

基于属性消减的模糊概念格渐进式构造算法

王黎明 姜 琴 张 卓

(郑州大学信息工程学院 郑州 450051)

摘 要 当前模糊概念格的直接构造具有指数时间复杂度,且随着真值集合 L 大小的增加,模糊概念格的规模变得越来越大。为此提出了 FMBUAD 算法,它能够在原有模糊概念格的基础上消去多个消减属性(冗余或者无效属性)得到新的模糊概念格,且不考虑真值集合 L 的大小。基于模糊概念格的基础理论证明了 FMBUAD 算法的正确性。该算法首先将所有概念节点内涵中的消减属性隶属度移除;然后找出模糊概念格中所有的删除节点;最后集中处理删除节点父子节点之间的偏序关系。理论证明和实验结果表明:FMBUAD 算法构造 L -模糊概念格具有较好的时间性能。

关键词 模糊概念格构造,消减属性,隶属度,广度优先,渐进式构造

中图法分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.8.044

Incremental Algorithm for Constructing Fuzzy Concept Lattices Based on Attributes Decrement

WANG Li-ming JIANG Qin ZHANG Zhuo

(School of Information Engineering, Zhengzhou University, Zhengzhou 450051, China)

Abstract Constructing fuzzy lattices directly is always with an exponential time complexity currently. With the increase of the set of truth values L , the size of fuzzy concept lattice becomes larger and larger. This paper proposed an algorithm called FMBUAD. A new fuzzy concept lattice can be gotten by removing the membership degree of deleting attributes which become redundant and useless in original concepts, without reconstructing the fuzzy concept lattice from scratch and considering the set of truth values. We proved the correctness of FMBUAD based on the theoretical basis of fuzzy concept lattices. In order to update the original fuzzy concept lattice with the deleting attributes, FMBUAD removes the membership degree of deleting attributes of all concepts firstly. Secondly, it visits the fuzzy lattice to identify all of the nodes which have to be deleted. Finally, the algorithm has to deal with the partial order between the deleted nodes' parents and children. The theory and experimental results show that FMBUAD has excellent performance for saving time.

Keywords Fuzzy concept lattice, Deleting attributes, Membership degree, Breadth first search, Incremental construction

1 引言

形式概念分析(FCA)^[1]提供了一种数学的方法来分析数据,这些数据是以形式背景^[1]的方式表示的。概念格作为 FCA 的核心数据结构,已经在信息检索、数据挖掘^[2]、机器学习等领域得到应用。概念格分为经典概念格^[1,3]与模糊概念格^[4-7],模糊概念格比经典概念格具有更广泛的应用,提高模糊概念格的构造效率更有实际意义。

由于经典概念格的研究已经相对成熟,很多研究人员将经典概念格算法运用在模糊概念格的构造上。Belohlavek 提出了将模糊形式背景转化为经典形式背景,运用经典概念格的构造算法构造概念格,这种简化使得每个模糊概念格都可被看成经典概念格^[5]。Belohlavek 在文献[6]中将 Ganter 的 NextClosure 算法^[1]用文献[5]简化的方法构造模糊概念。但是这种简化的方法带来了另外一个弊端,即构造效率比较低,研究人员通过将经典概念格的构造算法进行改进,可以直接

构造模糊概念格^[6,7]。文献[6]将 NextClosure 算法^[1]进行改进,可以直接在模糊形式背景构造模糊概念。文献[7]通过设定分级属性,将 Lindig 的 Lattice 算法^[8]进行扩展从而构造模糊概念格。当前研究也集中在根据模糊特性构造模糊概念格^[9,10]。文献[9]通过研究 Popescu 提出的构造模糊概念格的方法与 Krajci 方法的等价性,给出了一种在不同类情况下直接运用 Popescu 方法产生模糊概念格的方法。文献[10]用一个模糊图形表示模糊形式背景,通过寻找图中的所有最小非连接特征来得到模糊概念格。知识约简也是目前概念格领域的研究热点之一^[11-14]。文献[11]提出了在精确的二元关系与模糊的二元关系共同构成的模糊决策形式背景的基础上,通过定义 (α, β) 约简找出协调集和冗余的属性集。文献[12]通过 K-means 聚类算法减小概念格的规模。文献[13]根据某个真值度 σ 将属性划分为 3 类,通过属性类型进行属性约简以简化模糊概念格。文献[14]将直觉模糊理论引入到概念格中,并根据可辨识矩阵和可识别函数进行模糊概念格的属

到稿日期:2015-07-23 返修日期:2016-01-07 本文受国家青年科学基金项目(61303044)资助。

王黎明(1963—),男,博士,教授,CCF 高级会员,主要研究方向为现代软件工程技术、分布式人工智能和数据挖掘等;姜 琴(1989—),女,硕士,CCF 会员,主要研究方向为形式概念分析及应用、数据挖掘等;张 卓(1978—),男,博士,讲师,主要研究方向为形式概念分析及应用等, E-mail: jiezhangzhuo@zzu.edu.cn(通信作者)。

性约简。这些方法是从形式背景(数据)本身的特性进行知识约简,并没有从已有的格结构和概念节点隶属度之间的关系去分析并且构造新的模糊概念格。张磊等人^[15,16]研究了消去一个消减属性或者一个对象时,如何在原有经典概念格的基础上得到新的经典概念格。为此我们在张磊等人^[15]研究的基础上提出了 FMBUAD 算法,它是 BUAD^[15]算法的扩展。FMBUAD 算法能在原有模糊概念格的基础上消去任意多个消减属性而得到新的模糊概念格。该算法的执行过程与真值集合 L 无关,只与模糊概念格的初始规模^[17,18]和消减属性的个数有关,在很大程度上提高了模糊概念格的构造效率。

本文提出的 FMBUAD 具有以下特点:1)在原始模糊概念格的基础上,通过消去内涵中消减属性的隶属度以及对概念节点类型的判断直接得到新的模糊概念格,且并不产生新的模糊概念(概念节点的外延没有变化);2)对于一个给定的模糊概念格,由属性消减得到新的模糊概念格,这个算法的执行过程并不关心真值集合 L 的精度,即算法只关心消减属性的个数以及模糊概念节点的内涵,该算法也可以认为是模糊概念格的维护算法。

