

粒计算中基于属性分类的形式概念属性约简

徐 怡^{1,2†}, 王 泉², 霍思林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: 针对目前已有的形式概念属性约简算法的不足(如属性约简的时间复杂度偏高、属性及属性值比较过程中存在冗余计算、存储开销大等问题), 结合粒计算思想, 提出基于属性分类的形式概念属性约简模型. 首先, 通过定义两个算子来划分属性之间分类关系; 然后, 由属性分类关系制定约简规则, 并在此基础上提出基于属性分类的形式概念约简算法, 该算法在保持目前最低时间复杂度不变的情况下, 减少了冗余计算和存储开销, 提高了属性约简的计算效率; 最后, 通过实例和仿真实验对基于属性分类关系的形式概念属性约简算法的有效性进行了验证.

关键词: 形式概念分析; 粒计算; 属性分类; 属性约简

中图分类号: TP18

文献标志码: A

Formal concept attribute reduction model based on attribute classification relation

XU Yi^{1,2†}, WANG Quan², HUO Si-lin²

(1. Key Laboratory of Intelligent Computing and Signal Processing, Ministry of Education, Anhui University, Hefei 230039, China; 2. School of Computer Science and Technology, Anhui University, Hefei 230601, China)

Abstract: In view of the shortcomings of existing formal concept attribute reduction algorithms, such as the problems that the time complexity of attribute reduction is too high, redundancy calculation exists in attribute and attribute value comparison, and storage overhead is great and so on, by combining the idea of granular computing this paper proposes a conceptual attribute reduction model based on attribute classification. Firstly, two operators are defined to classify the relations among attributes. Then, a reduction rule is established by attribute classification relation. On this basis, this paper proposes a formal concept reduction algorithm based on attribute classification, which reduces the redundancy calculation and storage overhead while keeping the current minimum time complexity constant, which improves the computational efficiency of attribute reduction. Finally, the validity of the formal attribute reduction algorithm based on classification is verified by examples and simulation experiments.

Keywords: formal concept analysis; granular computing; attribute classification; attribute reduction

0 引 言

形式概念分析理论^[1-2], 是一种能够从形式背景进行数据分析和规则提取的强有力工具. 形式概念分析理论是根据概念的哲学思想提出的一种数学方法^[3-4], 从新的视角对知识发现进行定义, 对问题背景中的对象、属性以及对象属性关系等用形式化的语境表述, 并根据语境构造格结构, 一般称为概念格. 通过研究概念格的结构达到知识发现、知识分析和获取知识的目的. 形式概念分析广泛应用于软件工程、

信息处理、知识发现、数据挖掘等相关领域^[5-7]. 形式概念分析中的一个重要研究方向是形式概念属性约简. 虽然每个属性都具有不同的作用, 对应的特征也不相同, 但是在大数据的背景下如何提高数据分析的效率至关重要, 而属性约简就是其中的一个重要研究方向. 目前, 关于形式概念的属性约简算法的研究大多是基于属性的可辨识矩阵或可辨识函数进行的, 如 Zhang 等^[8] 提出了一种新的属性约简概念, 可在形式背景中保留所有概念及其层次结构, 同时给出了

收稿日期: 2017-07-10; 修回日期: 2017-09-06.

基金项目: 国家自然科学基金项目(61402005); 安徽省自然科学基金项目(1308085QF114); 安徽省高等学校省级自然科学基金项目(KJ2013A015); 国家留学基金委员会资助项目(201606505034); 安徽大学计算智能与信号处理教育部重点实验室课题项目.

作者简介: 徐怡(1981—), 女, 副教授, 博士, 从事智能信息处理、粒计算、粗糙集理论等研究; 王泉(1992—), 男, 硕士生, 从事粒计算的研究.

†通讯作者. E-mail: xuyi1023@126.com

一种基于差别矩阵的属性约简方法;Wei等^[9]提出了基于粗糙集理论的概念格属性约简,利用概念格的依赖空间找到最小的属性子集;基于粒计算的思想,Wu等^[10]利用信息粒的方式研究了形式概念的属性约简;Medina^[11]研究了面向对象和面向属性的概念格的属性约简.以上这些形式概念属性约简算法的时间复杂度都偏高,因此,Chen等^[12]提出了概念格的属性约简算法,基于覆盖粗糙集的思想,利用覆盖交约简的方法实现概念格的属性约简,降低计算时间复杂度.然而,该方法在属性约简过程中存在冗余计算和存储开销大的问题,因此,本文结合粒计算的思想,提出了基于属性分类的形式概念约简算法.该算法根据定义的两个算子计算出属性之间的分类关系,并通过属性分类关系制定出约简规则,实现了形式概念属性约简.该算法在保持目前最低时间复杂度不变的情况下,减少了冗余计算和存储开销,提高了形式概念属性约简的计算效率.

