

多尺度决策系统中高效下近似跳转算法研究

张 晴

烟台大学计算机与控制工程学院, 山东 烟台

收稿日期: 2025年3月22日; 录用日期: 2025年4月22日; 发布日期: 2025年4月29日

摘 要

经典粗糙集理论只能从单个尺度处理分析数据, 多尺度决策系统则可以从粗细不同尺度深度挖掘数据中隐含的信息, 该系统的提出拓宽了经典粗糙集的应用范围, 成为粗糙集领域的重要研究方向之一, 受到学者广泛关注。由于传统方法下各尺度间下近似的转换仍需要消耗较多资源, 因此本文提出一种下近似快速跳转方法。在基于乐观多粒度的多尺度决策系统的背景下, 利用多尺度决策系统中特有的粒度转换函数实现了尺度由细向粗跳转时下近似的快速更新, 并提出了相应的更新算法, 有效避免了重新对数据表的遍历。为了验证算法的有效性, 实验使用9个UCI数据集从下近似结果及时间方面分别进行验证, 结果表明本文所提出的算法可行且具有更高的效率。

关键词

粒计算, 下近似, 多尺度决策系统, 多粒度

Research on Efficient Lower Approximation Transition Algorithm Based on Multi-Scale Decision Systems

Qing Zhang

School of Computer and Control Engineering, Yantai University, Yantai Shandong

Received: Mar. 22nd, 2025; accepted: Apr. 22nd, 2025; published: Apr. 29th, 2025

Abstract

Classical rough set theory can only process and analyze data from a single scale, whereas a multi-scale decision system can deeply mine the information hidden in the data from different coarse and fine scales. The introduction of multi-scale decision systems has expanded the application scope of classical rough set theory, establishing it as a significant research direction within the field and garnering widespread attention. However, traditional methods for transitioning lower approximations

across scales still consume considerable resources. To address this, this paper proposes a rapid transition method for lower approximations. Initially, the multi-scale decision system is integrated with optimistic multigranulation rough sets. Building on this foundation, the method leverages the granular transformation functions of multi-scale decision systems to facilitate swift updates of lower approximations during scale transitions from fine to coarse. This approach introduces a fast update algorithm for lower approximations as scales coarsen, effectively circumventing the need for repeated traversal of data tables. To validate the algorithm's efficacy, experiments were conducted using nine UCI datasets, assessing both the results of lower approximations and computational time. The findings demonstrate that the proposed algorithm is not only feasible but also exhibits superior efficiency.

Keywords

Granular Computing, Lower Approximation, Multi-Scale Decision Systems, Multigranulation

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

粗糙集理论[1]作为处理不确定性数据的有效工具,在数据挖掘[2]、冲突分析[3]、决策理论[4]及特征选择[5]等领域广泛应用,引起了持续关注与探索。随着研究的不断深入,经典粗糙集理论也在持续发展完善,不断衍生出新的模型和应用方向,以更好地处理多样化的数据,为解决复杂问题提供了强大的理论支撑和工具支持。

为克服传统粗糙集在问题处理上的局限性和单一性,多尺度决策系统[6]应运而生。该系统通过将条件属性划分为具有粗细关系的多个尺度,从不同层次对信息进行分析与处理,从而提升数据处理的灵活性和适应性,具体来说,在细尺度下处理问题利于捕捉细节信息,而在粗尺度下处理问题则能更好地把握全局信息。2013年,研究者从多粒度[7]的角度提出了多尺度信息系统中的多粒度粗糙集[8],并基于乐观多粒度和悲观多粒度两种视角进行分析,以满足不同的需求。在多尺度决策系统中,最优尺度的选择[9]与属性约简[10]是其核心研究方向,而关于不同尺度之间关联性的挖掘研究仍较为有限。

现有研究尚未涉及利用粒度转换函数在不同尺度间实现下近似跳转的机制与方法,因此本文在基于乐观多粒度的多尺度决策系统中,提出了从细尺度到粗尺度的下近似快速更新方法,并构建了多尺度决策系统下基于乐观多粒度的下近似快速跳转算法。最终,通过设计对比实验验证了本文所提算法的正确性与高效性。

