

一种多线程概念约简算法

祁斌¹, 祁建军^{1,2}, 李俊安¹, 赵思雨³, 折延宏⁴

(1. 西安电子科技大学计算机科学与技术学院, 710071, 西安; 2. 新疆政法学院信息网络安全学院, 844000, 新疆图木舒克; 3. 咸阳师范学院数学与统计学院, 712000, 陕西咸阳;
4. 西安石油大学理学院, 710065, 西安)

摘要: 针对现有概念约简算法计算过程较为繁琐且执行效率低的问题,提出了一种多线程概念约简算法 MTCR。MTCR 以提高计算概念约简的效率为首要目标,在多核环境下利用多线程技术并行计算概念约简。首先,MTCR 算法使用两个线程分别计算单个对象的对象代表概念集和单个属性的属性代表概念集;然后,将形式背景中的对象(属性)依次放入 p 个队列,并为每个队列创建线程;最后,通过多线程方式并行计算任意对象的对象代表概念集和任意属性的属性代表概念集,以及两类代表概念集的交集,进而构建代表概念矩阵,据此计算出所有概念约简。在 MTCR 算法中,使用多线程的每个阶段的数据相对独立,使得多线程之间不需要频繁的同步操作,从而减少了线程之间的竞争和等待。这样可充分有效地利用计算资源,大大提升算法的性能。UCI 数据集和随机数据集上的实验表明:MTCR 算法可以准确得到概念约简结果,在使用单线程情况下执行速度与串行概念约简算法 SCR 相近;当线程数不超过 8 时,线程数每增加 1 倍,MTCR 算法执行速度可提高 30% 以上。

关键词: 形式概念分析;概念格;概念约简;多线程并行算法

中图分类号: TP18 **文献标志码:** A

DOI: 10.7652/xjtuxb202402017 **文章编号:** 0253-987X(2024)02-0164-08

A Multi-Thread Algorithm for Concept Reduction

QI Bin¹, QI Jianjun^{1,2}, LI Jun'an¹, ZHAO Siyu³, SHE Yanhong⁴

(1. School of Computer Science and Technology, Xidian University, Xi'an 710071, China;
2. School of Information Network Security, Xinjiang University of Political Science and Law, Tumushuke, Xinjiang 844000, China; 3. School of Mathematics and Statistics, Xianyang Normal University, Xianyang, Shaanxi 712000, China;
4. College of Science, Xi'an Shiyou University, Xi'an 710065, China)

Abstract: Aiming at the issue that the existing concept reduction algorithms are complex and the efficiency is low, a multi-thread concept reduction algorithm (MTCR) is proposed. MTCR takes improving the efficiency of concept reduction algorithms as the primary goal and employs multi-thread technology to parallelly calculate concept reduction in a multi-core environment. The MTCR algorithm initially uses two threads to calculate the object representative concept set of a single object and the attribute representative concept set of a single attribute, then puts the objects (attributes) into p queues in turn, and creates a thread for each queue. Finally, the object representative concept set of any object and the attribute representative concept set of any attribute are calculated parallelly using multi-threaded technology, and the representative concept matrix is

constructed by computing the intersection of these two different types of representative concept sets, on the basis of which all the concept reducts are computed. In the MTCR algorithm, the data of each stage using multiple threads is relatively independent, which avoids frequent synchronization operations between multiple threads, thereby reducing the competition and waiting time between threads. This mechanism enables the full and effective utilization of computing resources, which greatly improves the performance of the algorithm. Experiments on UCI datasets and random datasets show that the MTCR algorithm can accurately obtain all concept reduction, and the running time is close to the existing SCR algorithm when using a single thread. Specially, when the number of threads is less than 8, the running time of the MTCR algorithm can be improved by more than 30% for each doubling of the number of threads.

Keywords: formal concept analysis; concept lattice; concept reduction; multi-thread parallel algorithm

德国数学家 Wille 于 20 世纪八十年代为重构格理论提出了形式概念分析(formal concept analysis, FCA)^[1-2]。该理论自提出至今得到了很大的发展,已经突破了格论自身,被广泛应用到诸如数据挖掘^[3-4]、软件工程^[5-6]、知识工程^[7-8]、推荐系统^[9-10]和信息检索^[11-12]等领域,为解决知识发现问题的重要理论和工具之一。

概念约简是形式概念分析理论的一个新的研究方向,由魏玲等^[13-14]于 2018 年提出。概念约简的目标是获取保持二元关系不变的极小概念集,这样可以在完全不损失形式背景原始信息的情况下,减少概念数量,降低使用形式概念分析解决问题的复杂性。目前,概念约简算法相对较少,魏玲等^[14]在 2020 年提出了基于二元关系的概念约简算法。Zhao 等^[15]在文献[14]的基础上进一步研究,提出代表概念矩阵的定义,简明直观地展示了二元关系与概念之间的联系,并给出了基于代表概念矩阵的串行概念约简算法(serial concept reduction, SCR)。谢小贤等^[16]从布尔运算的角度研究了保持二元关系不变的概念约简。马文胜等^[17]针对计算部分概念约简的问题,提出同效关系,并给出通过同效关系求解概念约简的方法。李俊余等^[18]提出三元概念约简,可保持三元背景中所有的三元关系不变。王霞等^[19]提出一种基于概念可辨识矩阵对概念类别进行判定的方法。刘津等^[20]提出了基于概念可辨识矩阵的对偶三支概念约简方法。

