基于标记可辨识矩阵的增量式属性约简算法

尹林子 [1] 阳春华 [2] 王晓丽 [2] 桂卫华 [2]

摘 要 针对现有增量式属性约简算法中存在的约简传承性差以及不完备现象, 提出基于标记可辨识矩阵的增量式属性约简算法. 本文首先定义了标记函数, 对样本之间的可辨识性进行分类, 并将之引入一个新的可辨识矩阵, 在新增样本时, 结合标记信息可以快速识别可辨识矩阵元素集的异动, 获得强传承性的约简超集, 在此基础上, 设计与标记可辨识矩阵匹配的必要矩阵, 用以快速判断并删除冗余属性, 确保约简的完备性. 理论分析以及实验测试表明, 本算法具有约简传承性强, 约简集完备等特点, 具有较强的实用性.

关键词 标记可辨识矩阵, 必要矩阵, 增量式约简, 约简传承性

引用格式 尹林子, 阳春华, 王晓丽, 桂卫华. 基于标记可辨识矩阵的增量式属性约简算法. 自动化学报, 2014, **40**(3): 397−404

DOI 10.3724/SP.J.1004.2014.00397

An Incremental Algorithm for Attribute Reduction Based on Labeled Discernibility Matrix

YIN Lin-Zi[1] YANG Chun-Hua[2] WANG Xiao-Li[2] GUI Wei-Hua[2]

Abstract In order to improve the inheritance rate of reducts and obtain the complete reducts, an incremental algorithm for attribute reduction based on labeled discernibility matrix is proposed. The label function is first defined to classify the discernibility relationships of all object pairs. The labeled discernibility matrix is then proposed to find out the changed elements and compute a supper set of reducts quickly when a new object is added. At the same time, a necessary matrix corresponding to the labeled discernibility matrix is presented. It is used to delete the redundant attributes for a complete reduct. Theoretical analysis and experimental results show that the reducts calculated by the proposed algorithm are complete and have the characteristic of high inheritance rate.

Key words Labeled discernibility matrix, necessary matrix, incremental reduction, inheritance rate of reduct

Citation Yin Lin-Zi, Yang Chun-Hua, Wang Xiao-Li, Gui Wei-Hua. An incremental algorithm for attribute reduction based on labeled discernibility matrix. *Acta Automatica Sinica*, 2014, **40**(3): 397−404

属性约简是粗糙集理论重要的应用之一[1−2], 也是知识获取的关键步骤, 根据应用的区别, 主要分为静态约简与动态约简, 前者针对固定的数据集进行知识获取, 后者则面临不断增加的新样本, 需及时修正约简集. 由于知识处于一个动态变化的过程, 增量式约简算法具有一个重要的区别于静态约简算法的评价指标: 约简集的传承性. 传承性指新约简集相对于旧约简集的继承程度, 是知识延续性的一种体现. 如果传承性强, 则新样本对规则集的影响小, 反之, 约简剧烈变化则可能导致原来的规则集需要全部更新[3], 因此, 在基于增量式约简的应用中 (如文献 [4−5]), 约简的传承性是非常重要的.

增量式约简吸引了许多研究者的兴趣. 文献 [6] 针对增量样本可能产生的不一致现象, 提出了基于正域的增量式属性约简算法. 文献 [7] 基于正域的决策值以及边界域分解决策表, 提出了分布式增量属性约简模型, 实现了全部约简的增量式计算. 为了增加动态特性, 提高计算速度, 文献 [8] 引入计数分类算法处理冗余以及不一致的样本, 降低计算复杂度. 文献 [9] 提出增量式核更新算法以及属性约简更新算法, 分类新增样本对约简的影响, 快速更新核属性集, 获取了完备的约简. 文献 [10] 提出不使用可辨识矩阵的增量式核更新算法以及约简算法, 具有较低的时间、空间复杂度. 在传承性方面, 文献 [9, 11] 均有目的地利用候选约简来辨识新样本以提高传承性, 但并没有给出具体的衡量指标, 而且, 在候选约简不足以胜任新数据集时, 传承性将难以保证.

收稿日期 2012-07-03 录用日期 2012-10-25
Manuscript received July 3, 2012; accepted October 25, 2012
国家自然科学基金 (61025015, 61273159, 61321003), 国家科技支撑计划 (2012BAF03B05) 资助
Supported by National Nature Science Foundation of China (61025015, 61273159, 61321003), Projects in the National Science Technology Pillar Program During the Twelfth Five-year Plan Period (2012BAF03B05)
本文责任编委 方海涛
Recommended by Associate Editor FANG Hai-Tao
1. 中南大学物理与电子学院 长沙 410083 2. 中南大学信息科学与工程学院 长沙 410083
1. School of Physics and Electronic, Central South University, Changsha 410083 2. School of Information Science and Engineering, Central South University, Changsha 410083

为提高增量式算法的约简传承性, 本文构建了新的标记可辨识矩阵, 通过对元素进行分类并加以标记, 快速提取变化的元素集, 并以此更新原有约简集, 既提高了约简的传承性, 又有效利用原有知识, 提高了计算速度, 在此基础上, 设计与标记矩阵匹配的必要矩阵, 应用于冗余属性检测, 保证约简集的完备性.

1 粗糙集概念

定义 1[12]**.** 一个决策表定义为一个四元组 $S = \langle U, At = C \cup D, V, f\rangle$, 其中, U 为非空有限样本集, C 为条件属性集, D 为决策属性集, V 为非空属性值集, $f : U \to V$ 为映射函数.

