

控制与决策

Control and Decision

一种基于约束格维护概念模型一致性的方法

刘勇, 胡宇鹏, 李学庆

引用本文:

刘勇, 胡宇鹏, 李学庆. 一种基于约束格维护概念模型一致性的方法[J]. *控制与决策*, 2020, 35(2): 367–374.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2018.0414>

您可能感兴趣的其他文章

Articles you may be interested in

[基于积分滑模控制的非理想变时滞神经网络有限时间同步](#)

Finite-time synchronization of nonidentical neural networks with time-varying delay based on integral sliding mode control

控制与决策. 2019, 34(7): 1559–1564 <https://doi.org/10.13195/j.kzyjc.2017.1689>

[2-DOF绳索驱动并联机构轨迹跟踪控制](#)

Trajectory tracking control of a 2-DOF cable-driven parallel mechanism

控制与决策. 2019, 34(5): 1103–1108 <https://doi.org/10.13195/j.kzyjc.2017.1460>

[用于图像分割的局部区域一致性流形约束MRF模型](#)

Local region consistency manifold constrained MRF model for image segmentation

控制与决策. 2019, 34(5): 997–1003 <https://doi.org/10.13195/j.kzyjc.2017.1453>

[基于邻域粗糙互信息熵的非单调性属性约简](#)

Non-monotonic attribute reduction based on neighborhood rough mutual information entropy

控制与决策. 2019, 34(2): 353–361 <https://doi.org/10.13195/j.kzyjc.2017.1065>

[基于RVMD的牵引电机转子初期断条故障监测](#)

Monitoring of incipient rotor bars broken fault in traction motors based on RVMD method

控制与决策. 2018, 33(3): 497–502 <https://doi.org/10.13195/j.kzyjc.2017.0090>

[自动导引车系统防碰撞及死锁的形式化控制方法](#)

Collision prevention and deadlock formal control method for automated guided vehicle system

控制与决策. 2017, 32(9): 1628–1634 <https://doi.org/10.13195/j.kzyjc.2016.0613>

[基于Pythagorean不确定语言的扩展VIKOR多属性群决策方法](#)

Extended VIKOR method for multi-attribute group decision making based on Pythagorean uncertain linguistic information

控制与决策. 2017, 32(12): 2145–2152 <https://doi.org/10.13195/j.kzyjc.2016.1535>

[一种结合空间信息和稀疏字典优化的目标跟踪算法](#)

An object tracking algorithm combining spatial information and sparse dictionary optimization

控制与决策. 2016, 31(12): 2170–2176 <https://doi.org/10.13195/j.kzyjc.2015.1489>

一种基于约束格维护概念模型一致性的方法

刘 勇^{1,2}, 胡宇鹏¹, 李学庆^{1†}

(1. 山东大学 计算机科学与技术学院, 济南 250101; 2. 昌吉学院 计算机工程系, 新疆 昌吉 831100)

摘 要: 针对以形式概念分析理论为基础的概念建模过程中知识表示存在差距的问题, 提出一种整合专家知识到概念格结构中的形式化模型. 首先, 将一组属性依赖与概念格提供的一系列蕴涵对齐, 对原始格进行修订, 然后, 通过使用外延投影建立约束格来提供变化轨迹, 并在此基础上, 提出基于形式概念分析约束格理论弥补这一差距的建模方法, 以维护概念模型的一致性. 该方法不仅提供了领域专家修订概念模型的途径, 还保留了原始格和最终约束格之间的变化轨迹. 通过这些变化, 专家可以访问实践中的概念如何与数据自动发布的概念相关联. 最后, 结合示例对基于约束格维护概念模型一致性方法的有效性进行了验证.

关键词: 概念建模; 形式概念分析; 概念格; 约束格; 投影; 依赖关系

中图分类号: TP273

文献标志码: A

A method of maintaining consistency of conceptual model with constrained lattice

LIU Yong^{1,2}, HU Yu-peng¹, LI Xue-qing^{1†}

(1. School of Computer Science and Technology, Shandong University, Jinan 250101, China; 2. Department of Computer Engineering, Changji University, Changji 831100, China)

Abstract: In order to make up for the gap in knowledge representation in the process of concept modeling based on the formal concept analysis theory, a formal model of integrating expert knowledge into concept lattice structure is proposed. Firstly, the original lattice is revised by aligning a set of attribute dependencies with a set of implications provided by the concept lattice. Then, the constraint lattice is created by using the extent projection to provide change traces. On this basis, this paper proposes a modeling method based on the formal concept analysis constraint lattice theory to bridge this gap, so as to maintain the consistency of the conceptual model. This method not only provides a way for domain experts to revise the conceptual model, but also retains the traces of changes between the original lattice and the final constraint lattice. Through these changes, experts can track how the concept of the real world is associated with the concept of automatic derivation. Finally, examples are given to illustrate the effectiveness of the method to maintain the consistency of the conceptual model based on constraint lattice.

Keywords: conceptual modeling; formal concept analysis; concept lattice; constraint lattice; projection; dependency relationship

0 引 言

形式概念分析 (Formal concept analysis, FCA)^[1] 是一种分类方法, 有助于构建本体的概念化步骤^[2]. FCA 从数据中以一组属性蕴涵或概念格的形式抽取出概念格. 概念格是概念层次的形式化模型, 形式概念和概念格的逻辑结构可以有效地支持人类推理^[3]. 然而, 构建知识模型是一个认知过程, 并不遵守严格的形式规则, 因为领域专家可能以不同于数据表示的方式来理解特定领域. 因此, 基于概念格的表示模型与领域专家所理解的表示模型之间通常存在差

距. 为了弥补这一差距, 文献 [4-6] 尝试将专家知识以属性之间的依赖关系集成到格结构中并形式化. 在这些方法中, 属性依赖作为约束条件, 向专家提供满足约束条件的形式化概念, 忽略不满足约束条件的形式化概念, 使得形式概念的结构更加易于理解. 这种格结构在支持人类推理方面是有效的.

Vladimir 等^[7-8] 尝试在数据挖掘方法提取的信息单元中整合领域专家需求的迭代框架. 数据挖掘方法被用来从数据中提取信息单元, 并将这些信息单元呈现给用户; 接下来, 系统从用户的反馈中再学

收稿日期: 2018-04-04; 修回日期: 2018-09-20.

基金项目: 山东省重点研发计划项目 (2015GGX101009).

†通讯作者. E-mail: xqli@sdu.edu.cn.

