

概念格多属性渐减式构造*

马 垣, 马文胜

(辽宁科技大学 软件学院, 辽宁 鞍山 114051)

通讯作者: 马垣, E-mail: mayuanasas@163.com



摘要: 渐进式算法是概念格构造的重要方法之一,但以前的渐进式算法均为渐增式算法,即对象或属性都是增加的.实践表明,很多场合需要属性减少后的概念格.2013年,减少单个属性的渐减式算法已有研究,然而该算法只适用于单个属性,减少多个属性时,该算法需要反复执行多次.研究了减少多个属性的一次性渐减式算法,该算法与减少单个属性的渐减式算法有相同的时间复杂度,但当减少多个属性时,单属性的渐减式算法需要反复执行多次,而该算法只需执行一次.

关键词: 形式概念;概念格;减少属性;渐进式算法;概念格构造

中图法分类号: TP18

中文引用格式: 马垣,马文胜.概念格多属性渐减式构造.软件学报,2015,26(12):3162-3173. <http://www.jos.org.cn/1000-9825/4818.htm>

英文引用格式: Ma Y, Ma WS. Construction of multi-attributes decrement for concept lattice. Ruan Jian Xue Bao/Journal of Software, 2015, 26(12): 3162-3173 (in Chinese). <http://www.jos.org.cn/1000-9825/4818.htm>

Construction of Multi-Attributes Decrement for Concept Lattice

MA Yuan, MA Weng-Sheng

(School of Software Engineering, University of Science and Technology Liaoning, Anshan 114051, China)

Abstract: Incremental algorithms are important methods for the construction of concept lattices. But previous incremental algorithms are for the addition of objects or attributes. Concept lattices decrementing some attributes are needed in practical problems. Incremental algorithm decrementing single attribute were studied in 2013. But the algorithm only applies to decrement single attribute. When decrementing multi-attributes, the algorithm must be implemented again and again. This paper studies an incremental algorithm decrementing multi-attributes. This algorithm has as same time complexity as the algorithm decrementing single attribute. But when decrementing multi-attributes, the algorithm decrementing single attribute must be implemented on many times, the presented algorithm only need to be implemented single execution.

Key words: formal concept; concept lattice; decrementing attribute; incremental algorithm; concept lattice construction

德国数学家 Wille 教授于 1982 年提出了形式概念的理论^[1]. 设 U 是对象集合, M 是属性集合, $I \subseteq U \times M$ 是 U 与 M 间的关系, 则称三元组 $\mathcal{K} = (U, M, I)$ 为一个形式背景. 若 $A \subseteq U, B \subseteq M$, 令 $f(A) = \{m \in M \mid \forall u \in A: (u, m) \in I\}$, $g(B) = \{u \in U \mid \forall m \in B: (u, m) \in I\}$, 并认为若 $f(A) = B, g(B) = A$, 则 (A, B) 是一个形式概念. A 是这个概念的外延, B 是这个概念的内涵. \mathcal{K} 的全部概念的集合记作 $\mathcal{B}(\mathcal{K})$. 若 $(A_1, B_1), (A_2, B_2) \in \mathcal{B}(\mathcal{K})$, 且 $A_1 \subseteq A_2$ (可以证明, 此时必有 $B_2 \subseteq B_1$) 则称 (A_1, B_1) 是 (A_2, B_2) 的子概念, (A_2, B_2) 是 (A_1, B_1) 的父概念, 并且记作 $(A_1, B_1) \leq (A_2, B_2)$. 已证明^[1] 集合 $\mathcal{B}(\mathcal{K})$ 以 \leq 为偏序是一个格 (请见本文第 1 节形式概念的基础定理). 因此, 形式概念的理论又称为概念格理论. 若 A_1 是 A_2 的真子集, 则记 $(A_1, B_1) < (A_2, B_2)$. 若 $(A_1, B_1) < (A_2, B_2)$, 且不存在概念 (A_3, B_3) 使 $(A_1, B_1) < (A_3, B_3) < (A_2, B_2)$, 则称 (A_1, B_1) 是 (A_2, B_2) 的直接子

* 基金项目: 国家自然科学基金(61273019)

Foundation item: National Natural Science Foundation of China (61273019)

收稿时间: 2014-06-18; 定稿时间: 2014-09-24

概念, (A_2, B_2) 是 (A_1, B_1) 的直接父概念, 记作 $(A_1, B_1) \prec (A_2, B_2)$. 有向图 (S, L) 称为是概念格 $\mathfrak{B}(\mathcal{K})$ 的 Hasse 图, 其中, $S = \mathfrak{B}(\mathcal{K})$ 是点的集合, L 是有向边的集合, 当且仅当 $(A_1, B_1) \succ (A_2, B_2)$ 时, $((A_1, B_1), (A_2, B_2)) \in L$. 且在绘制时, 要求父概念总在子概念上面, 并由此而略去有向边的箭头. 由一个形式背景求出它的全部概念及这些概念形成的 Hasse 图, 称为概念格的构造.

现在, 形式概念已得到了广泛的应用, 例如在软件工程^[2]、地理信息系统^[3]、入侵检测^[4]、Web 日志挖掘^[5]、客户群聚类^[6]、Apsect 挖掘^[7]、动态知识学习^[8]、空间聚类^[9]、病症智能诊断^[10]、Folksonomy^[11]、访问控制权限管理^[12]、软件演化分析^[13]、二值命题逻辑约简^[14]等很多领域都有了成功的应用. 然而, 概念的个数是随背景的尺寸指数级增长的, 因此, 概念格的构造一直是很受关注的课题. 目前, 构造概念格的方法可分为两种: 批处理算法^[15]及渐进式算法^[16-19].

直到 2013 年, 渐进式构造算法都是背景增加属性或增加对象时的渐增式构造算法. 然而实际应用中, 有时背景的变化并不是增加属性(或对象), 而是减少属性(或对象), 例如在 Web 信息检索的背景中, 对象是 Web 文档, 属性是关键词. 随着时间的推移, 有些关键词不再被人们所关注了, 若继续将它们保存在背景中, 长期积累下来将会使概念格越来越大, 无效的概念越来越多, 信息检索的速度越来越下降. 因此, 应该将不再受关注的关键词从背景中删除, 压缩概念格的规模, 提高检索速度.