定义 2[13]**.** 设矩阵 $M = \{M(x, y)\}$ 为决策表对应的可辨识矩阵, 约简 R 必须满足以下条件:

1) 对 $\forall M(x, y)$, $M(x, y) \neq \emptyset \Rightarrow R \cap M(x, y) \neq \emptyset$;

2) 对 $\forall P \subset R$, $\exists M(x, y) \neq \emptyset \Rightarrow M(x, y) \cap P = \emptyset$;

如果属性集 R 满足条件 1), 但不满足或无法判断是否满足条件 2), 则称之为约简超集.

2 标记可辨识矩阵与约简传承率

考虑到新样本可能引发的不一致问题, 本文采用文献 [9] 的分类方法, 从决策表中提取两类样本: 精确样本集 U_1 (包含所有不重复的精确的样本) 以及粗糙样本集 U_2' (从每个粗糙粒子中选取一个样本构成的集合), 并对原始决策表进行预处理, 包括:

1) 删除重复的样本;

2) 删除 $U - U_1 - U_2'$ 中的样本 (即每个粗糙粒子只保留一个样本, 其余的删除).

针对预处理之后的决策表, 本文设计可辨识矩阵 $M_1 = \{m_{ij}\}$, 其元素定义如下:

$$m_{ij} = \begin{cases} \{a \in C : f(x_i, a) \neq f(x_j, a)\}, \\ \qquad x_i \in U_1, x_j \in (U_1 \cup U_2') \\ \emptyset, \qquad \text{其他} \end{cases}$$

该矩阵要求每一行对应的样本 x_i 为精确样本, 并且, 对相同决策值的对象对 (Object pair) 也进行辨识. 将可辨识矩阵 M_1 中的元素分别给予不同的标记, 标记函数定义如定义 3 所示.

定义 3. 对于决策表 S, 其可辨识矩阵 M_1 中非空元素的标记函数 $Label(\cdot)$ 定义为

$$Label(m_{ij}) = \begin{cases} 0, & d(x_i) = d(x_j) \text{ 且 } x_j \in U_1 \\ 1, & x_j \in U_2' \\ 2, & \text{其他} \end{cases}$$

定义 4. 将可辨识矩阵 M_1 中的非空元素利用标记函数 $Label()$ 进行标记即可获得决策表 S 的标记可辨识矩阵 (Labeled discernibility matrix), 记为 LM.

在标记可辨识矩阵中, 元素被分为三类: 0 标记具有相同决策值的精确样本之间的辨识关系; 1 标记精确样本与粗糙样本之间的辨识关系; 2 标记不同决策值的精确样本之间的辨识关系.

标记可辨识矩阵与 M_1 的大小一致, 均为 $|U_1 \cup U_2'| \times |U_1| \times |C|$. 它具有以下性质:

性质 1. 标记可辨识矩阵中所有偶数标记列构成了一个对称阵;

性质 2. 每一行对应于一个精确样本;

性质 3. 在基于正域的静态属性约简方法中, 标记为 0 的元素可以忽略;

性质 4. 标记为偶数的元素均会出现两次, 标记为 1 的元素只出现一次, 空集元素没有标记;

性质 5. 粗糙样本对应列的标记均为奇数, 精确样本对应列的标记均为偶数.

实例 1. 下面通过一个实例说明标记可辨识矩阵的构建效果. 考虑决策表 S 如表 1 所示.
其中, $x_1 \in U_2'$, $x_3, x_4, x_5 \in U_1, x_2$ 可以忽略, 表 1 对应的标记可辨识矩阵如下:

$$\begin{bmatrix} \{c_2c_4\}^1 & \emptyset & \{c_1c_3\}^2 & \{c_2c_3c_4\}^2 \\ \{c_1c_2c_3c_4\}^1 & \{c_1c_3\}^2 & \emptyset & \{c_1c_2c_4\}^0 \\ \{c_2c_3\}^1 & \{c_2c_3c_4\}^2 & \{c_1c_2c_4\}^0 & \emptyset \end{bmatrix}$$

前 3 行分别描绘了精确样本 x_3, x_4, x_5 的辨识关系, 第 1 列代表了粗糙样本 x_1 与其他精确样本的辨识关系, 第 2 ∼ 4 列构成了 1 个对称阵.

表 1 一个典型决策表

Table 1 A classical decision table

样本	c_1	c_2	c_3	c_4	d
x_1	1	0	1	0	2
x_2	1	0	1	0	1
x_3	1	1	1	1	3
x_4	0	1	0	1	2
x_5	1	2	0	0	2

基于静态约简算法[13], 可知本决策表包含三个约简 $\{c_1c_2\}, \{c_2c_3\}, \{c_3c_4\}$, 称之为候选约简, 对于增量式约简算法, 每次只需要一个候选约简.

定义 5. 设 R 为决策表 S 的候选约简, R' 为新增样本之后的新约简, 则单次增量式约简的传承率 (Inheritance rate, IR) 定义为

$$IR = \frac{|R \cap R'|}{\min(|R|, |R'|)}$$

假设进行了 k 次增量式约简, 则平均传承率

(Inheritance rate average, IRA) 定义为

$$IRA = \sum_{i=1}^{k} \frac{IR_i}{k}$$

在约简过程中, 传承率越高则约简集的变化越小, 相应地, 对规则集的影响也越小. 作为极致情况, 如果传承率为 0, 则新的约简集与原来的约简集完全不同, 此时, 规则集需要全部更新.

