

基于约束的模糊概念格构造算法

崔芳婷 王黎明 张卓

(郑州大学信息工程学院 郑州 450001)

摘要 一般的模糊概念格在构造过程中没有考虑用户的需求,用户对模糊概念格节点中一些属性集形成的内涵并不感兴趣。为了增强模糊概念格的针对性,降低模糊概念格构造的时空复杂性,构造满足用户需求的模糊概念格,首先将用户感兴趣的背景知识定义为约束条件,根据用户关心的属性间关系,将约束条件分为3类:单约束、与约束及或约束,并采用谓词公式表示,进而提出了基于约束的模糊概念格(Constrained Fuzzy Concept Lattice, CFCL)构造算法。该算法自底向上构造模糊概念格,利用模糊概念格父子节点内涵的单调关系,采用剪枝技术来减少构造过程中判断模糊概念是否满足约束的次数,提高了模糊概念格的构造效率。实验结果表明,该算法能够有效地减少模糊概念格的存储空间和构格时间。

关键词 模糊概念格,谓词逻辑,约束,构造算法

中图分类号 TP18 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.8.059

Construction Algorithm of Fuzzy Concept Lattice Based on Constraints

CUI Fang-ting WANG Li-ming ZHANG Zhuo

(School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China)

Abstract The general process of constructing fuzzy concept lattice does not take user's requirement into account. Users sometimes are not interested in all intensions of attribute sets in fuzzy concept lattice node. In order to enhance the pertinence of fuzzy concept lattice, reduce the time and space complexity and construct fuzzy concept lattice which can meet the need of users, first of all, the background knowledge which users are interested was defined as a constraint condition. We divided the constraint into three categories: single-constraint, and-constraint and or-constraint by the relation among attributes. Then we formalized the constraint by predicate formula. And a construction algorithm based on constraints (Constrained Fuzzy Concept Lattice, CFCL) was presented. The algorithm presents a bottom-up method to compute the fuzzy concept lattice. By making use of the monotone relation between father fuzzy concept's intent and child fuzzy concept's intent, this method reduces the judgment operations between fuzzy concepts and the constraints. Thus the efficiency of building the fuzzy concept lattice is improved. Finally, the experiment results verify that the proposed algorithm can reduce the storage space and time of fuzzy concept lattice construction.

Keywords Fuzzy concept lattice, Predicate logic, Constraint, Construction algorithm

1 引言

概念格^[1]中的每个节点是一个形式概念,由内涵(属性)和外延(对象)两部分构成,是一种有效的数据分析和知识提取工具,已广泛应用于机器学习、数据挖掘^[2-4]、知识发现、信息检索、软件工程^[5]等诸多领域。一般的概念格构造算法是以形式背景作为单一输入构造出完备的概念格,构造过程并没有考虑用户的需求。为构造满足用户不同需求的概念格,国内外学者进行了多方面的研究。国外的 Belohlavek 团队在文献^[6]中将属性依赖规则定义为约束,把约束作为额外的输入引入到概念格的构造过程中,约简了概念格的规模;文献^[7]采用闭包算子表示用户自定义的需求,进而构造满足用户需求的概念格;文献^[8]将用户自定义的属性优先级定义为约

束,进而提取符合约束的概念。国内的张继福团队将用户感兴趣的属性集定义为约束条件,并将其引入到概念格中,提出了约束概念格理论^[9],并证明了约束概念格的完备性,给出了约束概念格的构造方法。

概念格只是针对精确的形式概念,然而现实应用中的大多数信息都是模糊的、不确定的,研究学者将形式概念分析^[1]和模糊逻辑结合在一起,使用隶属度来表示不确定的信息,提出了模糊形式概念,进而产生了模糊概念格^[10]这一新的研究领域。Burusco^[11]首先提出了使用三角模对传统的伽罗瓦联系进行模糊扩展的方法,但是其定义的对象与属性间的映射并不满足闭包特性,失去了与传统概念格的兼容性。国内学者范世青^[12]在交换伴随对与对合剩余格的条件下,讨论了模糊概念格的4种定义形式;张文修^[13]等讨论了变精度概念格

到稿日期:2014-08-31 返修日期:2014-11-21 本文受国家青年科学基金项目(61303044)资助。

崔芳婷(1990—),女,硕士生,主要研究方向为机器学习、数据挖掘,E-mail:happy_ftcui@163.com;王黎明(1963—),男,博士,教授,CCF高级会员,主要研究方向为分布式人工智能、现代软件工程技术等;张卓(1978—),男,博士,讲师,CCF会员,主要研究方向为形式概念分析及其应用等,E-mail:iezhangzhuo@zzu.edu.cn(通信作者)。

的构造与性质。国外的 Belohlavek 提出了以完备剩余格为真值度的 L-伽罗瓦连接^[14]的定义,使用模糊逻辑的剩余算子及其模糊二元关系,将精确概念格的特性扩展到模糊概念格上,从而保持了与传统概念格兼容的数学特性。基于 Belohlavek 所定义的模糊概念格的直接构造算法并不多;文献^[15]受到 Ganter 的 NextClosure 算法^[1]启发,提出了 FuzzyNextClosure 算法构造 L-模糊概念,但并没有产生格结构;文献^[16]提出了构造 L-模糊概念格的 Lattice 算法,该方法是自底向上产生模糊概念,并生成格结构。由于模糊概念格的构造过程具有指数级的时间复杂度和空间复杂度,为提高模糊概念的构造效率,文献^[17]将模糊集合组合空间映射为自然数空间,充分利用计算资源,提出模糊概念并行构造算法。为减少模糊概念格的存储空间,文献^[18]提出了基于障篱的模糊概念格约简方法;文献^[19]提出了通过模糊闭包操作,约简模糊概念格规模的方法。

模糊概念格虽然可以直接处理模糊、不确定数据,但是在模糊概念格的构造过程中,并没有考虑到用户的需求。为了构造满足用户需求的模糊概念格,增强模糊概念格的针对性,降低模糊概念格构造的时空复杂性,本文将用户感兴趣的背景知识定义为约束,采用谓词逻辑表示约束,并将约束引入到模糊概念格结构中,在此基础上,提出一种基于约束的模糊概念格构造算法。实验验证了该算法能够有效地节约模糊概念格的存储空间和建格时间。