2 相关理论

2.1 模糊概念格的基本理论

定义 1^[4] (模糊闭包运算) 模糊形式背景是一个四元组 $\langle L, X, Y, I \rangle$, $L = \{0 = a_0, a_1, \dots, a_n = 1\}$ ($a_0 < \dots < a_n$), L 是由真值集构成的完备剩余格, X 为对象集, Y 为属性集, 二元关系 $I \in L^{X \times Y}$ 。对 $\forall A \in L^X, \forall B \in L^Y$, 引入运算 $\uparrow: L^X \rightarrow L^Y$ 与 $\downarrow: L^Y \rightarrow L^X$

$$A^\uparrow(y) = \bigwedge_{x \in X} (A(x) \rightarrow I(x, y)) \quad (1)$$

$$B^\downarrow(x) = \bigwedge_{y \in Y} (B(y) \rightarrow I(x, y)) \quad (2)$$

其中, $A^\uparrow(y)$ 的含义为模糊集合 A 中对象共有属性 y 的真值度, $B^\downarrow(x)$ 的含义为对象 x 拥有 B 中所有属性的真值度。因此, $CS(X, Y, I) = CS(K) = \{\langle A, B \rangle \mid A^\uparrow = B, B^\downarrow = A\}$ 被称为模糊概念格, $\langle A, B \rangle$ 被称为模糊概念。 $CS(K)$ 的偏序关系 $\leq$ 被定义为:

$$\langle A_1, B_1 \rangle \leq \langle A_2, B_2 \rangle \text{ 当且仅当 } A_1 \subseteq A_2 \text{ (当且仅当 } B_1 \supseteq B_2) \quad (3)$$

若 $\forall x \in X, A_1(x) < A_2(x)$, 则 $A_1 \subseteq A_2$ 。

定义 2^[4] 真值集合完备 L 具有完备剩余格 $L = (L, \wedge, \vee, \otimes, \rightarrow, 0, 1)$ 的代数结构, $L \subseteq [0, 1]$, 本文采用的 L 是卢卡谢维奇代数表, $\wedge$ 是取最小值, $\vee$ 是取最大值, $\otimes$ 与 $\rightarrow$ 被定义为:

$$a \otimes b = \max(0, a + b - 1) \quad (4)$$

$$a \rightarrow b = \min(1 - a + b, 1) \quad (5)$$

$CS(K)$ 上的概念有如下基本性质 ($\forall A, A_1, A_2 \in L^X, \forall B, B_1, B_2 \in L^Y$)。

性质 1^[4] $A_1 \subseteq A_2 \Rightarrow A_2^\uparrow \subseteq A_1^\uparrow, B_1 \subseteq B_2 \Rightarrow B_2^\downarrow \subseteq B_1^\downarrow$ 。

性质 2^[4] $A \subseteq (A^\uparrow)^\downarrow, B \subseteq (B^\downarrow)^\uparrow$ 。

2.2 模糊概念格属性消减基本理论

$CS(K)$ 是由模糊形式背景产生的模糊概念格, $C = \langle A, B \rangle$ 是 $CS(K)$ 中的一个概念, 其中 $A \in L^X, B \in L^Y$ 。文中 $B - Y_n$ 在与新产生的格中的节点比较时的含义为: $\forall y \in Y_n$, 将 $B(y)$ 从 B 中删除; $B - Y_n$ 在与原始概念格中的节点进行比较时的含义为: $\forall y \in Y_n$, 将 $B(y)$ 置 0。 $C_1 = \langle A_1, B_1 \rangle \in CS(K), B = B_1$ 应满足 $\forall y \in Y, B(y) = B_1(y), B \neq B_1$ 满足 $\exists y \in Y, B(y) \neq$

$B_1(y)$ 。将集合 Y_n 中包含的属性列从形式背景 K 中删除得到的形式背景记为 K^* , 其中 $K^* = K - Y_n$, 消去消减属性后得到的模糊概念格记为 $CS(K^*)$ 。

定义 3 (保留节点) 若 $C = \langle A, B \rangle \in CS(K), \forall y \in Y_n, B(y) = 0$, 则 C 被称为保留节点, 原始概念格 $CS(K)$ 中所有保留节点的集合记为 $RS_{-Y_n}(K)$ 。

定义 4 (更新节点) 若 $C \in CS(K), \forall C_1 \in CS(K) \neq C$, 都有 $Int(C_1) - Y_n \neq Int(C) - Y_n$, 则称 C 为更新节点, 形式背景 K 上所有更新节点的集合为 $MS_{-Y_n}(K)$ 。

定义 5 (删除节点) 若 $C \in CS(K), \exists C_1 \in CS(K) \neq C$, 满足 $Int(C_1) - Y_n = Int(C) - Y_n$, 则称 C 为删除节点, 形式背景 K 上所有删除节点的集合为 $DS_{-Y_n}(K)$ 。

由以上定义可知, $CS(K) = RS_{-Y_n}(K) \cup MS_{-Y_n}(K) \cup DS_{-Y_n}(K)$ 。

定义 6 (删除基节点) 若概念节点 $C_d = \langle A_d, B_d \rangle \in CS(K), \exists C_b = \langle A_b, B_b \rangle \in C_p$ 满足 $B_b - Y_n = B_d - Y_n$, 其中 C_p 为 C_d 的父节点, 则称 C_b 为删除基节点。

形式背景 $K^* = K - Y_n$, 其上的闭包映射记为: $\uparrow^*$ 和 $\downarrow^*$ 。

性质 3 $A^{\uparrow^*} = A^\uparrow - Y_n$ 。

证明: 形式背景 K^* 上的属性集 $Y^* = Y - Y_n, \forall y \in Y^*, A^{\uparrow^*}(y) = \bigwedge_{x \in X} (A(x) \rightarrow I(x, y)) = A^\uparrow(y)$ 。 $\forall y_1 \in Y_n, A^\uparrow - Y_n$ 为将 $A^\uparrow(y_1)$ 从 $A^\uparrow$ 中移除, 所以 $A^{\uparrow^*} = A^\uparrow - Y_n$ 。

性质 4 当 $\forall y \in Y_n, B(y) = 0$ 时, $B^{\downarrow^*} = B^\downarrow$ 。

证明: 由形式背景 K^* 的对象集 $X, \forall x \in X, B^{\downarrow^*}(x) = \bigwedge_{y \in Y - Y_n} (B(y) \rightarrow I(x, y)), B^\downarrow(x) = \bigwedge_{y \in Y} (B(y) \rightarrow I(x, y))$, 由于 $\forall y \in Y_n, B(y) = 0$, 再由式 (5) 可知, $\forall y \in Y_n, (B(y) \rightarrow I(x, y)) = 1$, 因此由 $B^\downarrow(x) = \bigwedge_{y \in Y - Y_n} (B(y) \rightarrow I(x, y))$ 得到 $B^{\downarrow^*} = B^\downarrow$ 。

定理 1 若 $C = \langle A, B \rangle \in MS_{-Y_n}(K)$, 则有 $(B - Y_n)^\downarrow = B^\downarrow = A$ 。