3　增量式属性约简算法

在实际应用中, 新样本的增加方式有多种可能性, 与文献 [14] 类似, 本文假设每次只增加一个样本, 且会多次增加. 因此, 本算法不仅需要计算新的属性约简, 也需要保存一些信息以简化下一次的增量式约简计算. 对于数据不一致现象, 采用基于正域的知识选择方法, 忽略粗糙样本之间的辨识性. 为保证算法的传承性、完备性以及持续可循环性, 算法运行前需要先分析原始决策表, 生成分析数据, 主要有标记可辨识矩阵 LM、候选约简 R 等. 具体的增量式约简算法则包括增量式超集 R' 计算与 LM 修正, 冗余属性检测等模块.

3.1　原始决策表分析

为适应增量式属性约简计算, 需要对最原始决策表 S 进行一系列处理, 主要包括:

1) 删除重复样本;

2) 删除 $U - U_1 - U_2'$ 中的样本. 由于标记可辨识矩阵能够识别粗糙样本 (性质 5), 因此, 每一个粗糙粒子只需要保留一个样本;

3) 根据决策表生成标记可辨识矩阵 LM, 并计算一个完备约简 R 作为候选约简.

初始约简计算属于静态约简计算过程, 有大量成熟的算法可以参考, 本算法要求 R 至少为一个约简超集, 至于 R 是否为最小约简则并不关注.

3.2　LM 修正与数据提取

考虑属性集 R 为决策表 S 的候选约简, 当增加一个新的样本 x 时, 可能会出现三种情况:

1) 样本 x 与 $x_i \in U_1$ 冲突, 即 $d(x) \neq d(x_i)$ 且 $\forall c \in C$, 有 $c(x) = c(x_i)$;

2) 样本 x 为新增样本, 即 $\forall x_i \in (U_1 \cup U_2')$, $\exists c \in C$, 有 $c(x) \neq c(x_i)$;

3) 其他情况.

对于情况 1), 样本 x_i 将不属于样本集 U_1, 转而属于样本集 U_2', 换而言之, 样本 x_i 将从某个决策等价类的正域移到边界域, 因此, x_i 与 U_2' 中样本 ($\forall x_j \in U_2'$) 的辨识关系将被忽略, 与同决策值精确样本 ($\forall x_k \in U_1, d(x_k) = d(x_i)$) 的辨识关系则需要重新纳入计算. 对于 LM 而言, 样本 x_i 对应的行与列中, 标记 1 的元素将不再需要辨识, 标记 0 的元素将需要纳入约简计算, 标记为 2 的元素仍需要计算, 但是标记需要修改为 1. 故 LM 的整体修正方案为: 删除 x_i 对应的行, 然后将 x_i 对应列中非空元素的标记置为 1. 与修改前相比, 增加的元素为 x_i 对应列中标记为 0 的元素, 减少的元素 (本文称之为失效元素) 为 x_i 对应行中标记大于 0 的元素.

对于情况 2), 新样本 x 将属于样本集 U_1, 此时, 标记可辨识矩阵需要增加新的一行以及新的一列, 没有失效元素.

情况 3) 中, x 为重复样本或者与 U_2' 集关于 C 一致的样本, 此时, R、LM 均不变.

上述分析表明, 当增加一个新的样本时, 矩阵中需要辨识的元素可能会增加或者减少, 利用新增元素集可以快速计算约简超集, 而失效元素集则与冗余属性检测有关. 在新增样本 x 的情况下, LM 修正以及增量约简计算数据: 新增元素集 (New element, NE) 与失效元素集 (Deletion element, DE) 提取算法如算法 1 所示.

算法 1. 增量式数据提取与 *LM* 修正算法

输入. 决策表 S, LM, 候选约简 R, 新增样本 x.

输出. 新增元素集 NE, 失效元素集 DE.

步骤 1. NE, DE 为空.

步骤 2. 新增样本 x 与决策表 S 中的样本比较.

For $i = 1$: $|U_1 + U_2'|$

比较样本 x 与样本 x_i, 获得对应的辨识元素 elm 并打上标记;

$NE(i) =$ elm;

If elm 为空, **Then** 判断冲突类型.

步骤 3. 修正 LM, 提取 NE 与 DE.

对于情况 1), $DE = x_i$ 对应行, 将 x_i 对应行置为空. $NE = x_i$ 对应列中标记为 0 的元素集合, 将 x_i 对应列中非空元素的标记改为 1; 对于情况 2), 将 NE 作为新的一行增加到 LM 中, 将 NE 中标记不为 1 的元素作为新的一列依次增加到 LM 中.

步骤 4. 结束.

算法 1 主要完成 LM 的修正, 提取 NE 用以计算约简超集, DE 则用以实现冗余属性的删除.

复杂度分析: 算法的时间复杂度主要集中于步骤 2, 为 $O(|U_1 \cup U_2'| \times |C|)$, 由于决策表 S 中不存在重复的样本, elm 为空的机会最多只出现一次, 因此, 可以在 elm 为空时直接退出循环以加快速度. 空间复杂度则主要为 LM 存储需求, 为 $O(|U_1 \cup U_2'| \times |U_1| \times |C|)$.

算法 1 表明, 在增加新样本时, 由于标记的作用, 并不需要重新计算整个矩阵, 只需要修正部分元

素集, 避免了对大部分样本对 ($|U_1 \cup U_2'| \times |U_1|$ 个样本对) 的二次辨识, 计算速度快.

3.3 超集计算

由于候选约简 R 可以辨识原始 LM 中所有标记大于 0 的元素, 因此, 新增样本引发标记可辨识矩阵改变时, 计算新增元素集 NE 即可获得约简超集.

