The Journal of Engineering

The 3rd Asian Conference on Artificial Intelligence Technology (ACAIT 2019)

Attribute reduction based on improving DIT in interval-valued ordered information system

eISSN 2051-3305
Received on 11th October 2019
Accepted on 19th November 2019
E-First on 27th July 2020
doi: 10.1049/joe.2019.1158
www.ietdl.org

Lei Yang[1], Xiaoyan Zhang[2] ✉, Weihua Xu[2], Binbin Sang[3]
[1]*School of Science, Chongqing University of Technology, Chongqing, 400054, People's Republic of China*
[2]*College of Artificial Intelligence, Southwest University, Chongqing, 400715, People's Republic of China*
[3]*School of Information Science and Technology, Southwest Jiaotong University, Chengdu, 611756, People's Republic of China*
✉ *E-mail: zxy19790915@163.com*

Abstract: Traditional attribute reduction based on discernibility information tree (DIT) is proposed under the equivalence relation, which effectively realises the compression and storage of discernible matrix. However, it has not been studied whether it is applicable under the dominant relation. In this study, under the background of an interval-valued ordered information system, the improved DIT (IDIT) based on a discernible matrix is constructed by combining the core attributes and attribute significance. Furthermore, the compressed storage of the discernible matrix is realised. Moreover, only one reduction of the information system can be found through traditional DIT. By changing the ranking order of attributes in the information system, this study finds all reductions in the interval-valued ordered information system by the IDIT.

1 Introduction

Rough set theory was founded in the 1980s by Pawlak [1], a professor of Warsaw University of Technology and academician of the Polish Academy of Sciences. After more than 30 years of rapid development, a relatively complete theoretical system has been formed [2–7], and a very rich application research results have been achieved. In particular, compared with other methods, a rough set theory is more suitable for dealing with uncertainties contained in information systems. At present, the rough set has been successfully applied in many fields, such as industrial control [8], biomedicine [9], fault diagnosis [10], market forecasting [11] etc. Rough set theory has developed rapidly and achieved fruitful results in recent years. Chinese scholars have also made important systematic research results [12–15], which have attracted the attention of foreign counterparts. The extension of rough set has two aspects, one is the extension of theory, the other is the extension of application. The extension of the theory is based on the extension of equivalence relation [16], approximated set and inclusion operator [5]. The extension of the application is mainly based on the extension of attribute reduction method and pattern classification.

Attribute reduction is an important research topic in rough set theory. Generally speaking, knowledge (attributes) in the knowledge base (information system) is not equally important; even some conditional attributes are redundant. The existence of redundancy, on the one hand, is a waste of resources (need storage space and processing time); on the other hand, it is not conducive to people to make correct and concise decisions. Attribute reduction simply seeks to delete irrelevant or unimportant attributes while keeping the classification and decision-making ability of the knowledge base unchanged. Many researchers have done this work [17–20]. At present, there are many attribute reduction algorithms based on rough sets, many of which are heuristic methods. Most heuristic algorithms select attributes according to the importance of attributes. According to the different measurement methods of attribute importance, the existing algorithms can be divided into three categories. The first one is based on the discernible matrix (DM), in which the importance of attributes is measured by the frequency of attributes appearing in the DM. The second one is based on Pawlak's concept of attribute importance, which defines attribute importance from an algebraic point. The third method is based on information entropy, which uses information entropy to describe attribute importance [21]. In [22], attribute reduction algorithms based on DM, information entropy, distribution, maximum distribution, approximation, and positive regions are studied, and their relationships are obtained. Most of these reduction algorithms are based on positive regions. Recently, some reduction algorithms based on boundary regions have been proposed [23–26].

In addition to the heuristic method, the discernibility matrix also provides a good idea for attribute reduction. The discernibility matrix concentrates the discernibility information of all attributes in the information system into a matrix. A discernibility matrix obtained by an information system under equivalence relation is a symmetric matrix with empty diagonal, but it does not satisfy the symmetry if the discernibility matrix under dominance relation. The discernibility matrix was proposed by Skowron and Rauszer [27]. At present, many excellent attribute reduction methods have been proposed based on discernibility matrix and discernibility function. Based on the discernibility matrix, an attribute reduction method was proposed by Konecny in concept lattices [28]. Based on discernibility matrix simplification, a reduct construction method was proposed by Yao and Zhao [29]. Dai *et al.* [30] proposed maximal discernibility pairs based approach to attribute reduction in fuzzy rough sets. In order to simplify the discernibility function of a decision table, Liang et al. [31] proposed a method of sample pair selection based on rough set. It was found that only the minimal elements in the discernibility matrix were employed to find reductions. Du and Hu [32] presented a theoretical method based on the discernibility matrix to enumerate all reducts and a practical approach on the basis of significance to find one reduct. Zhang *et al.* [33] proposed the attribute reduction method in concept lattice. Also, using the idea similar to Skowron and Rauszer's discernibility matrix, the discernibility matrix and function of a concept lattice were defined. However, when the number of objects or attributes increases, the storage cost of the discernibility matrix is too high. In order to effectively reduce the storage cost of the discernibility matrix, after introducing the concentrate tree (C-Tree), Yang and Yang [34] proposed the attribute reduction algorithm based on C-Tree, which stores every non-empty element of the discernibility matrix on the corresponding path of C-Tree. In order to satisfy the dynamic change of information system, the improved algorithm of C-Tree was given in [35], which effectively reduces the storage cost. In order to satisfy the dynamic change of information system, Yang and Jing [36] proposed an incremental updating algorithm based on C-Tree. The C-tree in [34] mainly achieves compressed storage of discernibility matrix by eliminating repetitive elements, but it can't eliminate the redundant parent set elements. In order to eliminate redundant parent set elements, Jiang [37] proposed a discernibility information tree (DIT). Later, the core attributes and the importance of attributes were introduced to improve the DIT [38]. However, all these studies are based on the equivalence relation, and the DIT can only be a reduction of the information system. Yang *et al.* [39] introduced a dominance relation to attribute reduction based on DIT in the interval-valued ordered information system. In this study, the attribute reduction method based on improved DIT (IDIT) in the interval-valued ordered information system is studied by introducing core attributes and attribute importance. Further compression and storage of DM are realised, and the disadvantage that only one reduction can be obtained through DIT is solved.

The structure of this paper is divided into the following parts: the introduction section mainly introduces the work done by predecessors related to this topic and the motivation of our research on this topic. In Section 2, the basic knowledge of attribute reduction and DIT is introduced. In Section 3, we study the construction of DIT in the initial order and other order and propose an improved algorithm for the construction of DIT in the interval-valued ordered information system based on the core attributes and the importance of attributes. In Section 4, an attribute reduction algorithm based on improved DIT is presented, and all attribute reduction of the interval-valued information system is solved by changing the ranking order of attributes. The conclusion of this paper is given in Section 5.

2 Preliminaries

In this section, we briefly introduce some basic concepts about attribute reduction, distinguish matrix, distinguish function, attribute reduction of the interval-valued ordered information system, and DIT.