习,得到关于用户兴趣性的模式排名.该框架允许用户在数据挖掘方法提取的模式中选择其感兴趣的模式.但是,选择提取的模式并不总是领域专家的最佳选择.如果仅仅基于从数据中提取的信息,则在提取的模式^[7]或提取的规则^[8]中进行选择,是不足以弥补数据挖掘方法建立的概念模型与概念获取过程领域专家的知识模型之间差距的.

以概念格中属性之间依赖关系的形式表示概念模型,在国内已有学者进行了相关研究.文献[9-11]将属性探索扩展到包括背景影响,即专家已知的影响是有效的.在这些方法中,研究者相信原始数据,领域专家必须提供新的对象作为反例.为了处理这样的情况,文献[10]通过从数据中提取的形势背景和由领域专家提供的属性层次结构来构建概念层次结构.文献[11]提出了类似的方法,作者引入了属性依赖关于单个属性的概念,并提出了一种从递增算法改编的用于计算约束条件的算法.

本文引入并优化了将专家约束集成到概念格的形式化方法,将一组给定的属性依赖与概念格提供的一系列蕴涵“对齐”,实现在原始格中的修订.在不改变数据的情况下建立约束格,通过在格上使用外延投影来提供变化的轨迹,并评估实践中的概念与数据自动发布的概念的相关性.

1 属性蕴涵与依赖关系

下面介绍一些重要的定义.首先,引入属性蕴涵和依赖关系;然后,用概念格来描述投影.形式化过程将使用文献[1]中的FCA符号定义,其中形式背景 (G, M, I) 由一组对象 G ,一组属性 M 和一个关联集 $I \subseteq G \times M$ 组成.表1给出了形式背景的一个示例.

表1 形式背景示例

	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9
g_1 : bear	×						×	×	
g_2 : carp		×			×			×	×
g_3 : chicken	×	×	×	×		×		×	
g_4 : crab		×							×
g_5 : dolphin			×		×		×	×	×
g_6 : honeybee		×	×	×					
g_7 : penguin	×	×		×		×		×	×
g_8 : wallaby	×						×	×	

表1中的行标题表示对象集 G ,列标题表示属性集 M ,符号 \times 表示对象所具备的属性.为便于表述,符号化了对对象及属性名称,其中 m_1 : has-two-legs, m_2 : lays-eggs, m_3 : can-fly, m_4 : has-wings, m_5 : has-fins, m_6 : has-feathers, m_7 : has-milk, m_8 : has-backbone, m_9 : lives-in-water.

1.1 属性蕴涵

在形式背景中的蕴涵表示数据中存在的属性之间的依赖关系.给出蕴涵 $x \rightarrow y$,可以理解为“所有有 x 的对象也有 y ”.属性蕴涵可以直接从形式背景中读出,如定义1所述.

定义1(属性蕴涵^[1]) 在形式背景 (G, M, I) 中的属性集 $X, Y \subseteq M$ 之间的蕴涵由 $X \rightarrow Y$ 表示,其中具有来自 X 的所有属性的每个对象也具有来自 Y 的所有属性,即 $X' \subseteq Y'$.

命题1 如果 $Y \subseteq X''$ (其中 X' 是具有 X 中的所有属性的对象的集合, X'' 代表 (X')),则属性 $X, Y \subseteq M$ 在 (G, M, I) 中成立的蕴涵 $X \rightarrow Y$ 是在 X' 中所有对象共有的属性的集合,在所有概念内涵的集合中也自动成立.

命题2 如果 $X \rightarrow m$ 对于每个 $m \in Y$ 成立,则在属性集合 $X, Y \subseteq M$ 之间的蕴涵 $X \rightarrow Y$ 在格中成立.如果 $(m', m'') \geq (X', X'')$,其中 X' 是 X 中具有所有属性的对象集合, X'' 代表 (X') 是 X' 中所有对象共有的属性集合, m' 是具有属性 m 的对象集合, m'' 代表 $(m)'$ 是 m' 中所有对象共有的属性集合,则 $X \rightarrow m$ 成立, (m', m'') 是 m 的属性概念.

根据定义1,归因于命题1和命题2,可以在形式背景和概念格中验证属性蕴涵.以表1所示的形式背景为例,蕴涵 $m_6 \rightarrow m_8$ 成立,因为具有属性 m_6 的每个对象也具有属性 m_8 ;蕴涵 $\{m_5\} \rightarrow \{m_8, m_9\}$ 成立,因为每个具有属性 m_5 的对象也具有来自属性集 $\{m_8, m_9\}$ 的所有属性.

1.2 属性依赖

属性依赖与属性蕴涵不同,属性依赖不会从数据中产生.属性依赖是专家期望作为属性蕴涵存在的依赖关系.属性依赖关系代表领域专家根据属性对对象进行分类时的知识,使用重要属性形成上层概念,用较普遍的属性形成下层概念.属性依赖由Baixeries等^[12-13]先后扩展,作为属性之间依赖关系的形式化方法.

定义2(属性依赖^[12]) 属性集合 x 和 y 之间的依赖关系的形式是 $x \prec y$,其中属性集合 x 不如属性集合 y 重要,并且在没有 y 的情况下 x 的存在没有意义.

定义3 当 $x, y \in B$ 时,形式概念 (A, B) 满足属性 x 和 y 之间的属性依赖关系 $x \prec y$.

1) 由属性依赖 $x \prec y$ 约束的概念格 \mathcal{L} 是来自满足 $x \prec y$ 的所有形式概念的集合;

2) 由一组属性依赖 D 约束的概念格 \mathcal{L} 是来自格 \mathcal{L} 的所有形式概念的集合,其满足 D 中的所有属性依赖关系.

