Information Sciences 541 (2020) 475–501

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Incremental approaches for heterogeneous feature selection in dynamic ordered data

Binbin Sang [a,b], Hongmei Chen [a,b,*], Tianrui Li [a,b], Weihua Xu [c], Hong Yu [d]

[a] Institute of Artificial Intelligence, School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, PR China
[b] National Engineering Laboratory of Integrated Transportation Big Data Application Technology, Southwest Jiaotong University, Chengdu 611756, PR China
[c] School of Artificial Intelligence, Southwest University, Chongqing 400715, PR China
[d] Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing 400065, PR China

ARTICLE INFO

Article history:
Received 14 February 2020
Received in revised form 23 June 2020
Accepted 25 June 2020
Available online 9 July 2020

Keywords:
Heterogeneous ordered decision system
Dominance-based neighborhood rough set
Feature selection
Matrix-based incremental algorithm

ABSTRACT

Feature selection can identify essential features and reduce the dimensionality of features, improving the classification ability of a learning model. In this study, we consider data with a preference-order relation, i.e., ordered data. In the big data era, ordered data contain noise and exhibit heterogeneous features (including numerical and categorical features) and dynamic characteristics (i.e., new objects are added and obsolete objects are removed with evolving time). The dominance-based neighborhood rough set (DNRS) considers the preference order relation of heterogeneous features and demonstrates fault tolerance; thus, it can be applied well to heterogeneous feature selection in ordered data. At present, DNRS-based heterogeneous feature selection methods are only used for static ordered data. For dynamic ordered data, existing heterogeneous feature selection approaches are highly time-consuming because they are required to recalculate knowledge from scratch when multiple objects vary. Motivated by this issue, we utilize a matrix-based method in this work to study incremental heterogeneous feature selection based on DNRS in dynamic ordered data. First, we define neighborhood dominance conditional entropy (NDCE) as the uncertainty measure and introduce a non-monotonic feature selection strategy based on this measure. Second, the neighborhood dominance relation matrix and its diagonal matrix are defined to calculate NDCE in matrix form. Third, the updating mechanisms of the diagonal matrix are studied when objects vary and used to update NDCE. Lastly, two incremental feature selection algorithms are proposed when multiple objects are added to or deleted from heterogeneous ordered data. Experiments are performed on public data sets. Experimental results verify that the proposed incremental algorithms are effective and efficient for updating feature subsets in dynamic heterogeneous ordered data.

1. Introduction

Rough set theory (RST), which was proposed by Pawlak in 1982, is an effective mathematical tool for classification learning in uncertain and incomplete data [22]. Objects with the same description based on equivalence relations compose basic granules of knowledge, which are used to describe the concepts approximately. Thus, knowledge granules are the basic

* Corresponding author.
E-mail address: hmchen@swjtu.edu.cn (H. Chen).

https://doi.org/10.1016/j.ins.2020.06.051

concept of RST. To investigate the general properties of knowledge granules, Polkowski et al. proposed two research schemes: logical approximate reasoning [27,23] and rough neural computing [24]. Furthermore, Polkowski et al. developed a systematic study of granule within rough mereology [25,26]. The continuous improvements in granulation technology in recent years, such as the relation among different granular mechanisms [6,45], multi-granulation decision-theoretic rough sets [14], a lattice model based on knowledge distance [31], and grouping granular structures [28], have promoted the development of RST [33,29]. The dominance-based rough set approach (DRSA) proposed by Greco et al. is an important extension of RST that has been successfully applied to multi-criteria decision-making (MCDM) [11]. The strict partial order relation is implemented between attribute values in DRSA; thus, this model is not fault-tolerant when processing MCDM with numerical data. The primary reason for this defect is that a minor fluctuation of an attribute value in DRSA may change the relations between objects due to an error, causing basic knowledge granules to change, and ultimately enabling decision makers to obtain wrong decision information. Such errors are inevitable during data collection, particularly for numerical data. To overcome this deficiency, Chen et al. introduced the idea of neighborhood into DRSA and proposed the dominance-based neighborhood rough set (DNRS) [4]. In DNRS, neighborhood dominance relation considers the partial order of numerical and categorical data and the degree of preference between them. This is, DNRS qualitatively and quantitatively considers the partial order relation of heterogeneous data. Hence, DNRS is more suitable for multi-criteria decision analysis. In particular, DNRS-based feature selection is effective for heterogeneous ordered data. Feature selection approaches suitable for dynamic heterogeneous ordered data sets have not been studied at present. Accordingly, the current work focuses on an incremental heterogeneous feature selection approach based on DNRS for dynamic ordered data.

In the field of data mining, uncertainty measures are important evaluating tools that can quantify data inconsistency. As a common uncertainty measure, information entropy has been widely used in feature selection algorithms. Related studies have been extended since Shannon proposed information entropy [32]. Liang et al. presented complementary and combination entropies based on RST [30]. Hu et al. studied the rank entropies based on DRSA [12]. However, existing information entropy measures are unsuitable for the classification of ordered heterogeneous data sets because they do not simultaneously consider partial order relation and degree of preference among samples. To address this deficiency, we propose NDCE as the uncertainty measure of the feature selection algorithm in our research. Moreover, given that the matrix form of information can simplify the calculation process and intuitively represent the construction of a method, matrix-based computing technology is widely used in incremental learning. For example, the researchers in [16,40] used matrix-based methods to study incremental learning approaches. Considering the advantages of matrix form, the current work also utilizes this form to study the incremental mechanism of feature selection.

As a common data preprocessing technology, feature selection has elicited widespread attention. This technique can delete redundant features and achieve the objectives of reducing dimensionality and improving classification accuracy. RST-based feature selection (also called attribute reduction) methods have been extensively studied in the past decades [7,37,8,20,49]. With the diversification of data types, heterogeneous (mixed) feature selection methods has become a popular research topic [13,47,50,2]. However, the partial order relation of heterogeneous features is not considered in these previous studies. Chen et al. proposed a DNRS-based feature selection method that can efficiently complete heterogeneous feature selection tasks in a given ordered data (i.e., static ordered data) [4]. Technologies for gathering, storing, and processing information continues to evolve with the development of the information age. In general, heterogeneous ordered data sets dynamically evolve over time. For example, students' grades are typical heterogeneous ordered data that include numerical (e.g. score) and categorical (e.g. grade A, B, and C) data. With the graduation and enrollment of students, this data exhibit dynamic characteristics, making it dynamic heterogeneous ordered data. However, existing DNRS-based heterogeneous feature selection methods for such dynamic data sets are required to recalculate existing knowledge. Such process is time-consuming or even infeasible. Therefore, an effective incremental feature selection approach is urgently necessary to efficiently obtain a new feature subset from dynamic heterogeneous ordered data.

As a technique for quickly acquiring knowledge from dynamic data, incremental learning has been widely studied by scholars [5,46,42,15]. Researchers have proposed many incremental learning algorithms in the past decade; among which, incremental feature selection algorithms are among the important representatives. Numerous incremental feature selection algorithms are available, and they can be roughly divided into three categories: object-oriented, feature-oriented, and feature value-oriented varieties. (1) Object-oriented varieties. On the basis of information entropy, Liang et al. developed an incremental updating feature subset approach when adding a group sample [19]. Chen et al. extended two incremental attribute reduction algorithms based on variable precision rough set model and fuzzy rough set model, respectively [3,44]. Yang et al. studied an incremental feature selection approach based on an active sample selection principle [43]. Shu et al. introduced dependency-based incremental updating methods to derive a new reduct [34]. Jing et al. developed a knowledge granularity-based incremental feature selection method with a multi-granulation view [17]. (2) Feature-oriented varieties. Wang et al. proposed an effective attribute reduction algorithm based on information entropy for data sets with dynamically increasing attributes [39]. In dynamic covering decision information systems, Lang et al. studied incremental algorithms based on related families [18]. Zeng et al. investigated a fuzzy rough set-based incremental attribute reduction method on hybrid data [48]. (3) Feature value-oriented varieties. Wang et al. proposed an effective feature selection algorithm by using three representative entropies [38]. Wei et al. presented a dynamic feature selection approach by using discernibility matrix [41]. Cai et al. developed two incremental attribute reduction approaches for coarsening and refining covering granularity [1]. Shu et al. proposed RST-based incremental feature selection methods for a dynamic incomplete information system [35]. Notably, the aforementioned incremental feature selection methods are only applicable to dynamic data sets with equivalence or similarity relation. By contrast, these methods are evidently ineffective for dynamic heterogeneous data sets with partial order relation.

As indicated above, no research about DNRS-based incremental approaches for heterogeneous feature selection has been reported, motivating the current study. Accordingly, we investigate DNRS-based incremental heterogeneous feature selection approaches in an ordered decision system with the variation of multiple objects. The major contributions of this study are as follows. (i) We propose NDCE and then introduce a non-monotonic feature selection method based on it. (ii) We present the definitions of the neighborhood dominance relation matrix and its diagonal matrix and then propose a method for calculating NDCE in matrix form. (iii) The updating mechanisms of matrix-based NDCE are investigated when multiple objects are added to or deleted from a heterogeneous ordered decision system. On this basis, two incremental algorithms for feature selection are further designed. (iv) The experimental comparison with a heuristic algorithm indicates that the proposed algorithms can efficiently obtain a feature subset with a comparable classification effect.

The remainder of this paper is organized as follows. Section 2 reviews related studies. In Section 3, NDCE is defined and NDCE-based non-monotonic feature selection method is proposed. Then, the neighborhood dominance relation matrix and its diagonal matrix are defined, several related corollaries are presented and proved, and a matrix-based heuristic feature selection algorithm is introduced. The updating mechanisms of the matrix-based NDCE are proposed in Section 4. In Section 5, two incremental feature selection algorithms are developed. The effectiveness and efficiency of the developed incremental algorithms are experimentally demonstrated in Section 6. The final section concludes the study and outlines future research.

2. Preliminaries

In this section, we briefly review relevant knowledge in DNRS [4].

2.1. Heterogeneous ordered decision system and its normalization

A heterogeneous decision system is a 4-tuple $H = (U, C \cup \{d\}, V, f)$, where $U = \{x_1, x_2, \ldots, x_n\}$ is a non-empty finite set of objects; $C \cup \{d\}$ is a non-empty finite set of features, with $C = C^{nu} \cup C^{ca}$ denoting a conditional feature set, C^{nu} denoting a numerical feature set, C^{ca} denoting a categorical feature set, d denoting a decision feature, and $C \cap d = \varnothing$; $V = \bigcup_{a_k \in C \cup \{d\}} V_{a_k}, V_{a_k}$ is the domain of feature a_k; $f : U \times (C \cup \{d\}) \rightarrow V$ is the information function with $f(x_i, a_k) \in V_{a_k}$, and $f(x_i, a_k)$ is the feature value of x_i under a_k, which is also denoted by v_{ik}.