2013 年, 张磊等人^[20]研究了当形式概念的某些属性削减之后, 如何快速有效地在原有的概念格上进行调整, 得到新形式背景的概念格, 而不是传统方式下的重新构造的算法. 张磊等人给出了减少一个属性时自底向上与自顶向下的渐减式算法, 能够同时对概念格及 Hasse 图进行调整, 而且有良好的时间性能. 在这个方向上做了开创性的工作. 但该算法每次只能减少一个属性, 而实际问题中只减少一个属性的情况是较少的, 一般是要减少一些属性, 例如要删除与某个问题相关的一组关键词. 这时, 减少一个属性的算法必须反复执行多次. 能否将每次减少一个属性的算法推广为每次减少多个属性的算法呢? 研究发现: 每次减少一个属性的算法不能简单地推广为每次减少多个属性的算法, 因为减少一个属性时的许多性质在减少多个属性时不再成立. 例如, 一个删除基只对应一个删除节点, 删除节点必然在某个保留节点与某个更新节点之间, ..., 等等, 在减少多个属性时都不再成立. 因此, 一次减少多个属性时必须研究新方法. 张磊等人在文献[20]中指出研究同时删除一个以上属性的概念格的属性渐减算法, 无疑也是一件很有意义的事情. 本文将解决这个问题, 给出一次减少多个属性时快速有效地在原有概念格的基础上得到新背景的概念格的方法.

本文第 1 节给出形式概念的基础定理. 第 2 节给出多属性减少的基本原理. 第 3 节介绍多属性减少的算法. 第 4 节是验证. 第 5 节是后记.

1 形式概念的基础定理

定理 1^[1]. 设 $\mathcal{K} = (U, M, I)$ 为一个形式背景. 若 $A, A_1, A_2 \subseteq U, B, B_1, B_2 \subseteq M$, 则函数 f, g 有以下性质:

- (1) $A_1 \subseteq A_2$, 则 $f(A_1) \supseteq f(A_2)$;
- (2) $B_1 \subseteq B_2$, 则 $g(B_1) \supseteq g(B_2)$;
- (3) $f(A_1 \cup A_2) = f(A_1) \cap f(A_2)$;
- (4) $g(B_1 \cup B_2) = g(B_1) \cap g(B_2)$;
- (5) $A \subseteq g(f(A))$;
- (6) $B \subseteq f(g(B))$;
- (7) $f(A) = f(g(f(A)))$;
- (8) $g(B) = g(f(g(B)))$.

而且由性质(8)可知: 对于 M 的任何子集 $B, (g(B), f(g(B)))$ 一定是概念, $g(B)$ 一定是外延. 特别是 $(g(M), M)$ 一定是概念, 称其为零元概念.

对于 $\mathcal{D} \subseteq \mathfrak{B}(\mathcal{K})$, 若 $(A_0, B_0) \in \mathfrak{B}(\mathcal{K})$ 满足 $\forall (A, B) \in \mathcal{D}: (A, B) \geq (A_0, B_0)$, 则称 (A_0, B_0) 是 \mathcal{D} 的一个下界; 若 (A_0, B_0) 满足 $\forall (A, B) \in \mathcal{D}: (A, B) \leq (A_0, B_0)$, 则称 (A_0, B_0) 是 \mathcal{D} 的一个上界.

定理 2. \mathcal{D} 所有下界的集合有最大元素,称该元素为 \mathcal{D} 的下确界,记作 $\wedge \mathcal{D}$,且 $\wedge \mathcal{D} = (\bigcap_{(A,B) \in \mathcal{D}} A, f(\bigcap_{(A,B) \in \mathcal{D}} A))$; \mathcal{D} 所有上界的集合有最小元素,称该元素为 \mathcal{D} 的上确界,记作 $\vee \mathcal{D}$,且 $\vee \mathcal{D} = (g(\bigcap_{(A,B) \in \mathcal{D}} B), \bigcap_{(A,B) \in \mathcal{D}} B)$.

证明:首先,对任何 $(C,D) \in \mathcal{D}$ 都有 $C \supseteq \bigcap_{(A,B) \in \mathcal{D}} A$,于是 $(\bigcap_{(A,B) \in \mathcal{D}} A, f(\bigcap_{(A,B) \in \mathcal{D}} A)) \leq (C,D)$,即是 \mathcal{D} 的下界之一.若 (E,F) 是 \mathcal{D} 的任意一个下界,则对所有 $(A,B) \in \mathcal{D}$ 都有 $A \supseteq E$,于是 $E \subseteq \bigcap_{(A,B) \in \mathcal{D}} A$.这样,对 \mathcal{D} 的任意一个下界 (E,F) ,我们都有 $(E,F) \leq (\bigcap_{(A,B) \in \mathcal{D}} A, f(\bigcap_{(A,B) \in \mathcal{D}} A))$,所以 $(\bigcap_{(A,B) \in \mathcal{D}} A, f(\bigcap_{(A,B) \in \mathcal{D}} A))$ 是 \mathcal{D} 的下确界.

对偶可证 $(g(\bigcap_{(A,B) \in \mathcal{D}} B), \bigcap_{(A,B) \in \mathcal{D}} B)$ 是 \mathcal{D} 的一个上确界. □

定理 3. 设 $N \subseteq M$,称背景 $\mathcal{K}_1 = (U, N, I \cup U \times N)$ 为 \mathcal{K} 的子背景.若我们记对 \mathcal{K}_1 求的 f 及 g 为 f_1 及 g_1 ,则容易看出:对所有 $A \subseteq U$,都有 $f_1(A) = f(A) \cap N$,以及对所有 $B \subseteq N$,都有 $g_1(B) = g(B)$.

2 多属性减少的基本原理

定义 1. 设背景 $\mathcal{K} = (U, M, I), N \subseteq M$,则称 $\mathcal{B}(\mathcal{K})$ 上的关系 $R_N = \{((A,B), (C,D)) | B \cap N = D \cap N\}$ 为 N 决定的等价关系 (R_N 是等价关系的证明简单,略),称 R_N 在 $\mathcal{B}(\mathcal{K})$ 上划分的任意个等价类都为 N 决定的等价类.

按定义 1,同一等价类中的概念内涵与 N 的交都是相同的.

例 1:设 $U = \{1,2,3,4,5,6,7,8\}, M = \{a,b,c,d,e,h,q\}$,背景 $\mathcal{K} = (U, M, I)$ 见表 1,概念格如图 1 所示,若 $N = \{a,c,q\}$,则 N 决定的等价类如图 2 所示.其中,每一个闭曲线中的概念是一个等价类,同一个等价类中的概念的内涵与 N 交的共同值写在了该等价类闭曲线的旁边.具体如下:等价类 $\{\#1, \#3, \#7\}$ 中的概念的内涵与 N 的交都是空,等价类 $\{\#2, \#5, \#6, \#8, \#9, \#10\}$ 中的概念的内涵与 N 的交都是 a ,等价类 $\{\#4\}$ 中的概念的内涵与 N 的交都是 c ,等价类 $\{\#11\}$ 中的概念的内涵与 N 的交都是 cq ,等价类 $\{\#12, \#13\}$ 中的概念的内涵与 N 的交都是 acq .