这里两个集合都是原始格 \mathcal{L} 的部分有序子集,将

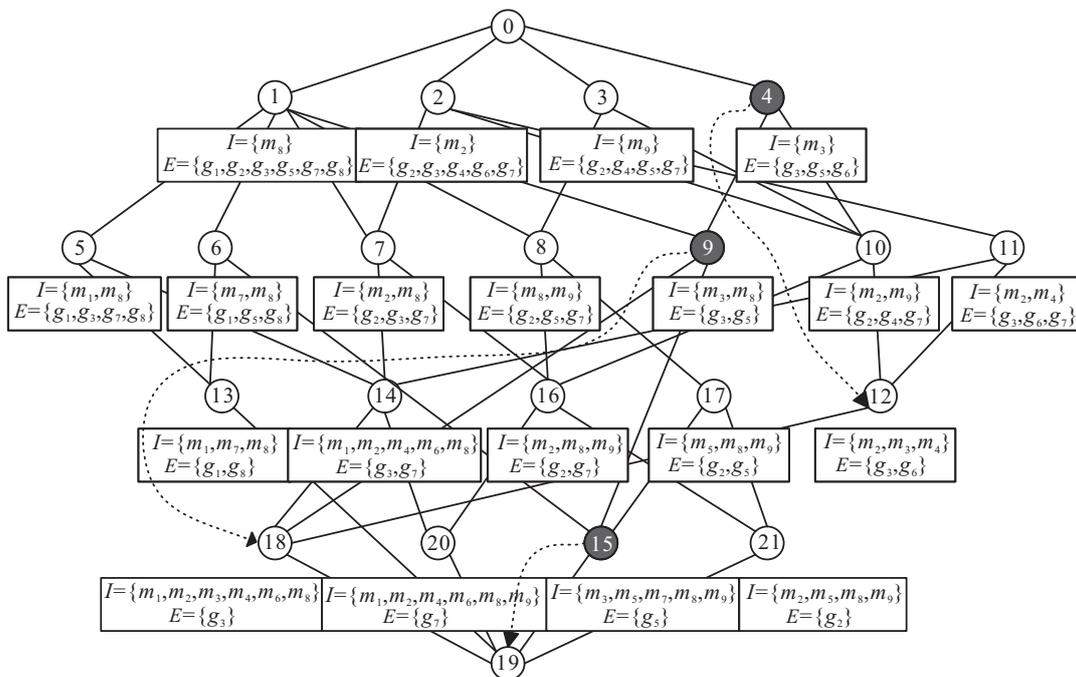


图1 由表1的形式背景所构建的概念格

这些集合称为格 \mathcal{L} 的约束条件,其中每个依赖项 $x \prec y$ (或者与依赖关系集 D 相关).为说明定义2和定义3,分析表1中的形式背景,相应的概念格如图1所示.

假设领域专家以属性依赖 $m_3 \prec m_4$ 的形式提供了领域的概念,由属性依赖 $m_3 \prec m_4$ 约束的概念格 \mathcal{L} 的偏序集是图1中所示格的所有形式概念的集合.图1中空圈标记的形式概念的集合满足这一属性依赖关系,其中不满足属性依赖 $m_3 \prec m_4$ 的概念 C_4 、 C_9 和 C_{15} 将被忽略.

属性蕴涵和属性依赖表示属性之间的依赖关系.属性蕴涵来自于数据,而属性依赖则对领域专家的知识进行编码.领域专家或许会提出与数据不同的内容,因此,属性依赖可能不符合从数据中提取的任何含义,从而产生基于概念格的表示模型与领域专家表示模型之间的差距.

2 生成约束格的投影

2.1 外延投影

投影是一种数学函数,可以通过将形式背景映射到外延和内涵来简化概念格,包括外延投影和内涵投影^[13-14].最初,投影用于简化模式结构概念描述^[15].在序理论中,投影被称为核运算符或算子,本文使用外延投影.

定义4(外延投影) 如果对于 \mathcal{L} 的每一对 (A_1, A_2) , ψ 是 \mathcal{L} 的外延投影,则有:

- 1) 如果 $A_1 \subseteq A_2$,则 $\psi A_1 \subseteq \psi A_2$ (单调性);
- 2) $\psi(A_1) \subseteq \psi A_1$ (收缩性);
- 3) $\psi(\psi(A_1)) = \psi(A_1)$ (幂等性).

设 \mathcal{L} 是一个格, ψ 是 \mathcal{L} 的外延投影,则 \mathcal{L} 中的外延集合 E 可以分为两个集合,即

$$E = \{e \in E \mid \psi(e) = e\} \cup \{e \in E \mid \psi(e) \neq e\},$$

集合 $\{e \in E \mid \psi(e) = e\}$ 称为 ψ 的不动点.

命题3 设 \mathcal{L} 是一个格, ψ 是 \mathcal{L} 的外延投影,则将概念 (A, B) 投影到概念 (A_1, B_1) 对应的格 $\psi(\mathcal{L})$ 上,使得 $A_1 = \psi(A), B_1 = A'_1$.

根据命题3,可以计算格中的概念到投影格中相应概念的映射.值得注意的是,投影的结果实际上是一个概念格.就本文的目的而言,其优点是通过投影约束格的结果保留了格结构.在本文后续部分,将通过投影约束格的结果称为约束格.

2.2 约束格

如前所述,在给定的形式背景中,存在一些表示属性之间属性依赖关系的蕴涵.但是,从领域专家的角度来看,某些蕴涵可能因为不同的原因而不能在数据中呈现.例如,某些对象可能缺少属性关联或可能错误地分配了属性.因此,在数据的属性关系和领域专家理解的属性关系之间常常存在差距.

为了弥补基于概念格的表示模型和领域专家表示模型之间的差距,本文将属性依赖集与概念格提供的蕴涵集“对齐”.根据定义1和定义2,如果蕴涵 $x \rightarrow y$ 在概念格中成立,则该格满足属性依赖 $x \prec y$.为了建立满足属性依赖集的格,需要寻找拥有相应属性蕴涵集合的格.这里提供了一种约束格结构的方法,对原始格中的一些概念修订或消失的原因给

予了解释说明,并使原始格结构得以保留.这些解释说明以这些概念映射到约束版本中的相应概念的形式提供.本文将这些映射称为原始格和最终受约束版本中发生的变化轨迹,并通过在格上使用外延投影来实现.下面举例来说明这种情况,格与受属性依赖约束的格之间的映射函数如图2所示.