SCR 算法将概念通过对象和属性分别聚类的方式构建代表概念矩阵,并使用代表概念矩阵计算概念约简,其时间复杂度低于文献[14]和文献[16]中概念约简算法的时间复杂度,较二者具有更快的执行速度,但 SCR 算法中并没有使用多线程方法,

执行效率仍然不高。针对此问题,提出一种多线程概念约简算法(multi-thread concept reduction, MTCR)。MTCR 算法使用构建代表概念矩阵的思想,通过两个线程分别对对象集和属性集单独处理,计算每个对象的对象代表概念集和每个属性的属性代表概念集,并以多线程并行计算方式获取任意对象的对象代表概念集和任意属性的属性代表概念集中共有概念,从而构建出代表概念矩阵。本文是在多核环境下运行多线程程序,通过使用多线程,提高概念约简算法的执行效率。

1 概念约简基础理论

先给出形式概念分析的基础知识。

定义 1^[1] 称 (G, M, I) 为一个形式背景,其中 $G = \{g_1, \dots, g_p\}$ 为对象集,每个 $g_i (i \leq p)$ 称为一个对象; $M = \{m_1, \dots, m_q\}$ 为属性集,每个 $m_j (j \leq q)$ 称为一个属性; I 为 G 和 M 之间的二元关系,即 $I \subseteq G \times M$ 。若 $(g, m) \in I$,则表示对象 g 具有属性 m ,也记为 gIm 。

例 1:表 1 是一个形式背景 (G, M, I) ,其对象集 $G = \{1, 2, 3, 4, 5\}$,属性集 $M = \{a, b, c, d, e\}$ 。表中将 $(g, m) \in I$ 记为 1, $(g, m) \notin I$ 记为 0。

表 1 一个形式背景 (G, M, I)
Table 1 A formal context (G, M, I)

对象	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
1	1	0	1	0	1
2	1	1	0	0	0
3	0	1	0	0	1
4	1	0	0	1	0
5	1	1	0	0	1

对于形式背景 (G, M, I) , 在对象子集 $X \subseteq G$ 和属性子集 $B \subseteq M$ 上可以定义一对对偶算子

$$X^* = \{m \mid m \in M, \forall g \in X, gIm\}$$

$$B^* = \{g \mid g \in G, \forall m \in B, gIm\}$$

其中 X^* 表示 X 中所有对象共同具有的属性集合, B^* 表示共同具有 B 中所有属性的对象集合。

对于任意的 $X \subseteq G, B \subseteq M$, 若 $X^* = B$ 且 $B^* = X$, 则称 (X, B) 为形式概念, 简称概念。其中 X 称为概念的外延, B 称为概念的内涵。

将形式背景 (G, M, I) 所有形式概念的集合记为 $L(G, M, I)$ 。对于任意的 $(X_1, B_1), (X_2, B_2) \in L(G, M, I)$, 定义如下的偏序关系

$$(X_1, B_1) \leq (X_2, B_2) \Leftrightarrow X_1 \subseteq X_2 (B_1 \supseteq B_2)$$

任意两个概念的下确界、上确界为

$$(X_1, B_1) \wedge (X_2, B_2) = (X_1 \cap X_2, (B_1 \cup B_2)^{**})$$

$$(X_1, B_1) \vee (X_2, B_2) = ((X_1 \cup X_2)^{**}, B_1 \cap B_2)$$

从而 $L(G, M, I)$ 形成一个完备格, 被称为概念格。

例2: 例1中形式背景构建的概念格如图1所示。为了方便, 本文中所有概念的外延、内涵都以罗列其中元素的序列形式给出, 并对所有概念进行编号。

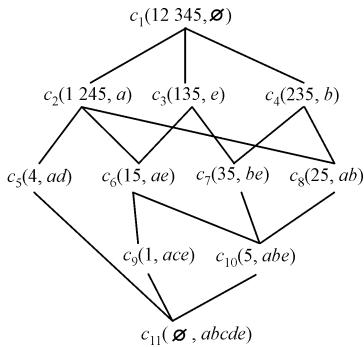


图1 $L(G, M, I)$

Fig. 1 $L(G, M, I)$

概念格虽然可以清晰地展示形式背景中概念之间的层次关系, 但是在知识还原方面, 概念格中的概念信息存在冗余, 而通过概念约简仅用部分概念就可以完全还原形式背景的全部信息。下面给出概念约简的相关定义。