Table 1 A formal context

表 1 一个形式背景

	a	b	c	d	e	h	q
1	×						
2	×	×					
3			×				
4	×	×		×	×		
5			×		×	×	×
6	×	×		×	×	×	
7	×		×		×	×	×
8					×		

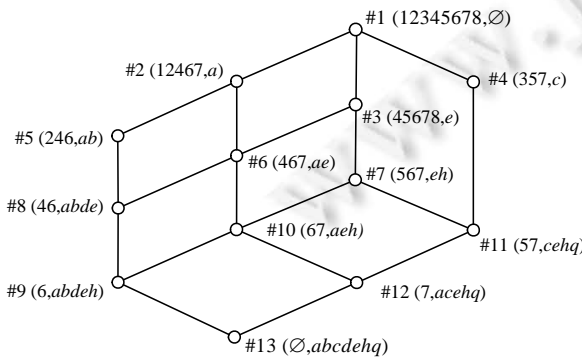


Fig.1 Concept lattice of formal context

图 1 形式背景的概念格

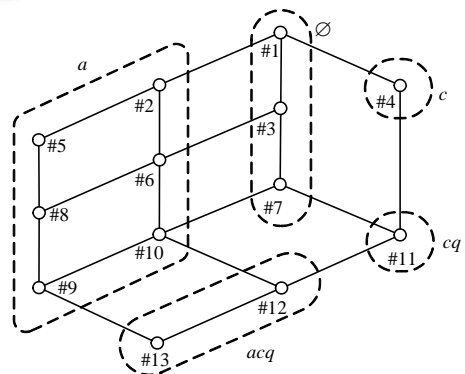


Fig.2 N determining equivalent classes

图 2 N 决定的等价类

我们将逐步证明: N 决定的每个等价类都有最大元素,而最大元素与子背景中的概念一一对应.若原背景中

有最大元素(A,B),则子背景中有概念(A,B∩N).这个事实先请见例 2.

例 2:对于例 1 中的 \mathcal{K} 及 N ,子背景 $\mathcal{K}_1=(U,N,I\cap U\times N)$ 见表 2,子背景的概念格如图 3 所示,原概念格中等价类的最大元素是#1,#2,#4,#11,#12 概念,如图 4 中的实心圆所示.

其中,#1 概念(12345678,∅)对应子背景中的(12345678,∅∩N)=(12345678,∅),#2 概念(12467,a)对应(12467,a∩N)=(12467,a),#4 概念(357,c)对应(357,c∩N)=(357,c),#11 概念(57,cehq)对应(57,cehq∩N)=(57,cq),#12 概念(7,acehq)对应(7,acehq∩N)=(7,acq).

Table 2 A subcontext

表 2 一个子背景

	a	c	q
1	×		
2	×		
3		×	
4	×		
5		×	×
6	×		
7	×	×	×
8			

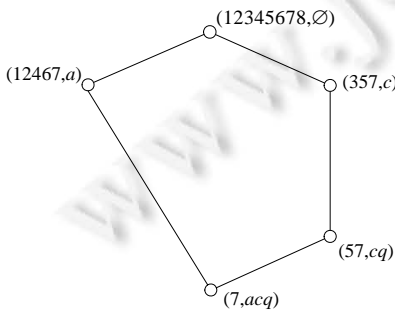


Fig.3 Concept lattice of subcontext

图 3 子背景的概念格

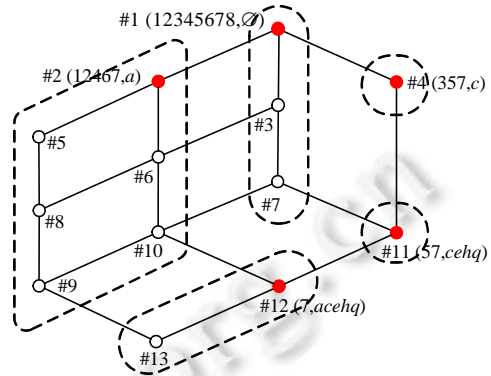


Fig.4 Maximal elements in equivalent classes

图 4 等价类的最大元素

下面逐步证明这个事实.

引理 1. 设 $\mathcal{K}=(U,M,I)$ 是一个背景, $N\subseteq M$, ω 是 N 决定的一个等价类,则 $\forall \omega \in \omega$,且 $\forall \omega$ 是 ω 的最大元素.

证明:设 $\omega=\{(A_t, B_t) | t \in T\}$ (T 是等价类 ω 的索引集),于是,对 $\forall t \in T, B_t \cap N$ 都是相同的,令 $B_t \cap N = P$.

由于 $\forall \omega = \bigvee \{(A_t, B_t) | t \in T\} = (g(\bigcap_{i \in T} B_i), \bigcap_{i \in T} B_i)$,而对每一个 $t \in T$ 都有 $B_t \cap N = P$,所以,

$$(\bigcap_{i \in T} B_i) \cap N = \bigcap_{i \in T} (B_i \cap N) = P.$$

由于 $(g(\bigcap_{i \in T_0} B_i), \bigcap_{i \in T_0} B_i)$ 的内涵与 N 的交也为 P ,所以 $(g(\bigcap_{i \in T_0} B_i), \bigcap_{i \in T_0} B_i) \in \omega$.由于 $\forall \omega$ 是 ω 的上界,所以 $\forall t \in T$ 都有 $(A_t, B_t) \leq \forall \omega$,所以, $\forall \omega$ 是 ω 的最大元素. □

引理 2. 设 $\mathcal{K}=(U,M,I)$ 是一个背景, $N\subseteq M, (A, B) \in \mathcal{B}(\mathcal{K})$,则:

$$f(g(B \cap N)) \cap N = B \cap N.$$

证明:因 $B \cap N \subseteq f(g(B \cap N))$,所以 $(B \cap N) \cap N \subseteq f(g(B \cap N)) \cap N$,即

$$B \cap N \subseteq f(g(B \cap N)) \cap N \tag{1}$$

另外, $B \cap N \subseteq B, f(g(B \cap N)) \subseteq f(g(B))$,而 (A, B) 是概念,所以 $f(g(B)) = B$,这样, $f(g(B \cap N)) \subseteq B$.由此:

$$f(g(B \cap N)) \cap N \subseteq B \cap N \tag{2}$$

由公式(1)、公式(2)知:

$$f(g(B \cap N)) \cap N = B \cap N. \quad \square$$

引理 3. 设 $\omega = \{(A_t, B_t) | t \in T\}$ 是 N 决定的一个等价类, 则对每个 $(A_t, B_t) \in \omega$, 都有:

$$\forall \omega = (g(B_t \cap N), f(g(B_t \cap N))).$$

证明: 由于 ω 是 N 决定的一个等价类, 所以对于 $\forall t \in T, B_t \cap N$ 都是相同的, 所以 $\forall t \in T, (g(B_t \cap N), f(g(B_t \cap N)))$ 都是相同的, 设为 (C, D) , 即, 对 $\forall t$ 都有:

$$g(B_t \cap N) = C \quad (3)$$

$$f(g(B_t \cap N)) = D \quad (4)$$

而且, 由公式(4)及引理 2 还知: $\forall t$, 都有:

$$D \cap N = f(g(B_t \cap N)) \cap N = B_t \cap N \quad (5)$$

所以 $(C, D) \in \omega$. 这样, $D \in \{B_t | t \in T\}$. 于是:

$$\bigcap_{t \in T} B_t \subseteq D \quad (6)$$

并且另一方面, 还由公式(5)得知: $\forall t, D \cap N \subseteq B_t$. 于是:

$$D \cap N \subseteq \bigcap_{t \in T} B_t \quad (7)$$

然而, 还由公式(5)及公式(3)得:

$$g(D \cap N) = g(B_t \cap N) = C = g(D) \quad (8)$$

这样, 由公式(6)~公式(8), 我们有:

$$g(D \cap N) = g(\bigcap_{t \in T} B_t) = g(D) = C \quad (9)$$

因 (C, D) 及 $(g(\bigcap_{t \in T} B_t), \bigcap_{t \in T} B_t)$ 都是概念, 所以:

$$D = f(C) = f(g(\bigcap_{t \in T} B_t)) = \bigcap_{t \in T} B_t \quad (10)$$

由公式(9)、公式(10)及公式(3)、公式(4)得:

$$\forall \omega = \{(A_t, B_t) | t \in T\} = (g(\bigcap_{t \in T} B_t), \bigcap_{t \in T} B_t) = (C, D) = (g(B_t \cap N), f(g(B_t \cap N))). \quad \square$$

定理 4. 设 $\mathcal{K} = (U, M, I)$ 是一个背景, $N \subseteq M, \mathcal{K}_1 = (U, N, I \cup U \times N)$ 是子背景, (A_1, B_1) 是 \mathcal{K}_1 的概念, 当且仅当存在 N 决定的某个等价类的最大元素 (A, B) 满足 $A_1 = A, B_1 = B \cap N$.

证明:

(当): 设 (A, B) 是 N 决定的某个等价类 ω 的最大元素, 令

$$A_1 = g_1(B \cap N), B_1 = f_1(g_1(B \cap N)).$$

首先, 由于 $B \cap N \subseteq N$, 所以 (A_1, B_1) 是 \mathcal{K}_1 的概念; 其次, 因为 (A, B) 是最大元素, 由引理 3 中的公式(8)得知, $g(B) = g(B \cap N)$; 又因 $g(B) = A$, 所以 $A = g(B \cap N)$. 因 $B \cap N \subseteq N$, 所以 $g(B \cap N) = g_1(B \cap N) = A_1$, 所以 $A = A_1$; 最后, $B_1 = f_1(g_1(B \cap N)) = f_1(g(B \cap N))$, 而 $f_1(g(B \cap N)) = f(g(B \cap N)) \cap N$, 所以 $B_1 = f(g(B \cap N)) \cap N$. 再由引理 3 中的公式(8)知 $g(B \cap N) = g(B)$, 所以 $B_1 = f(g(B)) \cap N$, 因 (A, B) 是概念, 所以 $f(g(B)) = B$, 所以 $B_1 = B \cap N$.

(仅当): 设 (A_1, B_1) 是 \mathcal{K}_1 的一个概念. 令 $A = A_1, B = f(A_1)$.

首先, $B \cap N = f(A_1) \cap N = f_1(A_1) = B_1$; 其次, 我们证明 (A, B) 一定是 \mathcal{K} 的一个概念, 因为一方面 $B = f(A_1) = f(A)$, 另一方面 $A = A_1 = g(B_1) = g(f(g(B_1)))$. 因 $B_1 \subseteq N$, 所以 $g(B_1) = g_1(B_1) = A_1$. 因此, $A = g(f(g(B_1))) = g(f(A_1)) = g(B)$, 所以 (A, B) 是一个概念; 最后, 证明 (A, B) 一定是一个最大元素. 因 $g(B \cap N) = g_1(B \cap N)$ 且 $B \cap N = f(A) \cap N = f_1(A_1) = B_1$, 所以 $g(D \cap N) = g_1(B \cap N) = g_1(B_1) = A_1 = A, f(g(B \cap N)) = f(A) = B$. 所以 $(g(B \cap N), f(g(B \cap N))) = (A, B)$, 由引理 1 及引理 3 得知, (A, B) 是一个最大元素. \square

由定理 4 求出 $\mathcal{B}(\mathcal{K})$ 的所有最大元素 (A, B) , 则所有 $(A, B \cap N)$ 就是子背景 \mathcal{K}_1 的所有概念. 我们称 (A, B) 与 $(A, B \cap N)$ 互为对应概念.

为了在原背景基础上得到子背景 \mathcal{K}_1 中概念的直接父子关系,做出子背景概念格的 Hasse 图,我们还有以下定理.

定理 5. 若 $(C_1, D_1), (C_2, D_2) \in \mathcal{B}(\mathcal{K}_1), (C_2, D_2) \succ (C_1, D_1), (C_1, D_1), (C_2, D_2)$ 在 $\mathcal{B}(\mathcal{K})$ 中对应的概念分别是 (A_1, B_1) 及 (A_2, B_2) , 则 (A_2, B_2) 是 (A_1, B_1) 某个直接父概念所在等价类的最大元素.

证明:因 (A_1, B_1) 及 (A_2, B_2) 分别是 $(C_1, D_1), (C_2, D_2)$ 的对应概念, 则 $A_1=C_1, A_2=C_2$. 由于 $(C_2, D_2) \succ (C_1, D_1)$, 所以 $C_2 \supset C_1$. 于是, $A_2 \supset A_1$. 这样, $(A_2, B_2) \succ (A_1, B_1)$, 因此有 (A_1, B_1) 的直接父概念 (A_3, B_3) 满足 $(A_2, B_2) \geq (A_3, B_3) \succ (A_1, B_1)$. 于是 $B_2 \subseteq B_3, B_2 \cap N \subseteq B_3 \cap N$ 及 $g(B_2 \cap N) \supseteq g(B_3 \cap N)$. 此时, 或 $g(B_2 \cap N) = g(B_3 \cap N)$, 或 $g(B_2 \cap N) \supset g(B_3 \cap N)$.

