Applied Intelligence (2022) 52:9148–9173
https://doi.org/10.1007/s10489-021-02861-x

Dynamic updating approximations of local generalized multigranulation neighborhood rough set

Weihua Xu[1] · Kehua Yuan[2] · Wentao Li[1]

Accepted: 17 September 2021 / Published online: 3 January 2022

Abstract

The approximation space in rough set theory is important for dealing with uncertainties. As the information contained in various information systems is constantly updated and changed with the development of information technology, how to effectively obtain the approximation space in dynamic environments is essential. The local rough set as an excellent model avoids unnecessary calculation of information granules, and can significantly improve learning efficiency. In this paper, we mainly investigate a dynamic approximation update mechanism of multigranulation data from local viewpoint. We first define a support and inclusion function to construct local generalized multigranulation neighborhood rough set model. Then, the dynamic updating process of global rough set and local rough set is analyzed when object chandes. Meanwhile, the corresponding dynamic update algorithms for dynamic objects are proposed based on local generalized multigranulation rough set model. The complexity analysis about them theoretically proves the efficiency of local dynamic algorithm compared with global algorithm and static algorithm. To illustrate the effectiveness of proposed algorithms, twelve datasets from UCI are adopted to contrast experiments.

Keywords Approximation space · Changing objects · Dynamic updating · Generalized multigranulation rough set · Local rough set · Neighborhood information system

1 Introduction

The Rough set(RS) theory [14] proposed by Pawlak is an important mathematical tool for handling uncertain knowledge, in which concept approximation [5, 10, 20] and attribute reduction [8, 20, 34, 35] are two important research aspects. Since the rough set does not require any prior knowledge other than information in the studied data, it has been widely applied to data mining [3, 4, 6], individual classification [32], uncertain reasoning [32, 34], and so on [1, 11, 12, 23]. In classical rough set theory, we need to obtain all information classes of the objects coming from universe for approximating target concept, it is complicated and time-consuming [15, 18, 36]. Also, the classical rough set model is based on equivalence relation and equivalence classes, which is limited in real life and has poor fault tolerance for real data [13, 15]. In order to solve these issues, many scholars have further studied the properties of rough set model and extended the equivalent relation to deal with various complex data [7, 18, 23, 25].

With the development of technology, massive data with various features is emerging. In RS theory, we need to compute all equivalence classes to obtain the lower and upper approximations, which is not convenient in big dataset. Luckily, the proposal of local rough set(LRS) has solved this issue well [16, 18]. In LRS theory, we only need to obtain the information classes determined by the objects from target concept, this avoids unnecessary computation between the elements outside target concept and significantly improves approximate efficiency [15, 16, 18]. Also, scholar Qian and others explored its application in concept approximation and attribute reduction, it significantly reduces time complexity

✉ Weihua Xu
chxuwh@gmail.com

Kehua Yuan
yuan945527632@163.com

Wentao Li
drliwentao@gmail.com

1 College of Artificial Intelligence, Southwest University, Chongqing, 400715, People's Republic of China

2 School of Mathematics Statistics, Southwest University, Chongqing, 400715, People's Republic of China

and saves memory, moreover, the local though thinking can be further applied to semi-supervised study for processing unlabeled data [16]. Guo et al. proposed local logical disjunction double-quantitative rough sets model and found it is an effective model for decision through exploring the difference between them and other models [5]. The local rough set model theory is a new perspective to learn the uncertainties in big data.

With the development of information science, the diversity of stored data increases. To flexibly handle these complex data, many scholars have further extended the equivalence relation to dominance relation [1, 10], neighborhood relation [6, 13, 19], fuzzy relation [20, 31, 33, 37]. Especially, neighborhood rough set(NRS) based on neighborhood relation has strong processing ability for complex data through defining distance function. In NRS theory, the neighborhood radius is an important parameter for distinguishing individuals. Jiang et al. proposed the intra-class radius and inter-class radius to distinguish samples considering their labels, and then designed accelerator attribute reduction algorithm based on supervised neighborhood rough set. Their experimental results show the accelerator reduction algorithm is effective and efficient for feature selection [6]. To improve the reduction efficiency, Chen et al. utilized the parallel of neighborhood distance matrix to design parallel algorithm for computing reduction in DNRS, the corresponding reduced subset behaves well in classification accuracy and consume time [1]. Also, based on Gaussian function, Yang et al. proposed a comprehensive granularity research method on horizontal granularity and vertical granularity from the viewpoint of neighborhood rough set, and the efficiency of multilevel neighborhood sequential three-way decision is verified by the acceptable accuracy and reduction of processing cost [6]. Moreover, Wang et al. processed attribute reduction in a semi-supervised method based on local neighborhood rough set model which significantly reduced the complexity and improved the utilization of information [18].

In practical application situations, objects usually need to be depicted from multiple aspects, but the classical single rough set model can only describe object from one point of view. For this issue, Qian et al. first extended the classical single rough set to multigranulation rough set model ($MGRS$) to deal with multi-aspect information [17, 24, 29]. Subsequently, many scholars explored it in many areas, such as approximate concept [17, 19, 22], granularity selection [6, 21], decision-making [15, 28], and data mining [27, 33, 37]. So far, there have developed two models: optimistic multi-granularity rough set($OMGRS$) and pessimistic multigranulation rough set($PMGRS$) [33, 37]. Zhou et al. used multigranulation approximation regions to solve uncertainty associated with the fuzzifier parameter to realize fuzzy clustering [37]. As we all know, the approximate operators are too loose and tight, which can not refine the attribute information flexible. The raise of generalized multigranulation rough set by Xu et al. have made it possible, the generalized model can describe samples more detail and flexible [21]. Meanwhile, it is noted that the approximation space reflecting the certainty and possibility can be applied to practical situations such as decision-making. In 2019, Fujita et al. first made resilience analysis in critical infrastructures based on three-way decision-making and granular computing thinking [4]. Then, they further proposed a new interactive method to analyze and evaluate hypotheses to terrorism events based on three-way decisions and fuzzy probabilistic rough sets, which further verifies the broadness of rough set theory in application scenarios [3]. Therefore, how to effectively update approximate space in dynamic environment is important in data mining [9, 26, 30]. To improve the efficiency of obtaining approximation space, scholars have investigated many strategies. Chen et al. proposed parallel algorithm based on matrix parallel computing, which significantly improves the reduction efficiency [1]. Jiang et al. designed accelerator for feature selection combining the properties of feature indicator with neighborhood radius [6]. Yang et al. proposed a unified dynamic framework of decision-theoretic rough sets for incrementally updating three-way probabilistic regions [26]. Liu et al. introduced Multi-Granularity Attribute Selector to the framework of heuristic algorithm to improve the reduction efficiency [12]. These dynamic updating algorithms save a lot of time produced by these repeated computing based on dynamic information system compared with traditional algorithms. In this paper, we mainly investigate a dynamic update mechanism for approximate space when the object is changing.

Considering the fault tolerance and flexibility of neighborhood rough set, we construct local generalized multigranulation neighborhood rough set model to characterise uncertainty and imprecise knowledge in real data from multi-aspects, and design the dynamic update mechanism for approximate space when the object set changes. The main contributions of this paper are as follows: (1) The support and inclusion functions in neighborhood information system are proposed to construct local multigranulation neighborhood rough set model. This model could describe objects in target concept more detail from multiple viewpoints and avoid necessary computing. (2) We analyze the dynamic updating process and design corresponding algorithm for updating approximate space when object changes, and then give an example to illustrate the dynamic update process. (3) The local dynamic algorithm is a valid strategy by complexity analysis, which is further verified by the experimental results running on twelve public datasets.

The rest of the paper is organized as follows: Some related concepts are reviewed in Section 2. In Section 3, we first define the support function and inclusion function, and then construct local generalized multigranulation neighborhood rough set model. Section 4 shows the dynamic update mechanism when the object changes, and give an example to illustrate the update process. In Section 5, we design the dynamic update algorithms of computing approximations and analyze their complexity in theory. Moreover, the related experiments on UCI data sets are carried in Section 6. Through comparing the time consumption between static and dynamic updating, we find the local dynamic update algorithm is an efficient strategy for approximation update. Finally, we draw some conclusions in this paper in Section 7.

To conveniently illustrate the fundamental idea of this paper, we give the following chart as shown in Fig. 1.

2 Related work

In this section, we review some concepts about rough set, local rough set, neighborhood rough set, classical multigranulation rough set and generalized multigranulation rough set [14, 17, 19].

2.1 Rough set

Rough set is an important tool for dealing with uncertain information [16]. In classical rough set theory, we obtain the approximations of a target concept based on the all information classes determined by the objects from universe. The rough set model is defined as follows.

Definition 1 [16] Let $I = (U, N)$ be an information system, where $U = \{x_1, x_2, x_3, \ldots, x_n\}$ denotes the set of finite objects and $N = \{a_1, a_2, a_3, \ldots, a_m\}$ denotes a set of attributes. The $[x]_R$ is the equivalence classes under equivalent relation R defined on U, then for any $X \subseteq U$, the lower and upper approximations of X on R are defined by

$$\begin{aligned} \underline{R}(X) &= \cup\{[x]_R \mid [x]_R \subseteq X, x \in U\}, \\ \overline{R}(X) &= \cup\{[x]_R \mid [x]_R \cap X \neq \emptyset, x \in U\}. \end{aligned} \tag{1}$$

If $\underline{R}(X) = \overline{R}(X)$, the target concept X is a definable set; and if $\underline{R}(X) \neq \overline{R}(X)$, then $< \underline{R}(X), \overline{R}(X) >$ is called rough set. For target concept X, the positive region is $POS(X) = \underline{R}(X)$, the negative region is $NEG(X) = U - \overline{R}(X)$ and the boundary region is $BON(X) = \overline{R}(X) - \underline{R}(X)$. The approximation space also can be written as [14]

$$\begin{aligned} \underline{R}(X) &= \{x \mid [x]_R \subseteq X, x \in U\}, \\ \overline{R}(X) &= \{x \mid [x]_R \cap X \neq \emptyset, x \in U\}. \end{aligned} \tag{2}$$

2.2 Local rough set

In classical rough set theory, we need to consider all the information classes from universe, it is time-consuming and memory intensive, especially in big data. For this issue, scholars have proposed local rough set model. In local rough set, we only need to compute the information classes determined by these objects coming from target concept, it significantly reduces the complexity of approximating X.

Definition 2 [16] Let $I = (U, N)$ be an information system, and $\mathcal{D}$ be an inclusion degree defined on $P(D) \times P(D)$. For any $X \subseteq U$, the lower and upper approximations of X are defined by

$$\begin{aligned} \underline{R_{L,\alpha,\beta}}(X) &= \{x \mid \mathcal{D}(X/[x]_R) \geq \alpha, x \in X\}, \\ \overline{R_{L,\alpha,\beta}}(X) &= \cup\{[x]_R \mid \mathcal{D}(X/[x]_R) > \beta, x \in X\}. \end{aligned} \tag{3}$$

Where $P(U)$ is the power set of U, $0 \leq \beta < 0.5$, $0.5 \leq \alpha \leq 1$ are two parameters in approximation space. The pair $<$

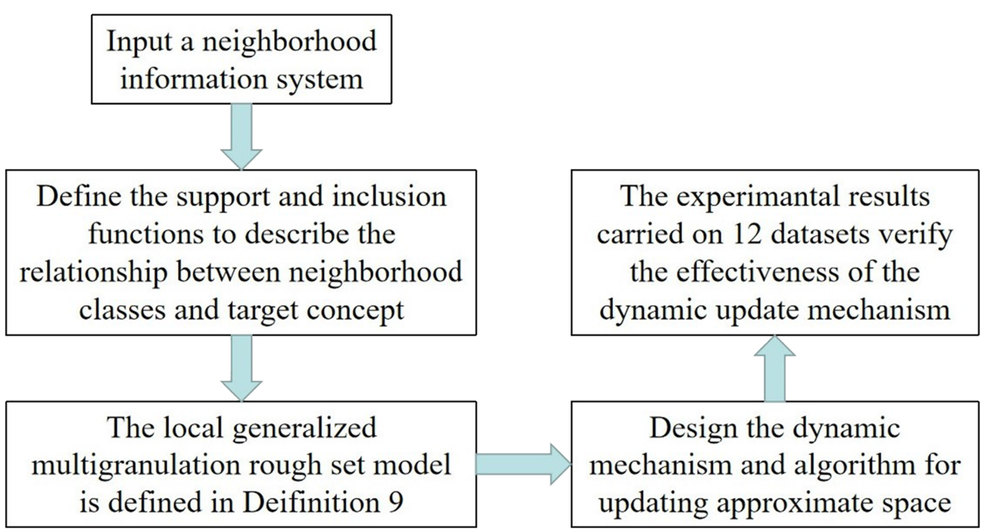

Fig. 1 The chart of proposed approach

$<\underline{R_{L,\alpha,\beta}}(X), \overline{R_{L,\alpha,\beta}}(X)>$ is called local rough set. The local positive region is $POS_{L,\alpha,\beta}(X) = \underline{R_{L,\alpha,\beta}}(X)$, the local negative region is $NEG_{L,\alpha,\beta}(X) = U - \overline{R_{L,\alpha,\beta}}(X)$ and the local boundary region is $BON_{L,\alpha,\beta}(X) = \overline{R_{L,\alpha,\beta}}(X) - \underline{R_{L,\alpha,\beta}}(X)$. Meanwhile, when $\alpha = 1$, $\beta = 0$, the local rough set model will degenerate to classical rough set model as follows.

$$\begin{aligned} \underline{R_{L,1,0}}(X) &= \{x|\mathcal{D}(X/[x]_R) \geq 1, x \in X\} = \{x|[x]_R \subseteq X, x \in U\} = \underline{R}(X), \\ \overline{R_{L,1,0}}(X) &= \cup\{[x]_R|\mathcal{D}(X/[x]_R) > 0, x \in X\} = \{x|[x]_R \cap X \neq \emptyset, x \in U\} \\ &= \overline{R}(X). \end{aligned} \quad (4)$$

2.3 Neighborhood rough set

In an information system $I = (U, N)$, the attribute set $N = C \cup D$, where C is a conditional attribute set and D is a decision attribute set. For any $x \in U$ and $a \in N$, the $a(x)$ denotes the property value of object x on attribute a. If the attributes in C are all numerical, then I is named neighborhood information system [16]. In neighborhood information system, in order to represent the proximity between x_i and x_j in attribute set $N' \subseteq N$, we choose the distance function[19] as follows

$$\Delta(x_i, x_j) = (\sum_{a \in N'} |a(x_i) - a(x_j)|^k)^{1/k}. \quad (5)$$

In this paper, we select $k = 2$ called Euclidean distance to compute the distance between any two objects. Let δ is the radius of neighborhood, the neighborhood classes of $x \in U$ in attribute set $N' \subseteq N$ is defined as

$$\delta_R(x) = \{x_s|\Delta(x, x_s) \leq \delta, x_s \in U\}. \quad (6)$$

Definition 3 [16] Let $I = (U, N)$ be a neighborhood information system. Suppose the radius of neighborhood is δ and the $\delta_i(x)$ is the neighborhood classes under neighborhood relation $R_{R_{N_i}}$ defined on $N_i \subseteq N$, then for any $X \subseteq U$, the lower and upper approximations of X under R_{N_i} are defined by

$$\begin{aligned} \underline{R_{R_{N_i}}}(X) &= \{x|\delta_i(x) \subseteq X, x \in U\}, \\ \overline{R_{R_{N_i}}}(X) &= \{x|\delta_i(x) \cap X \neq \emptyset, x \in U\}. \end{aligned} \quad (7)$$

2.4 Optimistic and pessimistic multigranulation neighborhood rough set

Due to the complexity of things, it is limited to describe an object from one viewpoint. To describe the objects more detail, Qian et al. first proposed multigranulation rough set [17]. Moreover, optimistic multigranulation and pessimistic multigranulation rough approximations are proposed in 2012 by Xu et al. [22], they are defined as follows:

Definition 4 [22] Let $I = (U, N)$ be a neighborhood information system, where the radius of neighborhood is δ, and the $\delta_i(x)$ is neighborhood classes under $R_{R_{N_i}}$ for $X \subseteq U$. Then the lower and upper approximations in optimistic multigranulation of X are defined by

$$\begin{aligned} \underline{OM_{\sum_{i=1}^m R_{R_{N_i}}}}(X) &= \left\{x \in U \mid \vee_{i=1}^m (\delta_i(x) \subseteq X)\right\}, \\ \overline{OM_{\sum_{i=1}^m R_{R_{N_i}}}}(X) &= \left\{x \in U \mid \wedge_{i=1}^m (\delta_i(x) \cap X \neq \emptyset)\right\}. \end{aligned} \quad (8)$$

Where "$\vee$" denotes *or*, "$\wedge$" denotes *and*. If $\underline{OM_{\sum_{i=1}^m R_{R_{N_i}}}} \neq \overline{OM_{\sum_{i=1}^m R_{R_{N_i}}}}(X)$, then $< \underline{OM_{\sum_{i=1}^m R_{R_{N_i}}}}(X), \overline{OM_{\sum_{i=1}^m R_{R_{N_i}}}}(X) >$ is called optimistic multigranulation rough set in neighborhood information system and X is rough.

Definition 5 [22] Continue to the Definition 4, the lower and upper approximations in pessimistic multigranulation of X are as follows.

$$\begin{aligned} \underline{PM_{\sum_{i=1}^m R_{R_{N_i}}}}(X) &= \left\{x \in U \mid \wedge_{i=1}^m (\delta_i(x) \subseteq X)\right\}, \\ \overline{PM_{\sum_{i=1}^m R_{R_{N_i}}}}(X) &= \left\{x \in U \mid \vee_{i=1}^m (\delta_i(x) \cap X \neq \emptyset)\right\}. \end{aligned} \quad (9)$$

If $\underline{PM_{\sum_{i=1}^m R_{R_{N_i}}}} \neq \overline{PM_{\sum_{i=1}^m R_{R_{N_i}}}}$, then $< \underline{PM_{\sum_{i=1}^m R_{R_{N_i}}}}, \overline{PM_{\sum_{i=1}^m R_{R_{N_i}}}} >$ is pessimistic multigranulation rough set in neighborhood information system.