2.1 Attribute reduction

In the knowledge base, $K = (U, R)$, deleting some relations does not weaken its classification ability. For example, when a doctor treats a patient, he does not require a patient to have a general examination before drawing a conclusion. If a doctor does that, it will inevitably delay the patient's time and greatly increase the patient's medical expenses. This shows that some relationships in the knowledge base are redundant. In the process of knowledge processing, these redundant relationships will inevitably lead to unnecessary computational load. Therefore, we need to delete these redundant relations before knowledge processing, and then reduce the computational load.

Let R be a family of equivalence relations and $r \in R$. If $\text{ind}\,(R) = \text{ind}\,(R - \{r\})$, then r is called an unnecessary relation in R, otherwise r is called a necessary relation in R. If each $r \in R$ is necessary, then R is called independent, otherwise, R is not independent. Let $P \subseteq R$, if P is independent and $\text{ind}(P) = \text{ind}(R)$, then P is called a reduction of R.

The intersection of all reductions of R constitutes the core of R, i.e. $\text{Core}(R) = \bigcap \text{Red}(R)$, where $\text{Core}(R)$ denotes the core of R and $\text{Red}(R)$ denotes all reductions of R.

It can be seen that the use of Core has two aspects: firstly, it can be used as the basis of all reductions, because Core is included in all reductions, and the calculation can be carried out directly; secondly, it can be interpreted as a set of attributes that cannot be deleted in attribute reduction.

2.2 Distinguish matrix and distinguish function

There are many advantages in using a discernibility matrix to represent knowledge, especially it can calculate reduction and core easily.

Let $\boldsymbol{S} = (U, A, V, f)$ be a knowledge representation system, $|U| = n$. The distinguish matrix $\boldsymbol{S}$ is of $n \times n$, and any element of $\boldsymbol{S}$ is

$$\alpha(x, y) = \{a \in A \,|\, f(x, a) \neq f(y, a)\}.$$

Therefore, $\alpha(x, y)$ is a collection of all attributes that distinguish object x from object y.

Next, a Boolean function is introduced, which is called a discernible function and is represented by Δ. For each $a \in A$, we specify a Boolean variable 'a'. If $a(x, y) = \{a_1, a_2, \ldots, a_k\} \neq \varnothing$, a Boolean function $a_1 \vee a_2 \vee \cdots \vee a_k$ is specified and expressed in $\sum^a (x, y)$; if $a(x, y) = \varnothing$, a Boolean constant 1 is specified. The discernible function can be expressed as follows:

$$\Delta = \prod_{(x, y) \in U \times U} \sum^a (x, y).$$

The discriminant function Δ has the following properties: all conjunctions in the minimal disjunctive paradigm of function Δ are all reductions of the attribute set A. In other words, the reduction is a minimal subset of attributes of the entire attribute set and can distinguish all objects.

It is easy to see that if $B \subseteq A$ is a minimal subset satisfying condition

$$B \bigcap a\ (x, y) \neq \varnothing, \forall a\ (x, y) \neq \varnothing,$$

then B is a reduction of A.

The Core is a set of all the individual elements in the discernibility matrix, i.e.

$$\text{Core}(A) = \{a \in A \,|\, a(x, y) = \{a\}, x \in U, y \in U\}.$$

2.3 Attribute reduction of interval-valued ordered information system

In the interval-valued ordered information system ($\text{I}_\text{v}\text{OIS}$), attribute reduction is to remove some unnecessary criteria from the information system based on the interval-valued dominant relation, obtain a minimum set of criteria, and simplify knowledge expression.

The triple $I = (U, A, F)$ is called an information system, where U is a finite set of objects, A is a finite set of conditional attributes, F is a set of relations between U and A. If $\forall f \in F, a \in A$, and $x_i \in U$ have $f(x_i, a) = \left[a^L(x_i), a^U(x_i)\right]$, then $I = (U, A, F)$ is the interval-valued information system, where both $a^L(x_i)$ and $a^U(x_i)$ are real numbers and satisfy $a^L(x_i) \leq a^U(x_i)$; $f(x_i, a)$ is the attribute value of object x_i under attribute a and $f(x_i, a)$ is an interval number. When $a^L(x_i) = a^U(x_i)$, the attribute value $f(x_i, a)$ is a real number, so the interval value information system is a generalisation of the single value information system.

Let $I = (U, A, F)$ be an interval-valued information system. For any $a \in A$, the attribute values in the interval-valued information system can be compared. Then two definitions are given

$$\begin{aligned} f(x_i, a) \trianglelefteq f(x_i, a) &\Leftrightarrow (\forall a \in A) \\ &\times \left[a^L(x_i) \leq a^L(x_j), a^U(x_i) \leq a^U(x_j)\right], \\ f(x_i, a) \trianglerighteq f(x_i, a) &\Leftrightarrow (\forall a \in A) \\ &\times \left[a^L(x_i) \geq a^L(x_j), a^U(x_i) \geq a^U(x_j)\right], \end{aligned}$$

where $\trianglelefteq$ and $\trianglerighteq$ can construct increasing and decreasing partial orders, respectively, in the interval-valued information system. If the range of an attribute in the interval value information system is increasing or decreasing, it is called the criterion. In this study, we consider the case that the dominant relation is composed of increasing partial order, and the case of decreasing partial order can be given in the same way. $I = (U, A, F)$ is called $\text{I}_\text{v}\text{OIS}$ if all conditional attributes in the interval-valued information system are criteria, recorded as $I^{\geq}$.

In the $\text{I}_\text{v}\text{OIS}$, A is the criterion set. $\forall a \in A$, existing advantage relation '$\geq_a$', $x_j \geq_a x_i$ denotes that x_j is better than x_i about a. Then, we can have $x_j \geq_A x_i \Leftrightarrow (\forall a \in A)[x_j \geq_a x_i]$. The interval-valued dominance relation $R_A^{\gtrsim}$ is defined as follows:

$$\begin{aligned} R_A^{\gtrsim} &= \{(x_i, x_j) \in U \times U \mid x_j \geq_a x_i, \forall a \in A\} \\ &= \{(x_i, x_j) \in U \times U \mid (\forall a \in A) \\ &\quad \times [a^L(x_i) \leq a^L(x_j), a^U(x_i) \leq a^U(x_j)]\}. \end{aligned}$$

The interval dominance class $[x_i]_A^{\gtrsim}$ induced by interval-valued dominance relationship $R_A^{\gtrsim}$ is

$$\begin{aligned} [x_i]_A^{\gtrsim} &= \{x_j \in U \mid (x_i, x_j) \in R_A^{\gtrsim}\} \\ &= \{x_j \in U \mid (\forall a \in A) \\ &\quad \times [a^L(x_i) \leq a^L(x_j), a^U(x_i) \leq a^U(x_j)]\}. \end{aligned}$$

Note 1: referring to different practical meanings, we can define different interval dominance relations, such as

- $R_A^{\gtrsim 1} = \{(x_i, x_j) \in U \times U \mid a^U(x_i) \leq a^L(x_j), \forall a \in A\}$,
- $R_A^{\gtrsim 2} = \{(x_i, x_j) \in U \times U \mid a^L(x_i) \leq a^L(x_j), \forall a \in A\}$,
- $R_A^{\gtrsim 3} = \{(x_i, x_j) \in U \times U \mid a^U(x_i) \leq a^U(x_j), \forall a \in A\}$.

