

【电子与信息科学 / Electronics and Information Science】

基于改进差别矩阵的增量式属性约简算法

冯少荣, 张东站

厦门大学信息科学与技术学院, 福建 厦门 361005

摘要: 研究目前粗糙集中求属性核和属性约简存在的效率低下问题, 提出基于改进差别矩阵的核增量式更新算法, 用于解决对象动态增加情况下核的更新问题. 为降低现有增量式属性约简算法的时间和空间复杂度, 提出一种不存储差别矩阵的高效属性约简算法, 用于处理对象动态增加情况下属性约简的更新问题. 理论及实验结果表明, 该算法可明显降低时间和空间的复杂度.

关键词: 粗糙集理论; 属性约简; 差别矩阵; 属性核; 决策表; 动态更新; 增量式算法; 知识约简; 时间复杂性; 空间复杂性;

中图分类号: TP 311

文献标识码: A

doi: 10.3724/SP.J.1249.2012.05405

Increment algorithm for attribute reduction based on improvement of discernibility matrix

FENG Shao-rong[†] and ZHANG Dong-zhan

School of Information Science and Technology, Xiamen University, Xiamen, Fujian 361005, P. R. China

Abstract: An incremental updating algorithm for computing core based on an improved discernibility matrix definition is proposed to improve the efficiency of computing attribute core and attribute reduction in rough sets. This new algorithm is mainly used to solve core updating when objects are dynamically increased. The purpose of this said algorithm is to decrease the complexity of time and space on the existing incremental attribute reduction algorithm. The discernibility matrix is not necessary to be stored and therefore the attribute reduction is updated when objects are dynamically increased. Theoretical analysis and experimental results have shown that this new algorithm is feasible and effective.

Key words: rough set theory; attribute reduction; discernibility matrix; attribute core; decision table; dynamic updating; Incremental algorithm; knowledge reduction; time complexity; space complexity

知识约简是粗糙集理论^[1]的核心内容之一. 在粗糙集理论中, 知识约简分为属性约简和属性值约简. 随着数据库系统中数据的不断增加, 属性约简相对属性值约简更加有效. 它大大简化了数据库结构的复杂度, 提高了人们对隐含在数据库庞大数据

量下的各种信息的认识程度. 因此, 属性约简成为目前粗糙集的研究热点之一^[2-5]. 决策表的属性约简通常是不唯一的, 人们希望能找到具有最少属性的最佳约简. 然而, 找到最佳约简是一个 NP-hard 问题, 导致 NP-hard 的主因是属性的组合爆炸, 解

Received: 2010-05-04; Revised: 2012-06-03; Accepted: 2012-07-27

Foundation: National Natural Science Foundation of China (50604012)

[†] Corresponding author: Associate professor FENG Shao-rong. E-mail: shaorong@xmu.edu.cn

Citation: FENG Shao-rong, ZHANG Dong-zhan. Increment algorithm for attribute reduction based on improvement of discernibility matrix [J]. Journal of Shenzhen University Science and Engineering, 2012, 29(5): 405-411. (in Chinese)

<http://journal.szu.edu.cn>

决该问题通常采用启发式搜索方法, 求出最佳或次最佳约简^[6-8]. 现有的属性约简大体上可分为, 基于差别矩阵或基于差别矩阵的改进的属性约简算法、基于正区域的属性约简算法及启发式的属性约简算法, 其共同特点是: ① 利用属性的重要性作为启发式信息, 但并不完备^[9]; ② 这些算法大都针对静态的信息系统或决策表, 不适合信息系统或决策表动态变化的情况. 现有的有关属性约简的增量式算法不多, 已有的动态增量属性约简算法, 其时间和空间复杂度都较高. 很多属性约简是从求属性核开始的, 所以求核是粗糙集理论的关键之一, 求解属性核也面临同样问题.

本研究围绕粗糙集中的求属性核和属性约简问题, 分析现有的求核算法和属性约简算法^[10-11]时间和空间复杂度高的原因, 基于差别矩阵对称性, 通过改进差别矩阵定义, 改进增量式求属性核算法及高效的属性约简完备算法. 理论分析和实验验证都表明, 本研究提出的算法有效可行, 可明显降低时

间和空间复杂度.

1 改进的差别矩阵定义及求核算法

1.1 改进的差别矩阵定义

按照文献 [11] 差别矩阵定义, U_1 为论域, x_i 和 x_j 是论域 U_1 中的元素, m_{ij} 和 m_{ji} 是差别矩阵 M 中的元素项, 在计算差别矩阵时, 当 $x_i \in U_1, x_j \in U_1$ 时, $m_{ij} = m_{ji}$, 文献 [11] 同时计算了 m_{ij} 及 m_{ji} . 由于差别矩阵是对称的, 因此只需计算 m_{ij} 就能正确求核. 当 $|U_1|$ 较大时, 可显著降低计算量, 特别是当决策表一致时, 减少的不必要计算量是求核实际所需计算量的 1/2.