2. 基本概念

定义 1 [6]: 二元组 $S = (U, C \cup D)$ 表示一个多尺度决策信息系统, $U = \{x_1, x_2, \dots, x_s\}$ 是一个非空有限集合,表示论域; $D = \{d\}$ 为决策属性集; $C = \{a_j \mid j = 1, 2, \dots, n\}$ 表示具有 n 个条件属性的集合,其中每个条件属性 $a_j \in C$ 在多尺度条件下具有 m 个尺度,即 $a_j = \{a_j^1, a_j^2, \dots, a_j^m\}$ 。条件属性 a_j 的相邻的两个尺度之间存在粒度转换函数 $g_{a_j}^{u-1,u}$,使得 $a_j^u(x) = g_{a_j}^{u-1,u}(a_j^{u-1}(x))$,其中 $g_{a_j}^{u-1,u}$ 为满射函数,因此条件属性集合在多尺度条件下可表示为 $C = \{a_j^u \mid j = 1, 2, \dots, n, u = 1, 2, \dots, m\}$ 。

定义 2 [11]: $S = (U, C \cup D)$ 为一个多尺度决策信息系统,其中 $C = \{a_j^u \mid j = 1, 2, \dots, n, u = 1, 2, \dots, m\}$ 。

给定 $k = (u_1, u_2, \dots, u_n)$ 表示第 i 个条件属性的尺度被限制为 u_i ，则条件属性集 C 在 k 下形成的单尺度信息表表示为： $S^k = (U, C^k \cup \{d\})$ ，其中 $C^k = (a_1^{u_1}, a_2^{u_2}, \dots, a_n^{u_n})$ 。此时称 k 为 S 的一个尺度组合。 S 中所有的 k 组成的集合称为尺度空间，记作 K 。

定义 3 [11]： $S = (U, C \cup D)$ 为一个多尺度决策信息系统， K 为尺度空间，对于尺度组合 $k = (u_1, u_2, \dots, u_n)$ ， $k' = (u'_1, u'_2, \dots, u'_n) \in K$ ，若对于任意的 $i \in [1, n]$ ，均满足 $u_i \leq u'_i$ ，称尺度组合 k' 较 k 更粗，记作 $k < k'$ 。

定义 4 [8]：二元组 $S = (U, C \cup D)$ 为一个多尺度决策信息系统。当尺度组合为 $v \in K$ 时，得到信息表 $S^v = (U, C^v \cup \{d\})$ 。粒度集 $A^v = \{A_1^v, A_2^v, \dots, A_t^v\}$ ，满足 $\forall A_i^v \in A^v, A_i^v \subseteq C^v$ 。对于 $X \subseteq U$ ， X 在粒度集 A^v 下的乐观多粒度下、上近似定义分别为：

$$\underline{Apr}_{A^v}^O(X) = \{x \in U, x \in \underline{Apr}_{A_1^v}(X) \vee x \in \underline{Apr}_{A_2^v}(X) \vee \dots \vee x \in \underline{Apr}_{A_t^v}(X)\} \quad (1)$$

$$\overline{Apr}_{A^v}^O(X) = \sim (\underline{Apr}_{A^v}^O(\sim X)) \quad (2)$$

则 X 的乐观多粒度正域定义如下：

$$POS_{A^v}^O(X) = \underline{Apr}_{A^v}^O(X) \quad (3)$$

性质 1：在多尺度决策信息系统 $S = (U, C \cup D) = (U, \{a_j^u \mid j = 1, 2, \dots, n, u = 1, 2, \dots, m\} \cup \{d\})$ 中，对于尺度组合 $v, v' \in K$ ， $v < v'$ ，对于粒度集 A 中的任意粒度 $A_i \subseteq A (i \in [1, t])$ ，满足以下性质：

$$\underline{Apr}_{A^v}^O(X) \subseteq \underline{Apr}_{A^{v'}}^O(X) \quad \underline{Apr}_{A_i^v}^O(X) \subseteq \underline{Apr}_{A_i^{v'}}^O(X) \quad (4)$$