Let

$$I^{\gtrsim} = (U, A, F)$$

be an $\mathrm{I_vOIS}$,

$$\mathrm{Dis}_{\geq A}(x_i, x_j) = \{a \in A \mid (x_i, x_j) \notin R_a^{\gtrsim}\}$$

is the distinguishable attribute set of objects x_i, x_j in $I^{\gtrsim}$ about $R_A^{\gtrsim}$. Record

$$\mathrm{Dis}_{\geq A} = (\mathrm{Dis}_{eA}(x_i, x_j))_{|U| \times |U|},$$

where $\mathrm{Dis}_{\geq A}$ is the DM of the object in $I^{\gtrsim}$ with respect to $R_A^{\gtrsim}$. In particular, for any $x_i, x_j \in U$ there are

$$\mathrm{Dis}_{\geq A}(x_i, x_i) = \varnothing$$

and get:

$$\mathrm{Dis}_{\geq A}(x_i, x_j) \bigcap \mathrm{Dis}_{\geq A}(x_j, x_i) = \varnothing.$$

$$M_{\geq A} = \wedge \{ \vee \{a \mid a \in \mathrm{Dis}_{\geq A}(x_i, x_j)\}\}$$

is called the identifiable formula of $I^{\gtrsim}$ for $R_A^{\gtrsim}$. Next, according to identifiable formula, the definition of attribute reduction in IvOIS is shown as follows [40].

Definition 1: Let $I^{\gtrsim} = (U, A, F)$ be an $\mathrm{I_vOIS}$. The minimal disjunctive paradigm corresponding to

$$M_{\geq A} = \wedge \{ \vee \{a \mid a \in \mathrm{Dis}_{\geq A}(x_i, x_j)\}\}$$

is defined as follows:

$$M_{\geq \min} = \bigvee_{k=1}^{p} \left(\bigwedge_{s=1}^{q_k} a_s \right).$$

Note $B_k = \{a_s \mid s = 1, 2, \ldots, q_k\}$, then $\{B_k \mid k = 1, 2, \ldots, p\}$ is the set of all reductions in the $\mathrm{I_vOIS}$ $I^{\gtrsim}$.

2.4 Discernibility information tree (DIT)

For the identifiable matrix corresponding to the $\mathrm{I_vOIS}$, any element contained in the identifiable matrix is a subset of the condition attributes in the information system. There are many redundant repetitive elements and parent set elements in the given identifiable matrix. These elements have no contribution to find the reduction, but their existence occupies a lot of storage space and increases the computational complexity of solving attribute reduction. In order to realise the compression storage of the difference matrix without losing the reduction information contained in the discernibility matrix, a construction method of DIT based on a discernibility matrix is proposed in [37]. Next, the definition of discernibility information tree in reference [37] is shown as follows.

Definition 2: The DIT is an ordered tree. Each node in the tree has at most $|A|$ sub-trees ($|A|$ is the number of attributes in the information system). The sub-trees of the DIT are ordered trees, and the order cannot be arbitrarily reversed. The specific definitions are as follows:

(1) Each node in the DIT is mainly composed of four parts: an attribute name, a father pointer, a child pointer, and a homonym pointer. The attribute name identifies the condition attribute corresponding to the node; the father pointer points to the parent node of the node; the child pointer points to the child node of the node; the homonym pointer points to the node in the other path having the same attribute name as the node.
(2) The attribute pointer header table consists of two parts: the attribute name and the homonymous pointer. The attribute name identifies the condition attribute corresponding to the entry; the homonymous pointer points to the leftmost node that has the same attribute name as the entry in the DIT.

Algorithm 1 (see Fig. 1) introduces the construction method of DIT based on the discernibility matrix in the interval-valued ordered information system in [37].

3 Construction of IDIT based on core attribute and attribute significance

3.1 DIT constructed under the initial order of attributes

The initial order of attributes is an arrangement of conditional attributes in the information system. In this study, the initial order of attributes is defined as the left-to-right sequence $\mathcal{S}$ of conditional attributes in the information system. Next, we have given an illustrative example (Example 1) with the sequence: a, b, c, d, and e, denoted by $\mathcal{S} = \langle abcde \rangle$.

Example 1: Given an interval-valued ordered information system $I^{\gtrsim} = (U, AT, F)$ as shown in Table 1, where $U = \{u_1, u_2, \ldots, u_{10}\}$, $AT = \{a, b, c, d, e\}$.

Let $\mathcal{S} = \langle abcde \rangle$. The DIT based on the DM is given by Algorithm 1 (Fig. 1) as follows. Firstly, the root node of the DIT is created, marked as 'null'. Then, the DM of the interval-valued ordered information system is calculated, and the results are shown in Table 2. Finally, the elements in the DM (identifiable attribute sets) are inserted into the tree. It should be noted that the elements in the identifiable attribute set are arranged in sequence $S = \langle abcde \rangle$. The specific insertion process is as follows.

1. The path (abce) corresponding to the first non-empty identifiable attribute set $\{a, b, c, e\}$ in the identifiable matrix is constructed and inserted into the DIT based on the DM.
2. For the second non-empty identifiable attribute set $\{a, b, c, e\}$, because the path ($abce$) corresponding to $\{a, b, c, e\}$ already exists in the DIT based on the distinguishable matrix, no new path is constructed. Similarly, all identical distinguishable attribute sets are mapped to the same path.
3. For distinguishable attribute set $\{a, c\}$, its corresponding path (ac) is created, which has the same prefix (a) as path ($abce$).

4. For distinguishable attribute set $\{d\}$, its corresponding path (d) is created.
5. For distinguishable attribute set $A = \{a, b, c, d, e\}$, its corresponding path ($abcde$) is created, which has the same prefix (abc) as path ($abce$).
6. For distinguishable attribute set $A = \{a, c, d\}$, it completely contains the distinguishable attribute set $A = \{a, c\}$ corresponding to path (ac). Therefore, the non-extended path strategy is adopted, and the node d is not constructed.
7. For distinguishable attribute set $\{b, e\}$, its corresponding path (b, e) is created.

As shown in Fig. 2, according to the distinguishable matrix in Table 2, we construct a DIT based on the DM in the $\mathrm{I_vOIS}$.

Input : An interval-valued ordered information system$I^{\geq} = \{U, AT, F\}$
Output : DIT based on discernible matrix in interval-valued ordered information system
begin
  Step1 : Constructing the root node TN of DIT based on discernible matrix, marked as "*null*";
  Step2 : Calculating the discernible matrix; the order of the elements in identifiable attribute set in the discernible matrix is arranged in the order of condition attributes in the interval-valued ordered information system.
  Step3 :
  for $i = 1 : n$ **do**
    for $j = 1 : n$ **do**
      while $Dis_{\geq A}(x_i, x_j) \neq \emptyset$ **do**
        A. Let attribute a be the leftmost element arranged in $Dis_{\geq A}(x_i, x_j)$.
        B.
        if *There is a child node N in TN and the attribute name of N is a.* **then**
          if *N is a leaf node* **then**
            Do not create the node corresponding to the remaining attributes in $Dis_{\geq A}(x_i, x_j)$ (no extended path strategy);
          end
          if *a is the last element in* $Dis_{\geq A}(x_i, x_j)$ **then**
            Delete the subtree rooted at N from the constructed DIT but retain N(Delete subtree strategy);
          end
          else
            $TN = N$
          end
        end
        else
          Create a new node N', node N' as a child of TN, and let the node name of N' be a. And N' is connected to the node with the same attribute name as N' passing the homonymy pointer of the node, thus forming a chain of the same name attribute node.
          Let $TN = N'$.
        end
        C. $Dis_{\geq A}(x_i, x_j) = Dis_{\geq A}(x_i, x_j) - \{a\}$
      end
      Note: each time an identifiable attribute set is built, TN is returned to the root node.
    end
  end
end