2 相关概念

本文的工作是在 Belohlavek 定义的模糊概念格的基础上进行的,文中使用的模糊概念、模糊伽罗瓦联系和模糊概念格的相关定义和性质不再赘述,具体的形式化描述可参考文献^[14]。将用户感兴趣的背景知识定义为约束条件,根据用户关心的模糊概念格节点中属性间的关系,将约束条件分为 3 类:单约束、与约束及或约束。采用谓词逻辑表示约束,首先定义描述背景知识的谓词,然后通过连接词(\wedge 、 \vee 、 \rightarrow 、 \forall 、 \exists)将相关的谓词连接起来,形成谓词公式,表达相应的约束条件。

定义 1 一元谓词 $F(z)$:表示模糊概念格^[16]节点集合,其中 z 表示模糊概念格节点。

定义 2 四元谓词 $Concept(z, x, y, \mu)$:表示模糊概念格节点 z 的外延为 x ,内涵为 y ,隶属度为 μ 。

定义 3 二元谓词 $Contain(y, k)$:表示属性 k 包含在模糊概念格节点的内涵 y 中。

定义 4 二元谓词 $Care(k, \lambda_k)$:表示用户关心的属性为 k ,并且满足隶属度 $\mu_k \geq \lambda_k$ 。

定义 5 一元谓词 $Constraint(z)$:表示满足约束的模糊概念格节点,其中 z 表示模糊概念格节点。

定义 6 单约束 $P_1(Z)$: $P_1(Z) = \forall z((F(z) \wedge Concept(z, x, y, \mu) \wedge Contain(y, k_0) \wedge Care(k_0, \lambda_k)) \rightarrow Constraint(z))$ 为一谓词公式。表示对于一个模糊概念格节点,如果其内涵 y 包含用户关心的属性 k_0 ,且属性 k_0 的隶属度满足条件,则该节点为满足单约束的模糊概念格节点。

定义 7 与约束 $P_2(Z)$: $P_2(Z) = \forall z((F(z) \wedge Concept(z, x, y, \mu) \wedge (Contain(y, k_0) \wedge Contain(y, k_1)) \wedge (Care(k_0, \lambda_0) \wedge Care(k_1, \lambda_1))) \rightarrow Constraint(z))$ 为一谓词公式。表示对

于一个模糊概念格节点,如果其内涵 y 包含用户关心的属性 k_0 和 k_1 ,且属性 k_0 和 k_1 均满足条件,则该节点为满足与约束的模糊概念格节点。

定义 8 或约束 $P_3(Z)$: $P_3(Z) = \forall z((F(z) \wedge Concept(z, x, y, \mu) \wedge (Contain(y, k_0) \vee Contain(y, k_1)) \wedge (Care(k_0, \lambda_0) \vee Care(k_1, \lambda_1))) \rightarrow Constraint(z))$ 为一谓词公式。表示对于一个模糊概念格节点,如果其内涵 y 包含用户关心的属性集 k_0 或者 k_1 ,且属性 k_0 或者 k_1 满足条件,则该节点为满足或约束的模糊概念格节点。

定义 9(满足约束的模糊概念) 给定模糊形式背景 (X, Y, I) ,基于约束的模糊概念的形式为三元组 $H((A, B), P)$,其中 $(A, B) \in L^X \times L^{Y[14]}$, P 为约束条件,如果满足 $A^+ = B$, $B^+ = A$,且 $P((A, B)) = True$,则称 $H((A, B), P)$ 为满足约束的模糊概念。

定义 10(满足约束的模糊概念格) 将所有满足约束的模糊概念节点 $H((A, B), P)$ 及其偏序关系 \leq ^[14] 构成的有序集合 (H, \leq) 称为满足约束的模糊概念格。

3 基于约束的模糊概念格构造算法

在相关概念的基础上,提出基于约束的模糊概念格构造算法,简称 CFCL 算法。下面详细介绍算法所涉及的相关定理。

3.1 理论基础

定理 1 设 (X, Y, I) 为一模糊形式背景,当约束条件 $P = \emptyset$ 时,满足约束的模糊概念格退化为一般的模糊概念格。

证明:给定模糊形式背景 (X, Y, I) 下的一般的模糊概念^[14]节点 $C(A, B)$,由于 $P = \emptyset$,则 $P(C) = True$,因此 $((A, B), P)$ 为满足约束的模糊概念格节点。设 $C_1(A_1, B_1)$ 和 $C_2(A_2, B_2)$ 是模糊形式背景下的两个不同的一般模糊概念格节点,满足 $C_1 \leq C_2$ 。又因 $P = \emptyset$,所以 $((A_1, B_1), P)$ 和 $((A_2, B_2), P)$ 为满足用户约束的两个模糊概念,且有 $((A_1, B_1), P) \leq ((A_2, B_2), P)$ 。故得证。

定理 2 设 (X, Y, I) 为一模糊形式背景, $C(X, Y, I)$ 为基于模糊形式背景构造的模糊概念格^[14],对于任意模糊概念格节点 $C_1(A_1, B_1), C_2(A_2, B_2) \in C(X, Y, I)$,满足 $C_1 \leq C_2$,则

