍有关 Rough 集理论和属性序的基本概念; 第 3 节提出一种新的属性约简算法, 该算法充分结合了改进的分辨矩阵和分治法的思想; 第 4 节给出在 KDDCUP99 数据集上的实验测试结果, 并通过自定义数据集, 在 $|U| \gg |C|$ 和 $|U| \ll |C|$ 两种情况下进行海量数据集的测试, 且对实验结果进行分析; 最后一节总结全文.

2 Rough 集和属性序的基本概念

定义 1(决策表^[2]). 一个决策表 $S = \langle U, A = C \cup D, V, f \rangle$, 其中 U 是对象的集合, 也称为论域, $A = C \cup D$ 是属性集合, C 和 D 分别称为条件属性集和决策属性集, $D \neq \emptyset$, V 是属性值的集合, $f: U \times A \rightarrow V$ 是一个信息函数, 它指定了 U 中每个对象 x 的属性值.

给定决策表 $S = \langle U, A = C \cup D, V, f \rangle$, Skowron 给出了信息系统的分辨矩阵^[8], Hu 给出了分辨矩阵的另一种形式^[3]. 然而, Hu 给出的分辨矩阵在不相容决策表中是不完备的, 文献[12-13]中给出了改进后的 Skowron 分辨矩阵.

定义 2^[14]. 给定决策表 $S = \langle U, A = C \cup D, V, f \rangle$, 设改进后的 Skowron 分辨矩阵为 M , M 中的分辨矩阵元素可以定义为 $B_{xy}^s = \{a \in C \mid f(x, a) \neq$

$f(y, a) \text{ 且 } w(x, y) = 1\}$, 其中 $x, y \in U$;

$$w(x, y) = \begin{cases} 1, & x \in POS_C(D), y \notin POS_C(D) \\ 1, & x \notin POS_C(D), y \in POS_C(D) \\ 1, & x, y \in POS_C(D), d(x) \neq d(y) \\ 0, & \text{其它情况} \end{cases}.$$

本文将改进后的 Skowron 分辨矩阵简称为 Skowron 分辨矩阵或者分辨矩阵. 此外, 为了描述简洁, 本文将 $f(x, a) (x \in U, a \in C)$ 记做 $a(x)$.

命题 1(基于 Skowron 分辨矩阵的约简规则^[9]). 令 M 是决策表 $S = \langle U, A = C \cup D, V, f \rangle$ 的分辨矩阵. $R (R \subseteq A \wedge R \neq \emptyset)$ 是一个约简, 当且仅当 $\forall_{a \in M} (a \neq \emptyset \rightarrow a \cap R \neq \emptyset)$.

我们给整个条件属性集合定义一个完整的序关系. 根据这个序关系, 我们可以给出 Skowron 分辨矩阵的等价关系. 为了本文后续介绍的方便, 在此先将文献[9]中关于属性序的基本概念进行简单介绍.

令 $S = \langle U, A = C \cup D, V, f \rangle$ 是一个决策表. 我们在 C 上定义了一个完整的序关系“ $<$ ”, 同时, 我们为 C 中的所有属性分别标上 1 到 $|C|$. 这样, 我们在 C 上就得到了一个关于属性的序列, 在本文中称为“属性序” $SO: c_1 < c_2 < \dots < c_{|C|}$ ^[9].

令 M 是 $S = \langle U, A = C \cup D, V, f \rangle$ 的 Skowron 分辨矩阵. $\forall \delta \in M$, δ 中的属性从左到右继承着序列 SO , 例如: $\delta = c_j B$, $c_j \in C$, B 是 C 中的一个属性子集, c_j 是在序 SO 下 δ 的第一个属性, 在这里, 我们把 c_j 叫做 δ 的标签属性.

对于 c_j , 我们定义集合: $L(SO) = \{\delta \mid \delta = c_j B, \delta \text{ 从左到右继承着序列 } SO, \delta \in M\}$. 容易验证: $L(SO)$ 是 M 上的等价关系, 而且将 M 划分为多个等价类. 可以用商集来表示它的划分: $M/L(SO) = \{[c_1], [c_2], \dots, [c_{|C|}]\}$. 因为 $[c_i] \cap [c_j] = \emptyset$ (当 $i \neq j$ 时), 因而这个划分是唯一的. 所以, Skowron 分辨矩阵中的每一个元素只属于一个等价类, 且该等价类由分辨矩阵元素的标签属性决定. 这个划分可以用属性的下标来表示: $M/L(SO) = \{[1], [2], \dots, [|C|]\}$ ^[9].

在由 $L(SO)$ 划分得到的等价类中, 下标最大的标签属性对属性约简起着非常重要的作用. 令 $N = \max\{|M/L(SO)|\}$, $1 \leq N \leq |C|$. 它的标签属性是 a_N . R 是一个约简, 假定 $R = \emptyset$.