我们先证明 $g(B_2 \cap N) \supset g(B_3 \cap N)$ 不可能, 因为这将有 $(g(B_2 \cap N), f(g(B_2 \cap N))) \succ (g(B_3 \cap N), f(g(B_3 \cap N)))$, 而 (A_2, B_2) 是 (C_2, D_2) 的对应概念, 所以本身就是最大元素, 所以由引理 1 及引理 3 得知 $(A_2, B_2) = (g(B_2 \cap N), f(g(B_2 \cap N)))$, 于是 $(A_2, B_2) \succ (g(B_3 \cap N), f(g(B_3 \cap N)))$. 再注意到 $(A_3, B_3) \succ (A_1, B_1)$, 而 $(g(B_3 \cap N), f(g(B_3 \cap N)))$ 是 (A_3, B_3) 所在等价类的最大元素, 所以,

$$(g(B_3 \cap N), f(g(B_3 \cap N))) \geq (A_3, B_3) \succ (A_1, B_1).$$

这样我们有:

$$(A_2, B_2) \succ (g(B_3 \cap N), f(g(B_3 \cap N))) \geq (A_1, B_1), (A_2, B_2 \cap N) \succ (g(B_3 \cap N), f(g(B_3 \cap N))) \cap N \succ (A_1, B_1 \cap N),$$

即:

$$(C_2, D_2) \succ (g(B_3 \cap N), f(g(B_3 \cap N))) \cap N \succ (C_1, D_1).$$

因 $(g(B_3 \cap N), f(g(B_3 \cap N)))$ 是最大元素, 所以 $(g(B_3 \cap N), f(g(B_3 \cap N))) \cap N$ 是子背景的概念, 这与 (C_2, D_2) 是 (C_1, D_1) 的直接父概念矛盾, 于是只能是 $g(B_2 \cap N) = g(B_3 \cap N)$. 由此, 显然有 $f(g(B_2 \cap N)) \cap N = f(g(B_3 \cap N)) \cap N$. 而由引理 2 得知, 这时 $B_2 \cap N = B_3 \cap N$, 即 (A_2, B_2) 与 (A_3, B_3) 属于同一个等价类, 于是, (A_2, B_2) 就是 (A_1, B_1) 的直接父概念 (A_3, B_3) 所在等价类的最大元素. □

3 多属性减少的算法

本文采用自底向上的算法, 自顶向下的算法对偶可得.

我们用 \mathcal{M} 存放子背景的概念, 用 \mathcal{L} 存放子背景的概念格 Hasse 图的边, 用 \mathcal{S} 存放待处理的那些最大元素.

子背景零元概念对应的概念显然是原背景零元概念 $(g(M), M)$ 所在等价类的最大元素. 这个概念是

$$(g(M \cap N), f(g(M \cap N))) = (g(N), f(g(N))).$$

例如在例 2 中, 子背景零元概念的对应概念是

$$(g(acq), f(g(acq))) = (7, acehq).$$

我们从原背景这个最大元素 $(g(N), f(g(N)))$ 开始逐步向上处理.

若正在处理某个最大元素 (A, B) , 则 $(A, B \cap N)$ 就是子背景的一个概念, 送入 \mathcal{M} , 然后考察 (A, B) 的所有直接父概念 (C, D) . 显然, $(g(D \cap N), f(g(D \cap N)))$ 是 (C, D) 所在等价类的最大元素. 如果 $(g(D \cap N), f(g(D \cap N)))$ 不曾在 \mathcal{S} 中, 则加到 \mathcal{S} 尾. 由于 $(g(D \cap N), f(g(D \cap N))) \cap N$ 就是 $(g(D \cap N), f(g(D \cap N)))$ 对应的子背景的概念, 并且根据定理 5 得知, $(A, B \cap N)$ 的直接父概念就在由 (A, B) 的所有直接父概念 (C, D) 得出的 $(g(D \cap N), f(g(D \cap N))) \cap N$ 这些概念中. 例如在例 2 中, 如果正在处理的最大元素 (A, B) 是 #11 概念 $(57, cehq)$, 它对应的概念是子背景中的 $(57, cq)$, #11 概念有两个直接父概念: #7 概念 $(567, eh)$ 及 #4 概念 $(357, c)$, 而 #7 概念及 #4 概念所在等价类的最大元素分别是 #1 概念 $(12345678, \emptyset)$ 及 #4 概念 $(357, c)$, #1 概念及 #4 概念对应的子背景的概念是 $(12345678, \emptyset)$ 及 $(357, c)$, 于是, 子背景的 $(57, cq)$ 的直接父概念必在 $(12345678, \emptyset)$ 及 $(357, c)$ 之中.

将 (A, B) 的所有直接父概念 (C, D) 得到的的 $(g(D \cap N), f(g(D \cap N))) \cap N$ 送入临时变量 \mathcal{I} 中, 并进行“超集吸收”, 即: 在 \mathcal{I} 中若有概念 (I, J) 的外延 I 是概念 (E, F) 的外延 E 的超集, 则将 (I, J) 删掉. 例如在例 2 中, 将 $(12345678, \emptyset)$ 及 $(357, c)$ 送入变量 \mathcal{I} 中, 并因 $(12345678, \emptyset)$ 的外延 12345678 是 $(357, c)$ 的外延 357 的超集, 于是在 \mathcal{I} 中将 $(12345678,$

∅)删掉.

显然, \mathcal{P} 中存放的将是 $(A, B \cap N)$ 的全部直接父概念.将 \mathcal{P} 中的每个元素 (E, F) 都分别与 $(A, B \cap N)$ 形成二元组:

$$((E, F), (A, B \cap N)).$$

即是子背景的概念格 Hasse 图的一个边,将其送入 \mathcal{L} .处理完 \mathcal{S} 的一个元素,再处理下一个元素,最终所有最大元素都处理完毕时,输出 \mathcal{D} 及 $(\mathcal{D}, \mathcal{L})$ 就是子背景的全部概念及其 Hasse 图.

具体算法如下:

算法. 概念格多属性渐减式构造算法.

输入:背景 $\mathcal{K}=(U, M, I)$,概念格 $\mathcal{B}(\mathcal{K})$ 及减少的属性集 D ;

输出:子背景 $\mathcal{K}_1=(U, N, I \cap U \times N)$ 的全部概念 \mathcal{D} 及其 Hasse 图 $(\mathcal{D}, \mathcal{L})$,这里, $N=M-D$.

方法:

1. $N:=M-D$

$$\mathcal{S}:=\{(g(N), f(g(N)))\},$$

$$\mathcal{D}:=\emptyset$$

$$\mathcal{L}:=\emptyset$$

2. WHILE $\mathcal{S} \neq \emptyset$

3. $\mathcal{S}[0]$ 移到 (A, B)

4. $\mathcal{P}:=\emptyset$

5. $\mathcal{D}:=\mathcal{D} \cup \{(A, B \cap N)\}$

6. FOR (A, B) 的每个直接父概念 (C, D)

7. $P_1:=\{(g(D \cap N), f(g(D \cap N)))\}$

8. $P_2:=\{(g(D \cap N), f(g(D \cap N)) \cap N)\}$

9. IF P_1 不曾在 \mathcal{S} 中 THEN

10. P_1 加到 \mathcal{S} 尾

11. END IF

12. $\mathcal{P}:=\mathcal{P} \overset{\circ}{\cup} \{P_2\}$ //这里, $\overset{\circ}{\cup}$ 是超集吸收的并