(1)对于单约束 P_1 : $P_1(C_2) = True \Rightarrow P_1(C_1) = True$

(2)对于与约束 P_2 : $P_2(C_2) = True \Rightarrow P_2(C_1) = True$

证明:根据模糊概念格的性质,若 $C_1 \leq C_2$,则 $A_1 \sqsubset A_2$ 或者 $B_1 \supset B_2$ 。根据定义 6,给定单约束 P_1 ,若 $P_1(C_2) = True$,则对于用户约束的属性 $k \in B_2$,将 k 记为 k_2 ,且满足 $\mu_{k_2} \geq \lambda_k$ 。那么,由于 $B_1 \supset B_2 \Rightarrow$ 对于 $k \in B_1, \mu_{k_1} \geq \mu_{k_2} \geq \lambda_k$ 成立 $\Rightarrow P_1(C_1) = True$ 。同理,对于与约束 P_2 ,若 $P_2(C_2) = True$,则对于用户约束的属性 $k, k' \in B_2$,将 k 和 k' 分别记为 k_2 和 k_2' ,满足 $\mu_{k_2} \geq \lambda_k$ 且 $\mu_{k_2'} \geq \lambda_{k'}$ 。那么,由于 $B_1 \supset B_2 \Rightarrow$ 对于 $k, k' \in B_1$,分别记为 k_1 和 k_1' ,有 $\mu_{k_1} \geq \mu_{k_2} \geq \lambda_k$ 且 $\mu_{k_1'} \geq \mu_{k_2'} \geq \lambda_{k'}$ 成立 $\Rightarrow P_2(C_1) = True$,故定理 2 得证。

推论 设 (X, Y, I) 为一模糊形式背景, $C(X, Y, I)$ 为基于模糊形式背景构造的模糊概念格,对于任意模糊概念格节点 $C_1(A_1, B_1), C_2(A_2, B_2) \in C(X, Y, I)$,满足 $C_1 \leq C_2$,则

(1)对于单约束 P_1 : $P_1(C_1) = False \Rightarrow P_1(C_2) = False$

(2)对于与约束 P_2 : $P_2(C_1) = False \Rightarrow P_2(C_2) = False$

证明:由定理 2 的逆否命题即得证。

定理 3 设 (X, Y, I) 为一模糊形式背景, $C(X, Y, I)$ 为基于模糊形式背景构造的模糊概念格, 对于任意模糊概念格节点 $C_1(A_1, B_1), C_2(A_2, B_2) \in C(X, Y, I)$, 满足 $C_1 \leq C_2$, 则

对于或约束 $P_3: P_3(C_1) = \text{False} \Rightarrow P_3(C_2) = \text{False}$

证明: 根据模糊概念格的性质, 若 $C_1 \leq C_2$, 则 $A_1 \subseteq A_2$ 或者 $B_1 \supseteq B_2$. 根据定义 8, 给定用户约束条件 P_3 , 若 $P_3(C_1) = \text{False}$, 那么对于用户约束的属性 $k, k' \in B_1$, 将 k 和 k' 分别记为 k_1 和 k'_1 , 必满足以下 3 个条件: (1) $\mu_{k_1} < \lambda_k$; (2) $\mu_{k'_1} < \lambda_{k'}$; (3) $\mu_{k_1} < \lambda_k$ 且 $\mu_{k'_1} < \lambda_{k'}$. 下面分别对以上 3 种情况进行讨论: 若满足条件(1), $\mu_{k_1} < \lambda_k$, 那么由于 $B_1 \supseteq B_2 \Rightarrow$ 对于 $k \in B_2, \mu_{k_2} < \mu_{k_1} < \lambda_k$ 成立 $\Rightarrow P_3(C_2) = \text{False}$. 同理, 条件(2)也可得证. 对于条件(3), $\mu_{k_1} < \lambda_k$ 且 $\mu_{k'_1} < \lambda_{k'}$, 那么由于 $B_1 \supseteq B_2 \Rightarrow$ 对于 $k, k' \in B_2, \mu_{k_2} < \mu_{k_1} < \lambda_k$ 且 $\mu_{k'_2} < \mu_{k'_1} < \lambda_{k'}$ 成立 $\Rightarrow P_3(C_2) = \text{False}$. 综上, 定理 3 得证.

3.2 基于约束的模糊概念格构造过程

本文是在 Belohlavek 的模糊概念格构造方法^[16]的基础上进行的, 根据用户给定的约束, 采用自底向上的方式构造满足约束的模糊概念格. 在构造满足约束的模糊概念格的过程中, 根据定理 2 的推论以及定理 3, 对于不满足约束条件的子节点, 该子节点的父节点概念也不满足约束, 因此将不满足约束的子节点删除, 仅将满足约束条件的模糊概念节点加入到格中, 并生成其全部父节点, 然后对其父节点依次判断是否满足约束, 直到生成模糊概念顶节点. 在模糊概念格的更新过程中, 删除不满足约束的概念节点, 可以避免产生一些冗余信息, 同时减少判断模糊概念节点是否满足约束的次数, 提高构格效率. 现将基于约束的模糊概念格构造过程描述如下:

步骤 1 根据模糊形式背景, 确定模糊概念格底节点, 生成其全部父节点;

步骤 2 判断父节点是否满足约束;

步骤 3 标记满足约束的模糊概念父节点, 将模糊概念父节点及其子节点确立偏序关系, 加入到模糊概念格结构中;

步骤 4 依次生成标记模糊概念节点的全部父节点, 重复步骤 2 和步骤 3, 直到生成模糊概念顶节点.

下面通过一个例子来说明本文提出的基于约束的模糊概念格构造的过程.

例 1 表 1 为模糊形式背景, 其中对象集 $X = \{x_1, x_2, x_3\}$, 属性集 $Y = \{y_1, y_2, y_3, y_4, y_5\}$, 精度为 $L_3 = \{0, 0.5, 1\}$. 由该模糊形式背景构造出的一般的模糊概念格如图 1 所示, 其中模糊概念格中的节点分别为: $C_0 = \{(1, 1, 1), (0, 0, 0.5, 0.5, 0.5)\}$, $C_1 = \{(1, 1, 0.5), (0.5, 0.5, 0.5, 1, 0.5)\}$, $C_2 = \{(1, 0.5, 1), (0, 0, 0.5, 0.5, 1)\}$, $C_3 = \{(1, 1, 0), (1, 0.5, 0.5, 1, 0.5)\}$, $C_4 = \{(0.5, 1, 0.5), (0.5, 0.5, 1, 1, 0.5)\}$, $C_5 = \{(1, 0.5, 0.5), (0.5, 0.5, 0.5, 1, 1)\}$, $C_6 = \{(0.5, 1, 0), (1, 1, 1, 0.5)\}$, $C_7 = \{(1, 0.5, 0), (1, 0.5, 0.5, 1, 1)\}$, $C_8 = \{(0.5, 0.5, 0.5), (0.5, 0.5, 1, 1, 1)\}$, $C_9 = \{(0.5, 0.5, 0), (1, 1, 1, 1, 1)\}$.