3. 基于多粒度的多尺度决策系统中下近似快速跳转方法

多粒度下近似的计算建立在单个粒度下近似计算的基础之上，而尺度由细向粗变化时，传统更新单粒度下近似的方法需要重新遍历整个信息表以确定每个对象的等价类，导致计算效率较低。针对这一问题，本研究将多尺度决策系统特有的粒度转换函数与哈希表相结合，提出了一种基于多尺度决策系统中基于粒度转换函数的下近似快速更新算法。该算法避免了大规模数据遍历，从而显著提升了尺度转换过程中下近似更新的计算效率。本研究以单个属性为一个粒度进行分析。

定义 5：在粗糙集理论中哈希查找借助哈希表对元素进行分类存储，能够实现数据的快速存取，具体可表示为 $key: \{value\} = key: \{x, d, T/F\}$ 。其中 key 为对象的属性值， $value$ 为三元组，在 $value$ 中， x 表示对象的等价类， d 为等价类的广义决策值， T/F 为判断该对象是否在下近似中的标识项。

性质 2：在多尺度决策信息系统 $S = (U, C \cup D) = (U, \{a_j^u \mid j = 1, 2, \dots, n, u = 1, 2, \dots, m\} \cup \{d\})$ 中，对于 $\forall x \in U$ ，属性 a_j 在尺度 u 下所划分的等价类表示为 $[x]_{a_j^u}$ ， $g_{a_j}^{u, u'}$ 表示条件属性 a_j 在尺度 u 与 u' 之间的粒度转换函数，其中 $u < u'$ ，则当尺度由细变粗，即由 u 变化为 u' 时，对于 $\forall x, y \in U$ ，可能发生的情况如下：

- (1) $g_{a_j}^{u, u'}(a_j^u(x)) \neq g_{a_j}^{u, u'}(a_j^u(y))$;
- (2) $g_{a_j}^{u, u'}(a_j^u(x)) = g_{a_j}^{u, u'}(a_j^u(y))$ 且 $x \notin \underline{Apr}_{a_j^u}(D)$, $y \notin \underline{Apr}_{a_j^u}(D)$;
- (3) $g_{a_j}^{u, u'}(a_j^u(x)) = g_{a_j}^{u, u'}(a_j^u(y))$ 且 $x \in \underline{Apr}_{a_j^u}(D)$, $y \notin \underline{Apr}_{a_j^u}(D)$;
- (4) $g_{a_j}^{u, u'}(a_j^u(x)) = g_{a_j}^{u, u'}(a_j^u(y))$ 且 $x \in \underline{Apr}_{a_j^u}(D)$, $y \in \underline{Apr}_{a_j^u}(D)$ 。

其中 $g_{a_j}^{u,u'}(a_j^u(x)) = g_{a_j}^{u,u'}(a_j^u(y))$ 表示尺度变化时 x, y 所在的等价类会发生合并。对于以上四种情况, 只有当发生情况(3)或发生情况(4)且满足 x, y 在尺度 u 下的决策值不相同, 对象 x 不再属于尺度 u' 的下近似, 此时需要从下近似中移除该对象。情况(1)和(2)不会对下近似产生影响。

证明:

情况(1)不在考虑范围之内; 情况(2)显然成立在此不做证明;

情况(3): 因为 $y \notin \underline{Apr}_{a_j}^{u'}(D)$, 所以 $\exists o \in U, o \in [y]_{a_j^{u'}}$ 并且 $d(o) \neq d(y)$; 因为 $g_{a_j}^{u,u'}(a_j^u(x)) = g_{a_j}^{u,u'}(a_j^u(y))$, 所以 $o, y, x \in [x]_{a_j^{u'}}$ 并且 $[o]_{a_j^{u'}} = [x]_{a_j^{u'}} = [y]_{a_j^{u'}}$ 。已知 $d(o) \neq d(y)$, 故而得知 $[x]_{a_j^{u'}} \not\subset \underline{Apr}_{a_j}^{u'}(D)$ 。情况(4)同理可证。