2.5 Generalized multigranulation neighborhood rough set

For an object $x \in U$, if its neighborhood classes is the subset of target concept in all granulation, x is considered to be a member of the pessimistic lower approximation; and if there exists a neighborhood relation R_{N_i} satisfying $\delta_i(x) \subseteq X$, then $x \in \underline{OM_{\sum_{i=1}^m R_{R_{N_i}}}}(X)$. It is too loose while the pessimistic ones is strict in approximating target concept. They are all not effective in practice. Thus, Xu et al. have come up with generalized multigranulation rough sets on equivalence relation in 2017, it is more practical and flexible compared with other models [21]. Inspired by this, we further study the generalized multigranulation neighborhood rough sets in this paper. To illustrate the models, we will first define a support function.

Definition 6 [21] Let $I = (U, N)$ be a neighborhood information system. A support function $P_X^{R_{R_{N_i}}}(x) (i = 1, 2, \ldots, m)$ of x describes the inclusion relation between the neighborhood class $\delta_i(x)$ and X, which is defined as follows:

$$P_X^{R_{R_{N_i}}}(x) = \begin{cases} 1, & \delta_i(x) \subseteq X \\ 0, & others \end{cases} \quad (10)$$

According to the definition of support function, the $x \in \overline{PM_{\sum_{i=1}^{m} R_{R_{N_i}}}(X)}$ only $P_X^{R_{R_{N_i}}}(x) = 1$ holds for $i = 1, 2, \cdots, m$, it is too strict to acquire approximations, especially when the number of granulations is large. Inversely, if there exists a neighborhood relation $R_{R_{N_i}}$ satisfying $P_X^{R_{R_{N_i}}}(x) = 1$, then x is the element of optimistic multigranulation lower approximation, it is also not effective. Thus, we introduce a parameter β, i.e., the ratio of the support function value to m. The higher β is, the stricter our requirements are. The generalized multigranulation neighborhood rough set with $\beta \in (0.5, 1]$ is described in the following definition.

Definition 7 [21] Let $I = (U, N)$ be a neighborhood information system. For support functions $P_X^{R_{R_{N_i}}}(x)(i = 1, 2, \ldots, m)$ and the parameter $\beta \in (0.5, 1]$, the generalize lower and upper multigranulation approximations are defined as follows:

$$\begin{aligned} \underline{P_\beta}(X) &= \left\{ x \in U \middle| \frac{\sum_{i=1}^{m} P_X^{R_{N_i}}(x)}{m} \geq \beta \right\}, \\ \overline{P_\beta}(X) &= \left\{ x \in U \middle| \frac{\sum_{i=1}^{m} (1 - P_{\sim X}^{R_{N_i}}(x))}{m} > 1 - \beta \right\}. \end{aligned} \tag{11}$$

If $\underline{P_\beta}(X) \neq \overline{P_\beta}(X)$, then X is rough and $< \underline{P_\beta}(X), \overline{P_\beta}(X) >$ is called a generalized multigranulation neighborhood rough set with the level β. The boundary region $Bn_\beta(X) = \overline{P_\beta}(X) - \underline{P_\beta}(X)$, otherwise, the X is a definable subset. Obviously, the optimistic multigranulation rough set is a special case when $\beta = 1/m$.

Example 1 A neighborhood information system is shown in Table 1, where $\delta = 0.1$. The universe of we studied is $U = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ and the conditional attributes a_1, a_2, a_3, a_4, a_5 represent *Length*, *Width*, *Popularity*, *Fiber* and *Cotton*, respectively. Suppose the target concept $X = \{x_3, x_5, x_6\}$ and three granules N_1, N_2, N_3, where $N_1 = \{a_1, a_2\}$, $N_2 = \{a_3\}$ and $N_3 = \{a_4, a_5\}$. Obviously, they show the size, popularity and quality of clothes. Each granule denotes different characters, thus we should study the lower and upper approximations from three granules.

From the viewpoint of global generalized multigranulation rough set, we need to obtain all the neighborhood classes determined by objects from U. We first compute the distance matrices based on three granulation.

According to $N_1 = \{a_1, a_2\}$, the distance matrix D_1 is:

$$D_1 = \begin{pmatrix} 0 & 0.0707 & 0.3722 & 0.1044 & 0.3501 & 0.4400 \\ 0.0707 & 0 & 0.3015 & 0.0500 & 0.2807 & 0.3701 \\ 0.3722 & 0.3015 & 0 & 0.2789 & 0.0539 & 0.0806 \\ 0.1044 & 0.0500 & 0.2789 & 0 & 0.2508 & 0.3413 \\ 0.3501 & 0.2807 & 0.0539 & 0.2508 & 0 & 0.0906 \\ 0.4400 & 0.3701 & 0.0806 & 0.3413 & 0.0906 & 0 \end{pmatrix}.$$

Let $\delta = 0.1$, the neighborhood classes on N_1 are:

$\delta_1(x_1) = \{x_1, x_2\}$, $\delta_1(x_2) = \{x_1, x_2, x_4\}$, $\delta_1(x_3) = \{x_3, x_5, x_6\}$,

$\delta_1(x_4) = \{x_2, x_4\}$, $\delta_1(x_5) = \{x_3, x_5, x_6\}$, $\delta_1(x_6) = \{x_3, x_5, x_6\}$.

According to $N_2 = \{a_3\}$, the distance matrix D_2 is:

$$D_2 = \begin{pmatrix} 0 & 0.0200 & 0.0400 & 0.0200 & 0.0700 & 0.0100 \\ 0.0200 & 0 & 0.0600 & 0 & 0.0900 & 0.0300 \\ 0.0400 & 0.0600 & 0 & 0.0600 & 0.0300 & 0.0300 \\ 0.0200 & 0 & 0.0600 & 0 & 0.0900 & 0.0300 \\ 0.0700 & 0.0900 & 0.0300 & 0.0900 & 0 & 0.0600 \\ 0.0100 & 0.0300 & 0.0300 & 0.0300 & 0.0600 & 0 \end{pmatrix}.$$

Similarly, the neighborhood classes on N_2 are:

$\delta_2(x_1) = \{x_1, x_2, x_3, x_4, x_5, x_6\}$, $\delta_2(x_2) = \{x_1, x_2, x_3, x_4, x_5, x_6\}$, $\delta_2(x_3) = \{x_1, x_2, x_3, x_4, x_5, x_6\}$,

$\delta_2(x_4) = \{x_1, x_2, x_3, x_4, x_5, x_6\}$, $\delta_2(x_5) = \{x_1, x_2, x_3, x_4, x_5, x_6\}$, $\delta_2(x_6) = \{x_1, x_2, x_3, x_4, x_5, x_6\}$.

According to $N_3 = \{a_4, a_5\}$, the distance matrix D_3 is:

$$D_3 = \begin{pmatrix} 0 & 0.0608 & 0.3890 & 0.0632 & 0.3354 & 0.0300 \\ 0.0608 & 0 & 0.4205 & 0.1063 & 0.3744 & 0.0316 \\ 0.3890 & 0.4205 & 0 & 0.3257 & 0.0762 & 0.3992 \\ 0.0632 & 0.1063 & 0.3257 & 0 & 0.2729 & 0.0781 \\ 0.3354 & 0.3744 & 0.0762 & 0.2729 & 0 & 0.3499 \\ 0.0300 & 0.0316 & 0.3992 & 0.0781 & 0.3499 & 0 \end{pmatrix}.$$

Table 1 A new neighborhood information system

U	a_1	a_2	a_3	a_4	a_5
x_1	0.58	0.28	0.65	0.56	0.35
x_2	0.59	0.35	0.67	0.50	0.34
x_3	0.62	0.65	0.61	0.68	0.72
x_4	0.55	0.38	0.67	0.58	0.41
x_5	0.57	0.63	0.58	0.71	0.65
x_6	0.58	0.72	0.64	0.53	0.35

We obtain the neighborhood classes on N_3 are:

$\delta_3(x_1) = \{x_1, x_2, x_4, x_6\}$, $\delta_3(x_2) = \{x_1, x_2, x_6\}$, $\delta_3(x_3) = \{x_3, x_5\}$,

$\delta_3(x_4) = \{x_1, x_4, x_6\}$, $\delta_3(x_5) = \{x_3, x_5\}$, $\delta_3(x_6) = \{x_1, x_2, x_4, x_6\}$.

Let $\beta = 0.6$, we obtain that

$$\underline{P_{0.6}}(X) = \{x_3, x_5\}, \quad \overline{P_{0.6}}(X) = \{x_1, x_2, x_3, x_4, x_5, x_6\}.$$

3 Local generalized multigranulation neighborhood rough set model

In order to observe the object in more detail and improve approximation efficiency, we will introduce local generalized multigranulation rough set models in neighborhood information system. And the first we need to introduce a inclusion function to describe the inclusion relation between an object and the union set of neighborhood classes of $x \in X$ on a neighborhood relation.

Definition 8 Let $I = (U, N)$ be a neighborhood information system. The radius of neighborhood is δ and $R_{N_i} \subseteq N(i = 1, 2, 3, \ldots, m)$ are neighborhood relations. The inclusion function $S_X^{R_{N_i}}(x)$ is defined as follows:

$$S_X^{R_{N_i}}(x) = \begin{cases} 1, & x \in N_{iL}(X), \\ 0, & x \notin N_{iL}(X). \end{cases} \tag{12}$$

Where $N_{iL}(X) = \bigcup\{\delta_i(x) | \delta_i(x) \cap X \neq \emptyset, x \in X\}$. The larger the number of $S_X^{R_{N_i}}(x) = 1$ on $R_{N_i}(i = 1, 2, \ldots, m)$, the more likely x is the member of upper approximation.

Definition 9 Let $I = (U, N)$ be a neighborhood information system. Given the neighborhood radius δ and a parameter β, for support functions $P_X^{R_{N_i}}(x)(i = 1, 2, \ldots, m)$ and the parameter $\beta \in (0.5, 1]$, the local generalized multigranulation lower and upper approximations of X under neighborhood relation $R_{N_i}(i = 1, 2, \ldots, m)$ with the level β are respect defined by

$$\begin{aligned} \underline{P_{L,\beta}}(X) &= \left\{ x \in X \left| \frac{\sum_{i=1}^{m} P_X^{R_{N_i}}(x)}{m} \geq \beta \right. \right\}, \\ \overline{P_{L,\beta}}(X) &= \left\{ x \in N_L \left| \frac{\sum_{i=1}^{m} S_X^{R_{N_i}}(x)}{m} > 1 - \beta \right. \right\}. \end{aligned} \tag{13}$$

Where $N_L = \bigcup_{i=1}^{m}\{N_{iL}\}$. If $\underline{P_{L,\beta}}(X) \neq \overline{P_{L,\beta}}(X)$, then X is rough and $< \underline{P_\beta}(X), \overline{P_\beta}(X) >$ is called a local generalized multigranulation neighborhood rough set with the level β. The boundary region $Bn_{L,\beta}(X) = \overline{P_{L,\beta}}(X) - \underline{P_{L,\beta}}(X)$.

Compared with classical rough set, the local generalized rough set defined in Definition 9 possesses some interesting properties.

Proposition 1 Given the neighborhood information system $I = (U, N)$, and the corresponding approximate space $< \underline{P_{L,\beta}}(X), \overline{P_{L,\beta}}(X) >$. Then, for any $X, Y \subseteq U$ and $0.5 < \beta_1 \leq \beta_2 \leq 1$, the following propertied hold:

(1) $\underline{P_{L,\beta}}(X) \subseteq X \subseteq \overline{P_{L,\beta}}(X)$;

(2) $\underline{P_{L,\beta}}(\emptyset) = \overline{P_{L,\beta}}(\emptyset) = \emptyset$, $\underline{P_{L,\beta}}(U) = \overline{P_{L,\beta}}(U) = U$;

(3) $X \subseteq Y$, $\underline{P_{L,\beta}}(X) \subseteq \underline{P_{L,\beta}}(Y)$, $\overline{P_{L,\beta}}(X) \subseteq \overline{P_{L,\beta}}(Y)$;

(4) $\underline{P_{L,\beta}}(X \cap Y) \subseteq \underline{P_{L,\beta}}(X) \cap \underline{P_{L,\beta}}(Y)$, $\overline{P_{L,\beta}}(X \cap Y) \subseteq \overline{P_{L,\beta}}(X) \cap \overline{P_{L,\beta}}(Y)$;

(5) $\underline{P_{L,\beta}}(X \cup Y) \supseteq \underline{P_{L,\beta}}(X) \cup \underline{P_{L,\beta}}(Y)$, $\overline{P_{L,\beta}}(X \cup Y) \supseteq \overline{P_{L,\beta}}(X) \cup \overline{P_{L,\beta}}(Y)$;

(6) $0.5 < \beta_1 \leq \beta_2 \leq 1$, $\underline{P_{L,\beta_1}}(X) \supseteq \underline{P_{L,\beta_2}}(X)$, $\overline{P_{L,\beta_1}}(X) \subseteq \overline{P_{L,\beta_2}}(X)$.

Proof:

(1) $\underline{P_{L,\beta}}(X) \subseteq X$ is obviously to be obtained according to the Definition 9. For $\forall x \in X$, its neighborhood class $\delta_i(x) \cap X \neq \emptyset$ for any granulation R_{N_i}, we can get $\frac{\sum_{i=1}^{m} S_X^{R_{N_i}}(x)}{m} = 1$, thus $x \in \overline{P_{L,\beta}}(X)$ and $X \subseteq \overline{P_{L,\beta}}(X)$ holds.

(2) According to (1), we can get $\underline{P_{L,\beta}}(\emptyset) \subseteq \emptyset \subseteq \overline{P_{L,\beta}}(\emptyset)$, thus the $\underline{P_{L,\beta}}(\emptyset) = \emptyset$. Also, the union set $N_{iL}(X) = \emptyset$ under granulation R_{N_i} when $X = \emptyset$, thus the $\overline{P_{L,\beta}}(\emptyset) = \emptyset$. Analogously, we can prove that $\underline{P_{L,\beta}}(U) = \overline{P_{L,\beta}}(U) = U$.

(3) For $x \in \underline{P_{L,\beta}}(X)$, we have $\delta_i(x) \subseteq X$ in at least $m\beta$ neighborhood relations, since $X \subseteq Y$, we can obtain $\delta_i(x) \subseteq Y$ satisfies in at least $m\beta$ neighborhood relations, thus $x \in \underline{P_{L,\beta}}(Y)$. Furthermore, when $x \in \overline{P_{L,\beta}}(X)$, $x \in N_{iL}(X)$ holds in at least $m(1-\beta)$ granularity. Meanwhile, $X \subseteq Y \Rightarrow N_{iL}(X) \subseteq N_{iL}(Y)$, thus $x \in N_{iL}(X) \Rightarrow x \in N_{iL}(Y)$. So, one can get that $\overline{P_{L,\beta}}(X) \subseteq \overline{P_{L,\beta}}(Y)$.

(4) For any $x \in \underline{P_{L,\beta}}(X \cap Y)$, we can get that

$x \in \underline{P_{L,\beta}}(X \cap Y) \Rightarrow \frac{\sum_{i=1}^{m} P_{X\cap Y}^{R_{N_i}}(x)}{m} \geq \beta \Rightarrow \delta_i(x) \subseteq X \cap Y$ in at least $m\beta$ granularity

$\Rightarrow \delta_i(x) \subseteq X \wedge \delta_i(x) \subseteq Y$ in at least $m\beta$ granularity $\Rightarrow x \in \underline{P_{L,\beta}}(X) \wedge \underline{P_{L,\beta}}(Y) \Rightarrow x \in \underline{P_{L,\beta}}(X) \cap \underline{P_{L,\beta}}(Y)$.

Analogously, for any $x \in \overline{P_{L,\beta}}(X \cap Y)$ we can get that

$x \in \overline{P_{L,\beta}}(X \cap Y) \Rightarrow \frac{\sum_{i=1}^{m} S_{X\cap Y}^{R_{N_i}}(x)}{m} > 1 - \beta$

$\Rightarrow x \in N_{iL}(X \cap Y)$ in at least $m(1-\beta)$ granularity $\Rightarrow x \in N_{iL}(X) \cap N_{iL}(Y)$ in at least $m(1-\beta)$ granularity

$\Rightarrow x \in \overline{P_{L,\beta}}(X) \wedge x \in \overline{P_{L,\beta}}(Y)$

$\Rightarrow x \in \overline{P_{L,\beta}}(X) \cap \overline{P_{L,\beta}}(Y)$.

(5) For any $x \in \underline{P_{L,\beta}}(X) \cup \underline{P_{L,\beta}}(Y)$, we can get that

$x \in \underline{P_{L,\beta}}(X) \cup \underline{P_{L,\beta}}(Y) \Rightarrow x \in \underline{P_{L,\beta}}(X) \vee x \in \underline{P_{L,\beta}}(Y)$

$\Rightarrow \delta_i(x) \subseteq X$ or $\delta_i(x) \subseteq Y$ holds in at least $m\beta$ granularity $\Rightarrow \delta_i(x) \subseteq X \cup Y$ in at least $m\beta$ granularity

$\Rightarrow x \in \underline{P_{L,\beta}}(X \cup Y)$.

