改进的基于简化二进制分辨矩阵的属性约简方法

王亚琦 范年柏
(湖南大学信息科学与工程学院 长沙 410082)

摘 要 在基于二进制分辨矩阵的属性约简方法中,删除法即从属性全集中依次删除冗余属性,直至剩余的属性集是一个最小约简。针对传统的基于二进制分辨矩阵的删除法效率较低且得不到最小约简的问题,提出一种改进的二进制分辨矩阵属性约简方法。首先对决策表进行简化,然后给出一种改进的简化二进制分辨矩阵方法;其次通过一个新的属性约简度量方法一次性删除多个属性,并从理论上分析了该方法的可行性;最后通过实验证明了得到的约简结果是最小约简。

关键词 粗糙集,二进制分辨矩阵,属性约简,决策系统
中图法分类号 TP181 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.6.044

Improved Algorithms for Attribute Reduction Based on Simple Binary Discernibility Matrix

WANG Ya-qi FAN Nian-bai
(School of Information Science and Engineering, Hunan University, Changsha 410082, China)

Abstract In the algorithms for attribute reduction based on simple binary discernibility matrix, elimination means that redundant attributes are excluded from reduct sets one by one until the last is a minimum reduct. The traditional elimination based on simple binary discernibility has the following shortcomings: lower efficiency and being not able to get the optimal solution. In view of those problems, an improved algorithm for attribute reduction based on simple binary discernibility matrix was presented. Firstly, the decision table was simplified. Secondly, an improved algorithm to simplify binary discernibility matrix was proposed. Lastly, for attribute reduction, we presented a new measure which can delete more than one redundant attributes and proved the feasibility of the measure. Moreover, the experiment results prove the correctness of the method.

Keywords Rough set, Binary discernibility matrix, Attribute reduction, Decision system

1 引言

20世纪80年代初期波兰数学家Pawlak提出了粗糙集[1]的概念,其被广泛运用于处理不确定性、不一致和不完全信息系统。其中,属性约简[2]是粗糙集理论模型中研究的核心之一。目前已有的属性约简方法有基于正区域的属性约简方法[3]、基于信息熵的约简方法[4]以及基于分辨矩阵的约简方法[5]。

Skowron[5]在不可分辨关系的基础上,提出了分辨矩阵的概念,其中矩阵中的每个元素是所有能够区分实例对象对的属性的集合,并在此基础上,提出了基于分辨矩阵的属性约简方法,即通过化简分辨函数建立的析取逻辑表达式,得到属性约简结果。因其在化简逻辑表达式上的计算复杂度非常大,Hu、Cereone N[6]针对Skowron的基于分辨矩阵的属性约简方法进行了改进,利用分辨矩阵先求核属性,再进行约简。Yamaguchi[7]基于分辨矩阵和条件属性的频率,提出了一种新的属性依赖度模型,该模型在不一致决策表中能更有效地计算属性重要度。Felix[8]等人在1999年提出了一种新的分辨矩阵,其矩阵元素只由0和1组成,称为二进制分辨矩阵。

支天云[9]在文献[8]的基础上对二进制分辨矩阵进行了各种变换,并将包含1的个数最多的列对应的属性归入约简。李龙澍[10]在简化二进制分辨矩阵的基础上,每次选择列方向包含1的个数最多的属性归入约简,所以其本质与文献[9]的度量方式相同。陈宸[11]基于二进制分辨矩阵,以矩阵的行与列两个方向的特征作为度量属性重要性的依据。徐章艳、任倩[12,13]针对不一致决策表,提出了基于二进制分辨矩阵的属性约简方法。以上方法,在对属性约简的过程中均是根据某种度量重要度的函数,以空集为初始约简集或以核为初始点,使用某种启发式信息作为衡量属性重要度的标准,逐次选择属性重要度较大的条件属性添加到约简集中。

蒙祖强[14]在简化二进制分辨矩阵的基础上进行矩阵变换,并每次选择一个列方向1的个数最少的非当前核属性进行删除。类似地,桂现才[15]在二进制分辨矩阵的基础上,每次选择任意一个非核属性,并将该属性对应的列上值1改为0。杨传健[16]针对蒙祖强的方法,将部分数据存于外存中,降低了其空间复杂度,但由于内外存交互的时间开销,其算法的

到稿日期:2014-06-17 返修日期:2014-10-08
王亚琦(1990-),女,硕士生,主要研究方向为粗糙集、数据挖掘,E-mail:397953969@qq.com;范年柏(1962-),男,博士,副教授,主要研究方向为粗糙集、模糊集、数据挖掘。

时间效率并未提高。因此，以上算法在做属性删除时效率并不高，且未能得到最小属性约简。

为了解决传统的二进制分辨矩阵删除法效率较低且得不到最小属性约简的问题，本文提出了一种改进的基于二进制分辨矩阵的属性约简方法，该方法首先运用蒙祖强[14]的方法对决策表进行简化，生成二进制分辨矩阵；然后指出蒙祖强[14]、李龙澍[10]在对二进制分辨矩阵简化过程中的可完善之处，并作出改进；其次提出一个新的属性约简度量方法，一次性将矩阵中多个列的1置为0，此算法通过一种新的度量属性约简重要度的方式，能够有效地提高删除法效率，并得到最小属性约简。

