计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.190500125

基于优化可辨识矩阵和改进差别信息树的属性约简算法

徐　怡[1,2]　唐静昕[2]
1 安徽大学计算智能与信号处理教育部重点实验室　合肥 230039
2 安徽大学计算机科学与技术学院　合肥 230601

摘　要　运用可辨识矩阵表示信息系统中所有对象的区分信息，为研究属性约简提供了新方向。然而，传统的可辨识矩阵在构造结束后才利用核属性消除冗余元素项，忽略了核属性在矩阵构建过程中的作用。针对这一问题，文中做了以下研究：1)优化可辨识矩阵的构造方式，在计算任意两个对象的区分信息之前，先判断核属性上的取值是否相等，如果不相等，则直接将对应元素项记为 ϕ，忽略对其他条件属性的判断；2)提出属性加权重要度的概念，综合考虑每个条件属性占可辨识矩阵中非空元素项的比率(称为宏观重要度)与每个属性对区分对象的贡献程度(称为微观重要度)，并通过例子说明了该度量方法的合理性；3)针对优化后的矩阵仍然存在大量冗余元素和空集这一缺陷，结合差别信息树的概念提出基于优化可辨识矩阵和属性加权重要度的差别信息树。按照属性加权重要度对优化可辨识矩阵中所有非空元素项进行排序，使得重要度高的属性被更多的节点共享；且在构建过程中将不包含核属性的元素项映射到树中的一条路径上，而包含核属性的元素项则被直接忽略。最后，提出基于优化可辨识矩阵和改进差别信息树的约简算法 HSDI-tree。在 UCI 的 5 个数据集上分别比较了 HSDI-tree 算法与 CDI-tree，DI-tree 和 IDI-tree 算法的约简结果和节点个数，实验结果表明 HSDI-tree 算法能有效找到最小属性约简且空间压缩能力更好。

关键词：粗糙集；属性重要度；可辨识矩阵；属性约简；差别信息树
中图法分类号　TP181

Attribute Reduction Algorithm Based on Optimized Discernibility Matrix and Improving Discernibility Information Tree

XU Yi[1,2] and TANG Jing-xin[2]
1 Key Laboratory of Intelligent Computing and Signal Processing and Ministry of Education, Anhui University, Hefei 230039, China
2 College of Computer Science and Technology, Anhui University, Hefei 230601, China

Abstract　Discernibility matrix expresses the distinguishing information of all objects in the information system with matrix elements, which provides a new idea for attribute reduction. However, the traditional discernibility matrix uses the core attributes to eliminate redundant element items after the construction is finished, ignoring the role of the core attributes in the matrix construction process. In response to this problem, the following research is done. Firstly, the definition of the discernibility matrix is optimized. Before calculating the distinguishing information of any two objects, it is first determined whether the values on the core attributes are equal. If not, the corresponding element items are directly recorded as ϕ, and the judgment of other attributes is ignored. Secondly, the concept of attribute weighted importance is proposed. The ratio of each condition attribute to the non-empty element term in the discernibility matrix (called macro importance) and the contribution of each attribute to the distinguishing object (called micro Importance) are comprehensively considered, and the rationality of the measurement method is illustrated by an example. Thirdly, aiming at the disadvantages that there are a lot of redundant elements and empty sets in the optimized discernibility matrix, by combining the concept of discernibility information tree, discernibility information tree based on optimized discernibility matrix and attribute weighted importance is proposed. All non-empty element items in the optimized discernibility matrix are sorted according to attribute weighted importance, so that attributes with high importance are shared by more nodes. Element items that do not contain core attributes are mapped to a path in the tree during the build process, while element items that contain core attributes are ignored. Finally, a reduction algorithm HSDI-tree based on optimized discernibility matrix and improving discernibility information tree is proposed. This paper compared the reduction results and the number of nodes of the HS-

到稿日期：2019-05-22　返修日期：2019-09-21　本文已加入开放科学计划(OSID)，请扫描上方二维码获取补充信息。
基金项目：国家自然科学基金 (61402005)；安徽省自然科学基金(1308085QF114)；安徽省高等学校省级自然科学基金(KJ2013A015)
This work was supported by the National Natural Science Foundation of China under Grant(61402005), Natural Science Foundation of Anhui Province under Grant (1308085QF114) and provincial Natural Science Foundation of Anhui higher education institute under Grant (KJ2013A015).
通信作者：徐怡(xuyi1023@126.com)

DI-tree algorithm, CDI-tree, DI-tree and IDI-tree algorithms on the five data sets of UCI. The experimental results show that the HSDI-tree algorithm can effectively find the minimum attribute reduction and has better space compression ability.