故当 $x_i \in U_1, x_j \in U_1$ 时, 通过限定 $i > j$, 只计算 m_{ij} , 而不需计算与其对称且相等的 m_{ji} . 因此, 本研究对给定信息系统 (information system, IS), 将文献 [11] 的差别矩阵 $M = \{m_{ij}\}$ 重新定义为

$$m_{ij} = \begin{cases} \{a \in C : f(x_i, a) \neq f(x_j, a)\}, & f(x_i, D) \neq f(x_j, D) \text{ 且 } x_i \in U_1, x_j \in U_1, i > j \\ \{a \in C : f(x_i, a) \neq f(x_j, a)\}, & x_i \in U_1, x_j \in U_2 \\ \emptyset & \text{其他} \end{cases}$$

其中, $U_1 = \bigcup_{i=1}^k C_0(\Psi_i)$, Ψ_i 是基于 C_* 导出的 k 个划分; $U_2 = U - U_1$; $U_2' = \text{delrep}(U_2)$, $\text{delrep}()$ 是利用 U_2 产生 U_2' 的函数, 具体过程同文献 [11].

1.2 改进的求核算法

针对动态增加的情况, 文献 [11] 中算法 2 (optimization for incremental updating algorithm of a core, OIUAC) 的时间复杂度为 $O(5 \times |U_1| + 3 \times |U_2'|)$, 时间复杂度较低. 该算法的空间复杂度为 $O(|U_1| \times (|U_1| + |U_2'|))$, 由于该算法在计算核时使用了差别矩阵, 导致空间复杂度较高, 特别当决策一致时, $U_1 = U, U_2' = 0$, U 为论域, 空间复杂度为 $O(|U|^2)$, 不能有效处理大数据集. 为此, 本研究通过改进文献 [11] 中的算法 2, 由于差别矩阵是对称矩阵, 只存储差别矩阵下三角部分, 降低空间复杂度.

基于本研究重新定义的差别矩阵, 新的求属性核的算法流程如图 1. 其中, C 为条件属性集合; f 为属性集到值域集的映射; $\text{count}(\geq 1)$ 为 m_{ij} 在差别矩阵 M 中出现的次数; $\text{DMSC}(C)$ 为 C 的核.

2 改进的核增量式算法

2.1 算法原理

在只存储差别矩阵下三角部分情况下, 针对文献 [11] 的 3 种情况, 本研究分别做如下处理:

① 当 x 与 $(U_1 \cup U_2')$ 一致时, 计算 x 与 $U_1 \cup U_2'$ 间的 m_{ij} , 并按如图 2 的处理过程, 将 m_{ij} 增至 $\text{DMSC}(C)$, $U_1 = U_1 \cup \{x\}$.

② 当 x 与 U_1 不一致时, 在 U_1 中找出与 x 不一致的 y , 计算 y 与 $U_1 \cup U_2'$ 之间的 m_{ij} , 并把 $\text{DMSC}(C)$ 中相应的 m_{ij} 依据图 2 处理过程删除; 令 $U_2' = U_2' \cup \{y\}$, $U_1 = U_1 - \{y\}$, 并计算 y 与 U_1 之间的 m_{ij} , 按图 2 处理过程, 将 m_{ij} 增至 $\text{DMSC}(C)$.

③ 当 x 与 U_2' 不一致时, $\text{DMSC}(C)$ 保持不变.

改进的核增量式算法 (improved optimization for incremental updating algorithm of a core, IOIUAC) 流程图如图 2. 其中, $\text{core}(C)$ 表示最终得到的条件属性集 C 的核.