表 1 模糊形式背景

	y_1	y_2	y_3	y_4	y_5
x_1	1	0.5	0.5	1	1
x_2	1	1	1	1	0.5
x_3	0	0	0.5	0.5	1

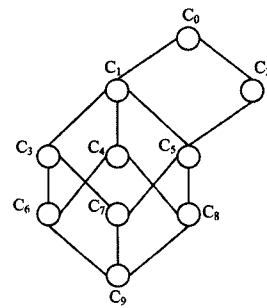


图 1 一般的模糊概念格

以表 1 的模糊形式背景为例, 如果用户关心模糊概念格节点内涵中的属性 y_5 , 且满足隶属度 $\mu_5 \geq 1$, 即约束条件为单约束 $P_1(Z) = \forall z((F(z) \wedge \text{Concept}(z, x, y, \mu) \wedge \text{Contain}(y, y_5) \wedge \text{Care}(y_5, 1)) \rightarrow \text{Constraint}(z))$, 由上述步骤可以构造满足用户约束的模糊概念格: 首先确定模糊概念格底节点 C_9 , 判断模糊概念格节点 C_9 满足约束条件, 将其加入概念格中, 然后生成 C_9 的全部父节点 C_6, C_7 和 C_8 , 依次判断父节点是否满足约束, 发现 C_6 节点不满足约束条件, 因此将 C_6 节点删除, 并且不再生成 C_6 节点的父节点 C_3 和 C_4 , 将满足约束的概念节点 C_7 和 C_8 加入格中, 并与 C_9 节点建立偏序关系. 重复上面的步骤, 最终构造的满足单约束 P_1 的模糊概念格如图 2 所示, 其中 $C_0 = \{(1, 0.5, 1), (0, 0, 0.5, 0.5, 1)\}$, $C_1 = \{(1, 0.5, 0.5), (0.5, 0.5, 0.5, 1, 1)\}$, $C_2 = \{(1, 0.5, 0), (1, 0.5, 0.5, 1, 1)\}$, $C_3 = \{(0.5, 0.5, 0.5), (0.5, 0.5, 1, 1, 1)\}$, $C_4 = \{(0.5, 0.5, 0), (1, 1, 1, 1, 1)\}$.

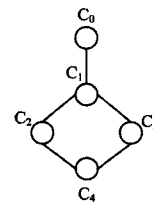


图 2 满足单约束 P_1 的模糊概念格

如果用户关心模糊概念格节点内涵中属性 y_1 和属性 y_3 , 且满足隶属度 $\mu_1 \geq 0.5$ 和 $\mu_3 \geq 1$, 即约束条件为与约束 $P_2(Z) = \forall z((F(z) \wedge \text{Concept}(z, x, y, \mu) \wedge (\text{Contain}(y, y_1) \wedge \text{Contain}(y, y_2)) \wedge (\text{Care}(y_1, 0.5) \wedge \text{Care}(y_3, 1)))) \rightarrow \text{Constraint}(z))$, 依据上述步骤构造的满足与约束 P_2 的模糊概念格如图 3 所示, 其中 $C_0 = \{(0.5, 1, 0.5), (0.5, 0.5, 1, 1, 0.5)\}$, $C_1 = \{(0.5, 1, 0), (1, 1, 1, 0.5)\}$, $C_2 = \{(0.5, 0.5, 0.5), (0.5, 0.5, 1, 1, 1)\}$, $C_3 = \{(0.5, 0.5, 0), (1, 1, 1, 1, 1)\}$.

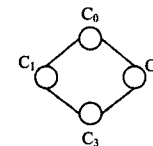


图 3 满足与约束 P_2 的模糊概念格

如果用户关心模糊概念格节点内涵中属性 y_1 或者 y_3 , 且满足隶属度 $\mu_1 \geq 0.5$ 或者 $\mu_3 \geq 1$, 即约束条件为或约束 $P_3(Z) = \forall z((F(z) \wedge \text{Concept}(z, x, y, \mu) \wedge (\text{Contain}(y, y_1) \vee \text{Contain}(y, y_3)) \wedge (\text{Care}(y_1, 0.5) \vee \text{Care}(y_3, 1)))) \rightarrow \text{Constraint}(z))$, 依据上述步骤构造的满足或约束 P_3 的模糊概念格如图 4 所示, 其中 $C_0 = \{(1, 1, 0.5), (0.5, 0.5, 0.5, 1, 1)\}$.

0.5)}, C₁ = {(1,1,0), (1,0.5,0.5,1,0.5)}, C₂ = {(0.5,1,0.5), (0.5,0.5,1,1,0.5)}, C₃ = {(1,0.5,0.5), (0.5,0.5,0.5,1,1)}, C₄ = {(0.5,1,0), (1,1,1,0.5)}, C₅ = {(1,0.5,0), (1,0.5,0.5,1,1)}, C₆ = {(0.5,0.5,0.5), (0.5,0.5,1,1,1)}, C₇ = {(0.5,0.5,0), (1,1,1,1,1)}.