Keywords Rough set, Attribute importance, Discernibility matrix, Attribute reduction, Discernibility information tree

1 引言

粗糙集理论是一种处理不完整与不确定性数据的有效工具，由波兰数学家 Pawlak 于 1982 年提出[1]，被广泛应用于数据挖掘、知识发现、机器学习和规则提取等领域。属性约简作为粗糙集理论的重要研究部分[2]，旨在保持知识库分类能力的情况下删除冗余的条件属性，并从特征信息中提取有效信息，进而简化知识的表现形式，提高决策效率。属性约简在数据量大、数据维度高的情况下，对于简化后续问题处理和提高求解速度等发挥着重要作用。目前，属性约简的方法主要有基于正域[3]、基于信息熵[4]、基于可辨识矩阵[5]、基于基因[6]和基于蚁群[7]等。其中，属性重要度的度量方法对属性约简的优劣起着至关重要的作用。能否准确地计算决策表中所有属性的重要度关乎着约简算法的实现效率与准确度。

Skowron 等[8]提出了被众多学者广泛关注[9-10]的可辨识矩阵的概念，以及基于可辨识矩阵的属性约简方法；Yamaguchi[11]在 Skowron 可辨识矩阵的基础上，结合条件属性频率，提出了一种属性依赖性模型；Felix 等[12]提出了矩阵元素仅由 0 和 1 组成的二进制可辨识矩阵；文献[13]提出的度量属性重要性的方法仅依赖于矩阵列方向，而矩阵的行特征对属性重要性的影响不予考虑；文献[14]提出以矩阵行为主、列为辅的属性重要性策略。而研究表明，行方向对属性重要性的影响相比列方向而言是次要特征，这导致了以该度量策略为基础的约简算法存在性能缺陷。以上文献在二进制可辨识矩阵的基础上进行研究，对矩阵的行或行列两个方向的特征分别进行排序以选择属性，排序耗时相对较长。

本文在前人研究的基础上，首先优化了可辨识矩阵的定义：在计算任意两个对象的区分信息之前，先判断核属性上的取值是否相等，如果不相等，则直接将对应元素项置为 Ø，忽略对其他条件属性的判断。在核属性出现早、数量多的情况下，这种定义方式能大大缩短构造矩阵所消耗的时间。与此同时，基于可辨识矩阵提出了属性加权重要度的概念：综合考虑每个条件属性占可辨识矩阵中非空元素项的比率（称为宏观重要度）与每个属性对区分对象的贡献程度（称为微观重要度），并通过例子说明了该度量方法的合理性。其次，针对优化后的矩阵仍然存在大量冗余元素和空集这一缺陷，结合文献[15]介绍的差别信息树的概念，提出了基于优化可辨识矩阵和属性加权重要度的改进差别信息树：对优化可辨识矩阵中的所有非空元素项按属性加权重要度重新进行降序排列，并以此为基础构建差别信息树，使得重要度高的属性被更多的节点共享；并且，该树在构建过程中将不包含核属性的元素项映射到一条路径上，包含核属性的元素项则直接被忽略，相同的元素项映射到同一路径，具有父子集关系的元素项共享子集对应的路径。最后，提出了基于优化可辨识矩阵和改进差别信息树的属性约简算法：在算法每次迭代的过程中，删除当前重要度最高的属性所对应的所有路径，直到得到一棵空树。该算法为属性约简策略提供了一种新思路。

本文第 2 节介绍粗糙集理论的基本概念；第 3 节介绍可辨识矩阵并对其进行优化；第 4 节提出属性加权重要度的计算方法；第 5 节提出改进的差别信息树；第 6 节提出基于优化可辨识矩阵和改进差别信息树的属性约简算法；第 7 节通过实验验证算法的有效性；最后总结全文。

2 基本概念

本节介绍 Pawlak 约简以及可辨识矩阵等基本概念[16-17]。

定义 1 设 $S=(U, AT=C\cup D, \{V_a \mid a\in AT\}, \{I_a \mid a\in AT\})$为一个决策表，其中 U 代表非空有限的元素集，即论域；C 为非空有限的条件属性集，D 为决策属性，$C\cap D=\varnothing$；V_a 代表属性 a 的值域；$I_a: U\rightarrow V_a$ 代表一个信息函数。