In a heterogeneous decision system, if the domain of a feature is arranged in accordance with an increasing or decreasing preference, then the feature is a criterion. A heterogeneous decision system is called a heterogeneous ordered decision system (HODS) if all the features are criteria. We denote a HODS as $H^{\succeq} = (U, C \cup \{d\}, V, f)$.

In HODS, we need to normalize the data values to [0,1] to calculate the distance between objects. In the following paragraphs, we introduce the normalization of numerical and categorical data in HODS.

- Normalization of numerical data. Given a HODS, for any $a_k \in C^{nu}$ and $x_i \in U, min(V_{a_k})$ denotes the minimum value in $V_{a_k}, max(V_{a_k})$ denotes the maximum value in V_{a_k}, and the normalized value of v_{ik} is defined as

$$\hat{v}_{ik} = \frac{v_{ik} - \lambda_1 min(V_{a_k})}{\lambda_2 max(V_{a_k}) - \lambda_1 min(V_{a_k})}. \tag{1}$$

- Normalization of categorical data. Given a HODS, for any $a_l \in C^{ca}$, the feature value range of a_l is $V_{a_l} = \{v_{l_1}, v_{l_2}, \ldots, v_{l_r}, \ldots, v_{l_q}\}$, where $v_{l_1} \prec v_{l_2} \prec \ldots \prec v_{l_r} \ldots \prec v_{l_q}$, the normalized value of v_{l_r} is defined as

$$\hat{v}_{l_r} = \frac{r - \lambda_1}{\lambda_2 |V_{a_l}| - \lambda_1}, \tag{2}$$

  where $0 < \lambda_1 < 1, \lambda_2 > 1$.

  In practical applications, the value of a criterion is typically within a certain range, such as physical examination indicators, students test scores, and risk level assessments. Therefore, when multiple objects are added to a HODS, we normalize the added object set by using the $min(V_{a_l}), max(V_{a_l})$, and $|V_{a_l}|$ of the original HODS as fixed values. When multiple objects are deleted from the HODS, we delete the objects from the original normalized HODS without renormalizing the HODS after the objects are deleted. To avoid further errors, we set the corresponding parameters λ_1 and λ_2 to reduce $min(V_{a_l})$ and expand $max(V_{a_l}), |V_{a_l}|$. In summary, the $\lambda_1 min(V_{a_l}), \lambda_2 max(V_{a_l})$, and $\lambda_2 |V_{a_l}|$ of the original HODS are used as fixed values to normalize the changed HODS. Normalized distance between objects. Given a HODS, for any $x_i, x_j \in U, P \subseteq C$, the normalized distance between x_i and x_j under P is defined as

$$\hat{d}_P(x_i, x_j) = \sqrt[q]{\sum_{k=1}^{|P|} |\hat{f}(x_i, a_k) - \hat{f}(x_j, a_k)|^q}, \tag{3}$$

where (1) it is called the Manhattan distance if $q = 1$ and (2) the Euclidean distance if $q = 2$. Furthermore, the Chebychev distance is defined as $\hat{d}_P(x_i, x_j) = max_{a_k \in P}\left(|\hat{f}(x_i, a_k) - \hat{f}(x_j, a_k)|\right)$.

2.2. Approximations in DNRS

Given a HODS $H^{\succeq} = (U, C \cup \{d\}, V, f)$, for any $P \subseteq C$ and $P \neq \emptyset$, the neighborhood dominance relation $N^{\succeq}_{P_\delta}$ [4] on P is defined as

$$N^{\succeq}_{P_\delta} = \left\{(x, y) \in U \times U | \hat{d}_P(x, y) \geqslant \delta \wedge \hat{f}(x, a) \geqslant \hat{f}(y, a), \forall a \in P\right\}, \tag{4}$$

where $0 \leqslant \delta \leqslant 1$. Note that objects in the HODS follow the traditional dominance relation under decision feature d, which is denoted as

$$N^{\succeq}_d = \{(x, y) \in U \times U | f(x, d) \geqslant f(y, d)\}, \tag{5}$$

and the neighborhood dominance relation under feature set $P \cup d$ is denoted as

$$N^{\succeq}_{P_\delta \cup d} = \left\{(x, y) \in U \times U | \hat{d}_P(x, y) \geqslant \delta \wedge \hat{f}(x, a) \geqslant \hat{f}(y, a) \wedge f(x, d) \geqslant f(y, d), \forall a \in P\right\}. \tag{6}$$

From Eq. (4), the neighborhood dominance relation $N^{\succeq}_{P_\delta}$ degenerates into the traditional dominance relation when $\delta = 0$, i.e., $N^{\succeq}_{P_\delta} = \left\{(x, y) \in U \times U | \hat{f}(x, a) \geqslant \hat{f}(y, a), \forall a \in P\right\}$, which is reflexive, antisymmetric, and transitive. The neighborhood dominance relation $N^{\succeq}_{P_\delta}$ is anti-reflexive, antisymmetric, and transitive when $0 < \delta \leqslant 1$. Hence, the neighborhood dominance relation is a generalized dominance relation. In this study, we set $0 < \delta \leqslant 1$ when calculating the neighborhood dominance relation between objects.

Knowledge granules are generated via binary relation. In DNRS, knowledge granules are called the neighborhood dominating and dominated sets [4], which are defined respectively as follows

$$N^+_{P_\delta}(x) = \left\{y \in U | y N^{\succeq}_{P_\delta} x\right\}; \tag{7}$$

$$N^-_{P_\delta}(x) = \left\{y \in U | x N^{\succeq}_{P_\delta} y\right\}. \tag{8}$$

Analogously, the dominating and dominated sets of x induced by d and $P \cup d$ are denoted respectively as $N^+_d(x) = \{y \in U | y N^{\succeq}_d x\}, N^+_{P_\delta \cup d}(x) = \left\{y \in U | y N^{\succeq}_{P_\delta \cup d} x\right\}$ and $N^-_d(x) = \{y \in U | x N^{\succeq}_d y\}, N^-_{P_\delta \cup d}(x) = \left\{y \in U | x N^{\succeq}_{P_\delta \cup d} y\right\}$.

The neighborhood dominating and dominated sets are illustrated in Fig. 1, where the conditional feature set is $P = \{a_1, a_2\}$. In Fig. 1, the x-coordinate and y-coordinate are the domains of features a_1 and a_2, respectively. Points denote objects in the universe, where x_k is an arbitrary object and δ is distance. In Fig. 1, objects in the universe are divided into five groups: G_1, G_2, G_3, G_4, and G_5. From Fig. 1, we can clearly see that the neighborhood dominating and dominated sets of x_k under feature set P are G_2 and G_4, respectively, i.e., $N^+_{P_\delta}(x_k) = G_2$ and $N^-_{P_\delta}(x_k) = G_4$.

Property 1. For neighborhood dominating and dominated sets, the following properties hold.

(1) Given $P \subseteq C$, if $0 < \delta_1 < \delta_2 \leqslant 1$, then $N^+_{P_{\delta_2}}(x) \subseteq N^+_{P_{\delta_1}}(x)$, $N^-_{P_{\delta_2}}(x) \subseteq N^-_{P_{\delta_1}}(x)$, $N^+_{P_{\delta_2} \cup d}(x) \subseteq N^+_{P_{\delta_1} \cup d}(x)$, and $N^-_{P_{\delta_2} \cup d}(x) \subseteq N^-_{P_{\delta_1} \cup d}(x)$;
(2) Given $0 < \delta \leqslant 1$, for any $P \subseteq C$, $N^+_{P_\delta}(x) \cap N^+_d(x) = N^+_{P_\delta \cup d}(x)$ and $N^-_{P_\delta}(x) \cap N^-_d(x) = N^-_{P_\delta \cup d}(x)$.

The Property 1 (1) introduces the monotonicity characteristic between the neighborhood dominating/dominated set and δ, which indicates that the cardinality of the neighborhood dominating/dominated set decreases as δ increases. The Property 1 (2) shows the relation between the neighborhood dominating/dominated set induced independently by different feature subsets and that induced by a combination of these feature subsets. The dominating/dominated set and the relation between the dominating/dominated set induced by different feature subsets are important elements in calculating NDCE. The aforementioned properties lay the foundation for further discussing the properties of NDCE in Proposition 1.

Given that d is a categorical feature, U can be divided into a family of equivalent classes by d, denoted as $Cl = \{Cl_t, t \in T\}$, where $T = \{1, 2, \ldots, |V_d|\}$. The decision classes are also preference-ordered, i.e., $\forall r, s \in T$, if $r > s$, then $\forall x \in Cl_r$ is preferred over $\forall y \in Cl_s$. In DNRS, the sets to be approximated are upward and downward unions [11], which are denoted as $Cl^{\succeq}_t = \bigcup_{t' \geqslant t} Cl_{t'}; Cl^{\preceq}_t = \bigcup_{t' \leqslant t} Cl_{t'} (\forall t, t' \in T)$. The statement $x \in Cl^{\succeq}_t$ indicates that x belongs to at least class Cl_t, whereas $x \in Cl^{\preceq}_t$ indicates that x belongs to at most class Cl_t.

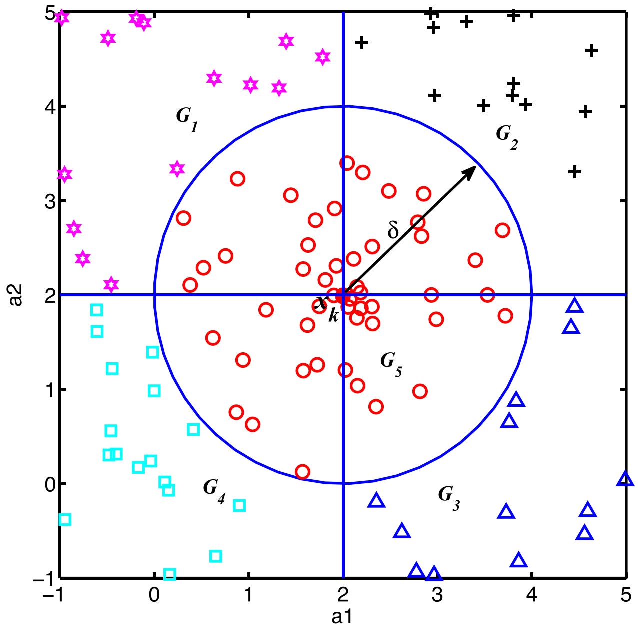

Fig. 1. The neighborhood dominating set and neighborhood dominated set.