定义2^[14] 设 (G, M, I) 是一个形式背景, $L(G, M, I)$ 是概念格, $F \subseteq L(G, M, I)$ 。如果 $I = \bigcup_{(X, B) \in F} (X \times B)$, 则称 F 为保持二元关系 I 不变的概念协调集; 若对任意的 $(X_i, B_i) \in F, F' = F \setminus \{(X_i, B_i)\}$, 有 $I \neq \bigcup_{(X, B) \in F'} (X \times B)$, 则称 F 为保持二元关系 I 不变的概念约简。

注 由文献[15]可知, 概念格中顶元与底元均不存在于概念约简中。因此, 这两种概念在之后的讨论中不予考虑。

定义3^[15] 设 (G, M, I) 为形式背景, $g_i \in G, m_j \in M$ 。如果存在 $(X, B) \in L(G, M, I)$, 满足 $g_i \in X$ 且 $m_j \in B$, 那么称 (X, B) 为 (g_i, m_j) 的一个代表概念。记 $\text{REP}((g_i, m_j)) = \{(X, B) \in L(G, M, I) \mid (g_i, m_j) \in X \times B\}$, 称 $\text{REP}((g_i, m_j))$ 为 (g_i, m_j) 的代表概念集; 记 $\mathbf{A} = (\text{REP}((g_i, m_j)), g_i \in G, m_j \in M)_{|G| \times |M|}$ 为代表概念集 $\text{REP}((g_i, m_j))$ 构成的矩阵, 称 $\mathbf{A}$ 为 (G, M, I) 的代表概念矩阵。

定义4 设 (G, M, I) 为形式背景, $g_i \in G, m_j \in M$ 。如果存在 $(X, B) \in L(G, M, I)$, 满足 $g_i \in X$, 那么称 (X, B) 为对象 g_i 的对象代表概念, 记 $\text{REP}(g_i) = \{(X, B) \in L(G, M, I) \mid g_i \in X\}$, 称 $\text{REP}(g_i)$ 为对象 g_i 的对象代表概念集; 若存在 $(X, B) \in L(G, M, I)$, 满足 $m_j \in B$, 那么称 (X, B) 为属性 m_j 的属性代表概念, 记 $\text{REP}(m_j) = \{(X, B) \in L(G, M, I) \mid m_j \in B\}$, 称 $\text{REP}(m_j)$ 为属性 m_j 的属性代表概念集。

根据对象 g_i 的对象代表概念集和属性 m_j 的属性代表概念集的定义, 可以得到如下性质。

性质1 设 (G, M, I) 为形式背景, $g_i \in G, m_j \in M$, 则有 $\text{REP}((g_i, m_j)) = \text{REP}(g_i) \cap \text{REP}(m_j)$ 。

例3: $L(G, M, I)$ 中每一个对象的对象代表概念集和每一个属性的属性代表概念集分别如表2和表3所示。

表2 例1中的 $\text{REP}(g_i)$

Table 2 $\text{REP}(g_i)$ in example 1

g_i	$\text{REP}(g_i)$
1	$\{c_2, c_3, c_6, c_9\}$
2	$\{c_2, c_4, c_8\}$
3	$\{c_3, c_4, c_7\}$
4	$\{c_2, c_5\}$
5	$\{c_2, c_3, c_4, c_6, c_7, c_8, c_{10}\}$

表3 例1中的 $\text{REP}(m_j)$

Table 3 $\text{REP}(m_j)$ in example 1

m_j	$\text{REP}(m_j)$
a	$\{c_2, c_5, c_6, c_8, c_9, c_{10}\}$
b	$\{c_4, c_7, c_8, c_{10}\}$
c	$\{c_9\}$
d	$\{c_5\}$
e	$\{c_3, c_6, c_7, c_9, c_{10}\}$

例1中代表概念矩阵 $\mathbf{A}$ 如下所示。

$$\mathbf{A} = \begin{bmatrix} \{c_2, c_6, c_9\} & \emptyset & \{c_9\} & \emptyset & \{c_3, c_6, c_9\} \\ \{c_2, c_8\} & \{c_4, c_8\} & \emptyset & \emptyset & \emptyset \\ \emptyset & \{c_4, c_7\} & \emptyset & \emptyset & \{c_3, c_7\} \\ \{c_2, c_5\} & \emptyset & \emptyset & \{c_5\} & \emptyset \\ \{c_2, c_6, c_8, c_{10}\} & \{c_4, c_7, c_8, c_{10}\} & \emptyset & \emptyset & \{c_3, c_6, c_7, c_{10}\} \end{bmatrix}_{5 \times 5}$$

矩阵共有 5 行 5 列。第 i 行第 j 列的代表概念集表示外延包含第 i 个对象 g_i , 同时内涵包含第 j 个属性 m_j 的所有代表概念。代表概念由对象 g_i 的对象代表概念集 $\text{REP}(g_i)$ 和属性 m_j 的属性代表概念集 $\text{REP}(m_j)$ 做交运算获得。例如代表概念矩阵第 1 行第 1 列的概念集合由对象 1 的对象代表概念集 $\{c_2, c_3, c_6, c_9\}$ 和属性 a 的属性代表概念集 $\{c_2, c_5, c_6, c_8, c_9, c_{10}\}$ 做交运算获得, 即满足外延包含对象 1 同时内涵包含属性 a 的所有概念为 $\{c_2, c_6, c_9\}$ 。空集 $\emptyset$ 表示不存在相应的代表概念。