定义 2 设 $S=(U, AT=C\cup D, \{V_a \mid a\in AT\}, \{I_a \mid a\in AT\})$为一个决策表，$P\subseteq(C\cup D)$，$P$ 在 U 上的不可分辨关系定义为 $IND(P)=\{(x,y)\in U\times U \mid \forall a\in P, I_a(x)=I_a(y)\}$。

定义 3 设 $S=(U, AT=C\cup D, \{V_a \mid a\in AT\}, \{I_a \mid a\in AT\})$为一个决策表，对于 $\forall R\subseteq C$，如果 R 满足以下两个条件，那么 R 为 Pawlak 约简。

$$POS_R(D)=POS_C(D) \tag{1}$$

$$\forall r\in R, POS_{R-\{r\}}(D)\neq POS_R(D) \tag{2}$$

定义 4 决策表 $S=(U, C\cup D, \{V_a\}, \{I_a\})$，其中 $|U|=n$，$U=\{x_1, x_2, \cdots, x_n\}$。$I_a(x)$ 代表对象 x 在属性 a 上的取值。S 的可辨识矩阵为 $\boldsymbol{M}=(m(x,y))$，其定义如下：

$$\boldsymbol{M}=(m(x,y))=\begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{21} & m_{22} & \cdots & m_{2n} \\ \vdots & \vdots & & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{nn} \end{bmatrix} \tag{3}$$

其中，$m(x,y)$定义为：

$$m(x,y)=\begin{cases} \{a\in C \mid I_a(x)\neq I_a(y)\}, & D(x)\neq D(y) \\ \varnothing, & \text{otherwise} \end{cases} \tag{4}$$

3 优化可辨识矩阵

Skowron 可辨识矩阵很好地阐述了对象之间的区分关系，但矩阵的元素项较多，内存开销大。针对 Skowron 可辨识矩阵存在的问题，本文优化了矩阵的构造方式，在缩短构造时间的同时，减少了矩阵中有效元素项的个数。本节首先给出优化可辨识矩阵构造算法，再通过一个例子来说明优化可辨识矩阵的创建过程。

优化可辨识矩阵的构造算法如算法 1 所示。

算法 1 优化可辨识矩阵的构造算法

输入：决策表 S

输出：优化可辨识矩阵 $\mathbf{M}'_{ij}$

1. 数组 C 存储条件属性。令 CORE=Ø，m=0；

2. for(i=2;i<=|U|;i++)

```
3.    for(j=1;j<i;j++)
4.      if(D(x_i)≠D(x_j))
5.        if(CORE≠∅∧(∃a∈CORE,I_a(x_i)≠I_a(x_j)))
6.          m(x_i,x_j)′=∅;
7.        else
8.        for(k=0;k<|C|;k++)
9.          if(I_C[k](x_i)≠I_C[k](x_j))
10.           m(x_i,x_j)′=m(x_i,x_j)′∪C[k];
11.           m++;
12.       if(m==1)
13.         CORE=CORE∪m(x_i,x_j)′;
14. else
15. m(x_i,x_j)′=∅
16. 输出 M′_ij，算法结束。
```

下面通过一个例子来阐述优化可辨识矩阵的构造过程。

例 1　给定某一决策表 S，其中论域为 $U=\{O_1,O_2,O_3,O_4,O_5\}$，条件属性集为 $C=\{a,b,c,d,e\}$，决策属性为 D。S 的具体信息如表 1 所列。

表 1　决策表 S

Table 1　Decision information S

	a	b	c	d	e	D
O_1	3	1	2	0	1	1
O_2	1	0	1	2	2	2
O_3	3	2	2	0	1	2
O_4	1	0	1	1	3	3
O_5	2	0	3	3	1	3

1）根据 Skowron 可辨识矩阵的定义，得到 S 的可辨识矩阵 $\boldsymbol{M}$，如表 2 所列。

表 2　Skowron 的可辨识矩阵 $\boldsymbol{M}$

Table 2　Skowron discernibility matrix $\boldsymbol{M}$

	O_1	O_2	O_3	O_4	O_5
O_1	∅				
O_2	$\{a,b,c,d,e\}$	∅			
O_3	$\{b\}$	∅	∅		
O_4	$\{a,b,c,d,e\}$	$\{d,e\}$	$\{a,b,c,d,e\}$	∅	
O_5	$\{a,b,c,d\}$	$\{a,c,d,e\}$	$\{a,b,c,d\}$	∅	∅