2 粗糙集理论

定义 1[3] 元组 $S=\langle U,C\cup D,V,f\rangle$ 是一个决策表，其中 U 是论域；C、D 分别是条件属性集和决策属性集，$V=\bigcup_{a\in(C\cup D)}Va$，$Va$ 表示属性 a 的值域；$f:U\times(C\cup D)\to V$ 是信息函数，若 $u_i\in U,a\in(C\cup D)$，记 $f(u_i,a)=a(u_i)$。

定义 2[3] 给定 $S=\langle U,C\cup D,V,f\rangle$ 和 $P\subseteq C\cup D$，P 在 U 上的不可分辨关系定义为 $IND(P)=\{(x,y)\mid(x,y)\in U\times U\wedge(\forall b\in P,b(x)=b(y))\}$。

定义 3[12] 在决策表 $S=\langle U,C\cup D,V,f\rangle$ 中，记 $U/C=\{[u_1']c,[u_2']c,\cdots,[u_{n'}']c\}$，记 $U'=\{u_1',u_2',\cdots,u_{n'}'\}$，$S'=(U',C\cup D,V,f)$ 为简化的决策表。

定义 4[8] 设决策表 $S=\langle U,C\cup D,V,f\rangle$，其中 $U=\{u_1,u_2,\cdots,u_n\}$，$C=\{c_1,c_2,\cdots,c_m\}$，$D=\{d\}$，则决策表 S 的二进制分辨矩阵 $BM=(BM(i,j,c_k))_{n(n-1)/2\times m}$。

$$BM(i,j,c_k)=\begin{cases}1, & c_k(u_i)\neq c_k(u_j)\wedge d(u_i)\neq d(u_j)\\ 0, & \text{其他}\end{cases}$$

其中，$BM((i,j),c_k)$ 表示决策表的第 i 和第 j 个样本是否能够通过第 k 个条件属性区分出，若能区分且这两个样本的决策属性值不一致，则 $BM(i,j,c_k)=1$，否则 $BM(i,j,c_k)=0$。

定义 5 $S'=(U',C\cup D,V,f)$ 为简化的决策表，$U'=\{u_1',u_2',\cdots,u_{n'}'\}$，$C=\{c_1,c_2,\cdots,c_m\}$，$D=\{d\}$，其二进制分辨矩阵 $BM'=(BM'(i,j,c_k))_{n'(n'-1)/2\times m}$。

$$BM'(i,j,c_k)=\begin{cases}1, & c_k(u_i')\neq c_k(u_j')\wedge d(u_i')\neq d(u_j')\\ 0, & \text{其他}\end{cases}$$

定义 6[14] 在二进制分辨矩阵 BM 中，若行 (s_i,s_j) 中1的个数小于行 (s_i',s_j') 中1的个数，且行 (s_i,s_j) 中列的值为1时，行 (s_i',s_j') 在该列也为1，则称行 (s_i',s_j') 逻辑包含行 (s_i,s_j)，若行 (s_i',s_j') 逻辑包含行 (s_i,s_j)，则删除行 (s_i',s_j')。这种不含逻辑包含行的二进制分辨矩阵称为简化二进制分辨矩阵。

定义 7 在决策表 $S=\langle U,C\cup D,V,f\rangle$ 中，$C=\{c_1,c_2,\cdots,c_m\}$，$P=\{c_p,c_q,\cdots,c_k\}$，P 是 S 的约简，$R_p=(\lambda_1,\lambda_2,\cdots,\lambda_m)$，$\lambda_i=\begin{cases}1, & c_i\in P\\ 0, & c_i\notin P\end{cases}$，$0<i\leqslant m$，其中 R_p 为 m 维向量，称 R_p 是 P 的二进制向量表示。例如，$C=\{c_1,c_2,c_3,c_4,c_5\}$，$P=\{c_1,c_3,c_5\}$，则 $R_p=(1,0,1,0,1)$。

定理 1[14] 对于一致决策系统而言，$S'=(U',C\cup D,V,f)$ 为简化的决策表，$U'=\{u_1',u_2',\cdots u_{n'}'\}$，$BM'_{n'\times m}$ 为 S' 的二进制分辨矩阵，若存在唯一的 $k\in\{1,2,\cdots,m\}$，使得 $BM'(i,j,c_k)=1$，而对于 $\forall a\in\{1,2,\cdots,k-1,k+1,\cdots,m\}$，均有 $BM'(i,j,c_a)=0$，则 k 所在的列为 S' 的核属性。

3 属性约简方法的分析

基于简化的决策表，文献[10,14]中给出了一个简化二进制分辨矩阵算法，但该算法仍有可改进之处，下面将简述该算法，然后分析说明该算法的可完善之处。

算法 1 简化二进制分辨矩阵算法[10,14]