证毕。

图1以可视化的形式展示了上述性质的图示说明, 不同的等价类由不同形状表示, 上述性质中的(1)、(2)、(3)与图中一一相对应, 从图1可以看出, 情况(1)和情况(2) (a)会对下近似值产生影响, 并且在更新过程中可以删除整个等价类, 而不是单独删除对象, 可以简化计算流程并提高更新效率。

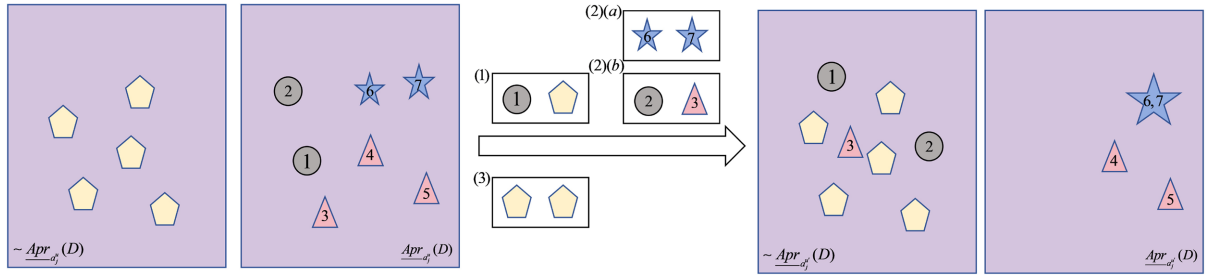


Figure 1. Change diagram of lower approximation

图1. 尺度由细到粗转换时下近似变化图

定义6: 在多尺度决策信息系统 $S = (U, C \cup D) = (U, \{a_j^u \mid j = 1, 2, \dots, n, u = 1, 2, \dots, m\} \cup \{d\})$ 中, K 为尺度空间, 给定尺度组合 $v, v' \in K$ 且满足 $v < v'$ 。以单个属性作为一个粒度, 尺度组合 $v = \{u_1, u_2, \dots, u_n\}$ 和 $v' = \{u'_1, u'_2, \dots, u'_n\}$ 的多粒度结构分别为 $A^v = \{A_1^v, A_2^v, \dots, A_n^v\}$ 和 $A^{v'} = \{A_1^{v'}, A_2^{v'}, \dots, A_n^{v'}\}$, 其中 $A_i^v, A_i^{v'} \in C^v$ 。各条件属性在两个尺度间的粒度转换函数表示为 $g_{a_1}^{u_1, u'_1}, g_{a_2}^{u_2, u'_2}, \dots, g_{a_n}^{u_n, u'_n}$ 。对于决策属性 D , $\underline{Apr}_{A_i^v}^{v'}(D)$ 表示尺度组合 v' 下单个粒度 A_i^v 关于 D 的下近似。当细粒度向粗粒度转换时, 尺度组合 v' 的下近似快速更新公式为:

$$\underline{Apr}_{A_i^{v'}}^{v'}(D) = \underline{Apr}_{A_i^v}^{v'}(D) - \bigcup \left\{ [x]_{A_i^v} \mid x \in \underline{Apr}_{A_i^v}^{v'}(D), \exists y \in \Delta, d(y) \neq d(x) \right\} \quad (5)$$

其中, Δ 为满足 $g_{a_i}^{u_i, u'_i}(A_i^v(x)) = g_{a_i}^{u_i, u'_i}(A_i^v(y))$ 的对象集合, 即在尺度从细向粗转换过程中, 等价类发生合并的对象集合, $d(x)$ 为对象 x 的决策值。