2）根据优化可辨识矩阵的定义，得到 S 的优化可辨识矩阵 $\boldsymbol{M}'_{ij}$，如表 3 所列。

表 3　优化可辨识矩阵 $\boldsymbol{M}'_{ij}$

Table 3　Optimized discernibility matrix $\boldsymbol{M}'_{ij}$

	O_1	O_2	O_3	O_4	O_5
O_1	∅				
O_2	$\{a,b,c,d,e\}$	∅			
O_3	$\{b\}$	∅	∅		
O_4	∅	$\{d,e\}$	∅	∅	
O_5	∅	$\{a,c,d,e\}$	∅	∅	∅

由表 2 和表 3 可以看出，在构造优化可辨识矩阵的过程中，一旦判断条件属性 b 满足定义加入 $CORE$ 后，对象 O_4O_1，O_4O_3，O_5O_1 和 O_5O_3 对应的元素项便不再计算，而直接记为 ∅。可见，优化后的可辨识矩阵中的有效元素减少，便于简化后续的约简计算。

4　属性加权重要度

属性重要度被广泛应用于属性约简，是粗糙集理论中的重要概念之一[18]。能否准确、快速地度量属性重要度关系着属性约简算法效率的高低与约简结果的优劣。

由可辨识矩阵的定义可知，所有出现在矩阵中的属性对于对象的区分都是有贡献的，只是不同属性的重要度不一定相同。因此，本文提出基于可辨识矩阵的属性加权重要度的计算方法（$Wsig$）。该方法综合考虑每个条件属性占分辨矩阵的比率（宏观重要度）与每个条件属性对区分信息的贡献大小（微观重要度）。属性加权重要度的计算方法如下。

定义 5　$S=(U,C\cup D,V,f)$ 为一个决策表，$\boldsymbol{M}$ 为决策表 S 的可辨识矩阵，m_{ij} 对应可辨识矩阵第 i 行 j 列的非空元素，$|M_{ij}|$ 表示可辨识矩阵中非空元素项的个数。对于 $\forall a\in C$，属性 a 对于 D 的宏观重要度 $Ssig$ 定义为：

$$Ssig(a)=\frac{N_a}{|M_{ij}|} \tag{5}$$

其中，N_a 表示属性 a 在可辨识矩阵中出现的次数。

定义 6　$S=(U,C\cup D,V,f)$ 为一个决策表，$\boldsymbol{M}$ 为决策表 S 的可辨识矩阵，m_{ij} 对应可辨识矩阵第 i 行 j 列的非空元素，$|M_{ij}|$ 表示可辨识矩阵中非空元素项的个数。对于 $\forall a\in C$，属性 a 对于 D 的微观重要度 $Hsig$ 定义为：

$$Hsig(a)=\frac{\sum\limits_{i,j=1,2,\cdots,n} a\cap m_{ij}}{N} \tag{6}$$

其中，$a\cap m_{ij}=\begin{cases}\dfrac{1}{|m_{ij}|}, & a\cap m_{ij}\neq\varnothing,i,j=1,2,\cdots,n\\ 0, & a\cap m_{ij}=\varnothing,i,j=1,2,\cdots,n\end{cases}$。$|m_{ij}|$ 表示非空元素项 m_{ij} 中属性的个数，N 表示包含属性 a 的元素项的个数。

定义 7　在可辨识矩阵 $\boldsymbol{M}$ 中，对于 $\forall a\in C$，属性 a 的加权重要度 $Wsig$ 定义为：

$$Wsig(a)=\beta\times Ssig(a)+(1-\beta)\times Hsig(a) \tag{7}$$

当某个属性多次出现时，表示它能分辨的对象多，相应的重要性也大。按照常理，宏观重要度 $Ssig$ 的权重比微观重要度 $Wsig$ 的权重大，因此假定 $\beta>0.5$。根据以上定义，得到如下性质。

性质 1　在决策表 $S=(U,C\cup D,V,f)$ 中，m_{ij} 为可辨识矩阵 $\boldsymbol{M}$ 中的非空元素。对于 $\forall a\in C$，有 $0<Ssig(a)\leqslant 1$。