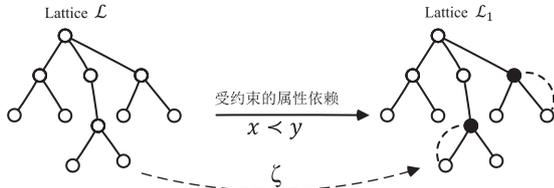


图2 格与受属性依赖约束的格之间的映射函数

格 \mathcal{L} 被映射到格 \mathcal{L}_1 上,格 \mathcal{L}_1 是一个受约束的版本,相对于属性依赖 $x < y$,称这个映射为 ζ .观察 ζ 的以下特征:1) ζ 缩小了格 \mathcal{L} 的大小,因为约束格 \mathcal{L}_1 不包含 \mathcal{L} 中不满足属性依赖关系 $x < y$ 的形式概念.2)根据格结构, ζ 减少概念外延,同时增加了内涵,如果一个概念 $C \in \mathcal{L}$ 满足属性依赖 $x < y$,则 $\zeta(C) = C$,如果一个概念 $C \in \mathcal{L}$ 不满足 $x < y$ 的内涵包含 x ,而不是 y ,可以用 $\text{int}(C)$ 表示概念 C 的内涵,为了满足 \mathcal{L}_1 中的 $x < y$, $\text{int}(\zeta(C)) = \text{int}(C) \cup K$,其中 K 是包含 y 的属性集合.为了得到满足约束条件的 \mathcal{L} 中的形式概念,映射函数 ζ 将格中的概念外延替换为仍具有外延的较小对象集合.算法1描述了这一形式概念更新过程.

算法1 概念更新算法

Input: C, S Output: S .

```

1 foreach  $C_1 \in S$  do
2   if  $\text{ext}(C) = \text{ext}(C_1)$  then
3     if  $\text{int}(C_1) \subseteq \text{int}(C)$  then
4        $C_1 \leftarrow (\text{ext}(C_1), \text{int}(C))$ ;
5       标记 $C_1$ 为原始概念;
6     else
7       if  $\text{int}(C_1) \subset \text{int}(C)$  then
8          $C_1 \leftarrow (\text{ext}(C_1), \text{int}(C) \cup \text{int}(C_1))$ ;
9         标记 $C_1$ 为已更新概念;
10  if  $\text{int}(C_1) \not\subseteq \text{int}(C)$  then
11    if  $\text{ext}(C) \subset \text{ext}(C_1)$  then
12      标记 $C$ 为已删除概念,从堆栈 $S$ 中移除 $C$ ;
13       $\text{ext}(C), \text{int}(C) \cup \text{int}(C_1) \Rightarrow S$ ;
14    if  $\text{ext}(C) \supset \text{ext}(C_1)$  then
15       $C_1 \leftarrow (\text{ext}(C_1), \text{int}(C) \cup \text{int}(C_1))$ ;
16  if  $\nexists C_2 \in S : \text{ext}(C_2) = \text{ext}(C) \cap \text{ext}(C_1)$ 
    then

```

```

17   $\text{ext}(C) \cap \text{ext}(C_1), \text{int}(C) \cup \text{int}(C_1) \Rightarrow S$ ;
18  if  $\text{int}(C) \cup \text{int}(C_1) \not\subseteq \text{int}(C_2)$  then
19     $C_2 \leftarrow (\text{ext}(C_2), \text{int}(C_2) \cup (\text{int}(C) \cup \text{int}(C_1)))$ ;

```

20 Return S

其中 C 是 \mathcal{L} 的概念,概念 C 的外延和内涵分别用 $\text{ext}(C)$ 和 $\text{int}(C)$ 表示.如果在 \mathcal{L} 中存在一个与 C 具有相同外延和内涵的概念,则 C 是原始概念;若在 \mathcal{L} 中存在与 $\text{int}(C)$ 具有相同内涵但外延与 $\text{ext}(C)$ 不同的概念,则 C 是更新的概念.算法中的循环用于扫描堆栈 S 中的一组概念,并根据命题3来标识.第10~第19行用于更新操作.

通过修订更新,从现有的格 \mathcal{L} 生成新的格 \mathcal{L}_1 上,为领域专家提供了形式概念变化的结果,以便专家可以确认 \mathcal{L} 和 \mathcal{L}_1 之间的区别,避免修订了重要的概念.这种更新可能会导致信息丢失.因此,变化轨迹有助于领域专家发现概念格中一些重要对象的缺失.图2所示映射的两个特征是事实的结果,是 ζ 的外延投影. ζ 是文献[16]的一个特例.这种外延投影不会创建新的外延,它会以更小的外延来替换格中的概念外延.下一节将描述如何在属性依赖和属性依赖集两种不同情况下定义外延投影,用以维护概念模型的一致性.

3 约束格的应用

3.1 约束格的投影与属性之间的依赖关系

对于外延投影 $\psi: \mathcal{L} \rightarrow \mathcal{L}_1$,其中 \mathcal{L} 是不满足属性 $x, y \in M$ 之间的属性蕴涵 $x \rightarrow y$ 的概念格, \mathcal{L}_1 是满足蕴涵 $x \rightarrow y$ 的 \mathcal{L} 的投影格.下面的命题说明了外延投影 ψ 的主要性质.

命题4 设 \mathcal{L} 是不满足属性 x 和 y 之间的属性蕴涵 $x \rightarrow y$ 的概念格,有 $x' \not\subseteq y' \Rightarrow x' \cap y' \subset x'$.

命题5 设 \mathcal{L} 是不满足属性 x 和 y 之间的属性蕴涵 $x \rightarrow y$ 的概念格,并且 ψ 是 \mathcal{L} 的外延投影,使得投影的格满足 $x \rightarrow y$,有 $\psi(x') = x' \cap y'$.

证明 由于投影格满足 $x \rightarrow y$,根据命题2,可得到 $\psi(x') \subseteq \psi(y')$;结合投影的收缩性质 $\psi(y') \subseteq y'$ (定义4),可得到 $\psi(x') \subseteq \psi(y') \subseteq y'$;同理结合 $\psi(x') \subseteq x'$,可得到 $\psi(x') \subseteq x' \cap y'$.

根据命题2, $\psi(x')$ 是投影格子 \mathcal{L}_1 中具有属性 x 的对象的最大集合.因为 $x' \cap y'$ 是 \mathcal{L} 中的一个外延,并且这个集合中的对象同时具有属性 x 和 y ,即 \mathcal{L} 中的外延 $x' \cap y'$ 的概念满足属性蕴涵 $x \rightarrow y$ 并且在 \mathcal{L}_1 中保持相同.因此,在 \mathcal{L}_1 中存在外延 $x' \cap y'$,并且这个集合中的对象包含属性 x ,即 $x' \cap y' \subseteq \psi(x')$;再

结合 $\psi(x') \subseteq x' \cap y'$, 可得 $\psi(x') = x' \cap y'$. \square

命题5给出了外延投影 ψ 观察格 \mathcal{L} 变化的一个重要性质:把格 \mathcal{L} 中的 x' 投影到 $x' \cap y'$ 上,由不满足蕴涵 $x \rightarrow y$ 的格 \mathcal{L} , 可得到满足蕴涵 $x \rightarrow y$ 的投影格.

