International Journal of Computational Intelligence Systems
Vol. **13(1)**, 2020, *pp.* 1473–1482
DOI: https://doi.org/10.2991/ijcis.d.200915.004; ISSN: 1875-6891; eISSN: 1875-6883
https://www.atlantis-press.com/journals/ijcis/

Research Article

Attribute Reduction of Boolean Matrix in Neighborhood Rough Set Model

Yan Gao, Changwei Lv, Zhengjiang Wu*,

College of Computer Science and Technology, Henan Polytechnic University, No. 2001 Century Avenue, Jiaozuo, Henan 454003, P.R. China

ARTICLE INFO

Article History
Received 02 Jan 2020
Accepted 02 Sep 2020

Keywords
Neighborhood rough set
Boolean matrix
Attribute reduction
GPU

ABSTRACT

Neighborhood rough set is a powerful tool to deal with continuous value information systems. Graphics processing unit (GPU) computing can efficiently accelerate the calculation of the attribute reduction and approximation sets based on matrix. In this paper, we rewrite neighborhood approximation sets in the matrix-based form. Based on the matrix-based neighborhood approximation sets, we propose the relative dependency degree of attributes and the corresponding algorithm (DBM). Furthermore, we design the reduction algorithm (ARNI) for continuous value information systems. Compared with other algorithms, ARNI can effectively remove redundant attributes, and less affect the classification accuracy. On the other hand, the experiment shows ARNI based on the matrixing rough set model can significantly speed up by GPU. The speedup is many times over the central processing unit implementation.

1. INTRODUCTION

The explosive growth of data volume increases the complexity of data, and makes data processing more difficult than before. By reducing unnecessary or irrelevant attributes in the data, the efficiency of decision-making can be improved. The rough set theory proposed by Pawlak can quickly deal with uncertain and inaccurate problems [1], which has been widely used in machine learning, data mining, and other fields [2]. To construct an equivalence relation of Pawlak rough set for continuous value information system, the continuous values need to be clustered into some mutually disjoint blocks. However, discretizing the continuous values exists some uncertainty and may lose some essential information. To solve this problem, many rough set models have been proposed, such as fuzzy rough sets [3–6], covering rough sets [7–9], semi-monolayer cover rough set [10], neighborhood rough sets [11–14], granule-based rough sets [15–17]. Neighborhood rough set is a feasible model to handle continuous values without discretization.

In neighborhood rough sets, the calculation, whether two elements are neighborhoods or not, becomes easier and more localized compared to other methods [11]. It offers better potential for parallel and distributed computation [18]. Furthermore, by marking the neighborhood of the element, we can label some data to improve the usability for lack-of-label big data [19]. However, the complexity of the calculation is an unavoidable question on neighborhood rough set [19,20].

Matrix description makes the calculation of approximation sets more efficient. Huang *et al.* [21] defined three composition operations, and studied their characteristic matrices, and investigated the relationship between the characteristic matrices and covering approximation operators. Wang *et al.* [22] represented three types of existing covering approximation operators with the Boolean matrix. Ma [23] selected more covering approximation model and rewrote them in a high level.

Graphics processing unit (GPU) acceleration is widely used not only in deep learning [24] but also in the operations of the dense matrices and blocks [25], such as minimizing the encountered in the transformation of tensor contractions into matrix multiplications [26], object sorting [27], computation of equivalence classes, and approximation sets [28]. Generally, GPU is at least 10 times faster than the coetaneous central processing unit (CPU) [29]. Zhang *et al.* [30] adopted a multi-GPU solution to accelerate their algorithm about a parallel method for computing approximations based on matrix and achieved 334.9 times of acceleration compared to the CPU.

The matrixing of the whole computation process is the premise of the speedup by GPU. For neighborhood rough approximation operators, the relation of disjunction and conjunction among Boolean matrices can represent the relation of union and intersection between sets. Through a sequence of substitution operations, the operation of the neighborhood rough set will be transformed from element-based form to matrix-based form. Furthermore, we proposed the relative dependency degree of attributes based on matrixing neighborhood approximation sets, and the corresponding reduction algorithms (DBM and ARNI).

* *Corresponding author. Email: wuzhengjiang@hpu.edu.cn*

This paper is organized as follows: In Section 2, some basic concepts about neighborhood rough sets are introduced. In Section 3, we propose attribute reduction algorithms DBM and ARNI based on the Boolean matrix. In Section 4, a series of comparative experiments are designed to show the feasibility and efficiency of ARNI. Section 5 is the conclusion of this paper.

2. PRELIMINARIES

In information systems, the elements can easily cluster into some information granules by different methods. How to describe them systematically in a framework is a basic question about the construction of approximation space in rough set theory. The binary relation and neighborhood system are two common entry points. Binary relation can summarize the principle of generating information granules. Covering or neighborhood systems directly organize the related elements into a block or a neighborhood. In classic Pawlak rough set, the equivalence relation and equivalence classes can transform mutually. However, in other models, the transformation is no longer smooth [31]. The choice binary relation or neighborhood system depends almost entirely on the character in information systems. For example, in set-valued or incomplete information systems, we can use not only tolerance relation and its generalized tolerance relation [18] but also maximal consistent blocks [11] or semi-monolayer covering [10] to organize the blocks on the system. For the information system with noise, the variable precision binary relation [32] and probabilistic binary relation [33] are frequently mentioned. For continuous information systems, we can build a hypersphere neighborhood to describe a similar degree among the elements, as shown in Figure 1.

In this section, we introduce the basic concepts about neighborhood rough set and two Boolean matrix operations.

2.1. Neighborhood Rough Set

In information systems, let $U = \{x_1, x_2, x_3, \cdots, x_n\}$ be a nonempty and finite set, $A = \{a_1, a_2, a_3, \cdots, a_m\}$ be the finite condition-attribute set of U, $d = \{d_1, d_2, d_3, \cdots, d_k\}$ be the decision attribute set of U. If a family of neighborhood relations can be generated by condition-attribute set A, then, $NS = (U, A)$ is called a neighborhood system, and $NDS = (U, A, d)$ is called a neighborhood decision system.

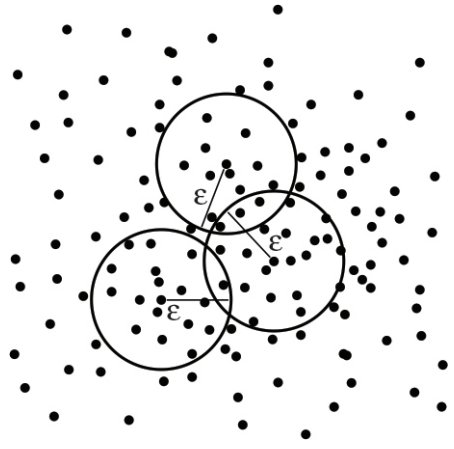

Figure 1 Neighborhood of the elements ε.