算法 1^[9]. 属性序下的约简算法.

- (1) 令 c_N 是一个约简属性, $R = R \cup \{c_N\}$;
- (2) $E = \{a; a \cap \{c_N\} = \emptyset, a \in M\}$, $M = E$;
- (3) $N' = \max\{|M/L(SO)|\}$, $N = N'$;
- (4) 重复以上步骤, 直到 $M = \emptyset$.

则 R 是 $S = \langle U, A = C \cup D, V, f \rangle$ 的一个约简.

引理 1^[9]. 令 k_1, k_2 是属性下标, $[k_1], [k_2] \in \mathbf{M}/L(S)$. 它们的标签属性分别是 b_1 和 b_2 . 如果 $k_1 < k_2$, 那么, 都有 $\{b_1\} \cap \gamma = \emptyset$.

引理 2^[9]. 对于一个给定的属性序 SO , 假定用算法 1 计算得到的约简为 R , 则 c_N 是 R 必不可少的.

命题 2^[9]. 算法 1 对于 Pawlak 约简是完备的.

命题 3^[9]. 给定属性集 C 的一个序, 算法 1 得到的属性约简结果是唯一的.

3 属性序下决策表的属性约简

在基于决策表正区域不变的属性约简中^[3,5-9], 计算决策表的正区域是最重要的计算之一. 文献[7]给出了基于基数排序计算正区域的算法, 其时间复杂度为 $O(|C| \times |U|)$, 空间复杂度为 $O(|U| \times |C|)$; 文献[5-6, 15]给出了用快速排序的方法(快速排序的方法属于分治法^[16])计算正区域, 并指出其时间复杂度为 $O(|C| \times |U| \times \log |U|)$, 然而, 我们发现, 用快速排序对二维表排序的平均时间复杂度为 $O(|U| \times (|C| + \log |U|))$, 空间复杂度为 $O(|U|)$ ^[17]. 考虑到空间复杂度的问题, 本文采用快速排序的方法计算决策表的正区域, 具体方法参见文献[5-6, 17], 这里不再赘述, 下面给出属性序下的属性约简算法.

3.1 属性序下的属性约简算法

命题 4. \mathbf{M} 中的分辨矩阵元素 B_{xy}^i 不为空的充要条件是: 对象 x, y 中至少有一个属于 $POS_C(D)$ 且 $d(x) \neq d(y)$.

命题 5. 令 $[k] \in \mathbf{M}/L(S)$, 则 $[k] \neq \emptyset$ 的充要条件是: $\exists x, y \in U$, 满足以下两个条件:

- (1) \mathbf{M} 中的分辨矩阵元素 B_{xy}^i 不为空;
- (2) $\forall_{k_1(1 \leq k_1 < k)} (c_{k_1}(x) = c_{k_1}(y)) \wedge (c_k(x) \neq c_k(y))$.

命题 6. 给定决策表 $S = \langle U, A = C \cup D, V, f \rangle$, 给定属性序 $SO: c_1 < c_2 < \dots < c_{|C|}$, 属性 $c_k (c_k \in C \wedge 2 \leq k \leq |C|)$ 是非空标签属性的充要条件是: 在 Skowron 分辨矩阵中依次去掉包含属性 c_1, c_2, \dots, c_{k-1} 的分辨矩阵元素后, $\exists B_{xy}^i \in \mathbf{M}$ 满足 $c_k \in B_{xy}^i$.

结合定义 2、属性序的基本概念和引理 1、引理 2 的证明过程(见文献[9]), 可以证明命题 4、命题 5、命题 6. 由于篇幅的原因, 这里省略了命题 4~命题 6 的证明过程.

在给定属性序下, 算法 1 给出了一个求唯一 Pawlak 属性约简的完备算法, 但是该算法的平均、最坏时间复杂度都是 $O(|U|^2 \times |C|)$, 空间复杂度

也是 $O(|U|^2 \times |C|)$, 算法的时间复杂度是平方级的, 保持了与 $|C|$ 的线性关系, 但是空间复杂度太大, 当决策表的数据量较大时, 处理速度相当慢. 因此, 改进此算法, 获得与算法 1 等价的约简结果且时空复杂度较小的算法是很有必要的.

使用快速排序后, 计算正区域的算法的时间复杂度、空间复杂度都降低了, 如果我们把分治法的思想也加入到属性约简过程中, 就可以降低算法的复杂度. 分析算法 1 的复杂度后发现, 在进行算法 1 对决策表进行处理之前, 需要保存一个 Skowron 分辨矩阵, 这直接导致了算法 1 的时间、空间复杂度都必须在 $O(|U|^2 \times |C|)$. 因此, 必须找到一条途径, 使得不需要存储 Skowron 分辨矩阵, 同时又能满足算法 1 所需要的数据.