证明: $Y_n = \{y_1, y_2, \dots, y_n\} (n \leq \|Y\|)$, 用归纳法证明。1) 首先证明 $n = 1, Y_n = \{y_1\}$ 时, $(B - \{y_1\})^\downarrow = B^\downarrow$ 。根据文献 [15] 引理 1 的部分证明并结合定义 1 和定义 2 推知, $B - \{y_1\} \subseteq (B - \{y_1\})^{\downarrow\uparrow} \subseteq B^{\downarrow\uparrow} = B$, 由模糊概念格的性质可知, $(B - \{y_1\})^{\downarrow\uparrow}$ 为 $CS(K)$ 中某个概念的内涵。由于 $C \in MS_{-\{y_1\}}(K)$, 假设 $CS(K)$ 存在概念节点 $C_1 = \langle A_1, B_1 \rangle$, 其中 $B_1 = \{a_1, a_2, \dots, a_n\} (n = \|Y\|, a_1, a_2, \dots, a_n \in L)$ 满足 $a_1 = B_1(y_1) < B(y_1), \forall y \neq y_1, B_1(y) = B(y)$, 与 $C \in MS_{-\{y_1\}}(K)$ 矛盾, 即 $C_1 \notin CS(K)$ 且 $B - \{y_1\} \neq B - \{y_1\}^{\downarrow\uparrow}$, 则 $(B - \{y_1\})^{\downarrow\uparrow} = B$ 。由 $(B - \{y_1\})^\downarrow = (B - \{y_1\})^{\downarrow\uparrow\downarrow}$, 所以 $(B - \{y_1\})^\downarrow = B^\downarrow = A$ 。2) 假设若 $n = k, Y_k = \{y_1, y_2, \dots, y_k\}$, 则 $(B - Y_k)^\downarrow = B^\downarrow = A$ 成立。3) 若 $n = k + 1$, 则 $Y_{k+1} = \{y_1, y_2, \dots, y_{k+1}\}$, 因为 $C \in MS_{-Y_n}(K)$, 由定义 4 可知, 不存在 $C_1 = \langle A_1, B_1 \rangle \in CS(K)$ 使得 $B - Y_n = B_1 - Y_n$, 即也不存在 $C_2 = \langle A_2, B_2 \rangle \in CS(K)$ 使得 $B - \{y_1, y_2, \dots, y_k\} = B_2 - \{y_1, y_2, \dots, y_k\}$, 则 $C \in MS_{-Y_k}(K)$ 。由 2) 可知, 在消去消减属性集合 Y_k 后, 节点 $C = \langle A, B - Y_k \rangle \in MS_{-Y_k}(K)$, 在此基础上消去 $\{y_{k+1}\}$, 由 1) 可得 $(B - Y_k - y_{k+1})^\downarrow = (B - Y_k)^\downarrow$, 即 $(B - Y_n)^\downarrow = B^\downarrow = A$ 。

由定理 1 推出: 若 $C = \langle A, B \rangle \in RS_{-Y_n}(K)$, 则有 $(B - Y_n)^\downarrow = B^\downarrow = A$ 。

定理 2 设 $C_d = \langle A_d, B_d \rangle \in DS_{-Y_n}(K), C_b$ 为 C_d 的删除基节点, 若 $C = \langle A, B \rangle \in C_p$ 满足 $C \neq C_b$, 其中 C_p 为 C_d 的父节

点,则 $C \in DS_{-Y_n}(K)$ 。

证明:由于 C_p 可以同时删除多个消减属性,则删除基节点的类型并不唯一,删除基节点类型可以为保留节点、更新节点、删除节点。由 $CS(K) = RS_{-Y_n}(K) \cup MS_{-Y_n}(K) \cup DS_{-Y_n}(K)$ 可知,证明 $C \in DS_{-Y_n}(K)$, 仅需证明 $C \notin RS_{-Y_n}(K)$ 和 $C \notin MS_{-Y_n}(K)$ 即可,下文证明中概念节点内涵减去消减属性的含义为将消减属性隶属度对应位置置 0。

1) 证 $C \notin RS_{-Y_n}(K)$, 若 $C \in RS_{-Y_n}(K)$, 由 $C \in C_p$ 可知, $B \subseteq B_d$, 由于 B 中消减属性的隶属度为 0, 因此 $B - Y_n \subseteq B_d - Y_n$, 由于 $C_d \in DS_{-Y_n}(K)$, 因此 $B_d - Y_n = B_b - Y_n$, 所以 $B \subset B_d - Y_n \subset B_b - Y_n$, 则 $C > C_b$, 与 C 和 C_b 不存在偏序关系矛盾, 所以 $C \notin RS_{-Y_n}(K)$ 。

2) 证 $C \notin MS_{-Y_n}(K)$, 若 $C \in MS_{-Y_n}(K)$, 由 $C_d < C$, $C_d < C_b$ 可知, C 与 C_b 在 $CS(K)$ 中不存在偏序关系。由 $C_d < C \in C_p$ 可知, $B \subseteq B_d$, 则 $B - Y_n \subseteq B_d - Y_n = B_b - Y_n$ 。若 $C_b \in RS_{-Y_n}(K)$ 或 $C_b \in MS_{-Y_n}(K)$, 节点更新后, C_b 的内涵与外延不做任何改变, $C > C_b$ 与 C 和 C_b 不存在偏序关系矛盾; 若 $C_b \in DS_{-Y_n}(K)$, 由定理 6 可知, $CS(K)$ 中存在一个概念节点 $C_1 = (A_1, B_1)$ 是 C_b 的删除基节点, C_1 为更新节点或者保留节点。 $B - Y_n \subseteq B_d - Y_n = B_b - Y_n = B_1 - Y_n$, 节点更新后 $A \supset A_1$, B 为 B_1 的父节点, 由于更新节点前后, 节点的外延并没有发生改变, 则在节点更新前, $A \supset A_1 \supset A_b$, 与 C 和 C_b 不存在偏序关系矛盾, 因此 $C \notin MS_{-Y_n}(K)$ 。

定理 3 设 $K^* = K - Y_n, \forall C^* = (A^*, B^*) \in CS(K^*)$, 则 $\exists C = (A, B) \in CS(K)$ 满足 $A^* = A, B^* = B - Y_n$ 。

证明:由文献[15]引理 2 可知,若 K 为经典形式背景, Y_n 为单个消减属性 y , 定理 3 得证。下面沿用其证明思路, 证明 $\|Y_n\| \geq 1$ 时, K 为模糊形式背景的情形。 $\forall C^* = (A^*, B^*) \in CS(K^*)$ 满足 $A^{*\uparrow*} = B^*, B^{*\uparrow*} = A^*$, 其中 $(\uparrow*, \downarrow*)$ 为 $CS(K^*)$ 中的模糊运算。设 $A = A^{*\uparrow\downarrow}, B = A^{*\uparrow}$, 由于 $A^* \subseteq L^X$, 再由文献[4]可知, $(A, B) \in CS(K)$ 。