1 基本概念

下面介绍将要用到的相关概念^[12-13].

定义1 形式背景 K 是一个三元组, $K = (G, M, I)$,其中 G 为所有对象的集合, M 为所有属性的集合, $I \subseteq G \times M$ 为 G 和 M 中元素之间的关系集合.对于 $g \in G, m \in M, (g, m) \in I$ 或者 gIm ,表示“对象 g 拥有属性 m ”.

定义2 设 $K = (G, M, I)$ 为一个形式背景,对于集合 $A \subseteq G$,记

$$A^I = \{m \in M | (g, m) \in I, \forall g \in A\}.$$

相应地,对于集合 $B \subseteq M$,记

$$B^I = \{g \in G | (g, m) \in I, \forall m \in B\}.$$

定义3 设 $K = (G, M, I)$ 为一个形式背景,根据集合 $A \subseteq G$ 和 $B \subseteq M$,如果 $A^I = B$ 且 $B^I = A$,则称 $X = (A, B)$ 是一个“形式概念”,此时,集合 A 为 X 的外延,集合 B 为 X 的内涵.特别地,用 $B(K)$ 表示 K 上所有形式概念组成的集合.

定义4 设 $K = (G, M, I)$ 是一个形式背景,形式概念所有的属性约简集合被定义为 $\text{reduct}(K) = \{M_i : 0 < i \leq |M|\}$, M_i 为约简后的属性核.此时属性集合 M 可被划分为3个部分:

- 1) 绝对不必要的属性集合 $A_M = M - \bigcup M_i$;
- 2) 相对必要的属性集合 $B_M = \bigcup M_i - \bigcap M_i$;
- 3) 绝对必要的属性集合 $C_M = \bigcap M_i$.

定理1 设 $K = (G, M, I)$ 是一个形式背景,对于 $D \subset M, D \neq \emptyset, E = M - D$, D 被称为概念格一致性集合,当且仅当 $e \in E$ 时,有 $((e^I)^I \cap D)^I = e^I$.

定理2 设 $K = (G, M, I)$ 是一个形式背景,对

于 $D \subset M, D \neq \emptyset, E = M - D$, D 被称为概念格的约简,当且仅当 $e \in E$,且 $d \in D$ 时,有 $((d^I)^I \cap (D - \{d\}))^I \neq d^I$.

2 基于属性分类关系的形式概念分析属性约简模型

目前已有的形式概念属性约简算法的时间复杂度偏高,为此,Chen等^[12]提出了基于广义覆盖粗糙集的形式概念属性约简算法,将覆盖交约简的方法应用于形式概念模型,降低了属性约简的时间复杂度.然而,该方法在属性约简过程中存在冗余计算和系统存储开销大的问题,主要表现为:1) 属性等价类和属性约简是分开计算的,该计算过程中存在属性值重复遍历和重复计算;2) 将形式背景知识向覆盖知识转换时,增加了系统存储的开销.因此,本文结合粒计算的思想,通过定义两个算子计算出属性的分类关系(等价、可约、独立),并由属性分类关系制定出约简规则,实现形式概念的属性约简,有效减少了属性约简在计算过程中存在冗余计算和存储开销的问题.

2.1 模型介绍

给定一个形式背景,对于属性 $m \in M$,对象 $g \in G, Vm(g)$ 表示对象 g 在属性 m 上的属性值.将数值形式 $Vm(g)$ 表示为形式背景中对象属性之间的关系 (g, m) ,若 $(g, m) \in I$,则 $Vm(g) = 1$;反之,若 $(g, m) \notin I$,则 $Vm(g) = 0$.