输入：决策系统〈U,C∪D〉；

输出：简化的二进制分辨矩阵 M″；

1)求解 U/C；

2)在 U/C 的每个等价类中抽取一个对象，组成简化决策表〈U′,C∪D〉，$U'=\{s_1',s_2',\cdots,s_p'\}$；

3)令 M″=空集；

4)For i=1 to |U′|−1 do

Begin

For j=i to |U′|−1 do

Begin

If M″ 中存在逻辑包含 (s_i',s_j') 的行，Then 将包含 (s_i',s_j') 的行全部删除，然后将 (s_i',s_j') 添加到 M″ 中；

else if (s_i',s_j') 不包含 M″ 中任意的行，Then 将 (s_i',s_j') 添加到 M″ 中；

end

end

上述算法在构造简化二进制分辨矩阵的过程中，每次判断 M'' 是否逻辑包含 (s_i',s_j')，以及 (s_i',s_j') 是否逻辑包含 M' 时，需要逐位进行逻辑加运算；而事实上，只需要判断 M'' 每行或者 (s_i',s_j') 行中1所在的列，(s_i',s_j') 或者 M' 在该列是否也为1即可，如果是，则删除 (s_i',s_j') 所在的行或者 M'' 所在的行。

算法 2 基于二进制分辨矩阵的决策系统的启发式约简算法[14]

输入：决策系统〈U,C∪D〉；

输出：C 相对于 D 的一个约简 Red；

1. 利用算法1产生简化二进制分辨矩阵 M″；
2. 计算 C 中每个非当前核属性 a 的 f(a)值(如果 C 中存在非当前核属性)，其中 $f(a)=|\{(s_i,s_j)\mid$ 行 (s_i,s_j) 在列 a 处的值为1，$(s_i,s_j)\in M''\}|$；
3. 如果 C 中存在非当前核属性，则转4，否则转5；
4. 选择 f( )值最小的非当前核属性，设为 a′，将 a′对应的列从 M″中删除，转3；
5. 取 M″中剩下的属性组成属性集 Red，Red 即为所求的约简。

上述基于删除法的属性约简算法中，每次选择一个列方向上1的个数最少的属性进行删除，它的迭代次数是非常高的；同时，如果有多个属性的 $f(a)$ 值同时达到最小值，则随机选择一个删除，这种处理方式只考虑了二进制分辨矩阵列方向的情况，即该属性在所有实例对中出现的次数，而没有综合考虑行方向的情况，因此并不完善。

4 改进算法的分析

文献[17]给出了基于差别矩阵的属性约简算法需满足的两个条件，针对二进制分辨矩阵属性约简，这两个条件等价

于：一致决策表 $S=\langle U,C\cup D,V,f\rangle$，$BM_{n(n-1)/2\times m}$ 为决策表 S 的二进制分辨矩阵，$BM=(BM(i,j,c_k))_{n(n-1)/2\times m}$，$u_i\ \overline{IND}u_j$ 表示实例对 u_i,u_j 关于条件属性集 C 是不可分辨的，$BM(i,j,\_)$ 为实例对 u_i,u_j 所在行的行向量，P 是 S 的约简，$R_p=(\lambda_1,\lambda_2,\cdots,\lambda_k,\cdots,\lambda_m)$ 是 P 的二进制向量表示，需满足下列两个条件：

(1) $\forall(u_i,u_j)\in U\times U\wedge u_i\ \overline{IND(C)}u_j$，$BM(i,j,\_)\cdot(\lambda_1,\lambda_2,\cdots,\lambda_k,\cdots,\lambda_m)^{\mathrm{T}}\neq 0$

(2) $\exists(u_i,u_j)\in U\times U\wedge u_i\ \overline{IND(C)}u_j$，若 $\lambda_k=1$，$BM(i,j,\_)\cdot(\lambda_1,\lambda_2,\cdots,\overline{\lambda_k},\cdots,\lambda_m)^{\mathrm{T}}=0$

其中，条件(1)说明 P 为决策表 S 的一个约简，而条件(2)说明 P 中每一个属性都是必要的，因此改进算法的约简结果需满足以上两个条件。

特别地，对于简化的一致决策表而言，二进制分辨矩阵 BM 中不存在全零行，因此，条件(1)、(2)可简化为：

(1) $\forall(u_i,u_j)\in U\times U$，$BM(i,j,\_)\cdot(\lambda_1,\lambda_2,\cdots,\lambda_k,\cdots,\lambda_m)^{\mathrm{T}}\neq 0$

(2) $\exists(u_i,u_j)\in U\times U$，若 $\lambda_k=1$，$BM(i,j,\_)\cdot(\lambda_1,\lambda_2,\cdots,\overline{\lambda_k},\cdots,\lambda_m)^{\mathrm{T}}=0$

定理 2 决策表 $S=\langle U,C\cup D,V,f\rangle$，$BM_{n(n-1)/2\times m}$ 为决策表 S 的二进制分辨矩阵，决策表 S 是可约简的，其必要条件是 $\sum_{k=1}^{|C|}BM(i,j,c_k)>1$。

文献[14]给出的基于二进制分辨矩阵的属性约简算法，每次选择一个列方向 1 的个数最少的属性置 0，其迭代次数是非常高的，因此，本文算法考虑一次性将多个列置 0，但是此操作可能会导致二进制分辨矩阵出现新的全 0 行，于是有如下定理：

定理 3 决策表 $S=\langle U,C\cup D,V,f\rangle$，$BM_{n(n-1)/2\times m}$ 为决策表 S 的二进制分辨矩阵，属性集合 C' 中所有属性是可删除的，其充分条件是，若将 C' 所在的列置 0，BM 中不出现新的全 0 行。