Meanwhile, for $x \in \overline{P_{L,\beta}}(X) \cup \overline{P_{L,\beta}}(Y)$, we can get that $x \in \overline{P_{L,\beta}}(X) \cup \overline{P_{L,\beta}}(Y) \Rightarrow x \in N_{iL}(X)$ or $x \in N_{iL}(Y)$ in at least $m(1-\beta)$ granularity

$\Rightarrow x \in N_{iL}(X \cup Y)$ in at least $m(1-\beta)$ granularity $\Rightarrow x \in \overline{P_{L,\beta}}(X \cup Y)$.

(6) When $0.5 < \beta_1 \le \beta_2 \le 1$, if $x \in \underline{P_{L,\beta_2}}(X)$, we know $\frac{\sum_{i=1}^{m} P_X^{R_{N_i}}(x)}{m} \ge \beta_2 \ge \beta_1$, then $x \in \underline{P_{L,\beta_1}}(X)$. Analogously, we can prove $\overline{P_{L,\beta_1}}(X) \subseteq \overline{P_{L,\beta_2}}(X)$.

Example 2 For the neighborhood information system shown in Table 1. The local generalized multigranulation approximations only need to compute the neighborhood classes of objects in target concept. It is effective compared with global ones.

Based on granulation N_1, the distance matrix D_4 is:

$$D_4 = \begin{pmatrix} 0.3722 & 0.3015 & 0 & 0.2789 & 0.0539 & 0.0806 \\ 0.3501 & 0.2807 & 0.0539 & 0.2508 & 0 & 0.0906 \\ 0.4400 & 0.3701 & 0.0806 & 0.3413 & 0.0906 & 0 \end{pmatrix}.$$

Due to $\delta = 0.1$, the neighborhood classes of x_3, x_5, x_6 on N_1 are:

$$\delta_1(x_3) = \{x_3, x_5, x_6\}, \quad \delta_1(x_5) = \{x_3, x_5, x_6\}, \quad \delta_1(x_6) = \{x_3, x_5, x_6\}.$$

Based on N_2, the distance matrix D_5 is:

$$D_5 = \begin{pmatrix} 0.0400 & 0.0600 & 0 & 0.0600 & 0.0300 & 0.0300 \\ 0.0700 & 0.0900 & 0.0300 & 0.0900 & 0 & 0.0600 \\ 0.0100 & 0.0300 & 0.0300 & 0.0300 & 0.0600 & 0 \end{pmatrix}.$$

Similarly, the neighborhood classes of x_3, x_5, x_6 on N_2 are:

$$\delta_2(x_3) = \{x_1, x_2, x_3, x_4, x_5, x_6\}, \quad \delta_2(x_5) = \{x_1, x_2, x_3, x_4, x_5, x_6\},$$
$$\delta_2(x_6) = \{x_1, x_2, x_3, x_4, x_5, x_6\}.$$

Based on granulation N_3, the distance matrix D_6 is:

$$D_6 = \begin{pmatrix} 0.3890 & 0.4205 & 0 & 0.3257 & 0.0762 & 0.3992 \\ 0.3354 & 0.3744 & 0.0762 & 0.2729 & 0 & 0.3499 \\ 0.0300 & 0.0316 & 0.3992 & 0.0781 & 0.3499 & 0 \end{pmatrix}.$$

The neighborhood classes of x_3, x_5, x_6 on N_3 are:

$$\delta_3(x_3) = \{x_3, x_5\}, \quad \delta_3(x_5) = \{x_3, x_5\}, \quad \delta_3(x_6) = \{x_1, x_2, x_4, x_6\}.$$

According to the above results, we know $N_{iL} = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ for all granularity. Give $\beta = 0.6$, based on the definitions of local generalized multigranulation lower and upper approximations, we obtain that

$$\underline{P_{L,0.6}}(X) = \{x_3, x_5\}, \quad \overline{P_{L,0.6}}(X) = \{x_1, x_2, x_3, x_4, x_5, x_6\}$$

.

4 Dynamic updating approximations theories in neighborhood information system

With the development of science and technology, the information in dataset is updated constantly, which includes the variations of objects and attributions in information system. It will be time-consuming if we use a static method to study the target approximation in updated information system. In fact, the updated approximation space has a strong relationship with the original approximation space, thus it would be an effective strategy to update approximate space according to the relationship between updated information system and original one.

To more flexible study the complexity of computing target approximation, in this section, we will focus on the variation of objects which includes adding new objects and deleting original objects from the perspectives of global and local.

4.1 Dynamic updating approximations theories on global generalized multigranulation neighborhood rough set

In this section, we will study the dynamic updating approximations of varied objects from global viewpoint.

Let $I = (U, N)$ be the initial neighborhood approximation space, where U is the set of objects investigated and $R_{N_i} (i = 1, 2, \ldots, m)$ is the neighborhood relation on U. Suppose X is target concept, $\sim X$ is the complementary set in the universe U, and $\delta_i(x)$ is the neighborhood classes of object x under the neighborhood relation $R_{N_i}, i = 1, 2, ..., m$. The $\underline{P_\beta}(X)$ and $\overline{P_\beta}(X)$ represent the initial global generalized multigranulation lower and upper approximations, respectively. When the universe U updated, the neighborhood approximation space I, support and inclusion functions will change accordingly. Thus, define $I' = (U', N)$ be the updated neighborhood approximation space, $P_X^{R_{N_i}}(x)'$ and $S_X^{R_{N_i}}(x)'$ are updated functions, moreover, define $\delta_i(x)'$ is the updated neighborhood classes of object x. Furthermore, $\underline{P_\beta}(X)'$ and $\overline{P_\beta}(X)'$ represent the updated generalized multigranulation lower and upper approximations on global.

4.1.1 Dynamic adding objects on global generalized multigranulation neighborhood rough set

In the initial information system, the target concept X is given in advance, thus it will not change in the updated information system. However, all the neighborhood classes will be affected by the added objects, and the global generalized multigranulation approxiamtions will change accordingly.

Proposition 2 Let $I = (U, N)$ be a neighborhood information system, $R_{N_i}(i = 1, 2, \ldots, m)$ is a neighborhood relation on U, and $X \subseteq U$ is the target concept. Suppose the added object is x_a, the approximations of X will be changed as follows.

1. $\forall x_s \in \underline{P_\beta}(X)$, if $\frac{\sum_{i=1}^{m} P_X^{R_{N_i}}(x_s)'}{m} < \beta$, then $\underline{P_\beta}(X)' = \underline{P_\beta}(X) - \{x_s\}$; $\forall x_s \notin \underline{P_\beta}(X)$, $x_s \notin \underline{P_\beta}(X)'$.
2. If there exists some granularity satisfying $\frac{\sum_{i=1}^{m} P_{\sim X}^{R_{N_i}}(x_a)'}{m} > 1 - \beta$, then $\overline{P_\beta}(X)' = \overline{P_\beta}(X) \cup \{x_a\}$; $\forall x_s \notin \overline{P_\beta}(X)$, $x_s \notin \overline{P_\beta}(X)'$ whether x_a is the element of $\delta_i(x_s)'(i = 1, 2, \ldots, m)$.

Proof:

1. $x_s \in \underline{P_\beta}(X)$ means that there exists enough granulation satisfying $\frac{\sum_{i=1}^{m} P_X^{R_{N_i}}(x_s)}{m} \geq \beta$. After adding x_a to U, the neighborhood classes will change, and if $\frac{\sum_{i=1}^{m} P_X^{R_{N_i}}(x_s)'}{m} < 1 - \beta$, x_s will not the number of lower approximation according to the Definition 7. Also, if $x_s \notin \underline{P_\beta}(X)$, there will not enough granularity satisfying $\delta_i(\overline{x_s}) \subseteq X$. And because the number of neighborhood classes will increase or constant, $\frac{\sum_{i=1}^{m} P_X^{R_{N_i}}(x_s)'}{m} < 1 - \beta$, i.e. $x_s \notin \underline{P_\beta}(X)'$.
2. It is easily to obtain that if $\frac{\sum_{i=1}^{m} P_{\sim X}^{R_{N_i}}(x_a)'}{m} > 1 - \beta$, the $x_a \in \overline{P_\beta}(X)'$. And for $x_s \notin \overline{P_\beta}(X)$. Due to $x_a \notin X$, if $\delta_i(x_s) \nsubseteq X$, $\delta_i(x_s)' \nsubseteq X$ holds, thus x_s still not the number of updated generalized multigranulation upper approximation.

Example 3 From Table 1, it is easy to obtain the generalized multigranulation approximations of $X = \{x_3, x_5, x_6\}$ according to the definitions. Moreover, we study the approximations and their challenges after adding the new object x_7. And the new neighborhood information system is shown in Table 2.

For global generalized multigranulation neighborhood rough set, we need to obtain all neighborhood classes determined by the objects from U. We should calculate the distance matrix D_7, D_8 and D_9 of objects from U on each granulation.

According to granulation N_1, the distance matrix D_7 is:

$$D_7 = \begin{pmatrix} 0 & 0.0707 & 0.3722 & 0.1044 & 0.3501 & 0.4400 & 0.4105 \\ 0.0707 & 0 & 0.3015 & 0.0500 & 0.2807 & 0.3701 & 0.3401 \\ 0.3722 & 0.3015 & 0 & 0.2789 & 0.0539 & 0.0806 & 0.0447 \\ 0.1044 & 0.0500 & 0.2789 & 0 & 0.2508 & 0.3413 & 0.3140 \\ 0.3501 & 0.2807 & 0.0539 & 0.2508 & 0 & 0.0906 & 0.0671 \\ 0.4400 & 0.3701 & 0.0806 & 0.3413 & 0.0906 & 0 & 0.0361 \\ 0.4105 & 0.3401 & 0.0447 & 0.3140 & 0.0671 & 0.0361 & 0 \end{pmatrix}.$$

Take $\delta = 0.1$, the neighborhood classes on N_1 are:

$\delta_1(x_1)' = \{x_1, x_2\}$, $\delta_1(x_2)' = \{x_1, x_2, x_4\}$, $\delta_1(x_3)' = \{x_3, x_5, x_6, x_7\}$,

$\delta_1(x_4)' = \{x_2, x_4\}$, $\delta_1(x_5)' = \{x_3, x_5, x_6, x_7\}$,
$\delta_1(x_6)' = \{x_3, x_5, x_6, x_7\}$, $\delta_1(x_7)' = \{x_3, x_5, x_6, x_7\}$.

Based on granulation N_2, the distance matrix D_8 is:

$$D_8 = \begin{pmatrix} 0 & 0.0200 & 0.0400 & 0.0200 & 0.0700 & 0.0100 & 0.0200 \\ 0.0200 & 0 & 0.0600 & 0 & 0.0900 & 0.0300 & 0 \\ 0.0400 & 0.0600 & 0 & 0.0600 & 0.0300 & 0.0300 & 0.0600 \\ 0.0200 & 0 & 0.0600 & 0 & 0.0900 & 0.0300 & 0 \\ 0.0700 & 0.0900 & 0.0300 & 0.0900 & 0 & 0.0600 & 0.0900 \\ 0.0100 & 0.0300 & 0.0300 & 0.0300 & 0.0600 & 0 & 0.0300 \\ 0.0200 & 0 & 0.0600 & 0 & 0.0900 & 0.0300 & 0 \end{pmatrix}.$$

Similarly, the neighborhood classes on N_2 are:

$\delta_2(x_1)' = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$, $\delta_2(x_2)' = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$,

$\delta_2(x_3)' = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$, $\delta_2(x_4)' = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$,

$\delta_2(x_5)' = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$, $\delta_2(x_6)' = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$,

$\delta_2(x_7)' = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$.

Based on granulation N_3, the distance matrix D_9 is:

$$D_9 = \begin{pmatrix} 0 & 0.0608 & 0.3890 & 0.0632 & 0.3354 & 0.0300 & 0.1208 \\ 0.0608 & 0 & 0.4205 & 0.1063 & 0.3744 & 0.0316 & 0.0640 \\ 0.3890 & 0.4205 & 0 & 0.3257 & 0.0762 & 0.3992 & 0.4789 \\ 0.0632 & 0.1063 & 0.3257 & 0 & 0.2729 & 0.0781 & 0.1703 \\ 0.3354 & 0.3744 & 0.0762 & 0.2729 & 0 & 0.3499 & 0.4360 \\ 0.0300 & 0.0316 & 0.3992 & 0.0781 & 0.3499 & 0 & 0.0943 \\ 0.1208 & 0.0640 & 0.4789 & 0.1703 & 0.4360 & 0.0943 & 0 \end{pmatrix}.$$

Table 2 The new neighborhood information system

U	a_1	a_2	a_3	a_4	a_5
x_1	0.58	0.28	0.65	0.56	0.35
x_2	0.59	0.35	0.67	0.50	0.34
x_3	0.62	0.65	0.61	0.68	0.72
x_4	0.55	0.38	0.67	0.58	0.41
x_5	0.57	0.63	0.58	0.71	0.65
x_6	0.58	0.22	0.64	0.53	0.35
x_7	0.60	0.69	0.67	0.45	0.30

The neighborhood classes on granulation N_3 are:

$\delta_3(x_1)' = \{x_1, x_2, x_4, x_6\}$, $\delta_3(x_2)' = \{x_1, x_2, x_6, x_7\}$,
$\delta_3(x_3)' = \{x_3, x_5\}$, $\delta_3(x_4)' = \{x_1, x_4, x_6\}$,

$\delta_3(x_5)' = \{x_3, x_5\}$, $\delta_3(x_6)' = \{x_1, x_2, x_4, x_6, x_7\}$, $\delta_3(x_7)' = \{x_2, x_6, x_7\}$.

Let $\beta = 0.6$, we obtain that

$$\underline{P_{0.6}}(X)' = \emptyset, \qquad \overline{P_{0.6}}(X)' = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}.$$

Combining the results in the Example 1, due to $x_7 \in \delta_1(x_3)'$, $x_7 \in \delta_2(x_3)'$, $\frac{\sum_{i=1}^{m} P_X^{R_{N_i}}(x_3)'}{m} = \frac{1}{3} < \beta$, thus x_3 is not the number of lower approximation. Similarly, $x_5 \notin \underline{P_{0.6}}(X)'$, i.e. $\underline{P'_{0.6}}(X) = \underline{P_{0.6}}(X) - \{x_3, x_5\} = \emptyset$. And because $\delta_i(x_7)' \cap X \neq \emptyset (i = 3, 5, 6)$, $\frac{\sum_{i=1}^{m} P_{\sim X}^{R'_{N_i}}(x_7)}{m} = 1 > 1 - \beta$, thus x_7 is the number of $\overline{P_{0.6}}(X)'$.

The above results further verify the corrections of Proposition 2.

4.1.2 Dynamic deleting objects on global generalized multigranulation neighborhood rough set

The dynamic updating of objects not only include adding new objects but also deleting old objects in the universe U. Due to the removed objects are randomly, the removed objects may be the number of X. Thus, we will discuss the dynamic updating in the deleted objects whether they belong to X.

Proposition 3 Let $I = (U, N)$ be a neighborhood information system, $R_{N_i} (i = 1, 2, \ldots, m)$ is a neighborhood relation on U, and $X \subseteq U$ is the target concept. Suppose the object removed is $x_d (x_d \in X)$, the approximations of X will change as follows.

1. If $x_d \in \underline{P_\beta}(X)$, then $\underline{P_\beta}(X)' = \underline{P_\beta}(X) - \{x_d\}$; $\forall x_s \notin \underline{P_\beta}(X)$, after deleting $x_d \in X$, $x_s \notin \underline{P_\beta}(X)$.
2. If $x_d \in \overline{P_\beta}(X)$, then $\overline{P_\beta}(X)' = \overline{P_\beta}(X) - \{x_d\}$; $\forall x_s \in \overline{P_\beta}(X)(x_s \neq x_d)$, if there exists some granulation satisfying $\frac{\sum_{i=1}^{m} P_{\sim X}^{R_{N_i}}(x_s)'}{m} \leq 1 - \beta$, then $\overline{P_\beta}(X)' = \overline{P_\beta}(X) - \{x_s\}$.

Proof:

1. Obviously, if the deleted object $x_d \in \underline{P_\beta}(X)$, after removing it, the updated lower approximation will not include x_d. An object x_s is not in the initial lower approximation, which shows that there are not enough granulation such that $\delta_i(x_s) \subseteq X$, also $x_d \in X$, thus when it is deleted, there still not enough granulation satisfying $\frac{\sum_{i=1}^{m} P_X^{R_{N_i}}(x_s)'}{m} \geq \beta$, and then $x_s \notin \underline{P_\beta}(X)'$.
2. After removing x_d, the updated upper approximation will delete x_d from the initial approximation correspondingly. For $x_s \in \overline{P_\beta}(X)(x_s \neq x_d)$, since the deleted object $x_d \in X$, the updated neighborhood classes $\delta_i(x_s)' \cap X$ may be $\emptyset$, if $\frac{\sum_{i=1}^{m} P_{\sim X}^{R_{N_i}}(x_s)'}{m} \leq 1 - \beta$, then the x_s will be not the number of updated generalized multigranulation upper approximation.

Example 4 Continue to Example 1. Removing object $x_d \in X$, the generalized multigranulation approximations will change consistently. And the changes will be different whether x_d is the element of X. In this example, we discuss the updated approximations of generalized multigranulation on global when the removed object is $x_3 \in X$.

According to the neighborhood information system Table 1 and distance matrices in Example 1, we obtain the updated neighborhood classes on three granulation as follows:

$\delta_1(x_1)' = \{x_1, x_2\}, \delta_1(x_2)' = \{x_1, x_2, x_4\}, \delta_1(x_4)' = \{x_2, x_4\}$,
$\delta_1(x_5)' = \{x_5, x_6\}, \delta_1(x_6)' = \{x_5, x_6\}$.
$\delta_2(x_1)' = \{x_1, x_2, x_4, x_5, x_6\}, \delta_2(x_2)' = \{x_1, x_2, x_4, x_5, x_6\}$,
$\delta_2(x_4)' = \{x_1, x_2, x_4, x_5, x_6\}$,
$\delta_2(x_5)' = \{x_1, x_2, x_4, x_5, x_6\}, \delta_2(x_6)' = \{x_1, x_2, x_4, x_5, x_6\}$.
$\delta_3(x_1)' = \{x_1, x_2, x_4, x_6\}, \delta_3(x_2)' = \{x_1, x_2, x_6\}, \delta_3(x_4)' = \{x_1, x_4, x_6\}$,
$\delta_3(x_5)' = \{x_5\}, \delta_3(x_6)' = \{x_1, x_2, x_4, x_6\}$.