Definition 1. [12,13] In the neighborhood system $NS = (U, A)$, $\forall x_i \in U$, the neighborhood of x_i is defined as

$$\varepsilon(x_i) = \left\{x_j | x_j \in U, \delta_p(x_i, x_j) \leq \varepsilon\right\} \tag{1}$$

where ε is the radius of the hypersphere neighborhood, δ_p is the Minkowsky distance between x_i and x_j, (1) it is called Manhattan distance if $p = 1$, (2) Euclidean distance if $p = 2$, (3) Chebyshev distance if $p = \infty$. The Euclidean distance is selected in this paper.

Hu proposed the definition of the neighborhood rough sets [13,14], which is a specialized covering rough model in neighborhood approximation space (Definition 2).

Definition 2. [11–14,34] In the neighborhood system $NS = (U, A)$, for any $X \subseteq U$, the upper approximation $\overline{N}(X)$, the lower approximation $\underline{N}(X)$, and the boundary region $BN(X)$ of the neighborhood of X are defined as follows, respectively:

$$\overline{N}(X) = \{x_i | \varepsilon(x_i) \cap X \neq \varnothing\} \tag{2}$$

$$\underline{N}(X) = \{x_i | \varepsilon(x_i) \subseteq X\} \tag{3}$$

$$BN(X) = \overline{N}(X) - \underline{N}(X) \tag{4}$$

Example 1. Given a neighborhood decision system $NDS = (U, A, d)$ at Table 1, where d is the decision attribute, $a_i \in A$ is the condition attributes. We suppose that $\varepsilon = 0.5$ and the neighborhood calculation formula of x under A is

$$\delta_p(x_i, x_j) = \sqrt{\sum_{k=1}^{4} \left[v(x_i, a_k) - v(x_j, a_k)\right]^2},$$

$$\varepsilon(x_i) = \left\{x_j | x_j \in U, \delta_p(x_i, x_j) \leq 0.5\right\}$$

where, $i, j = 1, 2, 3, 4, 5$, then

The values of neighborhoods are

$$\varepsilon(x_1) = \{x_1\}, \varepsilon(x_2) = \{x_2, x_3\},$$

$$\varepsilon(x_3) = \{x_2, x_3\}, (x_4) = \{x_4\}, \varepsilon(x_5) = \{x_5\}$$

By (2) and (3), the upper and lower approximation sets of the neighborhood rough set are as follows:

$$\overline{N}(d_1) = \{x_i | \varepsilon(x_i) \cap d_1 \neq \varnothing\} = \{x_1, x_2, x_3\},$$

Table 1 Decision information system.

U	a_1	a_2	a_3	a_4	d
x_1	0.9	0.6	1.4	1.3	0
x_2	0.5	1.0	1.5	1.1	0
x_3	0.6	1.3	1.4	1.0	1
x_4	1.0	1.2	1.7	1.5	1
x_5	1.1	0.9	2.5	1.4	1

$$\overline{N}(d_2) = \{x_2, x_3, x_4, x_5\}$$

and

$$\underline{N}(d_1) = \{x_i | \varepsilon(x_i) \subseteq d_1\} = \{x_1\},$$

$$\underline{N}(d_2) = \{x_4, x_5\}$$

2.2. Boolean Matrix Operation

Boolean matrix has been used to describe the classic binary relation or neighborhood system frequently. To matrix the upper and lower neighborhood approximation operations, we need to upgrade some operations based on traditional disjunction and conjunction to calculate the union and intersection of two sets.

Definition 3. [23,34] Let $\odot$ be an invariant operation between two Boolean matrices, $\boldsymbol{A_{n\times m}} = (a_{ik})_{n\times m}$ and $\boldsymbol{B_{m\times l}} = \left(b_{kj}\right)_{m\times l}$ be two Boolean matrices. Then, the calculation of $\boldsymbol{C_{n\times l}} = \left(c_{ij}\right)_{n\times l} = A \odot B$ is defined as follows:

$$c_{ij} = \vee_{k=1}^{m} \left(a_{ik} \wedge b_{kj}\right) \tag{5}$$

where $i = 1, 2, 3, \cdots, n$, $j = 1, 2, 3, \cdots, l$; $\vee$ and $\wedge$ are Boolean logic operations, which represent the disjunctive and conjunctive, respectively.

Definition 4. [23,34] Let $\otimes$ be a matrix transformational operations between two Boolean matrices, $\boldsymbol{A}_{n\times m} = (a_{ik})_{n\times m}$ and $\boldsymbol{B}_{m\times l} = \left(b_{kj}\right)_{m\times l}$ be two Boolean matrices. Then, the calculation of $\boldsymbol{D}_{n\times l} = \left(d_{ij}\right)_{n\times l} = \boldsymbol{A} \otimes \boldsymbol{B}$ is defined as follows:

$$d_{ij} = \wedge_{k=1}^{m} \left[(1 - a_{ik}) \vee b_{kj}\right] \tag{6}$$

where $i = 1, 2, 3, \cdots, n$, $j = 1, 2, 3, \cdots, l$.

3. ATTRIBUTE REDUCTION BASED ON BOOLEAN MATRIX

In this section, Boolean matrix operations ($\odot$ and $\otimes$) have been constructed to matrix the set operations (union and intersection) in neighborhood rough approximation operations. Furthermore, we define a dependency degree in the neighborhood decision system and propose a new attribute reduction. The algorithm can keep the same consistency of matrix-based calculation and element-based calculation in neighborhood approximation operations.

3.1. Boolean Matrix Representation of Approximations

In this section, we rewrite the neighborhood approximation operation by the Boolean matrix and its matrix operations.

Definition 5. [34] In the neighborhood system $NS = (U, A)$, $\forall B \subseteq A$, $\forall x_i, x_j \in U$. Then, the Boolean matrix $\boldsymbol{M}_B = \left(m_{ij}\right)_{n\times n}$ of the neighbor N_B corresponding to B is defined as

$$m_{ij} = \begin{cases} 1, x_j \in \varepsilon(x_i) \\ 0, x_j \notin \varepsilon(x_i) \end{cases} \tag{7}$$

where, $i, j = 1, 2, 3, \cdots, n$.

Obviously, this definition indicates that $\boldsymbol{M}_B$ is a Boolean matrix, N_B is the neighborhood cluster of U with respect to condition-attribute set B, (U, N_B) is the neighborhood approximation space, $\boldsymbol{M}_B$ is its Boolean matrix representation, and $M(U)$ is a set of Boolean matrices on U.

For any $X \subseteq U = \{x_1, x_2, x_3, \cdots, x_n\}$, the eigenvector $\boldsymbol{\lambda}_X = (a_1, a_2, a_3, \cdots, a_n)^{\mathrm{T}}$ [24,25,34] of X includes only 1 and 0. For $\forall x_i \in U$, if $x_i \in X$, $a_i = 1$; otherwise $a_i = 0$. For example, suppose $U = \{x_1, x_2, x_3, x_4\}$, and $X = \{x_2, x_4\}$, then $\boldsymbol{\lambda}_X = (0\,1\,0\,1)^{\mathrm{T}}$. $\boldsymbol{\lambda}_X$ is a Boolean vector, the two operations ($\odot$ and $\otimes$) are also applicable to the operation between Boolean matrix and Boolean vector.