考虑知识的传承性, 优先利用候选约简 R 辨识 NE, 并将其分解为两个部分, $NE = NE_1 \cup NE_2$, $NE_1 \cap NE_2 = \emptyset$, 且 $NE_1 = \{m(i,j) : m(i,j) \in NE \wedge m(i,j) \cap R \neq \emptyset\}$.

如果 NE_2 为空, 则 R 为新样本集的超集, 否则, 需要针对 NE_2 计算约简. 约简超集计算算法如下.

算法 2. 增量式约简超集计算

输入. 新增元素集 NE, 候选约简 R.

输出. 超集 R'.

步骤 1. **If** NE 为空, **Then** $R' = R$, 结束.

步骤 2. 将 NE 中与 R 相交的, 且标记大于 0 的元素删除, 获得 NE_2.

步骤 3. $Tmp = NE_2$; NR 为空.

While Tmp 不为空

计算核属性 CE; $NR = NR \cup CE$;

删除 Tmp 中包含核属性的元素;

选择 Tmp 中出现频率最大的属性 c, 将其从 Tmp 元素中删除.

步骤 4. $R' = R \cup NR$.

算法 2 只针对 NE 进行计算, 由于步骤 3 每次选取的属性均为核属性, 因此, NR 为 NE_2 的一个约简. 又由于 R 为原始 LM 的一个约简, 算法 2 的计算结果为新 LM 的一个约简超集. 其次, 由于 $R \subseteq R'$, 算法 2 的知识传承性比较强, 对规则集的影响比较小.

算法 2 复杂度分析: 空间需求主要来自 NE, 由算法 1 可知, NE 最大为 LM 的一行, 有 $|NE| < |U_1 \cup U_2|$, 故空间复杂度为 $\mathrm{O}(|U_1 \cup U_2'| \times |C|)$, 时间需求则主要集中于步骤 3, 计算核属性需要遍历 NE, 为 $\mathrm{O}(|NE| \times |C|)$, 因此, 最坏情况下接近 $\mathrm{O}(|U_1 \cup U_2'| \times |C|)$.

算法 2 表明, 基于 LM 提取的新增元素集, 可以实现约简超集的增量式计算, 避免对 LM 重新遍历, 计算速度快.

3.4 冗余属性检测

冗余属性检测是属性约简的一个重要组成部分, 用以保证算法的完备性.

定义 6. 设决策表 S 对应的标记可辨识矩阵为 LM, $R \subseteq C$, 如果 $\exists m_{ij} \cap R = \{c\}$ 且 $Label(m_{ij}) > 0$, 则称属性 c 相对于 R 是必要的 (Necessary), 记为 $c \in NEC(R)$.

定理 1. 考虑约简超集 R', 对于 $\forall c \in R'$, 如果 $c \notin NEC(R')$, 则属性 c 为一个候选的冗余属性.

证明. 因为 $c \notin NEC(R')$, 则对于任意标记大于 0 的元素 m_{ij}, 有 $m_{ij} \cap R' \neq \{c\}$, 即 $c \notin m_{ij} \cap R'$ 或者 $|m_{ij} \cap R'| > 1$, 前者意味着 m_{ij} 无法由 c 辨识, 后者意味着 m_{ij} 可以由 $R' - \{c\}$ 中的属性辨识, 因此, 属性集 $R - \{c\}$ 与属性集 R 具有相同的辨识能力, 故属性 c 可以视为冗余属性. □

在实际计算过程中, 超集 R' 中可能会出现多个候选的冗余属性, 此时, 只能将其中的一个作为真正的冗余属性. 这是因为一旦删除某一个冗余属性, 则 R' 会改变, 因此, 每次只能选择一个冗余属性.

定理 2. 对于约简超集 R', 如果对于 $\forall c \in R'$, 有 $c \in NEC(R')$, 则 R' 是决策表的一个约简.

证明. 由于 $c \in NEC(R')$, 则 $\exists m_{ij} \in LM$, 有 $Label(m_{ij}) > 0$ 且 $m_{ij} \cap R' = \{c\}$, 因此, 对于任意 $P \subseteq R' - \{c\}$, 有 $m_{ij} \neq \emptyset \wedge m_{ij} \cap P = \emptyset$, 故 P 不是一个约简, 由于所有的 c 均满足 $c \in NEC(R')$, 因此, R' 的任意子集均不是约简, 又由于 R' 为一个约简超集, 故 R' 是一个约简. □

定理 3. 设 R 为决策表的候选约简, NR 为 NE_2 的一个约简, 则 $\forall c \in NR = R' - R$, 有 $c \in NEC(R')$.

证明. 由于 NR 为 NE_2 的一个约简, 对于 $\forall c \in NR$, $\exists m_{ij} \in NE$, 有 $Label(m_{ij}) > 0$ 且 $m_{ij} \cap NR = \{c\}$, 由于 $NR \cap R = \emptyset$, 有 $m_{ij} \cap R' = \{c\}$, 故 $c \in NEC(R')$. □

定理 1 和定理 2 分别证明了冗余属性检测的判断条件与终止条件, 定理 3 指明超集中的部分非冗余属性, 因此, 可以循环以下步骤获取完备的约简: 1) 对 LM 进行遍历求解所有的候选冗余属性; 2) 如果不存在冗余属性则算法终止, 否则删除一个冗余属性. 该方法简单、直观, 但是计算速度不快, 不管有没有冗余属性, 均需要至少遍历一次 LM, 又由算法 1 可知, 新增样本只会对 LM 进行部分修改, 因此, 可以存储以前对 LM 的遍历结果, 只针对修正的元素进行必要性分析以加快计算速度. 为此, 提出了必要矩阵的概念以简化冗余属性检测过程, 定义如下:

定义 7. 设 LM 为决策表 S 的标记可辨识矩阵, $R \subseteq C$, 则 R 的必要矩阵 (Necessary matrix, NM) 记为 $NM(R) = \{nm_{ij}\}$, 其元素定义为

$$nm_{ij} = \begin{cases} k, & Label(m_{ij}) > 0 \wedge m_{ij} \cap R = \{c_k\} \\ 0, & \text{其他} \end{cases}$$

显然, 如果存在 $nm_{ij} = k$, 则属性 c_k 不是冗余元素; 必要矩阵与标记矩阵具有一一对应关系, 因此, 在 LM 修正时可以实现 NM 的同步修正. 其

次, 由于 LM 的元素为包含了 $|C|$ 个属性的属性集, NM 的元素为单个数字 (0 到 $|C|$), 因此, 遍历 NM 的耗时为遍历 LM 耗时的 $1/|C|$. 基于必要矩阵, 提出了冗余属性检测算法如下 (需要在初始分析中计算初始 NM).

算法 3. 冗余属性检测与 NM 修正

输入. 超集 R', NE, NR, DE, 初始 NM.

输出. 完备约简.

步骤 1. 利用 NE, DE 以及 NR 修正 NM.

对 NE 的元素进行必要性分析, 并修正 NM; 将 DE 对应的 NM 元素删除; 利用 NR 修正 NM 中的非零元素.

步骤 2. 遍历 NM, 获取候选冗余集 TR.

$$TR = \{c_k \in R' : \forall nm_{ij} \in NM, nm_{ij} \neq k\}$$

步骤 3. 如果 TR 为空, 结束. 否则, 从 TR 选择频率最大的属性作为冗余属性, 将其从 R' 中删除.

步骤 4. 对于 LM 中所有包含冗余属性的元素重新计算其必要性, 修正 NM; 返回步骤 2.

复杂度分析: 步骤 1 与 2 需遍历 NM, 时间复杂度为 $\mathrm{O}(|U_1| \times (|U_1| + |U_2'|))$, 步骤 4 的复杂度与 LM 中包含冗余属性的元素数目有关, 最坏情况下为 $\mathrm{O}(|U_1| \times |U_1 \cup U_2'| \times |C|)$, 因此, 当 TR 集为空时, 算法 3 的时间复杂度为 $\mathrm{O}(|U_1| \times (|U_1| + |U_2'|))$, 如果 TR 中存在多个候选冗余属性, 则最坏情况下需要循环 $|TR|$ 次, 总时间复杂度为 $\mathrm{O}(|U_1| \times (|U_1| + |U_2'|) \times |C| \times |TR|)$.

完备性分析: 由定理 2 可知, 当算法 3 达到 TR 为空的终止条件时, 超集 R' 即为约简, 因此, 算法 3 的输出是一个完备约简.

3.5　基于标记可辨识矩阵的增量式约简算法

完整的增量式约简算法如图 1 所示. 该算法以 LM、候选约简 R 以及必要矩阵 NM 为增量式计算的核心, 除第一次由原始决策表提取之外, 其他情况全部由增量式算法进行修正. 当有新样本 x 出现时, 首先利用算法 1 修正 LM, 快速提取 NE, DE 集; 算法 2 则利用 NE 计算约简超集 R'; 算法 3 删除冗余属性, 输出完备的约简, 并完成对 NM 的修正, 以等待下次的新样本, 使程序可以无限循环执行.

算法复杂度分析: 算法 1 与算法 2 的时间复杂度均为 $\mathrm{O}(|U_1 \cup U_2'| \times |C|)$, 算法 3 则取决于 TR 集的大小, 介于 $\mathrm{O}(|U_1| \times (|U_1 \cup U_2'|))$ 与 $\mathrm{O}(|U_1| \times (|U_1 \cup U_2'|) \times |C| \times |TR|)$ 之间. 可见, 本算法主要耗时集中于冗余属性检测. 空间存储需求主要来自于标记可辨识矩阵 LM, 为 $\mathrm{O}(|U_1| \times (|U_1 \cup U_2'|) \times |C|)$.

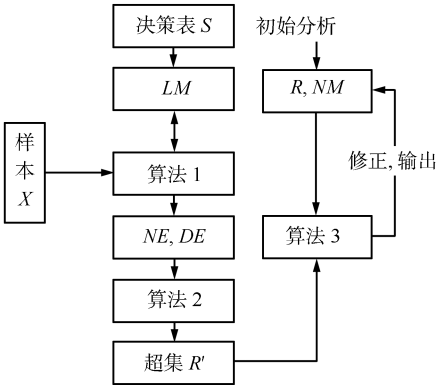


图 1　基于标记可辨识矩阵的增量式约简算法

Fig. 1　Incremental reduction algorithm based on LM

与同类型算法比较, 文献 [10] 只能求解超集, 其时间复杂度为 $\mathrm{O}(|C|^2 \times |U|)$, 而本文超集计算 (算法 1 与算法 2) 的复杂度为 $\mathrm{O}(|U_1 \cup U_2'| \times |C|)$, 明显低于前者. 文献 [9] 的复杂度比本算法略高, 文献 [15] 的复杂度为 $\mathrm{O}(|U|^2|C|^2)$, 远高于本文算法的最大耗时.

本算法的特点主要体现在两个方面: 1) 设计了合理的 LM, 快速提取 NE, 并只针对 NE 计算约简, 避免了对绝大部分样本对的二次辨识, 快速获得约简超集; 2) 设计了必要矩阵 NM, 快速判断是否存在冗余属性. 本算法可以输出完备约简 R, 约简传承性好且平均每次的计算复杂度低, 比较适合大量次数的增量式计算.