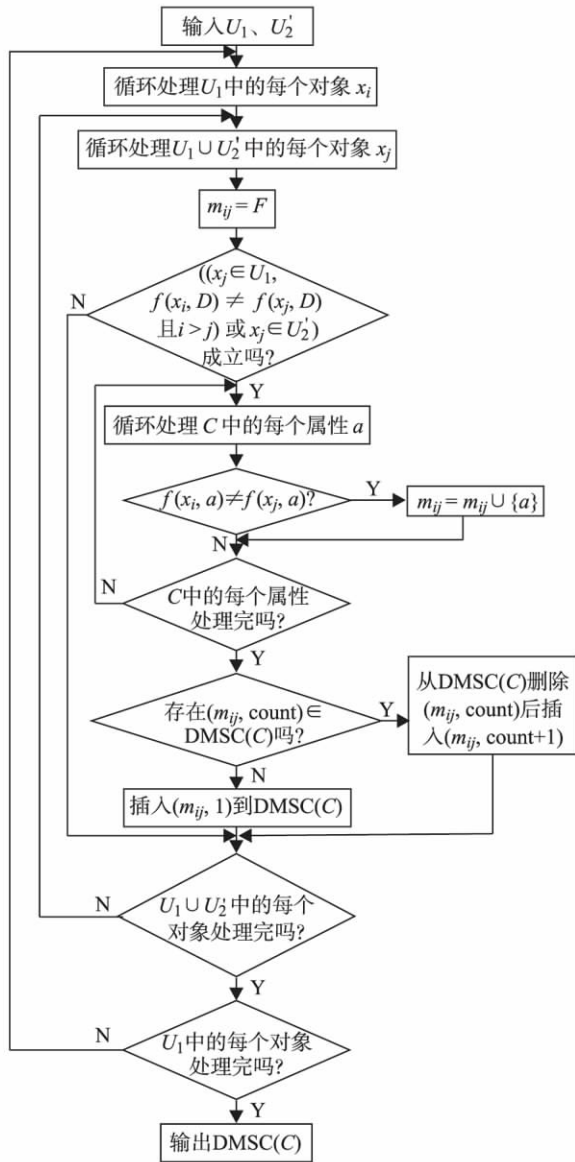


图 1 改进的求核算法

Fig. 1 Improved algorithm of the computation of a core

2.2 算法分析

① 当 x 与 $(U_1 \cup U_2)$ 一致时, 将 x 加入 U_1 , 文献 [11] 对差别矩阵增加了 1 行 1 列, 由于 $m_{ij} = m_{ji}$, 当 $x_i \in U_1, x_j \in U_1$ 时, 增加的列对于核计算来说是多余的, 所以 IOIUAC 只需计算相应的行; 此时计算量为 $|C| \times (|U_1| + |U_2|)$. 判断 x 与 $(U_1 \cup U_2)$ 是否一致, 所需计算量为 $|U_1| + |U_2|$. 故在此情况下, IOIUAC 的总时间复杂度为 $2 \times |C| \times (|U_1| + |U_2|)$.

② 当 x 与 U_1 不一致时, 计算 U_1 中与 x 不一致的对象 y , 并删除 DMSC(C) 中 y 所在行的单个属性,

其计算量为 $|U_1| + |U_2|$. 然后 y 作为 U_2 中的对象计算相应的 DMSC(C), 其计算量为 $|U_1|$. 判断一致性的时间为 $|C| \times (|U_1| + |U_2|)$, 所以 x 与 U_1 不一致时总的时间为 $|C| \times (3 \times |U_1| + 2 \times |U_2|)$. 故 IOIUAC 的时间复杂度为 $O(|C| \times (3 \times |U_1| + 2 \times |U_2|))$, 小于文献 [11] 中算法 2 的时间复杂度 $O(|C| \times (5 \times |U_1| + 3 \times |U_2|))$.

③ 当 x 与 U_2 不一致时, 文献 [11] 算法 2 空间复杂度为 $O(|U_1| \times (|U_1| + |U_2|))$, 而算法 IOIUAC 空间复杂度为 $O(|C|)$, $|C|$ 为条件属性数. 由于 $|C| \ll |U_1| + |U_2|$, 故 IOIUAC 较文献 [11] 中算法 2 的空间复杂度有显著改善.

2.3 实验

在内存为 1 024 MB, CPU 为 PIV 2.9 GHz, 操作系统为 Windows XP 的联想 PC 上, Eclipse 下 Java 实现文献 [11] 中算法 OIUAC 及本研究提出的 IOIUAC. 利用 UCI 提供的蘑菇数据库 (mushroom) 进行实验, 该数据库有 8 124 个对象. 将蘑菇数据库看作决策表, 并进行 2 个实验.

实验 1 从 8 124 个对象中选择 7 000 个对象作为基准决策表 (基准决策表即该表生成的差别矩阵作为 OIUAC 和 IOIUAC 的输入), 从剩下 1 124 个对象中依次选择 200、500、800 和 1 124 个对象作为增量, 执行 OIUAC 和 IOIUAC, 运行结果如图 3.

实验 2 由蘑菇数据库生成 8 000 个对象, 其中有 1 000 个不一致对象, 从生成的 8 000 个对象中选择 7 500 条作为基准决策表, 从剩下的 500 个对象中依次选择 100、200、300 和 500 个对象作为增量, 执行 OIUAC 和 IOIUAC, 运行结果如图 4.