Fig. 1 *Algorithm 1: construction of DIT based on the DM in interval-valued ordered information system*

Table 1 An interval-valued ordered information system

U	a	b	c	d	e
u_1	1	[0, 1]	2	1	[1, 2]
u_2	[0, 1]	0	[1, 2]	2	1
u_3	[0,1]	0	[1, 2]	1	1
u_4	0	0	1	2	1
u_5	2	[1, 2]	3	[1, 2]	[2, 3]
u_6	[0, 2]	[1, 2]	[1, 3]	[1, 2]	[2, 3]
u_7	1	1	2	1	2
u_8	[1, 2]	[1, 2]	[2, 3]	2	[2, 3]
u_9	[1,2]	2	[2, 3]	[0, 2]	3
u_{10}	2	2	3	[0, 1]	3

Table 2 Identifiable matrix of the interval-valued ordered information system in Table 1

$\mathrm{Dis}_{\geq \mathrm{AT}}$	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}
u_1	$\emptyset$	$\{a,b,c,e\}$	$\{a,b,c,e\}$	$\{a,b,c,e\}$	$\emptyset$	$\{a,c\}$	$\emptyset$	$\emptyset$	$\{d\}$	$\{d\}$
u_2	$\{d\}$	$\emptyset$	$\{d\}$	$\{a,c\}$	$\{d\}$	$\{d\}$	$\{d\}$	$\emptyset$	$\{d\}$	$\{d\}$
u_3	$\emptyset$	$\emptyset$	$\emptyset$	$\{a,c\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\{d\}$	$\{d\}$
u_4	$\{d\}$	$\emptyset$	$\{d\}$	$\emptyset$	$\{d\}$	$\{d\}$	$\{d\}$	$\emptyset$	$\{d\}$	$\{d\}$
u_5	AT	$\{a,b,c,e\}$	AT	$\{a,b,c,e\}$	$\emptyset$	$\{a,c\}$	AT	$\{a,c\}$	$\{a,c,d\}$	$\{d\}$
u_6	AT	$\{a,b,c,e\}$	AT	$\{a,b,c,e\}$	$\emptyset$	$\emptyset$	AT	$\emptyset$	$\{d\}$	$\{d\}$
u_7	$\{b,e\}$	$\{a,b,c,e\}$	$\{a,b,c,e\}$	$\{a,b,c,e\}$	$\emptyset$	$\{a,c\}$	$\emptyset$	$\emptyset$	$\{d\}$	$\{d\}$
u_8	AT	$\{a,b,c,e\}$	AT	$\{a,b,c,e\}$	$\{d\}$	$\{a,c,d\}$	AT	$\emptyset$	$\{d\}$	$\{d\}$
u_9	AT	$\{a,b,c,e\}$	AT	$\{a,b,c,e\}$	$\{b,e\}$	$\{a,b,c,e\}$	AT	$\{b,e\}$	$\emptyset$	$\{d\}$
u_{10}	$\{a,b,c,e\}$	$\{a,b,c,e\}$	$\{a,b,c,e\}$	$\{a,b,c,e\}$	$\{b,e\}$	$\{a,b,c,e\}$	$\{a,b,c,e\}$	$\{a,b,c,e\}$	$\{a,c\}$	$\emptyset$

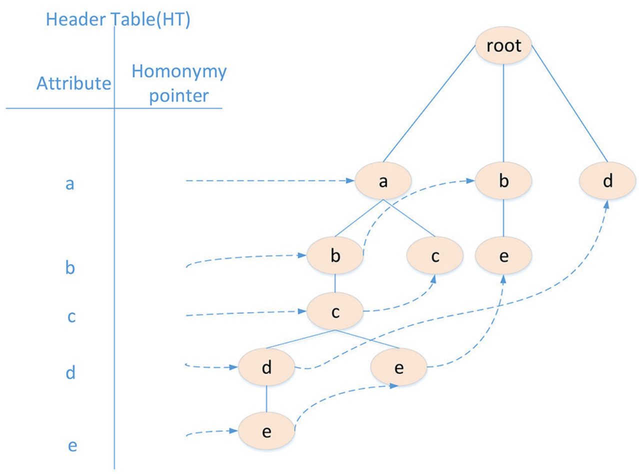

Fig. 2 *DIT based on the DM in Table 2 under the initial order of attributes*

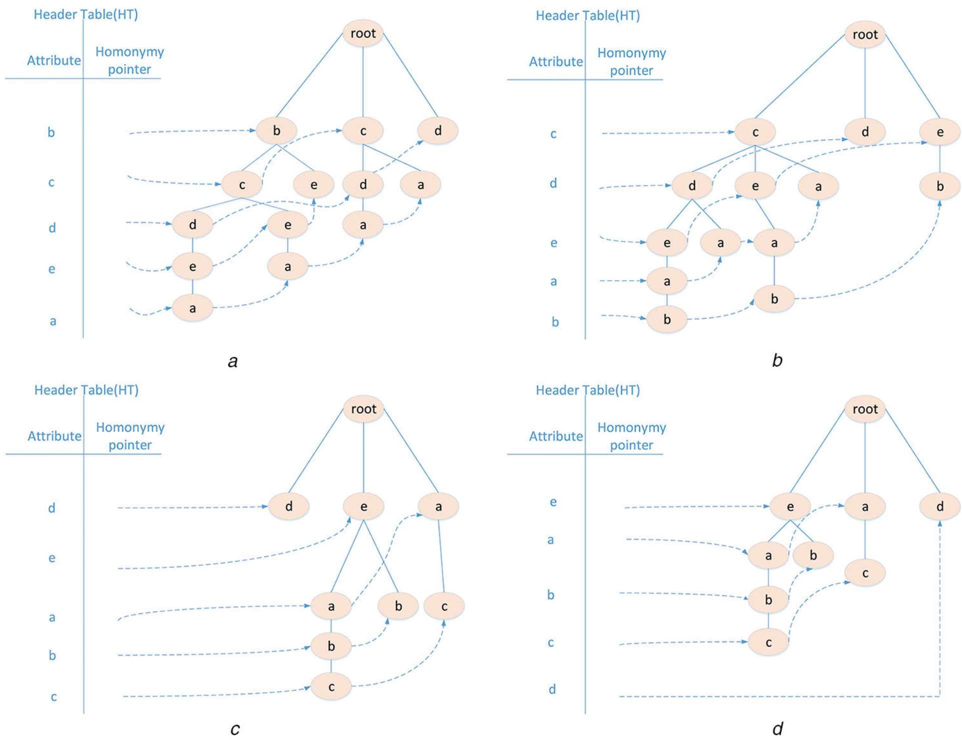

Fig. 3 *DIT based on the DM in Table 2 under the other orders of attributes*

From Fig. 2, it appears that the DIT based on the DM has no duplicate elements. Moreover, the distinguishable attribute sets $\{a, b, c, e\}$, $\{a, b, c, d, e\}$ and $\{a, c\}$ share the prefix (a), and the distinguishable attribute sets $\{a, b, c, e\}$ and $\{a, b, c, d, e\}$ share the prefix (abc). Thus, the compressed storage of the identifiable matrix is realised. However, in this section, only the construction of the DIT based on the DM in the initial order is considered. The construction of the DIT based on the DM under the other order of attributes and its effect on the compression storage of the difference matrix is not studied.