实例 2. 对于表 1 所示决策表, 选择不同的候选约简以及新样本进行增量式计算.

假设候选约简为$\{c_2c_3\}$, 新增样本 x_6 为 01011, 则 x_6 与 x_4 冲突. LM 修正步骤主要包括: 1) 删除 x_4 对应的行 (第 2 行), 2) 将 x_4 对应列 (第 3 列) 中所有非空元素的标记变为 1. 此时, LM 为

$$\begin{bmatrix} \{c_2c_4\}^1 & \emptyset & \{c_1c_3\}^1 & \{c_2c_3c_4\}^2 \\ \{c_2c_3\}^1 & \{c_2c_3c_4\}^2 & \{c_1c_2c_4\}^1 & \emptyset \end{bmatrix}$$

新增样本集 $NE = \{\{c_1c_2c_4\}\}$, 用候选约简进行辨识, 可知 NR 为空. 基于算法 3 检测冗余属性, NM 修正为 $\begin{bmatrix} 2 & 0 & 3 & 0 \\ 0 & 0 & 2 & 0 \end{bmatrix}$, TR 为空, 故$\{c_2c_3\}$为完备约简, 算法结束, 等待下一次的新样本. 与之对比, 文献 [10] 约简结果为$\{c_1c_2\}$, 文献 [9] 计算结果为$\{c_2c_3\}$.

假设新样本 x_6 为 00111, 候选约简$\{c_3c_4\}$, 则 x_6 满足情况 2), 此时 LM 修正为

$$\begin{bmatrix} \{c_2c_4\}^1 & \emptyset & \{c_1c_3\}^2 & \{c_2c_3c_4\}^2 & \{c_1c_2\}^2 \\ \{c_1c_2c_3c_4\}^1 & \{c_1c_3\}^2 & \emptyset & \{c_1c_2c_4\}^0 & \{c_2c_3\}^2 \\ \{c_2c_3\}^1 & \{c_2c_3c_4\}^2 & \{c_1c_2c_4\}^0 & \emptyset & \{c_1c_2c_3c_4\}^2 \\ \{c_1c_4\}^1 & \{c_1c_2\}^2 & \{c_2c_3\}^2 & \{c_1c_2c_3c_4\}^2 & \emptyset \end{bmatrix}$$

$NE = \{\{c_1c_4\}, \{c_1c_2\}, \{c_2c_3\}, \{c_1c_2c_3c_4\}\}$, 利用候

选约简进行辨识, 获得 $NE_2 = \{\{c_1c_2\}\}$, 由于两个属性的频率一致, 删除排在后面的 c_2, 获得 $NR = \{c_1\}$. 算法 2 输出 $\{c_1c_3c_4\}$. 算法 3, 修正必要矩阵为 [4 0 0 0 1; 0 0 0 0 3; 3 0 0 0 0; 0 1 3 0 0], TR 为空, 输出完备约简 $\{c_1c_3c_4\}$, 传承度 100 %. 与之对比, 文献 [10] 为 $\{c_1c_2c_3\}$, 不完备, 文献 [9] 为 $\{c_1c_2\}$, 传承度为 0.

4 实验验证

4.1 对比测试

本文算法与文献 [9−10] 算法进行测试比较, 检查算法完备性与传承性方面的差别. 数据集如表 1 所示, 针对不同的候选约简集, 单次增量式对比测试结果如表 2 所示.

表 2 传承性对比测试

Table 2 Comparison of inheritance rate

	候选约简 c_2c_3			候选约简 c_3c_4		
新样本	算法 A	算法 B	算法 C	算法 A	算法 B	算法 C
11111	c_3	c_3	c_3	c_3	c_3	c_3
00111	c_1c_2	c_1c_2	$c_1c_2c_3$*	$c_1c_3c_4$	c_1c_2	$c_1c_2c_3$*
01011	c_2c_3	c_2c_3	c_1c_2	c_3c_4	c_3c_4	c_1c_2
12001	c_2c_3	c_2c_3	c_1c_4	c_3c_4	c_3c_4	c_1c_4

表 2 中, "*" 表示该约简不完备, 算法 A 为本文算法, 算法 B 为文献 [9] 算法, 算法 C 为文献 [10] 算法. 显然, 本文算法不仅完备, 传承率也全面优于对比算法. 文献 [9] 强调利用候选约简, 如果候选约简能够胜任, 则传承性比较好, 否则, 则需要重新从核开始计算. 文献 [10] 则出现了不完备现象.

4.2 增量性能测试

选取 UCI 标准数据集进行测试. 首先以 wdbc 数据集为例, 说明本算法的性能. 由于数据包含实数数据, 采用了均匀离散化操作, 为保持一定的样本冲突率, 采取 4 段均匀离散化操作, 即每个属性的属性值均匀分为 4 类. 本实验将前 40 个样本作为原始数据集, 然后实现了 529 次增量式操作, 新样本冲突类型如图 2 所示.

图 2 中, 490 个为新样本 (Type = 0), 与精确样本冲突了 27 次 (Type = 1), 与粗糙样本冲突了 12 次 (Type = 2). 因此, 在总计 529 个新样本中, 需要更新约简的情况出现了 517 次.

传承率以及约简数目变化如图 3. 该示例中, 上面的线为传承率变化曲线, 传承率变化范围为 [0.6, 1], 平均传承率 (IRA) 为 99.65 %. 下面的曲线描绘了约简数目的变化趋势, 考虑到该数据集 $|C| = 30$, 最小时约简集只包含 3 个属性, 最多包含了 18 个属性. 约简数目变化表明, 随着样本的不断增加, 知识越来越丰富, 约简集呈增加趋势.