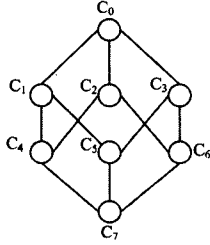


图4 满足或约束 P₃ 的模糊概念格

由上面例子可以看出,不同的约束条件可以满足用户的不同需求,并且不同的条件对模糊概念格的约束程度也不相同。对于不同的约束条件,约束程度由强到弱依次是:与约束,单约束,或约束。对于同一约束条件,用户约束的属性隶属度取值越大,约束模糊概念格的程度就越强;用户约束的属性隶属度取值越小,约束的程度就越弱。约束的程度越强,构造的满足约束的模糊概念格节点数目相应越少;反之则越多。本例对于与约束和或约束仅使用两个属性的与、或关系进行约束,基于多个属性的与约束和或约束的模糊概念格的构造过程是类似的,不再举例说明。

3.3 算法描述

根据上述对基于约束的模糊概念格构造过程的分析,首先给出模糊概念的直接前序(父节点)生成函数(UpperNeighbors)。Belohlavek 在文献[16]中给出了模糊概念的直接后序(子节点)构造算法(NEIGHBORS Algorithm)。该算法的主要步骤是:根据一个给定模糊概念的内涵 B,通过模糊闭包运算 $C(\downarrow \uparrow)$ 构造出该模糊概念的所有直接后序(子概念)的内涵集 U。需要注意的是,NEIGHBORS 算法仅构造出所有直接后序概念的内涵。该算法中假设对象集合 $X = \{1, 2, \dots, n\}$, 真值集合 $L = \{0 = a_1 < a_2 < \dots < a_n = 1\}$, 对于 $i < k$, 本文使用 a^{i+} 表示 a^{i+1} 。对于任意内涵模糊集合 $B \in L^Y, y \in Y$, 并且 $B(y) < 1$, 那么标记 $[y]_B^{\uparrow} = (B \cup \{B(y)^+ / y\})^{\uparrow \uparrow}$, 标记 $[y]_B^{\uparrow}$ 表示在内涵 y 上产生的后序概念的内涵集,但 $[y]_B^{\uparrow}$ 并不一定是概念 B 的直接后序,在此基础上进行筛选,求出所有直接后序概念的内涵集,具体证明请参照文献[16]。

由于模糊概念格是完备的,因此在 NEIGHBORS 算法的基础上,可以根据给定模糊概念的外延 A,通过闭包运算 $D(\uparrow \downarrow)$ 构造出该模糊概念的所有直接前序概念(父概念)的外延集 U。对于任意外延模糊集合 $A \in L^X, x \in X$, 并且 $A(x) < 1$, 那么使用标记 $[x]_A^{\downarrow} = (A \cup \{A(x)^+ / x\})^{\downarrow \downarrow}$ 。函数描述如下:

函数 1 UpperNeighbors

输入:模糊概念的外延 A

输出:模糊概念的所有直接前序概念的外延集合 U

1. $U \leftarrow \emptyset$;
2. $\text{Min}(\{x \in X \mid A(x) < 1\})$;
3. For $x=1$ to n
4. If $A(x) < 1$
5. $E \leftarrow [x]_A^{\downarrow}$;

6. $\text{Increased} \leftarrow \{z \in X \mid z \neq x \text{ and } A(x) < E(z)\}$;
7. If $\text{Min} \cap \text{Increased} = \emptyset$
8. Add E to U;
9. Else
10. Remove z form Min;
11. End If
12. End If
13. End For

在基于约束的模糊概念格的构造过程中,根据约束条件判断模糊概念是否满足约束,进而生成基于约束的模糊概念格结构。根据本文第 2 节给出的约束定义,给出判断模糊概念节点是否满足约束的函数 Constraint,描述如下:

函数 2 Constraint

输入:模糊概念节点的内涵 B,约束条件 P

输出:是否满足约束 Flag

1. Switch(P) Do
2. Case $P=P_1$:
3. If $(k \in B \text{ and } \mu_k \geq \lambda_k)$
4. Flag = True;
5. Else Flag = False;
6. End If
7. Case $P=P_2$:
8. If $((k_1 \in B \text{ and } \mu_{k_1} \geq \lambda_{k_1}) \text{ and } (k_2 \in B \text{ and } \mu_{k_2} \geq \lambda_{k_2}))$
9. Flag = True;
10. Else Flag = False;
11. End If
12. Case $P=P_3$:
13. If $((k_1 \in B \text{ and } \mu_{k_1} \geq \lambda_{k_1}) \text{ or } (k_2 \in B \text{ and } \mu_{k_2} \geq \lambda_{k_2}))$
14. Flag = True;
15. Else Flag = False;
16. End If
17. End Switch

在函数 1 和函数 2 的基础上,现给出基于约束的模糊概念格 CFCL 构造算法。该算法中使用到以下变量: A^* 表示模糊概念节点外延 A 的直接前序集合; A_* 表示模糊概念节点外延 A 的直接后继集合; F 表示生成的满足约束条件的所有模糊概念的外延集合。

算法 CFCL

输入:模糊形式背景 (X, Y, I) , 约束条件 P

输出:满足约束的模糊概念的外延集 F, 格结构 $\{(A^*, A_*) \mid A \in F\}$

1. $F \leftarrow \emptyset$;
2. $A \leftarrow \emptyset^{\uparrow \downarrow}$;
3. Add A to F;
4. CREATEFROM(A);
5. CREATEFROM(A)
6. While $A \neq X$ do
7. If Constraint(A^*, P)
8. $A^* \leftarrow \text{UpperNeighbors}(A)$
9. $N \leftarrow A^* - F$;
10. For each $D \in A^*$ do
11. If Constraint(D^*, P)
12. Add A to D_* ;
13. If $D \in N$
14. Add D to F;

```

15.           End If
16.         End If
17.       End For
18.     For each D ∈ N do
19.       CREATFROM(D)
20.     End For
21. End While

```

算法 CFCL 分为两个部分:第 1 行—第 3 行给出了算法的初始条件,确定了底概念结点的外延;第 4 行调用了 *CREATFROM()* 函数,该函数的作用是生成满足约束条件的模糊概念节点外延,并生成格结构;第 7 行根据概念结点的内涵是否满足约束条件确定是否生成直接前序集合,根据推论和定理 3,当模糊概念结点的内涵不满足约束条件时,不生成该模糊概念的直接前序,从而减少了后续约束的重复判断,提高了建格效率。该算法构造出的是模糊概念格的外延集合,并生成格结构,对模糊概念格的外延做一次模糊伽罗瓦连接^[14]运算,即可构造完整的模糊概念格。