为了得到概念约简, 可以将代表概念矩阵中的代表概念集罗列, 并在每个代表概念集中任取一个概念进行组合, 得到概念协调集, 其中极小的集合就是概念约简。具体方法由命题 1 给出。

命题 1^[15] 设 (G, M, I) 为形式背景, $L(G, M, I)$ 是其概念格, 对任意的 $(g_i, m_j) \in I$, 记代表概念函数 $f(\mathbf{A}) = \mathbf{A}_{\text{REP}(g_i, m_j)} \in \mathbf{A}(\bigvee_{(X, B) \in \text{REP}(g_i, m_j)} (X, B))$, 则 $f(\mathbf{A})$ 所对应的极小析取范式中的所有合取式为该形式背景所对应的概念约简全体。

在代表概念矩阵中可以很明显发现, 代表概念集之间可能存在包含关系。这种包含关系使代表概念矩阵中存在大量重复信息, 造成信息冗余。为了降低概念约简的计算复杂度, 提升概念约简执行效率, 可以通过删除包含元素, 保留被包含元素的方式, 将代表概念矩阵简化。简化后的代表概念矩阵称为最小代表概念矩阵。上面的代表概念矩阵简化结果如下

$$\mathbf{A}_{\min} = \begin{bmatrix} \emptyset & \emptyset & \{c_9\} & \emptyset & \emptyset \\ \{c_2, c_8\} & \{c_4, c_8\} & \emptyset & \emptyset & \emptyset \\ \emptyset & \{c_4, c_7\} & \emptyset & \emptyset & \{c_3, c_7\} \\ \emptyset & \emptyset & \emptyset & \{c_5\} & \emptyset \\ \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \end{bmatrix}_{5 \times 5}$$

由例 3 中最小代表概念矩阵 $\mathbf{A}_{\min}$, 可以得到 4 个概念约简结果为 $\{c_5, c_7, c_8, c_9\}$ 、 $\{c_2, c_4, c_5, c_7, c_9\}$ 、 $\{c_2, c_3, c_4, c_5, c_9\}$ 和 $\{c_3, c_4, c_5, c_8, c_9\}$ 。

2 MTCR 概念约简算法

2.1 SCR 概念约简算法

图 2 为使用 SCR 算法计算概念约简的流程图,

主要包含 4 个步骤。

(1) 计算概念: 将输入的形式背景通过概念格构建算法计算生成所有形式概念。

(2) 计算代表概念矩阵: 计算所有对象的对象代表概念集和所有属性的属性代表概念集, 并通过任意对象的对象代表概念集和任意属性的属性代表概念集中的共有概念构建代表概念矩阵。

(3) 计算最小代表概念矩阵: 判断代表概念矩阵中代表概念集之间的包含关系。若两个不同的代表概念集之间存在包含关系, 则删除包含的代表概念集, 保留被包含的代表概念集。若代表概念矩阵中任意两个代表概念集之间均不存在包含关系, 则该矩阵为最小代表概念矩阵。

(4) 计算概念约简结果: 计算最小代表概念矩阵中所有非空代表概念集极小析取范式中的所有合取式获得概念约简。

在上述 4 个步骤中, 因为第 1 步和第 4 步的算法已相对成熟^[21-25], 所以在概念约简中使用现有算法, SCR 算法只提出第 2 步和第 3 步的算法。第 2 步是 SCR 算法的核心步骤, 也是占用时间最长的步骤, 因此 MTCR 概念约简算法对第 2 步实现多线程并行计算方式改进, 其余步骤与 SCR 算法保持一致。

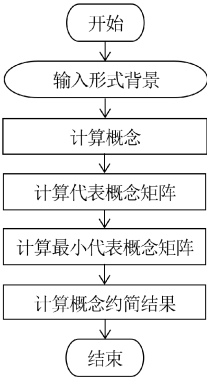


图 2 概念约简计算过程流程图

Fig. 2 The flow chart of concept reduct calculation

2.2 MTCR 概念约简算法

计算代表概念矩阵分为两步, 先计算对象代表概念集 $\text{REP}(g_i)$ 和属性代表概念集 $\text{REP}(m_j)$, 再计算 $\text{REP}(g_i)$ 和 $\text{REP}(m_j)$ 中的共有概念。

MTCR 算法在计算代表概念矩阵时,先以对象和属性为出发点,通过不同线程并行计算每一个对象 g_i 的对象代表概念集 $REP(g_i)$ 和每一个属性 m_j 的属性代表概念集 $REP(m_j)$ 。实现过程如算法 1 所示。