证明：若 $\sum_{k=1}^{|C|}BM(i,j,k)>1$，不妨设 $BM(i,j,c_p)=1$，$BM(i,j,c_q)=1,\cdots,BM(i,j,c_k)=1$，属性集合 $C'\subset\{c_p,c_q,\cdots,c_k\}$，若将 C' 所在的列置 0，出现新的全 0 行，即 $\exists(u_i,u_j)\in U\times U\wedge u_i\ \overline{IND(C)}u_j$，使得 $BM(i,j,\_)=0$，则对于决策表 S 的任意约简 $(\lambda_1,\lambda_2,\cdots,\lambda_m)$，都有 $(\lambda_1,\lambda_2,\cdots,\lambda_m)^{\mathrm{T}}\cdot BM(i,j,\_)=0$，不满足二进制分辨矩阵属性约简条件(1)，说明属性集合 C' 中所有属性是不可删除的。

由定理 3 可知，为了使得属性集合 C' 中所有属性是可删除的，必须保证每次将多个列置 0 前，二进制分辨矩阵是简化的，即不存在相互逻辑包含的行。

推论 1 二进制分辨矩阵 BM 中，若存在若干行 1 的个数相同，不妨设行 (u_i,u_j) 中，$c_1,c_2,\cdots,c_k$ 列的值为 1，属性集合 $C'\subset\{c_1,c_2,\cdots,c_k\}$，若将 C' 所在的列置 0，与行 (u_i,u_j) 中 1 的个数相同的行不为全 0。

由推论 1 可知，对于 1 的个数相同的若干行，一次性将多个列置 0 后，这些行中不出现全 0，则在二进制分辨矩阵简化过程中，1 的个数相同的行之间不需要比较，因此，本文算法在对二进制分辨矩阵简化时，根据每行 1 的个数将 BM 各行划分等价类，这样，只需要比较不同等价类之间的实例对，从而提高了二进制分辨矩阵简化效率。

定义 8 $R_{\max}=\{(p,q,t)\mid t=\max_{1\leqslant i<j\leqslant|U|}\{\sum_{k=1}^{|C|}BM(i,j,k)\}\}$，其中 $|U|$ 表示决策表 S 的样本个数，$|C|$ 为条件属性总数。$R_{\max}$ 表示二进制分辨矩阵 BM 中，实例对 (x_p,x_q) 所在行中 1 的总数最多，且总数为 t。

定义 9 若 $BM(p,q,c)=1$，$C(p,q,c)=\sum_{1\leqslant i<j\leqslant|U|}BM(i,j,c)$，其中 $c\in C$，$|U|$ 为决策表 S 的样本数，$C(p,q,c)$ 表示实例对 (x_p,x_q) 所在的行在属性 c 上的值为 1 时，属性 c 对应的列中 1 的总数，这里用 $\mathrm{Max}(C(c))$ 表示 1 的总数最多的列。

定义 10 若 $BM(i,j,c)=1$，$CR(c)=\sum_{1\leqslant i<j\leqslant|U|}\frac{BM(i,j,c)}{\sum_{k=1}^{|C|}BM(i,j,c_k)}$，其中，$c\in C$，$\sum_{k=1}^{|C|}BM(i,j,c_k)$ 表示实例对 (x_i,x_j) 所在行中 1 的总数，$\frac{BM(i,j,c_k)}{\sum_{k=1}^{|C|}BM(i,j,c_k)}$ 表示实例对 (x_i,x_j) 所在行中，属性 c_k 占该行中 1 的总数的比重，$CR(c)$ 表示在所有行中，属性 c 占各行 1 的总数的比重的和。$CR(c)$ 的值越大，表示属性 c 越重要。这里，用 $\mathrm{Max}(CR(c))$ 表示 $CR(c)$ 最大的值。

在这里，为了保证每次删除尽量多的冗余属性，选择了二进制分辨矩阵中每行 1 的总数最多的行，即 $R_{\max}$；其次，在将哪几列属性置 0 时，选择保留 $\mathrm{Max}(C(c))$，将其余的相对不重要的属性所在的列置 0。若出现多个 $\mathrm{Max}(C(c))$，算法将综合考虑行列两个方向的情况，计算它们的 $CR(c)$ 值，并保留 $\mathrm{Max}(CR(c))$ 所在的列，因为 $C(c)$ 只考虑了某个属性在所有实例对中出现的次数。同时，还需要考虑任意一个实例对能被多少个属性区分。