基于以上分析, 下面给出一个属性序下的快速约简算法.

3.1.1 计算非空标签等价类

从算法 1 可知: 计算属性序下的约简, 在得到决策表的正区域后, 需要得到非空标签属性集合. 命题 5 给出了计算非空标签属性集合的方法. 在这里, 首先给出计算非空标签属性集合的递归函数(递归函数 1)和具体的算法(算法 2).

递归函数 1(用于计算非空标签属性集合).

```
Void NonEmptyLabelAttr (int r, ObjectSet OSet)
//r 为属性的编号 (1 ≤ r ≤ |C|), OSet (OSet ∈ 2U)
//为对象集
{
    while (1 ≤ r ≤ |C|)
    {
        if |OSet| = 1, then return ;
        Elseif ∀x ∈ OSet x ∉ POSC(D), then return ;
        //根据命题 4 和命题 5
        Elseif ∀x, y ∈ OSet cr(x) = cr(y), then NonEmpty-
            LabelAttr (r+1, OSet);
        Else
        {
            NonEmptyLabel[r] = 1; //根据命题 5, 判断
            //属性 Cr 是否非空标签属性, 值为 1 表示
            //是非空标签属性
            将 OSet 划分成两部分 OSet1r, OSet2r; 设  $\bar{X}$  为
            OSet 中所有对象在属性 r 上属性值的平均
            值, 则 OSet1r, OSet2r 满足 ∀x ∈ OSet1r, y ∈ OS-
            et2r, 有
            cr(x) ≤  $\bar{X}$  < cr(y);
            NonEmptyLabelAttr (r, OSet1r);
            NonEmptyLabelAttr (r, OSet2r);
        }
    }
}
```

算法 2. 求 $L(SO)$ 划分得到的等价类的非空标签属性集合.

输入: 决策表 $S = \langle U, A = C \cup D, V, f \rangle$

输出: 决策表 S 的非空标签属性集合 R_1

1. 设 $C = \{c_1, c_2, \dots, c_{|C|}\}$, 给定一个属性序 $SO: c_1 < c_2 < \dots < c_{|C|}$.

2. $R_1 = \emptyset, r = 1, OSet_1 = U$;

For $j = 1$ to $|C|$ do

$NonEmptyLabel[j] = 0$;

3. 调用 $NonEmptyLabelAttr(1, OSet_1)$;

4. For $j = 1$ to $|C|$ do

 If $NonEmptyLabel[j] = 1$, then

$R_1 = R_1 \cup \{c_j\}$;

5. Return R_1 .

算法 2 的复杂度分析: 文献[17]中给出了二维表快速排序的平均时间复杂度和空间复杂度, 而算法 2 的时间复杂度与二维表快速排序的时间复杂度相同, 故算法 2 的平均时间复杂度为 $O(|U| \times (|C| + \log|U|))$, 空间复杂度为 $O(|U| + |C|)$.

3.1.2 基于属性序和分治法的属性约简算法

在得到决策表在属性序下的非空标签属性集合后, 接下来给出基于属性序和分治法的属性约简算法.

算法 3. 基于属性序和分治法的属性约简算法.

输入: 决策表 $S = \langle U, A = C \cup D, V, f \rangle$

输出: 决策表 S 的属性约简 Red

1. 设 $C = \{c_1, c_2, \dots, c_{|C|}\}$, 给定一个属性序 $SO: c_1 < c_2 < \dots < c_{|C|}$, $r = 1, U_1^1 = U, Red = \emptyset$;

2. 利用快速排序计算决策表的正区域 $POS_C(D)$;

3. 调用算法 2 计算决策表的非空标签属性集合 R_1 ;

4. 设 $c_{N'}$ 为 R_1 中标号最大的标签属性,

 If $c_{N'} \in Red$, then 转步 5;

 Else

$Red = Red \cup \{c_{N'}\}$, 且 $c_{N'}$ 放在 Red 的最后一位;

$R_1 = R_1 - \{c_{N'}\}$;

$C' = \emptyset$;

 合并属性集 Red 和 R_1 , 先将 Red 中的属性按标签属性下标的降序排列依次加入到 C' 中, 然后将 R_1 中属性按标签属性下标的升序排序依次加入到 C' 中;

$C = \emptyset$; $C = C'$;

 调用算法 2 计算决策表在新属性序下的非空标签属性集合 R_1 ;

 转步 4;

 End If

5. Return Red .

算法 3 的复杂度分析: 在算法 3 中, 步 2 的平均时间复杂度是 $O(|U| \times (|C| + \log|U|))$, 空间复杂

度为 $O(|U|)$; 步 3 的平均时间复杂度为 $O(|U| \times (|C| + \log|U|))$, 空间复杂度为 $O(|U| + |C|)$; 步 4 的平均时间复杂度为 $O(|U| \times |C| \times (|C| + \log|U|))$, 空间复杂度为 $O(|U| + |C|)$. 故算法 3 的平均时间复杂度为 $O(|U| \times |C| \times (|C| + \log|U|))$. 空间复杂度为 $O(|U| + |C|)$.