1) 证 $B^* = B - Y_n$, 由性质 3 可知, $A^{*\uparrow*} = A^{*\uparrow} - Y_n \Rightarrow B^* = B - Y_n$ 。

2) 证 $A^* = A$, 由性质 2 可知, $A^* \subseteq A^{*\uparrow\downarrow} = A$, 由 $A^{*\uparrow} \supseteq A^{*\uparrow*} = B^*$ 和性质 2 可知, $A = A^{*\uparrow\downarrow} \subseteq B^{*\uparrow} = B^{*\uparrow*} = A^*$, 所以 $A^* = A$ 。

定理 4 若 $K^* = K - Y_n$, 其中 K^* 为模糊形式背景, 则集合 $RS_{-Y_n} \cup MS_{-Y_n}$ 中的概念节点和集合 $CS(K^*)$ 中的概念节点是一一对应的。

证明:由定理 1 可知, 若 $C = (A, B) \in RS_{-Y_n}(K) \cup MS_{-Y_n}(K)$, 则在新概念格 $CS(K^*)$ 中, 存在节点 $C^* = (A^*, B^*) \in CS(K^*)$, 满足 $B - Y_n = B^*, A = A^*$, 则说明 $CS(K)$ 中的保留节点与更新节点消去消减属性后存在于 $CS(K^*)$ 中。由定义 5 可知, 若 $C \in DS_{-Y_n}(K)$, 则 $\exists C_1 \in CS(K) \neq C$, 满足 $Int(C_1) - Y_n = Int(C) - Y_n$, 则模糊概念节点 C 需要从原始模糊概念格中删除, 否则在节点消去消减属性后, 集合 $CS(K^*)$ 中存在 C_1 , 它与 C 内涵相同而外延不同, 与模糊概念格的定义矛盾, 则 $CS(K^*)$ 中并不存在与 $C \in DS_{-Y_n}(K)$ 的节点对应的节点, 则定理 4 得证。

由定理 4 可知, 原始模糊概念格消去消减属性后得到的新的概念格中节点是由原始模糊概念格中的保留节点和更新节点得来的, 并没有新的模糊概念节点产生。定理 5 的证明

是基于模糊概念格的基础理论, 也适用于经典概念格的证明。

定理 5 设 $C_d = (A_d, B_d) \in DS_{-Y_n}(K)$, $Int1$ 为消减属性在 B_d 中的隶属度, C_1 为 C_d 的删除基节点, 若 $\exists C_b = (A_b, B_b), C_b \in Parent(Child(C_d))$, $Int2$ 为消减属性在 C_b 中的隶属度, 满足 $Int1 \subseteq Int2$, 则 $C_b \leq C_1$ 不成立。

证明:(反证法)若 $C_b \leq C_1$ 成立, Int 为 C_1 中所含有的消减属性隶属度集合, 由模糊概念格偏序关系的定义可知 $Int \subseteq Int1$ 。由 C_1 为 C_d 的删除基节点可知, $B_d = B_1 \cup Int1$, 由 $C_b \leq C_1$ 可知, $B_1 \subseteq B_b \Rightarrow B_1 \cup Int2 \subseteq B_b \cup Int2$, 由于 $Int1 \subseteq Int2$, 则 $B_d = B_1 \cup Int1 \subseteq B_1 \cup Int2 \subseteq B_b$, 因此有 $C_b \leq C_d$, 与 $C_b \in Parent(Child(C_d))$ 矛盾, 则定理 5 成立。

定理 6 若一个节点为删除节点, 则在原始格中能找到它的删除基节点。

证明:设 $C_d = (A_d, B_d) \in DS_{-Y_n}(K)$, 由删除节点的定义可知, 只有找到删除基节点 C_b , 才能判定其为删除节点, 若 C_b 也是删除节点, 其删除基节点为 C_1 且节点 C_b 先于 C_d 被删除, C_b 的删除基节点也是 C_d 的删除基节点, 因为 $Int(C_d) - Y_n = Int(C_b) - Y_n = Int(C_1) - Y_n$, 且总存在一个 C_1 为更新节点或者是保留节点, 则定理 6 得证。

2.3 FMBUAD 算法实例

模糊概念格的形式背景如表 1 所列。

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 绘制了由表 1 产生的模糊概念格, 如图 1(a) 所示, 消减属性集 $Y_n = \{y_1, y_2\}$, 消去消减属性集合产生的概念格如图 1(b) 所示。

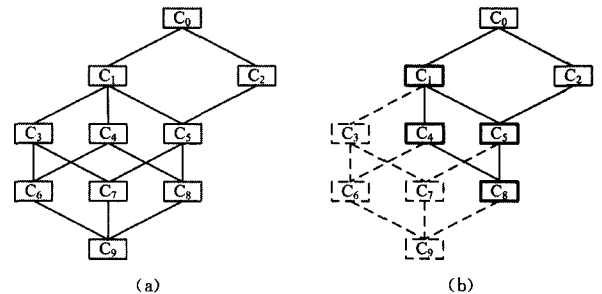


图 1 模糊概念格属性消减前后的对比

消减 Y_n 前的模糊概念:

$$C_0 = \langle \{x_1, x_2, x_3\}, \{0.5/y_3, 0.5/y_4, 0.5/y_5\} \rangle$$

$$C_1 = \langle \{x_1, x_2, 0.5/x_3\}, \{0.5/y_1, 0.5/y_2, 0.5/y_3, y_4, 0.5/y_5\} \rangle$$

$$C_2 = \langle \{x_1, 0.5/x_2, x_3\}, \{0.5/y_3, 0.5/y_4, y_5\} \rangle$$

$$C_3 = \langle \{x_1, x_2\}, \{y_1, 0.5/y_2, 0.5/y_3, y_4, 0.5/y_5\} \rangle$$

$$C_4 = \langle \{0.5/x_1, x_2, 0.5/x_3\}, \{0.5/y_1, 0.5/y_2, y_3, y_4, 0.5/y_5\} \rangle$$

$$C_5 = \langle \{x_1, 0.5/x_2, 0.5/x_3\}, \{0.5/y_1, 0.5/y_2, 0.5/y_3, y_4, y_5\} \rangle$$