证明：对于 $\forall a\in C$，若属性 a 出现在可辨识矩阵的所有元素项 m_{ij} 中，则 $Ssig(a)=\frac{N_a}{|M_{ij}|}=1$；对于 $\forall a\in C$，因为属性 a 至少出现一次，所以 $Ssig(a)=\frac{N_a}{|M_{ij}|}>0$；对于 $\forall a\in C$，若属性 a 存在于部分可辨识矩阵的元素项中，那么 $Ssig(a)=\frac{N_a}{|M_{ij}|}<1$。综上所述，$0<Ssig(a)\leqslant 1$。

性质 2　在决策表 $S=(U,C\cup D,V,f)$ 中，m_{ij} 为可辨识矩阵 $\boldsymbol{M}$ 中的非空元素。对于 $\forall a\in C$ 且 $a\in m_{ij}$，有 $0<Hsig(a)\leqslant 1$。

证明：对于 $\forall a\in C$ 且 $a\in m_{ij}$，因为 m_{ij} 为可辨识矩阵 $\boldsymbol{M}$ 中的非空元素，由定义 6 直接得到 $a\cap m_{ij}>0$，所以 $Hsig(a)=\frac{\sum\limits_{i,j=1,2,\cdots,n} a\cap m_{ij}}{N}>0$；对于 $\forall a\in C$ 且 $a\in m_{ij}$，若可辨识矩阵 $\boldsymbol{M}$ 中的元素都为单个属性，即均为核属性时，有 $\frac{1}{|m_{ij}|}=1$，即

$a \cap m_{ij} \frac{1}{|m_{ij}|}=1$。那么，$Hsig(a)=\frac{\sum\limits_{i,j=1,2,\cdots,n} a \cap m_{ij}}{N}=1$。对于$\forall a \in C$且$a \in m_{ij}$，若可辨识矩阵$\boldsymbol{M}$中的元素不都是单个属性，也存在多个属性构成的元素项，则必然有$0<\frac{1}{|m_{ij}|}<1$，也即$0<\sum\limits_{i,j=1,2,\cdots,n} a \cap m_{ij}<N$，那么$Hsig(a)=\frac{\sum\limits_{i,j=1,2,\cdots,n} a \cap m_{ij}}{N}<\frac{N}{N}=1$。综上所述，$0<Hsig(a) \leqslant 1$。

5 改进的差别信息树

优化的可辨识矩阵虽然在一定程度上减少了元素项个数，但是矩阵中的重复元素和空集占据了大量的存储空间；而差别信息树能在保证不丢失约简信息的条件下，对矩阵中的非空元素进行压缩存储。

本节先介绍差别信息树的概念，再结合优化可辨识矩阵和属性加权重要度给出改进差别信息树的构建算法。

定义 8[18] 差别信息树是一棵有序的前缀树，基于条件属性从左向右构建，顺序不能颠倒。

差别信息树有以下特征[19]：

1)差别信息树中每个节点至多有$|C|$个孩子节点。

2)差别信息树中的每个节点由 4 个部分组成：节点名称、前缀指针、后继指针和同名指针。前缀指针指向该节点的父亲节点，后继指针指向该节点的孩子节点，同名指针指向树中其他包含该节点名称的路径。

构建差别信息树的算法请见文献[19]，这里不再赘述。

在构建差别信息树的过程中删除核属性及核属性所在的路径，可以进一步压缩存储信息。同时，差别信息树中不含核属性的各子树之间也可能存在包含关系，而合理的属性顺序对差别信息树的结构有很大影响。例如，对于子树$\langle d,e\rangle$与$\langle a,c,d,e\rangle$，如果按照$\{a,c,d,e\}$的属性顺序，那么对应差别信息树的两条路径分别为$\langle d,e\rangle$和$\langle a,c,d,e\rangle$；但如果属性顺序为$\{d,e,a,c\}$，由于$\{d,e\} \subset \{d,e,a,c\}$，那么$\langle d,e,a,c\rangle$将映射到路径$\langle d,e\rangle$上。可见，在构建差别信息树时结合属性重要度，将会在保持区分信息的情况下更大程度地减小存储空间。根据上述思想，本节基于优化可辨识矩阵和属性加权重要度对差别信息树进行改进。改进差别信息树的构造算法如算法 2 所示。

算法 2 改进差别信息树的构造算法(HSDI)