3.2 算法 3 与算法 1 输出结果的等价性证明

接下来, 我们证明: 算法 3 的输出结果与算法 1 的输出结果是相同的.

给定决策表 $S = \langle U, A = C \cup D, V, f \rangle$, S 的分辨矩阵为 M . 条件属性子集 $C_1 \subseteq C$, 设 \overline{M}^{C_1} 为这样一个集合: $\overline{M}^{C_1} = \{\alpha_1, \alpha_2, \dots, \alpha_t\}$, 满足: (1) $\forall \alpha_i \in \overline{M}^{C_1} \alpha_i \in M (1 \leq i \leq t)$; (2) $\forall c \in C_1$, 如果存在 $\alpha \in M$, 满足 $c \in \alpha$, 则 $\alpha \in \overline{M}^{C_1}$.

即, 在分辨矩阵 M 中, \overline{M}^{C_1} 是包含了 C_1 中至少一个属性的分辨矩阵元素的集合.

命题 7. 给定决策表 $S = \langle U, A = C \cup D, V, f \rangle$, 条件属性子集 $C_1 \subseteq C$. 根据 U 中所有对象在 C_1 上的取值将 U 划分成 $U_1, U_2, \dots, U_{|U/C_1|}$, 并根据 U/C_1 的划分将决策表 S 分解成 $|U/C_1|$ 个子决策表 $S_1, S_2, \dots, S_{|U/C_1|}$, 其中 $S_i = \langle U_i, C \cup D, V, f \rangle (1 \leq i \leq |U/C_1|)$. 设 S 的分辨矩阵为 M , S_i 的分辨矩阵为 $M_i (1 \leq i \leq |U/C_1|)$, 则 $\bigcup_{\alpha \in M} \alpha = \bigcup_{i=1}^{|U/C_1|} \bigcup_{\beta \in M_i} \beta \cup \bigcup_{\gamma \in \overline{M}^{C_1}} \gamma$ (注: 若 $\forall x, y \in U_i \forall c_1 \in C_1 (x) = c_1(y)$, 则 $M_i = \emptyset$).

证明. 先证明 $\bigcup_{\alpha \in M} \alpha \subseteq \bigcup_{i=1}^{|U/C_1|} \bigcup_{\beta \in M_i} \beta \cup \bigcup_{\gamma \in \overline{M}^{C_1}} \gamma$.

$\forall \alpha \in M$, 设 α 是对象 $x (x \in U)$ 和 $y (y \in U)$ 的分辨属性集合. 经过 U/C_1 的划分后, 设 x 和 y 分别被划分到子决策表 $S_i (1 \leq i \leq |U/C_1|)$ 和 $S_j (1 \leq j \leq |U/C_1|)$ 中.

若 $i = j$, 则 x 和 y 属于同一个子决策表 S_i , 显然有 $\alpha \in M_i$, 且 $\alpha \in \bigcup_{i=1}^{|U/C_1|} \bigcup_{\beta \in M_i} \beta$.

否则, $i \neq j$, x 和 y 不属于同一个子决策表, 则 $\alpha \in \overline{M}^{C_1}$, 且 $\alpha \in \bigcup_{\gamma \in \overline{M}^{C_1}} \gamma$.

因而, $\alpha \subseteq \bigcup_{i=1}^{|U/C_1|} \bigcup_{\beta \in M_i} \beta \cup \bigcup_{\gamma \in \overline{M}^{C_1}} \gamma$, 即 $\bigcup_{\alpha \in M} \alpha \subseteq \bigcup_{i=1}^{|U/C_1|} \bigcup_{\beta \in M_i} \beta \cup \bigcup_{\gamma \in \overline{M}^{C_1}} \gamma$.

同理可证 $\bigcup_{\alpha \in M} \alpha \supseteq \bigcup_{i=1}^{|U/C_1|} \bigcup_{\beta \in M_i} \beta \cup \bigcup_{\gamma \in \overline{M}^{C_1}} \gamma$. 故命题 7

得证.

证毕.