算法 1: 计算对象代表概念集和属性代表概念集。输入概念集合 $concepts$, 输出为对象代表概念集 $objs[i]$ 和属性代表概念集 $attrs[j]$ 。

```

1 Process1;
2 for  $i = 0$  to  $|G|$ 
3   for  $(X, B)$  in  $concepts$ 
4     if  $g_i \in X$ 
5       Add  $(X, B)$  to  $objs[i]$ ;
6     end if
7   end for
8 end for
9 Process2;
10 for  $j = 0$  to  $|M|$ 
11   for  $(X, B)$  in  $concepts$ 
12     if  $m_j \in B$ 
13       Add  $(X, B)$  to  $attrs[j]$ ;
14     end if
15   end for
16 end for

```

算法 1 创建了线程 1、线程 2, 两个线程并行执行。算法第 1 行到第 8 行, 创建线程 1 并计算每一个对象的对象代表概念集。遍历 $L(G, M, I)$ 中的概念, 将外延中包含对象 g_i 的所有概念的标志以位运算的方式存于 $objs[i]$ 中。

算法第 9 行到第 16 行, 创建线程 2 并计算每一个属性的属性代表概念集。遍历 $L(G, M, I)$ 中的概念, 将内涵中包含属性 m_j 的所有概念的标志以位运算的方式存于 $attrs[j]$ 中。

通过对象代表概念集 $REP(g_i)$ 和属性代表概念集 $REP(m_j)$ 可以计算出 (g_i, m_j) 的代表概念集。在计算过程中, 每计算一个代表概念集, 都会查询相应的对象代表概念集和属性代表概念集, 但不同代表概念集的计算过程互不影响。因此为了提高计算效率, 可以使用多线程方式并行计算不同代表概念集。将形式背景中的对象依次放入 p 个队列, 并为每个队列分配线程, 通过多线程方式并行计算出任意对象的对象代表概念集和任意属性的属性代表概念集的交集, 最终得到代表概念矩阵。实现过程如算法 2 所示。

算法 2: 计算代表概念矩阵。输入对象代表概念集 $objs[i]$ 和属性代表概念集 $attrs[j]$, 输出代表概念矩阵 $represented[i][j]$ 。

```

1 for obj in context
2   count ++;
3   Store(obj, queue[count%p]);
4 end for
5 for  $q = 0$  to  $p - 1$ 
6   new process;
7   for obj in queue[q]
8      $i \leftarrow obj$ ;
9     for  $j = 0$  to  $|M|$ 
10      if  $objs[i] \cap attrs[j] \neq \emptyset$ 
11         $represented[i][j] = objs[i] \cap attrs[j]$ ;
12      end if
13    end for
14  end for
15 end for

```

算法第 1 行到第 4 行, 将形式背景中的对象依次存入 p 个队列。

算法第 5 行到第 15 行, 为每个队列分配线程, 分别计算队列中所含对象的对象代表概念集和与每一个属性的属性代表概念集中的共有概念。由性质 1 可知, 结果即为外延具有特定对象且内涵具有特定属性的概念集合, 并将其存于代表概念矩阵 $represented$ 中。

算法 2 以形式背景中对象为分配数据依次放入 p 个队列, 计算任意对象的对象代表概念集与任意属性的属性代表概念集中的共有概念构建代表概念矩阵。若将算法 2 中的对象与属性互换, 以属性作为分配数据放入 p 个队列, 同样可以得到代表概念矩阵, 并且与算法 2 计算得到的代表概念矩阵一致。为了使多线程负载均衡, 当形式背景中对象数大于等于属性数时, MTCR 算法使用以对象为分配数据的方式进行计算; 当形式背景中的对象数小于属性数时, MTCR 算法以属性为分配数据的方式进行计算。

通过多线程并行计算对象代表概念集和属性代表概念集的方法计算代表概念矩阵, 极大地提高了执行效率。

3 实验结果与分析

实验在 Apple M1 Pro 10 核、主频为 3.20 GHz、内存为 16 GB 的便携式计算机上进行, 分别对例 1 中形式背景和 4 种 UCI 数据集进行处理, 分析 SCR 算法与 MTCR 算法在性能上的区别, 表明 MTCR 算法可以准确得到概念约简结果, 并且在运行速度上快于 SCR 算法。在对比实验 2 中使用随机数据集, 分别考虑属性数量、对象数量和密度大小对 MTCR 算法性能的影响。

实验中计算概念步骤和计算概念约简结果步骤

分别使用 Andrews^[24]提出的 InClose3 算法和智慧来等^[25]提出的极小覆盖法。为了直观对比 MTCR 算法相对于 SCR 算法在执行速度上的提升,我们仅比较计算代表概念矩阵和计算最小代表概念矩阵消耗的总时间,并计算提升效率。

例 1 中使用 MTCR 算法的实验结果如表 4 所示,其中 t 为计算概念约简所需时间。表 4 中概念约简结果与例 3 中计算得到的概念约简结果相同,表明 MTCR 算法的准确性。