5 改进的基于二进制分辨矩阵的属性约简算法描述

经过上述分析，针对文献[10,14]中给出的属性约简方法的不足，现提出改进的基于二进制分辨矩阵的算法(算法 3)。

算法 3

输入：决策系统〈U,C∪D〉；

输出：C 相对于 D 的一个约简 Red；

步骤 1 求解 U/C；

步骤 2 在 U/C 的每个等价类中抽取一个对象，组成简化决策表〈U′,C∪D〉，$U'=\{u_1',u_2',\cdots,u_{n'}'\}$；

步骤 3 根据简化决策表构造简化二进制分辨矩阵；

步骤 4 计算 $R_{max}=(p,q,t)$；

若 t==1

Red 为二进制分辨矩阵 BM 中 1 所在列的属性集合。

若 t!=1

计算 R_{max} 实例对所在的行中值为 1 所在的列 $C(p,q,c_{c_1})$，$C(p,q,c_{c_2})$，…，$C(p,q,c_{c_k+1})$，保留 $\mathrm{Max}(c_{c_k})$；若存在多个 $\mathrm{Max}(c_{c_k})$，如 $\mathrm{Max}(c_{c_k-1})$，$\mathrm{Max}(c_{c_k})$，$\mathrm{Max}(c_{c_k+1})$，则计算 $CR(c_{c_k-1})$，$CR(c_{c_k})$，$CR(c_{c_k+1})$，选取 $\mathrm{Max}(CR(c_{c_k}))$ 保留，同时将 $c_{c_1},c_{c_2},\cdots,c_{c_k-1},c_{c_k+1}$ 所在的列置 0。

步骤 5 利用 hash 散列方法根据 BM 中各行 1 的总数将 BM 每行记录划分等价类，得到 $sum_1,sum_2,\cdots,sum_i,\cdots,sum_k$，其中 sum_i 表示具有相同 h_i 值的实例对集合，且每行记录的 hash 编码值为该行 1 的总数，记 $sum_i=\{(x_a,x_e)\mid hash(x_a,x_e\}=h_i\}$，$h_1<h_2<\cdots<h_i<\cdots<h_k$。

```
for(j=1;j≤k;j++)
  for(i=j+1;i<=k;i++)
    if(BM(x1,y1)∈sumj ∧ BM(x2,y2)∈sumi)
      {
        BM(x1,y1)中,行值为1所在的列col[1],col[2],…,
        col[k],对于BM(x2,y2)中,col[1],col[2],…,col[k]也
        为1,则删除BM(x2,y2)所在的行。
      }
    End
End
```

转至步骤4。

算法说明:

每次执行完步骤4后,都要执行步骤5来简化二进制分辨矩阵,这是为了防止步骤4操作使矩阵出现新的全0行,以至于违背定理2的内容。至于如何简化二进制分辨矩阵,根据推论1可知,对于二进制分辨矩阵 BM,1的总数相同的若干行,将多个列置0不会出现全0,因此步骤5利用hash散列方法根据每行1的总数划分等价类,且等价类根据每行1的总数的大小由低向高排列,最后从不同的等价类中选择实例对与行值为1所在的列进行比较,并删除逻辑包含行。

算法3的步骤1、步骤2时间复杂度均为 $O(|C||U|)$,步骤3中每次只比较1元素所在的列,时间复杂度为 $O(|U/C|^2|BM|C_i)$,C_i 表示记录行中1的总数,步骤4时间复杂度近似为 $O(|BM||C|)$,步骤5时间复杂度为 $O(\sum_{j=1}^{k}\sum_{i=j+1}^{k}h_j|sum_j||sum_i|)$,在最坏情况下其复杂度近似为 $O(|BM|^2|C|)$,所以本文算法的时间复杂度为 $O(|U/C|^2|BM||C|)$,空间复杂度为 $O(|U/C|^2|BM||C|)$。

6 实例分析

给定决策表 $S=\langle U,C\cup D,V,f\rangle$ 如表1所列,其中论域 $U=\{x_1,x_2,x_3,x_4,x_5,x_6\}$,条件属性 $C=\{a_1,a_2,a_3,a_4,a_5,a_6\}$,决策属性 $D=\{d\}$。

表1 决策表

对象	条件属性						
	a_1	a_2	a_3	a_4	a_5	a_6	d
x_1	1	2	1	2	0	1	1
x_2	1	1	1	2	0	2	0
x_3	2	0	1	0	0	2	1
x_4	0	1	1	2	1	0	0
x_5	3	2	0	2	2	3	0
x_6	1	2	2	1	0	1	0

S的二进制分辨矩阵如表2所列。

表2 二进制分辨矩阵

对象	条件属性					
	a_1	a_2	a_3	a_4	a_5	a_6
(x_1,x_2)	0	1	0	0	0	1
(x_1,x_4)	1	1	0	0	1	1
(x_1,x_5)	1	0	1	0	1	1
(x_1,x_6)	0	0	1	1	0	0
(x_2,x_3)	1	1	0	1	0	0
(x_3,x_4)	1	1	0	1	1	1
(x_3,x_5)	1	1	1	1	1	1
(x_3,x_6)	1	1	1	1	0	1

简化二进制分辨矩阵如表3所列。

表3 简化二进制分辨矩阵

对象	条件属性					
	a_1	a_2	a_3	a_4	a_5	a_6
(x_1,x_2)	0	1	0	0	0	1
(x_1,x_5)	1	0	1	0	1	1
(x_1,x_6)	0	0	1	1	0	0
(x_2,x_3)	1	1	0	1	0	0

先运用文献[12]中给出的方法得 $f(a_1)=2,f(a_2)=2,f(a_3)=2,f(a_4)=2,f(a_5)=1,f(a_6)=2$,因为 $f(a_5)=1$,将 a_5 所在的列值置0,得到的矩阵如表4所列。