定义5 给定形式背景 $K = (G, M, I)$,属性 $m_1, m_2 (m_1, m_2 \in M)$,对于 $Vm_1(g), Vm_2(g)$,有

$$\text{sub}_{(m_1, m_2)}(g) = Vm_1(g) - Vm_2(g).$$

$\text{sub}_{(m_1, m_2)}(g)$ 表示属性 m_1 和 m_2 在对象 g 上的差值,反映属性 m_1 和属性 m_2 两者之间的关系(等价关系、包含关系).这里,若 $\text{sub}_{(m_1, m_2)}(g) = 0$,则表示属性 m_1 与 m_2 计算出的概念是相同的,即在对象 g 上属性 m_1 与 m_2 等价;若 $\text{sub}_{(m_1, m_2)}(g) = 1$,则表示属性 m_2 构造出的形式概念是属性 m_1 构造形式概念的子概念,即在对象 g 上属性 m_1 包含属性 m_2 ;若 $\text{sub}_{(m_1, m_2)}(g) = -1$,则表示属性 m_2 构造出的形式概念是属性 m_1 构造形式概念的父概念,即在对象 g 上属性 m_2 包含属性 m_1 .

定义6 给定形式背景 $K = (G, M, I)$,定义两个算子 H_1 和 H_2 ,并设置初始值为0,若 $\text{sub}_{(m_1, m_2)}(g) = 0$,则设置 H_1 和 H_2 都不变;若 $\text{sub}_{(m_1, m_2)}(g) = 1$,则设置 $H_1 = 1, H_2$ 值不变;若 $\text{sub}_{(m_1, m_2)}(g) = -1$,则设置 H_1 值不变, $H_2 = 1$.对于所有对象 $g \in G$ 而言,有以下性质:

- 1) 若 $H_1 = 0, H_2 = 0$,则在对象 G 上, $m_1^I = m_2^I$;

2) 若 $H_1 = 1, H_2 = 0$, 则在对象 G 上, $m_1^I \subseteq m_2^I$;
 3) 若 $H_1 = 0, H_2 = 1$, 则在对象 G 上, $m_2^I \subseteq m_1^I$;
 4) 若 $H_1 = 1, H_2 = 1$, 则在对象 G 上, m_1^I 与 m_2^I 无关. 其中, 对于所有对象 $g \in G$, $\text{sub}_{(m_1, m_2)}(g)$ 在比较过程中, 第4个性质一旦满足, 则此时对象遍历终止; 否则, 将所有的对象全部遍历结束.

定义7 在定义6的基础上, 若 $\forall g \in G$, 属性 m_1 和 m_2 满足 $H_1 = 0, H_2 = 0$, 则 $\text{NE}_0(m_1) = \{m_2\}$, $\text{NE}_0(m_2) = \{m_1\}$. 其中 $\text{NE}_0(m)$ 表示关于属性 m 的等价类集合.

定义8 在定义6的基础上, 对于所有对象 $g \in G$, 属性 m_1 和 m_2 : 1) 若满足 $H_1 = 1, H_2 = 0$, 则 $\text{NE}_1(m_2) = \{m_1\}$; 2) 若满足 $H_1 = 0, H_2 = 1$, 则 $\text{NE}_1(m_1) = \{m_2\}$. 其中 $\text{NE}_1(m)$ 表示关于属性 m 的包含类集合.

定义9 给定形式背景 $K = (G, M, I)$, 对于属性 $m \in M$, 若 $\text{NE}_1(m) = \{m_1, m_2, \dots, m_n\}$, 则 $\text{Mul}(m) = \sum_{g \in G} (Vm_1(g) \otimes Vm_2(g) \otimes \dots \otimes Vm_n(g))$, 即 $\text{Mul}(m) = |m|$, 属性 m 可约, 否则属性 m 不可约.

性质1 设 $K = (G, M, I)$ 是一个形式背景, 对于 $m \in M, P(m) = |\text{NE}_1(m)| \geq 2$ 且 $\text{Mul}(m) = |m|, Q(m) = |\text{NE}_0(m)|$, 有以下结论:

- 1) m 是一个核属性, 当且仅当 $Q(m) < 1, P(m) = 0$;
- 2) m 是一个绝对不必要的属性, 当且仅当 $Q(m) < 1, P(m) = 1$;
- 3) m 是一个相对必要的属性, 当且仅当 $Q(m) \geq 1, P(m) = 0$.