输入：优化可辨识矩阵 $\mathbf{M}'_{ij}$，CORE

输出：改进的差别信息树 HSDI

1. 初始化约简集 REDU=∅，SList 为条件属性集合；
2. 建立差别信息树的根节点 ROOT，令 ROOT 为空；
3. for(i=2;i<=n;i++) /* n 为对象个数 */
4. for(j=1;j<i;j++)
5. 相应元素项记为 HI_{ij}；
6. if($HI_{ij} \cap CORE=\varnothing$)
7. if($HI_{ij} \neq \varnothing$)
8. 按照计算得出的属性加权重要度对 HI_{ij}中的属性按降序重新排列；
9. While($HI_{ij} \neq \varnothing$)
{
10. 将 HI_{ij}中最左侧的元素标记为 a；
11. if(ROOT 的某个后继节点 TN 的属性名 a)
{
12. if(TN 为叶子节点，返回)
13. else ROOT=TN；
}
14. else
{
15. 创建新节点 TN′，并将属性名标记为 a. 将同名的指针指向有相同属名的节点上，形成同名属性节点链；
16. 令 ROOT=TN′；
}
17. $H_{ij} \leftarrow H_{ij}-\{a\}$
}
18. 输出 HSDI，算法结束。

6 基于优化可辨识矩阵和改进差别信息树的属性约简算法

本节给出了基于优化可辨识矩阵和改进差别信息树的属性约简算法，并举例阐述了该算法的具体执行过程。

基于优化可辨识矩阵和改进差别信息树的属性约简算法如算法 3 所示。

算法 3 基于优化可辨识矩阵和改进的差别信息树约简算法(HSDI-tree)

输入：改进差别信息树 HSDI，SList

输出：属性约简 REDU

1. 初始化约简集 REDU=∅；
2. 扫描 HSDI 中的节点，将属性指针头表中没有指向的属性从 SList 中删除；
3. While(SList≠∅ ∧ SDI-tree≠∅)
{
4. 令 SList 中第一个属性为 S_i；
5. 根据指针头表搜索同名指针构成的指针链，在搜索过程中若存在属性名相同的节点，则删除含有该节点的路径，且 REDU=REDU∪S_i；
6. SList←SList−{s_i}；
}
7. 输出 REDU，算法结束。

下面通过例 2 来简述基于优化可辨识矩阵和改进差别信息树的属性约简的计算过程。

例 2(接例 1) 计算决策表 S 的约简集 $REDU$。

首先计算各条件属性的加权重要度。本文假设$\beta=0.6$，则$Ssig(b)=\frac{N_b}{|M'_{ij}|}=\frac{1}{2}$，$Hsig(b)=\frac{\sum\limits_{i,j=1,2,\cdots,5} b \cap m_{ij}}{N}=\frac{\frac{1}{5}+1}{2}=0.6$，$Wsig(b)=\beta \times Ssig(b)+(1-\beta) \times Hsig(b)=0.54$。同理可得，$Wsig(a)=Wsig(c)=0.39$，$Wsig(d)=Wsig(e)=0.577$。按重要度降序排列为$\{d,e,b,a,c\}$，其中$CORE=\{b\}$。如果用传统的度量属性重要度的方法，即以属性出现的频率作为属性重要度，那么将计算得到的结果按降序排序为$\{d,e,a,b,c\}$，其中属性 a,b,c 的重要度相同，无法分辨。而按照本

文给出的属性加权重要度的计算方法，属性 b 与属性 a，c 的重要度能够被区分，且核属性 b 的重要度远大于属性 a，c 的重要度。通过这个例子可以看出，属性加权重要度的概念是合理的。优化可辨识矩阵中的元素项按属性重要度排序后的结果如表 4 所列。

表 4　更改后的优化可辨识矩阵

Table 4　Optimized discernibility matrix after updating

	O_1	O_2	O_3	O_4	O_5
O_1	Ø				
O_2	$\{d,e,b,a,c\}$	Ø			
O_3	$\{b\}$	Ø	Ø		
O_4	Ø	$\{d,e\}$	Ø	Ø	
O_5	Ø	$\{d,e,a,c\}$	Ø	Ø	Ø

根据算法 2 的主要思想，构造改进差别信息树的过程如下：

1）初始时，$CORE=\{b\}$，$REDU=\emptyset$。

2）创建根节点 *root*。

3）当构建第一个元素项 $\{d,e,b,a,c\}$ 对应的第一条路径时，因为 $CORE\cap\{d,e,b,a,c\}\neq\emptyset$，所以路径 $\langle d,e,b,a,c\rangle$ 不用构建。