CFCL 算法在构造模糊概念格的过程中,当生成的所有模糊概念都满足约束条件时,是最坏的情况,必须对每一个模糊概念格节点做判断,将所有模糊概念格节点都遍历一遍,所以算法 CFCL 的时间复杂度为 $O(|X|^2 * |Y| * |E|)(|E|$ 为模糊概念格节点个数)。但是在算法 CFCL 中,通过第 7 行的约束判断,当模糊概念不满足约束条件时,不生成其直接前序节点,也不再对这些前序节点进行约束判断操作,模糊概念格节点的数量就会减少,因此算法的时间复杂度一定小于 $O(|X|^2 * |Y| * |E|)$ 。所以 CFCL 算法能有效地节约模糊概念格的存储空间和建格时间。

4 实验设计与分析

实验环境:3.4GHz CPU,4GB 内存,Win7 操作系统,用 MATLAB 实现了 CFCL 算法。实验目的:测试基于约束的模糊概念格构造算法 CFCL,进一步分析该算法的特性。实验数据为随机产生的稀疏数据集和稠密数据集,具体说明如下。

数据集 1:稀疏数据集。拥有 1000 个对象,10 个属性,平均非零项占 20%,属性值分别具有 3 值、5 值和 11 值,以对应 L_3 、 L_5 和 L_{11} 3 种精度。

数据集 2:稀疏数据集。拥有 1000 个对象,10 个属性,平均非零项占 30%,属性值分别具有 3 值、5 值和 11 值,以对应 L_3 、 L_5 和 L_{11} 3 种精度。

数据集 3:稠密数据集。拥有 100 个对象,10 个属性,平均非零项占 70%,属性值分别具有 3 值、5 值和 11 值,以对应 L_3 、 L_5 和 L_{11} 3 种精度。

数据集 4:稠密数据集。拥有 100 个对象,10 个属性,平均非零项占 80%,属性值分别具有 3 值、5 值和 11 值,以对应 L_3 、 L_5 和 L_{11} 3 种精度。

实验采用的 3 种不同精度的真值集合 $L: L_3 = \{0, 0.5, 1\}$, $L_5 = \{0, 0.25, 0.5, 0.75, 1\}$, $L_{11} = \{0, 0.1, 0.2, \dots, 0.9, 1\}$ 。

4.1 CFCL 算法完备性验证

表 2—表 5 分别为数据集 1—数据集 4 在不同精度下,通过 Belohlavek 的 Lattice 算法^[18]和 $P = \emptyset$ 时,CFCL 算法构造

的模糊概念格节点数量的对比。

表 2 数据集 1 在不同精度下通过不同算法构造的模糊概念格节点数量

	L_3	L_5	L_{11}
Lattice	627	2931	8815
CFCL	627	2931	8815

表 3 数据集 2 在不同精度下通过不同算法构造的模糊概念格节点数量

	L_3	L_5	L_{11}
Lattice	1209	7954	37516
CFCL	1209	7954	37516

表 4 数据集 3 在不同精度下通过不同算法构造的模糊概念格节点数量

	L_3	L_5	L_{11}
Lattice	1002	9412	103309
CFCL	1002	9412	103309

表 5 数据集 4 在不同精度下通过不同算法构造的模糊概念格节点数量

	L_3	L_5	L_{11}
Lattice	1638	10886	311725
CFCL	1638	10886	311725

由表 2—表 5 可以看出:对于稀疏数据集和稠密数据集,在相同的模糊形式背景下,CFCL 算法在约束条件为空即无约束时,构造出的模糊概念格与 Belohlavek 的 Lattice 算法构造出的模糊概念格是相同的,从而验证了 CFCL 算法的完备性。

4.2 CFCL 算法在单约束条件下的实验结果

根据本文约束的定义,用户约束的是模糊概念节点的属性及其隶属度。由于模糊概念的属性隶属度与数据集精度有关,下面实验以精度 L_{11} 的数据集为例,其他精度在此不一一列举。

图 5、图 6 显示了稀疏数据集 1 和稀疏数据集 2 在精度为 L_{11} 、对于相同的模糊形式背景、对 3 个任意属性的不同单约束条件下,CFCL 算法构造的模糊概念格节点数目的比较。

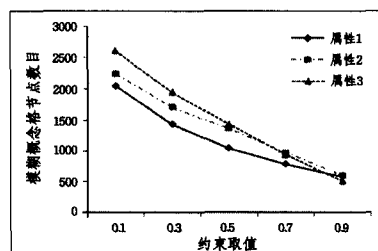


图 5 数据集 1 在不同单约束条件下构造的模糊概念格节点数目

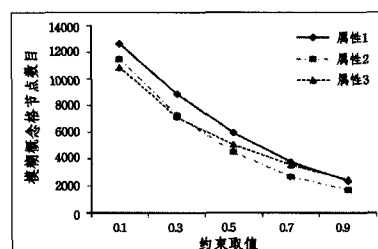


图 6 数据集 2 在不同单约束条件下构造的模糊概念格节点数目

图 7、图 8 显示了稠密数据集 3 和稠密数据集 4 在精度为

L_{11} 、对于相同的模糊形式背景、对 3 个任意属的性质不同单约束条件下,CFCL 算法构造的模糊概念格节点数目的比较。

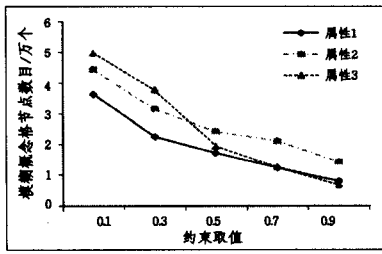


图 7 数据集 3 在不同单约束条件下构造的模糊概念格节点数目

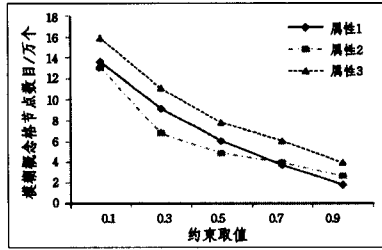


图 8 数据集 4 在不同单约束条件下构造的模糊概念格节点数目

由图 5—图 8 可以看出,对于稀疏数据集和稠密数据集,在相同的模糊形式背景下,约束条件不同时,CFCL 算法构造出的模糊概念格节点数目也不相同。对于同一属性,约束属性隶属度的取值越大,约束的程度越强,构造的满足约束的模糊概念格节点数目越少;约束属性隶属度的取值越小,约束的程度越弱,构造的满足约束的模糊概念格节点数目越多。同时,从图中还可以观察到,对于不同的属性,当约束隶属度的取值相同时,构造的满足约束的模糊概念格节点数目也不相同。