给定不满足属性 x 和 y 之间的属性蕴涵 $x \rightarrow y$ 的概念格 \mathcal{L} 和它的外延投影 ψ , 使得 $(x') = x' \cap y'$, 格 \mathcal{L} 中的外延 A 可被划分为3个类别:类别I包含所有被 $x' \cap y'$ 包括在内的外延 A , 即 $A \subseteq (x' \cap y') \subset x'$; 类别II包含所有被 x' 包括在内, 但不被 $x' \cap y'$ 包括的外延 A , 即 $A \subseteq x', A \not\subseteq (x' \cap y')$; 类别III包含不在类别I和类别II中的外延 A , 即 $A \not\subseteq x'$.

考虑类别I中 $A \subseteq (x' \cap y') \subset x'$ 的元素 A , $x' \cap y'$ 是 \mathcal{L} 中的外延, 并且该集合中的对象具有属性 x 和 y , 即具有外延 $x' \cap y'$ 的概念满足属性蕴涵 $x \rightarrow y$. 因为 $x' \cap y'$ 包含 A , 所以具有外延 A 的概念满足属性蕴涵 $x \rightarrow y$. 因此, 在 \mathcal{L} 中的外延 A 的概念在投影格中保持相同, 即 $\psi(A) = A$. 类别I是投影不动点的组成部分.

考虑类别III $A \not\subseteq x'$ 中的元素 A , 因为 A 中的对象没有属性 x , 所以在 \mathcal{L} 中 A 的概念是满足属性蕴涵的 $x \rightarrow y$. 这些概念在投影格中保持不变, 即 $\psi(A) = A$. 类别III也是投影的定点部分.

考虑类别II中 $A \subseteq x', A \not\subseteq (x' \cap y')$ 的元素 A , 则有: 1) $\psi(A) \subseteq A$ (投影的收缩性质, 定义4); 2) 因为 $A \subseteq x', \psi(A) \subseteq \psi(x')$ (投影的幂等性质, 定义4), 而且, $\psi(x') = x' \cap y'$ (依据命题5), 所以 $\psi(A) \subseteq x' \cap y'$. 由1)和2)可以得到 $\psi(A) \subseteq A \cap (x' \cap y')$.

为了得到最大的不动点, 设 $\psi(A) = A \cap (x' \cap y')$, $\psi(A) = A \cap (x' \cap y')$ 符合投影的性质, $\psi(A) = A \cap (x' \cap y')$ 的概念满足蕴涵 $x \rightarrow y$, 因为 $\psi(A) = A \cap (x' \cap y')$ 中的对象同时具有属性 x 和 y , 从而引出这样一个事实, 即类别II约束了以 $\psi(A) = A \cap (x' \cap y')$ 的方式投影的概念. 因此, 由 $\psi(x') = x' \cap y'$ 给出的投影中具有最大不动点的外延投影如下所示:

$$\psi(A) = \begin{cases} A \cap (x' \cap y'), & A \subseteq x', A \not\subseteq (x' \cap y'); \\ A, & \text{otherwise.} \end{cases} \quad (1)$$

该投影给出了满足属性蕴涵 $x \rightarrow y$ 的投影格. 这一投影可以扩展到支持形式 X 的属性集之间的依赖关系. 给定不满足属性集 $X, Y \subseteq M$ 之间的蕴含 $X \rightarrow Y$ 的概念格 \mathcal{L} , 用于生成满足来自 \mathcal{L} 的蕴含 $X \rightarrow Y$ 的投影格的外延投影如下所示:

$$\psi(A) = \begin{cases} A \cap (X' \cap Y'), & A \subseteq X', A \not\subseteq (X' \cap Y'); \\ A, & \text{otherwise.} \end{cases} \quad (2)$$

例1 分析领域专家以属性依赖关系 $m_3 \prec m_4$ 的形式提供知识的运行示例, 根据表1中的数据: $m'_3 = g_3, g_5, g_6; m'_4 = g_3, g_6, g_7; m'_4 \cap m'_3 = g_3, g_6$. 应用式(2), 生成由来自原始格的依赖性 $m_3 \prec m_4$ 约束的格的外延投影 $\psi(A)$, 当 $A \subseteq \{g_3, g_5, g_6\}$ 且 $A \not\subseteq \{g_3, g_6\}$ 时, $\psi(A) = A \cap \{g_3, g_6\}$, 否则 $\psi(A) = A$.

在图1中也同时描绘了约束格和由投影 ψ 提供的变化轨迹. 变化的轨迹包含格 \mathcal{L} 中的 C_4 在约束格 \mathcal{L}_1 中变为 C_{12} , \mathcal{L} 中的 C_9 变为 \mathcal{L}_1 中的 C_{18} , \mathcal{L} 中的 C_{15} 变为 \mathcal{L}_1 中的 C_{19} . 其中 \mathcal{L} 中的 C_4 到 \mathcal{L}_1 中 C_{12} 的变化说明如下: 对象 g_3 和 g_6 与对象 g_5 一起形成概念 C_4 , 其内涵是 $\{m_3\}$. 依据领域专家意见, 可以飞的动物应该有翅膀 ($m_3 \prec m_4$), 所以对象 g_5 不应该与 g_3 和 g_6 一起形成一个概念. 因此以概念 C_{12} 表示的外延为代表, 其外延是 $\{g_3, g_6\}$, 内涵是 $\{m_2, m_3, m_4\}$. 通过观察 \mathcal{L} 中的 C_4 到 \mathcal{L}_1 中 C_{12} 的变化, 可以发现数据中包含有“海豚可以飞行”的噪声元素.

3.2 依赖关系集合

当领域专家以依赖关系集合的形式描述领域知识时, 针对约束格以及变更轨迹的问题, 首先为每个蕴涵生成约束格, 然后通过这些约束格的交运算来获得最终约束格.

定义5 设 \mathcal{L} 是一个概念格, ψ_1 和 ψ_2 是 \mathcal{L} 的两个投影, 则 $\psi_1 \leq \psi_2$, 如果在 $\psi_2(\mathcal{L})$ 上有一个投影 ψ , 使得对于 \mathcal{L} 中的所有 A , $\psi_1(A) = \psi \circ \psi_2(A)$, $\psi_1(\mathcal{L}) \subseteq \psi_2(\mathcal{L})$, 则 $\psi_1 \leq \psi_2$.