证明 对于性质1中的结论1), 由定义7和定义8可知, 当 $Q(m) < 1, P(m) = 0$ 时, 表示属性 m 没有等价类集合, 同时也没有包含类结合, 即属性 m 相对属性集中其他属性是独立的, 则认为属性 m 是一个核属性. 同理, 结论2) 和结论3) 易可证. \square

性质1的结论是属性分类关系的依据, 本文通过研究属性分类关系来实现形式概念属性约简.

2.2 基于属性分类的形式概念属性约简算法

本节提出的基于属性分类的形式概念属性约简算法的思想是: 首先, 定义两个算子 H_1 和 H_2 , 并分别初始化为0. 然后, 将属性集中任意两个属性进行属性值比较运算, 计算出 H_1 和 H_2 的值, 若 $H_1 = H_2 = 0$, 则表示这两个属性之间满足等价关系, 并设置等价类集合 $\text{NE}_0(m)$; 若 $H_1 = H_2 = 1$, 则表示这两个属性之间存在独立关系; 若 $H_1 = 0 \& H_2 = 1$ 或 $H_1 = 1 \& H_2 = 0$, 则表示这两个属性之间存在包含关系, 并设置包含类集合 $\text{NE}_1(m)$, 直到所有的属性两两比较结束. 最后, 根据属性的分类关系实现形式概念的

属性约简. 在计算某个属性的等价类过程中可以直接将该属性等价类中的元素进行约简, 减少冗余计算, 提高计算效率. 对于属性 m 是否可约, 若属性 m 的包含类集合满足条件 $|\text{NE}_1(m)| \geq 2 \& \text{Mul}(m_i) = |m|$, 则属性 m 可约.

算法1 基于属性分类的形式概念属性约简算法.

Input: 形式背景 $K = (G, M, I)$;

Output: K 中一个属性约简核.

Step 1: 令 $H_1 = 0, H_2 = 0, \text{red} \leftarrow M$;

Step 2: For each $m_i \in M (1 \leq i < n)$, 记 $|M| = n$

Step 3: if $\text{NE}_0(m_i) == \text{null}$

Step 4: $j = i + 1$;

Step 5: while(ture)

Step 6: if $\text{NE}_0(m_j) == \text{null}$

Step 7: For each $g \in G$

Step 8: if $H_1 = 1 \& H_2 = 1$ break;

Step 9: if $\text{sub}_{(m_i, m_j)}(g) = 1, H_1 = 1$;

Step 10: else if $\text{sub}_{(m_i, m_j)}(g) = -1, H_2 = 1$;

Step 11: if $H_1 = 0 \& H_2 = 0$

Step 12: $\text{NE}_0(m_i) \leftarrow \text{NE}_0(m_i) + m_j$;

Step 13: $\text{NE}_0(m_j) \leftarrow \text{NE}_0(m_j) + m_i$;

Step 14: $\text{red} \leftarrow \text{red} - m_j$;

Step 15: else if $H_1 = 1 \& H_2 = 0$

Step 16: $\text{NE}_1(m_j) \leftarrow \text{NE}_1(m_j) + m_i$;

Step 17: else if $H_1 = 0 \& H_2 = 1$

Step 18: $\text{NE}_1(m_i) \leftarrow \text{NE}_1(m_i) + m_j$;

Step 19: if $j < n, j++$;

Step 20: else break;

Step 21: for each $m \in M$

Step 22: if $|\text{NE}_1(m)| \geq 2 \& \text{Mul}(m) = |m|$

Step 23: $\text{red} \leftarrow \text{red} - m$;

Step 24: 输出 red.

算法1是计算形式概念的属性约简, 基于粒计算的思想, 通过两个启发式算子计算出属性之间的分类关系, 并根据属性之间分类关系的划分实现形式概念的属性约简. 其中, 在Step 22中判断属性 m 的包含类集合个数是否大于2, 若存在这样的属性并结合定义9即能判断属性 m 是否可约, Step 2~Step 20 计算属性之间的分类关系(等价、包含、独立), 时间复杂度为 $O(nn)$; Step 21~Step 24 计算核属性集, 时间复杂度为 $O(m)$. 即该算法在整个计算过程中时间复杂度为 $O(nn)$, m 为对象个数, n 为属性个数.

3 实验研究

3.1 实例分析

为了考察算法的有效性,选择文献[12]中已知的形式背景 $K = (G, M, I)$ 进行属性约简分析. 如表1所示, $M = \{a, b, c, d, e\}$ 表示5个属性, $G = \{1, 2, 3, 4\}$ 表示4个对象.