Definition 6. [22,23] In the approximation space (U, N), $\boldsymbol{M}_B$ is its Boolean matrix representation. The relation matrix of the approximation space is defined as $\boldsymbol{R}_B = (r_{ik})_{n\times n}$

$$\boldsymbol{R}_B = \boldsymbol{M}_B \odot \boldsymbol{M}_B^T \tag{8}$$

where, $\boldsymbol{R}_B$ represents the invariance of the operation between the Boolean matrices and its transpose.

Proposition 1. *In the neighborhood approximation space (U, N), $\boldsymbol{R}_B$ is its relation matrix, $U/d = \{d_1, d_2, d_3, \cdots, d_k\}$. For $\forall B \subseteq A$, $\boldsymbol{\lambda}_{d_i}$ is the eigenvectors of d_i, the upper and lower approximations of d_1 of B corresponds to the eigenvectors are defined as follows, respectively:*

$$\boldsymbol{\lambda}_{\overline{N}(d_i)} = \boldsymbol{R}_B \odot \boldsymbol{\lambda}_{d_i} \tag{9}$$

$$\boldsymbol{\lambda}_{\underline{N}(d_i)} = \boldsymbol{R}_B \otimes \boldsymbol{\lambda}_{d_i} \tag{10}$$

where, $i = 1, 2, 3, \cdots, k$.

Proof ($\Rightarrow$): $\forall t \in \{1, 2, 3, \cdots, s\}$, if $x_t \in \overline{N}(d_i)$, also known as $a_t = 1$, then $\varepsilon(x_t) \cap d_i \neq \varnothing$, and $\exists x_l \in d_i$, $x_l \in \varepsilon(x_t)$. Thus, $r_{tl} = 1$ and $a_t = 1$, and $\vee_{t=1}^{s} (r_{tl} \wedge a_t) = 1$. Hence, $\forall t \in \{1, 2, 3, \cdots, n\}$, $r_i \rightarrow \vee_{k=1}^{m} (r_{ik} \wedge a_k)$.

($\Leftarrow$): $\forall t \in \{1, 2, 3, \cdots, s\}$, if $\vee_{t=1}^{s} (r_{tl} \wedge a_t) = 1$, in other words $\exists l \in \{1, 2, 3, \cdots, s\}$, $r_{tl} = 1$ and $a_t = 1$. Thus, $\exists x_l \in \varepsilon(x_t)$ and $x_l \in d_i$, $\varepsilon(x_t) \cap d_i \neq \varnothing$, then $x_t \in \overline{N}(d_i)$ and $a_t = 1$. Hence, $\forall t \in \{1, 2, 3, \cdots, n\}$, $\vee_{k=1}^{m} (r_{ik} \wedge a_k) \rightarrow r_i$.

Thus, $\forall t \in \{1, 2, 3, \cdots, s\}$, $r_i = \vee_{t=1}^{s} \left(r_{tl} \wedge a_{x_t}\right) = 1$. namely, $\boldsymbol{\lambda}_{\overline{N}(d_i)} = \boldsymbol{R}_B \odot \boldsymbol{\lambda}_{d_i}$, the proof of formula (9) is completed.

The proof of formula (10) is similar to that of formula (9), then proposition 1 is held.

Definition 7. [34] In the neighborhood approximation space (U, N), $U/d = \{d_1, d_2, d_3, \cdots, d_k\}$, $\boldsymbol{\lambda}_{\overline{N}(d_i)}$ and $\boldsymbol{\lambda}_{\underline{N}(d_i)}$ are the eigenvectors corresponding to the upper and lower approximations of d_i.

Then, upper approximation $\overline{N}(d_i)$ and lower approximation $\underline{N}(d_i)$ of d_i are defined as follows, respectively:

$$\overline{N}(d_i) = \cup_{j=1}^{k} \left\{x_j | a_j = 1\right\} \tag{11}$$

$$\overline{N}(d_i) = \cup_{j=1}^{k} \left\{x_j | a_j = 1, a_j \in \lambda_{\overline{N}(d_i)}\right\} \tag{12}$$

Definition 8. In the neighborhood approximation space (U, N), $U/d = \{d_1, d_2, d_3, \cdots, d_k\}$. $\lambda_{\overline{N}(d_i)}$ and $\lambda_{\underline{N}(d_i)}$ are the eigenvectors corresponding to the upper and lower approximations of d_i. Then, $\lambda_{\overline{N}(d)}$ and $\lambda_{\underline{N}(d)}$ of d are defined as follows, respectively:

$$\underline{N}(d_i) = \cup_{j=1}^{k} \left\{x_j | a_j = 1, a_j \in \lambda_{\underline{N}(d_i)}\right\} \tag{13}$$

$$\lambda_{\underline{N}(d)} = \cup_{i=1}^{k} \lambda_{\underline{N}(d_i)} \tag{14}$$

Suppose that eigenvectors

$$\lambda_X = (a_1, a_2, \ldots, a_n)^{\mathrm{T}}, \lambda_Y = (b_1, b_2, \ldots, b_n)^{\mathrm{T}}$$

then

$$\lambda_X \cup \lambda_Y = (\max(a_1, b_1), \max(a_2, b_2), \ldots, \max(a_n, b_n))$$

So far, we have accomplished the matrix description of the neighborhood rough set.

Example 2 (Continuing Example 1). Let $NDS = (U, A, d)$ be a neighborhood decision system the same as that of **Example 1**. The Boolean matrix method is used to calculate the upper and lower approximation sets of the neighborhood rough set. Thus, the corresponding Boolean matrix of the neighborhood is

$$\boldsymbol{M}_B = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\lambda_{d_1} = (1\,1\,0\,0\,0)^{\mathrm{T}},$$

$$\lambda_{d_2} = (0\,0\,1\,1\,1)^{\mathrm{T}}$$

The relation matrix $\boldsymbol{R}_B$ is

$$\boldsymbol{R}_B = \boldsymbol{M}_B \odot \boldsymbol{M}_B^T = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

The upper and lower approximation sets of d_1 corresponding eigenvectors are

$$\lambda_{\overline{N}(d_1)} = \boldsymbol{R}_B \odot \lambda_{d_1} = (1\,1\,1\,0\,0)^{\mathrm{T}},$$

$$\lambda_{\underline{N}(d_1)} = \boldsymbol{R}_B \otimes \lambda_{d_1} = (1\,0\,0\,0\,0)^{\mathrm{T}};$$

the upper and lower approximation sets of d_1:

$$\overline{N}(d_1) = \{x_1, x_2, x_3\}, \underline{N}(d_1) = \{x_1\}.$$

Similarly,

$$\lambda_{\overline{N}(d_2)} = (0\,1\,1\,1\,1)^{\mathrm{T}},$$

$$\lambda_{\underline{N}(d_2)} = (0\,0\,0\,1\,1)^{\mathrm{T}};$$

$$\overline{N}(d_2) = \{x_2, x_3, x_4, x_5\} \text{ and } \underline{N}(d_2) = \{x_4, x_5\}.$$

By the matrixing method, we still get the same result as Example 1.