分析递归函数 1 可知, 在条件属性子集 $C_1 (C_1 \subseteq C)$ 上, 通过递归函数 1 的反复调用, 可以将 U 分解为 $U_1^{C_1}, U_2^{C_1}, \dots, U_p^{C_1} (p$ 为递归函数 1 在条件属性子

集 C_1 将 U 划分的子集的数目, $1 \leq p \leq |U/C_1|$, 并满足如下条件: $U = U_1^{C_1} \cup U_2^{C_1} \cup \dots \cup U_p^{C_1}$ ($U_i^{C_1} \cap U_j^{C_1} = \emptyset$ ($1 \leq i < j \leq p$)) 且有 $\forall_{x,y \in U_k^{C_1}} \forall_{c \in C_1} c(x) = c(y)$ ($1 \leq k \leq p$) 和 $\forall_{x \in U_i^{C_1}} \forall_{z \in U_l^{C_1}} \exists_{c \in C_1} c(x) \neq c(z)$ ($1 \leq i < l \leq p$).

命题 8. 给定决策表 $S = \langle U, A = C \cup D, V, f \rangle$, 条件属性子集 $C_1 \subseteq C$. 根据 U 中所有对象在 C_1 上的取值将 U 划分成 $U_1, U_2, \dots, U_{|U/C_1|}$, 与之对应的子决策表为 S_i , S_i 的分辨矩阵为 $\mathbf{M}_i^{C_1}$ ($1 \leq i \leq |U/C_1|$). 在 C_1 上, 通过递归函数 1 的递归调用, 将 U 分解为 $U_1^{C_1}, U_2^{C_1}, \dots, U_p^{C_1}$ ($1 \leq p \leq |U/C_1|$), 与之对应的子决策表为 $S_1^{C_1}, S_j^{C_1}$ 的分辨矩阵为 $\mathbf{M}_j^{C_1}$ ($1 \leq j \leq p$). 则 $\bigcup_{i=1}^{|U/C_1|} \bigcup_{\alpha \in \mathbf{M}_i} \alpha = \bigcup_{i=1}^p \bigcup_{\beta \in \mathbf{M}_i^{C_1}} \beta$ (注: 如果 $\forall_{x,y \in U_j^{C_1}} \forall_{c \in C} c(x) = c(y)$, 则 $\mathbf{M}_j^{C_1} = \emptyset$).

证明. 先证 $\bigcup_{i=1}^{|U/C_1|} \bigcup_{\alpha \in \mathbf{M}_i} \alpha \supseteq \bigcup_{i=1}^p \bigcup_{\beta \in \mathbf{M}_i^{C_1}} \beta$.
对于 $U_k^{C_1}$ ($1 \leq k \leq p$), 有 3 种情况: (1) 如果 $\forall_{x \in U_k^{C_1}} x \notin POS_C(D)$, 则 $\mathbf{M}_k^{C_1} = \emptyset$; (2) 如果 $\forall_{x,y \in U_j} \forall_{c \in C} c(x) = c(y)$, 则 $\mathbf{M}_j^{C_1} = \emptyset$; (3) $\exists_{x,y \in U_k^{C_1}} (x \in POS_C(D) \wedge \exists_{c \in C} c(x) \neq c(y))$, 此时 $\mathbf{M}_j^{C_1} \neq \emptyset$. 我们只需要分析第三种情况. 根据递归函数 1 的调用过程可知, 必然存在 U_q ($1 \leq q \leq |U/C_1|$), 满足 $U_k^{C_1} = U_j$, 即 $\mathbf{M}_k^{C_1} = \mathbf{M}_q$, 故 $\bigcup_{i=1}^{|U/C_1|} \bigcup_{\alpha \in \mathbf{M}_i} \alpha \supseteq \bigcup_{i=1}^p \bigcup_{\beta \in \mathbf{M}_i^{C_1}} \beta$.

同理可得 $\bigcup_{i=1}^{|U/C_1|} \bigcup_{\alpha \in \mathbf{M}_i} \alpha \subseteq \bigcup_{i=1}^p \bigcup_{\beta \in \mathbf{M}_i^{C_1}} \beta$, 故命题 8 得证.
证毕.

命题 9. 给定决策表 $S = \langle U, A = C \cup D, V, f \rangle$, 给定属性序 $SO: c_1 < c_2 < \dots < c_{|C|}$, $C_1 = \{c_1, c_2, \dots, c_k\}$ ($1 \leq k \leq |C|$). 则, 在 C_1 上调用递归函数 1 的过程实质上是消除等价类 $[c_1], [c_2], \dots, [c_k]$, 即在 Skowron 分辨矩阵 \mathbf{M} 中不断消除所有标签属性为 c_1, c_2, \dots, c_k 的分辨矩阵元素的过程.

证明. 从命题 8 可知: 从消除 Skowron 分辨矩阵元素的角度分析, 在 C_1 上调用递归函数 1 的过程是划分等价类 U/C_1 的过程. 根据第 3 节属性序的基本知识易知命题成立.
证毕.