则结合公式(1)和(5), 由细尺度向粗尺度转换时尺度组合 v' 下的乐观多粒度下近似快速更新公式表示如下:

$$\underline{Apr}_{A^{v'}}^{v'}(D) = \bigcup_{A_i^v \in A^v} \left\{ \underline{Apr}_{A_i^v}^{v'}(D) - \bigcup \left\{ [x]_{A_i^v} \mid x \in \underline{Apr}_{A_i^v}^{v'}(D), \exists y \in \Delta, d(y) \neq d(x) \right\} \right\} \quad (6)$$

通过遍历粒度转换函数, 找出尺度变换后取值相同的元素, 并利用哈希表快速检索这些值对应的对

象。检查这些对象决策值的一致性,若存在不一致的情况,则将该对象所在的等价类从下近似集合中移除,从而获得更新后的下近似结果。这种方式能够在获得细尺度下近似后,快速计算更粗尺度的下近似,拥有更高的时间效率。根据上述定义和分析,下面给出尺度由细到粗跳转时乐观多粒度下近似的快速更新算法,如表1所示。

Table 1. Fast update algorithm for multigranulation lower approximations from fine to coarse scales

表 1. 尺度由细到粗跳转时多粒度下近似的快速更新算法

输入: 多尺度决策系统 $S = (U, C \cup D)$, 尺度组合 $v = \{u_1, u_2, \dots, u_n\}$ 和 $v' = \{u'_1, u'_2, \dots, u'_n\}$ 且 $v < v'$, 粒度转换函数为 $g_{a_1}^{u_1, u'_1}, g_{a_2}^{u_2, u'_2}, \dots, g_{a_n}^{u_n, u'_n}$, 多粒度结构 $A^v = \{A_1^v, A_2^v, \dots, A_n^v\}$ 和 $A^{v'} = \{A_1^{v'}, A_2^{v'}, \dots, A_n^{v'}\}$ 。

输出: $\underline{Apr}_{A^{v'}}^O(D)$ 。

1. 初始化: $\underline{Apr}_{A^{v'}}^O(D) = \emptyset$, $\underline{Apr}_{A^{v'}}^P(D) = \emptyset$;
2. 对于 $\forall A_i^v \in A^v$, 执行以下操作:
 - 2.1. 初始化字典 $dict = \{key: \{\}, value: \{\}\}$, $\underline{Apr}_{A_i^v}(D) = \emptyset$;
 - 2.2. $\forall x \in U$, 执行以下步骤:
 - 2.2.1. 若 $A_i^v(x)$ 不在 $dict$ 索引中: 添加索引 $A_i^v(x)$;
 - 2.2.2. 更新 $dict$ 中 key 为 $A_i^v(x)$ 的 $value$;
 - 2.3. 通过 $dict$ 计算 $\underline{Apr}_{A_i^v}(D)$;
 - 2.4. 遍历粒度转换函数 $g_i^{u_i, u'_i}$, 通过 $dict$ 找出需要合并的等价类集合, 合并后的集合记为 $Z_1, Z_2, \dots, Z_s$;
 - 2.5. 对于 $Z_i (1 \leq i \leq s)$, 执行以下步骤:
 - 2.5.1. 对于 $\forall x \in Z_i$, 若 $dict$ 中的标识项存在 T 且 d 值不相同: $\underline{Apr}_{A_i^{v'}}(D) = \underline{Apr}_{A_i^v}(D) - Z_i$;
3. 根据公式(6)计算 $\underline{Apr}_{A^{v'}}^O(D)$;
4. 输出粗尺度下乐观多粒度下近似 $\underline{Apr}_{A^{v'}}^O(D)$;
5. 算法结束。

4. 实验分析

本节从结果正确性及时间效率两个方面对本文提出的算法进行验证。实验环境为 Windows 11 操作系统, 配备 1.6GHz 的 CPU 和 8GB 的内存, 使用编程软件为 Pycharm2022.1.3。在实验中, 采用 9 组 UCI 数据集(<https://archive.ics.uci.edu/datasets>)进行对比实验, 详细信息见表 2, 包括数据集名称、样本数及属性数。