$$C_6 = \langle \{0.5/x_1, x_2\}, \{y_1, y_2, y_3, y_4, 0.5/y_5\} \rangle$$

$$C_7 = \langle \{x_1, 0.5/x_2\}, \{y_1, 0.5/y_2, 0.5/y_3, y_4, y_5\} \rangle$$

$$C_8 = \langle \{0.5/x_1, 0.5/x_2, 0.5/x_3\}, \{0.5/y_1, 0.5/y_2, y_3, y_4, y_5\} \rangle$$

$$C_9 = \langle \{0.5/x_1, 0.5/x_2\}, \{y_1, y_2, y_3, y_4, y_5\} \rangle$$

消减 Y_n 后的模糊概念:

$$C_0 = \langle \{x_1, x_2, x_3\}, \{0.5/y_3, 0.5/y_4, 0.5/y_5\} \rangle$$

$$C_1 = \langle \{x_1, x_2, 0.5/x_3\}, \{0.5/y_3, y_4, 0.5/y_5\} \rangle$$

$$C_2 = \langle \{x_1, 0.5/x_2, x_3\}, \{0.5/y_3, 0.5/y_4, y_5\} \rangle$$

$$C_4 = \langle \{0.5/x_1, x_2, 0.5/x_3\}, \{y_3, y_4, 0.5/y_5\} \rangle$$

$$C_5 = \langle \{x_1, 0.5/x_2, 0.5/x_3\}, \{0.5/y_3, y_4, y_5\} \rangle$$

$$C_8 = \langle \{0.5/x_1, 0.5/x_2, 0.5/x_3\}, \{y_3, y_4, y_5\} \rangle$$

图中的虚线框和虚线代表删除节点和删除的边,加粗的框为更新节点。由图 1 可知,新的模糊概念格是由原始概念格删除某些概念节点直接得到的。由消去消减属性前后模糊概念的对比可知,新概念格中的概念是由原始概念格中的某些概念节点的内涵消去消减属性的隶属度得到的,并且这些概念的外延并没有改变。

3 模糊概念格构造算法

在本文工作之前,我们研究了基于多属性消减的经典概念格构造算法,采用了自底向上和自顶向下两种方式遍历概念格。根据理论证明,模糊概念格属性消减构造算法与经典概念格构造算法单个属性消减^[15]的证明思路相似,但理论基础不同。基于这些工作,本文以自底向上的方式遍历原有模糊概念格中的节点,从而构造新的模糊概念格。

3.1 算法描述

基于属性消减的模糊概念格构造算法的思想是:对于概念格的每个节点 C , $\forall y \in Y_n$, 将 $\text{Int}(C)$ 中 y 的隶属度删除,并将删除的隶属度保留在单独的变量中,然后将 inf 节点放入队列集合 $Cset$ 中。 CP 为 inf 节点的父节点,若 $\exists C_p \in CP$, 满足 $\text{Int}(C_p) = \text{Int}(\text{inf})$, 则 inf 为删除节点, C_p 为 inf 的删除基节点,若其父节点保留的消减属性隶属度存在大于 0 的属性,则将其放入 $Cset$ 集合中。以此处理 $Cset$ 集合中的每个概念节点,最后统一处理删除节点父子节点之间边的关系。

算法 FMBUAD()

输入:原始模糊概念格 $L(K)$, 消减属性集合 Y_n

输出:删除消减属性集合后得到的格 $L(K - Y_n)$

```
1. begin
2. DealConcepts(L(K));
3. FindAllDeLConcepts();
4. DeleteDelConcepts();
5. end
```

算法共分为 3 个模块。函数 1 DealConcepts($L(K)$)是遍历整个概念格,将格中的每个概念节点内涵所含有的消减属性删除,并保存每个概念节点所含有的消减属性的隶属度。函数 2 FindAllDeLConcepts()查找概念格删除节点,并记录每个删除节点删除基节点。函数 3 DeleteDelConcepts()调整删除节点父子节点之间边的关系,并将删除节点从概念格中删除。

函数 1 DealConcepts()

输入:原始模糊概念格($L(K)$),消减属性集合 Y_n

输出:消去消减属性集合 Y_n 的概念格 $L'(K)$, $ReAll$ 集合

```
1. begin
2. ReAll={}; dealC={sup(L)};
3. while dealC≠∅ do
4. C=dealC.pop();
5. for each y in  $Y_n$ 
```

```
6. 取出 y 在  $\text{Int}(C)$  中的隶属度 a;
7. 添加 a 至 re 尾部
8. 移除 y 在  $\text{Int}(C)$  中的隶属度 a;
9. end for
10. 将 Re 添加至 ReAll 的尾部;
11. for each child in C. child
12. dealC.add(child);
13. end for
14. Re={};
15. end while
16. end
```

函数 1 中 $ReAll$ 记录的是所有概念节点消减属性的隶属度, Re 记录的是每个概念节点的消减属性的隶属度。

函数 2 FindAllDeLConcepts()

输入:概念格($L'(K)$)

输出:所有删除节点以及删除基节点

```
1. begin
2. 比较属性个数与对象个数的大小,并标记 mark //mark=1 则属性
   个数较多,mark=0 则对象个数较多
3. 找出格的最大下界节点 inf(内涵隶属度均为 1 的概念节点), Cset:=
   {inf};
4. while Cset≠∅ do
5. 取出 Cset 集合中的概念节点 C,比较  $\text{Int}(C)$  与父节点  $C_p$  内涵是
   否相等,若相等则将 C 放入 DelList 集合中,并将  $C_p$  放入 DbList
   中,若 C 的父节点中保存的消减属性的隶属度大于 0,则将这些
   父节点放入 Cset 集合中
6. end while
7. end
```

函数 2 中在格中按从下到上的顺序判断模糊概念格中节点的类型,将删除节点放入 $DelList$ 集合,并将其对应的删除基节点存入 $DbList$ 集合。第 2 行的原始属性个数是消去消减属性隶属度之前概念节点内涵的大小,比较原始概念属性个数与外延属性个数的大小,并作相应的标记,在函数 3 中用这个标记来提高整个算法的时间性能。

函数 3 DeleteDelConcepts()