Given a HODS $H^{\succeq} = (U, C \cup \{d\}, V, f)$, for any $P \subseteq C$, the lower and upper approximations of $Cl_t^{\succeq} (t \in T)$ are defined respectively as follows

$$\underline{N_{P_\delta}}\left(Cl_t^{\succeq}\right) = \left\{x \in U | N_{P_\delta}^{+}(x) \subseteq Cl_t^{\succeq}\right\}; \quad (9)$$

$$\overline{N_{P_\delta}}\left(Cl_t^{\succeq}\right) = \left\{x \in U | N_{P_\delta}^{+}(x) \cap Cl_t^{\succeq} \neq \varnothing\right\}. \quad (10)$$

The lower and upper approximations of $Cl_t^{\preceq} (t \in T)$ are defined respectively as follows

$$\underline{N_{P_\delta}}\left(Cl_t^{\preceq}\right) = \left\{x \in U | N_{P_\delta}^{-}(x) \subseteq Cl_n^{\preceq}\right\}; \quad (11)$$

$$\overline{N_{P_\delta}}\left(Cl_t^{\preceq}\right) = \left\{x \in U | N_{P_\delta}^{-}(x) \cap Cl_n^{\preceq} \neq \varnothing\right\}. \quad (12)$$

3. Non-monotonic feature selection in a HODS

In data mining, the redundant features in a data set may cause overfitting of the classification and increase computational costs. Feature selection is an efficient method for selecting the necessary features while keeping the discriminative capability of knowledge unchanged, which is a key step in data preprocessing. In this section, we propose a non-monotonic feature selection method based on NDCE. Considering that the matrix form of information can simplify the calculation process and intuitively represent the construction of a method, we present a matrix-based method for calculating NDCE, called matrix-based NDCE (MNDCE). On this basis of this method, we then introduce a matrix-based heuristic feature selection algorithm in a HODS.

3.1. Non-monotonic feature selection based on NDCE

In this subsection, we first define NDCE and prove its non-monotonicity. Then, we design a non-monotonic feature selection method for a HODS. Lastly, the inner and outer significance measures are presented. In [4], the knowledge granules generated via neighborhood dominance relation did not satisfy monotonicity because the authors used the Euclidean distance to construct neighborhood dominance relation. To overcome this deficiency, we define a new distance between objects in a HODS.

Definition 1. Given a HODS $H^{\succeq} = (U, C \cup \{d\}, V, f)$, for any $x_i, x_j \in U, P \subseteq C$, the distance between x_i and x_j under P is defined as

$$\hat{d}_P(x_i, x_j) = min_{a_k \in P} |\hat{f}(x_i, a_k) - \hat{f}(x_j, a_k)|. \quad (13)$$

Subsequently, we prove the monotonicity of the knowledge granules generated via neighborhood dominance relation based on the proposed distance.

Property 2. For any $P, Q \subseteq C$ and $\forall x \in U$, given $\delta \in (0,1]$ the following properties hold.

(1) If $P \subseteq Q$, then $N^+_{Q_\delta}(x) \subseteq N^+_{P_\delta}(x)$ and $N^-_{Q_\delta}(x) \subseteq N^-_{P_\delta}(x)$;
(2) $N^+_{P_\delta}(x) \cap N^+_{Q_\delta}(x) = N^+_{(P\cup Q)_\delta}(x)$ and $N^-_{P_\delta}(x) \cap N^-_{Q_\delta}(x) = N^-_{(P\cup Q)_\delta}(x)$.

Proof. (1) Assuming $\forall y \in N^+_{Q_\delta}(x)$, according to Eq. (4), we have $\hat{d}_Q(x,y) \geqslant \delta$ and $\hat{f}(x,a) \leqslant \hat{f}(y,a)$ $(\forall a \in Q)$. Known $P \subseteq Q$, based on Definition 1, we can determine that $\hat{d}_P(x,y) \geqslant \hat{d}_Q(x,y) \geqslant \delta$ and $\hat{f}(x,a) \leqslant \hat{f}(y,a)$ $(\forall a \in P)$ hold. According to Eq. (4), we have $y \in N^+_{P_\delta}(x)$. Thus, we can obtain $N^+_{Q_\delta}(x) \subseteq N^+_{P_\delta}(x)$. Analogously, the $N^-_{Q_\delta}(x) \subseteq N^-_{P_\delta}(x)$ can be proved. (2) First, we prove $N^+_{P_\delta}(x) \cap N^+_{Q_\delta}(x) \subseteq N^+_{(P\cup Q)_\delta}(x)$. Based on (1), we have $N^+_{P_\delta}(x) \subseteq N^+_{(P\cup Q)_\delta}(x)$ and $N^+_{Q_\delta}(x) \subseteq N^+_{(P\cup Q)_\delta}(x)$. Thus, we can easily determine that $N^+_{P_\delta}(x) \cap N^+_{Q_\delta}(x) \subseteq N^+_{(P\cup Q)_\delta}(x)$ holds. Second, we prove $N^+_{(P\cup Q)_\delta}(x) \subseteq N^+_{P_\delta}(x) \cap N^+_{Q_\delta}(x)$. Assuming $\forall y \in N^+_{(P\cup Q)_\delta}(x)$, according to Eq. (4), we have $\hat{d}_{P\cup Q}(x,y) \geqslant \delta$ and $\hat{f}(x,a) \leqslant \hat{f}(y,a)$ $(\forall a \in P \cup Q)$. Due to $P, Q \subseteq P \cup Q$, based on Definition 1, we can get that $\hat{d}_P(x,y) \geqslant \hat{d}_{P\cup Q}(x,y) \geqslant \delta, \hat{d}_Q(x,y) \geqslant \hat{d}_{P\cup Q}(x,y) \geqslant \delta, \hat{f}(x,a) \leqslant \hat{f}(y,a)$ $(\forall a \in P)$, and $\hat{f}(x,a) \leqslant \hat{f}(y,a)$ $(\forall a \in Q)$ hold. According to Eq. (4), we have $y \in N^+_{P_\delta}(x)$ and $y \in N^+_{Q_\delta}(x)$. Hence, the $N^+_{(P\cup Q)_\delta}(x) \subseteq N^+_{P_\delta}(x) \cap N^+_{Q_\delta}(x)$ holds. In summary, we can determine that $N^+_{P_\delta}(x) \cap N^+_{Q_\delta}(x) = N^+_{(P\cup Q)_\delta}(x)$ holds. Similarly, we also determine $N^-_{P_\delta}(x) \cap N^-_{Q_\delta}(x) = N^{\delta-}_{P\cup Q}(x)$ holds. $\square$

The Property 2 (1) indicates the monotonicity characteristic between the neighborhood dominating/dominated set and feature subsets. The Property 2 (2) presents the relation between the neighborhood dominating/dominated set induced by any two conditional feature subsets and that induced by a combination of the two conditional feature subsets, which will be used to prove Proposition 2.

In [12], ascending/decreasing rank conditional entropies were proposed to measure the ranking consistency of objects under conditional and decision features in an ordered decision system. In the current study, we only consider ascending rank conditional entropy (also called dominance conditional entropy) as the uncertainty measure. However, constructing this measure based on neighborhood dominance relation is inappropriate. Neighborhood dominance relation does not satisfy reflexivity; thus, knowledge granules generated by such relation may be an empty set in the worst case scenario. In such case, the denominator of the measure is zero, which is senseless. To overcome this issue, we propose NDCE based on the ascending rank conditional entropy.

Definition 2. Given a HODS $H^{\succeq} = (U, C \cup \{d\}, V, f)$, for any $P \subseteq C$, NDCE of P to d is defined as

$$NH^{\succeq}_{d|P_\delta}(U) = -\frac{1}{|U|}\sum_{i=1}^{n}\log\frac{|N^+_d(x_i) \cap N^+_{P_\delta}(x_i)| + 1}{|N^+_{P_\delta}(x_i)| + 1} = -\frac{1}{|U|}\sum_{i=1}^{n}\log\frac{|N^+_{d\cup P_\delta}(x_i)| + 1}{|N^+_{P_\delta}(x_i)| + 1}, \tag{14}$$

where $|*|$ represents the number of elements in the set $*$.

From Definition 2, we find that NDCE in a HODS reflects the degree of ranking consistency of object set, which is determined from the information provided by the conditional feature set and the decision feature. The formula $\frac{|N^+_d(x_i) \cap N^+_{P_\delta}(x_i)|+1}{|N^+_{P_\delta}(x_i)|+1}$ essentially determines the degree of ranking consistency. From Eq. (14), we easily determine that the value of $NH^{\succeq}_{d|P_\delta}(U)$ is inversely proportional to the degree of ranking consistency, where $NH^{\succeq}_{d|P_\delta}(U) \geqslant 0$. That is, the smaller the value of $NH^{\succeq}_{d|P_\delta}(U)$, the higher the degree of ranking consistency, also indicating that conditional feature set P provides more accurate ranking information for the object set, and vice versa.

Proposition 1. For any $P \subseteq Q \subseteq C, \forall \delta \in (0,1], NH^{\succeq}_{d|P_\delta}(U) \leqslant NH^{\succeq}_{d|Q_\delta}(U)$ or $NH^{\succeq}_{d|P_\delta}(U) \geqslant NH^{\succeq}_{d|Q_\delta}(U)$ is uncertain, i.e., NDCE dissatisfies monotonicity.

Proof. From Definition 2, we can get that

$$\begin{aligned}\triangle &= NH^{\succeq}_{d|Q_\delta}(U) - NH^{\succeq}_{d|P_\delta}(U)\\ &= -\tfrac{1}{|U|}\sum_{i=1}^{n}\log\tfrac{|N^+_{d\cup Q_\delta}(x_i)|+1}{|N^+_{Q_\delta}(x_i)|+1} + \tfrac{1}{|U|}\sum_{i=1}^{n}\log\tfrac{|N^+_{d\cup P_\delta}(x_i)|+1}{|N^+_{P_\delta}(x_i)|+1}\\ &= \tfrac{1}{|U|}\sum_{i=1}^{n}\left(\log\tfrac{|N^+_{d\cup P_\delta}(x_i)|+1}{|N^+_{P_\delta}(x_i)|+1} - \log\tfrac{|N^+_{d\cup Q_\delta}(x_i)|+1}{|N^+_{Q_\delta}(x_i)|+1}\right).\end{aligned}$$

Assuming that $g_1(x_i) = \frac{|N^+_{d\cup P_\delta}(x_i)|+1}{|N^+_{P_\delta}(x_i)|+1}$ and $g_2(x_i) = \frac{|N^+_{d\cup Q_\delta}(x_i)|+1}{|N^+_{Q_\delta}(x_i)|+1}$. According to Property 1 (2), we can obtain that $N^+_{d\cup P_\delta}(x_i) \subseteq N^+_{P_\delta}(x_i)$ and $N^+_{d\cup Q_\delta}(x_i) \subseteq N^+_{Q_\delta}(x_i)$. Obviously, $0 < g_1(x_i) \leqslant 1$ and $0 < g_2(x_i) \leqslant 1$. Hence, it can be obtained that $\triangle = \frac{1}{|U|}\sum_{i=1}^n(\log g_1(x_i) - \log g_2(x_i)) = \frac{1}{|U|}\sum_{i=1}^n \log\frac{g_1(x_i)}{g_2(x_i)}$. Due to $0 < g_1(x_i), g_2(x_i) \leqslant 1$, then the $\frac{g_1(x_i)}{g_2(x_i)} \geqslant 1$ $\left(\frac{g_1(x_i)}{g_2(x_i)} \leqslant 1\right)$ is uncertain. So the $\triangle \geqslant 0$ ($\triangle \leqslant 0$) is uncertain. Therefore, the NDCE does not satisfy monotonicity. □

Proposition 1 proves that NDCE is non-monotonic, and it is the basis for the definition of the following reduct. In [36], a non-monotonic feature selection method that is applicable to non-monotonic uncertainty measures was proposed. Considering that the proposed uncertainty measure is non-monotonic, we construct a feature selection method based on this measure, called non-monotonic feature selection based on NDCE.