定义5指出, 投影可以从较“一般”到较“具体”顺序排列. 如果 ψ_1 是投影在 ψ_2 的投影格上, 则 ψ_1 比 ψ_2 更具体, 或者 ψ_2 比 ψ_1 更一般. 如果 ψ_1 是投影在 ψ_2 的投影上, 则 ψ_1 比 ψ_2 更具体.

命题6^[17] 有序格 \mathcal{L} 的投影形成一个半格 (F, \wedge) , 其中 $\psi_1, \psi_2 \in F$ 之间的半格运算由 $\psi_1 \wedge \psi_2 = \psi_3$, if $\psi_3(\mathcal{L}) = \psi_1(\mathcal{L}) \cap \psi_2(\mathcal{L})$ 给出.

命题7 设 \mathcal{L} 是概念格, ψ_1 是 \mathcal{L} 的外延投影, 使得投影格满足蕴涵 $X_1 \rightarrow Y_1$, ψ_2 是 \mathcal{L} 的外延投影, 使得投影格满足蕴涵 $X_2 \rightarrow Y_2$, 其中 $X_1, Y_1, X_2, Y_2 \subseteq M$, 则有:

- 1) 如果 $X_1 \subseteq X_2, Y_1 \subseteq Y_2$, 则 $\psi_1 \leq \psi_2$;
- 2) 如果 $\psi_1 \leq \psi_2$, 则由 ψ_1 给出的投影格满足 $X_1 \rightarrow Y_1$ 和 $X_2 \rightarrow Y_2$;
- 3) 存在 \mathcal{L} 的外延投影 ψ_3 , 使得 $\psi_3(\mathcal{L}) = \psi_1(\mathcal{L}) \cap \psi_2(\mathcal{L})$, ψ_3 给出了满足 $X_1 \rightarrow Y_1$ 和 $X_2 \rightarrow Y_2$ 的约束格.

证明 令 $\psi_1(\mathcal{L})$ 为由 ψ_1 给出的投影格, $\psi_2(\mathcal{L})$ 为 ψ_2 给出的投影格, 由于 $\psi_1 \leq \psi_2$, 根据定义5, 在 $\psi_2(\mathcal{L})$ 上定义了一些投影 ψ , 对于 \mathcal{L} 中的所有 A , $\psi_1(A) = \psi \circ \psi_2(A)$, 于是 $\psi_1(\mathcal{L}) \subseteq \psi_2(\mathcal{L})$. 所以, 由 ψ_1 给出的

投影格满足 $X_1 \rightarrow Y_1$ 和 $X_2 \rightarrow Y_2$.

为了证明 $\psi_1(\mathcal{L}) \leq \psi_2(\mathcal{L})$, 在 $\psi_2(\mathcal{L})$ 上定义投影 ψ 使得 $\psi_1(\mathcal{L}) = \psi \circ \psi_2(\mathcal{L})$. 由命题6可以看出, \mathcal{L} 上的投影集 F 是一个半格, 一定存在 $\psi_3 = \psi_1 \wedge \psi_2$. \square

由命题6给出的格的投影存在偏序. 生成约束格的投影顺序可以用命题7来表述.

依据命题7的结论, 给定属性集之间的两个依赖关系, 可以按照以下顺序排列与这些依赖关系相对应的投影. 首先, 如果一个依赖项依赖于另一个依赖项的属性集中包含的属性集, 则该依赖项的投影比另一个依赖项的投影更具体. 其次, 如果一个投影比另一个更具体, 则可以使用更具体的投影来生成最终的约束格, 而不是使用所有的投影^[18].

再回到生成满足依赖集约束格的场景. 设 \mathcal{L} 是一个概念格, ψ_i 是在 \mathcal{L} 的外延投影, 使得投影格满足蕴含 $X_i \rightarrow Y_i$, 其中 $X_i, Y_i \subseteq M$. 这组投影 ψ_i 可以按照命题7给出的投影顺序来排序. 这个顺序形成了命题6给出的半格. 依据命题7, 按照这一顺序通过最具体投影的交运算给出的最终约束格满足这组蕴含.

3.3 应用示例

根据投影的顺序, 有两种可能的方法来生成约束格. 第1种方法使用投影集合中的最具体投影的交运算来生成最终的约束格. 第2种方法使用所有投影来生成约束格集合. 应用第1种或第2种方法来生成约

束格取决于专家需要什么. 使用最具体投影的交运算来生成最终约束格的第1种方式在计算上比使用所有投影来生成相应约束格的第2种方式更有效. 然而, 第1种方式仅提供原始格和最终约束格之间的变化轨迹. 在专家需要所有变化的轨迹的情况下, 就需要第2种方式使用所有的投影来生成相应的约束格.

例2 考虑运行示例, 其中专家以依赖集 d_i 的形式提供其领域知识, 即

$$d_1 : \{m_3, m_8\} \prec \{m_4\}; d_2 : m_3 \prec m_4;$$

$$d_3 : \{m_1, m_2\} \prec \{m_3\}; d_4 : m_4 \prec m_3.$$

设 ψ_i 是用于生成满足相关性 d_i 的约束格的外延投影, 应用式(2)分别得到

$$\psi_1(A) = \begin{cases} A \cap \{g_3\}, & A \subseteq \{g_3, g_5\}, A \not\subseteq \{g_3\}; \\ A, & \text{otherwise;} \end{cases} \quad (3)$$

$$\psi_2(A) = \begin{cases} A \cap \{g_3, g_6\}, & A \subseteq \{g_3, g_5, g_6\}, \\ & A \not\subseteq \{g_3, g_6\}; \\ A, & \text{otherwise;} \end{cases} \quad (4)$$

$$\psi_3(A) = \begin{cases} A \cap \{g_3\}, & A \subseteq \{g_3, g_7\}, A \not\subseteq \{g_3\}; \\ A, & \text{otherwise;} \end{cases} \quad (5)$$

$$\psi_4(A) = \begin{cases} A \cap \{g_3, g_6\}, & A \subseteq \{g_3, g_6, g_7\}, \\ & A \not\subseteq \{g_3, g_6\}; \\ A, & \text{otherwise.} \end{cases} \quad (6)$$