表4 删除属性 a_5 的二进制分辨矩阵

对象	条件属性					
	a_1	a_2	a_3	a_4	a_5	a_6
(x_1,x_2)	0	1	0	0	0	1
(x_1,x_5)	1	0	1	0	0	1
(x_1,x_6)	0	0	1	1	0	0
(x_2,x_3)	1	1	0	1	0	0

因为 $f(a_1)=f(a_2)=f(a_3)=f(a_4)=f(a_6)=2$,则任选一列如 a_6,将 a_6 所在的列置0得到的矩阵如表5所列。

表5 删除属性 a_6 的二进制分辨矩阵

对象	条件属性					
	a_1	a_2	a_3	a_4	a_5	a_6
(x_1,x_2)	0	1	0	0	0	0
(x_1,x_5)	1	0	1	0	0	0
(x_1,x_6)	0	0	1	1	0	0
(x_2,x_3)	1	1	0	1	0	0

第三次迭代选择 a_3,得到的矩阵如表6所列。

表6 删除属性 a_3 的二进制分辨矩阵

对象	条件属性					
	a_1	a_2	a_3	a_4	a_5	a_6
(x_1,x_2)	0	1	0	0	0	0
(x_1,x_5)	1	0	0	0	0	0
(x_1,x_6)	0	0	0	1	0	0
(x_2,x_3)	1	1	0	1	0	0

若干次迭代后,得到的矩阵如表7所列。

表7 最终的二进制分辨矩阵

对象	条件属性					
	a_1	a_2	a_3	a_4	a_5	a_6
(x_1,x_2)	0	1	0	0	0	0
(x_1,x_5)	1	0	0	0	0	0
(x_1,x_6)	0	0	0	1	0	0
(x_2,x_3)	0	1	0	0	0	0

由上表可以看出,根据文献[14]中采用的算法,约简 $P=\{a_1,a_2,a_4\}$。采用本文算法3,计算 $R_{max}=(1,5,4)$,选取实例对 (x_1,x_5) 所在的行,并计算 $C(1,5,c)$,其中 $C(1,5,a_1)=2$,$C(1,5,a_3)=2$,$C(1,5,a_5)=1$,$C(1,5,a_6)=2$,而 $C(1,5,a_1)=C(1,5,a_3)=C(1,5,a_6)=2$,故计算 $CR(a_1)=1/4+1/3$,$CR(a_3)=1/2+1/4$,$CR(a_6)=1/2+1/4$,因为 $CR(a_1)<CR(a_3)=CR(a_6)$,故将 a_1,a_3,a_5 列所在的1值置为0,得到矩阵如表8所列。

通过步骤5简化表8,根据每行1的个数利用hash散列方法划分等价类,其中 $hash(x_1,x_2)=2$,$hash(x_1,x_5)=1$,$hash(x_1,x_6)=1$,$hash(x_2,x_3)=2$,$sum_1=\{(x_1,x_5),(x_1,x_6)\}$,$sum_2=\{(x_1,x_2),(x_2,x_3)\}$,而行 (x_1,x_2) 逻辑包含行 (x_1,x_5),删除行 (x_1,x_2),行 (x_2,x_3) 逻辑包含行 (x_1,x_6),删

除行(x_2,x_3),得到的矩阵如表9所列。

表8 删除属性a_1,a_3,a_5的二进制分辨矩阵

对象	条件属性					
	a_1	a_2	a_3	a_4	a_5	a_6
(x_1,x_2)	0	1	0	0	0	1
(x_1,x_5)	0	0	0	0	0	1
(x_1,x_6)	0	0	0	1	0	0
(x_2,x_3)	0	1	0	1	0	0

表9 简化的二进制分辨矩阵

对象	条件属性					
	a_1	a_2	a_3	a_4	a_5	a_6
(x_1,x_5)	0	0	0	0	0	1
(x_1,x_6)	0	0	0	1	0	0

计算R_{max}中$t=1$,故采用本文算法得出的约简$P=\{a_4, a_6\}$,相比于文献[14]中给出的方法,本方法得到了最小约简,且迭代次数明显降低。

7 算法仿真测试及结果分析

本文选用了UCI中Zoo、Spectf、Spect、lymn、Heart数据集,在2.93GHz CPU、3.21G RAM、WinXP系统的计算机上的Matlab开发环境中进行实验。针对数据集lymn(条件属性18个,决策属性1个,实例数148个),采用本文算法(算法3)得出的约简结果为0,2,13,14,15,16,耗时7.125s,采用文献[14]中的算法,得出的约简结果为0,1,2,3,10,12,14,16,17,耗时16.609s,是本文算法时间消耗的两倍多。