Based on the definitions of generalized multigraulation approximations on global, we obtain that

$$\underline{P_\beta}(X)' = \{x_5\}, \quad \overline{P_\beta}(X)' = \{x_1, x_2, x_4, x_5, x_6\}.$$

Since the removed object $x_3 \in \underline{P_\beta}(X)$, updated lower approximation $\underline{P_\beta}(X)' = \underline{P_\beta}(X) - \{x_3\} = \{x_5\}$; and because $\frac{\sum_{i=1}^{m} P_{\sim X}^{R_{N_i}}(x_i)'}{m} > 1 - \beta (x_i \in U')$, $\overline{P_\beta}(X)' = \overline{P_\beta}(X) - \{x_3\} = \{x_1, x_2, x_4, x_5, x_6\}$ after deleting x_3.

The above results verify the Proposition 3.

Proposition 4 Let $I = (U, N)$ be a neighborhood approximation space, where $X \subseteq U$ is the target concept. Suppose the object removed is $x_d \notin X$, and the approximations of X will change as follows:

1. If $x_d \in \underline{P_\beta}(X)$, then $\underline{P_\beta}(X)' = \underline{P_\beta}(X) - \{x_d\}$; $\forall x_s \notin \underline{P_\beta(X)}(x_s \neq x_d)$, if there exists some granulation such that $\frac{\sum_{i=1}^{m} P_X^{R_{N_i}}(x_s)'}{m} \geq \beta$, then $\underline{P_\beta}(X)' = \underline{P_\beta}(X) \cup \{x_s\}$.
2. If $x_d \in \overline{P_\beta}(X)$, $\overline{P'_\beta}(X) = \overline{P_\beta}(X) - \{x_d\}$; $\forall x_s \notin \overline{P_\beta}(X)(x_s \neq x_d)$, then $x_s \notin \overline{P_\beta}(X)'$.

Proof:

1. Deleting x_d, then the updated approximations will remove it from the initial ones. And $x_s \notin \underline{P_\beta}$ shows $\frac{\sum_{i=1}^{m} P_X^{R_{N_i}}}{(} x_s)m < \beta$, after removing $x_d \in\sim X$, the

updated neighborhood classes $\delta_i(x_s)'$ may be the subset of X, if there exists enough multigranulation satisfying $\frac{\sum_{i=1}^{m} P_X^{R_{N_i}}(x_s)'}{m} \geq \beta$, then $x_s \in \underline{P_\beta}(X)'$.

2. It is easily to obtain that the updated upper approximation will remove the deleted object x_d from the initial one. Due to $x_s \in \sim X$, the updated neighborhood classes $\delta_i(x_s)' \cap X = \emptyset$ if $\delta_i(x_s) \cap X = \emptyset$. Therefore, if $x_s \notin \overline{P_\beta}(X)$, then $x_s \notin \overline{P_\beta}(X)'$.

Example 5 Continue to Example 1. In this example, we discuss the updated approximations of global generalized multigranulation when the removed object is $x_1 \notin X$. We obtain the updated neighborhood classes as follows.

$$\delta_1(x_2)' = \{x_2, x_4\}, \quad \delta_1(x_3)' = \{x_3, x_5, x_6\}, \quad \delta_1(x_4)' = \{x_2, x_4\},$$

$$\delta_1(x_5)' = \{x_3, x_5, x_6\}, \quad \delta_1(x_6)' = \{x_3, x_5, x_6\}.$$

$$\delta_2(x_2)' = \{x_2, x_3, x_4, x_5, x_6\}, \quad \delta_2(x_3)' = \{x_2, x_3, x_4, x_5, x_6\},$$
$$\delta_2(x_4)' = \{x_2, x_3, x_4, x_5, x_6\},$$

$$\delta_2(x_5)' = \{x_2, x_3, x_4, x_5, x_6\}, \quad \delta_2(x_6)' = \{x_2, x_3, x_4, x_5, x_6\}.$$

$$\delta_3(x_2)' = \{x_2, x_6\}, \quad \delta_3(x_3)' = \{x_3, x_5\}, \quad \delta_3(x_4)' = \{x_4, x_6\},$$

$$\delta_3(x_5)' = \{x_3, x_5\}, \delta_3(x_6)' = \{x_2, x_4, x_6\}.$$

Based on the definitions of lower and upper generalized multigranulation approximations, we obtain that

$$\underline{P_\beta}(X)' = \{x_3, x_5\}, \qquad \overline{P_\beta}(X)' = \{x_2, x_3, x_4, x_5, x_6\}.$$

After removing $x_1 \notin X$, we find $\frac{\sum_{i=1}^{m} P_X^{R_{N_i}}(x_j)}{m} \geq \beta (j = 3, 5)$, thus the updated lower approximation is the same as the initial one. Since $\frac{\sum_{i=1}^{m} P_{\sim X}^{R_{N_i}}(x_j)}{m} > 1 - \beta (j = 2, 3, 4, 5, 6)$, the updated generalized multigranulation upper approximation $\overline{P_\beta}(X)' = \overline{P_\beta}(X) - \{x_1\} = \{x_2, x_3, x_4, x_5, x_6\}$.

The above results verify the authenticity of Proposition 4.

4.2 Dynamic updating approximations theories on local generalized multigranulation neighborhood rough set

On local generalized multigranulation rough set, we only need to consider the neighborhood classes in target concept X, it is an efficient strategy compared with global ones. Moreover, the dynamic updating strategy would be a more efficient method for approximating target concept when the object set changes. In this section, we will continue to study the dynamic updating approximations of varied objects from local viewpoint.

The prerequisites are the same as those set out in Section 4.1, also, $\underline{P_{L,\beta}}(X)'$ and $\overline{P_{L,\beta}}(X)'$ denote the updated local generalized multigranulation approximations on local respectively.

4.2.1 Dynamic adding objects on local generalized multigranulation neighborhood rough set

In the initial neighborhood information system, the target concept X is given in advance and it will not change in the updated neighborhood information system. However, the neighborhood classes in X will be affected by the added objects, the local generalized multigranulation approximations will change accordingly.

Proposition 5 Let $I = (U, N)$ be a neighborhood information system, R_{N_i} $(i = 1, 2, \ldots, m)$ is the neighborhood relation on U, and X is the target concept. Suppose the added object is x_a, and the approximations of X will change as follows.

1. $\forall x_s \in \underline{P_\beta}(X)$, if $\frac{\sum_{i=1}^{m} P_X^{R_{N_i}}(x_s)'}{m} < \beta$, then $\underline{P'_\beta}(X) = \underline{P_\beta}(X) - \{x_s\}$; $\forall x_s \notin \underline{P_{L,\beta}}(x_s \in X)$, after adding x_a to $\underline{U}$, $x_s \notin \underline{P_{L,\beta}}'$.
2. If $\frac{\sum_{i=1}^{m} S_X^{R_{N_i}}(x_a)'}{m} > 1 - \beta$, then $\overline{P_\beta}(X)' = \overline{P_\beta}(X) \cup \{x_a\}$.

Proof:

1. Since the object $x_a \notin X$, if some neighborhood classes include it, then the inclusion relationship will not hold. Therefore, for $x_s \in \underline{P_\beta}(X)$, it would be removed from local generalized multigranulation lower approximation if $\frac{\sum_{i=1}^{m} P_X^{R_{N_i}}(x_s)'}{m} < \beta$.
2. According to the Definition 9, it is easily to obtain (2).

Example 6 Continue to the Example 3, the Table 2 is the new neighborhood information system after adding x_7 to U. As the same, we suppose the target concept $X = \{x_3, x_5, x_6\}$ and $\delta = 0.1$. For local generalized multigranulation rough set, we need to calculate the neighborhood classes for the object $x \in X$. According to the distance matrices in Example 3, we obtain that

$$\delta_1(x_3)' = \{x_3, x_5, x_6, x_7\}, \quad \delta_1(x_5)' = \{x_3, x_5, x_6, x_7\},$$
$$\delta_1(x_6)' = \{x_3, x_5, x_6, x_7\}.$$

$$\delta_2(x_3)' = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}, \quad \delta_2(x_5)' = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\},$$
$$\delta_2(x_6)' = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}.$$

$$\delta_3(x_3)' = \{x_3, x_5\}, \quad \delta_3(x_5)' = \{x_3, x_5\},$$
$$\delta_3(x_6)' = \{x_1, x_2, x_4, x_6, x_7\}.$$

Based on the above results, we obtain $N_{iL} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\} (i = 1, 2, 3)$. And according to the definitions of local generalized multigranulation approximations, let $\beta = 0.6$, we know that

$$\underline{P_{L,\beta}}(X)' = \emptyset, \quad \overline{P_{L,\beta}}(X)' = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}.$$

Due to $x_7 \in \delta_1(x_i)'$ and $\delta_2(x_i)'$ for $i = 3, 5$, thus $\frac{\sum_{i=1}^{m} P_X^{R_{N_i}}(x_3)'}{m} = \frac{\sum_{i=1}^{m} P_X^{R_{N_i}}(x_5)'}{m} = \frac{1}{3} < 0.6$, $\underline{P_{L,\beta}}(X)' = \underline{P_{L,\beta}}(X) - \{x_3, x_5\} = \emptyset$. Also, $\frac{\sum_{i=1}^{m} S_X^{R_{N_i}}(x_7)'}{m} = 1 > 0.4$, so $\overline{P_{L,\beta}}(X)' = \overline{P_{L,\beta}}(X) \cup \{x_7\}$.

The results further verify the authenticity of Proposition 5.

4.2.2 Dynamic deleting objects on local generalized multigranulation neighborhood rough set

Continue to the Example 4 and Proposition 4, we approximate the target concept X on local perspective. Because the deleted process is randomly, we will discuss the dynamic updating whether the removed object x_d is the element of X.

Proposition 6 Let $I = (U, N)$ be a neighborhood information system, where R_{N_i} $(i = 1, 2, \ldots, m)$ is a neighborhood relation on U. Suppose $X \subseteq U$ is the target concept and the object removed is $x_d \in X$, the local approximations of X will change as follows:

1. $\underline{P_{L,\beta}}(X)' = \underline{P_{L,\beta}}(X) - \{x_d\}$;
2. If $x_d \in \overline{P_{L,\beta}}(X)$, $\overline{P_{L,\beta}}(X)' = \overline{P_{L,\beta}}(X) - \{x_d\}$; $\forall x_s \in \overline{P_{L,\beta}}(X)$, if $\frac{\sum_{i=1}^{m} S_X^{R_{N_i}}(x_s)'}{m} \leq 1 - \beta$, then $\overline{P_{L,\beta}}(X)' = \overline{P_{L,\beta}}(X) - \{x_s\}$.

Proof:

1. It is easy to obtain that the updated lower approximation of generalized multigranulation rough set will remove the object x_s correspondingly after deleting $x_d \in X$.
2. After removing $x_d \in X$, it may be the intersections between some neighborhood classes and the target concept. Therefore, if $\frac{\sum_{i=1}^{m} S_X^{R_{N_i}}(x_s)'}{m} \leq 1 - \beta(x_s \neq x_d)$, the x_s should be deleted from the updated upper approximation.

Example 7 According to distance matrices in Example 4, When the removed object is $x_3 \in X$, the new neighborhood classes are:

$\delta_1(x_5)' = \{x_5, x_6\}$, $\delta_1(x_6)' = \{x_5, x_6\}$.

$\delta_2(x_5)' = \{x_1, x_2, x_4, x_5, x_6\}$, $\delta_2(x_6)' = \{x_1, x_2, x_4, x_5, x_6\}$.

$\delta_3(x_5)' = \{x_5\}, \delta_3(x_6)' = \{x_1, x_2, x_4, x_6\}$.

Let $\beta = 0.6$, we obtain that

$$\underline{P_{L,\beta}}(X)' = \{x_5\}, \quad \overline{P_{L,\beta}}(X)' = \{x_1, x_2, x_4, x_5, x_6\}.$$

Combining the above results with that in 4.2, we have $\underline{P_{L,\beta}}(X)' = \underline{P_{L,\beta}}(X) - \{x_3\} = \{x_5\}$. Also, $\frac{\sum_{i=1}^{m} S_X^{R_{N_i}}(x_j)'}{m} = 1 > \beta(j = 1, 2, 4, 5, 6)$ still holds, thus $\overline{P_{L,\beta}}(X)' = \overline{P_{L,\beta}}(X)$.

The results verify the Proposition 6.

Proposition 7 Let $I = (U, N)$ be a neighborhood information system, R_{N_i} $(i = 1, 2, \ldots, m)$ is a neighborhood relation on U. Suppose X is the target concept and the object removed is $x_d \notin X$, the approximations of X will change as follows.

1. $\forall x_s \notin \underline{P_{L,\beta}}(X)(x_s \in X)$, if there exists enough granulation satisfying $\frac{\sum_{i=1}^{m} P_X^{R_{N_i}}(x_s)'}{m} \geq \beta$, then $\underline{P_{L,\beta}}(X)' = \underline{P_{L,\beta}}(X) \cup \{x_s\}$.
2. $\overline{P_{L,\beta}}(X)' = \overline{P_{L,\beta}}(X) - \{x_d\}$.

Proof:

1. After removing $x_d \notin X$, the inclusion between the updated neighborhood classes and target concept may be hold. Therefore, for $x_s \notin \underline{P_{L,\beta}}(X)$, if $\frac{\sum_{i=1}^{m} P_X^{R_{N_i}}(x_s)'}{m} \geq \beta$, x_s will be the number of the updated upper approximation.
2. It is easy to obtain the results.

Example 8 Continue to the Examples 2 and 5, we discuss the updated approximations of generalized multigranulation on local when the removed object is $x_1 \notin X$.

According to the neighborhood information system and distance matrices in Example 2, we obtain the neighborhood classes as follows:

$\delta_1(x_3)' = \{x_3, x_5, x_6\}$, $\delta_1(x_5)' = \{x_3, x_5, x_6\}$, $\delta_1(x_6)' = \{x_3, x_5, x_6\}$.

$\delta_2(x_3)' = \{x_2, x_3, x_4, x_5, x_6\}$, $\delta_2(x_5)' = \{x_2, x_3, x_4, x_5, x_6\}$, $\delta_2(x_6)' = \{x_2, x_3, x_4, x_5, x_6\}$.

$\delta_3(x_3)' = \{x_3, x_5\}$, $\delta_3(x_5)' = \{x_3, x_5\}$, $\delta_3(x_6)' = \{x_2, x_4, x_6\}$.

Based on the definitions of local generalized multigranulation approximations, we have

$$\underline{P_{L,\beta}}(X)' = \{x_3, x_5\}, \quad \overline{P_{L,\beta}}(X)' = \{x_2, x_3, x_4, x_5, x_6\}.$$

After removing $x_1 \notin X$, we find $\frac{\sum_{i=1}^{m} P_X^{R_{N_i}}(x_6)}{m} = \frac{1}{3} < \beta$, thus the updated lower approximation is the same as the initial one. Also, $\overline{P_{L,\beta}}(X)' = \overline{P_{L,\beta}}(X) - \{x_1\} = \{x_2, x_3, x_4, x_5, x_6\}$.

The above results further verify the authenticity of Proposition 7.

5 The algorithms of obtaining updated generalized multigranulation approximations with varied objects

In this section, we will design algorithms about computing the updated approximations on global and local backgrounds. The challenges in objects which include adding and deleting elements will cause different results correspondingly. Moreover, the difference between static and dynamic updating algorithms will be compared.

5.1 Computing the updated approximations after adding objects

In this part, dynamic algorithms about computing updated approximations of target concept X after adding new objects to U are designed. From the definitions of lower/upper generalized multigranulation approximations, we need to recompute the all neighborhood classes on static viewpoint when the objets change while the dynamic one only need to depict the relations between the x_a and other existing objects. Here, we give the corresponding dynamic algorithms in below. Also, $d_i(x_d, x_s)(x_d \in Uadd)$ denotes the distance between x_d and x_s on neighborhood relation R_{N_i}.

Algorithm 1 Computing the updated global approximations by dynamic method after adding object.

Input:
1. A neighborhood information system $I = (U, N)$, target concept X, and the original neighborhood classes;
2. The original global generalized multigranulation lower and upper approximations, $GGMLA$ and $GGMUA$.