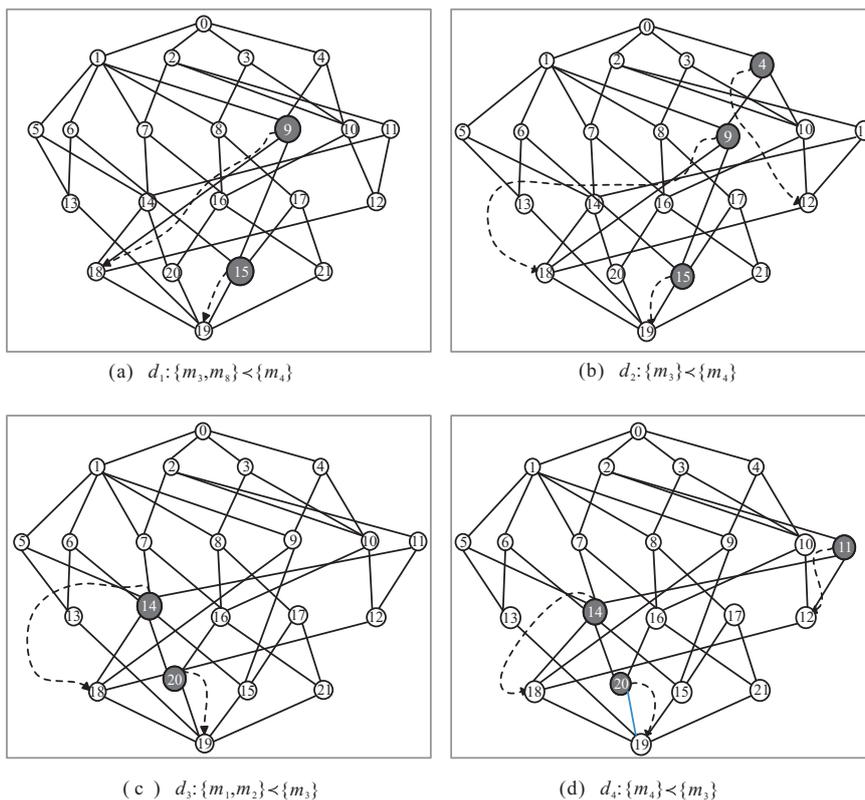


图3 受领域知识条件约束的格与变化轨迹

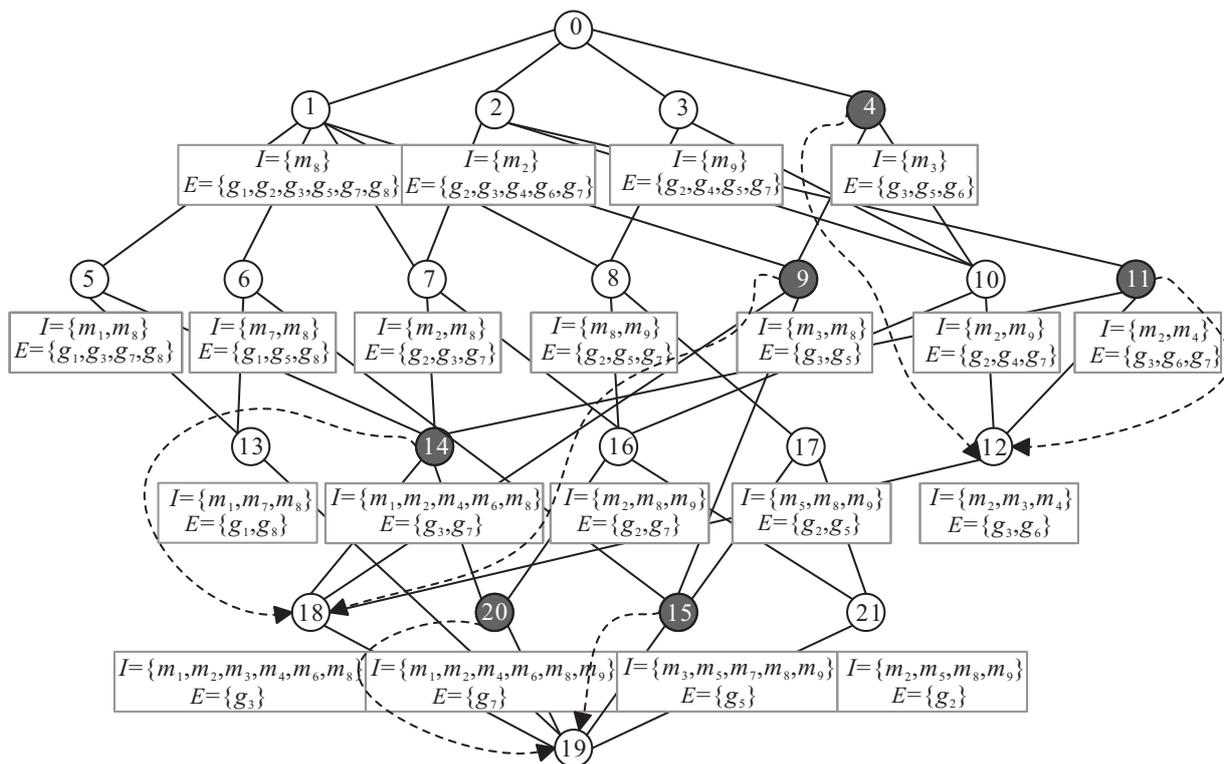


图4 最终的约束格和变化轨迹

由投影 $\psi_1, \psi_2, \psi_3, \psi_4$ 提供的约束格和变化的轨迹如图3所示. 在这组投影中, $\psi_2 \leq \psi_1$ 和 $\psi_4 \leq \psi_3$. 可以看到约束格 $\psi_2(\mathcal{L})$ 包含在 $\psi_1(\mathcal{L})$ 中, $\psi_4(\mathcal{L})$ 包含在 $\psi_3(\mathcal{L})$ 中.

设 ψ_5 为最具体投影的交运算的外延投影, 即 $\psi_5 = \psi_2 \wedge \psi_4$, 得到的 ψ_5 定义如下:

$$\psi_5(A) = \begin{cases} A \cap \{g_3, g_6\}, & A \subseteq \{g_3, g_6, g_7\}, A \not\subseteq \{g_3, g_6\}; \\ A \cap \{g_3, g_6\}, & A \subseteq \{g_3, g_5, g_6\}, A \not\subseteq \{g_3, g_6\}; \\ A, & \text{otherwise.} \end{cases} \quad (7)$$

这组投影 $\psi_1, \psi_2, \psi_3, \psi_4, \psi_5$ 形成一个半格. 有两种方法来生成最终的约束格. 第1种方式使用最具体的投影 ψ_5 的交运算. 图4描述了最终的约束格和原始格与由投影 ψ_5 提供的最终约束格之间的变化轨迹. 第2种方式是使用所有的投影来获得全部的变化轨迹.