输入:删除节点集合 $DelList$, 删除基节点集合 $DbList$ 和 mark 标记, $ReAll$ 集合

输出:消去删除节点后的模糊概念格

```
1. begin
2. for(int i=0; i<DelList.size(); i++)
3. delC=DelList.get(i);
4. DbC=DbList.get(i);
5. PofC={}; flag=1; currentC=delC;
6. while(DbC==null) do
7. if(PofC!=null)
8. currentC:=PofC.get(0);
9. PofC.pop(0);
10. end if
11. for each parent∈currentC. parent
12. if(Int(parent)==Int(delC))
13. DbC:=parent; break;
14. end if
15. PofC.add(parent);
16. end for
17. end while
18. for each ch∈Child(C)
19. cover:=True;
```

```

20.   for each cp∈Parent(ch)
21.       if(cp≠delC) then
22.           Flag=1;
23.           if(Int1⊆Int2 and Int2 不全为 0) then
24.               flag:=0; end if
25.           if(mark=1&flag=1&(Ext(cp)⊆Ext(DbC)))
26.               cover:=false; break;
27.           else if(flag=1&(Int(cp)⊆Int(DbC)))
28.               cover:=false; break;
29.           end if
30.       end if
31.   end for
32.   if(cover) then
33.       新增边 DbC→Ch; end if
34.   end for
35.   从 L(K)中移除并销毁节点 delC 和与它相连的边;
36. end for
37. end

```

函数 3 中 $Int1$ 为删除节点 $delC$ 的消减属性隶属度, $Int2$ 为删除节点子节点的父节点的消减属性隶属度, $Int1$ 和 $Int2$ 都是从 $ReAll$ 集合取出的, Int' 是消去消减属性隶属度之前的内涵。 $mark$ 标记的作用为: 如果属性个数较多, 比较概念节点对象的隶属度; 若对象个数较多, 比较概念节点属性的隶属度。第 3 行取出删除节点, 第 4 行取出删除基节点。第 6—17 行是在算法的执行过程中, 某些删除节点 C 的删除基节点的类型为删除节点且先于 C 删除, 可以利用定理 6 寻找新的删除基节点。第 18—36 行是在 BUAD^[15] 算法的基础上, 通过第 23, 24 行比较节点所含消减属性隶属度间的关系, 减少删除基节点与删除节点子节点的父节点之间是否含有偏序关系的判断次数。在第 25—29 行增加了一个 $mark$ 标记来提高了算法的性能。第 23, 24 行的依据可参见定理 5, 改进后的算法性能可参见第 4 节。

4 实验与分析

首先通过实验验证 FMBUAD 算法的正确性; 然后通过与其他算法的比较, 分析 FMBUAD 算法的运行效率。

4.1 完备性分析

最基本的验证模糊概念格构造算法正确性的方式是, 检验不同算法产生的概念格的所有概念节点个数是否相等、相应的外延和内涵是否相等以及上下界是否相等, 若这些条件都满足, 则算法是正确的。本文将 FMBUAD 算法与 Lattice 算法^[7]产生的概念格进行比较。FMBUAD 是在形式背景产生概念格的基础上消去所有消减属性; 而 Lattice 算法是形式背景消去前 6 个消减属性, 然后构造模糊概念格。运行结果如图 2 所示。

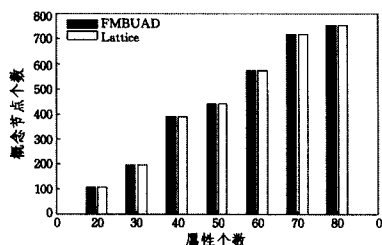


图2 FMBUAD算法的正确性验证

数据集 1: matlab 生成的随机数据, 对象个数为 10, 对象

个数从 20 增加到 80, 每次增加 10 个, 真值度集合为 $L_3 = \{0, 0.5, 1\}$, 消去 6 个消减属性。

由图 2 可知, 随着属性个数的增加, FMBUAD 消去消减属性集合后产生的概念格与 Lattice 在形式背景消去消减属性后产生的概念格个数相同。事实上, 我们在实验中也验证了概念节点的外延、内涵和上下界。实验表明: 本文提出的 FMBUAD 算法是正确的。

4.2 算法效率分析

由算法可知, 在消去多个消减属性构格时, 并不关心该模糊概念格真值集合 L 的大小和内容, 由于 L 的大小影响了原始概念格的规模^[17,18], 和其比较的算法^[6,7,15]的时间复杂度与概念格的真值集合 L 有关, 因此在以下实验中标出了概念格的 L 集合大小以及内容。本文基于 eclipse 平台用 java 实现了 FuNec 算法^[5]、Lattice 算法^[7]和 BUAD 算法^[15], 实验运行在单处理器多核 (Intel Xeon 8 Core) 计算环境下, 除操作系统以外, 无其它程序同时运行。

(1) 对于同一个概念格, 通过不断增加消减属性个数, 比较 FMBUAD、FuNec^[6]和 Lattice^[7]算法消去消减属性的性能。

FuNec^[6]和 Lattice 算法^[7]是在形式背景消去消减属性的情况下, 由形式背景中的数据重新构造模糊概念格; FMBUAD 是直接原始模糊概念格上消去消减属性。结果如图 3 所示。

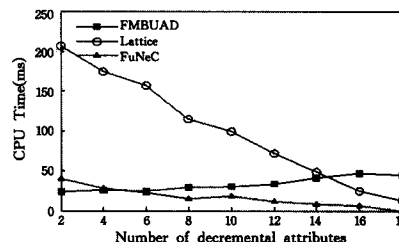


图3 $\|G\|=10, \|M\|=20$, FMBUAD, FuNec, Lattice 算法的性能比较

数据集 2: 由 matlab 生成的随机数据, 其中对象个数为 10, 属性个数为 20, 真值度集合 $L_5 = \{0, 0.25, 0.5, 0.75, 1\}$, 在格的大小为 339 个格节点。

由图 3 可知, 当消减属性个数小于 4 时, FMBUAD 的性能好于其它两个算法; 当消减属性个数超过 6 个时, FuNec 表现出较好的时间性能; 当消减属性个数超过 14 时, Lattice 的性能较好。由于 FuNec 并不产生格结构, 因此其时间性能优于其它两个算法。而当消减属性个数较多时 (如超过原有属性个数的 $2/3$ 时), 应利用 Lattice 算法直接由形式背景的数据构造格结构。

(2) 对于同规模的模糊概念格, 比较 FMBUAD 与 BUAD^[15]算法的性能之间。

FMBUAD 算法是 BUAD 算法的一个扩展, 也能处理精度为 L_2 的模糊概念格。在 FMBUAD 算法中, 组成概念的内涵和外延是由 0 和 1 构造的集合来表示, 0 表示该外延 (内涵) 不属于当前模糊集, 1 表示外延 (内涵) 属于当前模糊集。BUAD 算法中外延 (内涵) 是由其在形式背景中的位置表示的, 具体可参见文献^[15], 比较两个算法的性能对于处理由二元关系的形式背景构成的概念格以及决定用哪种数据结构表示概念节点有利于概念格后期的维护具有指导意义。在实验中, BUAD 是在原有概念格的基础上每次只减去一个消减属性得到新概念格, 然后在得到的新概念格的基础上再减一个,