使用RSES针对数据集lymn进行约简结果分析,结果如图1—图3所示。

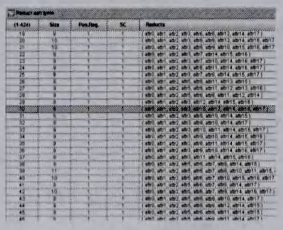

图1 使用RSES约简lymn数据

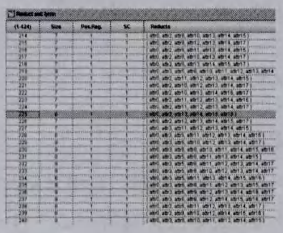

图2 使用RSES约简lymn数据集

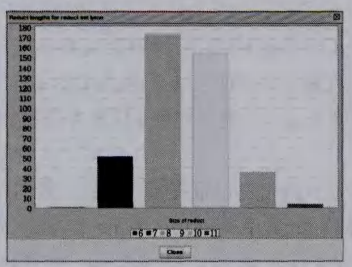

图3 lymn数据集约简结果分析

如图1、图2所示,使用RSES工具约简数据集lymn共得出424个约简结果,其中采用本文算法(算法3)得出的是第225个约简,约简个数为6,采用文献[14]中的算法得出的是第30个约简,约简个数为9。如图3所示,lymn数据集约简的最小个数为6,最大个数为11,因此本文算法相对于文献[14]中的算法,能够得到最小约简。为了进一步验证算法的性能,选用UCI中Zoo、Spectf、Spect、Heart、lymn数据集进行算法对比实验,实验结果如表10所列,算法的执行时间如图4所示。

表10 算法仿真测试结果

数据集	实例数	条件属性数	算法1,2(文献[14])			算法3		
			剩余属性	耗时(s)	是否最小约简	剩余属性	耗时(s)	是否最小约简
Zoo	101	16	6,13,4,8,9,12,16	5.7810	否	3,6,8,13,16	3.6410	是
Spectf	187	44	26,28,42	10.5630	否	33,42	5.8440	是
Spect	187	22	1,3,13,16,19,20,21,22,4,5,9,11	6.25	是	1,3,13,16,19,20,21,22,4,5,9,11	4.9840	是
Heart	270	13	1,5,8	32.2340	是	1,5,8	22.7500	是
lymn	148	18	0,1,2,3,10,12,14,16,17	16.609	否	0,2,13,14,15,16	7.125	是

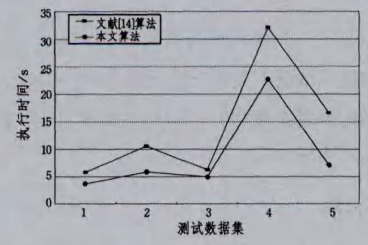

图4 两种算法执行时间对比

从表10可以看出,算法3在属性约简个数上取得了较好的结果,而文献[14]中给出的方法在Zoo、Spectf、lymn上未能得到最小约简,因为它在删除属性时,只考虑了二进制分辨矩阵列方向上1出现的个数。从图4可以发现,算法3一次性将多个属性置0,在条件属性个数、实例数较多的情况下,相比于文献[14]中给出的方法,有效地降低了删除法的迭代次数,提高了算法效率。

结束语 本文首先介绍了粗糙集、简化决策表、二进制分辨矩阵等相关概念;然后在文献[10,14]的基础上对二进制分辨矩阵的简化做了改进,即运用hash散列方法根据每行1的个数划分等价类,减少了矩阵中行与行之间的比较次数,在条件属性较多、数据量较大的情况下,可以有效地提高二进制分辨矩阵简化的效率;其次,针对文献[14]中基于二进制分辨矩阵删除法的属性约简效率不高,且得不到最小属性约简的情况,提出了一种改进的算法,其不仅能够一次性将多个属性列置0,且能够得到最小约简;最后,通过实例及仿真测试证明该算法在效率和约简结果上,都要优于传统的基于二进制分辨矩阵删除法的属性约简[10,14,15]。

下一步工作中,我们会尝试将该算法应用于大数据环境下,将其与Hadoop下的MapReduce框架相结合,使该算法能够适应于大规模数据集约简。

参 考 文 献

[1] Pawlak Z. Rough set[J]. Communications of the ACM, 1995, 38 (11): 89-95

[2] 王国胤. ROUGH 集理论与知识获取[M]. 西安: 西安交通大学出版社, 2001
Wang Guo-yin. Rough Set Theory and Knowledge Acquisition [M]. Xi'an: Xi'an Jiaotong University Press, 2001

[3] Pawlak Z, Skowron A. Rudiments of rough sets[J]. Information Sciences, 2007, 177: 3-27

[4] 钱文斌, 徐章艳, 黄丽宇, 等. 基于信息熵的二进制差别矩阵属性约简算法[J]. 计算机工程与应用, 2010, 46(6): 120-123
Qian Wen-bin, Xu Zhang-yan, Huang Li-yu, et al. Attribution reduction algorithm based on binary discernibility matrix of information entropy[J]. Computer Engineering and Applications, 2010, 46(6): 120-123

[5] Skowron A, Rauszer C. The discernibility matrices and functions in information systems[C]// Slowiński R, ed. Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory. Kluwer, Dordrecht, 1992, 11: 331-362

[6] Hu X H, Cereone N. Learning in Relational Database: A Rough Set Approach[J]. International Journal of Computational Intelligence, 1995, 11(2): 323-338

[7] Yamaguchi D. Attribute dependency functions considering data efficiency[J]. International Journal of Approximate Reasoning, 2009, 51(1): 89-98

[8] Felix R, Ushio T. Rough sets-based machine learning using a binary discernibility matrix[C]// Proceeding of 2nd International Conference on Intelligent Processing and Manufacturing of Materials. Ha-waii, 1999: 299-305