4）第二个元素项 $\{b\}$ 对应一条新的路径 $\langle b\rangle$。

5）第三个元素项 $\{d,e\}$ 对应新的路径 $\langle d,e\rangle$。

6）对于优化可辨识矩阵的第四个元素项 $\{d,e,a,c\}$，因为 $\{d,e\}\subset\{d,e,a,c\}$，所以 $\{d,e,a,c\}$ 对应的路径为 $\langle d,e\rangle$。

至此，优化可辨识矩阵中所有非空元素项都已映射到改进的差别信息树上。形成的改进差别信息树如图 1 所示。根据算法 3 进行属性约简的过程如下：

7）此时 $SList=\{d,e,b\}$，选取 *SList* 中的第一个属性 d，通过同名指针链删除包含属性 d 的路径，且 $REDU=REDU\cup d$，此时 *SList* 更改为 $SList=\{e,b\}$。

8）选取 *SList* 中的第一个属性 e，通过同名指针链判断没有含属性 e 的路径，因此直接从 *SList* 删除 e，此时 $SList=\{b\}$。

9）选取 *SList* 中的第一个属性 b，通过同名指针链判断存在含有属性 b 的路径，因此删除包含属性 b 的路径，且 $REDU=REDU\cup b$，此时 $SList=\emptyset$；

10）此时改进的差别信息树只有根节点，算法结束。最终的约简结果为 $REDU=\{d,b\}$。

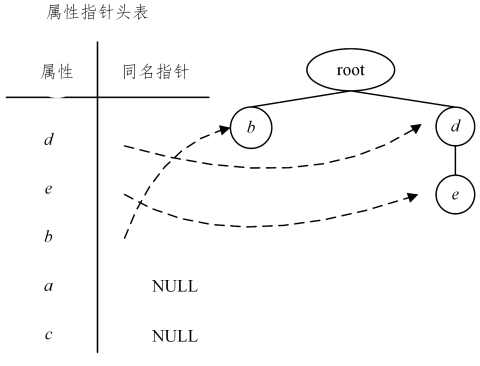

图 1　改进的差别信息树

Fig. 1　Improved discernibility information tree

7　实验结果与分析

为了验证本文提出的 HSDI-tree 算法的有效性，从 UCI 机器学习数据库中选用了 5 个数据集（数据集的具体信息如表 5 所列），实验环境为 Microsoft Windows 10，2.9 GHz Intel Core i5，8G RAM 和 Visual Stadio. NET 2010 平台的 VC++，与 IDI-tree 算法[18]、DI-tree 算法[19]和 CDI-tree 算法[20]进行了两组对比实验。第一组实验比较这 4 种算法进行属性约简的结果（见表 6）；第二组实验进行压缩能力的比较（见表 7）。

表 5　数据表的基本信息

Table 5　Basic information of data sets

Datasets	Size of objects	Number of feature
letter	20000	15
voting	435	16
tic-tac-toe	985	9
Poker hand	25010	10
Lymphography	148	18

表 6　4 种算法的约简结果

Table 6　Reduction results of four algorithms

Datasets	Size of core attributes	Reduction results			
		HSDI-tree	IDI-tree	DI-tree	CDI-tree
letter	4	11	11	12	12
voting	7	9	9	10	9
tic-tac-toe	0	7	7	7	7
Poker hand	5	6	6	8	6
Lymphography	0	7	7	8	8

表 6 列出了 4 种算法进行约简计算后得到的约简集的大小。从表 6 可以看出，HSDI-tree 约简算法与 IDI-tree 算法一样，比 CDI-tree 和 DI-tree 约简算法更能有效地找到最小属性约简。

表 7　4 种算法的节点个数

Table 7　Number of nodes of four algorithms

Datasets	Size of core	Reduction results			
		HSDI-tree	IDI-tree	DI-tree	CDI-tree
letter	20000	217	295	2075	545
voting	435	26	32	582	50
tic-tac-toe	985	45	45	45	45
Poker hand	25010	18	18	84	23
Lymphography	148	949	1512	1003	1003

由表 7 可知，与 CDI-tree，DI-tree 和 IDI-tree 算法相比，HSDI-tree 算法得到的节点数量更少，说明 HSDI-tree 进一步压缩了可辨识矩阵中非空元素所占的空间。因此，可以认为 HSDI-tree 具有更好的空间压缩能力。