Definition 3. Given a HODS $H^{\succeq} = (U, C \cup \{d\}, V, f)$, for any $B \subseteq C$, the B is a reduct of $H^{\succeq}$ if it satisfies

(1) $NH^{\succeq}_{d|B_\delta}(U) \leqslant NH^{\succeq}_{d|C_\delta}(U)$,
(2) $\forall a \in B, NH^{\succeq}_{d|(B-a)_\delta}(U) > NH^{\succeq}_{d|B_\delta}(U)$.

Condition (1) ensures that the selected feature subset has higher or at least the same ranking consistency power as the whole feature set; the condition (2) ensures that all features in the selected feature subset are indispensable, indicating that we cannot delete any features from the selected feature subset; otherwise, ranking consistency power will decrease. Therefore, the selected feature subset is called a reduct if it satisfies the two aforementioned conditions. If the selected feature subset only satisfies Condition (1), then it is a relative reduct.

In the feature selection process, the informative features can be obtained through the feature significance measures, which are defined as follows.

Definition 4. Given a HODS $H^{\succeq} = (U, C \cup \{d\}, V, f)$, for any $B \subseteq C$ and $\forall a \in B$, the inner significance of a in B is defined as

$$sig^{\succeq U}_{inner}(a, B, d) = NH^{\succeq}_{d|(B-a)_\delta}(U) - NH^{\succeq}_{d|B_\delta}(U). \tag{15}$$

In addition, the core of the feature set B is represented as $Core_B = \left\{a \in B | sig^{\succeq U}_{inner}(a, B, d) > 0\right\}$.

Definition 5. Given a HODS $H^{\succeq} = (U, C \cup \{d\}, V, f)$, for any $B \subseteq C$ and $\forall a \in (C - B)$, the outer significance of a to B is defined as

$$sig^{\succeq U}_{outer}(a, B, d) = NH^{\succeq}_{d|B_\delta}(U) - NH^{\succeq}_{d|(B\cup a)_\delta}(U). \tag{16}$$

In $H^{\succeq} = (U, C \cup \{d\}, V, f), \forall a \in C$, in accordance with the general heuristic feature selection strategy, we can determine that if $sig^{\succeq U}_{inner}(a, C, d) > 0$, then $a \in Core_C$, i.e., a is an indispensable feature. Then, a reduct can be gained based on $Core_C$ by gradually adding the selected feature subset with the highest outer significance to $Core_C$.

3.2. Matrix-based computation of NDCE

In this section, we first define the neighborhood dominance relation matrix and its diagonal matrix in a HODS. Then, several related properties and corollaries are presented and proved. We also provide an example to explain the proposed method.

Definition 6. Given a HODS $H^{\succeq} = (U, C \cup \{d\}, V, f)$, for any $P \subseteq C$, the neighborhood dominance relation matrix with respect to the neighborhood dominance relation $N^{\succeq}_{P_\delta}$ is defined as $\mathbb{R}^{\succeq P_\delta}_U = \left[r^{P_\delta}_{(i,j)}\right]_{n\times n}$, where

$$r^{P_\delta}_{(i,j)} = \begin{cases} 1, & x_j N^{\succeq}_{P_\delta} x_i; \\ 0, & otherwise. \end{cases} \tag{17}$$

Analogously, the dominance relation matrix $\mathbb{R}^{\succeq d}_U$ is induced by the traditional dominance relation $N^{\succeq}_d$, and the neighborhood dominance relation matrix $\mathbb{R}^{\succeq P_\delta \cup d}_U$ is induced by the neighborhood dominance relation $N^{\succeq}_{P_\delta \cup d}$. Their expressions are similar to Eq. (17). In fact, $\mathbb{R}^{\succeq P_\delta}_U$ is a matrix representation of the neighborhood dominance relation $N^{\succeq}_{P_\delta}$ on U.

Property 3. For $\mathbb{R}^{\succeq P_\delta}_U = \left[r^{P_\delta}_{(i,j)}\right]_{n\times n}$, the following properties hold.

(1) $n = |U|$;
(2) $r^{P_\delta}_{(i,i)} = 0$;

(3) $\sum_{j=1}^{n} r_{(i,j)}^{P_\delta} = |N_{P_\delta}^{+}(x_i)|$ and $\sum_{i=1}^{n} r_{(i,j)}^{P_\delta} = |N_{P_\delta}^{-}(x_j)|$.

Property 3 contains three properties of the neighborhood dominance relation matrix. These properties are used to construct NDCE in matrix form.

Definition 7. Given a HODS $H^{\succeq} = (U, C \cup \{d\}, V, f)$, for any $P, Q \subseteq C$, one can obtain two neighborhood dominance relation matrices $\mathbb{R}_U^{\succeq P_\delta} = \left[r_{(i,j)}^{P_\delta}\right]_{n\times n}$ and $\mathbb{R}_U^{\succeq Q_\delta} = \left[r_{(i,j)}^{Q_\delta}\right]_{n\times n}$, respectively. Then the "$\wedge$" operation between them is defined as

$$\mathbb{R}_U^{\succeq P_\delta} \wedge \mathbb{R}_U^{\succeq Q_\delta} = \left[min\left\{r_{(i,j)}^{P_\delta}, r_{(i,j)}^{Q_\delta}\right\}\right]_{n\times n}. \tag{18}$$

In particular, for any $P \subseteq C$, the "$\wedge$" operation between $\mathbb{R}_U^{\succeq P_\delta}$ and $\mathbb{R}_U^{\succeq d}$ is similarly defined as

$$\mathbb{R}_U^{\succeq P_\delta} \wedge \mathbb{R}_U^{\succeq d} = \left[min\left\{r_{(i,j)}^{P_\delta}, r_{(i,j)}^{d}\right\}\right]_{n\times n}. \tag{19}$$

Proposition 2. For any $P, Q \subseteq C$, the $\mathbb{R}_U^{\succeq (P\cup Q)_\delta} = \mathbb{R}_U^{\succeq P_\delta} \wedge \mathbb{R}_U^{\succeq Q_\delta}$ and $\mathbb{R}_U^{\succeq P_\delta \cup d} = \mathbb{R}_U^{\succeq P_\delta} \wedge \mathbb{R}_U^{\succeq d}$ hold.

Proof. From Definition 6, $\mathbb{R}_U^{\succeq (P\cup Q)_\delta} = \left[r_{(i,j)}^{(P\cup Q)_\delta}\right]_{n\times n}$. If $r_{(i,j)}^{(P\cup Q)_\delta} = 1$, i.e., $x_j \in N_{(P\cup Q)_\delta}^{+}(x_i)$. According to Property 2 (2), we have $x_j \in N_{P_\delta}^{+}(x_i)$ and $x_j \in N_{Q_\delta}^{+}(x_i)$, i.e., $r_{(i,j)}^{P_\delta} = 1$ and $r_{(i,j)}^{Q_\delta} = 1$. So we have $r_{(i,j)}^{(P\cup Q)_\delta} = min\left\{r_{(i,j)}^{P_\delta}, r_{(i,j)}^{Q_\delta}\right\} = 1$, and vice versa. If $r_{(i,j)}^{(P\cup Q)_\delta} = 0$, i.e., $x_j \notin N_{(P\cup Q)_\delta}^{+}(x_i)$, that is, $x_j \notin N_{P_\delta}^{+}(x_i)$ or $x_j \notin N_{Q_\delta}^{+}(x_i)$, i.e., $r_{(i,j)}^{P_\delta} = 0$ or $r_{(i,j)}^{Q_\delta} = 0$. So we can determine that $r_{(i,j)}^{(P\cup Q)_\delta} = min\left\{r_{(i,j)}^{P_\delta}, r_{(i,j)}^{Q_\delta}\right\} = 0$, and vice versa. Thus, we can obtain that $r_{(i,j)}^{(P\cup Q)_\delta} = min\left\{r_{(i,j)}^{P_\delta} \times r_{(i,j)}^{Q_\delta}\right\}$, i.e., $\mathbb{R}_U^{\succeq (P\cup Q)_\delta} = \mathbb{R}_U^{\succeq P_\delta} \wedge \mathbb{R}_U^{\succeq Q_\delta}$ holds. Similarly, we can prove $\mathbb{R}_U^{\succeq P_\delta \cup d} = \mathbb{R}_U^{\succeq P_\delta} \wedge \mathbb{R}_U^{\succeq d}$ holds. □

Proposition 2 proves the feasible of the "$\wedge$" operation between neighborhood dominance relation matrices with respect to two different feature subsets. Subsequently, we will provide the definition of the diagonal matrix of the neighborhood dominance relation matrix.