13. END FOR

14. FOR \mathcal{P} 中的每个元素 (E, F)

15. $\mathcal{L}:=\mathcal{L} \cup \{((E, F), (A, B \cap N))\}$

16. END FOR

17. END WHILE

18. 输出 \mathcal{D}

19. 输出 $(\mathcal{D}, \mathcal{L})$

其中,步骤 1 是初始化;步骤 2~步骤 17 的 WHILE 循环是依次处理 \mathcal{S} 中的各个最大元素,其中,

- 步骤 3 是将 \mathcal{S} 的队首从 \mathcal{S} 移出放入 (A, B) ,即:将外延放入 A ,将内涵放入 B ;
- 步骤 4 是将 \mathcal{P} 清空,为收集 (A, B) 对应的子背景概念 $(A, B \cap N)$ 的直接父概念做准备;
- 步骤 5 是将 (A, B) 对应的子背景概念 $(A, B \cap N)$ 送入子背景的概念集合 \mathcal{D} ;

- 步骤 6~步骤 13 的 FOR 循环是考察(A,B)的每个直接父概念(C,D),查找(A,B∩N)的直接父概念,其中,
 - 步骤 7、步骤 8 是求(C,D)所在等价类的最大元素及其对应的子背景概念(g(D∩N),f(g(D∩N)))及(g(D∩N),f(g(D∩N))∩N),并分别放入 P₁ 及 P₂;
 - 步骤 9~步骤 11 是判定 P₁ 作为最大元素是否曾在S中,如果不在,则加到S队尾;
 - 步骤 12 是将 P₂ 按超集吸收原则并入P,记号 ∪ 表示超集吸收的并,即:求并集时,P中是 P₂ 的超集的元素都删掉,如果 P₂ 是P中某个元素的超集,则 P₂ 删掉.这样,步骤 6~步骤 13 的 FOR 循环结束时,P中将是(A,B∩N)的全部直接父概念;
- 于是,在步骤 14~步骤 16 的 FOR 循环中,将它们与(A,B∩N)形成边送入L.

整个 WHILE 循环结束后, D将是子背景的全部概念,(D,L)将是子背景概念格的 Hasse 图.

定理 6. 概念格多属性渐减式构造算法的时间复杂度是 O(c×m×u).这里, c=|B(K)|是所有概念的个数, u=|U|是对象的个数,m=|M|是属性的个数.

证明:步骤 2~步骤 17 的 WHILE 循环的次数是原背景最大元素的个数(也即子背景概念的个数),显然小于原背景所有概念的个数 |B(K)|.步骤 6~步骤 13 的 FOR 循环的次数是(A,B)的直接父概念的个数,按文献[20],该数小于|B|,显然,最不利的情况是小于|M|.在此循环中,计算 P₁ 需要|D∩N|次交运算及|g(D∩N)|次交运算,显然,最不利的情况是小于|M|+|U|,在 P₁ 基础上计算 P₂ 需要 1 次交运算.步骤 12 在求并运算的同时还要做|P|次子集隶属关系的判定,|P|小于(A,B)的直接父概念的个数,显然,最不利的情况是小于|M|.同理,步骤 14~步骤 16 的 FOR 循环的次数最不利的情况是小于|M|,于是,整个算法的运算次数是

$$O(|B(K)| \times (|M| \times (|M| + |U| + |M|) + |M|)).$$

由于在实际问题中总是|M|<|U|,所以:

$$O(|M| + |U| + |M|) = O(|U|).$$

于是,

$$O(|M| \times (|M| + |U| + |M|)) = O(|M| \times |U|), O(|M| \times (|M| + |U| + |M|) + |M|) = O(|M| \times |U|).$$

这样,整个算法的时间复杂度是

$$O(|B(K)| \times |M| \times |U|) = O(c \times m \times u).$$

□

4 验证

我们以例 1 中的背景K及 D=bdeh 来对算法 1 进行验证.

初始化

$$N=M-D=acq, S=\{(7,acehq)\}, \text{即}, S=\{\#12\}, D=\emptyset, L=\emptyset$$

WHILE S≠∅

#12 概念

$$(A,B)=(7,acehq), (S=\emptyset), P=\emptyset, D=\{(7,acq)\}$$

FOR #12 的每个直接父概念

#10 概念

$$P_1=(12467,a), P_2=(12467,a), S=\{(12467,a)\} \text{即} S=\{\#2\}, P=\{(12467,a)\}$$

#11 概念

$$P_1=(57,cehq), P_2=(57,cq), S=\{(12467,a), (57,cehq)\}, \text{即}, S=\{\#2, \#11\}, P=\{(12467,a), (57,cq)\}$$

ENDFOR

FOR \mathcal{P} 中的每个元素(E,F):

$$\mathcal{L}=\{((12467,a),(7,acq)),((57,cq),(7,acq))\}.$$

ENDFOR

#2 概念

$$(A,B)=(12467,a),(S=\{(57,cehq)\}),\text{即},(S=\{\#11\}),\mathcal{P}=\emptyset,\mathcal{D}=\{(7,acq),(12467,a)\}$$

FOR #2 的每个直接父概念

#1 概念

$$P_1=(12345678,\emptyset),P_2=(12345678,\emptyset),S=\{(57,cehq),(12345678,\emptyset)\}\text{即 }S=\{\#11,\#1\}$$

$$\mathcal{P}=\{(12345678,\emptyset)\}$$

ENDFOR

FOR \mathcal{P} 中的每个元素(E,F):

$$\mathcal{L}=\{((12467,a),(7,acq)),((57,cq),(7,acq)),((12345678,\emptyset),(12467,a))\}.$$

ENDFOR

#11 概念

$$(A,B)=(57,cehq),(S=\{(12345678,\emptyset)\}),\text{即},(S=\{\#1\}),\mathcal{P}=\emptyset,\mathcal{D}=\{(7,acq),(12467,a),(57,cq)\}$$

FOR #11 的每个直接父概念

#7 概念

$$P_1=(12345678,\emptyset),P_2=(12345678,\emptyset),S=\{(12345678,\emptyset)\}\text{即 }S=\{\#1\},\mathcal{P}=\{(12345678,\emptyset)\}$$

#4 概念

$$P_1=(357,c),P_2=(357,c),S=\{(12345678,\emptyset),(357,c)\}\text{即 }S=\{\#1,\#4\},\mathcal{P}=\{(357,c)\} \quad //\text{超级吸收}$$

ENDFOR

FOR \mathcal{P} 中的每个元素(E,F):

$$\mathcal{L}=\{((12467,a),(7,acq)),((57,cq),(7,acq)),((12345678,\emptyset),(12467,a)),((357,c),(57,cq))\}.$$