[9] 支天云, 苗夺谦. 二进制可辨矩阵的变换及高效属性约简算法的构造[J]. 计算机科学, 2002, 29(2): 140-142
Zhi Tian-yun, Miao Duo-qian. The binary discernibility matrix's transformation and high efficiency attributes reduction algorithm's conformation[J]. Computer Science, 2002, 29(2): 140-142

[10] 李龙澍, 王慧萍, 徐怡. 二进制可分辨矩阵的最小属性约简算法[J]. 计算机技术与发展, 2010, 20(6): 93-96
Li Long-shu, Wang Hui-ping, Xu Yi. Algorithm for the least attribute reduction of binary discernibility matrix[J]. Computer Technology and Development, 2010, 20(6): 93-96

[11] 陈宸, 赵军. 一种新的基于二进制分辨矩阵的属性约简方法[J]. 计算机应用与软件, 2013, 30(9): 123-127
Chen Chen, Zhao Jun. A new attribute reduction method based on binary discernibility matrix [J]. Computer Applications and Software, 2013, 30(9): 123-127

[12] 徐章艳, 杨炳儒, 宋威. 基于简化的二进制差别矩阵的快速属性约简算法[J]. 计算机科学, 2006, 33(4): 155-158
Xu Zhang-yan, Yang Bing-ru, Song Wei. Quick attribution reduction algorithm based on simple binary discernibility matrix [J]. Computer Science, 2006, 33(4): 155-158

[13] 任倩, 罗月童. 一种二进制可分辨矩阵修正方法及其求核[J]. 小型微型计算机系统, 2013, 34(6): 1437-1440
Ren Qian, Luo Yue-tong. An new method for modifying binary discernibility matrix and computation of core[J]. Journal of Chinese Computer Systems, 2013, 34(6): 1437-1440

[14] 蒙祖强, 史忠植. 一种新的基于简化二进制可辨矩阵的相对约简算法[J]. 控制与决策, 2008, 23(9): 976-978
Meng Zu-qiang, Shi Zhong-zhi. Algorithm for relative reduction based on simplifried binary discernibility matrix[J]. Control and Decision, 2008, 23(9): 976-978

[15] 桂现才. 简化的二进制差别矩阵属性约简算法的改进[J]. 计算机工程与设计, 2007, 28(16): 3971-3973
Gui Xian-cai. Improved algorithm for attribute reduction based on simple binary discernibility matrix[J]. Computer Engineering and Design, 2007, 28(16): 3971-3973

[16] 杨传健, 葛浩, 李龙澍. 垂直划分二进制可分辨矩阵的属性约简[J]. 控制与决策, 2013, 28(4): 563-568
Yang Chuan-jian, Ge Hao, Li Long-shu. Attribute reduction of vertically partitioned binary discernibility matrix [J]. Control and Decision, 2013, 28(4): 563-568

[17] Wang Jue, Wang Ju. Reduction algorithms based on discernibility matrix: the ordered attributes method[J]. Journal of Computer Science and Technology, 2001, 16(6): 489-504

(上接第 188 页)

[5] Li Bing. Research on Key Technology of Self Adaptive Software [D]. Jilin: Jilin University, 2012

[6] Hu Qing-lei, Xiao Bing. Adaptive fault tolerant control using integral sliding mode strategy with application to flexible spacecraft [J]. International Journal of Systems Science, 2013, 44 (12): 2273-2286

[7] Zhang You-sheng. Software architecture [M]. Beijing: Tsinghua University press, 2006

[8] Li Jin-gang, Zhao Shi-lei, Du Ning. The theory and application of software architecture [M]. Beijing: Tsinghua University press, 2013

[9] Zhou Su, Peng Bin, Zhang Yong, et al. Software architecture and design [M]. Beijing: Tsinghua University press, 2013

[10] Chen Xiang-dong. New system based on event-driven and service-oriented business activity monitoring design and implementation[J]. Application Research of Computers, 2012, 29(3): 977-980

[11] Tang Shan, Li Li-ping, Tan Wen-an. Research on Runtime Monitoring for Self-adaptive and Reconfigurable Software Systems [J]. Computer Science, 2013, 40(11): 191-196

[12] Gao Jun, Shen Cai-liang, Zheng Mei-fang et al. Architecture description language of software oriented self-adaptive[J]. Application Research of Computers, 2010, 27(5): 1796-1801

[13] Chen Xiang-dong. Characteristics and Application of New Generation Websites System[J]. Journal of Beihua University(Natural Science), 2011, 12(3): 359-362

[14] Li Li, Liao Jian-wei, Ou Ling. Clound computing: an introduction [J]. Application Research of Computers, 2010, 27(12): 4419-4422

[15] Chen Quan, Deng Qian-ni. Clound computing and its key techniques[J]. Journal of Computer Applications, 2009, 29(9): 2562-2567

[16] Pan Jian, Zhou Yu, Luo Bin, et al. An Ontology-based Software Self-adaptation Mechanism [J]. Computer Science, 2007, 34 (11): 264-269

[17] Ding Bo, Wang Huai-min, Shi Dian-xi. Pervasive middleware technology [J]. Journal of Computer Science and Frontiers, 2007, 1(3): 241-254

[18] Liu Hui, Shi Dian-xi, Liu Ming, et al. Oriented Self-Adaptive Software Integrated Environment[J]. Computer Engineering & Science, 2010, 32(1): 105-108