3.2. Reduction of Neighborhood Rough Set

Dependency and importance degree are widely used in attribute reduction. For neighborhood rough set, we specified the two indexes based on the approximation sets.

Definition 9. In neighborhood approximation space (U, N), let A be a condition-attribute set on U, $U/d = \{d_1, d_2, d_3, \cdots, d_k\}$. For $\forall B \subseteq A$, the dependency degree $\gamma_B(d)$ of d with respect to B is defined as follows:

$$\gamma_B(d) = \frac{S\left(\lambda_{\underline{N}(d)}\right)}{|U|} \tag{15}$$

where, $S\left(\lambda_{\underline{N}(d)}\right)$ are the sum of all the elements in the eigenvector $\lambda_{\underline{N}(d)}$, it represents the number of elements in the lower approximation of d.

Unless particularly stated, the relative dependency degree is the dependency degree of the decision-making d with respect to all attributes A. Obviously, $\gamma_B(d) \in [0, 1]$, and the greater of the value, the more elements identified in U, the stronger classification ability of the attribute subset B.

Definition 10. [34,35] In the neighborhood decision system $NDS = (U, A, d)$, $\forall B \subseteq A$, $\forall a \in A$, then the neighborhood importance degree $\sigma_B(a)$ or $\sigma'_B(a)$ of a relative to B is

(a) If $a \in B$,

$$\sigma_B(a) = \gamma_B(d) - \gamma_{B-\{a\}}(d) \tag{16}$$

(b) If $a \in A - B$,

$$\sigma'_B(a) = \gamma_{B\cup\{a\}}(d) - \gamma_B(d) \tag{17}$$

Definition 11. In the neighborhood decision system $NDS = (U, A, d)$, $\forall C \subseteq A$, and $\forall a \in C$, if $\gamma_C(d) = \gamma_A(d)$ and $\sigma_C(a) > 0$, then we call C an attribute reduction of NDS.

Example 3 (Continuing Example 2). Let $NDS = (U, A, d)$ be a neighborhood decision system the same as Example 1, and the upper and lower approximations of the neighborhood rough set, respectively $\overline{N}(d_1) = \{x_1, x_2, x_3\}$, $\underline{N}(d_1) = \{x_1\}$, $\overline{N}(d_2) = \{x_2, x_3, x_4, x_5\}$, $\underline{N}(d_2) = \{x_4, x_5\}$.

The eigenvector corresponding to the upper and lower approximations of d are as follows:

$$\boldsymbol{\lambda}_{\overline{N}(d)} = \cup_{i=1}^{2} \boldsymbol{\lambda}_{\overline{N}(d_i)} = (1\,1\,1\,1\,1)^{\mathrm{T}},$$

$$\boldsymbol{\lambda}_{\underline{N}(d)} = \cup_{i=1}^{2} \boldsymbol{\lambda}_{\underline{N}(d_i)} = (1\,0\,0\,1\,1)^{\mathrm{T}}.$$

Thus, the dependency degree of d with respect to A is

$$\gamma_A(d) = \frac{S\left(\lambda_{\underline{N}(d)}\right)}{|U|} = \frac{3}{5}.$$

Similarly, the dependency degrees of other condition attributes are shown in Table 2. Because $\sigma_A(a_4) = 0$, a_4 is redundant, the lower approximation set of the data set doesn't change.

3.3. Attribute Reduction Algorithm Based on Boolean Matrix

Based on the previous propositions and definitions, we design two algorithms Dependency Degree based on Boolean Matrix (DBM) and Attribute Reduction based on Neighborhood Importance Degree (ARNI).

Algorithm DBM is used to calculate the dependency degree of the condition attribute with respect to decision attributes in the data set. Suppose that there are n samples, m condition attributes, and k classes in the data set. In line 2 of DBM, we initialize a matrix whose size is $n \times n$. The corresponding time complexity is $O\left(n^2\right)$. According to Definition 5, the complexity of lines 3–9 is $O\left(n^2\right)$. In line 10 of DBM, we initialize a vector whose size is $n \times 1$, and its time complexity is $O(n)$. In lines 11–15, we compute the $\gamma_A(d)$ of d with respect to A based on Definitions 7–9, the time complexity of these lines is $O\left(n^2k\right)$. Thus, the overall time complexity of the algorithm DBM is $O\left(n^2k\right)$. Algorithm DBM is an essential part of algorithm ARNI.

ARNI calls DBM to calculate the neighborhood importance degree of a relative to A at line 4. Its time complexity is $O\left(n^2k\right)$. The time complexity of step in line 9 to line 20 is $0.5m(m-1)$. Hence, the time complexity of algorithm ARNI is $O\left(n^2m^2k\right)$.

4. EXPERIMENTS

In the experiment, we use Algorithm DBM to calculate the relative dependency degree for every condition attribute. In Algorithm ARNI, the redundant attributes will be reduced from the continuous information systems. To verify the feasibility and effectiveness of ARNI, we design a series of comparative experiments. Specifically, we divide the experiments into two parts. One part is the reduction ability of the algorithm ARNI. In another part, we test the acceleration performance of GPU for the matrixing models.

4.1. Experiment Environment and Objects

The experimental data in this paper come from the UCI machine learning database [36], which is listed in Table 3. All values in those datasets will be normalized before we used.

Algorithms DBM and ARNI are programmed in Python 3.6.7. The experiments run on a computer with Windows 7 Ultimate, Intel i5-5200U CPU 2.2GHz, and 4GB memory, NVIDIA 820M.

4.2. Comparative Experiment between ARNI and Other Reduction Algorithms

To demonstrate the effectiveness of ARNI, we compare our algorithm with NRS, CRS, and PCA in terms of the classification

Algorithm 1: DBM

Input: $NDS = (U, A, d)$

Output: $\gamma_A(d)$ of d with respect to A

1 **begin**
2 Initialize $m_{ij} = (0)_{n\times n}$
3 **foreach** $i \in \{1, 2, 3, \ldots, n\}$ **do**
4 **foreach** $j \in \{1, 2, 3, \ldots, n\}$ **do**
5 **if** $\delta(x_i, x_j) \le \varepsilon$ **then**
6 $m_{ij} = 1$
7 **end**
8 **end**
9 **end**
10 Initialize $\boldsymbol{\lambda}_{\underline{N}(d)} = (0)_{n\times 1}$
11 **foreach** $i \in \{1, 2, 3, \ldots, k\}$ **do**
12 $\boldsymbol{\lambda}_{\underline{N}(d_i)} = \boldsymbol{M}_A \odot \boldsymbol{M}_A^T \otimes \boldsymbol{\lambda}_{d_i}$
13 $\boldsymbol{\lambda}_{\underline{N}(d)} = \boldsymbol{\lambda}_{\underline{N}(d)} \cup \boldsymbol{\lambda}_{\underline{N}(d_i)}$
14 **end**
15 $\gamma_A(d) = \frac{S(\boldsymbol{\lambda}_{\underline{N}(d)})}{|U|}$
16 **return** $\gamma_A(d)$
17 **end**

Table 2 The result of the subset of condition-attribute sets.