ENDFOR

#1 概念

$$(A,B)=(12345678,\emptyset),(S=\{(357,c)\}),\text{即},(S=\{\#4\}),\mathcal{P}=\emptyset,$$

$$\mathcal{D}=\{(7,acq),(12467,a),(57,cq),(12345678,\emptyset)\}.$$

FOR #1 的每个直接父概念

$$\text{无},P_1=nil,P_2=nil,S=\{(357,c)\},\text{即},S=\{\#4\},\mathcal{P}=\emptyset$$

ENDFOR

FOR \mathcal{P} 中的每个元素(E,F):

$$\mathcal{L}=\{((12467,a),(7,acq)),((57,cq),(7,acq)),((12345678,\emptyset),(12467,a)),((357,c),(57,cq))\}.$$

ENDFOR

#4 概念

$$(A,B)=(357,c),(S=\emptyset),\mathcal{P}=\emptyset,\mathcal{D}=\{(7,acq),(12467,a),(57,cq),(12345678,\emptyset),(357,c)\}$$

FOR #4 的每个直接父概念

#1 概念

$$P_1=(12345678,\emptyset), P_2=(12345678,\emptyset), S=\emptyset, \mathcal{P}=\{(12345678,\emptyset)\}$$

ENDFOR

FOR \mathcal{P} 中的每个元素 (E,F) :

$$\mathcal{L}=\{((12467,a),(7,acq)),((57,cq),(7,acq)),((12345678,\emptyset),(12467,a)),((357,c),(57,cq)),((12345678,\emptyset),(357,c))\}.$$

ENDFOR

ENDWHILE

我们看到:最终的结果 \mathcal{D} 及 \mathcal{L} 与例 2 中的子背景直接求得的概念及概念格完全一致,而且删除了 3 个属性算法也只需执行一次.

5 后 记

在减少属性 m 的渐减式构造算法中,对于原背景的概念 (A,B) 曾定义^[20]:

- (1) 若 $m \notin B$,则称该概念为保留概念;
- (2) 若 $m \in B$,但 $B-\{m\}$ 不再是任何概念的内涵,则称该概念为更新概念;
- (3) 若 $m \in B$,而 $B-\{m\}$ 是某个概念 (C,D) 的内涵,则称该概念为删除概念,称 (C,D) 为该概念的删除基.

这 3 个定义可根据以下原理转化为另一种等价形式.由于 $B-\{m\} \subseteq f(g(B-\{m\})) \subseteq f(g(B))$,而 $f(g(B))=B$,所以 $B-\{m\} \subseteq f(g(B-\{m\})) \subseteq B$,但 $B-\{m\}$ 与 B 最多只差一个元素,所以,或:

$$f(g(B-\{m\}))=B-\{m\} \tag{11}$$

或:

$$f(g(B-\{m\}))=B \tag{12}$$

对于以上定义的第 2 种情况,即更新概念的情况,公式(11)将不成立,因为 $f(g(B-\{m\}))=B-\{m\}$ 意味着 $B-\{m\}$ 是内涵,与 $B-\{m\}$ 不再是任何概念的内涵矛盾,所以这时只能是公式(12) $f(g(B-\{m\}))=B$ 成立,于是这时:

$$g(B-\{m\})=g(f(g(B-\{m\})))=g(B)=A.$$

对于以上定义的第 3 种情况,即删除概念的情况,显然,公式(11) $f(g(B-\{m\}))=B-\{m\}$ 成立,即,这时 $D=B-\{m\}=f(g(B-\{m\}))$ 及 $C=g(B-\{m\})$.而且公式(12) $f(g(B-\{m\}))=B$ 将不成立,但 $f(A)=B$,所以 $g(B-\{m\}) \neq A$.这样,前面的 3 种情况等价于:

- (1) 若 $m \notin B$,则称该概念为保留概念;
- (2) 若 $m \in B$,但 $g(B-\{m\})=A$,则称该概念为更新概念;
- (3) 若 $m \in B$,而 $g(B-\{m\}) \neq A$,则称该概念为删除概念,且删除基为 $(g(B-\{m\}),f(g(B-\{m\})))$.

与减少一个属性 m 对照,我们这里减少的是多个属性,即,减少的是一个属性集合 D .

于是, $m \notin B$ 对应 $D \cap B = \emptyset, g(B-\{m\})=A$ 对应 $g(B-D)=A$.若令 $N=M-D$,则对应 $g(B \cap N)=A$.

于是,与这 3 个等价定义对照,在这里是对于原背景的一个概念 (A,B) :

- (1) 保留概念是 $D \cap B = \emptyset$ (或 $B \subseteq N$);
- (2) 更新概念是 $D \cap B \neq \emptyset$ (或 $B \not\subseteq N$),但 $g(B \cap N)=A$;
- (3) 删除概念是 $D \cap B \neq \emptyset$ (或 $B \not\subseteq N$),且 $g(B \cap N) \neq A$,且删除基为 $(g(B \cap N),f(g(B \cap N)))$.

另外,对于第 1 种情况,显然 $g(B \cap N)=g(B)$,于是也有 $g(B \cap N)=A$.这样,对于第 1 种、第 2 种这两种情况都有 $A=g(B \cap N)$,进而 $B=f(g(B \cap N))$.这样, $(A,B)=(g(B \cap N),f(g(B \cap N)))$.于是,由定理 4 知, (A,B) 是最大元素.所以,我们这里的最大元素是保留概念及更新概念, $B \subseteq N$ 时是保留概念, $B \not\subseteq N$ 时是更新概念,不是最大元素的概念都是删除概念.另外,由定理 4 得知,删除基就是删除概念所在等价类的最大元素.

例如,对于例 1 中的 \mathcal{K} 及 N ,概念格如图 1 所示,概念格的等价类如图 2 所示,最大元素如图 4 所示.其中,#1, #2,#4 是保留概念,#11,#12 概念是更新概念,其他概念都是删除概念.#5,#6,#8,#9,#10 的删除基是#2 概念,#3,#7

的删除基是#1 概念,#13 的删除基是#12 概念。

减少一个属性时,一个删除基只有一个删除概念;减少多个属性时,一个删除基有多个删除概念.即:每个等价类中最大元素是删除基,其他众多元素都是删除概念.例如,例 2 中的删除基#2 概念有多达 5 个删除概念:#5, #6, #8, #9, #10.这样,减少多个属性时比减少一个属性时的删除概念要多许多.由于要删除的概念多,所以不宜采用逐个删除概念、逐个删除 Hasse 图边的方法.我们这里是采用在原概念格的基础上获取各个最大元素(其对应概念存入 \mathcal{M}),并进而获取 Hasse 图边(存入 \mathcal{L})的方法,大大缩小了对原背景概念的考察范围.例如,用本文的算法求例 2 中的子背景的概念格时,#5, #6, #8, #9 概念以及#3 概念都不会被涉及、不会被考察,并且因它们没被送入 \mathcal{M} 而自动被删除;同样,很多 Hasse 图的边因不会被送入 \mathcal{L} 而自动被删除.由于在这种算法中只需要区分最大元素及非最大元素,而区分保留概念、更新概念、删除概念的意义不太大,所以在本文算法中没做这方面的区分.本文给出的概念格多属性渐减式构造算法与文献[20]中的概念格减少单属性渐减式构造算法具有相同的时间复杂度,然而减少多个属性时,文献[20]的算法需要执行多次,本文算法只需执行一次.