考虑到原始数据集中缺乏多尺度特征, 因此在实验开始之前对原始数据进行预处理以构建具有粗细关系的三个尺度。具体构造方法为:

- 1) 首先基于文献[11]实验中所提及的方法构建第一尺度的值, 构造公式如下:

$$a^1(x) = \lfloor a(x) - m_a \rfloor / std(a) \quad (7)$$

在公式(7)中, $a(x)$ 表示对象 x 在条件属性 a 下的值, m_a 表示最小值, $std(a)$ 表示条件属性 a 下的标准差, 符号 $\lfloor a(x) - m_a \rfloor$ 表示向下取整函数, 将连续数据转换为离散数据。

- 2) 随后采用层次聚类构建剩余两个尺度, 构建过程中需要保证每个条件属性的类别数量不小于 2 个, 若无法满足该条件, 则终止当前尺度的聚类过程, 并将该尺度数据直接复制至下一尺度。

Table 2. UCI datasets
表 2. UCI 数据集

序号	数据集	样本数	属性数
1	Magic Gamma Telescope	19,020	10
2	Dry Bean Dataset	13,611	16
3	Raisin	900	7
4	Rice	3810	7
5	Iranian Churn Dataset	3150	13
6	Yeast	1484	8
7	Page Blocks Classification	5473	10
8	Liver Disorders	345	5
9	Iris	150	4

4.1. 算法正确性验证

Table 3. Lower approximation results
表 3. 下近似结果

序号	LAFTC		Hash		CL	
	下近似结果	长度	下近似结果	长度	下近似结果	长度
1	{12337, 12458, ..., 18979, 18981}	148	{12337, 12458, ..., 18979, 18981}	148	{12337, 12458, ..., 18979, 18981}	148
2	{4, 25, ..., 7426, 7427}	1660	{4, 25, ..., 7426, 7427}	1660	{4, 25, ..., 7426, 7427}	1660
3	{85, 290, ..., 892, 893}	151	{85, 290, ..., 892, 893}	151	{85, 290, ..., 892, 893}	151
4	{0, 7, ..., 3808, 3809}	462	{0, 7, ..., 3808, 3809}	462	{0, 7, ..., 3808, 3809}	462
5	{2, 6, ..., 3146, 3148}	1019	{2, 6, ..., 3146, 3148}	1019	{2, 6, ..., 3146, 3148}	1019
6	{114, 553, 881, 988, 989, 990, 1079}	7	{114, 553, 881, 988, 989, 990, 1079}	7	{114, 553, 881, 988, 989, 990, 1079}	7
7	{307, 323, ..., 5318, 5466}	34	{307, 323, ..., 5318, 5466}	34	{307, 323, ..., 5318, 5466}	34
8	{84, 223}	2	{84, 223}	2	{84, 223}	2
9	{0, 1, ..., 48, 49}	50	{0, 1, ..., 48, 49}	50	{0, 1, ..., 48, 49}	50

为了验证本文提出的 LAFTC 算法的有效性,我们将所提算法的计算结果与基于哈希[12]的求解乐观多粒度下近似方法(Hash)、传统的基于遍历[1]的求解乐观多粒度下近似方法(CL)进行了对比。实验结果表明,本文所提算法是正确且有效的。

在实验中,我们展示了第二尺度乐观多粒度下近似的对比结果,具体如表 3 所示。实验中分别采用三种算法计算从第一尺度向第二尺度跳转时乐观多粒度的下近似结果,下近似具体结果及其长度分别在表 3 中展示,三种算法求得乐观多粒度下近似结果完全一致,但由于下近似结果过长,无法全部展示,因此在表 3 中仅列出了结果下标的前两位及后两位。从表 3 的结果中可以看出,通过算法 LAFTC、Hash 及 CL 在 9 个数据集中所获得的下近似结果保持完全一致,这表明本文所提的 LAFTC 算法有效且正确。