结束语　本文首先改进了可辨识矩阵的定义方式，使得优化后的矩阵在核属性出现早、数量多的情况下能大大缩短构造矩阵所消耗的时间。其次，本文基于可辨识矩阵提出了属性加权重要度的概念，通过例子证明了该度量方法的合理性。由于优化后的可辨识矩阵仍然存在大量重复元素和空集，因此提出了基于优化可辨识矩阵和属性加权重要度的改进差别信息树，并提出了基于优化可辨识矩阵和改进差别信息树的属性约简算法。该方法结合了属性重要度和核属性的概念，在利用优化可辨识矩阵优势的同时，考虑了属性重要度

对压缩能力的影响，进一步优化了树结构。实验结果表明，本文提出的基于优化可辨识矩阵和改进差别信息树的属性约简算法在压缩能力上优于已有的改进差别信息树的约简算法。但是，该算法仅在单粒度环境下进行，没有考虑多粒度的情况。下一步将研究多粒度环境下能否结合差别信息树的知识提高属性约简效率，并实现压缩存储。

参 考 文 献

[1] PAWLAK Z. Rough set[J]. International Journal of Information and Computer Science, 1982, 11(5): 341-356.

[2] SHEN Q, JENSEN R. Selecting informative features with fuzzy-rough sets and its application for complex systems monitoring [J]. Pattern Recognition, 2004, 37(7): 1351-1363.

[3] WEN S D, BAO Q H. A fast heuristic attribute reduction approach to ordered decision systems[J]. European Journal of Operational Research, 2018, 264: 440-452.

[4] ZHENG J G, YAN R X. Attribute Reduction Based on Cross Entropy in Rough Set Theory[J]. Journal of Information & ComputationalScience, 2012, 9(3): 745-750.

[5] WEN S D, BAO Q H. A fast heuristic attribute reduction approach to ordered decision systems[J]. European Journal of Operational Research, 2018, 264: 440-452.

[6] MAJI P, PAUL S. Rough Set Based Maximum Relevance-Maximum Significance Criterion and Gene Selection from Microarray Data[J]. International Journal of Approximate Reasoning, 2011, 152: 408-420.

[7] ELEYAN D. Ant Colony Optimization based Feature Selection in Rough Set Theory[J]. International Journal of Computer Science and Electronics Engineering, 2013, 1(2): 244-247.

[8] SKOWRON A, RAUSZER C. The discernibility matrices and functions in information systems [M] // Intelligent Decision Support-handbook of Applicationsand Advances of the Rough Sets Theory. Dordrecht: Kluwer Academic Publisher, 1991: 331-362.

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

[10] PARTHALÁIN, NEIL M, SHEN Q, et al. distance measure approach to exploring the rough set boundary region for attribute reduction[J]. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(3): 305-317.

[11] YAMAGUCHI D. Atribute dependency functions considering data Efficiency[J]. International Journal of Approximate Reasoning, 2009, 51(1): 89-98.

[12] FELIX R, USHIO T. Rough sets-based machine learning using a binary discernibility matrix[C] // Proceeding of 2nd International Conference on Inteligent Procesing and Manufacturing of Materials. Ha-wai, 1999: 299-305.

[13] QIAN W B, XU Z Y, HUANG L Y, et al. Atribution reduction algorithm based on binary discernibility matrix of information entropy [J]. Computer Enginering and Applications, 2010, 46(6): 120-123.

[14] ZHI T Y, MIAO D Q. The Binary Discernibility Matrix's Transformation and High Efficiency Attributes Reduction Algorith m's Conformation[J]. Computer Science, 2002, 29(2): 140-143.

[15] JIANG Y. Attribute reduction with rough set based on discernibility information tree[J]. Control & Decision, 2015, 30(8): 1531-1536.

[16] PAWLAK Z, SKOWRON A. Rudiments of rough sets[J]. Information Sciences, 2007, 177: 3-27.

[17] PAWLAK Z. Rough Sets: Theoretical Aspects of Reasoning About Data[M]. Boston: Kluwer Academic Publishers, 1991.

[18] JIANG Y. Attribute reduction with rough set based on Improving discernibility information tree[J]. Control & Decision, 2019, 34(6): 135-140.

[19] YANG L, ZHANG X Y, XU W H. Attribute reduction of discernibility information tree in interval-valued ordered information system[J]. Journal of Frontiers of Computer Science & Technology, 2019(6): 1062-1069.

[20] JIANG Y, YU Y. Minimal attribute reduction with rough set based on compactness discernibility information tree[J]. Soft Computing, 2016, 20: 2233-2243.

XU Yi, born in 1981, Ph. D, associate professor, is member of China Computer Federation. Her main research interests include intelligent information processing and rough set.