根据命题 6、命题 9 和算法 2 可知, 算法 2 的输出结果 R_1 与文献[9]中非空标签属性集合的定义是一致的.

命题 10. 给定决策表 $S = \langle U, A = C \cup D, V, f \rangle$ 和属性序 $SO: c_1 < c_2 < \dots < c_{|C|}$, 算法 3 的输出结果 Red 和算法 1 的输出结果 R 是相同的.

证明. 由命题 6 和命题 9 可知, 算法 1 中的步

2 等价于在算法 3 中调用一次递归函数 1. 因此, 算法 1 的整个处理过程等价于算法 3 中的步 4, 故 $Red = R$.
证毕.

4 实验结果

为了验证本文方法的有效性, 我们进行了多组测试. 首先, 我们采用 KDDCUP99 数据集进行测试; 然后, 通过自定义数据集, 在 $|U| \gg |C|$ 和 $|U| \ll |C|$ 两种条件下分别测试算法 3 的性能.

4.1 KDDCUP99 数据集测试

为了考察算法 3 在大数据集环境下的运行性能, 我们采用了 KDDCUP99 入侵检测数据集进行测试 (数据集下载地址: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>), 原 KDDCUP99 数据集共有 4898432 条记录, 每条记录有 41 个属性. 我们分别从中随机选出 10%, 20%, ..., 100% 记录生成新的数据集, 采用等频率离散化方法^[2]对这 10 个数据集进行离散化处理, 所有记录在各个属性上的取值范围是 0~255, 这样, 只需要一个字节就可以存储一个属性值. 用算法 3 对这些数据集进行测试, 测试结果见表 1, 其中 T 表示算法的运行时间, 单位为 s; N 表示约简结果的条件属性个数, $MemUse$ 表示程序在运行过程中使用的最大内存, 单位为 KB. 本实验的硬件测试环境是: CPU 为 Intel Pentium4 2.4GHz, 内存为 512MB, 操作系统为 WindowsXP, 开发工具为 VC++ 6.0.

表 1 KDDCUP99 数据集测试结果

记录数比例/%	记录数	属性数	T/s	N	MemUse
10	489843	41	61.232417	29	198804
20	979686	41	135.023143	30	246972
30	1469529	41	218.743598	31	303232
40	1959372	41	283.406260	30	344984
50	2449216	41	384.723173	32	395144
60	2939059	41	469.103032	32	444576
70	3428902	41	602.920099	34	493672
80	3918745	41	661.663162	33	543172
90	4408588	41	753.895816	33	592372
100	4898432	41	13337.205877	34	641896

从表 1 可以看出, 对于前 9 个数据集, 算法 3 的处理速度是相当快的. 但是, 当记录数增加到 4898432 时, 算法 3 的运行时间突然急剧增加. 从算法 3 的时间复杂度的表达式分析, 不应该出现这样的运行时间急剧增加的情况. 分析整个运行过程, 我们发现: 当记录数增加到 4898432 时, 算法 3 使用的内存过大, 而此时 CPU 的利用率低于 10%, 使得算法 3 的性能急剧下降, 从而运行时间急剧增加. 由此

可见,对于大数据集的处理,算法的空间复杂度非常重要,在某些情况下可能比时间复杂度更重要.

4.2 自定义数据集测试

为了充分测试算法 3 的性能,在 $|U| \gg |C|$ 和 $|U| \ll |C|$ 两种情况下,我们自定义了两组数据进行了测试. 对这两组数据进行实验测试的硬件测试环境都是:CPU 为 Intel Pentium4 2.4GHz,内存为 256MB,操作系统为 WindowsXP,开发工具为 VC++ 6.0.

随机生成 20 个记录集,记录集的记录数由 1×10^5 逐渐增加到 3×10^6 ,每条记录含有 15 个条件属性和 1 个决策属性,每条记录的条件属性和决策属性都在(0~9)上随机取值,用算法 3 对这些数据集进行测试,测试结果见图 1. 属性序由随机生成的属性标号任意给定.

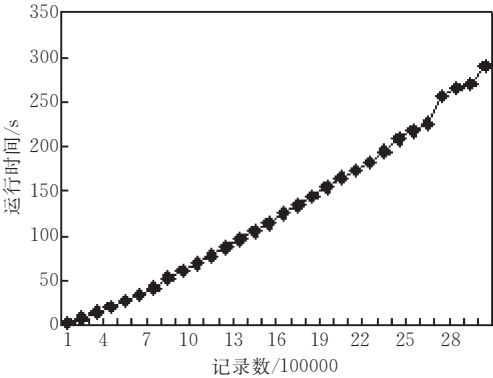


图 1 自定义数据集测试结果($|U| \gg |C|$)

从图 1 可以看出,算法 3 是高效的.