4.2. 算法高效性验证

在本节中，我们对 LAFTC 算法与基于哈希求解乐观多粒度下近似求解方法(Hash)、传统的基于遍历求解乐观多粒度下近似求解方法(CL)在计算乐观多粒度下近似时的时间效率方面进行了对比，实验结果见表 4 及图 2。

Table 4. Computation time (s)
表 4. 运行时间

序号	LAFTC	Hash	CL
1	0.0003	0.1223	0.2568
2	0.0013	0.1325	0.2241
3	0.0005	0.0043	0.0057
4	0.0004	0.0150	0.0236
5	0.0014	0.0229	0.0218
6	0.0002	0.0073	0.0172
7	0.0003	0.0290	0.0839
8	0.0001	0.0012	0.0029
9	0.0002	0.0004	0.0006
平均	0.0005	0.0372	0.0707

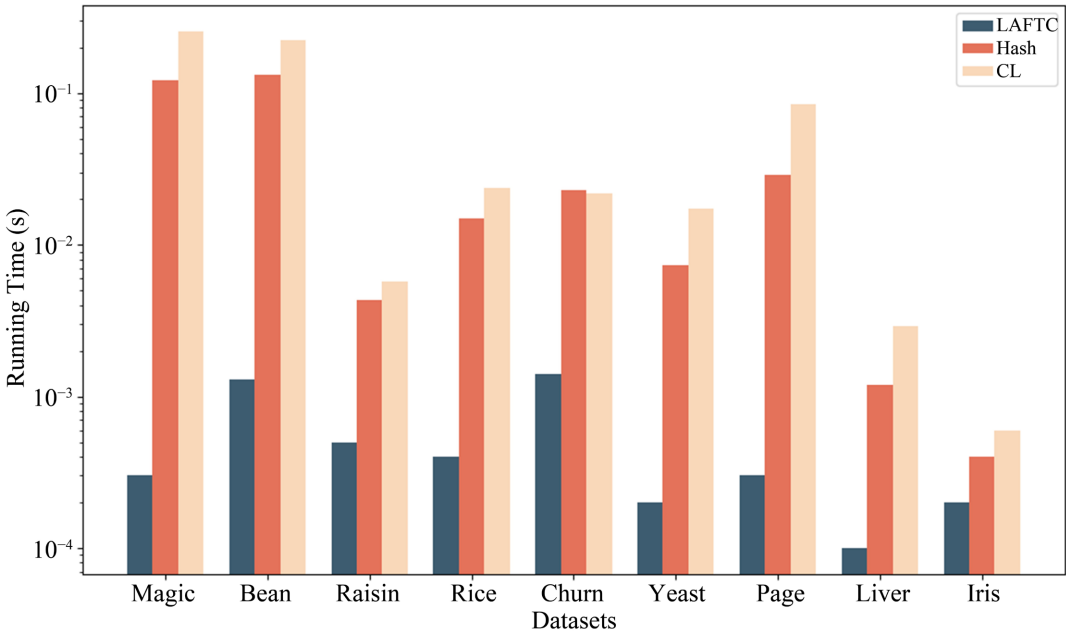


Figure 2. Optimistic multigranulation lower approximation time comparison
图 2. 计算乐观多粒度下近似时间对比图

表中结果均保留 4 位小数，时间单位为秒(s)，最后一行表示在 9 个数据集上计算乐观多粒度使用时间的平均值。图中使用对数坐标以更清晰地展示各算法之间的差异。由表和图可以看出，算法 LAFTC 在计算多粒度下近似时使用的时间平均最短，且在每个数据集中均取得了较高的效率。其中 Iris 数据集的

加速效果不如 Magic 数据集,这主要是由于 Iris 数据集的对象规模较小,导致不同算法之间的运行时间差异并不明显。相比之下,对于 Magic 和 Bean 等大规模数据集的加速效果较好,其中 Bean 数据集由于特征维度较高,其运行时间比 Magic 更长。从实验结果可以看出,本文提出的算法在处理大规模数据集时展现出显著的性能优势。综上所述,LAFTC 算法是有效的且在时间效率上表现优异。