表1 形式背景

G	a	b	c	d	e
1	0	0	1	0	0
2	0	1	0	0	0
3	1	0	1	1	1
4	1	1	0	1	0

根据所提出的基于属性分类关系的形式概念分析属性约简算法对表1进行形式概念属性约简,计算步骤如下所示.

Step 1: 令 $red \leftarrow \{a, b, c, d, e\}$, 其中 red 表示形式背景 K 中的核属性集(不可约的属性集合).

Step 2: 根据定义6, 以属性 a 和 b 进行属性值比较, 并计算 H_1 和 H_2 的值, 如表2所示. 此时, 当 $H_1 = 1$ 和 $H_2 = 1$ 时对象遍历结束, 属性 a 和 b 之间存在独立关系. 同理, 将属性集 M 中所有属性按照表2的方法进行两两比较. 在遍历属性的过程中需要结合表3中的数据, 最终结果如表4所示.

表2 属性 a 和属性 b 关系比较图

G	a	b	$Sub(a, b)$	H_1	H_2
1	0	0	1	0	0
2	0	1	-1	0	1
3	1	0	1	1	1
4	1	1			

表3 属性关系图

集合	a	b	c	d	e
$NE_0(m)$	d	#	#	a	#
$NE_1(m)$	#	#	#	#	$\{a, c\}$

表4 H_1 和 H_2 运行结果图

算子	(a, b)	(a, c)	(a, d)	(a, e)	(b, c)	(b, e)	(c, e)
H_1	1	1	0	1	1	1	1
H_2	1	1	0	0	1	1	0

Step 3: 由表4可知, 属性 (a, d) 满足 $H_1 = 0$ 和 $H_2 = 0$, 即属性 a 和属性 d 等价, 并将属性 d (默认只保存形式背景 K 中第1个遍历的属性) 去除, 即 $red \leftarrow red - d$. 由表4中的数据, 结合定义7和定义8可以得到 $NE_0(m)$ 和 $NE_1(m)$ 的结果, 如表3所示.

Step 4: 遍历表3, 由 $|NE_1(e)| \geq 2$, 并结合定义9, 计算 $Mul(e) = \sum_{g \in G} (Va(g) \otimes Vc(g)) = 1 = |e|$, 所以属性 e 是可约的, 即 $red \leftarrow red - e$; 最终形式背景 $K = (G, M, I)$ 的属性约简结果是 $red = \{a, b, c\}$.

3.2 仿真实验

为了测试基于属性分类关系的形式概念分析属性约简算法的有效性, 将其与文献[2,9,10,12]中相应的算法进行比较. 为了方便, 将文献[2,9,10,12]中的算法分别用 RMI, RAT, RPO 和 RCI 表示, 而本章提出的算法1用 RCL 表示. 本实验使用了10个数据集, 这些数据集均来自于UCI机器学习数据库^[13], 数据集的详细信息如表5所示. 将这10个数据集通过文献[13]中的方法转换成相应的10个形式背景(Data set1 ~ Data set10), 并将5个算法分别在10个形式背景(Data set1 ~ Data set10)上进行属性约简, 实验结果如表6所示.

表5 关于10个数据集的数据描述

No	Data sets	Objects	Condition attributes		
			Boolean	Discreten	Continuou
1	Zoo	101	15	1	0
2	Dermatology	358	1	33	0
3	Soybean	542	17	18	0
4	Car	1728	0	6	0
5	Kr-vs-kp	3196	0	36	0
6	Mushroom	5644	0	22	0
7	Musk	476	0	0	166
8	Semeion	1593	266	0	0
9	Cnae-9	1080	751	105	0
10	Madelon-test	1800	0	0	500

表6 算法的计算时间比较

Data sets	Running time/s				
	RAT	RPO	RMI	RCI	RCL
Data set1	0.048	0.010	0.012	0.001	0.001
Data set2	1.136	2.023	1.142	0.027	0.019
Data set3	0.925	0.353	0.327	0.014	0.009
Data set4	0.538	0.178	0.146	0.015	0.011
Data set5	9.447	1.385	1.364	0.108	0.086
Data set6	10.251	3.531	3.732	0.198	0.157
Data set7	12.165	13.271	10.942	0.354	0.294
Data set8	6.847	10.163	7.286	0.366	0.311
Data set9	112.187	78.135	32.879	1.287	1.027
Data set10	125.412	317.243	202.173	10.689	9.128

实验平台为Windows7操作系统、Intel Core i3 CPU 3.20 GHz, 2 G内存的微机. 在Eclipse中采用Java语言编程实现.