References:

- [1] Wille R. Restructuring lattice theory: An approach based on hierarchies of concepts. Rival I. Ordered Sets. Dordrecht-Boston: Reidel, 1982. 445–470.
- [2] Jiang P, Ren SB, Lin J. Using formal concept analysis for software engineering. Computer Technology and Development, 2008, 18(4):127–129, 213 (in Chinese with English abstract).
- [3] Chen X, Wu Y. Mining association rules of geographic information system based on concept lattice. Journal of Computer Application, 2011,31(3):686–689 (in Chinese with English abstract).
- [4] Xie LM, Li JM. Application research of concept lattice in intrusion detection. Computer Engineering and Design, 2010,31(5): 979–981, 998 (in Chinese with English abstract).
- [5] Xi HD, Yan H. The application of concept lattice in Web-log mining. Applies Computer Systems, 2006,15(9):21–24 (in Chinese with English abstract).
- [6] Xu T, Xu B. Application of concept lattice of customers group clustering. Modern Computer, 2008,285(6):70–73 (in Chinese with English abstract).
- [7] He LL, Bai HT, Zhang JC. Aspect mining using concept analysis. Computer Science, 2005,32(11):155–157 (in Chinese with English abstract).
- [8] Liu L, Zhang Y, Li M, Yang DS. Study of dynamic knowledge bases and concept lattices applied to intelligence disease diagnosis. Computer Engineering and Applications, 2007,43(28):233–236 (in Chinese with English abstract).
- [9] Yin JH, Li GQ, Chen Y, Deng M. A spatial clustering method based on concept lattices. Applies Computer Systems, 2011,20(6): 103–108 (in Chinese with English abstract).
- [10] Bal M, Sever H, Kalıpsız O. Modeling the symptom-disease relationship by using rough set theory and FormalConcept analysis. Proc. of the World Academy of Science, Engineering and Technology, 2007,26(12):517–521.
- [11] Shen L, Wang LM. Analysis of stability-based concept lattice for mining Folksonomy. Computer Engineering and Design, 2012, 33(3):1213–1217 (in Chinese with English abstract).
- [12] He YQ, Li JF. Permission management of RBAC based on concept lattice. Journal of Henan University (Natural Science), 2011, 41(3):1832–1844 (in Chinese with English abstract).
- [13] Xu JQ, Peng X, Zhao WY. An evolution analysis method based on fuzzy concept lattice and source code analysis. Chinese Journal of Computers, 2009,32(9):308–311 (in Chinese with English abstract).
- [14] Li LF, Zhang DX. The application of concept lattice theory in the reduction of the proposition set in two-valued propositional logic. Acta Electronica Sinica, 2007,35(8):1538–1542 (in Chinese with English abstract).
- [15] Godin R, Missaoui T, Aloui H. An incremental concept formation algorithm based on Galois (concept) lattices. Computational Intelligence, 1995,11(2):246–267.

- [16] van der Merwe D, Obiedkov S, Kourie D. AddIntent: A new incremental algorithm for constructing concept lattices. In: Proc. of the 2nd Int'l Conf. on Formal Concept Analysis. LNCS 2961, Berlin: Springer-Verlag, 2004. 372–385.
- [17] Zhi HL, Zhi DJ. Theory and algorithm of concept lattice incremental construction based on attributes. Computer Engineering and Application, 2012,48(26):17–21 (in Chinese with English abstract).
- [18] Andrews S. In-Close, a fast algorithm for computing formal concepts. In: Proc. of the 17th Int'l Conf. on Conceptual Structures (ICCS). Moscow: CEURWS, 2009. 1–14.
- [19] Liu ZT, Qiang Y, Zhou W, Li X, Huang ML. A fuzzy concept lattice model and its incremental construction algorithm. Chinese Journal of Computers, 2007,30(2):184–188 (in Chinese with English abstract).
- [20] Zhang L, Zhang HL, Yin LH, Han DJ. Theory and algorithms of attribute decrement for concept lattice. Journal of Computer Research and Development, 2013,50(2):248–259 (in Chinese with English abstract).

附中文参考文献:

- [2] 蒋平,任胜兵,林鹏.形式概念分析在软件工程中的应用.计算机技术与发展,2008,18(4):127–129,213.
- [3] 陈湘,吴跃.基于概念格挖掘 GIS 中的关联规则.计算机应用,2011,31(3):686–689.
- [4] 谢丽明,李建民.概念格在入侵检测中的应用.计算机工程与设计,2010,31(5):979–981,998.
- [5] 习慧单,严晖.概念格在 Web 日志挖掘中的应用.计算机系统应用,2006,15(9):21–24.
- [6] 许涛,徐彬.概念格在客户群聚类中的应用.现代计算机,2008,285(6):70–73.
- [7] 何丽莉,白洪涛,张家晨.用概念格方法挖掘 Aspect.计算机科学,2005,32(11):155–157.
- [8] 刘玲,张永,李明,杨德三.动态知识库和概念格在病症智能诊断中的应用.计算机工程与应用,2007,43(28)233–236.
- [9] 殷俊华,李光强,陈翼,邓敏.基于概念格的空间聚类方法.计算机系统应用,2011,20(6):103–108.
- [11] 申乐,王黎明.概念格稳定性分析及其在 Folksonomy 中的应用.计算机工程与设计,2012,33(3):1213–121.
- [12] 何云强,李建凤.RBAC 中基于概念格的权限管理研究.河南大学学报:自然科学版,2011,41(3):1832–1844.
- [13] 许佳卿,彭鑫,赵文耘.一种基于模糊概念格和代码分析的软件演化分析方法.计算机学报,2009,32(9):308–311.
- [14] 李立峰,张东晓.概念格在二值命题逻辑命题集约简中的应用.电子学报,2007,35(8):1538–1542.
- [17] 智慧来,智东杰.基于属性的概念格渐进式构造原理与算法.计算机工程与应用,2012,48(26)17–21.
- [19] 刘宗田,强宇,周文,李旭,黄美丽.一种模糊概念格模型及其渐进式构造算法.计算机学报,2007,30(2):184–188.
- [20] 张磊,张宏莉,殷丽华,韩道军.概念格的属性渐减原理与算法研究.计算机研究与发展,2013,50(2):248–259.



马垣(1941—),男,北京人,教授,主要研究领域为数据库理论,粗糙集理论,形式概念分析.



马文胜(1971—),男,副教授,主要研究领域为虚拟现实,地理信息系统,形式概念分析.