Output: The updated global generalized multigranulation lower and upper approximations $UGGMLA$ and $UGGMUA$

. **begin**
1:set $UGMLGA \leftarrow GGMLA$, $UGMUGA \leftarrow GMUG$; $\delta = 0.1$;
2:add the object x_a;
3:**for** $x_j \in U$ **do**
  for $i = 1 : m$ **do**
    if $d_i(x_j, x_a) \leq \delta$ **then**
      $\delta_i(x_j)' = \delta_i(x_j) \cup \{x_a\}$; `/* Dynamic update the neighborhood classes */`
    else
      $\delta_i(x_j)' = \delta_i(x_j)$;
    end
  end
end
4:**for** $x_j \in GMLGA$ **do**
  if $\frac{\sum_{i=1}^{m} P_X^{R_{N_i}}(x_j)'}{m} < \beta$ **then**
    $UGMLGA = UGMLGA - \{x_j\}$; `/* Dynamic update the global lower approximation */`
  else
    $UGMLGA = UGMLGA$;
  end
end
5:**for** x_a **do**
  if $\frac{\sum_{i=1}^{m} P_{\sim X}^{R_{N_i}}(x_a)'}{m} > 1 - \beta$ **then**
    $UGMUGA = UGMUGA \cup \{x_a\}$; `/* Dynamic update the global upper approximation */`
  else
    $UGMUGA = UGMUGA$;
  end
end
return: $UGMLGA, UGMUGA$;
end

In the global dynamic algorithm Algorithm 1. When the new object x_a added, we only compute the distance between x_a and $x_s \in U$ for all granulation while the static algorithm needs to compute the distance between any two objects in $U \cup \{x_a\}$, thus the complexity of step 3 of dynamic and static algorithms are $O(m \times |U|)$ and $O(m \times |U+1|^2)$. According to the theories of dynamic updating, in order to obtain the updated lower approximation, we only need to obtain the inclusion relation between $x_s \in GMLG$ on m granulation, so the complexity of step 4 is $O(m \times |GMLG|)$. Also, it only need to compare $\frac{\sum_{i=1}^{m} P_{\sim X}^{R_{N_i}}(x_a)'}{m}$ with $1-\beta$ for the added object, thus the complexity is $O(m \times 1)$ in step 5. However, for global static algorithm, the complexity of computing updated approximations are all $O(m \times |U|)$. According to the complexity, we find the dynamic updating method can significantly reduce time consumption.

In the local dynamic algorithm Algorithm 2. Step 3 needs to calculate whether x_d is the element of $\delta_i(x_s)(x_s \in X)$ for all granulation, thus complexity is $O(m \times |X|)$. But for local static algorithm, we need to recompute the distance between $x_s \in X$ and all objects in $U \cup \{x_a\}$, thus its complexity is $O(m \times |X|(|U|+1))$. Also, in the step 4, we need to delete $x_s \in GMLL$ satisfying $\frac{\sum_{i=1}^{m} P_X^{R_{N_i}}(x_s)'}{m} < \beta$ from the original lower approximation, so the complexity is $O(m \times |GMLL|)$, while the local one is $O(m \times |X|)$. Similarly to the Algorithm 1, the complexity of step 5 is $O(m \times 1)$ and the local static one is $O(m \times |N_L'|)$. Due $|GMLL| \leq |X|$ and $1 \leq |N_L'|$, thus the time consumption of local dynamic is less than the local static one. Moreover, the time complexity of updated algorithms on global and local after adding new object are shown in Table 3.

For dynamic algorithm, they only need to compare x_a with the original neighborhood classes, the complexity of global and local dynamic algorithms are $O(m \times |U| + m \times |GMLG| + m \times 1)$ and $O(m \times |X| + m \times |GMLL| + m \times 1)$ respectively. Since $|GMLL| = |GMLG|$ and $|X| < |U|$, the time consumption of local dynamic algorithm is less than global one. In conclusion, compared with static algorithms, the dynamic algorithms can significantly reduce the time complexity of approximating X.

5.2 Computing the updated approximations after deleting objects

In this part, we design the dynamic algorithms about computing updated approximations of target concept X when deleting existing objects. Since the removed $x_d \in U$ and the number of neighborhood classes only can decrease not rather increase, we only deleted the x_s from original information classes in dynamic computing process, which significantly reduce the time consumption.

Algorithm 2 Computing the updated local approximations by dynamic method after adding object.

Input:
1. A neighborhood information system $I = (U, N)$, a target concept X and the new object x_a;
2. The original local generalized multigranulation lower and upper approximations, $LGMLA$ and $LGMUA$.

Output: The updated local generalized multigranulation lower and upper approximations $ULGMLA$ and $ULGMUA$

. **begin**
  1:set $ULGMLA \leftarrow LGMLA$, $ULGMUA \leftarrow LGMUA$; $\delta = 0.1$;
  2:add the object x_a;
  3:**for** $x_j \in X$ **do**
    for $i = 1 : m$ **do**
      if $d_i(x_j, x_a) \leq \delta$ **then**
        $\delta_i(x_j)' = \delta_i(x_j) \cup \{x_a\}$; /* Dynamic update the neighborhood classes */
      else
        $\delta_i(x_j)' = \delta_i(x_j)$;
      end
    end
  end
  4:**for** $x_j \in ULGMLA$ **do**
    if $\frac{\sum_{i=1}^{m} P_X^{R_{N_i}}(x_j)}{m}' < \beta$ **then**
      $ULGMLA = ULGMLA - \{x_j\}$; /* Dynamic update the local lower approximation */
    else
      $ULGMLA = ULGMLA$;
    end
  end
  5:**for** x_a **do**
    if $\frac{\sum_{i=1}^{m} S_X^{R_{N_i}}}{(} x_a) m' > 1 - \beta$ **then**
      $ULGMUA = ULGMUA \cup \{x_a\}$; /* Dynamic update the local upper approximation */
    else
      $ULGMUA = ULGMUA$;
    end
  end
  return: $ULGMLA, ULGMUA$;
end

Algorithm 3 Computing the updated global approximations by dynamic method after deleting object.

Input:
1. A neighborhood information system $I = (U, N)$, target concept X, and the original neighborhood classes;
2. The original global generalized multigranulation lower and upper approximations $GGMLA$ and $GGMUA$.

Output: The updated global generalized multigranulation lower and upper approximations, $UGGMLA$ and $UGGMUA$.

begin
  1:set $UGGMLA \leftarrow GGMLA, UGGMUA \leftarrow \emptyset$; $\delta = 0.1$;
  2:delete the object x_d;
  3:$U = U - \{x_d\}$, $X = X - \{x_d\}$, $UGGMLA = UGGMLA - \{x_d\}$, $GGMUA = GGMUA - \{x_d\}$;
  4:**for** $x_j \in U$ **do**
    for $i = 1 : m$ **do**
      if $x_d \in \delta_i(x_j)$ **then**
        $\delta_i(x_j)' = \delta_i - \{x_d\}$; /* Dynamic update the neighborhood classes */
      else
        $\delta_i(x_j)' = \delta_i$;
      end
    end
  end
  5:**for** $x_j \in U - UGGMLA$ **do**
    if $\frac{\sum_{i=1}^{m} P_X^{R_{N_i}}(x_j)}{m}' \geq \beta$ **then**
      $UGGMLA = UGGMLA \cup \{x_j\}$; /* Dynamic update the global lower approximation */
    else
      $UGGMLA = UGGMLA$;
    end
  end
  6:**for** $x_j \in GGMUA$ **do**
    if $\frac{\sum_{i=1}^{m} P_X^{R_{N_i}}}{(} x_j) m' > 1 - \beta$ **then**
      $UGGMUA = UGGMUA \cup \{x_j\}$; /* Dynamic update the global upper approximation */
    else
      $UGGMUA = UGGMUA$;
    end
  end
  return: $UGGMLA, UGGMUA$;
end

Table 3 The time complexity of algorithms on global and local after adding new objects

Algorithms	Global	Local
Static	$O(m \times \lvert U+1\rvert^2 + 2 \times m \times \lvert U\rvert)$	$O(m \times \lvert X\rvert\lvert U+1\rvert + m \times \lvert X\rvert + m \times \lvert N_L'\rvert)$
Dynamic	$O(m \times \lvert U\rvert + m \times \lvert GMLG\rvert + m \times 1)$	$O(m \times \lvert X\rvert + m \times \lvert GMLL\rvert + m \times 1)$

The Algorithm 3 shows the process of computing updated approximations after deleting objects. In step 3, we delete x_d from the all original neighborhood classes and don't need to recompute the distance matric, thus the time complexity on global dynamic algorithm is $O(m(|U|-1))$, while the global static one is $O(|U|-1)^2$. For obtaining the updated lower approximation, the global static algorithms only needs to compare $|U|-1$ objects with target concept while the global dynamic only needs to observe the relationship between object in $U-UGGMLA$ and target concept, thus the complexity of global static and dynamic method is $O(m(|U|-1))$ and $O(m|U-UGGMLA|)$. Similarly, the time complexity of obtaining global upper approximation of static and dynamic algorithm is $O(m(|U|-1))$ and $O(m|UGGMUA|)$. Since $O(m(|U|-1)) \ll O(m(|U|-1)^2)$, $O(m|UGGMUA|) \ll O(m(|U|-1))$, and $O(m|U-UGGMLA|) \ll O(m(|U|-1))$, thus the global dynamic algorithm could significantly reduce the time when deleting objects.

The Algorithm 4 shows the dynamic updating process when deleting objects from local perspective. In step 3, it only needs to delete x_d from the original neighborhood classes determined by X', thus the time complexity is $O(m \times |X|)$ while the local static one is $O(m \times |X'|(|U|-1))$. According to the propositions of dynamic mechanism, we know the individual could be the member of lower approximation due to the reduction of non-target collection objects while the upper approximation may be decrease due to the reduction of target concept. Therefore, the local dynamic algorithm only needs to compute the function value of individual in $X-ULGMLA$ for obtaining the local lower approximation, its complexity is $O(m|X'-ULGMLA|)$ while that of local static algorithm is $O(m|X'|)$. In addition, for obtain the lower upper approximation, the number of objects observed in adynamic algorithm is $LGMUA$ and that of local static is $\cup\delta_i(x_j)'(x_j \in X', i = 1, 2, \cdots, m)$ according to the definition of local generalized multigranulation rough set, thus the time complexity of obtaining local upper approximation is $O \times (m)$ and $O \times (m| \cup \delta_i(x_j)'|)$. Since $|X'| \ll |X'|(|U|-1)$ and $1 \ll |\cup \delta_i(x_j)'|$, the time complexity of dynamic algorithm is less than static one on local. The more detail of complexity of updated algorithms on global and local after deleting object are shown in Table 4.

According to the above results, we know the time complexity of dynamic algorithms is less than that of static ones on global and local viewpoints. From the Table 4, because $|X'| \ll |U|-1$ and $1 \ll |UGGMUA|$, the local dynamic algorithm can significantly reduce the time complexity of approximating target concept compared with the global dynamic algorithm, especially in big data.

Algorithm 4 Computing the local updated approximations by dynamic method after deleting object.

Input:
1. A neighborhood information system $I=(U,N)$, a target concept X, and the original neighborhood classes;
2. The original local generalized multigranulation lower and upper approximation, $LGMLA$ and $LGMUA$.

Output: The updated local generalized multigranulation lower and upper approximations, $ULGMLA$ and $ULGMUA$.

begin
  1:set $ULGMLA \leftarrow LGMLA, ULGMUA \leftarrow \emptyset$; $\delta = 0.1$;
  2:delete the object x_d;
  3:$X = X - \{x_d\}, ULGMLA = ULGMLA - \{x_d\}, LGMUA = LGMUA - \{x_d\}$;
  4:**for** $x_j \in X$ **do**
    for $i = 1 : m$ **do**
      if $x_d \in \delta_i(x_j)$ **then**
        $\delta_i(x_j)' = \delta_i(x_j) - \{x_d\}$; /* Dynamic update the neighborhood classes */
      else
        $\delta_i(x_j)' = \delta_i(x_j)$;
      end
    end
  end
  5:**for** $x_j \in X - ULGMLA$ **do**
    if $\frac{\sum_{i=1}^{m} P_X^{R_{N_i}}(x_j)'}{m} \geq \beta$ **then**
      $ULGMLA = ULGMLA \cup \{x_j\}$; /* Dynamic update the local lower approximation */
    else
      $ULGMLA = ULGMLA$;
    end
  end
  6:**for** $x_j \in LGMUA$ **do**
    if $\frac{\sum_{i=1}^{m} S_X^{R_{N_i}}(x_j)'}{m} \leq 1 - \beta$ **then**
      $ULGMUA = ULGMUA \cup \{x_j\}$; /* Dynamic update the local upper approximation */
    else
      $ULGMUA = ULGMUA$;
    end
  end
  return: $ULGMLA, ULGMUA$;
end

6 Experiments and analysis

In Section 5, we have designed dynamic updating algorithms and illustrated its effectiveness by analysing their time complexity. Now, we will further verify the advantages

Table 4 The time complexity of algorithms on global and local after deleting object

Algorithms	Global	Local
Static	$O(m \times ((\|U\|+1)^2 + 2(\|U\|-1)))$	$O(m(\times\|X'\|(\|U\|-1)+\|X'\|+\|\cup\delta_i(x_j)'\|))$
Dynamic	$O(m \times ((\|U\|-1)+\|U-UGGMLA\|+\|UGGMUA\|))$	$O(m \times (\|X'\|+\|X'-ULGMLA\|+1))$

of local dynamic algorithm through the time consumption in experiments. The experiments are carried on two aspects: adding new objects and deleting existing ones.

6.1 Experiment design

All the algorithms are implemented in Matlab 2016b and carry out on a personal computer with Intel(R) Core(TM) i5-1135G7 CPU@2.40GH 2.42GH, and 16 GB memory. For verify the effective of dynamic algorithms while approximating concept on various objects, we choose twelve data sets from UCI [2] to carry experiments. And the detailed information for the six data sets is shown in Table 5.

In all experiments, for each data set, they are first preprocessed by normalizing each variable into the unit interval. First, we fix $\delta = 0.1$ and the target concept where its objects are the front 10% of these data sets, and then suppose three granulation and each of them is composed by third of all features in order. For these algorithms of adding new objects, we select 50% of the data sets as their basic sets and then add 10% of the remaining objects to them for ten times. Meanwhile, for the dynamic algorithms of deleting objects, we delete 5% of the objects in U for ten times. Also, the unit of time consumption is seconds. In order to detail observe the trend of time consumption, we take $\beta = 0.6, 0.8, 1$.

6.2 Experiments results when adding new objects

In this section, we verify the effectiveness of local dynamic algorithm when adding objects. The Tables 6, 7 and 8 record the consume time of adding new individuals with different parameters. In 12 datasets, we could obtain that the consume time increases with the number of objects increases on three algorithms. For local dynamic algorithm, it ignores the other redundant information about individuals out of target concept and further reduces the time due to dynamic computing, thus the consume time of local dynamic algorithm is always the lowest. Also, the more vivid comparison is shown in the Figs. 2, 3 and 4, where the blue, rose red and bright red lines represent the computational time on global dynamic, local static and dynamic algorithms, whose abbreviations are GDA, LSA and LDA respectively. The height of bar chart shows that the time consumption of approximating X on local dynamic algorithm is less than that on local static and global dynamic ones, which can further verify the effectiveness of dynamic updating algorithm and the advantages of local rough set.

Table 5 Datasets description

Nos	Datasets	Individuals	Features
set1	Airfoil Self-Noise	1503	5
set2	Combined Cycle Power Plant	9568	4
set3	Electrical Grid Stability Simulated Data	1000	13
set4	First-order theorem proving	6118	50
set5	FrogsMFCCs	7195	22
set6	Human Activity Recognition Using Smartphones	11934	17
set7	MAGIC Gamma Telescope MGT	19020	10
set8	page-blocks	5472	10
set9	SkillCraft1 Master Table	3395	19
set10	spambase	4601	57
set11	wilt	4839	5
set12	wine quality white	4898	11

Table 6 The consume time of adding objects by different approximate algorithms when $\beta = 0.6$

Dataset	Algorithm	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
	Local static	0.026	0.028	0.028	0.029	0.029	0.030	0.030	0.030	0.032	0.033
set1	Local dynamic	0.014	0.015	0.015	0.016	0.016	0.018	0.018	0.019	0.018	0.021
	Global dynamic	0.021	0.030	0.044	0.049	0.056	0.064	0.072	0.082	0.092	0.102
	Local static	0.905	0.973	1.054	1.123	1.190	1.263	1.336	1.403	1.471	1.575
set2	Local dynamic	0.125	0.188	0.246	0.319	0.400	0.471	0.551	0.629	0.676	0.741
	Global dynamic	0.325	0.638	1.109	1.446	1.867	2.298	2.788	3.249	3.687	4.091
	Local static	0.498	0.534	0.565	0.605	0.638	0.680	0.720	0.761	0.799	0.840
set3	Local dynamic	0.110	0.138	0.172	0.210	0.253	0.293	0.346	0.371	0.414	0.449
	Global dynamic	0.224	0.423	0.625	0.851	1.129	1.389	1.689	1.890	2.139	2.361
	Local static	0.182	0.194	0.205	0.218	0.229	0.241	0.252	0.264	0.275	0.286
set4	Local dynamic	0.056	0.066	0.074	0.088	0.098	0.111	0.124	0.137	0.149	0.159
	Global dynamic	0.122	0.237	0.334	0.448	0.566	0.683	0.824	0.954	1.185	1.465
	Local static	0.258	0.273	0.290	0.306	0.322	0.334	0.355	0.376	0.434	0.460
sest5	Local dynamic	0.072	0.076	0.083	0.102	0.105	0.116	0.129	0.144	0.170	0.177
	Global dynamic	0.123	0.240	0.344	0.456	0.587	0.722	0.886	1.013	1.171	1.285
	Local static	1.209	1.247	1.393	1.528	1.591	1.642	1.754	1.848	2.008	2.028
set6	Local dynamic	0.164	0.219	0.273	0.371	0.495	0.584	0.655	0.745	0.828	0.895
	Global dynamic	0.387	0.732	1.156	1.634	2.041	2.549	2.975	3.458	3.854	4.357
	Local static	3.601	3.711	4.008	4.166	4.386	4.638	4.907	5.184	5.363	5.607
set7	Local dynamic	0.409	0.661	1.015	1.328	1.545	1.801	2.019	2.262	2.580	2.743
	Global dynamic	0.989	2.022	3.343	4.567	5.693	6.836	8.174	9.405	10.781	12.347
	Local static	0.366	0.400	0.422	0.444	0.469	0.494	0.521	0.543	0.570	0.596
set8	Local dynamic	0.068	0.083	0.099	0.118	0.143	0.161	0.181	0.220	0.256	0.278
	Global dynamic	0.130	0.255	0.370	0.539	0.695	0.831	0.964	1.130	1.381	1.422
	Local static	0.097	0.101	0.108	0.112	0.120	0.125	0.128	0.138	0.148	0.153
set9	Local dynamic	0.041	0.040	0.047	0.058	0.063	0.062	0.073	0.076	0.081	0.085
	Global dynamic	0.053	0.111	0.131	0.169	0.200	0.242	0.276	0.317	0.354	0.394
	Local static	0.146	0.155	0.167	0.179	0.188	0.198	0.212	0.219	0.224	0.230
set10	Local dynamic	0.050	0.049	0.055	0.058	0.064	0.069	0.126	0.140	0.151	0.171
	Global dynamic	0.078	0.127	0.189	0.276	0.352	0.495	0.576	0.649	0.748	1.007
	Local static	0.301	0.321	0.346	0.368	0.390	0.412	0.435	0.455	0.478	0.498
set11	Local dynamic	0.058	0.073	0.090	0.111	0.126	0.146	0.177	0.192	0.218	0.235
	Global dynamic	0.101	0.189	0.297	0.369	0.453	0.699	0.809	0.925	1.040	1.161
	Local static	0.217	0.241	0.264	0.282	0.292	0.306	0.320	0.333	0.350	0.364
set12	Local dynamic	0.052	0.060	0.072	0.082	0.095	0.106	0.115	0.128	0.142	0.159
	Global dynamic	0.104	0.181	0.275	0.358	0.451	0.545	0.640	0.734	0.836	0.928