3.2 DIT constructed under the other order of attributes

For the identifiable matrix in Table 2, the elements in each identifiable attribute set are arranged in sequence $\mathcal{S} = \langle bcdea \rangle$, $\mathcal{S} = \langle cdeab \rangle$, $\mathcal{S} = \langle deabc \rangle$, $\mathcal{S} = \langle eabcd \rangle$. The DITs based on the identifiable matrix are obtained as shown in Fig. 3.

Figs. 3*a* and *b* contain 13 nodes. Figs. 3*c* and *d* only contain eight nodes. Obviously, from the perspective of storage cost, the DIT in Fig. 3*a* or Fig. 3*b* is the worst, while the DIT in Fig. 3*c* or Fig. 3*d* has the smallest size. As mentioned above, intuitively, we can find that if the core attributes play a role in the construction of DIT, the more compact the DIT is. In addition, we know that the higher the frequency of the attributes that appear in the DM, the more important the attribute is. Therefore, this study introduces the construction of a DIT based on core attributes and the construction of a DIT based on a new attribute ranking criterion, i.e. the order of attributes is arranged in descending order of importance.

3.3 Construction of IDIT)

3.3.1 Construction of IDIT based on a core attribute: The core of the interval-valued ordered information system is a subset of the condition attribute set, which is recorded as $\mathrm{Core}(A)$. It is the intersection of all reductions in the information system. For the discernible matrices, the union of all identifiable attribute sets only containing one element constitutes the core of the information system. For the DIT based on the DM, the union of the paths only containing a single node in the tree constitutes the core of the information system. As shown in Fig. 2, $\mathrm{Core}(A) = \{d\}$. The elements in $\mathrm{Core}(A)$ are indispensable components for us to obtain reductions. The reason why it is important is that if any element in $\mathrm{Core}(A)$ is deleted will affect the classification of the system. Based on this, when constructing the DIT based on the DM, the kernel is used as the heuristic information. We not only can eliminate the paths that have been built in the tree that contain core

```
Input   : An interval-valued ordered information system I^≥ = {U, AT, F}
Output  : The IDIT, Core(A)
begin
    Step1 : Let Core(A)=∅;
    Step2 : Constructing the root node TN of IDIT based on discernible matrix, marked as "null";
    Step3 : Calculating the discernible matrix;
    Step4 :
    for i = 1 : length(U) do
        for j = 1 : length(U) do
            if Dis_{≥A}(x_i, x_j) ∩ core(A) == ∅ then
                if |Dis_{≥A}(x_i, x_j)| == 1 then
                    core(A) = core(A) ∪ Dis_{≥A}(x_i, x_j) and delete all paths containing the core attribute according to the
                    attribute pointer header table.
                end
                while Dis_{≥A}(x_i, x_j) ≠ ∅ do
                    A. Let attribute a be the leftmost element arranged in Dis_{≥A}(x_i, x_j).
                    B.
                    if There is a child node N in TN and the attribute name of N is a. then
                        if N is a leaf node then
                            Do not create the node corresponding to the remaining attributes in Dis_{≥A}(x_i, x_j) (no
                            extended path strategy);
                        end
                        if a is the last element in Dis_{≥A}(x_i, x_j) then
                            Delete the subtree rooted at N from the constructed IDIT tree but retain N(Delete subtree
                            strategy);
                        end
                        else
                            TN = N
                        end
                    end
                    else
                        Create a new node N′, node N′ as a child of TN, and let the node name of N′ be a. And N′ is
                        connected to the node with the same attribute name as N′ passing the homonymy pointer of the
                        node, thus forming a chain of the same name attribute node.
                        Let TN = N′.
                    end
                    C. Dis_{≥A}(x_i, x_j) = Dis_{≥A}(x_i, x_j) − {a}.
                end
            end
            Note: Each time an identifiable attribute set is built, TN is returned to the root node.
        end
    end
end
```

Fig. 4 *Algorithm 2: construction of IDIT based on core attributes*

attributes but also do not need to construct the path for the discernible attributes set containing core attributes later than the core attributes. The compressive storage of DM is further realised compared with the method in [39].

Example 2: Based on Table 2 and Algorithm 2 (see Fig. 4), the specific construction process of the IDIT based on core attribute is given as follows.

(1) The path $(abce)$ corresponding to the first non-empty identifiable attribute set $\{a, b, c, e\}$ in the identifiable matrix is constructed and inserted into the IDIT based on the DM.
(2) For the second non-empty identifiable attribute set $\{a, b, c, e\}$, because the path $(abce)$ corresponding to $\{a, b, c, e\}$ already exists in the IDIT based on the distinguishable matrix, no new path is constructed. Similarly, all identical distinguishable attribute sets are mapped to the same path.
(3) For distinguishable attribute set $\{a, c\}$, its corresponding path (ac) is created, which has the same prefix (a) as path $(abce)$.
(4) For distinguishable attribute set $\{d\}$, the number of elements in $\{d\}$ is 1, so $\text{Core}(A) = d$. Then the path (d) is created.
(5) For distinguishable attribute set $A = \{a, b, c, d, e\}$, because $d \subseteq \{a, b, c, d, e\}$, path $(abcde)$ is not created.
(6) For distinguishable attribute set $A = \{a, c, d\}$, because $d \subseteq \{a, b, c, d, e\}$, path $(abcde)$ is not created.
(7) For distinguishable attribute set $\{b, e\}$, its corresponding path (b, e) is created.

Fig. 5 shows the IDIT constructed based on Algorithm 2 (Fig. 4) and Table 2.

Compared with Fig. 2, the IDIT based on DM constructed in Fig. 5 completely eliminates the parent set information of the core attribute. Also, in the process of building the tree, all non-empty elements in the identifiable matrix are not required to participate. As in Example 2, when the element d in $\text{Core}(A)$ appears, the discernible attribute sets containing d does not participate in constructing the IDIT.

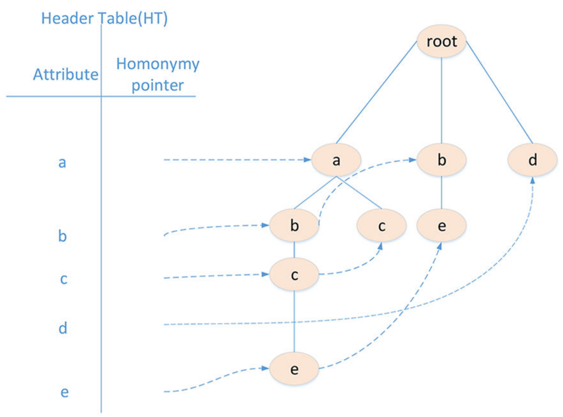

Fig. 5 *IDIT based on the DM in Table 2 in the initial order of attributes*

3.3.2 Construction of IDIT based on attribute significance: As analysed in Section 3.3.1, the IDIT based on core attributes completely eliminates the parent set information of core attributes, but it cannot ensure that the parent set information of other elements is deleted. Also, when we use a heuristic algorithm to get a reduction, we tend to incorporate the more important attributes into the reduction. So next we study the construction method of the IDIT based on attribute significance.