表6列出了RAT、RPO、RMI、RCI和RCL进行属性约简的计算时间. 由实验结果可明显看出,随着数据集中数据量的增加,属性约简的计算时间也相应增加. RCI和RCL好于其他3种算法,尤其数据集中数据量越大差异越明显. 例如,在Data set10中RCI只需要10.698 s, RCL只需要9.128 s, 而RAT, RPO, RMI分别需要125.412 s, 317.243 s, 202.173 s. 原因是RCI和RCL的算法时间复杂度是 $O(|U||A|)$, 而RAT, RMI, RPO的算法时间复杂度是 $O(|U||A|^2)$. 对比RCI和RCL算法可以发现, RCL比RCI花费的时间少, 原因是RCI算法在计算过程中将属性等价与属性约简分开计算时, 计算过程中存在属性与属性值重复遍历和重复计算的情况, 以及将形式背景知识向覆盖知识转换时增加了系统存储的开销. 然而, RCL算法避免了这些问题, 所以, 对于形式概念的属性约简问题, 可以得出RCL算法是有效的.

4 结论

对于目前已有的形式概念属性约简算法的时间复杂度偏高、存在冗余计算以及系统存储开销大的问题. 本文提出了基于属性分类的形式概念属性约简算法. 它从粒计算的思想出发, 通过给定的两个算子计算出属性之间的分类关系(等价、包含、独立), 并制定出相应的约简规则, 同时实现形式概念属性约简. 该方法不仅保持了目前属性约简算法中最低的计算时间复杂度, 而且还避免了冗余计算和系统存储开销大的问题, 从而提高了属性约简的效率. 最后通过实例和仿真实验验证了基于属性分类关系的形式概念属性约简算法的有效性.

参考文献(References)

- [1] Wille R. Restructuring lattice theory: An approach based on hierarchies of concepts[M]. Ordered Sets. Dordrecht: Springer Netherlands, 1982: 445-470.
- [2] Ganter B. Formal concept analysis: Mathematical foundations[M]. New York: Springer-Verlag, 1996: 1-269.
- [3] Belohlavek R, Sigmund E, Zaczal J. Evaluation of IPAQ questionnaires supported by formal concept analysis[J]. Information Sciences, 2011, 181(10): 1774-1786.
- [4] Nguyen T T, Hui S C, Chang K. A lattice-based approach for mathematical search using formal concept analysis[J]. Expert Systems with Applications, 2012, 39(5): 5820-5828.
- [5] Ferjani F, Elloumi S, Jaoua A, et al. Formal context coverage based on isolated labels: An efficient solution for text feature extraction[J]. Information Sciences, 2012, 188: 198-214.
- [6] Li J, Mei C, Lv Y. Knowledge reduction in real decision formal contexts[J]. Information Sciences, 2012, 189: 191-207.
- [7] Kaytoue M, Kuznetsov S O, Napoli A, et al. Mining gene expression data with pattern structures in formal concept analysis[J]. Information Sciences, 2011, 181(10): 1989-2001.
- [8] Zhang W X, Wei L, Qi J J. Attribute reduction in concept lattice based on discernibility matrix[C]. Int Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing. Berlin: Springer, 2005: 157-165.
- [9] Wei L, Qi J J. Relation between concept lattice reduction and rough set reduction[J]. Knowledge-Based Systems, 2010, 23(8): 934-938.
- [10] Wu W Z, Leung Y, Mi J S. Granular computing and knowledge reduction in formal contexts[J]. IEEE Trans on Knowledge & Data Engineering, 2008, 21(10): 1461-1474.
- [11] Medina J. Relating attribute reduction in formal, object-oriented and property-oriented concept lattices[J]. Computers & Mathematics with Applications, 2012, 64(6): 1992-2002.
- [12] Chen J, Li J, Lin Y, et al. Relations of reduction between covering generalized rough sets and concept lattices[J]. Information Sciences An Int J, 2015, 304(C): 16-27.
- [13] Bache K, Lichman M. UCI machine learning repository[EB/OL]. [2016-10-12]. <http://archive.ics.uci.edu/ml/datasets.html>.

(责任编辑: 孙艺红)