With the increasing of the number of objects, the computational time also increases, and the gap between local and global process becomes more obvious. That's because the global dynamic algorithm need to compare the added objects with original universe U, thus the computing time increases rapidly with the number of updates increases. However, the local dynamic only needs to compare the added objects with target concept X on the base of original approximations, so the time consumption increases relative slow. Also, since the dynamic algorithm only needs to compare the added objects with initial objects while the static one needs to recompute the neighborhood classes

Table 7 The consume time of adding objects by different approximate algorithms when $\beta = 0.8$

Dataset	Algorithm	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
	Local static	0.052	0.055	0.055	0.055	0.058	0.058	0.062	0.062	0.063	0.065
set1	Local dynamic	0.026	0.029	0.030	0.032	0.032	0.033	0.037	0.037	0.039	0.040
	Global dynamic	0.019	0.029	0.039	0.049	0.057	0.065	0.076	0.088	0.102	0.110
	Local static	0.908	0.978	1.054	1.127	1.195	1.270	1.337	1.411	1.476	1.542
set2	Local dynamic	0.128	0.184	0.244	0.300	0.391	0.471	0.549	0.623	0.673	0.742
	Global dynamic	0.321	0.621	1.067	1.438	1.846	2.264	2.738	3.172	3.586	3.979
	Local static	0.489	0.533	0.566	0.604	0.642	0.681	0.726	0.759	0.796	0.896
set3	Local dynamic	0.110	0.136	0.177	0.227	0.288	0.325	0.388	0.417	0.469	0.508
	Global dynamic	0.225	0.452	0.645	0.860	1.140	1.385	1.695	1.894	2.146	2.366
	Local static	0.185	0.193	0.204	0.217	0.229	0.240	0.253	0.262	0.272	0.283
set4	Local dynamic	0.056	0.067	0.075	0.085	0.098	0.108	0.131	0.155	0.163	0.184
	Global dynamic	0.123	0.234	0.355	0.515	0.638	0.800	0.939	1.101	1.120	1.222
	Local static	0.272	0.282	0.298	0.319	0.339	0.345	0.358	0.377	0.387	0.404
sest5	Local dynamic	0.063	0.067	0.075	0.097	0.102	0.111	0.126	0.137	0.164	0.167
	Global dynamic	0.127	0.254	0.354	0.461	0.581	0.740	0.879	1.013	1.194	1.341
	Local static	1.231	1.286	1.373	1.497	1.584	1.749	1.793	1.886	2.002	2.089
set6	Local dynamic	0.159	0.217	0.266	0.357	0.481	0.585	0.638	0.735	0.813	0.895
	Global dynamic	0.380	0.760	1.134	1.635	2.101	2.588	3.046	3.516	5.262	10.429
	Local static	3.628	3.838	4.043	4.233	4.460	4.720	4.941	5.254	5.520	5.668
set7	Local dynamic	0.416	0.686	1.059	1.323	1.562	1.821	2.018	2.279	2.521	2.776
	Global dynamic	1.045	2.034	3.332	4.588	5.783	6.966	8.213	9.545	10.943	12.379
	Local static	0.376	0.404	0.422	0.450	0.471	0.497	0.528	0.548	0.573	0.604
set8	Local dynamic	0.065	0.084	0.096	0.116	0.137	0.162	0.179	0.216	0.251	0.276
	Global dynamic	0.131	0.261	0.372	0.539	0.678	0.819	0.952	1.121	1.271	1.424
	Local static	0.092	0.100	0.106	0.112	0.114	0.120	0.127	0.134	0.141	0.149
set9	Local dynamic	0.041	0.048	0.050	0.053	0.062	0.064	0.072	0.078	0.081	0.085
	Global dynamic	0.056	0.090	0.115	0.160	0.197	0.241	0.276	0.322	0.358	0.401
	Local static	0.145	0.152	0.162	0.174	0.183	0.197	0.204	0.216	0.224	0.233
set10	Local dynamic	0.045	0.049	0.051	0.058	0.061	0.068	0.125	0.134	0.147	0.160
	Global dynamic	0.077	0.130	0.195	0.290	0.363	0.499	0.584	0.663	0.765	0.968
	Local static	0.309	0.330	0.347	0.370	0.389	0.410	0.431	0.454	0.493	0.507
set11	Local dynamic	0.061	0.074	0.094	0.108	0.127	0.148	0.170	0.191	0.211	0.232
	Global dynamic	0.103	0.193	0.289	0.367	0.452	0.689	0.805	0.927	1.054	1.163
	Local static	0.233	0.259	0.331	0.344	0.370	0.398	0.400	0.417	0.435	0.439
set12	Local dynamic	0.054	0.064	0.077	0.087	0.099	0.112	0.124	0.134	0.150	0.168
	Global dynamic	0.095	0.187	0.276	0.363	0.459	0.547	0.644	0.742	0.840	0.944

determined by the objects in U', thus the gap between static and dynamic is becomes larger with the increasing of U. The experimental results fully demonstrate the effectiveness of local dynamics algorithm. Thus, we can conclude that the local dynamic updating algorithm is a better method than the local static and global algorithms.

6.3 Experiments results when deleting objects

In this section, we further verify the effectiveness of local dynamic algorithm when deleting objects. The consume time results are recorded and described in Tables 9, 10 and 11 and Figs. 5, 6 and 7. From Tables 9, 10 and 11, we obtain that the consume time decreases as the number of reduced objects increases, and the global algorithm is slower than the local algorithm. The rank of running time by different algorithms is Global dynamic > Local static > Local dynamic. Theses time results reflect the effectiveness of local rough set and dynamic mechanism.

The Figs. 5, 6 and 7 show the time consumption in approximating target concept X on global dynamic, local static and dynamic algorithms under generalized

Table 8 The consume time of adding objects by different approximate algorithms when $\beta = 1$

Dataset	Algorithm	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
	Local static	0.027	0.027	0.028	0.029	0.029	0.030	0.031	0.031	0.032	0.032
set1	Local dynamic	0.013	0.015	0.015	0.016	0.016	0.016	0.017	0.018	0.019	0.019
	Global dynamic	0.020	0.030	0.042	0.048	0.059	0.065	0.075	0.088	0.100	0.110
	Local static	0.940	1.002	1.077	1.154	1.255	1.295	1.375	1.447	1.517	1.591
set2	Local dynamic	0.128	0.188	0.269	0.356	0.467	0.582	0.677	0.747	0.810	0.889
	Global dynamic	0.324	0.625	1.080	1.461	2.036	2.900	3.244	3.749	4.190	4.688
	Local static	0.488	0.525	0.559	0.596	0.636	0.674	0.718	0.752	0.791	0.832
set3	Local dynamic	0.111	0.142	0.175	0.219	0.257	0.303	0.351	0.377	0.411	0.462
	Global dynamic	0.226	0.451	0.644	0.851	1.134	1.397	1.692	1.915	2.145	2.389
	Local static	0.186	0.195	0.210	0.219	0.232	0.283	0.303	0.314	0.319	0.333
set4	Local dynamic	0.071	0.080	0.094	0.105	0.115	0.117	0.121	0.135	0.145	0.157
	Global dynamic	0.131	0.247	0.342	0.456	0.584	0.718	0.845	0.967	1.087	1.217
	Local static	0.252	0.268	0.285	0.305	0.315	0.332	0.349	0.363	0.378	0.395
sest5	Local dynamic	0.066	0.078	0.079	0.107	0.111	0.120	0.126	0.130	0.147	0.158
	Global dynamic	0.130	0.241	0.346	0.455	0.562	0.705	0.841	0.974	1.137	1.257
	Local static	1.193	1.235	1.324	1.460	1.541	1.627	1.724	1.947	2.054	2.208
set6	Local dynamic	0.172	0.209	0.282	0.358	0.481	0.585	0.623	0.692	0.786	0.862
	Global dynamic	0.393	0.771	1.159	1.639	2.007	3.722	6.798	7.867	8.523	9.833
	Local static	3.565	3.772	4.024	4.272	4.520	4.677	4.964	5.224	5.569	5.751
set7	Local dynamic	0.392	0.669	1.033	1.277	1.515	1.739	2.017	2.263	2.492	2.798
	Global dynamic	1.103	2.016	3.318	4.551	5.782	6.913	8.292	9.546	10.856	12.370
	Local static	0.366	0.397	0.413	0.444	0.469	0.481	0.504	0.530	0.561	0.587
set8	Local dynamic	0.066	0.085	0.102	0.119	0.142	0.164	0.197	0.237	0.289	0.318
	Global dynamic	0.134	0.246	0.350	0.518	0.656	0.816	1.110	1.281	1.452	1.487
	Local static	0.098	0.103	0.106	0.111	0.131	0.140	0.139	0.146	0.173	0.188
set9	Local dynamic	0.041	0.042	0.045	0.052	0.063	0.065	0.074	0.076	0.080	0.084
	Global dynamic	0.053	0.083	0.115	0.155	0.191	0.242	0.284	0.316	0.349	0.395
	Local static	0.143	0.151	0.156	0.172	0.182	0.193	0.202	0.214	0.221	0.230
set10	Local dynamic	0.049	0.048	0.053	0.058	0.063	0.070	0.128	0.136	0.152	0.165
	Global dynamic	0.076	0.129	0.196	0.281	0.365	0.506	0.581	0.659	0.762	0.949
	Local static	0.298	0.320	0.345	0.366	0.387	0.410	0.432	0.450	0.476	0.496
set11	Local dynamic	0.058	0.074	0.091	0.109	0.126	0.148	0.174	0.191	0.214	0.232
	Global dynamic	0.102	0.192	0.290	0.370	0.453	0.695	0.805	0.927	1.054	1.166
	Local static	0.247	0.308	0.323	0.334	0.350	0.364	0.393	0.395	0.410	0.431
set12	Local dynamic	0.055	0.061	0.073	0.087	0.102	0.108	0.119	0.133	0.157	0.178
	Global dynamic	0.099	0.190	0.284	0.377	0.478	0.630	0.762	0.884	0.988	1.012

multigranulation backgrounds. Meanwhile, the meanings of lines are the same with that in Figs. 2, 3 and 4. From these figures, we find the time consumption $LDA < LSA$ and $LDA < GDA$ on the same dataset, and they are all reduced as the data decreases. Since the local dynamic algorithm's complexity is less than local static and global dynamic ones, the gaps in complexity in them get smaller, thus the time gap between them becomes smaller with the size of data set narrows. The above experimental results further illustrate the effectiveness of the local dynamic update algorithm. In actual life, the proposed dynamic algorithm would be a powerful tool for describing uncertainties.

7 Conclusions and further work

With the development of science and technology, the amount of data is increasingly growing and updating faster. How to effectively deal with the updated information is a challenge for classical rough set. Since the local rough set

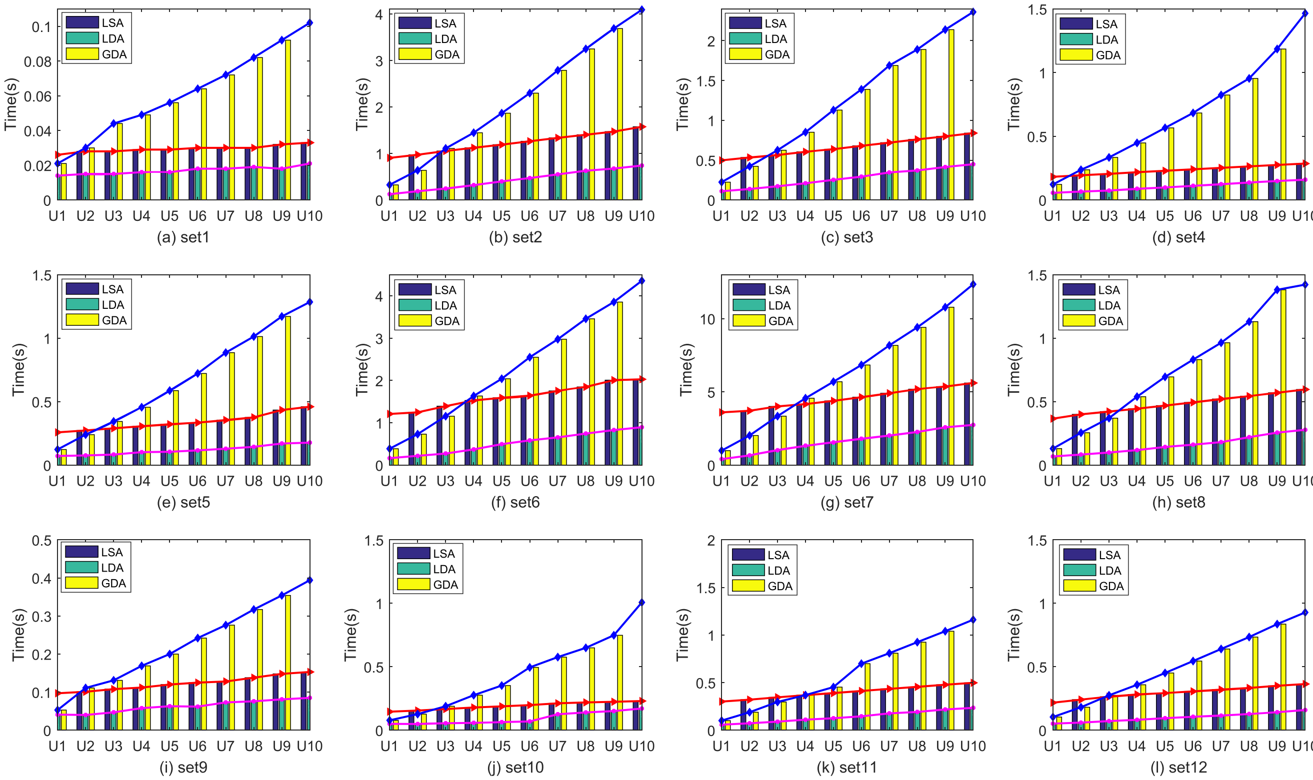

Fig. 2 The computational time of different algorithms with $\beta = 0.6$ when adding objects

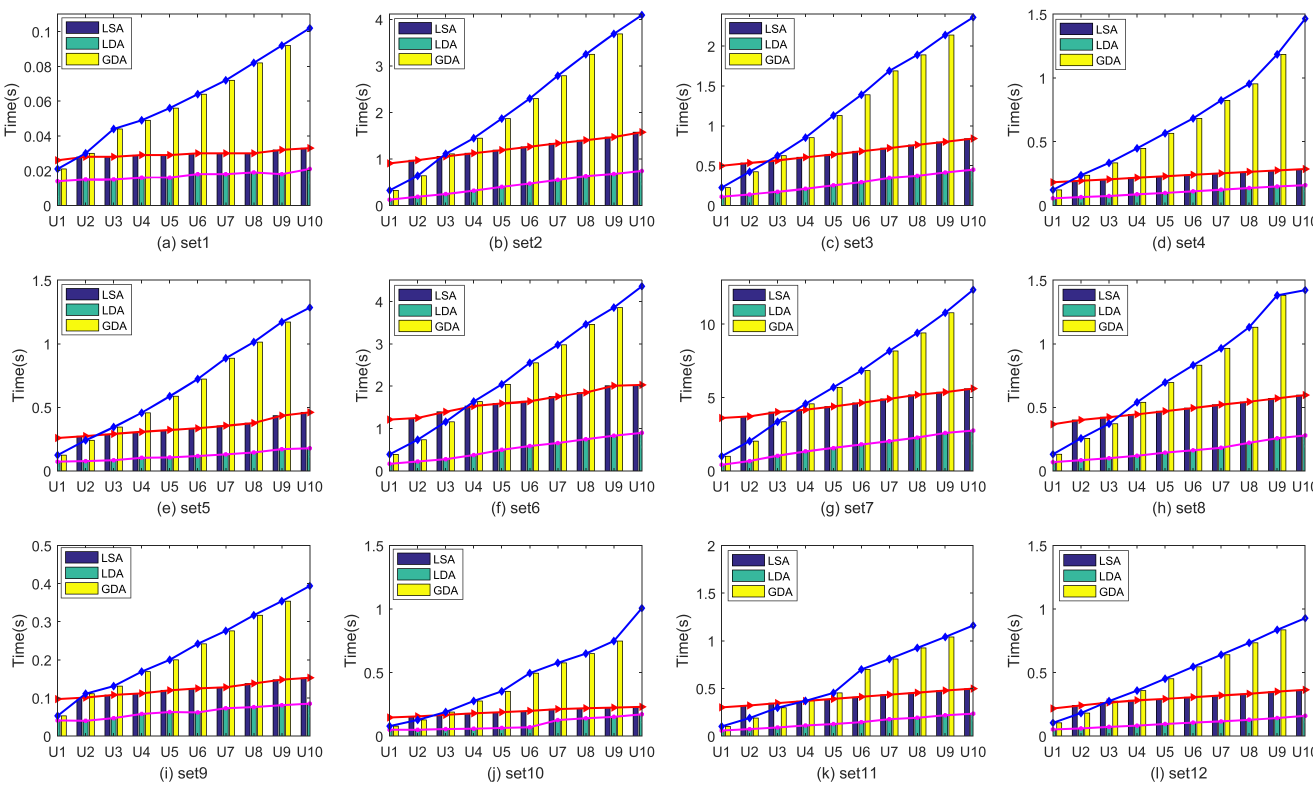

Fig. 3 The computational time of different algorithms with $\beta = 0.8$ when adding objects