In this study, attribute significance is defined as the frequency of attributes appearing in the DM. Based on this, the corresponding attributes in Table 2 are arranged in descending order of significance as $\mathcal{S} = \langle acdbe \rangle$. Attribute significance indicates how important an attribute is in the information system. The more significance an attribute is, the higher its position in the information system and vice versa. Then, from the process of the

IDIT, it can be known that after adding the attribute importance, the more important the attribute in the IDIT is, the closer it is to the root node. A further implementation realises the compressed storage of the identifiable matrix.

Example 3: Based on Table 2 and Algorithm 3 (see Fig. 6), the specific construction process of the IDIT based on attribute significance is given as follows

(1) The path ($acbe$) corresponding to the first non-empty identifiable attribute set $\{a,c,b,e\}$ in the identifiable matrix is constructed and inserted into the IDIT based on the DM.
(2) For the second non-empty identifiable attribute set $\{a,c,b,e\}$, because the path ($acbe$) corresponding to $\{a,c,b,e\}$ already exists in the IDIT based on the distinguishable matrix, no new path is constructed. Similarly, all identical distinguishable attribute sets are mapped to the same path.
(3) For distinguishable attribute set $\{a,c\}$, use the delete sub-tree strategy to modify the path ($acbe$) to (ac).
(4) For distinguishable attribute set $\{d\}$, the number of elements in $\{d\}$ is 1, so $\mathrm{Core}(A) = d$. Then the path (d) is created.
(5) For distinguishable attribute set $A = \{a,c,d,b,e\}$, because $d \subseteq \{a,b,c,d,e\}$, path ($abcde$) is not created.
(6) For distinguishable attribute set $A = \{a,c,d\}$, because $d \subseteq \{a,b,c,d,e\}$, path ($abcde$) is not created.
(7) For distinguishable attribute set $\{b,e\}$, its corresponding path (b,e) is created.

Fig. 7 shows the construction of IDIT based on Algorithm 3 (see Fig. 6) and Table 2.

Compared with Figs. 2–4, the IDIT in Fig. 7 has the fewest nodes, which greatly realises the compressed storage of the DM.

4 Attribute reduction based on IDIT in interval-valued ordered information system

In this section, in order to verify the attribute reduction algorithm of IDIT proposed in this study. The algorithm is described in detail as shown in Algorithm 4 (see Fig. 8).

Example 4: Based on Fig. 7, the attribute reduction process of the I_vOIS is solved by Algorithm 4 (see Fig. 8) as follows.

(1) Let $R = \emptyset$, $\mathrm{OS} = \{a,c,d,b,e\}$.
(2) Let $R = R \bigcup \{d\} = \{d\}$, $\mathrm{OS} = \mathrm{OS} - \{d\}$. Also, delete the path containing attribute $\{d\}$ from IDIT.
(3) Select the right-most attribute $\{e\}$ in OS and delete the leaf node containing attribute $\{e\}$ through the pointer chain of homonymy pointer. Also, let $\mathrm{OS} = \mathrm{OS} - \{e\}$.
(4) $R = R \bigcup \{b\}$; and the path containing the attribute $\{b\}$ is deleted in the IDIT. At the same time, let $\mathrm{OS} = \mathrm{OS} - \{b\}$.
(5) Select the right-most attribute $\{c\}$ in OS and delete the leaf node containing attribute $\{c\}$ through the pointer chain of homonymy pointer. Also, let $\mathrm{OS} = \mathrm{OS} - \{c\}$.
(6) $R = R \bigcup \{a\}$; and the path containing the attribute $\{a\}$ is deleted in the IDIT. At the same time, let $\mathrm{OS} = \mathrm{OS} - \{a\}$.
(7) $\mathrm{OS} = \emptyset$, and the IDIT only contains the root node. Output $R = \{d,b,a\}$, the algorithm ends.

As analysed in Example 4, the I_vOIS corresponding to Table 1 obtains a reduction $R = \{d,b,a\}$ by the IDIT. In order to verify the correctness of the results, we calculate all the attribute reductions of the I_vOIS by the definition of the minimum disjunction paradigm

Input : An interval-valued ordered information system $I^{\geq} = \{U, AT, F\}$
Output : $IDIT$, $Core(A)$
begin
  Step1 : Let $Core(A)=\emptyset$;
  Step2 : Constructing the root node TN of $IDIT$ based on discernible matrix, marked as "$null$";
  Step3 : Calculating the discernible matrix;
  Step4 : Calculating the frequency of each attribute in the discernible matrix, and arrange the attribute order in descending order of frequency. The order of the elements in each identifiable attribute set in the identifiable matrix is rearranged.
  Step5 : The rearranged identifiable attribute set is inserted into the tree according to the method of **Algorithm2**.
end

Fig. 6 *Algorithm 3: construction of IDIT based on attribute significances*

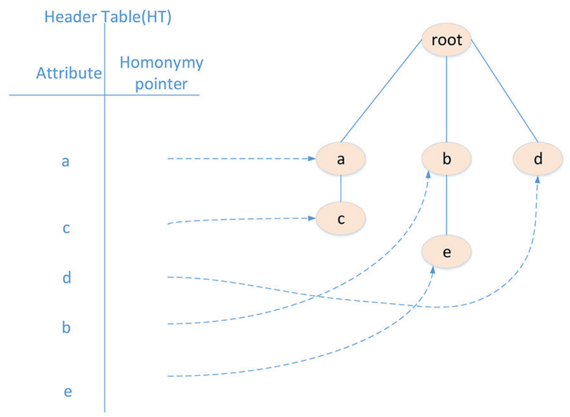

Fig. 7 *IDIT based on attribute significances*

Input : $IDIT$; $Core(A)$
Output : An attribute reduction of I_vOIS
begin
  Step1 : Let $R=\emptyset$; Get the header table HT of the $IDIT$ and make OS be an ordered sequence set of attributes in HT from top to bottom;
  Step2 :
  if $Core(A) \neq \emptyset$ **then**
    For any $a \in Core(A)$, delete all paths containing attribute a in the $IDIT$;
  end
  Step3 : Let $R \leftarrow Core(A)$, $OS \leftarrow OS - Core(A)$;
  Step4 :
  while $OS \neq \emptyset \wedge IDIT \neq \emptyset$ **do**
    A. Select the rightmost attribute a_i in the OS and let $OS \leftarrow OS - \{a_i\}$.
    B.
    if $HT[a_i] \neq \emptyset \wedge a_i \notin R$ **then**
      According to the homonymous pointer corresponding to header $HT[a_i]$, searching homonymous pointer constitutes a pointer chain. In the search process, if the node in the chain is a leaf node, the leaf node is deleted in the $IDIT$.
    end
    C. The attribute corresponding to the path of only one node in the $IDIT$ is added to R, and the path containing the attributes is deleted in the $IDIT$. These attributes (the attributes incorporated into R) are also removed in the OS.
  end
  return : R
end

Fig. 8 *Algorithm 4: attribute reduction algorithm based on IDIT in interval-valued ordered information system*

$$
\begin{aligned}
\mathrm{Red}_{\geq \min} &= (a \vee b \vee c \vee e) \wedge (a \vee c) \wedge d \\
&\quad \wedge (a \vee b \vee c \vee d \vee e) \wedge (a \vee c \vee d) \wedge (b \vee e) \\
&= (b \vee e) \wedge d \wedge (a \vee c) \\
&= (b \wedge d \wedge a) \vee (e \wedge d \wedge a) \vee (b \wedge d \wedge c) \\
&\quad \vee (e \wedge d \wedge c)
\end{aligned}
$$

From the result, it can be seen that the attribute reduction obtained by the IDIT is correct. However, it is obvious that only one reduction of the information table can be obtained by the IDIT. Next, we discuss how to solve other reductions of I_vOIS by the IDIT.