Definition 8. Given a HODS $H^{\succeq} = (U, C \cup d, V, f)$, for any $P \subseteq C$, the diagonal matrix of the neighborhood dominance relation matrix $\mathbb{R}_U^{\succeq P_\delta} = \left[r_{(i,j)}^{P_\delta}\right]_{n\times n}$ is defined as $\mathbb{D}_U^{\succeq P_\delta} = \left[d_{(i,j)}^{P_\delta}\right]_{n\times n}$, where

$$d_{(i,j)}^{P_\delta} = \begin{cases} \sum_{l=1}^{n} r_{(i,l)}^{P_\delta} + 1, & 1 \leqslant i,j \leqslant n, i = j; \\ 0, & 1 \leqslant i,j \leqslant n, i \neq j. \end{cases} \tag{20}$$

In addition, the determinant of diagonal matrix is expressed as $|\mathbb{D}_U^{\succeq P_\delta}| = \Pi_{i=j=1}^{n} d_{ij}^{P_\delta}$, the inverse matrix of the diagonal matrix is defined as $\left(\mathbb{D}_U^{\succeq P_\delta}\right)^{-1} = \left[\frac{1}{d_{(i,j)}^{P_\delta}}\right]_{n\times n}$, where

$$\frac{1}{d_{(i,j)}^{P_\delta}} = \begin{cases} \frac{1}{\sum\limits_{l=1}^{n} r_{(i,l)}^{P_\delta} + 1}, & 1 \leqslant i,j \leqslant n, i = j; \\ 0, & 1 \leqslant i,j \leqslant n, i \neq j. \end{cases} \tag{21}$$

For any two matrices $\mathbb{E} = \left[e_{(i,j)}\right]_{m\times n}$ and $\mathbb{F} = \left[f_{(i,j)}\right]_{n\times l}$, the multiplication "$\cdot$" operation between the two matrices is denoted as $\mathbb{E} \cdot \mathbb{F} = \left[g_{(i,j)}\right]_{m\times l}$, where $g_{(i,j)} = \sum_{k=1}^{n} e_{(i,k)} \times f_{(k,j)}$, and "$\times$" represents standard multiplication.

Corollary 1. For any $P \subseteq C$, the $\mathbb{D}_U^{\succeq P_\delta}$ and $\mathbb{D}_U^{\succeq P_\delta \cup d}$ are two diagonal matrices with respect to P and $P \cup d$, respectively. Then, NDCE based on matrix calculation is denoted as

$$MNH_{d|P_\delta}^{\succeq}(U) = -\frac{1}{|U|} \log |\mathbb{D}_U^{\succeq P_\delta \cup d} \cdot \left(\mathbb{D}_U^{\succeq P_\delta}\right)^{-1}|. \tag{22}$$

Proof. From Definition 2, we have $NH_{d|P_\delta}^{\succeq}(U) = -\frac{1}{|U|}\sum_{i=1}^{n} \log \frac{|N_{d\cup P_\delta}^{+}(x_i)|+1}{|N_{P_\delta}^{+}(x_i)|+1} = -\frac{1}{|U|} \log \frac{\Pi_{i=1}^{n}\left(|N_{d\cup P_\delta}^{+}(x_i)|+1\right)}{\Pi_{i=1}^{n}\left(|N_{P_\delta}^{+}(x_i)|+1\right)}$. According to Definitions 6 and 8, we can obtain that the diagonal matrices $\mathbb{D}_U^{\succeq P_\delta} = \left[d_{(i,j)}^{P_\delta}\right]_{n\times n}$ and $\mathbb{D}_U^{\succeq P_\delta \cup d} = \left[d_{(i,j)}^{P_\delta \cup d}\right]_{n\times n}$, where $d_{(i,j)}^{P_\delta} = |N_{P_\delta}^{+}(x_i)| + 1$ and

$d^{P_\delta\cup d}_{(i,j)} = |N^+_{P_\delta\cup d}(x_i)| + 1$. Then we can derive $|\mathbb{D}^{\succeq P_\delta\cup d}_U \cdot \left(\mathbb{D}^{\succeq P_\delta}_U\right)^{-1}| = \Pi^n_{i=1}\frac{d^{P_\delta\cup d}_{(i,j)}}{d^{P_\delta}_{(i,j)}} = \frac{\Pi^n_{i=1}d^{P_\delta\cup d}_{ij}}{\Pi^n_{i=1}d^{P_\delta}_{(i,j)}} = \frac{\Pi^n_{i=1}\left(|N^+_{d\cup P_\delta}(x_i)|+1\right)}{\Pi^n_{i=1}\left(|N^+_{P_\delta}(x_i)|+1\right)}$. Thus, we can determine that $NH^{\succeq}_{d|P_\delta}(U)= MNH^{\succeq}_{d|P_\delta}(U)$. In summary, the results of calculating NDCE based on matrix and non-matrix methods are consistent. □

Corollary 1 proposes a calculation method for NDCE in matrix form. Subsequently, an example is used to explain how NDCE is calculated using the matrix method.

Example 1. The data listed at the left in Table 1 is a HODS, where $U = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}, C^{ca} = \{a_1\}, C^{nu} = \{a_2, a_3, a_4\}$, and d is a decision feature. The values of the different criteria are ranked as follows. $V_{a_1}: low \prec mid \prec high \prec vhigh$; $V_{a_2}: 1 \prec \ldots \prec 20$; $V_{a_3}: 0 \prec \ldots \prec 2$; $V_{a_4}: 100 \prec \ldots \prec 500$; $V_d: E \prec D \prec C \prec B \prec A$.

First, we normalize the initial data by Eqs. (1) and (2), where $\lambda_1 = 0.8$ and $\lambda_2 = 1.5$. The normalized result is shown at the right side of Table 1.

Then, according to Definition 6, given $\delta = 0.05$, the neighborhood dominance relation matrix $\mathbb{R}^{\succeq C_\delta}_U$ and the dominance relation matrix $\mathbb{R}^{\succeq d}_U$ are calculated respectively as

$$\mathbb{R}^{\succeq C_\delta}_U = \begin{bmatrix} 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&1&0&0\\ 1&1&0&1&0&1&0&1\\ 0&0&0&0&0&1&0&0\\ 1&1&0&0&0&1&1&1\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&1&0&0 \end{bmatrix}_{8\times8}, \mathbb{R}^{\succeq d}_U = \begin{bmatrix} 1&1&1&1&1&1&0&1\\ 0&1&0&1&0&0&0&0\\ 0&1&1&1&0&1&0&0\\ 0&1&0&1&0&0&0&0\\ 1&1&1&1&1&1&0&1\\ 0&1&1&1&0&1&0&0\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&0&1 \end{bmatrix}_{8\times8}.$$

Taking $\mathbb{R}^{\succeq C_\delta}_U$ as an example, Property 3 is verified as follows.

(1) In the $\mathbb{R}^{\succeq C_\delta}_U = \left[r^{C_\delta}_{(i,j)}\right]_{8\times8}, |U| = 8$;
(2) For any $i \in [1, 8]$ and $i \in N^+, r^{C_\delta}_{(i,i)} = 0$;
(3) For any $i,j \in [1, 8]$ and $i,j \in N^+$, $\sum^8_{j=1} r^{C_\delta}_{(i,j)} = |N^+_{C_\delta}(x_i)|$ and $\sum^8_{i=1} r^{C_\delta}_{(i,j)} = |N^-_{C_\delta}(x_j)|$, e.g., when $i = 3, N^+_{C_\delta}(x_3) = \{x_1, x_2, x_4, x_6, x_8\}$, we have $\sum^8_{j=1} r^{C_\delta}_{(3,j)} = |N^+_{C_\delta}(x_3)| = 5$, when $j = 6, N^-_{C_\delta}(x_6) = \{x_1, x_2, x_3, x_4, x_5, x_7, x_8\}$, we have $\sum^8_{i=1} r^{C_\delta}_{(i,6)} = |N^-_{C_\delta}(x_6)| = 7$.

Next, according to Definition 7, the neighborhood dominance relation matrix $\mathbb{R}^{\succeq C_\delta\cup d}_U$ is calculated as

$$\mathbb{R}^{\succeq C_\delta\cup d}_U = \mathbb{R}^{\succeq C_\delta}_U \wedge \mathbb{R}^{\succeq d}_U = \begin{bmatrix} 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0\\ 0&1&0&1&0&1&0&0\\ 0&0&0&0&0&0&0&0\\ 1&1&0&0&0&1&0&1\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&1&0&0 \end{bmatrix}_{8\times8}.$$

Table 1
An example of initial and normalized HODS.

	The initial data					The normalized data				
U	a_1	a_2	a_3	a_4	d	a_1	a_2	a_3	a_4	d
x_1	high	8	1.10	410	D	0.4231	0.3299	0.4785	0.4301	D
x_2	high	10	0.90	380	B	0.4231	0.4330	0.3558	0.3750	B
x_3	low	2	0.40	280	C	0.0385	0.0206	0.0491	0.1912	C
x_4	high	5	0.72	380	B	0.4231	0.1753	0.2454	0.3750	B
x_5	low	6	0.50	220	D	0.0385	0.2268	0.1104	0.0809	D
x_6	vhigh	14	1.30	480	C	0.6154	0.6392	0.6012	0.5588	C
x_7	mid	12	0.90	300	E	0.2308	0.5361	0.3558	0.2279	E
x_8	high	7	0.94	390	D	0.4231	0.2784	0.3804	0.3934	D

Afterwards, according to Definition 8, the diagonal matrices $\mathbb{D}_U^{\succeq C_\delta}$ and $\mathbb{D}_U^{\succeq C_\delta \cup d}$ are calculated as

$$\mathbb{D}_U^{\succeq C_\delta} = \begin{bmatrix} 2&0&0&0&0&0&0&0\\ 0&2&0&0&0&0&0&0\\ 0&0&6&0&0&0&0&0\\ 0&0&0&2&0&0&0&0\\ 0&0&0&0&6&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&2&0\\ 0&0&0&0&0&0&0&2 \end{bmatrix}_{8\times 8}, \quad \mathbb{D}_U^{\succeq C_\delta \cup d} = \begin{bmatrix} 2&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&4&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&5&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&2&0\\ 0&0&0&0&0&0&0&2 \end{bmatrix}_{8\times 8},$$

and the inverse matrix of $\mathbb{D}_U^{\succeq C_\delta}$ is calculated as

$$\left(\mathbb{D}_U^{\succeq C_\delta}\right)^{-1} = \begin{bmatrix} 1/2&0&0&0&0&0&0&0\\ 0&1/2&0&0&0&0&0&0\\ 0&0&1/6&0&0&0&0&0\\ 0&0&0&1/2&0&0&0&0\\ 0&0&0&0&1/6&0&0&0\\ 0&0&0&0&0&1/1&0&0\\ 0&0&0&0&0&0&1/2&0\\ 0&0&0&0&0&0&0&1/2 \end{bmatrix}_{8\times 8}.$$

Finally, according to Corollary 1, NDCE of C to d can be calculated by using matrices $\mathbb{D}_U^{\succeq C_\delta \cup d}$ and $\left(\mathbb{D}_U^{\succeq C_\delta}\right)^{-1}$ as $MNH^{\succeq}_{d|C_\delta}(U) = -\frac{1}{8}\log|\mathbb{D}_U^{\succeq C_\delta \cup d}\cdot\left(\mathbb{D}_U^{\succeq C_\delta}\right)^{-1}| = -\frac{1}{8}\log(2\times\frac{1}{2}\times 1\times\frac{1}{2}\times 4\times\frac{1}{6}\times 1\times\frac{1}{2}\times 5\times\frac{1}{6}\times 1\times\frac{1}{1}\times 2\times\frac{1}{2}\times 2\times\frac{1}{2}) = -\frac{1}{8}\log\frac{5}{36} = 0.3560$.