5. 结论

本文在多尺度决策系统的研究背景下,结合哈希数据结构及粒度转换函数讨论了在尺度跳转时乐观多粒度下近似的变化机制。基于此,本文提出了尺度由细到粗跳转时乐观多粒度下近似的快速更新算法,实验部分通过对比不同算法在下近似结果上的一致性以及运行时间上的性能表现,验证了本文所提算法的正确性与高效性。实验结果表明本文算法不仅能够多尺度决策系统中准确计算乐观多粒度下近似,还在时间效率上具有显著优势。未来将进一步研究该算法在动态数据集的实现。

基金项目

本文受烟台市科技计划项目(编号:2022XDRH016)的资助。

参考文献

- [1] Pawlak, Z. (1982) Rough Sets. *International Journal of Computer & Information Sciences*, **11**, 341-356. <https://doi.org/10.1007/bf01001956>
- [2] Kong, Q., Wang, W., Xu, W. and Yan, C. (2024) A Method of Data Analysis Based on Division-Mining-Fusion Strategy. *Information Sciences*, **666**, Article 120450. <https://doi.org/10.1016/j.ins.2024.120450>
- [3] Zhi, H. and Li, J. (2024) Component Similarity Based Conflict Analysis: An Information Fusion Viewpoint. *Information Fusion*, **104**, Article 102157. <https://doi.org/10.1016/j.inffus.2023.102157>
- [4] Guo, Y., Tsang, E.C.C., Hu, M., Lin, X., Chen, D., Xu, W., et al. (2020) Incremental Updating Approximations for Double-Quantitative Decision-Theoretic Rough Sets with the Variation of Objects. *Knowledge-Based Systems*, **189**, Article 105082. <https://doi.org/10.1016/j.knosys.2019.105082>
- [5] Huang, Z. and Li, J. (2024) Covering Based Multi-Granulation Rough Fuzzy Sets with Applications to Feature Selection. *Expert Systems with Applications*, **238**, Article 121908. <https://doi.org/10.1016/j.eswa.2023.121908>
- [6] Wu, W. and Leung, Y. (2011) Theory and Applications of Granular Labelled Partitions in Multi-Scale Decision Tables. *Information Sciences*, **181**, 3878-3897. <https://doi.org/10.1016/j.ins.2011.04.047>
- [7] Qian, Y., Liang, J., Yao, Y. and Dang, C. (2010) MGRS: A Multi-Granulation Rough Set. *Information Sciences*, **180**, 949-970. <https://doi.org/10.1016/j.ins.2009.11.023>
- [8] Gu, S., Li, X., Wu, W. and Nian, H. (2013) Multi-Granulation Rough Sets in Multi-Scale Information Systems. 2013 *International Conference on Machine Learning and Cybernetics*, Tianjin, 14-17 July 2013, 108-113. <https://doi.org/10.1109/icmlc.2013.6890453>
- [9] Bao, H., Wu, W., Zheng, J. and Li, T. (2021) Entropy Based Optimal Scale Combination Selection for Generalized Multi-Scale Information Tables. *International Journal of Machine Learning and Cybernetics*, **12**, 1427-1437. <https://doi.org/10.1007/s13042-020-01243-y>
- [10] Wang, Z., Chen, H., Yuan, Z., Wan, J. and Li, T. (2023) Multiscale Fuzzy Entropy-Based Feature Selection. *IEEE Transactions on Fuzzy Systems*, **31**, 3248-3262. <https://doi.org/10.1109/tfuzz.2023.3250639>
- [11] Li, F. and Hu, B.Q. (2017) A New Approach of Optimal Scale Selection to Multi-Scale Decision Tables. *Information Sciences*, **381**, 193-208. <https://doi.org/10.1016/j.ins.2016.11.016>
- [12] 刘勇, 熊蓉, 褚健. Hash 快速属性约简算法[J]. 计算机学报, 2009(8): 1493-1499.