随机生成 10 个记录集,记录集的条件属性数由 0.5×10^4 逐渐增加到 5×10^4 ,决策属性个数为 1,每个记录集含有 1000 条记录,每条记录的条件属性和决策属性都在(0~4)上随机取值,用算法 3 对这些数据集进行测试,测试结果见图 2. 属性序由随机生成的属性标号任意给定.

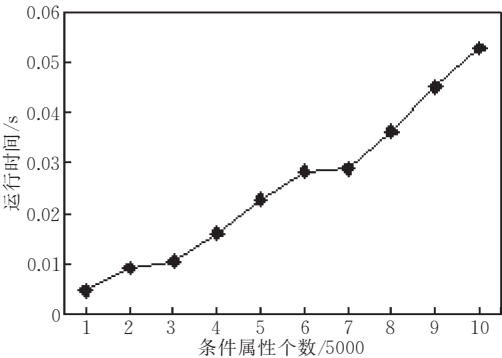


图 2 自定义数据集测试结果($|U| \ll |C|$)

从图 2 可以看出,当 $|U| \ll |C|$ 时,算法的执行速

度非常快. 从算法 3 的时间复杂度 $O(|U| \times |C| \times (|C| + \log|U|))$ 分析,其运算速度不应该这么快. 分析实验过程的每一步结果之后发现,随机生成的决策表为相容决策表,只需要大约 $\log|U|$ 个属性就可将数据集中所有对象区分开,因而算法 2 的时间复杂度降低到 $O(|U| \log|U|)$. 非空标签属性集合的势一般不超过 $\log|U|$,这导致算法 3 的时间复杂度下降到 $O(|U| \log^2|U|)$,这也是算法 3 运行效率高的原因.

此外,使用图 1 和图 2 中同样的数据集,我们对算法 1 进行了测试. 实验结果是:算法 1 对于 10 个数据集均产生内存不足的现象,程序非正常终止,主要原因还是算法 1 的空间复杂度太大.

5 结束语

虽然 Rough 集理论正日渐成熟,但是还没有能够在工业中取得广泛的应用. 一个重要原因是: Rough 集对于属性约简(特征提取)的优势,在海量数据集面前显得效率不够高. 已有的许多属性约简算法对于空间复杂度考虑不足,导致了算法不能适应大数据集的约简处理. 本文在属性序的基础上,将分治递归的思想融入到属性约简的过程中,设计了平均时间复杂度为 $O(|U| \times |C| \times (|C| + \log|U|))$ 、空间复杂度为 $O(|U| + |C|)$ 的约简算法. 该算法所得的属性约简结果与文献[9]中算法所得的结果相同. 实验结果表明,该算法在 $|U| \gg |C|$ 和 $|U| \ll |C|$ 两种条件下都能快速得到约简结果. 然而,算法本身存在着一个局限性:必须要以给定属性序为前提条件,当然,在面向领域用户的数据处理中,属性序的获得并不困难. 在无属性序的条件下,如何快速地利用分治法获取属性序将是我们进一步的研究工作. 另外,如何将分治法用于值约简等处理也是我们接下来需要研究的问题.

参 考 文 献

[1] Pawlak Z. Rough Set. International Journal of Computer and Information Sciences, 1982, 11: 341-356

[2] Wang Guo-Yin. Rough Set Theory and Knowledge Acquisition. Xi'an: Xi'an Jiaotong University Press, 2001 (in Chinese)

(王国胤. Rough 集理论与知识获取. 西安:西安交通大学出版社, 2001)

[3] Hu X H, Cercone N. Learning in relational database: A rough set approach. International Journal of Computational Intelligence, 1995, 11(2): 323-338