The following, we naturally derive the matrix-based inner and outer significance measures based on MNDCE.

Corollary 2. Given a HODS $H^{\succeq} = (U, C\cup\{d\}, V, f)$, for any $B \subseteq C$ and $\forall a \in B$, matrix calculation based inner significance of a in B is denoted as

$$Msig^{\succeq U}_{inner}(a,B,d) = MNH^{\succeq}_{d|(B-a)_\delta}(U) - MNH^{\succeq}_{d|B_\delta}(U). \tag{23}$$

Corollary 3. *Given a HODS $H^{\succeq} = (U, C\cup\{d\}, V, f)$, for any $B \subseteq C$ and $\forall a \in (C-B)$, matrix calculation based outer significance of a to B is denoted as*

$$Msig^{\succeq U}_{outer}(a,B,d) = MNH^{\succeq}_{d|B_\delta}(U) - MNH^{\succeq}_{d|(B\cup a)_\delta}(U). \tag{24}$$

3.3. Matrix-based heuristic feature selection algorithm in a HODS

A heuristic feature selection algorithm calculates a reduct from scratch when objects vary in a HODS, retraining the dynamic HODS as a new one. Thus, this algorithm is frequently referred to as a non-incremental feature selection algorithm

Table 2
A new HODS after adding objects.

	The initial data					The normalized data				
U	a_1	a_2	a_3	a_4	d	a_1	a_2	a_3	a_4	d
x_1	high	8	1.10	410	D	0.4231	0.3299	0.4785	0.4301	D
x_2	high	10	0.90	380	B	0.4231	0.4330	0.3558	0.3750	B
x_3	low	2	0.40	280	C	0.0385	0.0206	0.0491	0.1912	C
x_4	high	5	0.72	380	B	0.4231	0.1753	0.2454	0.3750	B
x_5	low	6	0.50	220	D	0.0385	0.2268	0.1104	0.0809	D
x_6	vhigh	14	1.30	480	C	0.6154	0.6392	0.6012	0.5588	C
x_7	mid	12	0.90	300	E	0.2308	0.5361	0.3558	0.2279	E
x_8	high	7	0.94	390	D	0.4231	0.2784	0.3804	0.3934	D
x_9	high	15	1.50	499	A	0.4231	0.6907	0.7239	0.5938	A
x_{10}	mid	4	0.88	280	B	0.2308	0.1237	0.3436	0.1921	B

in contrast with incremental algorithms. This subsection introduces a matrix-based heuristic feature selection (MHFS) algorithm based on Definition 3. The detailed steps are provided in Algorithm 1.

Algorithm 1 MHFS algorithm

Input: A $H^{\succeq} = (U, C \cup \{d\}, V, f)$, data normalized parameters λ_1, λ_2 and distance threshold δ.
Output: A reduct Red_U.
1: Initialize $Red_U \leftarrow \varnothing$;
2: Normalize HODS $H^{\succeq} = (U, C \cup \{d\}, V, f)$ by Eqs. (1) and (2);
3: Calculate MNDCE $MNH^{\succeq}_{d|C_\delta}(U)$ in U by Corollary 1;
4: **for** $k = 1$ to $|C|$ **do**
5: Calculate $Msig^{\succeq U}_{inner}(a_k, C, d)$ by Corollary 2;
6: **if** $Msig^{\succeq U}_{inner}(a_k, C, d) > 0$ **then**
7: $Red_U \leftarrow Red_U \cup \{a_k\}$;
8: **end if**
9: **end for**
10: Let $B \leftarrow Red_U$;
11: **while** $MNH^{\succeq}_{d|B_\delta}(U) > MNH^{\succeq}_{d|C_\delta}(U)$ **do**
12: **for** $l = 1$ to $|C - B|$
13: Calculate $Msig^{\succeq U}_{outer}(a_l, B, d)$ by Corollary 3;
14: **end for**
15: Select $a_0 = max\left\{Msig^{\succeq U}_{outer}(a_l, B, d), a_l \in (C - B)\right\}$;
16: $B \leftarrow B \cup a_0$
17: **end while**
18: **for** each $a \in B$
19: **if** $MNH^{\succeq}_{d|(B-a)_\delta}(U) \leqslant MNH^{\succeq}_{d|B_\delta}(U)$ then
20: $B \leftarrow B - a$;
21: **end if**
22: **end for**
23: $Red_U \leftarrow B$;
24: **return** Red_U;

The detailed explanation of the steps in Algorithm 1 and their time complexity are given as follows. Step 2 normalizes the HODS, and its time complexity is $O(|C||U|)$. Step 3 calculates MNDCE from scratch, and its time complexity is $O\left(|C||U|^2\right)$. Steps 4-9 obtain the indispensable feature a_k, i.e., a_k is a core feature of the HODS, and its time complexity is $O\left(|C|^2|U|^2\right)$. Steps 11-17 find the best candidate feature from the remaining feature set $C - B$ to the selected feature subset B until Step 11 no longer holds, i.e., relative reduct B is obtained, and its time complexity is $O\left(|C|^2|U|^2\right)$. Steps 18-22 delete redundant features from relative reduct B, and its time complexity is $O\left(|B|^2|U|^2\right)$. Steps 23-24 output a final reduct. In summary, the time complexity of Algorithm 1 is $O\left(|C||U| + |C||U|^2 + |C|^2|U|^2 + |C|^2|U|^2 + |B|^2|U|^2\right)$.

Here, we use an example to demonstrate how the reduct of a HODS is calculated in accordance with Algorithm 1.

Example 2. Continuing from Example 1, the process of calculating the reduct of the HODS in Table 1 by using Algorithm 1 is described as follows.

(1) We perform Step 2. The normalized HODS is shown at the right side of Table 1.
(2) We perform Step 3. MNDCE is calculated using Corollary 1 as $MNH^{\succeq}_{d|C_\delta}(U) = 0.3560$, it has been obtained in Example 1.
(3) We perform Steps 4-9. For any $a \in C, Msig^{\succeq U}_{inner}(a, C, d)$ can be calculated using Corollary 2 as $Msig^{\succeq U}_{inner}(a_1, C, d) = 0.0731 > 0, Msig^{\succeq U}_{inner}(a_2, C, d) = -0.0051 < 0, Msig^{\succeq U}_{inner}(a_3, C, d) = 0$, and $Msig^{\succeq U}_{inner}(a_4, C, d) = 0.0278 > 0$. Then, the core feature set is obtained as $B = \{a_1, a_4\}$.
(4) We perform Steps 11-17. $MNH^{\succeq}_{d|B_\delta}(U)$ is calculated using Corollary 1 as $MNH^{\succeq}_{d|B_\delta}(U) = 0.3509$. Since $MNH^{\succeq}_{d|C_\delta}(U) \geqslant MNH^{\succeq}_{d|B_\delta}(U)$, then proceed to Step 18.

(5) We perform Steps 18-22. For any $a \in B, MNH^{\succeq}_{d|(B-a)_\delta}(U)$ can be calculated using Corollary 1 as $MNH^{\succeq}_{d|(B-a_1)_\delta}(U) = 0.4934$ and $MNH^{\succeq}_{d|(B-a_4)_\delta}(U) = 0.3787$. Notably, for any $a \in B$, we have $MNH^{\succeq}_{d|(B-a)_\delta}(U) > MNH^{\succeq}_{d|B_\delta}(U)$. Thus, we can conclude that no redundant features occur in B.

(6) We perform Steps 23-24. The final reduct $Red_U = \{a_1, a_4\}$ is the output.

4. Updating mechanism of MNDCE with a variation of multiple objects

In the filter feature selection algorithm, the calculation of the uncertainty measure plays a key role that directly affects the efficiency of the algorithm. When multiple objects vary in a HODS, recomputing MNDCE is time-consuming, particularly in large data. To solve this issue, we propose the updating principles of the diagonal matrix when objects are added to or deleted from a HODS. On the basis of these principles, we introduce two updating mechanisms for calculating the new MNDCE in this section.

4.1. Updating mechanism of MNDCE while adding multiple objects

In this subsection, we discuss the incremental updating mechanism for calculating a new MNDCE when multiple objects are added to a HODS. The key step in the incremental calculation process is to update the diagonal matrix. In this, we introduce the updating principles of the diagonal matrix.

Proposition 3. *Given a HODS $H^{\succeq} = (U, C \cup \{d\}, V, f)$, where $U = \{x_1, x_2, \ldots, x_n\}$ and $P \subseteq C$. Suppose that the object set $U_{ad} = \{x_{n+1}, x_{n+2}, \ldots, x_{n+n'}\}$ is added to $H^{\succeq}$, and the new object set is denoted as $U' = U \cup U_{ad}$. The known previous neighborhood dominance relation matrix and its diagonal matrix on U with respect to P are $\mathbb{R}_U^{\succeq P_\delta} = \left[r^{P_\delta}_{(i,j)}\right]_{n\times n}$ and $\mathbb{D}_U^{\succeq P_\delta} = \left[d^{P_\delta}_{(i,j)}\right]_{n\times n}$, respectively. The updated diagonal matrix on U' with respect to P is denoted as $\mathbb{D}_{U'}^{\succeq P_\delta} = \left[d'^{P_\delta}_{(i,j)}\right]_{(n+n')\times(n+n')}$, where*

$$d'^{P_\delta}_{(i,j)} = \begin{cases} d^{P_\delta}_{(i,j)} + \sum\limits_{l=n+1}^{n+n'} r'^{P_\delta}_{(i,l)}, & 1 \leqslant i,j \leqslant n, i=j; \\ \sum_{l=1}^{n+n'} r'^{P_\delta}_{(i,l)} + 1, & n+1 \leqslant i,j \leqslant n+n', i=j; \\ 0, & 1 \leqslant i,j \leqslant n+n', i \neq j, \end{cases} \tag{25}$$

where

$$r'^{P_\delta}_{(i,j)} = \begin{cases} r^{P_\delta}_{(i,j)}, & (1 \leqslant i \leqslant n) \wedge (1 \leqslant j \leqslant n); \\ 1, & x_j N^{\succeq}_{P_\delta} x_i, (n+1 \leqslant i \leqslant n+n') \vee (n+1 \leqslant j \leqslant n+n'); \\ 0, & otherwise. \end{cases} \tag{26}$$

Proof. The proof process is divided into two parts. First, we prove Eq. (26). When U_{ad} is added to U, the new object set is $U' = \{x_1, x_2, \ldots, x_n, x_{n+1}, x_{n+2}, \ldots, x_{n+n'}\}$. From Definition 6, the new neighborhood dominance relation matrix $\mathbb{R}_{U'}^{\succeq P_\delta}$ can be divided into four parts, i.e., $\left[\begin{array}{c|c} [r^{1P_\delta}_{(i,j)}]_{n\times n} & [r^{2P_\delta}_{(i,j)}]_{n\times n'} \\ \hline [r^{3P_\delta}_{(i,j)}]_{n'\times n} & [r^{4P_\delta}_{(i,j)}]_{n'\times n'} \end{array}\right]$. We discuss the four sub-matrices as follows.