2.4 实验分析

实验 1 8 124 个对象为一致性对象. OIUAC 计算及遍历差别矩阵相应的行和列, 而 IOIUAC 只需计算 1 行, 又因我们优化了核属性计算算法, 所以 IOIUAC 的计算时间比较少.

实验 2 当 x 与 U_1 不一致时, OIUAC 首先遍历差别矩阵中与 x 相应的行和列, 删除 DMSC(C) 中相关的核属性, 然后把该对象插入 U_2 , 计算 U_2 中 x 相应的差别矩阵, 把核属性插入 DMSC(C). 此时, IOIUAC 的计算量明显少于 OIUAC 的计算量.

对实验过程的监测结果显示, OIUAC 因为存储差别矩阵, 随着对象数从 7 200 个增至 8 124 个, 内存使用量从 225 Mbit 增至 285 Mbit; 而 IOIUAC

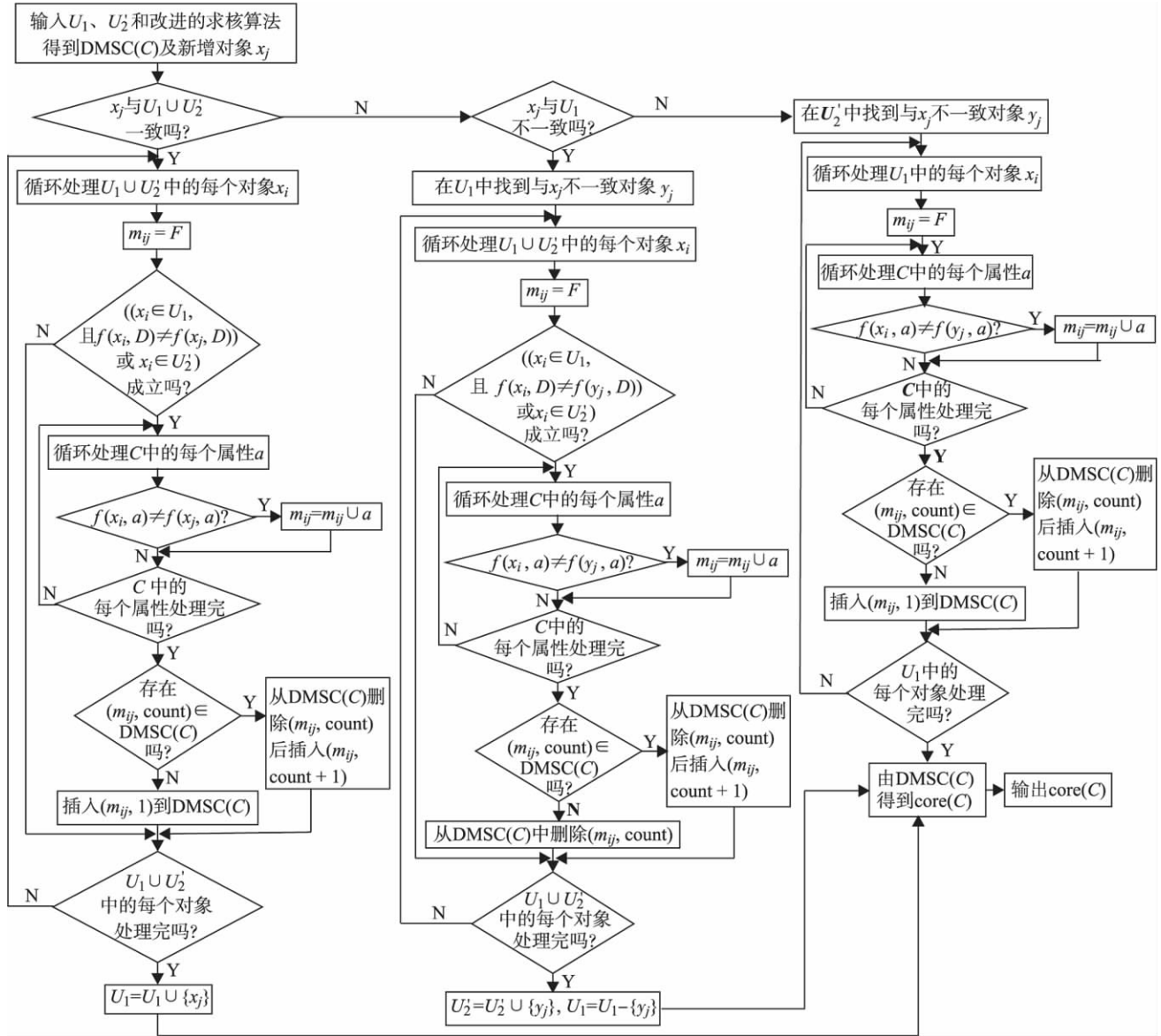


图 2 改进的核增量式算法

Fig. 2 Improved optimization for incremental updating algorithm of a core

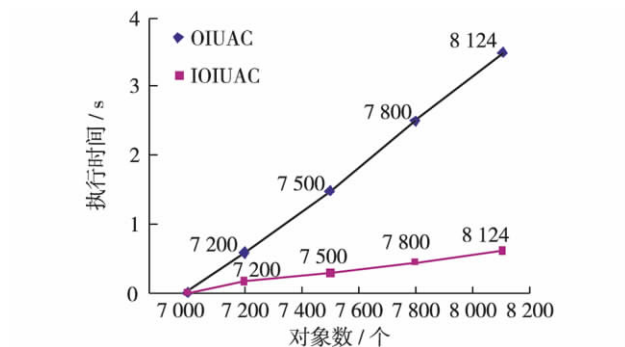


图 3 第 1 组实验算法的执行时间
Fig. 3 Algorithm running time in first group experiment

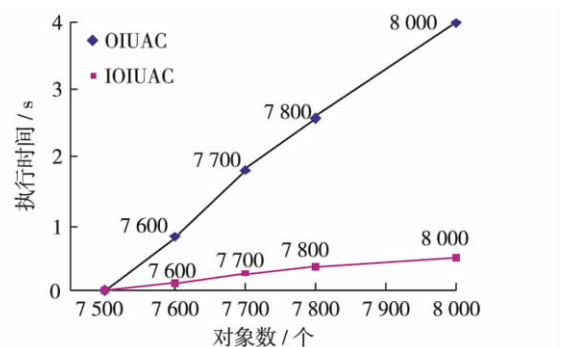


图 4 第 2 组实验算法的执行时间
Fig. 4 Algorithm running time in second group experiment

<http://journal.szu.edu.cn>

内存使用量一直保持小幅增加, 增幅仅约 20 M. 所以, IOIUAC 较 OIUAC 的另一重要优势为前者可应对大数据集的挑战.

3 改进的属性约简完备算法及其分析