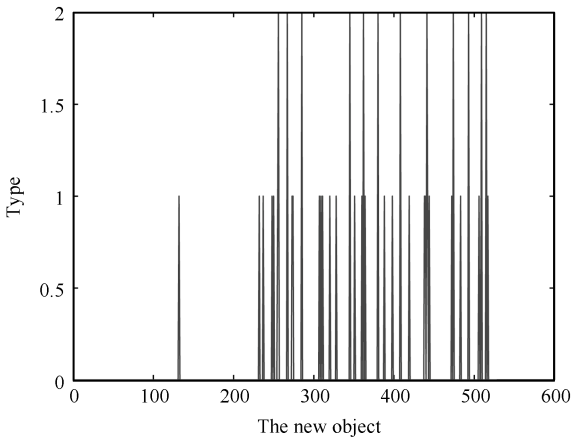


图 2 wdbc 数据集新样本冲突类型

Fig. 2 Type of new object in wdbc data set

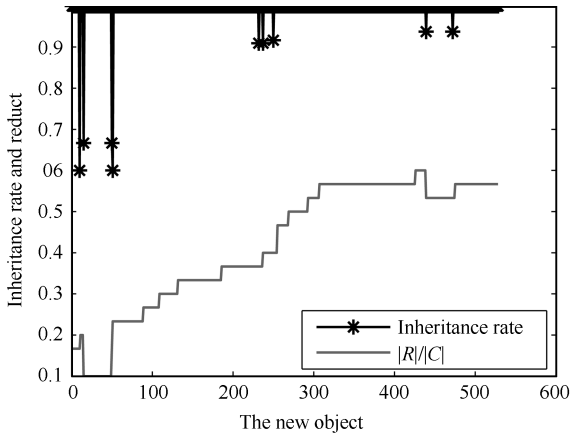


图 3 wdbc 数据集传承率与约简变化曲线图

Fig. 3 Inheritance rate and reducts of wdbc data set

算法复杂度分析: 在 549 次增量式计算中, TR 值有四次为 2, 一次为 3, 一次为 4, 换而言之, 需要删除冗余属性的情况只出现了 6 次该现象主要是因为本算法在计算 NR 时采取了删除频率最大属性的策略, 使 NR 中的属性比较多, 约简集的知识比较丰富, 增强了对于新样本的适应度. 该实例表明, 由于冗余属性计算次数比较少, 因此, 算法的平均实际耗时约为 $O(|U_1| \times (|U_1| + |U_2'|))$, 以 1.86 GB dual CPU, 1 GB 内存的笔记本电脑为例, 计算耗时约为 110 秒, 平均每一次增量式计算的耗时约为 0.198 秒.

同时, 本实验采用了多个 UCI 标准数据集进行测试, 每个数据集的前 40 个数据作为原始数据集, 其它的数据全部作为增量数据, 本文算法的测试结果如表 3 所示, 对比算法的测试结果见表 4.

其中, num 为增量约简次数, IRA 为平均传承

率, TRk 为 TR 大于 0 的次数, t 为平均每次增量式计算的耗时. 测试结果表明:

1) 本文算法的约简传承性非常好, 平均传承率 99 % 以上, 全面高于对比算法, 对规则集影响小.

2) 本文算法的计算速度远高于对比算法. 由于标记可辨识矩阵以及必要矩阵的原因, 本文算法避免了大部分样本的重复辨识, 而对比算法在候选约简无法胜任的时候则需要重新辨识, 因此, 本文算法在多次增量式计算时速度优势明显.

3) 本文算法单次增量式计算的平均耗时比较小, 但区别比较大. 这是因为 $|U_1|$ 与 $|U_2'|$ 与具体数据有关, 随着计算次数不断增加, LM 呈增大趋势, 会引发单次计算耗时不断增加, 因此, 增量次数越多, 平均单次耗时也越大.

4) 表 3 中 TR 大于 0 的次数相比于增量次数约为 1 ∼ 2 % 左右, 基本可以忽略不计, 故本文算法实际运行复杂度约为 $\mathrm{O}(|U_1| \times (|U_1| + |U_2'|))$, 计算速度快.

表 3　本文算法测试结果

Table 3　Experiment results of the proposed algorithm

Data set	$\|C\|$	num	IRA	TRk	t (s)
Wdbc	30	529	99.65	6	0.198
Wpbc	33	158	99.28	3	0.0409
Glass	9	174	99.83	2	0.0021
Iono	34	311	99.67	3	0.102
Wine	13	138	99.64	2	0.0174

表 4　对比算法测试结果

Table 4　Experiment results of two refreece algorithms

			算法 B		算法 C	
Data set	$\|C\|$	num	IRA	t (s)	IRA	t (s)
Wdbc	30	529	99.29	1.092	99.38	0.384
Wpbc	33	158	98.85	0.105	97.28	0.0968
Glass	9	174	99.42	0.009	99.42	0.005
Iono	34	311	99.49	0.31	97.88	0.197
Wine	13	138	95.69	0.0725	96.61	0.043

5　结论

在实际数据处理过程中, 约简只是一个中间结果, 规则集才是常见的输出, 因此, 增量式计算必须考虑约简对规则集的影响, 本文提出的传承率是一个重要的增量式约简性能衡量指标.