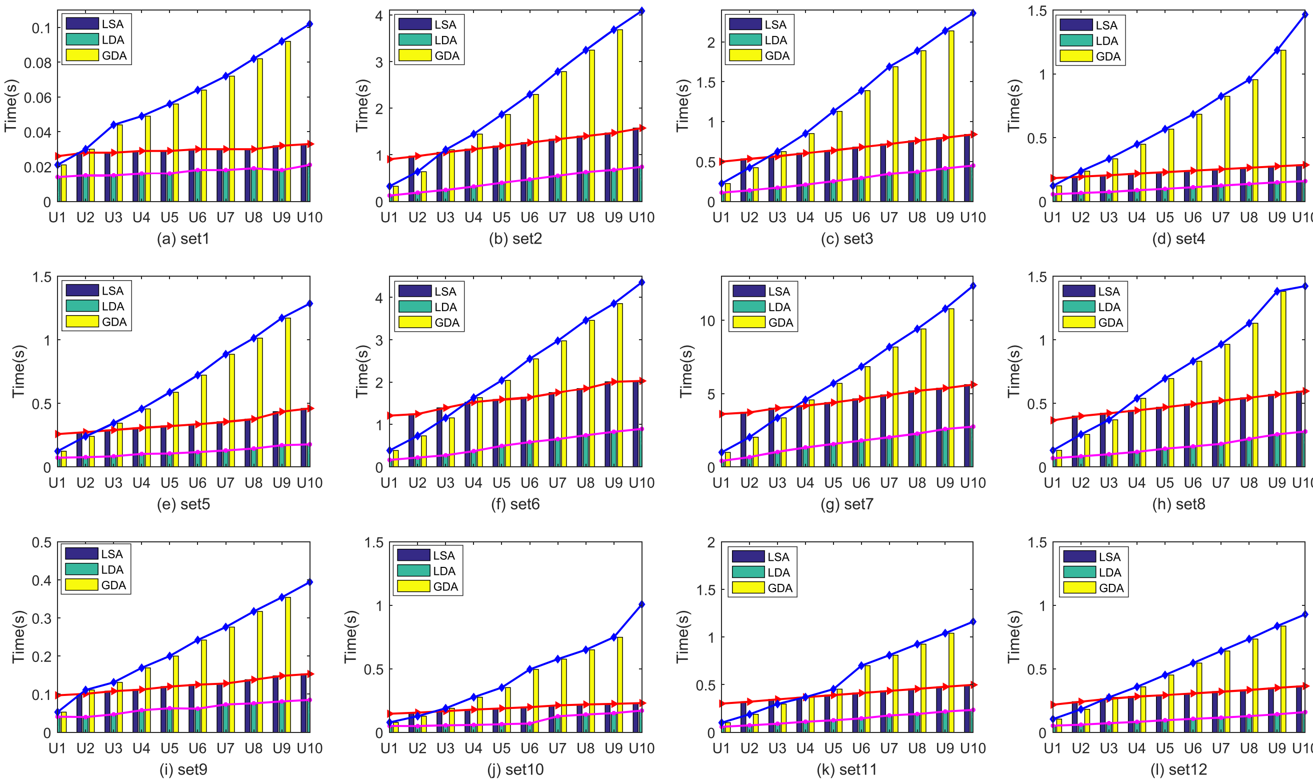

Fig. 4 The computational time of different algorithms with $\beta = 1$ when adding objects

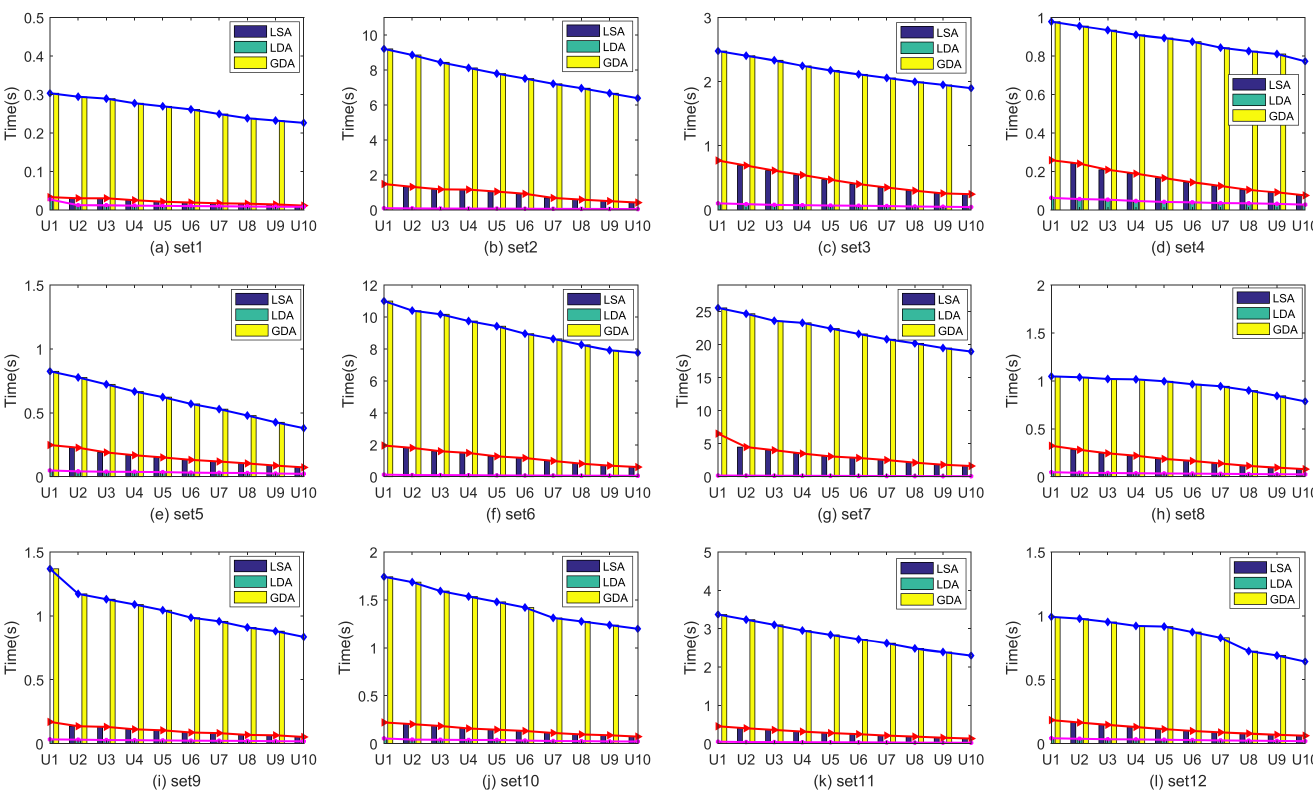

Fig. 5 The computational time of different algorithms with $\beta = 0.6$ when deleting objects

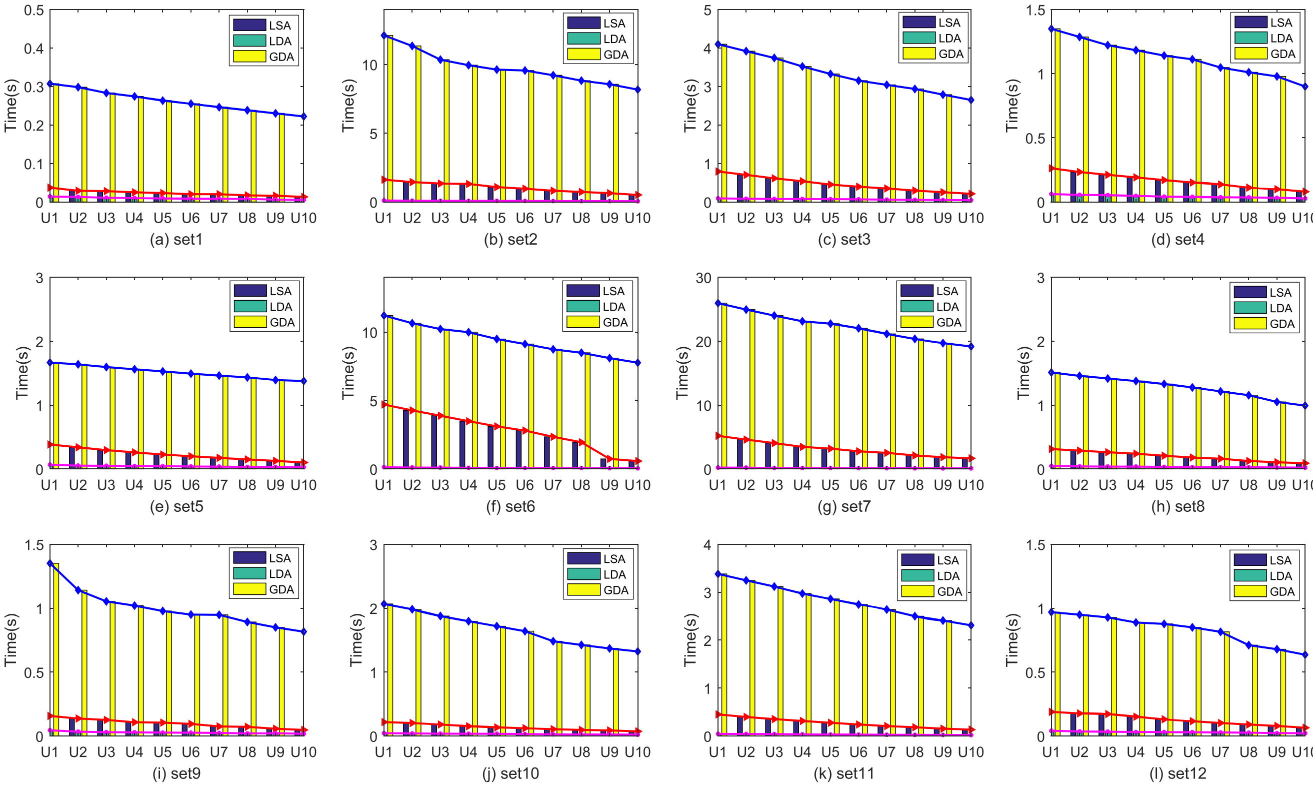

Fig. 6 The computational time of different algorithms with $\beta = 0.8$ when deleting objects

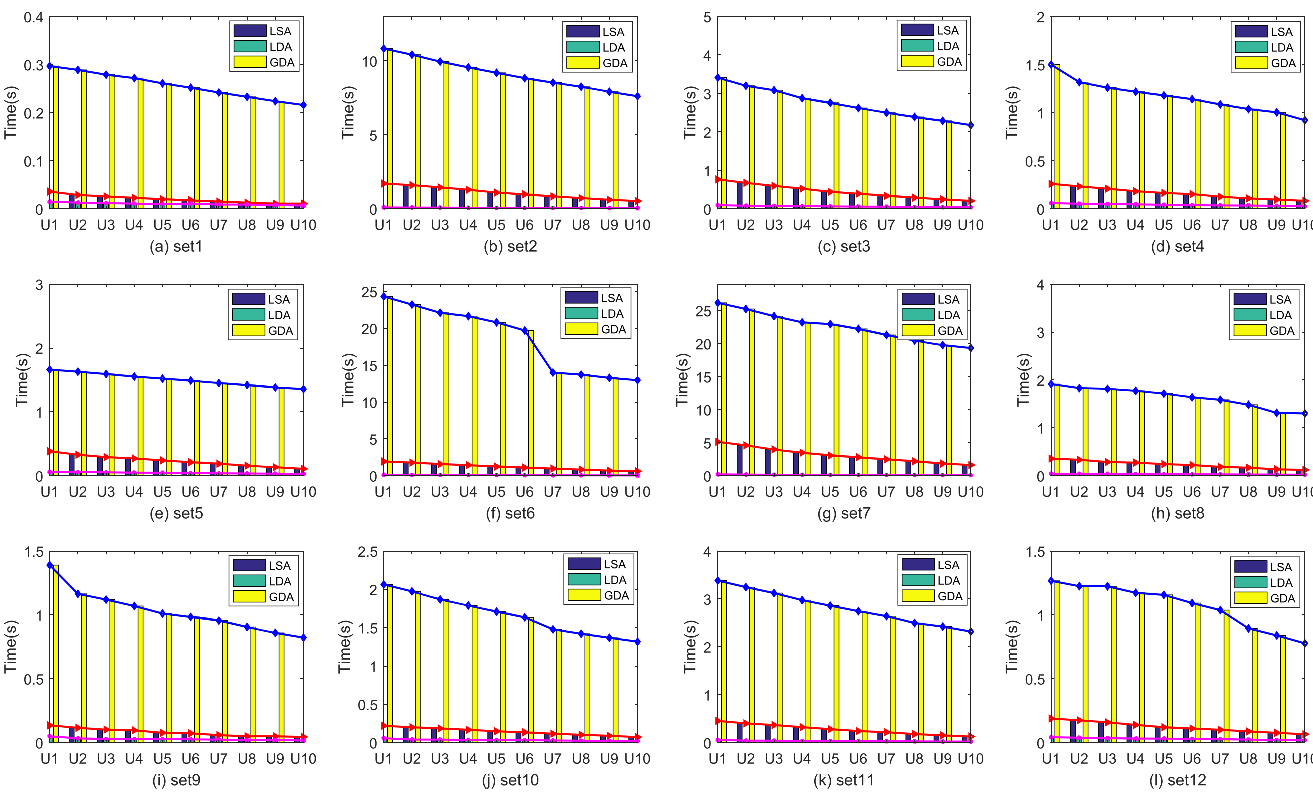

Fig. 7 The computational time of different algorithms with $\beta = 1$ when deleting objects

Table 9 The consume time of deleting objects by different approximate algorithms when $\beta = 0.6$

Dataset	Algorithm	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
	Local static	0.033	0.030	0.030	0.025	0.021	0.019	0.017	0.016	0.014	0.011
set1	Local dynamic	0.027	0.012	0.012	0.011	0.010	0.010	0.009	0.009	0.008	0.008
	Global dynamic	0.303	0.294	0.289	0.277	0.269	0.261	0.249	0.238	0.232	0.226
	Local static	1.476	1.313	1.170	1.157	1.042	0.916	0.680	0.584	0.498	0.414
set2	Local dynamic	0.094	0.084	0.074	0.071	0.071	0.063	0.056	0.054	0.049	0.043
	Global dynamic	9.202	8.853	8.435	8.114	7.791	7.506	7.207	6.954	6.666	6.383
	Local static	0.769	0.690	0.612	0.543	0.470	0.401	0.348	0.298	0.256	0.244
set3	Local dynamic	0.101	0.089	0.077	0.072	0.067	0.064	0.059	0.052	0.048	0.043
	Global dynamic	2.476	2.403	2.332	2.244	2.175	2.113	2.057	1.997	1.949	1.897
	Local static	0.258	0.240	0.208	0.188	0.165	0.143	0.124	0.104	0.091	0.075
set4	Local dynamic	0.062	0.056	0.053	0.046	0.041	0.039	0.035	0.034	0.031	0.027
	Global dynamic	0.978	0.955	0.934	0.910	0.893	0.874	0.843	0.825	0.810	0.772
	Local static	0.248	0.227	0.190	0.168	0.151	0.132	0.119	0.104	0.087	0.073
sest5	Local dynamic	0.049	0.042	0.039	0.038	0.036	0.032	0.030	0.029	0.025	0.023
	Global dynamic	0.825	0.777	0.723	0.667	0.623	0.570	0.529	0.479	0.426	0.380
	Local static	1.952	1.805	1.597	1.486	1.274	1.170	0.988	0.817	0.693	0.605
set6	Local dynamic	0.132	0.102	0.101	0.095	0.082	0.076	0.075	0.066	0.066	0.058
	Global dynamic	10.998	10.401	10.171	9.747	9.424	8.955	8.630	8.260	7.918	7.756
	Local static	6.517	4.485	4.011	3.503	3.074	2.839	2.522	2.128	1.814	1.622
set7	Local dynamic	0.178	0.164	0.153	0.141	0.134	0.119	0.108	0.099	0.090	0.081
	Global dynamic	25.545	24.657	23.594	23.286	22.428	21.606	20.801	20.182	19.472	18.941
	Local static	0.323	0.281	0.244	0.219	0.186	0.165	0.138	0.113	0.095	0.079
set8	Local dynamic	0.047	0.041	0.039	0.036	0.034	0.033	0.030	0.028	0.026	0.025
	Global dynamic	1.047	1.038	1.020	1.016	0.996	0.965	0.944	0.900	0.844	0.786
	Local static	0.170	0.136	0.131	0.112	0.104	0.087	0.082	0.068	0.065	0.052
set9	Local dynamic	0.033	0.031	0.028	0.027	0.025	0.023	0.022	0.021	0.019	0.018
	Global dynamic	1.368	1.172	1.130	1.089	1.044	0.986	0.957	0.909	0.880	0.835
	Local static	0.221	0.203	0.184	0.158	0.145	0.132	0.111	0.096	0.087	0.073
set10	Local dynamic	0.054	0.042	0.040	0.038	0.038	0.032	0.027	0.026	0.023	0.022
	Global dynamic	1.741	1.685	1.592	1.535	1.479	1.420	1.312	1.275	1.237	1.198
	Local static	0.452	0.400	0.357	0.314	0.276	0.246	0.209	0.183	0.154	0.132
set11	Local dynamic	0.050	0.041	0.041	0.037	0.035	0.034	0.031	0.026	0.025	0.022
	Global dynamic	3.368	3.234	3.098	2.950	2.836	2.719	2.621	2.484	2.390	2.296
	Local static	0.184	0.165	0.147	0.130	0.113	0.099	0.088	0.079	0.069	0.062
set12	Local dynamic	0.042	0.038	0.034	0.033	0.030	0.028	0.025	0.023	0.021	0.019
	Global dynamic	0.992	0.978	0.952	0.921	0.916	0.873	0.828	0.724	0.690	0.642

reduces some unnecessary computation about information classes, it could improve the efficiency for approximating target. To settle the approximate issue about multi-aspects numerical data in dynamic environment, we define the local generalized multigranulation neighborhood rough set model, and then design dynamic algorithms to obtain the updated approximations in this paper. Moreover, we have verified the effectiveness of dynamic algorithm through employing twelve data sets. The experimental results show that the proposed dynamic algorithm can significantly reduce the time consumption compared with the static algorithm, especially the local algorithm. It should be noted that