As discussed in Section 3.2, when we change the ordering of attributes, we will get a DIT of different structures. In order to achieve all the reductions of the I_vOIS by the IDIT, we construct the IDIT of Table 2 by Algorithm 2 (see Fig. 4) under different attribute sequences. The results are shown in Fig. 9.

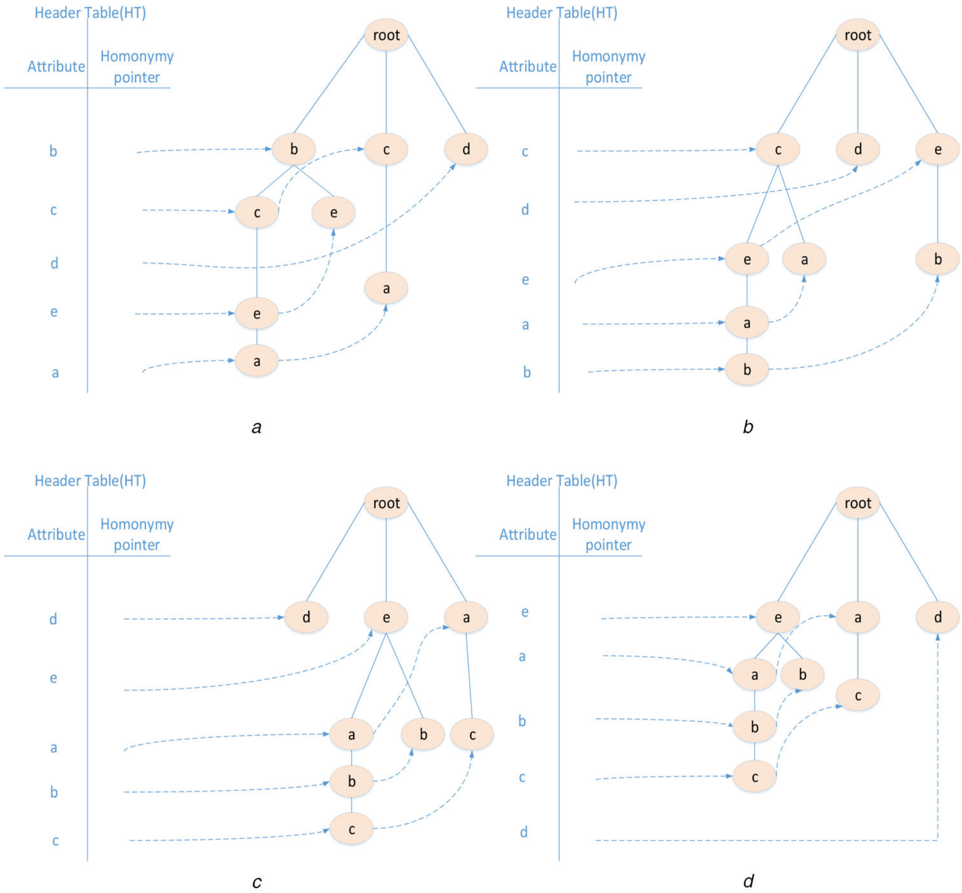

Fig. 9 *IDIT based on the DM in Table 2 in the other order of attributes*

Compared with Fig. 3, the number of nodes in Figs. 4*a* and *b* has changed from 13 to 8. Furthermore, the validity of our proposed IDIT is illustrated. Through Algorithm 4 (see Fig. 8), we get the reduction of IDIT in Fig. 9. Reduction R is $\{d, c, b\}$ under sequence $\mathscr{S} = \langle bcdea \rangle$, R is $\{d, e, c\}$ under sequence $\mathscr{S} = \langle cdeab \rangle$, R is $\{d, a, e\}$ under sequence $\mathscr{S} = \langle deabc \rangle$ and R is $\{d, a, e\}$ under sequence $\mathscr{S} = \langle eabcd \rangle$. In summary, we find all reductions of I_vOIS by the IDIT.

5 Conclusions

In this study, under the background of the interval value order information system, based on the DIT based on the discernibility matrix, the IDIT based on the discernibility matrix is constructed by combining the importance of attributes and core attributes. Furthermore, the compressed storage of the DM is realised. Moreover, by changing the ranking order of conditional attributes in the interval value order information system, the goal of finding all reductions in the information system by the IDIT is achieved. However, the IDIT in this study only takes the frequency of attributes as a measure, without considering the impact of other attribute importance on the construction of the IDIT. Therefore, our next step is to introduce different attribute importance, discuss its role in the IDIT in the interval-valued order information system, and whether it can compress and store the DM further.

6 Acknowledgments

This study is supported by the National Natural Science Foundation of China (nos. 61772002 and 61976245), the Fundamental Research Funds for the Central Universities (no. XDJK2019B029), and the Science and Technology Research Program of Chongqing Municipal Education Commission (no. KJ1709221). Graduate Student Innovation Program of Chongqing University of Technology (No. ycx20192080). Graduate Student Innovation foundation of Chongqing (No. CYS19323).