4.3 CFCL 算法在与约束条件下的实验结果

对于稀疏数据集,选取数据集 2 为例;对于稠密数据集,选取数据集 4 为例,精度均为 L_{11} 。图 9 和图 10 分别显示了数据集 2 和数据集 4 在两个任意属性上的不同与约束条件下,CFCL 算法构造的模糊概念格节点数目的比较。

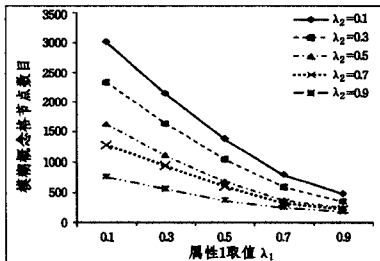


图 9 数据集 2 在不同与约束条件下构造的模糊概念格节点数目

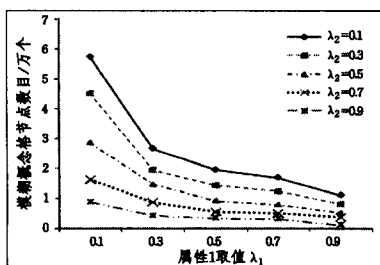


图 10 数据集 4 在不同与约束条件下构造的模糊概念格节点数目

由图 9 和图 10 可以看出,对于稀疏数据集和稠密数据集,当属性 2 的约束取值确定时,随着属性 1 的取值逐步增大,构造的模糊概念格节点数目递减;同样地,当属性 1 的约束取值确定时,随着属性 2 取值的逐步减小,构造的模糊概念格节点数目递增。也就是说,在相同的模糊形式背景下,针对两个属性的与约束,约束两个属性的隶属度取值越大,约束的程度越强,构造的满足约束的模糊概念格节点数目越少;约束两个属性的隶属度取值越小,约束的程度越弱,构造的满足约束的模糊概念格节点数目越多。

4.4 CFCL 算法在或约束条件下的实验结果

对于稀疏数据集,选取数据集 2 为例;对于稠密数据集,选取数据集 4 为例,精度均为 L_{11} 。图 11 和图 12 分别显示数据集 2 和数据集 4 在两个任意属性上的不同或约束条件下,CFCL 算法构造的模糊概念格节点数目的比较。

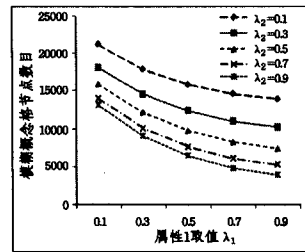


图 11 数据集 2 在不同或约束条件下产生的模糊概念格节点数目

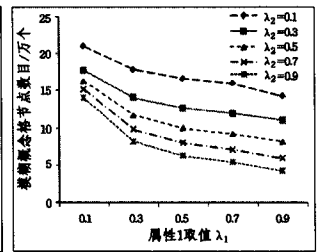


图 12 数据集 4 在不同或约束条件下产生的模糊概念格节点数目

由图 11 和图 12 可以看出,对于稀疏数据集和稠密数据集,在相同的模糊形式背景下,针对两个属性的或约束,约束两个属性的隶属度取值越大,约束的程度越强,构造的满足约束的模糊概念格节点数目越少;约束两个属性的隶属度取值越小,约束的程度越弱,构造的满足约束的模糊概念格节点数目越多。

4.5 CFCL 算法在不同约束条件下的实验结果

针对单约束、与约束和或约束的实验,均是对同一种约束条件的实验对比。下面给出算法 CFCL 在不同种约束条件下的实验结果,由于约束的隶属度取值灵活,在此仅以下列约束为例。表 6 和表 7 分别列出了稀疏数据集 2 和稠密数据集 4 在精度为 L_{11} 及不同的约束条件下,CFCL 算法构造出的模糊概念格节点数目及建格时间的比较。

表 6 数据集 2 在不同约束条件下的实验结果

约束条件	模糊概念格节点数目	建格时间/s
\emptyset	37516	3855
$(\lambda_1=0.1) \vee (\lambda_2=0.5)$	15785	1789
$\lambda_1=0.1$	12652	1301
$\lambda_2=0.5$	4510	489
$(\lambda_1=0.1) \wedge (\lambda_2=0.5)$	1377	174

表 7 数据集 4 在不同约束条件下的实验结果

约束条件	模糊概念格节点数目	建格时间/s
\emptyset	311725	5473
$(\lambda_1=0.3) \vee (\lambda_2=0.7)$	118985	2489
$\lambda_1=0.3$	91651	1896
$\lambda_2=0.7$	39577	908
$(\lambda_1=0.3) \wedge (\lambda_2=0.7)$	12243	328

(下转第 318 页)

[12] 于焯,陆建华,郑君里.一种新的彩色图像边缘检测算法[J].清华大学学报(自然科学版),2005,45(10):1339-1343
 Yu Ye, Lu Jian-hua, Zheng Jun-li. Color image edge detection algorithm[J]. Journal of Tsinghua University(Science and Tech-

nology), 2005, 45(10): 1339-1343

[13] Comanicu D, Ramesh V, Meer P. Real-Time Tracking of Non-Rigid Objects using Mean Shift[C]//IEEE Computer Vision and Pattern Recognition. 2000;142-149

(上接第 293 页)

对于不同的约束条件,约束程度由弱到强依次是:或约束,单约束,与约束;对于同一约束条件,约束属性隶属度的取值越大,约束程度越强。由表 6 和表 7 可以看出,约束条件从上到下约束程度依次增强。对于稀疏数据集和稠密数据集,在相同的模糊形式背景下,随着约束程度的增强,构造的满足约束的模糊概念格节点数目减少,建格时间明显减少;反之,随着约束程度的减弱,构造的模糊概念格节点数目增加,建格时间也增加。