文献 [12] 的增量式属性约简算法多次调用求核过程, 其时间复杂度为 $O(|U_1| \times (|U_1| + |U_2|))$, 这是引起时间复杂度高的一个重要原因; 又因该算法存储了差别矩阵, 求核遍历差别矩阵的时间复杂度为 $O(|U_1| \times (|U_1| + |U_2|))$, 故文献 [12] 使用差别矩阵是引起时间空间复杂度高的另一重要原因. 当决策表一致时, 有 $U_1 = U$, 文献 [12] 属性约简算法的时间和空间复杂度至少为 $O(|U|^2)$.

徐章艳^[13] 给出的属性约简算法 RedueBaseSig, 其时间复杂度为 $\max\{O(|C| \times |U|), O(|C|^2 \times |U|)\}$, 由于该算法未从求核出发, 在某些情况下, 获得并非属性的约简, 其中会包含冗余属性.

文献 [14] 对 RedueBaseSig 进行了改进, 提出一种基于分布计数的基数排序方法的等价类划分算法. 对决策表采用分布计数的基数排序, 按属性集 C 对决策表 S 排序, 该算法的时间复杂度也为 $O(|C| \times |U|)$, 空间复杂度为 $O(|U|)$.

3.1 改进的属性约简完备算法

改进的增量式属性约简算法 (improved incremental updating algorithm of attribute reduction for inserting, IIUAARI) 分 2 种情况: ① 新增对象 x , 若 x 与 U_2' 一致或 x 与 U_2' 中的某个对象不一致或 x 与 U_1 中的对象是 P 一致的, 则 P 是一个属性约简; ② 其他情况, 由增量式核算法 IOIUAC 计算得到的核开始重新计算属性约简.

IIUAARI 流程如图 5. 其中, $POS_C(D)$ 表示 D 的 C 正域, $POS_S(D)$ 表示 D 的 S 正域, $NEG_S(D)$ 表示 D 的 S 负域.