7 References

[1] Pawlak, Z.: 'Rough sets', *Int. J. Comput. Inf. Sci.*, 1982, **11**, (5), pp. 341–356
[2] Xu, W.H., Guo, Y.T.: 'Generalized multigranulation double-quantitative decision-theoretic rough set', *Knowl.-Based Syst.*, 2016, **105**, pp. 190–205
[3] Dai, J.H., Gao, S.C., Zheng, G.J.: 'Generalized rough set models determined by multiple neighborhoods generated from a similarity relation', *Soft Comput.*, 2018, **22**, (7), pp. 2081–2094
[4] Xu, T.H., Wang, G.Y.: 'Finding strongly connected components of simple digraphs based on generalized rough sets theory', *Knowl.-Based Syst.*, 2018, **149**, pp. 88–98
[5] Macparthalain, N., Shen, Q.: 'On rough sets, their recent extensions and applications', *Knowl. Eng. Rev.*, 2010, **25**, (4), pp. 365–395
[6] Wang, C.Z., Qi, Y.L., Shao, M.W.*, et al.*: 'A fitting model for feature selection with fuzzy rough sets', *IEEE Trans. Fuzzy Syst.*, 2017, **25**, (4), pp. 741–753
[7] Sang, B.B., Guo, Y.T., Shi, D.R.*, et al.*: 'Decision-theoretic rough set model of multi-source decision systems', *Int. J. Mach. Learn. Cybern.*, 2018, **9**, (1), pp. 1941–1954
[8] Mrzek, A.: 'Rough sets and dependency analysis among attributes in computer implementations of expert's inference models', *Int. J. Man-Mach. Stud.*, 1989, **30**, (4), pp. 457–473
[9] Xu, F.F., Miao, D.Q., Wei, L.: 'Fuzzy-rough attribute reduction via mutual information with an application to cancer classification', *Comput. Math. Appl.*, 2009, **57**, (6), pp. 1010–1017
[10] Shen, L.X., Tay, F.E. H., Qu, L.S.*, et al.*: 'Fault diagnosis using rough sets theory', *Comput. Ind.*, 2000, **43**, (1), pp. 61–72
[11] Tay, F.E. H., Shen, L.X.: 'Economic and financial prediction using rough sets model', *Eur. J. Oper. Res.*, 2002, **141**, (3), pp. 641–659
[12] Dai, J.H., Hu, Q.H., Hu, H.*, et al.*: 'Neighbor inconsistent pair selection for attribute reduction by rough set approach', *IEEE Trans. Fuzzy Syst.*, 2018, **26**, (2), pp. 937–950
[13] Jie, H., Li, T.R., Luo, C.*, et al.*: 'Incremental fuzzy cluster ensemble learning based on rough set theory', *Knowl.-Based Syst.*, 2017, **132**, pp. 144–155, S0950705117302940
[14] Hu, D., Yu, X.C., Wang, J.Y.: 'Statistical inference in rough set theory based on Kolmogorov-Smirnov goodness-of-fit test', *IEEE Trans. Fuzzy Syst.*, 2017, **25**, (4), pp. 799–812
[15] Xu, W.H., Li, W.T.: 'Granular computing approach to two-way learning based on formal concept analysis in fuzzy datasets', *IEEE Trans. Cybern.*, 2016, **46**, (2), pp. 366–379
[16] Zhu, W.: 'Generalized rough sets based on relations', *Inf. Sci.*, 2007, **177**, (22), pp. 4997–5011
[17] Yao, Y.Y., Zhao, Y.: 'Attribute reduction in decision-theoretic rough set models', *Inf. Sci.*, 2008, **178**, (17), pp. 3356–3373
[18] Fan, M., He, H.P., Qian, Y.H.*, et al.*: 'Test-cost-sensitive attribute reduction', *Inf. Sci. Int. J.*, 2011, **181**, (22), pp. 4928–4942

[19] Sang, B.B., Zhang, X.Y., Xu, W.H.: 'Attribute reduction of relative knowledge granularity in intuitionistic fuzzy ordered decision table', *FILOMAT*, 2018, **32**, (5), pp. 1727–1736

[20] Xu, W.H., Li, M.M., Wang, X.Z.: 'Information fusion based on information entropy in fuzzy multi-source incomplete information system', *Int. J. Fuzzy Syst.*, 2016, **19**, (4), pp. 1200–1216

[21] Wang, G.Y., Zhao, J., An, J.J., *et al.*: 'A comparative study of algebra viewpoint and information viewpoint in attribute reduction', *Fundam. Inf.*, 2005, **68**, (3), pp. 289–301

[22] Zhang-Yan, X.U., Yang, B.R., Song, W., *et al.*: 'Comparative research of different attribute reduction definitions', *J. Chi. Comput. Syst.*, 2008, **29**, (5), pp. 848–853

[23] Hu, Q.H., Zhao, H., Xie, Z.X., *et al.*: 'Consistency based attribute reduction', *Adv. Knowl. Discov. Data Mining*, 2007, **4426**, pp. 96–107

[24] Parthalain, N.M., Shen, Q.: 'Exploring the boundary region of tolerance rough sets for feature selection', *Pattern Recognit.*, 2009, **42**, (5), pp. 655–667

[25] Parthalain, N.M., Shen, Q., Jensen, R.: 'A distance measure approach to exploring the rough set boundary region for attribute reduction', *IEEE Trans. Knowl. Data Eng.*, 2010, **22**, (3), pp. 305–317

[26] Xu, W.H., Yu, J.H.: 'A novel approach to information fusion in multi-source datasets: a granular computing viewpoint', *Inf. Sci.*, 2017, **378**, pp. 410–423

[27] Skowron, A., Rauszer, C.: 'The discernibility matrices and functions in information systems', in '*Intelligent decision support, handbook of applications and advances of the rough sets theory*' (Springer, Dordrecht, Poland, 1991), pp. 331–362

[28] Konecny, J.: 'On attribute reduction in concept lattices: methods based on discernibility matrix are outperformed by basic clarification and reduction', *Inf. Sci.*, 2017, **415**, pp. 199–212

[29] Yao, Y.Y., Zhao, Y.: 'Discernibility matrix simplification for constructing attribute reducts', *Inf. Sci.*, 2009, **179**, (7), pp. 867–882

[30] Dai, J.H., Hu, H., Wu, W.Z., *et al.*: 'Maximal discernibility pairs based approach to attribute reduction in fuzzy rough sets', *IEEE Trans. Fuzzy Syst.*, 2018, **26**, (4), pp. 2174–2187

[31] Liang, J.K., Zhang, Y., Qu, Y.B.: 'A heuristic algorithm of attribute reduction in rough set', *J. Xian Univ. Post Telecommun.*, 2002, **36**, (4), pp. 478–481

[32] Du, W.S., Hu, B.Q.: 'Approximate distribution reducts in inconsistent interval-valued ordered decision tables', *Inf. Sci.*, 2014, **271**, (7), pp. 93–114

[33] Zhang, W.X., Wei, L., Qi, J.J.: 'Attribute reduction in concept lattice based on discernibility matrix', Int. Workshop on Rough Sets Fuzzy Sets Data Mining Granular Computing, Regina, SK, Canada, 31 August–3 September 2005, **3642**, pp. 157–165

[34] Yang, M., Yang, P.: 'A novel condensing tree structure for rough set feature selection', *Neurocomputing*, 2008, **71**, (4–6), pp. 1092–1100

[35] Yang, M., Yang, P.: 'A novel approach to improving C-tree for feature selection. *Appl. Soft Comput.*, 2011, **11**, (2), pp. 1924–1931

[36] Yang, M., Jing, L.V.: 'An incremental updating algorithm for attribute reduction based on C-tree', *Control Decis.*, 2012, **27**, (12), pp. 1769–1775

[37] Jiang, Y.: 'Attribute reduction with rough set based on discernibility information tree', *Control Decis.*, 2015, **30**, (8), pp. 1531–1536

[38] Jiang, Y., Yu, Y.: 'Minimal attribute reduction with rough set based on compactness discernibility information tree', *Soft Comput.*, 2016, **20**, (6), pp. 2233–2243

[39] Yang, L., Zhang, X.Y., Xu, W.H.: 'Attribute reduction of discernibility information tree in interval-valued ordered information system', *Comput. Sci. Technol.*, 2019, **13**, (6), pp. 1062–1069, doi: 10.3778/j.issn.1673-9418.1805037

[40] Xu, W.H.: '*Order information system and rough set*' (Science Press, Beijing, 2013)