	$\boldsymbol{\lambda}_{\overline{N}(d)}$	$\boldsymbol{\lambda}_{\underline{N}(d)}$	$\overline{N}(d)$	$\underline{N}(d)$	$\gamma_B(d)$	$\sigma_A(a)$
A	$(1\,1\,1\,1\,1)^{\mathrm{T}}$	$(1\,0\,0\,1\,1)^{\mathrm{T}}$	$\{x_1, x_2, x_3, x_4, x_5\}$	$\{x_1, x_4, x_5\}$	$\frac{3}{5}$	/
$A - \{a_1\}$	$(1\,1\,1\,1\,1)^{\mathrm{T}}$	$(1\,0\,0\,0\,1)^{\mathrm{T}}$	$\{x_1, x_2, x_3, x_4, x_5\}$	$\{x_1, x_5\}$	$\frac{2}{5}$	$\frac{1}{5}$
$A - \{a_2\}$	$(1\,1\,1\,1\,1)^{\mathrm{T}}$	$(0\,0\,0\,0\,1)^{\mathrm{T}}$	$\{x_1, x_2, x_3, x_4, x_5\}$	$\{x_5\}$	$\frac{1}{5}$	$\frac{2}{5}$
$A - \{a_3\}$	$(1\,1\,1\,1\,1)^{\mathrm{T}}$	$(0\,0\,0\,1\,0)^{\mathrm{T}}$	$\{x_1, x_2, x_3, x_4, x_5\}$	$\{x_4\}$	$\frac{1}{5}$	$\frac{2}{5}$
$A - \{a_4\}$	$(1\,1\,1\,1\,1)^{\mathrm{T}}$	$(1\,0\,0\,1\,1)^{\mathrm{T}}$	$\{x_1, x_2, x_3, x_4, x_5\}$	$\{x_1, x_4, x_5\}$	$\frac{3}{5}$	0

Algorithm 2: ARNI

Input: $NDS = (U, A, d)$
Output: The attribute reduction C of NDS

```
begin
    Initialize C = ∅
    foreach a ∈ A do
        σ(A, a) = γ_A(d) − γ_{A−a}(d)
        if σ(A, a) > 0 then
            C = C ∪ a
            P = A − a
        end
        while P ≠ ∅ do
            list ≠ ∅
            foreach a ∈ P do
                if σ'(C, a) = 0 then
                    P = P − a
                else
                    list = list ∪ σ'(C, a)
                end
                C = C ∪ a_index
                P = P − a_index
            end
        end
    end
    return C
end
```

Table 3 Description of data sets.

Data Sets	Logogram	Samples	Attributes	Classes
biodeg	BI	1055	41	2
debrecen	DR	1151	19	2
ForestTypes	FT	326	27	4
glass	GL	214	10	6
ionosphere	IP	351	33	2
iris	IR	150	4	3
movement_libras	ML	360	90	15
sonar	SN	208	60	2
wdbc	WD	569	30	2
wine	WI	178	13	9

performance of K-Nearest Neighbor, K = 3 (KNN) and Support Vector Machine (SVM).

NRS is another classic reduction algorithm in neighborhood rough set theory [12,13]. Table 4 shows the number of features after reduction by ARNI and NRS for the different radius of the hypersphere neighborhood ε. In general, the reduction performance of ARNI and NRS appears some differences. As shown in Table 4, in 19 of 30 experiments, the number of reduced attributes based on ARNI is no less than the number of NRS. The classification accuracy of KNN (SVM) based ARNI reduction is superior to the one based on NRS in 25 (24) of 30 experiments. Specifically, when $\varepsilon = 0.01$, although the number of reduction attribute subsets of algorithm ARNI is better than the NRS, its classification accuracy of classifiers is higher than NRS generally. When $\varepsilon = 0.05$, the advantage of ARNI in the number of reduction attribute subsets begins to appear, and its classification accuracy is higher than NRS in general. When $\varepsilon = 0.10$, the classification accuracy of ARNI is almost the same as that of NRS or better than NRS, but the number of attribute subsets by ARNI reduction is better.

Next, we provide a comprehensive comparison of the effects of ARNI, NRS, and other reduction algorithms based on non-neighborhood rough set theory. Those algorithms include the reduction algorithm classic rough set (CRS) [1] and principal component analysis (PCA) in the number of attributes and classification accuracy after reduction. For ARNI and NRS, we choose a moderate-performance parameter ($\varepsilon = 0.01$) as the radius of the neighborhood. For CRS, all continuous values in decision systems have been clustered into two categories by KMeans ($K = 2$). The reduction algorithm of CRS is the one based on the discernibility matrix. Because there are many results in the reduction sets, we randomly choose 3 results to calculate the average value as the final result in this paper. For PCA, to simplify the experimental analysis, the final number of attributes is set to be the same as ARNI.

The results of the comparative experiment are shown in Table 5. The classifiers perform better in the ARNI reduced decision systems than other algorithms. That is to say, the reduction of attributes based on ARNI provided better support for KNN and SVM than others.

4.3. Experiment about the Effect of ε

The granularity ε is a hyperparameter in ARNI. In this experiment, we will test the effect of attribute reduction and classification accuracy with different granularity ε. It will show the relationship between granularity ε and the number of attributes and classification accuracy after reduction.

The range of ε is from 0.02 to 1, and the step size is 0.02. The classifier is KNN ($K = 3$). The final accuracy is the average of 10-fold cross-validation. The detailed results are shown in Figure 2.

4.4. Experiment about GPU Acceleration

GPU architecture consists of a set of streaming multiprocessors (SMs), each of which contains a set of processor cores called streaming processors (SPs). Therefore, multiple threads are suitable for a large number of repeated or parallel operations. Using GPU acceleration operations can effectively improve computational efficiency.

Compute Unified Device Architecture (CUDA), introduced by NVIDIA, is a heterogeneous parallel computing model that involves both CPU and GPU. In this paper, we use CUDA V7.0 [37] and Pytorch framework [38] to recode ARNI in it. The program runs on CPU *versus* GPU. The acceleration times are shown in Figure 3. The speedup ratio ρ of GPU on different data sets is shown in Figure 4.

$$\rho = \frac{\text{CPU running time}}{\text{GPU running time}}$$

Table 4 Reduction results of two different algorithms.