- [4] Wang Guo-Yin, Yu Hong, Yang Da-Chun. Decision table reduction based on conditional information entropy. Chinese Journal of Computers, 2002, 25(7): 759-766(in Chinese)
(王国胤, 于洪, 杨大春. 基于条件信息熵的决策表约简. 计算机学报, 2002, 25(7): 759-766)
- [5] Nguyen S H et al. Some efficient algorithms for rough set methods//Proceedings of the Conference on Information Processing and Management of Uncertainty in Knowledge Based Systems. Granada, Spain. 1996: 1451-1456
- [6] Liu Shao-Hui, Sheng Qiu-Jian, Wu Bin. Research on efficient algorithms for rough set methods. Chinese Journal of Computers, 2003, 40(5): 637-642(in Chinese)
(刘少辉, 盛球战, 吴斌等. Rough 集理论高效算法的研究. 计算机学报, 2003, 5(26): 524-529)
- [7] Xu Zhang-Yan, Liu Zuo-Peng, Yang Bing-Ru, Song Wei. A quick attribute reduction algorithm with complexity of $\max\{O(|C||U|), O(|C|^2|U/C|)\}$. Chinese Journal of Computers, 2006, 29(3): 391-399(in Chinese)
(徐章艳, 刘作鹏, 杨炳儒, 宋威. 一个复杂度为 $\max\{O(|C||U|), O(|C|^2|U/C|)\}$ 的快速属性约简算法. 计算机学报, 2006, 29(3): 391-399)
- [8] Skowron A, Rauszer C. The discernibility functions matrices and functions in information systems//Slowinski R ed. Proceedings of the Intelligent Decision Support — Handbook of Applications and Advances of the Rough Sets Theory. Dordrecht: Kluwer Academic Publisher, 1991: 331-362
- [9] Wang Jue, Wang Ju. Reduction algorithms based on discernibility matrix: The order attributes method. Journal of Computer Science and Technology, 2001, 16(6): 489-504
- [10] Zhao Min. The data description based on reduct[Ph. D. dissertation]. Institute of Automation, Chinese Academy of Sciences, Beijing, 2004(in Chinese)
(赵岷. 基于 Reduct 理论的数据描述[博士学位论文]. 中国科学院自动化研究所, 北京, 2004)
- [11] Han S Q, Wang J. Reduct and attribute order. Journal of Computer Science and Technology, 2004, 19(4): 429-449
- [12] Ye Dong-Yi, Chen Zhao-Jiong. A new discernibility matrix and the computation of a core. Acta Electronica Sinica, 2002, 30(7): 1086-1088(in Chinese)
(叶东毅, 陈昭炯. 一个新的差别矩阵及其求核方法. 电子学报, 2002, 30(7): 1086-1088)
- [13] Wang Guo-Yin. The computation method of core attribute in decision table. Chinese Journal of Computers, 2003, 26(5): 611-615(in Chinese)
(王国胤. 决策表核属性的计算方法. 计算机学报, 2003, 26(5): 611-615)
- [14] Li Ding-Fang, Li Gui-Bin, Zhang Wen. $U/\{a\}$ partition based smallest reduction construction. Journal of Wuhan University (Nat Sci Ed), 2005, 51(3): 269-272(in Chinese)
(李订芳, 李贵斌, 章文. 基于 $U/\{a\}$ 划分的最小约简构造. 武汉大学学报(理学版), 2005, 51(3): 269-272)
- [15] (Hu Ke-Yun, Lu Yu-Chang, Shi Chun-Yi. Advances in rough set theory and its applications. Journal of Tsinghua University (Sci & Tech), 2001, 41(1): 64-68(in Chinese)
(胡可云, 陆玉昌, 石纯一. 粗糙集理论及其应用进展. 清华大学学报, 2001, 41(1): 64-68)
- [16] Yu Xiang-Xuan, Cui Guo-Hua, Zhou Hai-Ming. Fundamentals of Computer Algorithms. Wuhan: Huazhong Univ Press, 2001(in Chinese)
(余祥宣, 崔国华, 邹海明. 计算机算法基础. 武汉: 华中科技大学出版社, 2001)
- [17] Hu Feng, Wang Guo-Yin. Analysis of the complexity of quick sort for two dimension table. Chinese Journal of Computers, 2007, 30(6): 963-968(in Chinese)
(胡峰, 王国胤. 二维表快速排序的复杂度分析. 计算机学报, 2007, 30(6): 963-968)



HU Feng, born in 1978, Ph.D. candidate, lecturer. His research interests include intelligent information processing, data mining, etc.

WANG Guo-Yin, born in 1970, professor, Ph.D. supervisor. His research interests include intelligent information processing, data mining, rough set, neural network, etc.

Background

This paper is partially supported by the National Natural Science Foundation of China under Grants No. 60373111 and No. 60573068, Program for New Century Excellent Talents in University (NCET), Natural Science Foundation of Chongqing under grant No. 2005BA2003, Science & Technology Research Program of Chongqing Education Commission under grant No. KJ060517.

In the research of rough set theory, many researchers have proposed many algorithms for attribute reduction. However, it is difficult for the existed attribute reduction al-

gorithms to process huge data sets. There are two reasons. One is the time complexity, and the other is the space complexity. Especially in processing huge data set, the space complexity of an algorithm will affect greatly its efficiency. In this paper, a quick algorithm for attribute reduction of ordered attributes is proposed based on the divide and conquer method. Its average time complexity is $O(|U| \times |C| \times (|C| + \log|U|))$ and space complexity is $O(|U| + |C|)$. The algorithm is suitable for huge data processing. It may be helpful to put rough set theory into industry applications.