3.2 时间复杂度分析

运行 IIUAARI 算法时各中间步骤的时间复杂度分别为: 更新 $DMSC(C)$ 为 $O(|C| \times (|U_1| + |U_2|))$; 判断一致性为 $O(|C| \times (|U_1| + |U_2|))$; 由 $DMSC(C)$ 得到核为 $O(|C|)$; 判断 P 不一致性为 $O(U - POS_P(D)) < O(|U|)$; 计算 $POS_C(D)$ 、 $POS_S(D)$ 和 $NEG_S(D)$ 时, 3 个公式的复杂度都为 O

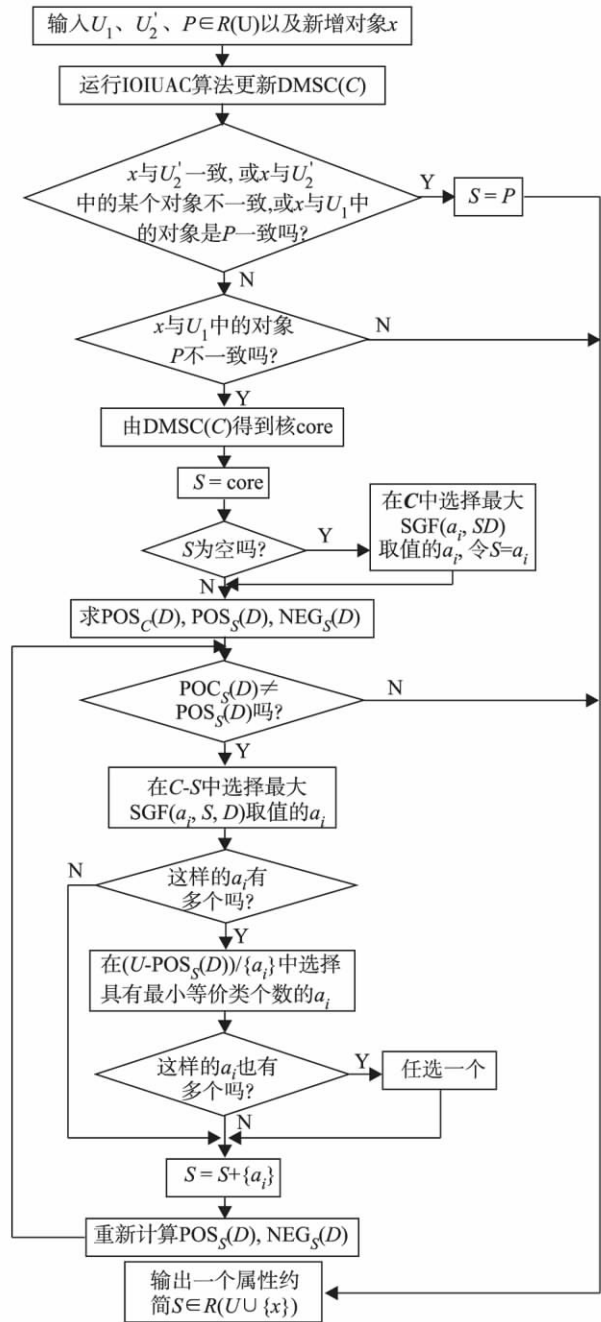


图 5 改进的增量式属性约简算法
Fig. 5 Improved incremental updating algorithm of attribute reduction

$(|C| \times |U|)$; $POS_C(D) \neq POS_S(D)$ 的循环处理为 $O(|C|^2 \times |U|)$. 综上可知, 算法 IIUAARI 总时间复杂度为 $O(|C|^2 \times |U|)$.

由于 IIUAARI 算法没有存储差别矩阵, 所以它的时间复杂度较文献 [12, 15] 的 $O(|C|^2 \times |U|^2)$ 有显著降低, 并与文献 [14] 的时空复杂度相当. 但

<http://journal.szu.edu.cn>

文献[14]的求核算法只能用于求一致性决策表的核;而IIUAARI算法充分利用已有的核,能减少计算量,适用于一致性、不一致性决策表的属性增量式约简;同时当新增对象 x , x 与 U_2 中的某个对象不一致或 x 与 U_1 中的对象是 P 一致时,IIUAARI的时间复杂度为 $O(|C| \times (|U_1| + |U_2|))$,能高效求得属性约简.

3.3 示例说明

表1是一张有5个对象和5个属性二值数据表,其中 $C = \{C_1, C_2, C_3, C_4\}$ 为条件属性集; D 为决策属性.