ε	Data Sets	NRS			ARNI		
		Reduction	Accuracy		Reduction	Accuracy	
			KNN	SVM		KNN	SVM
0.01	BI	**9**	**0.8216**	**0.7128**	**9**	**0.8216**	**0.7128**
	DR	**7**	0.6221	0.5951	8	**0.6413**	**0.6245**
	FT	**4**	0.8108	**0.7293**	6	**0.8234**	0.7183
	GL	**4**	**0.6889**	**0.5653**	**4**	**0.6889**	**0.5653**
	IP	**5**	**0.9007**	**0.8638**	**5**	**0.9007**	**0.8638**
	IR	**3**	**0.9533**	**0.9667**	**3**	**0.9533**	**0.9667**
	ML	**7**	0.5956	0.3980	12	**0.6811**	**0.4235**
	SN	**4**	0.5889	0.5374	7	**0.6218**	**0.6134**
	WD	**4**	0.8999	0.9177	6	**0.9189**	**0.9255**
	WI	**4**	0.8144	0.8147	6	**0.95131**	**0.96663**
0.05	BI	14	0.8284	0.6948	**13**	**0.8328**	**0.7493**
	DR	13	0.6230	0.6168	**11**	**0.6334**	**0.6213**
	FT	**7**	**0.8393**	**0.7414**	9	0.8306	0.7298
	GL	9	0.6555	0.4696	**8**	**0.6641**	**0.4958**
	IP	**6**	**0.9091**	**0.8867**	**6**	**0.9091**	**0.8867**
	IR	**3**	**0.9533**	**0.9667**	**3**	**0.9533**	**0.9667**
	ML	31	**0.7011**	0.4111	**22**	0.6989	**0.4564**
	SN	**7**	0.5794	0.5753	11	**0.6375**	**0.6384**
	WD	**9**	0.9316	**0.9230**	10	**0.9489**	0.9216
	WI	5	0.8262	0.8141	**4**	**0.8783**	**0.8818**
0.1	BI	23	0.8159	**0.7658**	**20**	**0.8200**	0.7625
	DR	19	**0.6229**	0.6142	**17**	**0.6213**	**0.6169**
	FT	**17**	0.8429	**0.7379**	19	**0.8436**	**0.7379**
	GL	9	**0.6555**	**0.4696**	**7**	0.6479	0.4563
	IP	**8**	**0.8952**	**0.8780**	**8**	**0.8952**	**0.8780**
	IR	4	**0.9533**	**0.9667**	**3**	**0.9533**	**0.9667**
	ML	23	0.7000	0.4089	**19**	**0.7167**	**0.4360**
	SN	**9**	**0.6493**	**0.6492**	11	0.6375	0.6384
	WD	**11**	**0.9403**	**0.9213**	**11**	**0.9403**	**0.9213**
	WI	6	0.8655	0.8655	**4**	**0.8783**	**0.8818**

ARNI, Attribute Reduction based on Neighborhood Importance Degree; KNN, K-nearest neighbor; SVM, support vector machine.

Table 5 Reduction results of different algorithms.

Data Sets	RAW			CRS			NRS			PCA			ARNI		
	Reduction	KNN	SVM	Reduction	KNN	SVM	Reduction	KNN	SVM	Reduction	KNN	SVM	Reduction	KNN	SVM
BI	41	0.8368	0.7725	33	**0.8368**	**0.7772**	**9**	0.8216	**0.7128**	**9**	**0.8226**	0.7008	**9**	0.8216	**0.7128**
DR	19	0.6229	0.6142	17	0.6203	0.6142	**7**	0.6221	0.5951	8	0.6117	0.6156	8	**0.6413**	**0.6245**
FT	27	0.8168	0.7042	18	0.7908	0.6374	**4**	0.8108	0.7293	6	0.8025	0.7178	6	**0.8234**	**0.7183**
GL	10	0.6555	0.4696	8	0.6379	0.4973	**4**	**0.6889**	**0.5653**	**4**	0.6690	0.5277	**4**	**0.6889**	**0.5653**
IP	33	0.8553	0.8523	27	0.8437	0.8664	**5**	**0.9007**	**0.8638**	**5**	0.8693	0.8809	**5**	**0.9007**	**0.8638**
IR	4	**0.9533**	**0.9667**	4	**0.9533**	**0.9667**	**3**	**0.9533**	**0.9667**	**3**	**0.9533**	**0.9600**	**3**	**0.9533**	**0.9667**
ML	90	**0.7956**	0.5389	17	0.7389	0.4678	**7**	0.5956	0.3980	12	0.6900	**0.5889**	12	0.6811	0.4235
SN	60	0.6432	0.6388	9	0.6498	**0.7024**	**4**	0.5889	0.5374	7	0.6644	0.6726	7	**0.6722**	0.6134
WD	30	**0.9702**	0.9527	23	0.9648	**0.9474**	**4**	0.8999	0.9177	6	0.9499	0.9422	6	0.9189	0.9255
WI	13	**0.9555**	**0.9781**	13	**0.9555**	**0.9781**	**4**	0.8144	0.8147	6	0.9513	0.9666	6	0.9513	0.9667

ARNI, Attribute Reduction based on Neighborhood Importance Degree; KNN, K-nearest neighbor; SVM, support vector machine; CRS, classic rough set; PCA, principal component analysis.

5. CONCLUSION

It is a core challenge to the rough set theory about how to reduce attributes effectively in information systems. The matrix description of rough sets and reduction algorithm can be executed more efficiently than other forms in the computer operation, especially in GPU. In this paper, we propose an attribute reduction algorithm for neighborhood rough set based on Boolean matrix. Specifically, we define the dependency degree of decision attribute sets. Based on matrix description of neighborhood rough set, we also give a solving method for the relative dependency degree by matrix computing (Algorithm DBM). Furthermore, we realized the attribute reduction by matrix computing (Algorithm ARNI). The results in UCI datasets show that comparing the existing attribute reduction