(1) The $\left[r^{1P_\delta}_{(i,j)}\right]_{n\times n}$ represents the neighborhood dominance relation matrix of $U \times U$ under P, where

$$r^{1P_\delta}_{(i,j)} = \begin{cases} 1, & x_j N^{\succeq}_{P_\delta} x_i, (n \leqslant i \leqslant n) \vee (n \leqslant j \leqslant n); \\ 0, & otherwise. \end{cases}$$

Notably, the $\left[r^{1P_\delta}_{(i,j)}\right]_{n\times n}$ is previous neighborhood dominance relation matrix $\mathbb{R}_U^{\succeq P_\delta} = \left[r^{P_\delta}_{(i,j)}\right]_{n\times n}$.

(2) The $\left[r^{2P_\delta}_{(i,j)}\right]_{n\times n'}$ represents the neighborhood dominance relation matrix of $U \times U_{ad}$ under P, where

$$r^{2P_\delta}_{(i,j)} = \begin{cases} 1, & x_j N^{\succeq}_{P_\delta} x_i, (1 \leqslant i \leqslant n) \wedge (n+1 \leqslant j \leqslant n+n'); \\ 0, & otherwise. \end{cases}$$

(3) The $\left[r^{3P_\delta}_{(i,j)}\right]_{n'\times n}$ represents the neighborhood dominance relation matrix of $U_{ad} \times U$ under P, where

$$r^{3P_\delta}_{(i,j)} = \begin{cases} 1, & x_j N^{\succeq}_{P_\delta} x_i, (n+1 \leqslant i \leqslant n+n') \wedge (1 \leqslant j \leqslant n); \\ 0, & otherwise. \end{cases}$$

(4) The $\left[r^{4P_\delta}_{(i,j)}\right]_{n' \times n'}$ represents the neighborhood dominance relation matrix of $U_{ad} \times U_{ad}$ under P, where

$$r^{4P_\delta}_{(i,j)} = \begin{cases} 1, & x_j N^{\succeq}_{P_\delta} x_i, (n+1 \leqslant i \leqslant n+n') \wedge (n+1 \leqslant j \leqslant n+n'); \\ 0, & otherwise. \end{cases}$$

Based on the above discussion, we find that $\left[r^{2P_\delta}_{(i,j)}\right]_{n \times n'}$, $\left[r^{3P_\delta}_{(i,j)}\right]_{n' \times n}$, and $\left[r^{4P_\delta}_{(i,j)}\right]_{n' \times n'}$ can be represented by

$$r'^{P_\delta}_{(i,j)} = \begin{cases} 1, & x_j N^{\succeq}_{P_\delta} x_i, (n+1 \leqslant i \leqslant n+n') \vee (n+1 \leqslant j \leqslant n+n'); \\ 0, & otherwise. \end{cases}$$

By integrating the previous and added neighborhood dominance relation matrices, we can derive the updated neighborhood dominance relation matrix $\mathbb{R}^{\succeq P_\delta}_{U'} = \left[r'^{P_\delta}_{(i,j)}\right]_{(n+n') \times (n+n')}$, where $r'^{P_\delta}_{(i,j)}$ is denoted as Eq. (26). Second, we prove Eq. (25). From Definition 8, we have $\mathbb{D}^{\succeq P_\delta}_{U'} = \left[d'^{P_\delta}_{(i,j)}\right]_{(n+n') \times (n+n')}$. In fact, all elements outside the diagonal in the diagonal matrix are zero, i.e., for any $1 \leqslant i,j \leqslant n+n', i \neq j, d'^{P_\delta}_{(i,j)} = 0$ holds. Thus, we can determine that $d'^{P_\delta}_{(i,j)}$ remain unchanged for any $1 \leqslant i,j \leqslant n, i \neq j$, i.e., $d'^{P_\delta}_{(i,j)} = d^{P_\delta}_{(i,j)}$. According to Eq. (20), for any $1 \leqslant i,j \leqslant n, i=j$, we can get that $d'^{P_\delta}_{(i,j)} = \sum_{l=1}^{n+n'} r'^{P_\delta}_{(i,l)} + 1 = \sum_{l=1}^{n} r'^{P_\delta}_{(i,l)} + \sum_{l=n+1}^{n+n'} r'^{P_\delta}_{(i,l)} + 1$. Based on Eq. (26), for any $1 \leqslant i,l \leqslant n, r'^{P_\delta}_{(i,l)} = r^{P_\delta}_{(i,l)}$ always hold. Hence, we can determine that $d'^{P_\delta}_{(i,j)} = \sum_{l=1}^{n} r^{P_\delta}_{(i,l)} + 1 + \sum_{l=n+1}^{n+n'} r'^{P_\delta}_{(i,l)} = d^{P_\delta}_{(i,j)} + \sum_{l=n+1}^{n+n'} r'^{P_\delta}_{(i,l)}$. Besides, for any $n+1 \leqslant i,j \leqslant n+n', i=j$, from Definition 8, $d'^{P_\delta}_{(i,j)}$ is calculated as$d'^{P_\delta}_{(i,j)} = \sum_{l=1}^{n+n'} r'^{P_\delta}_{(i,l)} + 1$. Thus, we can obtain the updated diagonal matrix $\mathbb{D}^{\succeq P_\delta}_{U'} = \left[d'^{P_\delta}_{(i,j)}\right]_{(n+n') \times (n+n')}$, where $d'^{P_\delta}_{(i,j)}$ is denoted as Eq. (25). □

Proposition 3 discusses the incremental updating mechanism of the proposed diagonal matrix when multiple objects are added to a HODS. Notably, the principle of updating the diagonal matrix $\mathbb{D}^{\succeq P_\delta \cup d}_{U'}$ on U' with respect to $P_\delta \cup d$ is similar to the that of Proposition 3.

Here we explain how to calculate a new MNDCE by updating the diagonal matrix when adding objects. Given a HODS $H^{\succeq} = (U, C \cup \{d\}, V, f)$, for any $P \subseteq C$, the known original matrices are $\mathbb{R}^{\succeq P_\delta}_{U}, \mathbb{R}^{\succeq P_\delta \cup d}_{U}, \mathbb{D}^{\succeq P_\delta}_{U}$, and $\mathbb{D}^{\succeq P_\delta \cup d}_{U}$. When U_{ad} is added to $H^{\succeq}$, in accordance with Proposition 3, we can obtain the updated diagonal matrices $\mathbb{D}^{\succeq P_\delta}_{U'}$ and $\mathbb{D}^{\succeq P_\delta \cup d}_{U'}$. On the basis of this matrices, calculating the new $MNH^{\succeq}_{d|P_\delta}(U')$ by using Corollary 1 is easy. In summary, after obtaining the updated diagonal matrices, we can easily derive the new $MNH^{\succeq}_{d|P_\delta}(U')$ via a simple matrix operation.

Subsequently, in accordance with Proposition 3, we will present an example to demonstrate how to update the neighborhood dominance relation matrix and its diagonal matrix. Then, the new MNDCE is calculated on the basis of these matrices by using Corollary 1.

Example 3. Continuing from Example 1, $U_{ad} = \{x_9, x_{10}\}$ is added to Table 1, a new object set is $U' = \{x_1, x_2, \ldots, x_{10}\}$. First, we just normalize the added data by using Eqs. (1) and (2). The results are shown in Table 2. First, according to Eq. (26), the neighborhood dominance relation matrix $\mathbb{R}^{\succeq C_\delta}_{U'}$ based on $\mathbb{R}^{\succeq C_\delta}_{U}$ is updated as

$$\mathbb{R}^{\succeq C_\delta}_{U'} = \left[\begin{array}{c|c} [r^{C_\delta}_{(i,j)}]_{8\times8} & [r^{2C_\delta}_{(i,j)}]_{8\times2} \\ \hline [r^{3C_\delta}_{(i,j)}]_{2\times8} & [r^{4C_\delta}_{(i,j)}]_{2\times2} \end{array}\right] = \left[\begin{array}{cccccccc|cc} 0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0 \\ 1&1&0&1&0&1&0&1&1&0 \\ 0&0&0&0&0&1&0&0&0&0 \\ 1&1&0&0&0&1&1&1&1&0 \\ 0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&1&0 \\ 0&0&0&0&0&1&0&0&0&0 \\ \hline 0&0&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&1&0&0&1&0 \end{array}\right]_{10\times10}, \quad \text{where} \quad \left[r^{C_\delta}_{(i,j)}\right]_{8\times8} = \begin{bmatrix} 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&1&0&0 \\ 1&1&0&1&0&1&0&1 \\ 0&0&0&0&0&1&0&0 \\ 1&1&0&0&0&1&1&1 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&1&0&0 \end{bmatrix}_{8\times8},$$

$$\left[r^{2C_\delta}_{(i,j)}\right]_{8\times2} = \begin{bmatrix} 0&0 \\ 0&0 \\ 1&0 \\ 0&0 \\ 1&0 \\ 0&0 \\ 1&0 \\ 0&0 \end{bmatrix}_{8\times2}, \left[r^{3C_\delta}_{(i,j)}\right]_{2\times8} = \begin{bmatrix} 0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&1&0&0 \end{bmatrix}_{2\times8}, \left[r^{4C_\delta}_{(i,j)}\right]_{2\times2} = \begin{bmatrix} 0&0 \\ 1&0 \end{bmatrix}_{2\times2}.$$

Analogously, we can obtain the updated matrix $\mathbb{R}^{\succeq C_\delta \cup d}_{U'}$ based on $\mathbb{R}^{\succeq C_\delta \cup d}_{U}$ by using Eq. (26) as

$$\mathbb{R}_{U'}^{\succeq C_\delta \cup d} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}_{10\times 10}.$$

Next, based on the updated neighborhood dominance relation matrices $\mathbb{R}_{U'}^{\succeq C_\delta}, \mathbb{R}_{U'}^{\succeq C_\delta \cup d}$, and their previous diagonal matrices $\mathbb{D}_{U}^{\succeq C_\delta}, \mathbb{D}_{U}^{\succeq C_\delta \cup d}$, we update the diagonal matrices $\mathbb{D}_{U'}^{\succeq C_\delta}$ and $\mathbb{D}_{U'}^{\succeq C_\delta \cup d}$ respectively by using Eq. (25) as $\mathbb{D}_{U'}^{\succeq C_\delta}$ =

$$\begin{bmatrix} 2+0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2+0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6+1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2+0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6+1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1+0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2+1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2+0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0+1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3+1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \end{bmatrix}_{10\times 10}$$

$\mathbb{D}_{U'}^{\succeq C_\delta \cup d}$ =

$$\begin{bmatrix} 2+0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1+0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4+1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1+0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5+1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1+0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2+1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2+0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0+1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1+1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}_{10\times 10}.$$

Final, we calculate new MNDCE by using Corollary 1 as $MNH_{d|C_\delta}^{\succeq}(U') = -\frac{1}{10}\log(2\times\frac{1}{2}\times 1\times\frac{1}{2}\times 5\times\frac{1}{7}\times 1\times\frac{1}{2}\times 6\times\frac{1}{7}\times 1\times\frac{1}{1}\times 3\times\frac{1}{3}\times 2\times\frac{1}{2}\times 1\times\frac{1}{1}\times 2\times\frac{1}{4}) = -\frac{1}{10}\log\frac{15}{196} = 0.3708$.