表1 二值数据表
Table 1 Two-value data table

| 对象 | 属性 | | | | D |
|-------|-------|-------|-------|-------|-----|
| | C_1 | C_2 | C_3 | C_4 | |
| x_1 | 1 | 0 | 1 | 0 | 2 |
| x_2 | 1 | 0 | 1 | 0 | 1 |
| x_3 | 1 | 1 | 1 | 0 | 3 |
| x_4 | 0 | 1 | 0 | 0 | 2 |
| x_5 | 0 | 1 | 1 | 1 | 2 |

由表1可知, $U_1 = \{x_3, x_4, x_5\}$, $U_2 = \{x_1, x_2\}$, $U_3 = \{x_1\}$,由文献[11]定义可得差别矩阵 M 为

$$M = \begin{matrix} & x_3 & x_4 & x_5 & x_1 \\ \begin{matrix} x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{pmatrix} \emptyset & \{C_1, C_3\} & \{C_1, C_4\} & \{C_2\} \\ \{C_1, C_3\} & \emptyset & \emptyset & \{C_1, C_2, C_3\} \\ \{C_1, C_4\} & \emptyset & \emptyset & \{C_1, C_2, C_4\} \end{pmatrix} \end{matrix}$$

由文献[11]可得,该表中核为 $\{C_2\}$.由文献[1]可知, $\{C_1, C_2\}$ 和 $\{C_2, C_3, C_4\}$ 为其2个属性约简.

为进一步说明算法IIUAARI,以下通过新增对象 x 说明插入对象后属性约简更新情况.

①若 $x = \{1, 0, 1, 0, 3\}$,则 x 与 x_1 不一致,由IIUAARI可知,核和属性约简不变.

②若 $x = \{1, 1, 1, 0, 1\}$,则 x 与 x_3 不一致;当IIUAARI通过IOIUAC更新DMSC(C),且 x 与 U_1 中的对象不一致时,由DMSC(C) = \emptyset 及 $S = \emptyset$,可得SGF(C_1, S, D) = 2/6, SGF(C_2, S, D) = 0, SGF(C_3, S, D) = 1/6, SGF(C_4, S, D) = 1/6;在 C 中选择 C_1 ,使得SGF(a_i, S, D)取值最大,则 $S = \{C_1\}$,由于POS $_C(D)$ = POS $_S(D)$,故 $S = \{C_1\}$ 为

一属性约简.

③若 $x = \{0, 1, 0, 0, 3\}$,则 x 与 x_4 不一致;当IIUAARI通过IOIUAC更新DMSC(C),且 x 与 U_1 中的对象不一致时,由DMSC(C) = $\{C_2\}$ 及 $S = \{C_2\}$,可得SGF(C_1, S, D) = 1/6, SGF(C_3, S, D) = 0, SGF(C_4, S, D) = 1/6; (U-POS $_S(D)$) / $\{C_1\}$ = 2,将 C_1 加入 S ,得 $S = \{C_1, C_2\}$;重新计算可得SGF(C_3, S, D) = 1/6, SGF(C_4, S, D) = 1/6; (U-POS $_S(D)$) / $\{C_1\}$ = 2,将 C_3 加入 S , $S = \{C_1, C_2, C_3\}$,此时POS $_C(D)$ = POS $_S(D)$,故 $S = \{C_1, C_2, C_3\}$ 为一属性约简.

④若 $x = \{1, 1, 0, 1, 3\}$,则 x 与原决策表中的对象 P 一致,由算法IIUAARI可知,核和属性约简不改变.

⑤若 $x = \{0, 1, 0, 1, 3\}$,则 x 与原决策表中的对象一致;IIUAARI通过IOIUAC更新DMSC(C);由DMSC(C) = $\{C_2, C_3, C_4\}$, $S = \{C_2, C_3, C_4\}$,POS $_C(D)$ = POS $_S(D)$,可得 $S = \{C_2, C_3, C_4\}$ 为一属性约简.

可见算法IIUAARI在各种情况下得到的约简与文献[12]算法相同,本文示例取自文献[12].

结 语

本研究基于已有增量式求核算法,提出不存储差别矩阵的改进增量式求核算法,由于差别矩阵是对称矩阵,本研究中的求核算法,通过只存储差别矩阵下三角部分,有效降低了空间复杂度.故能有效地处理大数据集.由于本文的IIUAARI算法没有存储差别矩阵,充分利用已有的核,显著减少计算量,降低了属性约简的复杂度,较已有的属性约简增量式算法^[12-18]更高效,而且它适用于一致性、不一致性决策表的属性增量式约简.故可以用于快速处理大数据集,能高效的求得属性约简.

下一步我们将继续完善属性约简算法,着重考虑对象删除情况下核增量式算法及属性约简的更新问题.

基金项目: 国家自然科学基金资助项目(50604012)

作者简介: 冯少荣(1964-),男(汉族),河北省南宫市人,厦门大学副教授、博士. E-mail: shaorong@xmu.edu.cn

引文: 冯少荣,张东.基于改进差别矩阵的增量式属性约简算法[J].深圳大学学报理工版,2012,29(5):405-411.