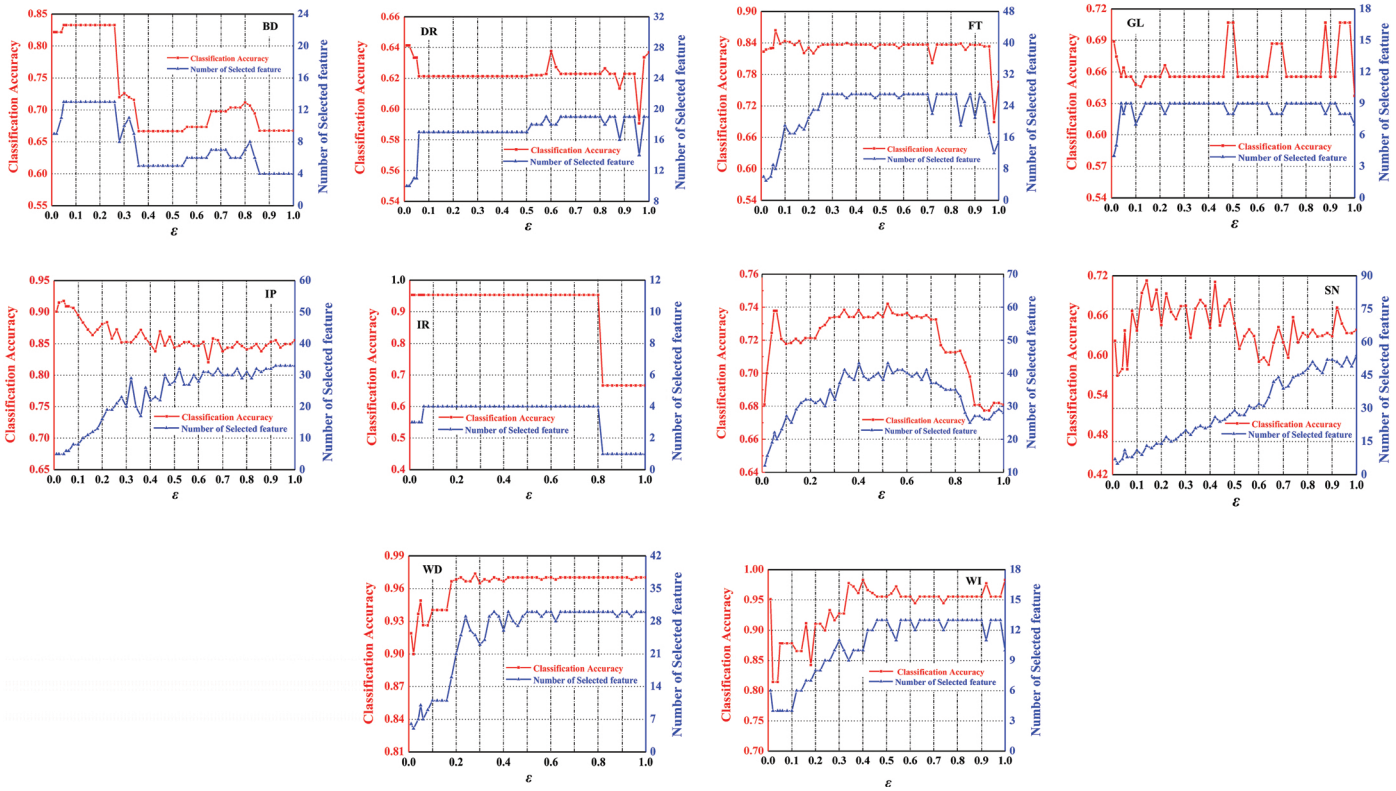

Figure 2 | Numbers of selected features and classification accuracy with the different granularity ε.

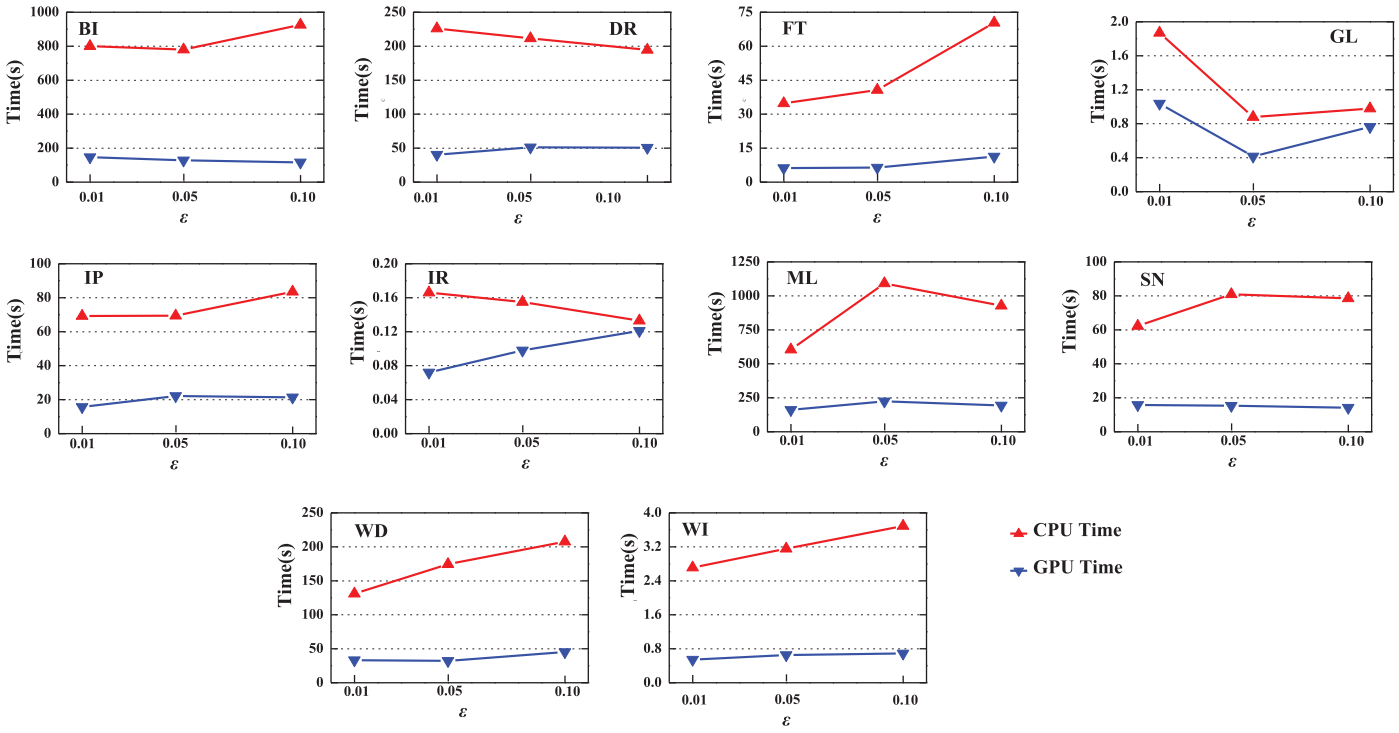

Figure 3 | Running time on using only central processing unit (CPU) and using graphics processing unit (GPU) acceleration with different data sets.

algorithms ARNI can reduce attributes less affecting the classification accuracy. Besides, we also show the acceleration performance of GPU for the matrixing model.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

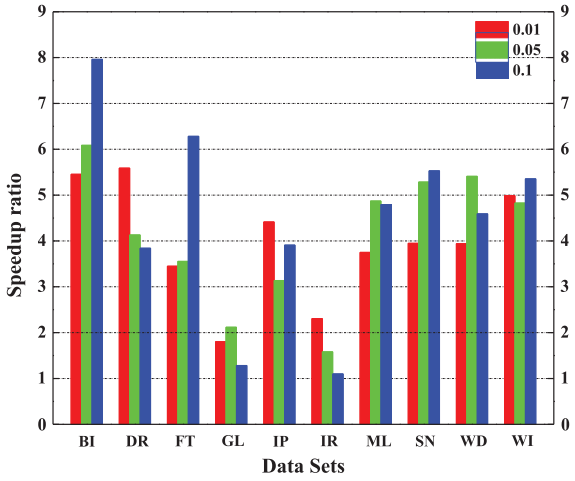

Figure 4 | Speedup ratio of graphics processing unit (GPU) with different datasets.