对比实验 1 使用的 UCI 数据集是 UCI Machine Learning Repository 网站中提供的 Balance-Scale、Car、Iris 和 Wine 这 4 个数据集,对比 SCR 算法和 MTCR 算法在执行时间上的差异,对比结果如表 5 所示,MTCR 运行时间为使用两个线程时所消耗的时间。SCR 算法分别与 1、2、4 和 8 个线程的 MTCR 算法对比结果如图 3 所示。

通过对 4 个 UCI 数据集进行实验,由图 3 可以明确观察到使用 MTCR 概念约简算法在计算概念约简时,使用单线程情况下,与 SCR 算法计算时间几乎相同;使用 2、4 和 8 线程时,MTCR 算法使用时间均少于 SCR 算法。当线程数不超过 8 时,线程数每增加 1 倍,计算速度可以提高约 30%。

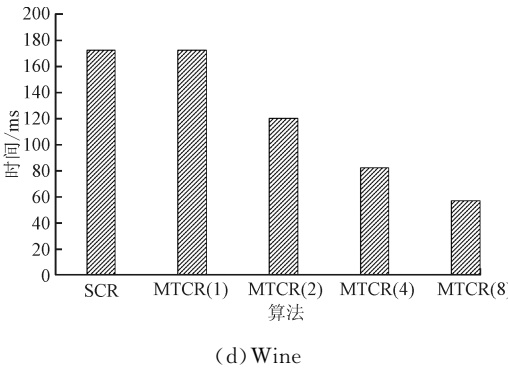
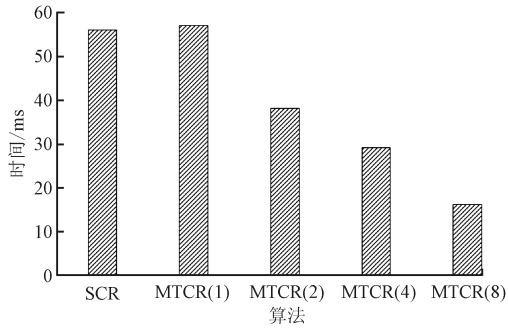
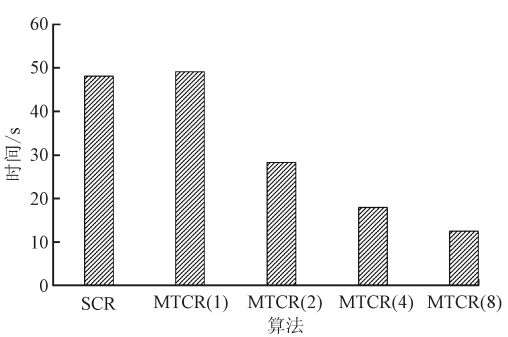
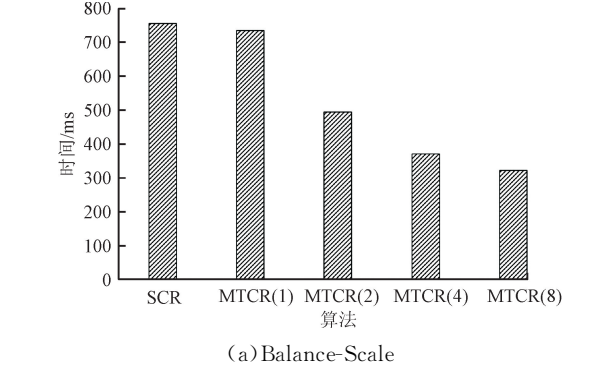


图 3 MTCR 算法与 SCR 算法对比实验
Fig. 3 Comparison test between MTCR and SCR

表 4 例 1 实验运行过程表

Table 4 The running process of the experiment in example 1

概念数量	t/ms	最小代表概念矩阵元素	概念约简
11	40	$\{c_5\}, \{c_9\}, \{c_2, c_8\}, \{c_3, c_7\},$ $\{c_4, c_7\}, \{c_4, c_8\}$	$\{c_5, c_7, c_8, c_9\}, \{c_2, c_4, c_5, c_7, c_9\},$ $\{c_2, c_3, c_4, c_5, c_9\}, \{c_3, c_4, c_5, c_8, c_9\}$

表 5 4 个 UCI 数据集实验结果

Table 5 Experimental results of 4 UCI datasets

UCI 数据集	对象数	属性数	概念总数	SCR 运行时间/ms	MTCR 运行时间/ms	提升效率/%
Balance-Scale	625	23	2 106	755	493	34.70
Car	1 728	25	12 640	48 114	28 282	41.22
Iris	150	126	445	56	38	32.14
Wine	178	1 279	984	172	120	30.23

对比实验 2,为了进一步验证 MTCR 算法的高效性,对随机数据进行实验,固定形式背景中其他变 量,分别改变属性数量,对象数量和密度大小,分析 对不同数量线程性能的影响,实验结果如图 4 所示。