如此往复一共减去 n 次 (n 等于消减属性的个数), 记录最终得到概念格的时间, 结果如图 4 所示。

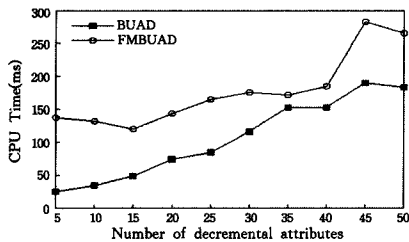


图 4 $\|L\|=3582$, FMBUAD 和 BUAD 算法之间的性能比较

数据集 3: 由 matlab 生成的随机数据, 对象个数为 80, 属性个数为 150, 密度为 0.3, 格的初始大小为 3582。

由图 4 可知, 对于同等规模的概念格, 随着消减属性个数的增加, FMBUAD 并没有表现出比 BUAD 更好的时间性能。由该实验可知, 若后期需消减大量无用的属性, 当由形式背景构造概念格时, 应考虑概念格的形式。若对象与属性之间的关系是单一的, 非 0 即 1, 则需构造经典概念格, 没有必要将概念节点表示成模糊概念的形式, 这样才更有利于概念格后期的维护。

(3) 比较利用 BUAD 算法^[15]结合文献^[5]构造模糊概念格和 FMBUAD 算法之间的时间性能。

任何一个模糊概念格 $B(X, Y, I)$ 都同构于一个经典概念格^[5] $B(X \times L, Y \times L, I^*)$, 其中 I^* 被定义为 $\langle \langle x, a \rangle, \langle y, b \rangle \rangle \in I^*$ 当且仅当 $a \otimes b \leq I(x, y)$ 。若构造一个模糊概念格 $B(X, Y, I)$, 构造经典概念格的算法需首先将模糊形式背景 $\langle X, Y, I \rangle$ 转化为 $\langle X \times L, Y \times L, I^* \rangle$, 然后采用经典概念格构造算法构造 $B(X \times L, Y \times L, I^*)$, 最后将经典概念格转化为 $B(X, Y, I)$ ^[6]。由于 BUAD 算法是在已有的经典概念格基础上运算的, 该部分实验假设已经利用 Godin 算法^[3]在模糊形式背景 $\langle X, Y, I \rangle$ 转化的 $\langle X \times L, Y \times L, I^* \rangle$ 背景中构造经典概念格 $B(X \times L, Y \times L, I^*)$, 然后利用 BUAD 算法在 $B(X \times L, Y \times L, I^*)$ 上逐步消去多个消减属性, 并记录该部分运算的时间, 最后利用文献^[5]中提供的方法将 $B(X \times L, Y \times L, I^*)$ 转换到模糊概念格 $B(X, Y, I)$ 。FMBUAD 算法是直接原有模糊概念格的基础上消去消减属性。结果如图 5 所示。

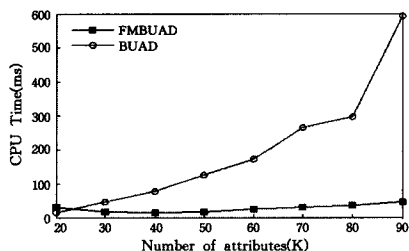


图 5 $\|G\|=300$ 时, BUAD 和 FMBUAD 算法之间的性能比较

数据集 4: 由 matlab 生成的随机数据, 形式背景对象个数为 20, 属性个数从 20 每次递增 10 到 90 为止, 密度为 0.4 (不为 0 的个数在整个形式背景中所占的比例), 构造初始概念格使用的真值度集合为 $L_3 = \{0, 0.5, 1\}$, 消减属性的个数为 10。

在图 5 中, 随着属性的增大, 原始概念格的规模逐渐增大, 实验中使用的消减属性的个数是固定的, BUAD 算法随着属性的个数增加消耗的时间越来越多。这是由于属性增加, $\|Y \times L\|$ 的个数就会增多, 在算法中寻找删除节点消耗

的时间就会增加, 因此导致整个算法的时间性能降低。当属性个数变化幅度不大时, FMBUAD 算法的时间性能变化不大, 有较好的时间性能。由于 FMBUAD 算法在整个算法的执行过程中是将消减属性的隶属度从概念节点内涵中移除的, 当属性个数的幅度变化较小时, 寻找删除节点消耗的时间只是有小幅度的增长。若删除节点个数增加不多, 则调整删除节点父节点和子节点之间的边所消耗的时间趋于稳定, 所以该实验中, FMBUAD 算法执行的时间趋于稳定。

(4) 比较 FMBUAD 和 Lattice 在 UCI 数据集上的时间性能。

实验结果如图 6 所示。

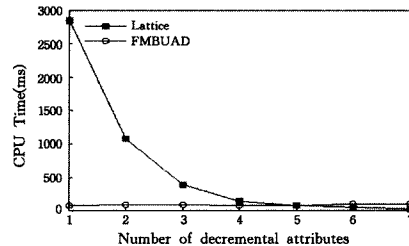


图 6 $\|G\|=1484$, $\|M\|=8$, FMBUAD 和 Lattice 算法的性能比较

数据集 5: UCI 数据集 Yeast^[19], 形式背景有 10 个属性, 8 个为实数值属性, 共有 1484 个事务, 所用的真值集合为 L_5 , 实验中只使用真值属性。

由图 6 可知, 在真实数据集 yeast 中, 当消减属性个数小于 5 时, FMBUAD 表现出了较好的时间性能; 当消减属性的个数超过 5 时, Lattice 算法的性能略好于 FMBUAD。

结束语 本文揭示了模糊概念格属性消减后概念集合和 Hasse 图的变化规律, 提出了 FMBUAD 算法构造新的模糊概念格, 实验结果和分析表明: 1) 该算法是有效的; 2) FMBUAD 具有较好的时间性能并能产生完整的 Hasse 图。在实际的应用领域中, 当处理小规模模糊概念格 (概念格的节点数量小于 100 万) 中某些无用的消减属性时, 可以利用本文提出的 FMBUAD 算法, 它能将多个无用的属性直接消去且比重新构造模糊概念格消耗的时间要少。随着概念格规模的日益增大, 以及多核计算环境的普及, 并行算法和并行计算提供了一种非常有效的方法来处理大规模的数据^[16,17]。张卓等人将模糊集合组合空间映射为自然数空间, 将搜索空间均分给各个并行线程的并行算法^[16], 并在文献^[17]中利用树状结构组织、遍历搜索空间, 逐级并行产生模糊概念、缩减搜索空间等, 同时通过重新划分搜索空间实现各个计算节点负载的均衡, 但文献^[16,17]并没有产生格结构。未来的工作主要集中在由若干计算机并行执行属性消减任务, 将原始格划分为若干个子格, 消去多个消减属性的任务由不同的计算节点完成, 以及实现各个计算节点负载均衡以进一步提高算法的效率且能产生完整的格结构。