根据投影的半格, $\psi_2 \leq \psi_1$ 和 $\psi_4 \leq \psi_3$ 为了获得全部的变化轨迹, 在 ψ_2 之前施加 ψ_1, ψ_4 之前施加 ψ_3 . 相比之下, 当 ψ_2 和 ψ_4 不兼容时, 首先应用 ψ_1 和 ψ_2 还是 ψ_3 和 ψ_4 并不重要. 因此, 可以根据 $\psi_1, \psi_2, \psi_3, \psi_4, \psi_5$ 或 $\psi_3, \psi_4, \psi_1, \psi_2, \psi_5$ 的顺序进行投影, 变化轨迹分别对应应如图3所示的 (a), (b), (c), (d) 顺序或 (c), (d), (a), (b) 顺序到图4所示的格链. 这两种情形, 专家均可以访问对应于每个依赖项的更改.

4 结论

本文提出了一种基于外延投影将专家知识整合到概念格中的方法, 从而保持格的结构和变化的轨迹. 专家知识被形式化为属性依赖关系集, 与由概念格提供的内涵集合对齐. 根据投影的不同排列顺序, 该方法提供了两种生成约束格的途径: 一是使用最具体投影的交运算来生成最终的约束格; 二是使用所有投影来生成一组约束格. 第1种途径生成约束格的计算效率更高, 但只提供原始格和最终约束格之间的变化轨迹, 而第2种途径则提供所有变化轨迹, 但生成约束格的计算效率较低.

未来的工作包括以类似的方式定义内涵投影, 以便集成不同对象之间的依赖关系, 并允许将对象的分类整合到概念格中. 当数据中遗漏了一些内涵时, 本文提出的方法可以用于增加领域专家期望存在的内涵来完成数据的定义.

参考文献(References)

[1] Ganter B, Stumme G, Wille R. Formal concept analysis: foundations and applications[M]. Berlin, Heidelberg: Springer-Verlag, 2005: 1-47.

[2] Sarmah A K, Hazarika S M, Sinha S K. Formal concept analysis: Current trends and directions[J]. Artificial Intelligence Review, 2015, 44(1): 47-86.

[3] Wille R. Restructuring lattice theory: An approach based on hierarchies of concepts[J]. Orderd Sets D Reidel, 1982,

- 83(1): 314-339.
- [4] Sumangali K, Ch A K, Li J. Concept compression in formal concept analysis using entropy-based attribute priority[J]. *Applied Artificial Intelligence*, 2017, 31(1): 1-28.
- [5] AitYakoub Zina, Djouadi Yassine, Dubois Didier, et al. Asymmetric composition of possibilistic operators in formal concept analysis: Application to the extraction of attribute implications from incomplete contexts[J]. *Int J of Intelligent Systems*, 2017, 32(12): 1285-1311.
- [6] Ganter B, Wille R, Borchmann D, et al. Implications and dependencies between attributes[C]. *Int Conf on Formal Concept Analysis*. Rennes: Springer, 2017: 23-35.
- [7] Vladimir Dzyuba, Matthijs van Leeuwen, Siegfried Nijssen, et al. Interactive learning of pattern rankings[J]. *Int J on Artificial Intelligence Tools*, 2014, 23(6): 1-31.
- [8] Dzyuba V, Leeuwen M V, Raedt L D. Flexible constrained sampling with guarantees for pattern mining[J]. *Data Mining & Knowledge Discovery*, 2017, 31(3): 1-28.
- [9] 徐怡, 王泉, 霍思林. 粒计算中基于属性分类的形式概念属性约简[J]. *控制与决策*, 2018, 33(12): 2203-2207. (Xu Y, Wang Q, Huo S L. The guide of the formal concept attribute reduction model based on attribute classification relation[J]. *Control and Decision*, 2018, 33(12): 2203-2207.)
- [10] 周莉, 王珏, 周勇. 函数依赖集在属性子集上投影的新方法[J]. *江西师范大学学报: 自然科学版*, 2013, 37(4): 387-391. (Zhou L, Wang J, Zhou Y. New method of projection of function dependencies onto attributes[J]. *J of Jiangxi Normal University: Natural Science*, 2013, 37(4): 387-391.)
- [11] 安秋生, 孔祥玉. 函数依赖与属性蕴含的关系研究[J]. *小型微型计算机系统*, 2017, 38(9): 2000-2005. (An Q S, Kong X Y. Relationship study of functional dependency and attribute implication[J]. *J of Chinese Computer Systems*, 2017, 38(9): 2000-2005.)
- [12] Baixeries J, Kaytoue M, Napoli A. Characterizing functional dependencies in formal concept analysis with pattern structures[J]. *Annals of Mathematics & Artificial Intelligence*, 2014, 72(1/2): 129-149.
- [13] Buzmakov A, Egho E, Jay N, et al. On mining complex sequential data by means of FCA and pattern structures[J]. *Int J of General Systems*, 2015, 45(2): 1-27.
- [14] Cherukuri Kumar. Knowledge discovery in data using formal concept analysis and random projections[J]. *Int J of Applied Mathematics & Computer Science*, 2011, 21(4): 745-756.
- [15] 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.
- [16] Buzmakov A, Egho E, Jay N, et al. The representation of sequential patterns and their projections within Formal Concept Analysis[C]. *Machine Learning and Knowledge Discovery in Databases*. Prague: Springer, 2013: 65-79.
- [17] Kaytoue M, Codocedo V, Buzmakov A, et al. Pattern structures and concept lattices for data mining and knowledge processing[C]. *Machine Learning and Knowledge Discovery in Databases*. Porto: Springer-Verlag, 2015: 227-231.
- [18] Stumme G. Formal concept analysis on its way from mathematics to computer science[M]. *Conceptual Structures: Integration and Interfaces*. Borovets: Springer-Verlag, 2002: 2-19.

作者简介

刘勇(1974—), 男, 副教授, 博士, 从事智能信息处理、知识工程及其应用等研究, E-mail: liuyong@sdu.edu.cn;

胡宇鹏(1983—), 男, 讲师, 博士, 从事知识工程、软件工程等研究, E-mail: huyupeng@sdu.edu.cn;

李学庆(1964—), 男, 教授, 博士生导师, 从事软件工程、知识工程和人机交互与虚拟现实等研究, E-mail: xqli@sdu.edu.cn.

(责任编辑: 孙艺红)