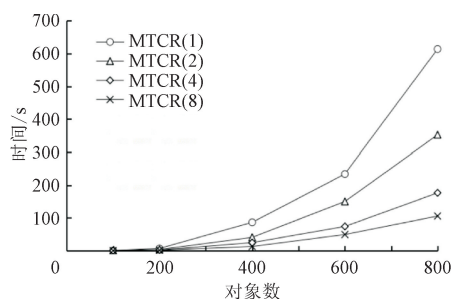
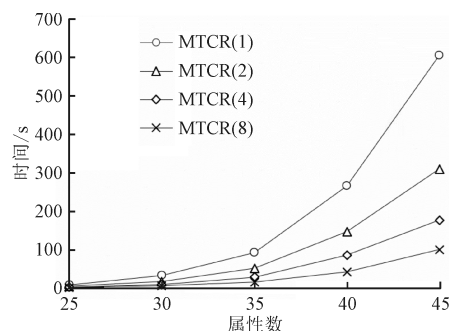
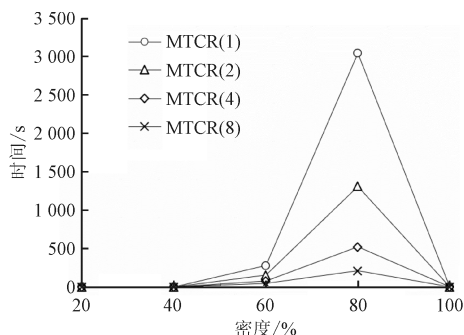
(a) 对象数 ($|M|=30, d=30\%$)(b) 属性数 ($|G|=300, d=30\%$)(c) 密度 ($|G|=100, |M|=20$)

图 4 不同变量对 MTCR 算法性能的影响

Fig. 4 Influence of different variables on the performance of MTCR

图 4(a)为固定形式背景的属性数量为 30,密度为 30%,对象数量的变化范围为 100~800;图 4(b)为固定形式背景的对象数量为 300,密度为 30%,属性数量的变化范围为 25~45;图 4(c)为固定形式背景对象数为 100,属性数为 20,密度大小的变化范围为 20%~100%。

在形式概念分析中,对象与属性呈对偶关系,对象数与属性数对 MTCR 算法的性能具有相同的影响。由图 4 中(a)、(b)可知,固定其他变量,仅改变对象数(属性数)时,MTCR 算法计算时间随对象数(属性数)增加而增加,当对象数和属性数都较大时,形式背景中的概念数量呈指数倍增长,MTCR 算法计算时间也呈指数倍增长。由图 4(c)可知,固定对象数和属性数不变,形式背景密度处于 80%左右

时,概念数量相对较多,MTCR 算法计算时间较长,密度趋于 0%和 100%时,概念数量相对较少,MTCR 算法计算时间较短。

通过对 UCI 数据集和随机数据集的实验分析可知,MTCR 算法使用的线程数量增长对算法效率的提升趋于稳定,当线程数不超过 8 时,随着线程数量每增加 1 倍,MTCR 算法计算效率均可提速 30%以上。但是,随着概念数量的增长,MTCR 算法在使用固定数量的多线程情况下,计算所需时间随概念数量的增长呈指数倍增长。在处理较大规模的数据集时,应增加线程数量,以提高计算效率。

4 结束语

本文基于通过构建代表概念矩阵计算概念约简的方法,对 SCR 算法中计算过程复杂,执行效率低的部分提出并行化处理方案。通过对例 1 中的形式背景、UCI 数据集和随机数据集的实验表明,MTCR 算法能准确、快速地计算出概念约简的结果,极大提高概念约简算法的执行效率。当线程数不超过 8 时,随着线程数量每增加 1 倍,MTCR 算法计算效率均可以提升 30%以上。

参考文献:

- [1] WILLE R. Restructuring lattice theory: an approach based on hierarchies of concepts [C]//Ordered Sets. Dordrecht: Springer Netherlands, 1982: 445-470.
- [2] GANTER B, WILLE R. Formal concept analysis: mathematical foundations [M]. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999.
- [3] RODRÍGUEZ-JIMÉNEZ J M, CORDERO P, ENCISO M, et al. Data mining algorithms to compute mixed concepts with negative attributes: an application to breast cancer data analysis [J]. Mathematical Methods in the Applied Sciences, 2016, 39(16): 4829-4845.
- [4] KUMAR C A. Fuzzy clustering-based formal concept analysis for association rules mining [J]. Applied Artificial Intelligence, 2012, 26(3): 274-301.
- [5] CARBONNEL J, HUCHARD M, NEBUT C. Modeling equivalence classes of feature models with concept lattices to assist their extraction from product descriptions [J]. Journal of Systems and Software, 2019, 152: 1-23.
- [6] AI-MSIE'DEEN R, BLASI A H. Software evolution understanding: automatic extraction of software identifiers map for object-oriented software systems [J]. Journal of Communications Software and Systems,