参考文献

- [1] Ganter B, Wille R. Formal concept analysis. mathematical foundations[M]. Berlin: Springer-Verlag, 1999: 66-68
- [2] 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 (in Chinese)

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

- [3] Missaoui G R, Alaoui H. Incremental concept formation algorithms based on Galois (concept) lattices[J]. Computational Intelligence, 1995, 11(2): 246-267
- [4] Belohlávek R. Fuzzy Galois Connections[J]. Mathematical Logic Quarterly, 1999, 45(4): 497-504
- [5] Belohlavek R. Reduction and a simple proof of characterization of fuzzy concept lattices[J]. Fundamenta Informaticae, 2001, 46(4): 277-285
- [6] Belohlavek R. Algorithms for fuzzy concept lattice[C]// Proceedings of the 4th International Conference on Recent Advances in Soft Computing. Nottingham, United Kingdom, 2002: 200-205
- [7] 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
- [8] Lindig C. Fast Concept Analysis[C]// Working with Conceptual Structures, 2000. Aachen: Shaker Verlag, 2000: 152-161
- [9] Pócs J. Note on generating fuzzy concept lattices via Galois connections[J]. Information Sciences, 2012, 185(1): 128-136
- [10] Ghosh P, Kundu K, Sarkar D. Fuzzy graph representation of a fuzzy concept lattice[J]. Fuzzy Sets and Systems, 2010, 161(12): 1669-1675
- [11] Pei D, Li M Z, Mi J S. Attribute reduction in fuzzy decision formal contexts[C]// International Conference on Machine Learning and Cybernetics (ICMLC). IEEE Press, New York, 2011: 204-208
- [12] Aswanikumar C, Srinivas S. Concept lattice reduction using fuzzy

K-Means clustering [J]. Expert Systems with Applications, 2010, 37(3): 2696-2704

- [13] Li L, Zhang J. Attribute reduction in fuzzy concept lattices based on the T implication[J]. Knowledge-Based Systems, 2010, 23(6): 497-503
- [14] Pang J, Zhang X, Xu W. Attribute Reduction in Intuitionistic Fuzzy Concept Lattices [J]. Abstract and Applied Analysis, 2013, 2013(54): 1-13
- [15] Zhang Lei, Zhang Hong-li, Yin Li-hua, et al. Theory and Algorithms of Attribute Decrement for Concept Lattice[J]. Journal of Computer Research and Development, 2013, 50(2): 248-259 (in Chinese)
张磊, 张宏莉, 殷丽华, 等. 概念格的属性渐减原理与算法研究[J]. 计算机研究与发展, 2013, 50(2): 248-259
- [16] Zhang L, Zhang H, Shen X, et al. An Incremental Algorithm for Removing Object from Concept Lattice[J]. Journal of Computational Information Systems, 2013, 9(9): 3363-3372
- [17] Zhang Zhou, 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 (in Chinese)
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
- [18] Zhang Zhuo, Du Juan, Wang Li-ming. Load balance-based algorithm for parallelly generating fuzzy formal concepts[J]. Control and Decision, 2014, 29(11): 1935-1942 (in Chinese)
张卓, 杜鹃, 王黎明. 基于负载均衡的模糊概念并行构造算法[J]. 控制与决策, 2014, 29(11): 1935-1942
- [19] Frank A, Asuncion A. UCI machine learning repository [EB/OL]. <http://www.ics.uci.edu>

(上接第 203 页)

念的基础上, 采用双优先队列模型分别计算浏览请求和交易请求的优先级, 提出了基于请求分类和事前过载判断的动态优先级机制和调度策略, 并通过仿真实验将其与传统 FCFS 调度机制进行了对比研究, 结果表明, 在电子商务服务器高载情况下, 提出的多目标动态优先调度算法在有效提高交易请求达成率的前提下平均响应时间有所降低。下一步将深入研究基于预测的事前过载判断。

参 考 文 献

- [1] Loh S, Wives L K, de Oliveira J P. Concept-based knowledge discovery in texts extracted from the Web[J]. SIGKDD Explorations, 2000, 2(1): 29-39
- [2] Schroeder B, Harchol-Baher M. Web servers under overload: how scheduling can help[J]. ACM Trans. on Internet Technology, 2006, 6(1): 20-52
- [3] Menasce D A, Almeida V A F, Riedi R, et al. In search of Invariants for e-business workload[C]// Proceedings of 2000 ACM Conference in E-commerce, Minneapolis, MN, 2000: 17-20
- [4] Elnikety S, Nahum E, Tracey J, et al. A Method for Transparent Admission Control and Request Scheduling in E-commerce Web Sites[C]// Proceedings of the 13th International Conference on World Wide Web. New York, USA: ACM Press, May 2004: 276-286
- [5] Cherkasova L. Scheduling strategy to improve response time for

web applications[C]// Proceedings High Performance Computing and Networking. Amsterdam, Apr. 1998: 305-314

- [6] Guitart J, Carrera D, Beltran V, et al. Session-based adaptive overload control for secure dynamic web applications[C]// Proceedings of the 2005 International Conference on Parallel Processing. Oslo, Norway, Jun. 2005: 341-349
- [7] Singhmar N, Mathur V, Apte V, et al. A combined LIFO-priority scheme for overload control of e-commerce Web servers[C]// International Infrastructure Survivability Workshop. Lisbon, Portugal, 2006: 5-8
- [8] Doshi B T, Heffes H. Overload performance of several processor queuing disciplines for the M/M/1 queue[J]. IEEE Transactions on Communications, 1986, 34(6): 538-546
- [9] Cao Lin-qi, Xiao Xiao-qiang, et al. DPSP: A client request scheduling policy based on Web content[J]. Journal of Computer Research and Development, 2002, 139(12): 142-147 (in Chinese)
曹林奇, 肖晓强, 等. DPSP: 一种基于内容的客户请求调度策略[J]. 计算机研究与发展, 2002, 139(12): 142-147
- [10] Martin F A, Carey L W. Web server workload characterization: The search for invariants[C]// Proceedings of ACM SIGMETRICS 1996. Philadelphia, 1996: 126-137
- [11] Guan He-qing, Zhang Wen-bo, et al. An application-aware web service requests scheduling strategy [J]. Chinese Journal of Computers, 2006, 29(7): 1189-1198 (in Chinese)
官荷卿, 张文博, 等. 一种应用敏感的 Web 服务请求调度策略[J]. 计算机学报, 2006, 29(7): 1189-1198