<http://journal.szu.edu.cn>

参考文献/References:

- [1] Pawlak Z. Rough sets [J]. International Journal of Information and Computer Science, 1982, 11(5): 341-356.
- [2] QIAN Yu-hua, LIANG Ji-ye, Pedrycz Witold, et al. Positive approximation: an accelerator for attribute reduction in rough set theory [J]. Artificial Intelligence, 2010, 174(9/10): 597-618.
- [3] HE Qiang, WU Cong-xin, CHEN De-gang, et al. Fuzzy rough set based attribute reduction for information systems with fuzzy decisions [J]. Knowledge-Based Systems, 2011, 24(5): 689-696.
- [4] QIAN Yu-hua, LIANG Ji-ye, Pedrycz Witold, et al. An efficient accelerator for attribute reduction from incomplete data in rough set framework [J]. Pattern Recognition, 2011, 44(8): 1658-1670.
- [5] XU Wei-hua, LI Yuan, LIAO Xiu-wu. Approaches to attribute reductions based on rough set and matrix computation in inconsistent ordered information systems [J]. Knowledge-Based Systems, 2012, 27: 78-91.
- [6] Wong S K M, Ziarko W. On optimal decision rules in decision tables [J]. Bulletin of Polish Academy of Sciences, 1985, 33(11/12): 663-676.
- [7] Hu X H, Cercone N. Mining knowledge rules from databases: a rough set approach [C]// Proceedings of the 12th International Conference on Data Engineering. New Orleans (USA): IEEE Press, 1996: 96-105.
- [8] Hu X H, Nick Cercone. Learning in relational databases: a rough set approach [J]. International Journal of Computational Intelligence, 1995, 11(2): 323-338.
- [9] WANG Guo-yin, YU Hong, YANG Da-chun. Decision table reduction based on conditional information entropy [J]. 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 [J]. 计算机学报, 2002, 25(7): 759-766.
- [10] YE Dong-yi, CHEN Zhao-jiong. A new discernibility matrix and the computation of a core [J]. 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 [J]. 电子学报, 2002, 30(7): 1086-1088.
- [11] YANG Ming. An incremental updating algorithm of the computation of a core based on the improved discernibility matrix [J]. Chinese Journal of Computers, 2006, 29(3): 407-413. (in Chinese)
杨明. 一种基于改进差别矩阵的核增量式更新算法 [J]. 计算机学报, 2006, 29(3): 407-413.
- [12] YANG Ming. An incremental updating algorithm for attribute reduction based on improved discernibility matrix [J]. Chinese Journal of Computers, 2007, 30(5): 815-822. (in Chinese)
杨明. 一种基于改进差别矩阵的属性约简增量式更新算法 [J]. 计算机学报, 2007, 30(5): 815-822.
- [13] XU Zhang-yan, YANG Bin-ru. Quickly attribution reduction algorithm based on decision table [J]. Journal of Chinese Computer Systems, 2006, 25(5): 858-861. (in Chinese)
徐章艳, 杨炳儒. 一个基于决策表的快速属性约简算法 [J]. 小型微型计算机系统, 2006, 25(5): 858-861.
- [14] GE Hao, LI Long-shu, YANG Chuan-jian. Improvement to quick attribution reduction algorithm [J]. Journal of Chinese Computer Systems, 2009, 30(2): 308-312. (in Chinese)
葛浩, 李龙澍, 杨传健. 改进的快速属性约简算法 [J]. 小型微型计算机系统, 2009, 30(2): 308-312.
- [15] LIU Yang, FENG Bo-qin, ZHOU Jiang-wei. Complete algorithm of increment for attribute reduction based on discernibility matrix [J]. Journal of Xi'an Jiaotong University, 2007, 41(2): 158-161. (in Chinese)
刘洋, 冯博琴, 周江卫. 基于差别矩阵的增量式属性约简完备算法 [J]. 西安交通大学学报, 2007, 41(2): 158-161.
- [16] QIAN Jin, YE Fei-Yue, LU Ping. An incremental attribute reduction algorithm in decision table [C]// Proceedings of the Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). Yantai (China): IEEE Press, 2010, 4: 1848-1852.
- [17] XU Yi-tian, WANG Lai-sheng, ZHANG Rui-yan. A dynamic attribute reduction algorithm based on 0-1 integer programming [J]. Knowledge-Based Systems, 2011, 24(8): 1341-1347.
- [18] ZHANG Jun-bo, LI Tian-rui, RUAN Da. Rough sets based incremental rule acquisition in set-valued information systems [J]. Autonomous Systems: Developments and Trends, 2012, 391: 135-146.

【中文责编: 英子; 英文责编: 雨辰】

<http://journal.szu.edu.cn>