- 2021, 17(1): 20-28.
- [7] ANANIAS K H A, MISSAOUI R, RUAS P H B, et al. Triadic concept approximation [J]. Information Sciences, 2021, 572: 126-146.
- [8] KUMAR C. Knowledge discovery in data using formal concept analysis and random projections [J]. International Journal of Applied Mathematics and Computer Science, 2011, 21(4): 745-756.
- [9] MUANGPRATHUB J, BOONJING V, CHAMNON-GTHAI K. Learning recommendation with formal concept analysis for intelligent tutoring system [J]. Heliyon, 2020, 6(10): E05227.
- [10] MEZNI H, ABDELJAOUED T. A cloud services recommendation system based on fuzzy formal concept analysis [J]. Data & Knowledge Engineering, 2018, 116: 100-123.
- [11] AROUR K, YEFERNY T. Formal concept analysis based user model for distributed systems [J]. Multimedia Tools and Applications, 2017, 76(15): 16085-16105.
- [12] 刘美玉, 祁建军, 刘伟. 三支概念格中的关联规则提取算法 [J]. 西安交通大学学报, 2021, 55(9): 189-196.
- LIU Meiyu, QI Jianjun, LIU Wei. Extracting association rules in three-way concept lattices [J]. Journal of Xi'an Jiaotong University, 2021, 55(9): 189-196.
- [13] 曹丽, 魏玲, 祁建军. 保持二元关系不变的概念约简 [J]. 模式识别与人工智能, 2018, 31(6): 516-524.
- CAO Li, WEI Ling, QI Jianjun. Concept reduction preserving binary relations [J]. Pattern Recognition and Artificial Intelligence, 2018, 31(6): 516-524.
- [14] 魏玲, 曹丽, 祁建军, 等. 形式概念分析中的概念约简与概念特征 [J]. 中国科学(信息科学), 2020, 50(12): 1817-1833.
- WEI Ling, CAO Li, QI Jianjun, et al. Concept reduction and concept characteristics in formal concept analysis [J]. Scientia Sinica (Informationis), 2020, 50(12): 1817-1833.
- [15] ZHAO Siyu, QI Jianjun, LI Junan, et al. Concept reduction in formal concept analysis based on representative concept matrix [J]. International Journal of Machine Learning and Cybernetics, 2023, 14(4): 1147-1160.
- [16] 谢小贤, 李进金, 陈东晓, 等. 基于布尔矩阵的保持二元关系不变的概念约简 [J]. 山东大学学报(理学版), 2020, 55(5): 32-45.
- XIE Xiaoxian, LI Jinjin, CHEN Dongxiao, et al. Concept reduction of preserving binary relations based on Boolean matrix [J]. Journal of Shandong University (Natural Science), 2020, 55(5): 32-45.
- [17] 马文胜, 侯锡林. 形式概念分析中的同效关系与概念约简 [J]. 计算机科学, 2023, 50(4): 63-76.
- MA Wensheng, HOU Xilin. Same effect relation and concept reduction in formal concept analysis [J]. Computer Science, 2023, 50(4): 63-76.
- [18] 李俊余, 李星璇, 王霞, 等. 基于三元因子分析的三元概念约简 [J]. 南京大学学报(自然科学版), 2020, 56(4): 480-493.
- LI Junyu, LI Xingxuan, WANG Xia, et al. Reduction of triadic concepts based on triadic factor analysis [J]. Journal of Nanjing University(Natural Science), 2020, 56(4): 480-493.
- [19] 王霞, 彭致华, 李俊余, 等. 一种基于概念可辨识矩阵的概念约简方法 [J]. 计算机科学, 2021, 48(1): 125-130.
- WANG Xia, PENG Zhihua, LI Junyu, et al. Method of concept reduction based on concept discernibility matrix [J]. Computer Science, 2021, 48(1): 125-130.
- [20] 刘津, 米据生, 李仲玲, 等. 基于概念复合的对偶三支概念格及其概念约简 [J]. 计算机科学, 2023, 50(6): 122-130.
- LIU Jin, MI Jusheng, LI Zhongling, et al. Dual three-way concept lattice based on composition of concepts and its concept reduction [J]. Computer Science, 2023, 50(6): 122-130.
- [21] KUZNETSOV S O. A fast algorithm for computing all intersections of objects in a finite semi-lattice [J]. Automatic Documentation and Mathematical Linguistics, 1993, 27(5): 11-21.
- [22] KRAJCA P, OUTRATA J, VYCHODIL V. Advances in algorithms based on CbO [J]. Concept Lattices and Their Applications, 2010: 325-337.
- [23] ANDREWS S. In-close, a fast algorithm for computing formal concepts [C]//Proceedings of the 17th International Conference on Conceptual Structures. [S.l.]: [s.n.], 2009: 1-14.
- [24] ANDREWS S. A 'Best-of-Breed' approach for designing a fast algorithm for computing fixpoints of Galois connections [J]. Information Sciences, 2015, 295: 633-649.
- [25] 智慧来, 智东杰, 刘宗田. 从合取范式到析取范式的转换研究 [J]. 计算机工程与应用, 2012, 48(2): 15-17, 29.
- ZHI Huilai, ZHI Dongjie, LIU Zongtian. Research on conversion from conjunctive normal form to disjunctive normal form [J]. Computer Engineering and Applications, 2012, 48(2): 15-17, 29.

(编辑 赵炜)