综合上面的实验结果可以看出,本文提出的 CFCL 算法在不同的约束条件下,能够构造满足用户不同需求的概念格,同时能够不同程度地节约模糊概念格的存储空间和建格时间,尤其在约束程度较强时效果更为突出。

结束语 当应用模糊概念格进行知识提取时,一些模糊概念格中节点并不满足用户的需求,同时随着处理数据的增多,构造模糊概念格的时空复杂度呈指数级增长。本文将用户的需求定义为约束,研究了一种基于约束的模糊概念格构造算法。该算法适用于文中定义的约束,利用约束指导模糊概念格的构造过程,使构造的模糊概念格更具针对性,并通过实验验证了该算法能够节约模糊概念格的存储空间和建格时间。本文定义的约束表达范围有限,能满足用户关心的属性隶属度大于某一取值的需求,下一步的工作是扩展约束条件的表示范畴,使得约束能够满足用户更多的需求,进而构造满足约束的模糊概念格。在此基础上,如何进一步提高模糊概念格的构造效率也是未来的工作方向。

参 考 文 献

[1] Ganter B, Wille R. Formal Concept Analysis[M]. Berlin, Heidelberg; Springer, 1999

[2] 柴玉梅,王春丽,王黎明.基于频繁项集的互补替代关系挖掘算法[J].模式识别与人工智能,2012,25(1):157-165
 Chai Yu-mei, Wang Chun-li, Wang Li-ming. An Algorithm for Mining Complement-Alternative Relationship Based on Frequent Itemsets [J]. Pattern Recognition and Artificial Intelligence, 2012, 25(1): 157-165

[3] 柴玉梅,张卓,王黎明.基于频繁概念直乘分布的全局闭频繁项集挖掘算法[J].计算机学报,2012,35(5):990-1001
 Chai Yu-mei, Zhang Zhuo, Wang Li-ming. An Algorithm for Mining Global Closed Frequent Itemsets Based on Distributed Frequent Concept Direct Product[J]. Chinese Journal of Computers, 2012, 35(5): 990-1001

[4] 王黎明,张卓.基于 Iceberg 概念格并置集成的闭频繁项集挖掘算法[J].计算机研究与发展,2007,44(7):1184-1190
 Wang Li-ming, Zhang Zhuo. An Algorithm for Mining Closed Frequent Itemsets Based on Apposition Assembly of Iceberg Concept Lattices [J]. Journal of Computer Research and Development, 2007, 44(7): 1184-1190

[5] Young P. Software Retrieval by Samples Using Concept Analysis [J]. The Journal of Systems and Computer, 2000, 54(3):

179-183

[6] Belohlavek R, Sklenar V. Formal Concept Analysis Constrained by Attribute-Dependency Formulas[C]//Proc of the 3rd International Conference on ICFA. Lens, France, 2005;176-191

[7] Belohlavek R, Vychodil V. Formal Concept Analysis with Constraints by Closure Operators[C]//Proc of 14th International Conference on Conceptual Structures (ICCS 2006). Aalborg, Denmark, 2006;131-143

[8] Belohlavek R, Vychodil V. Formal concept analysis with background knowledge; Attribute priorities [J]. IEEE Transactions on Systems, 2009, 39(4): 399-409

[9] 张继福,张素兰,蒋义勇.约束概念格的代数性质及其知识表示的完备性[J].模式识别与人工智能,2010,23(3):289-299
 Zhang Ji-fu, Zhang Su-lan, Jiang Yi-yong. Algebraic Properties of Constrained Concept Lattice and Its Completeness of Knowledge Representation [J]. Pattern Recognition and Artificial Intelligence, 2010, 23(3): 289-299

[10] Belohlavek R. What is a Fuzzy Concept Lattice? [C]//Proc of the 13th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing. Moscow, Russia, 2011; 19-26

[11] Burusco A, Fuentes-Gonzalez R. The Study of the L-Fuzzy Concept Lattice [J]. Mathware & Soft Computer, 1994, 1(3): 209-218

[12] Fan Shi-Qing, Zhang Wen-Xiu, Xu Wei. Fuzzy Inference Based on Fuzzy Concept Lattice [J]. Fuzzy Sets and Systems, 2006, 157(24): 3177-3187

[13] 宋笑雪,张文修,李红.变精度对象概念格的构造及其性质[J].计算机科学,2010,37(12):197-214
 Song Xiao-xue, Zhang Wen-xiu, Li Hong. Construction and Properties of Variable Threshold Object-oriented Concept Lattices [J]. Computer Science, 2010, 37(12): 197-214

[14] Belohlavek R. Fuzzy Galois Connections[J]. Mathematical Logic Quarterly, 1999, 45(4): 497-504

[15] Belohlavek R. Algorithms for Fuzzy Concept Lattices[C]//Proc of the 4th International Conference on Recent Advances in Soft Computing. Nottingham, UK, 2002; 200-205

[16] Belohlavek R, De Baets B, Outrata J, et al. Computing the Lattice of All Fixpoints of a Fuzzy Closure Operator [J]. IEEE Transactions on Fuzzy Systems, 2010, 18(3): 546-557

[17] 张卓,柴玉梅,王黎明,等.模糊形式概念并行构造算法[J].模式识别与人工智能,2013,26(3):260-269
 Zhang Zhuo, Chai Yu-mei, Wang Li-ming, et al. A Parallel Algorithm Generating Fuzzy Formal Concepts[J]. Pattern Recognition and Artificial Intelligence, 2013, 26(3): 260-269

[18] Belohlavek R, Vychodil V. Reducing The Size of Fuzzy Concept Lattices by Hedges[C]//The 14th IEEE International Conference on Fuzzy Systems. Reno, Nevada, USA, 2005; 663-668

[19] Belohlavek R, Vychodil V. Reducing The Size of Fuzzy Concept Lattices by Fuzzy Closure Operators[C]//ISIS 2006. Tokyo, Japan, 2006; 309-314