Table 10 The consume time of deleting objects by different approximate algorithms when $\beta = 0.8$

Dataset	Algorithm	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
	Local static	0.037	0.029	0.028	0.025	0.023	0.020	0.020	0.017	0.016	0.013
set1	Local dynamic	0.014	0.013	0.011	0.010	0.009	0.008	0.008	0.008	0.006	0.005
	Global dynamic	0.307	0.298	0.283	0.274	0.263	0.255	0.246	0.238	0.230	0.222
	Local static	1.615	1.443	1.337	1.305	1.075	0.959	0.812	0.727	0.635	0.509
set2	Local dynamic	0.104	0.089	0.082	0.078	0.076	0.069	0.062	0.059	0.059	0.051
	Global dynamic	12.109	11.341	10.344	9.942	9.621	9.557	9.210	8.818	8.556	8.167
	Local static	0.790	0.702	0.612	0.537	0.450	0.394	0.349	0.294	0.252	0.211
set3	Local dynamic	0.092	0.083	0.076	0.075	0.072	0.067	0.059	0.054	0.049	0.045
	Global dynamic	4.096	3.915	3.739	3.518	3.323	3.150	3.040	2.934	2.786	2.647
	Local static	0.262	0.233	0.211	0.191	0.170	0.151	0.136	0.111	0.099	0.080
set4	Local dynamic	0.061	0.055	0.053	0.047	0.041	0.040	0.036	0.035	0.032	0.027
	Global dynamic	1.349	1.285	1.221	1.182	1.140	1.111	1.047	1.010	0.978	0.899
	Local static	0.383	0.339	0.295	0.260	0.226	0.199	0.174	0.149	0.127	0.101
sest5	Local dynamic	0.066	0.052	0.050	0.046	0.043	0.039	0.036	0.033	0.033	0.028
	Global dynamic	1.665	1.637	1.594	1.560	1.526	1.491	1.461	1.432	1.391	1.376
	Local static	4.705	4.275	3.889	3.488	3.109	2.800	2.347	1.950	0.736	0.578
set6	Local dynamic	0.130	0.102	0.100	0.084	0.075	0.071	0.067	0.061	0.055	0.050
	Global dynamic	11.202	10.640	10.213	9.984	9.485	9.120	8.735	8.483	8.093	7.756
	Local static	5.183	4.592	4.050	3.458	3.184	2.765	2.524	2.113	1.847	1.662
set7	Local dynamic	0.215	0.203	0.155	0.141	0.132	0.122	0.113	0.104	0.097	0.097
	Global dynamic	25.941	24.932	23.991	23.100	22.728	22.009	21.126	20.346	19.689	19.158
	Local static	0.312	0.287	0.262	0.239	0.207	0.179	0.161	0.125	0.105	0.090
set8	Local dynamic	0.047	0.040	0.037	0.037	0.033	0.030	0.029	0.027	0.024	0.021
	Global dynamic	1.510	1.456	1.415	1.376	1.329	1.275	1.213	1.154	1.049	0.991
	Local static	0.157	0.137	0.126	0.108	0.105	0.095	0.075	0.072	0.057	0.048
set9	Local dynamic	0.045	0.034	0.030	0.029	0.027	0.026	0.024	0.021	0.022	0.018
	Global dynamic	1.351	1.141	1.053	1.020	0.978	0.950	0.948	0.892	0.850	0.816
	Local static	0.218	0.204	0.181	0.153	0.135	0.123	0.107	0.096	0.088	0.073
set10	Local dynamic	0.045	0.041	0.037	0.036	0.035	0.031	0.026	0.025	0.025	0.022
	Global dynamic	2.066	1.983	1.877	1.796	1.718	1.640	1.483	1.425	1.370	1.323
	Local static	0.451	0.398	0.353	0.315	0.277	0.241	0.208	0.183	0.154	0.134
set11	Local dynamic	0.044	0.043	0.041	0.035	0.033	0.031	0.027	0.025	0.023	0.021
	Global dynamic	3.381	3.247	3.118	2.971	2.860	2.743	2.639	2.500	2.409	2.309
	Local static	0.189	0.177	0.172	0.151	0.131	0.116	0.102	0.090	0.079	0.065
set12	Local dynamic	0.041	0.036	0.034	0.031	0.031	0.030	0.028	0.026	0.023	0.022
	Global dynamic	0.969	0.949	0.928	0.888	0.877	0.850	0.815	0.711	0.679	0.636

the attributes in the information system will also change (the number of attributes or the specific attribute value), thus how effective updated approximate space when attribute set changes is worth exploring. In addition, the neighbor radius is an important parameter related to approximate accuracy, so how to efficiently choose the optimal radius in the dynamic mechanism is also a problem that we need further research. Moreover, the approximate space is a basic tool for classification, thus the proposed dynamic mechanism provides a new idea for dynamic classification in dynamic environment.

Table 11 The consume time of deleting objects by different approximate algorithms when $\beta = 1$

Dataset	Algorithm	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
	Local static	0.036	0.029	0.026	0.023	0.020	0.018	0.015	0.013	0.011	0.011
set1	Local dynamic	0.015	0.013	0.012	0.011	0.010	0.011	0.009	0.009	0.008	0.007
	Global dynamic	0.297	0.289	0.279	0.272	0.261	0.252	0.242	0.233	0.224	0.216
	Local static	1.715	1.608	1.457	1.297	1.105	0.988	0.848	0.729	0.618	0.530
set2	Local dynamic	0.100	0.088	0.085	0.075	0.070	0.067	0.063	0.060	0.053	0.046
	Global dynamic	10.833	10.419	9.952	9.563	9.195	8.837	8.534	8.252	7.916	7.613
	Local static	0.769	0.676	0.599	0.527	0.444	0.398	0.339	0.293	0.248	0.208
set3	Local dynamic	0.097	0.083	0.079	0.071	0.065	0.059	0.057	0.049	0.044	0.043
	Global dynamic	3.411	3.200	3.085	2.878	2.758	2.622	2.499	2.390	2.289	2.177
	Local static	0.261	0.234	0.210	0.184	0.166	0.154	0.127	0.109	0.097	0.084
set4	Local dynamic	0.060	0.054	0.052	0.047	0.043	0.039	0.036	0.035	0.032	0.027
	Global dynamic	1.500	1.318	1.260	1.218	1.180	1.141	1.086	1.038	1.004	0.922
	Local static	0.384	0.329	0.292	0.271	0.241	0.212	0.189	0.158	0.134	0.109
sest5	Local dynamic	0.062	0.057	0.054	0.049	0.048	0.042	0.039	0.036	0.033	0.029
	Global dynamic	1.663	1.630	1.594	1.554	1.522	1.492	1.452	1.421	1.382	1.356
	Local static	1.944	1.786	1.590	1.448	1.242	1.113	0.978	0.847	0.703	0.613
set6	Local dynamic	0.136	0.120	0.104	0.094	0.085	0.078	0.069	0.069	0.057	0.052
	Global dynamic	24.324	23.224	22.111	21.655	20.816	19.710	14.007	13.729	13.289	12.973
	Local static	5.126	4.607	3.994	3.488	3.071	2.814	2.494	2.191	1.847	1.629
set7	Local dynamic	0.222	0.180	0.152	0.143	0.131	0.123	0.113	0.107	0.097	0.097
	Global dynamic	26.167	25.234	24.157	23.223	22.952	22.225	21.306	20.478	19.775	19.332
	Local static	0.359	0.334	0.287	0.274	0.244	0.224	0.186	0.169	0.134	0.123
set8	Local dynamic	0.043	0.042	0.042	0.036	0.034	0.031	0.030	0.027	0.025	0.022
	Global dynamic	1.915	1.829	1.812	1.772	1.714	1.638	1.585	1.481	1.312	1.301
	Local static	0.136	0.116	0.102	0.097	0.077	0.073	0.059	0.050	0.050	0.045
set9	Local dynamic	0.050	0.034	0.029	0.029	0.029	0.026	0.024	0.021	0.022	0.017
	Global dynamic	1.390	1.165	1.120	1.070	1.011	0.984	0.955	0.905	0.859	0.822
	Local static	0.220	0.201	0.186	0.168	0.149	0.135	0.117	0.103	0.090	0.073
set10	Local dynamic	0.058	0.041	0.040	0.037	0.036	0.033	0.029	0.026	0.022	0.020
	Global dynamic	2.064	1.974	1.869	1.790	1.710	1.635	1.478	1.420	1.368	1.317
	Local static	0.454	0.403	0.368	0.324	0.282	0.243	0.218	0.180	0.152	0.128
set11	Local dynamic	0.059	0.043	0.042	0.037	0.036	0.033	0.030	0.028	0.024	0.023
	Global dynamic	3.383	3.247	3.123	2.976	2.860	2.746	2.641	2.496	2.422	2.318
	Local static	0.189	0.175	0.159	0.139	0.121	0.111	0.101	0.088	0.077	0.066
set12	Local dynamic	0.043	0.039	0.035	0.032	0.032	0.030	0.027	0.024	0.021	0.020
	Global dynamic	1.267	1.225	1.224	1.173	1.157	1.093	1.038	0.894	0.839	0.777

Acknowledgements This paper is supported by the National Natural Science Foundation of China (Nos.61976245, 61472463, 61772002) and the Fundamental Research Funds for the Central Universities (No. XDJK2019B029).

References

1. Chen H, Li T, Cai Y et al (2016) Parallel attribute reduction in dominance-based neighborhood rough set. Inf Sci 373:351–368
2. Dua D, Graff C (2019) UCI Machine Learning Repository http://archive.ics.uci.edu/ml. Irvine, CA: University of California, School of Information and Computer Science
3. Fujita H, Gaeta A, Loia V et al (2020) Hypotheses analysis and assessment in counterterrorism activities: a method based on OWA and fuzzy probabilistic rough sets. IEEE Trans Fuzzy Syst 28(5):831–845
4. Fujita H, Gaeta A, Loia V et al (2019) Resilience analysis of critical infrastructures: a cognitive approach based on granular computing. IEEE Trans. Cybern. 49(5):1835–1848
5. Guo Y, Tsang ECC, Xu W, Chen D (2019) Local logical disjunction double-quantitative rough sets. Inf Sci 500:87–112

6. Jiang Z, Liu K, Yang X et al (2020) Accelerator for supervised neighborhood based attribute reduction. Int J Approx Reason 119:122–150
7. Ju H, Ding W, Yang X et al (2021) Robust supervised rough granular description model with the principle of justifiable granularity. Appl Soft Comput 110:107612
8. Kong Q, Zhang X, Xu W et al (2020) Attribute reducts of multi-granulation information system. Artif Intell Rev 53(2):1353–1371
9. Li W, Xu W, Zhang X et al (2021) Updating approximations with dynamic objects based on local multigranulation rough sets in ordered information systems. Artif Intell Rev, https://doi.org/10.1007/s10462-021-10053-9
10. Li W, Xue X, Xu W et al (2020) Double-quantitative variable consistency dominance-based rough set approach. Int J Approx Reason 124:1–16
11. Lin B, Zhang X (2019) Relative reduction of incomplete interval-valued decision information systems associated with evidence theory. J Inf Sci Eng 35(6):1377–1396
12. Liu K, Yang X, Fujita H et al (2019) An efficient selector for multi-granularity attribute reduction. Inf Sci 505:457–472
13. Luo S, Miao D, Zhang Z et al (2020) A neighborhood rough set model with nominal metric embedding. Inf Sci 520:373–388
14. Pawlak Z (1998) Rough set. Int J Comput Inf Sci 11(5):341–356
15. Qian Y, Liang X, Lin G et al (2017) Local multigranulation decision-theoretic rough sets. Int J Approx Reason 82:119–137
16. Qian Y, Liang X, Wang Q et al (2018) Local rough set: a solution to rough data analysis in big data. Int J Approx Reason 97:38–63
17. Qian Y, Liang J, Yao Y, et a. l. (2010) MGRS: A multi-granulation rough set. Inf Sci 180(6):949–970
18. Wang Q, Qian Y, Liang X et al (2018) Local neighborhood rough set. Knowl-Based Syst 153:53–64
19. Xu W, Guo Y (2016) Generalized multigranulation double-quantitative decision-theoretic rough set. Knowl-Based Syst 105:190–205
20. Xu W, Li W (2016) Granular computing approach to two-way learning based on formal concept analysis in fuzzy datasets. IEEE Trans Cybern 46(2):366–379
21. Xu W, Li W, Zhang X (2017) Generalized multigranulation rough sets and optimal granularity selection. GrC 2(4):271–288
22. Xu W, Sun W, Zhang X et al (2012) Multiple granulation rough set approach to ordered information systems. Int J Gen Syst 41(5):475–501
23. Xu W, Wang Q, Zhang X (2013) Multi-granulation rough sets based on tolerance relations. Soft Comput 17(7):1241–1252
24. Xu W, Yu J (2017) A novel approach to information fusion in multi-source datasets: a granular computing viewpoint. Inf Sci 378:410–423
25. Yang X, Li T, Liu D et al (2020) A multilevel neighborhood sequential decision approach of three-way granular computing. Inf Sci 538:119–141
26. Yang X, Li T, Liu D et al (2017) A unified framework of dynamic three-way probabilistic rough sets. Inf Sci 420:126–147
27. Yang W, Pawlak Z et al (1982) MGRS: A multi-granulation rough set. J Inform Technol 12(3):341–356
28. Yang L, Xu W, Zhang X et al (2020) Multi-granulation method for information fusion in multi-source decision information system. Int J Approx Reason 122:47–65
29. Yang L, Zhang X, Xu W et al (2019) Multi-granulation rough sets and uncertainty measurement for multi-source fuzzy information system. Int J Fuzzy Syst 21(6):1919–1937
30. Yu J, Chen M, Xu W (2017) Dynamic computing rough approximations approach to time-evolving information granule interval-valued ordered information system. Appl Soft Comput 60:18–29
31. Yuan K, Li W, Xu W et al (2021) A comparative experimental evaluation on performance of type-1 and interval type-2 Takagi-Sugeno fuzzy models. Int J Mach Learn Cyb 12:2135–2150
32. Zhang X, Gou H, Lv Z et al (2021) Double-quantitative distance measurement and classification learning based on the tri-level granular structure of neighborhood system. Knowl-Based Syst 217:106799
33. Zhang C, Li D, Zhai Y et al (2019) Multigranulation rough set model in hesitant fuzzy information systems and its application in person-job fit. Int J Mach Learn Cyb 10(4):717–729
34. Zhang X, Yao H, Lv Z et al (2021) Class-specific information measures and attribute reducts for hierarchy and systematicness. Inf Sci 563:196–225
35. Zhang J, Zhang X, Xu W (2018) Lower approximation reduction based on discernibility information tree in inconsistent ordered decision information systems. Symmetry 10(12):696
36. Zhang J, Zhang X, Xu W et al (2019) Local multigranulation decision-theoretic rough set in ordered information systems. Soft Comput 23(24):13247–13261
37. Zhou J, Lai Z, Miao D et al (2020) Multigranulation rough-fuzzy clustering based on shadowed sets. Inf Sci 507:553–573

Weihua Xu received the Ph.D. degree from School of Sciences, Xi'an Jiaotong University, Xi'an, China, in 2007, and the M.Sc. degree from School of Mathematics and Information Sciences, Guangxi University, Nanning, China, in 2004. He is currently a professor with the College of Artificial Intelligence, Southwest University, Chongqing, China. He serves in the editorial boards of several international journals, and has published 4 monographs and over 120 articles in international journals. His current research interests include granular computing, approximate reasoning, fuzzy sets, data mining, knowledge discovery, and cognitive computing.

Kehua Yuan received the B.Sc. degree from the School of Mathematics and Statistics, Ludong University, YanTai, China, in 2019. She is currently pursuing the M.Sc. degree from the school of mathematical and statistics, Southwest University, Chongqing, China. Her current research interests include target concept, attribute reduction and granular computing.

Wentao Li received the Ph.D. degree from the Department of Mathematics, Harbin Institute of Technology, Harbin, China, in 2019, and the M.Sc. degree from the School of Mathematics and Statistics, Chongqing University of Technology, Chongqing, China, in 2015. He is currently a lecture with the College of Artificial Intelligence, Southwest University, Chongqing, China. He has published over 20 articles in international journals. His current research interests include fuzzy sets, rough sets, and granular computing.