为提高传承率, 本文算法优先使用候选约简辨识新增加的元素集, 对于无法辨识的元素集则避免使用辨识能力强的属性, 扩大约简中属性数目, 降低 TR 大于 0 的概率, 有利于提高传承率. 本文提出的 LM 以及 NM, 实现了对元素集变动的快速记录, 避免了对样本的重复辨识, 提高了计算速度, 适合于大量次数的循环计算.

实验验证表明, 本文算法具有约简传承率高、完备以及实际运行速度快等特点, 具有良好的实用性.

References

1 Xiao Di, Hu Shou-Song. Real rough set theory and attribute reduction. *Acta Automatica Sinica*, 2007, **33**(3): 253−258
(肖迪, 胡寿松. 实域粗糙集理论及属性约简. 自动化学报, 2007, **33**(3): 253−258)

2 Yin Lin-Zi, Yang Chun-Hua, Gui Wei-Hua, Li Yong-Gang. Hierarchical reduction of rules. *CAAI Transactions on Intelligent Systems*, 2008, **3**(6): 492−497
(尹林子, 阳春华, 桂卫华, 李勇刚. 规则分层约简算法. 智能系统学报, 2008, **3**(6): 492−497)

3 Fan Y N, Tseng T L, Chern C C, Huang C C. Rule induction based on an incremental rough set. *Expert Systems with Applications*, 2009, **36**(9): 11439−11450

4 Fan Y N, Chern C C. An agent model for incremental rough set-based rule induction in customer relationship management. *Hybrid Artificial Intelligent Systems*, 2012, **7208**: 1−12

5 Zhang J B, Li T R, Ruan D. Rough sets based incremental rule acquisition in set-valued information systems. *Autonomous Systems: Developments and Trends*, 2012, **391**: 135−146

6 Hu F, Wang G Y, Huang H, Wu Y. Incremental attribute reduction based on elementary sets. In: Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. Regina, Canada: Springer-Verlag, 2005. 185−193

7 Hu Feng, Dai Jin, Wang Guo-Yin. Incremental algorithms for attribute reduction in decision table. *Control and Decision*, 2007, **22**(3): 268−272, 277
(胡峰, 代劲, 王国胤. 一种决策表增量属性约简算法. 控制与决策, 2007, **22**(3): 268−272, 277)

8 Qian J, Ye F Y, Lv P. An incremental attribute reduction algorithm in decision table. In: Proceedings of the 7th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). Yantai, China: IEEE, 2010, **4**: 1848−1852

9 Yang Ming. An incremental updating algorithm for attribute reduction based on improved discernibility matrix. *Chinese Journal of Computers*, 2007, **30**(5): 815−822
(杨明. 一种基于改进差别矩阵的属性约简增量式更新算法. 计算机学报, 2007, **30**(5): 815−822)

10 Feng Shao-Rong, Zhang Dong-Zhan. Effective increment algorithm for attribute reduction. *Control and Decision*, 2011, **26**(4): 495−500
(冯少荣, 张东站. 一种高效的增量式属性约简算法. 控制与决策, 2011, **26**(4): 495−500)

11 Xu Y T, Wang L S, Zhang R Y. A dynamic attribute reduction algorithm based on 0-1 integer programming. *Knowledge-Based Systems*, 2011, **24**(8): 1341−1347

12 Jiang Yun-Liang, Yang Zhang-Xian, Liu Yong. Quick distribution reduction algorithm in inconsistent information system. *Acta Automatica Sinica*, 2012, **38**(3): 382−388
(蒋云良, 杨章显, 刘勇. 不协调信息系统快速属性分布约简方法. 自动化学报, 2012, **38**(3): 382−388)

13 Yao Y Y, Zhao Y. Discernibility matrix simplification for constructing attribute reducts. *Information Sciences*, 2009, **179**(7): 867−882

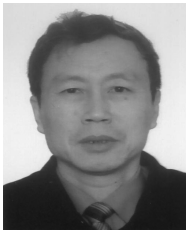
14 Zhang C S, Ruan J. An efficient incremental updating algorithm for knowledge reduction in information system. *Electronics and Signal Processing*, 2011, **97**: 263−271

15 Liu Yang, Feng Bo-Qin, Zhou Jiang-Wei. Complete algorithm of increment for attribute reduction based on discernibility matrix. *Journal of Xi′an Jiaotong University*, 2007, **41**(2): 158−161, 208
(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. 西安交通大学学报, 2007, **41**(2): 158−161, 208)

尹林子 中南大学博士研究生. 主要研究方向为智能数据处理, 粗糙集理论与应用. E-mail: nihaoylz@126.com
(**YIN Lin-Zi** Ph. D. candidate at Central South University. His research interest covers intelligent information processing and rough set theory.)

阳春华 中南大学教授. 主要研究方向为复杂工业过程建模与优化控制, 智能自动化系统. 本文通信作者.
E-mail: ychh@csu.edu.cn
(**YANG Chun-Hua** Professor at Central South University. Her research interest covers modeling and optimal control of industrial process, intelligent control systems, and fault-tolerant computing of real-time systems. Corresponding author of this paper.)

王晓丽 中南大学讲师. 主要研究方向为复杂工业过程建模, 智能数据处理.
E-mail: xlwang@csu.edu.cn
(**WANG Xiao-Li** Lecturer at Central South University. Her research interest covers modeling of complex industrial process and intelligent information processing.)



桂卫华 中南大学教授. 主要研究方向为复杂工业过程建模与优化控制, 分散鲁棒控制及故障诊断.
E-mail: gwh@csu.edu.cn
(**GUI Wei-Hua** Professor at Central South University. His research interest covers modeling and optimal control of complex industrial process, distributed robust control, and fault diagnoses.)