AUTHORS' CONTRIBUTIONS

Yan Gao conceived and designed the study. Changwei Lv performed the experiments and wrote the paper. Zhengjiang Wu reviewed and edited the manuscript. All authors read and approved the manuscript.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (No. 11601129, 61972134).

REFERENCES

[1] Z. Pawlak, Rough sets, Int. J. Comp. Inf. Sci. 11 (1982), 341–356.
[2] S. Angra, S. Ahuja, Machine learning and its applications: a review: in 2017, International Conference on Big Data Analytics and Computational Intelligence, Beijing, China, 2017, pp. 57–60.
[3] M. Aggarwal, Probabilistic variable precision fuzzy rough sets, IEEE Trans. Fuzzy Syst. 24 (2016), 29–39.
[4] Z. Pawlak, Rough sets and fuzzy sets, Fuzzy Set Syst. 17 (1985), 99–102.
[5] D. Meng, X. Zhang, K. Qin, Soft rough fuzzy sets and soft fuzzy rough sets, Cmput. Math. Appl. 62 (2011), 4635–4645.
[6] Y. Daren, A. Shuang, Q. Hu, Fuzzy mutual information based min-redundancy and max-relevance heterogeneous feature selection, Int. J. Comput. Intell. Sys. 4 (2011), 619–633.
[7] Z. Bonikowski, E. Bryniarski, U. Wybraniec-Skardowska, Extensions and intentions in the rough set theory, Inf. Sci. 107 (1998), 149–167.
[8] S.U. Kumar, H.H. Inbarani, PSO-based feature selection and neighborhood rough set-based classification for BCI multiclass motor imagery task, Neural Comput. Appl. 28 (2017), 3239–3258.
[9] W. Zhu, F. Wang, Reduction and axiomization of covering generalized rough sets, Inf. Sci. 152 (2003), 217–230.
[10] Z. Wu, N. Chen, Y Gao, Semi-monolayer cover rough set: Concept, property and granular algorithm, Inf. Sci. 456 (2018), 97–112.
[11] C. Wang, Q. He, M. Shao, Q. Hu, Feature selection based on maximal neighborhood discernibility, Int. J. Mach. Learn. Cybern. 9 (2018), 1929–1940.
[12] C. Wang, Q. Hu, X. Wang, D. Chen, Y. Qian, Feature selection based on neighborhood discrimination index, IEEE Trans. Neural Netw. Learn. Syst. 29 (2018), 2986–2999.
[13] Q. Hu, D. Yu, Z. Xie. Neighborhood classifiers, Expert Syst. Appl. 34 (2008), 866–876.
[14] Q. Hu, D. Yu, J. Liu, C. Wu, Neighborhood rough set based heterogeneous feature subset selection, Inf. Sci. 178 (2008), 3577–3594.
[15] G. Yu, Information structures in an incomplete information system: a granular computing viewpoint, Int. J. Comput. Intell. Syst. 11 (2018), 1179–1191.
[16] P. Yu, J. Li, W. Hongkun, G. Lin, Matrix-based approaches for updating approximations in multigranulation rough set while adding and deleting attributes, Int. J. Comput. Intell. Syst. 12 (2019), 855–872.
[17] X. Yang, S. Xu, H Dou, X. Song, H. Yu, J. Yang, Multigranulation rough set: a multiset based strategy, Int. J. Comput. Intell. Syst. 10 (2017), 277–292.
[18] Y. Qian, X. Liang, Q. Wang, J. Liang, B. Liu, A. Skowron, Y. Yao, J. Ma, C. Dang, Local rough set: a solution to rough data analysis in big data, Int. J. Approx. Reason. 97 (2018), 38–63.
[19] T. Lin, K. Huang, Q. Liu, Rough sets, neighborhood systems and approximation, in Proceedings of the Fifth International Symposium on Methodologies of Intelligent Systems, 1990, vol. 22, pp. 130–141.
[20] Q. Wang, Y. Qian, X. Liang, G. Guo, J. Liang, Local neighborhood rough set, Knowl. Based Syst. 153 (2018), 53–64.
[21] A. Huang, W. Zhu, Characteristic matrices of compound operations of coverings and their relationships with rough sets, Int. J. Mach. Learn. Cybern. 10 (2019), 75–85.
[22] S. Wang, W. Zhu, Q. Zhu, F. Min, Characteristic matrix of covering and its application to Boolean matrix decomposition, Inf. Sci. 263 (2014), 186–197.
[23] L. Ma, The investigation of covering rough sets by Boolean matrices, Int. J. Approx. Reason. 100 (2018), 69–84.
[24] J. Fan, T. Chow, Deep learning based matrix completion, Neurocomputing. 266 (2017), 540–549.
[25] N.L. Zamarashkin, D.A. Zheltkov, GPU based acceleration of parallel block Lancoz solver, Lobachevskii J. Math. 39 (2018), 596–602.
[26] D.I. Lyakh, An efficient tensor transpose algorithm for multicore CPU, Intel Xeon Phi, and NVidia Tesla GPU, Comput. Phys. Commun. 189 (2015), 84–91.
[27] R. Skinderowicz, The GPU-based parallel ant colony system, J. Parallel Distr. Comput. 98 (2016), 48–60.
[28] S. Jing, G. Li, K. Zeng, W. Pan, C. Liu, Efficient parallel algorithm for computing rough set approximation on GPU, Soft. Comput. 22 (2018), 7553–7569.
[29] A. Singh, K. Deep, P. Grover, A novel approach to accelerate calibration process of a k-nearest neighbours classifier using GPU, J. Parallel Distr. Comput. 104 (2017), 114–129.
[30] J. Zhang, Y. Zhu, Y. Pan, T. Li, Efficient parallel Boolean matrix based algorithms for computing composite rough set approximations, Inf. Sci. 329 (2016), 287–302.

[31] W. Zhu, Relationship between generalized rough sets based on binary relation and covering, Inf. Sci. 179 (2009), 210–225.
[32] W. Ziarko, Variable precision rough set model, J. Comput. Syst. Sci. 46 (1993), 39–59.
[33] Y. Yao, Probabilistic rough set approximations, Int. J. Approx. Reason. 49 (2008), 255–271.
[34] Y. Shi, Y. Huang, C. Wang, Q. He, Attribute reduction based on the Boolean matrix, Granul. Comput. 4 (2019), 313–322.
[35] G. Liu, L Li, J. Yang, Y. Feng, K. Zhu, Attribute reduction approaches for general relation decision systems, Pattern. Recogn. Lett. 65 (2015), 81–87.
[36] D. Dua, C. Graff, UCI Machine Learning Repository, 2019. https://archive.ics.uci.edu/ml/datasets.php
[37] Nvidia, CUDA Toolkit Documentation v8.0, 2017. https://docs.nvidia.com/cuda/archive/8.0
[38] PyTorch, PyTorch 0.4 Chinese document, 2019. https://pytorch.apachecn.org/docs/0.4