4.2. Updating mechanism of MNDCE while deleting multiple objects

This subsection presents an incremental method for computing a new MNDCE while deleting multiple objects from a HODS. We mostly discuss the updating mechanism of the diagonal matrix. The process of calculating a new MNDCE is similar to that presented in subSection 4.1, but the method for updating the diagonal matrix is different. The primary reason is that new content is not required to be calculated while deleting multiple objects from the original HODS. We must only move the position of the matrix elements in accordance with the position of the deleted objects and then obtain the updated matrices. The following paragraphs introduce the updating principles of the diagonal matrix.

Proposition 4. *Given a HODS $H^{\succeq} = (U, C\cup d, V, f)$, where $U = \{x_1, x_2, \ldots, x_n\}$ and $P \subseteq C$. $U_{de} = \{x_{q_1}, x_{q_2}, \ldots, x_{q_{n'}}\}$ is deleted from $H^{\succeq}$. The new object set is denoted as $U' = U - U_{de}$. The neighborhood dominance relation matrix and its diagonal matrix on U with respect to P are $\mathbb{R}_U^{\succeq P_\delta} = \left[r_{(i,j)}^{P_\delta}\right]_{n\times n}$ and $\mathbb{D}_U^{\succeq P_\delta} = \left[d_{(i,j)}^{P_\delta}\right]_{n\times n}$, respectively. The updated diagonal matrix on U' with respect to P is denoted as $\mathbb{D}_{U'}^{\succeq P_\delta} = \left[d'^{P_\delta}_{(i,j)}\right]_{(n-n')\times(n-n')}$, where*

$$d'^{P_\delta}_{(i,j)} = \begin{cases} d^{P_\delta}_{(i+k-1,j+k-1)} - \sum_{t=1}^{n'} r^{P_\delta}_{(i+k-1,q_t)}, & q_{k-1} - k + 2 \leqslant i,j < q_k - k + 1, i = j; \\ d^{P_\delta}_{(i+n',j+n')} - \sum_{t=1}^{n'} r^{P_\delta}_{(i+n',q_t)}, & q_{n'} - n' + 1 \leqslant i,j \leqslant n - n', i = j; \\ 0, & 1 \leqslant i,j \leqslant n - n', i \neq j, \end{cases} \tag{27}$$

where $1 \leqslant k \leqslant n'$.

Proof. After deleting object set U_{de}, the new object set is denoted as $U' = \{x_1, x_2, \ldots, x_{n-n'}\}$. From Eq. (20), we have $d'^{P_\delta}_{(i,j)} = \sum_{l=1}^{n-n'} r'^{P_\delta}_{(i,l)} + 1 = \sum_{l=1}^{n} r'^{P_\delta}_{(i,l)} - \sum_{t=1}^{n'} r'^{P_\delta}_{(i,t)} + 1$ for any $1 \leqslant i,j \leqslant n-n', i=j$. Subsequently, on the basis of the previous diagonal matrix $\mathbb{D}^{\succeq P_\delta}_{U}$, we discuss the three cases of changes of the elements in $\mathbb{D}^{\succeq P_\delta}_{U'}$ on U'.

(1) For any $1 \leqslant i,j \leqslant n-n', i \neq j, d'^{P_\delta}_{(i,j)} = 0$ is always true, that is, $d'^{P_\delta}_{(i,j)} = d^{P_\delta}_{(i,j)} = 0$ holds.

(2) Notably, if $1 \leqslant i,j \leqslant n$, then $r'^{P_\delta}_{(i,j)} = r^{P_\delta}_{(i,j)}$ holds. Thus, we have $d'^{P_\delta}_{(i,j)} = \sum_{l=1}^{n} r^{P_\delta}_{(i,l)} - \sum_{t=1}^{n'} r^{P_\delta}_{(i,t)} + 1 = d^{P_\delta}_{(i,j)} - \sum_{t=1}^{n'} r^{P_\delta}_{(i,t)}$. From Definition 8, for any $q_{k-1} \leqslant i,j < q_k, i=j$, the row and column coordinates of element $d^{P_\delta}_{(i,j)}$ should be shifted forward simultaneously by $k-1$ positions. Thus, we can determine that $d'^{P_\delta}_{(i,j)} = d^{P_\delta}_{(i+k-1,j+k-1)} - \sum_{t=1}^{n'} r^{P_\delta}_{(i+k-1,q_t)}$ for any $q_{k-1} - k + 2 \leqslant i,j < q_k - k + 1, i=j$.

(3) For any $q_{n'} - n' + 1 \leqslant i,j \leqslant n-n', i=j$, the row and column coordinates of element $d^{P_\delta}_{(i,j)}$ should be shifted forward simultaneously by n' positions. Thus, we can determine that $d'^{P_\delta}_{(i,j)} = d^{P_\delta}_{(i+n',j+n')} - \sum_{t=1}^{n'} r^{P_\delta}_{(i+n',q_t)}$ for any $q_{n'} - n' + 1 \leqslant i,j \leqslant n-n', i=j$.

In summary, we can obtain the updated diagonal matrix $\mathbb{D}^{\succeq P_\delta}_{U'}$, where $d'^{P_\delta}_{(i,j)}$ is denoted as Eq. (27). □

Proposition 4 discusses the incremental updating mechanism of the proposed diagonal matrix when multiple objects are deleted from a HODS. Similarly, we can update the diagonal matrix $\mathbb{D}^{\succeq P_\delta \cup d}_{U'}$ on U' with respect to $P_\delta \cup d$ in accordance with Proposition 4.

Here, an example is presented in accordance with Proposition 4 to demonstrate how the diagonal matrix can be updated when multiple objects are deleted from a HODS. Thereafter, we calculate the new MNDCE by using Corollary 1.

Example 4. Continuing from Example 1, $U_{de} = \{x_2, x_7\}$ is deleted from Table 1. Then, we obtain the new object set is $U' = \{x_1, x_3, x_4, x_5, x_6, x_8\}$ in Table 3. On the basis of previous neighborhood dominance relation matrix $\mathbb{R}^{\succeq C_\delta}_{U}$ and its diagonal matrix $\mathbb{D}^{\succeq C_\delta}_{U}$, the diagonal matrix $\mathbb{D}^{\succeq C_\delta}_{U'}$ is updated by using Proposition 4 as

$$\mathbb{D}^{\succeq C_\delta}_{U'} = \begin{bmatrix} 2-0 & \not{0} & 0 & 0 & 0 & 0 & \not{0} & 0 \\ \not{0} & \not{2} & \not{0} & \not{0} & \not{0} & \not{0} & \not{0} & \not{0} \\ 0 & \not{0} & 6-1 & 0 & 0 & 0 & \not{0} & 0 \\ 0 & \not{0} & 0 & 2-0 & 0 & 0 & \not{0} & 0 \\ 0 & \not{0} & 0 & 0 & 6-2 & 0 & \not{0} & 0 \\ 0 & \not{0} & 0 & 0 & 0 & 1-0 & \not{0} & 0 \\ \not{0} & \not{0} & \not{0} & \not{0} & \not{0} & \not{0} & \not{2} & \not{0} \\ 0 & \not{0} & 0 & 0 & 0 & 0 & \not{0} & 2-0 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}_{6\times 6}$$

Similarly, based on $\mathbb{R}^{\succeq C_\delta}_{U}$ and $\mathbb{D}^{\succeq C_\delta}_{U}$, we can obtain the updated diagonal matrix $\mathbb{D}^{\succeq C_\delta \cup d}_{U'}$ by using Proposition 4 as

$$\mathbb{D}^{\succeq C_\delta \cup d}_{U'} = \begin{bmatrix} 2-0 & \not{0} & 0 & 0 & 0 & 0 & \not{0} & 0 \\ \not{0} & \not{1} & \not{0} & \not{0} & \not{0} & \not{0} & \not{0} & \not{0} \\ 0 & \not{0} & 4-1 & 0 & 0 & 0 & \not{0} & 0 \\ 0 & \not{0} & 0 & 1-0 & 0 & 0 & \not{0} & 0 \\ 0 & \not{0} & 0 & 0 & 5-1 & 0 & \not{0} & 0 \\ 0 & \not{0} & 0 & 0 & 0 & 1-0 & \not{0} & 0 \\ \not{0} & \not{0} & \not{0} & \not{0} & \not{0} & \not{0} & \not{2} & \not{0} \\ 0 & \not{0} & 0 & 0 & 0 & 0 & \not{0} & 2-0 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}_{6\times 6}$$

Final, we calculate new MNDCE by using Corollary 1 as $MNH^{\succeq}_{d|C_\delta}(U') = -\frac{1}{6}\log(2 \times \frac{1}{2} \times 3 \times \frac{1}{5} \times 1 \times \frac{1}{2} \times 4 \times \frac{1}{4} \times 1 \times \frac{1}{1} \times 2 \times \frac{1}{2}) = -\frac{1}{6}\log\frac{3}{10} = 0.2895$.

Table 3
A new HODS after deleting object set.

	The initial data					The normalized data				
U	a_1	a_2	a_3	a_4	d	a_1	a_2	a_3	a_4	d
x_1	high	8	1.10	410	D	0.4231	0.3299	0.4785	0.4301	D
~~x_2~~	~~high~~	~~10~~	~~0.90~~	~~380~~	~~B~~	~~0.4231~~	~~0.4330~~	~~0.3558~~	~~0.3750~~	~~B~~
x_3	low	2	0.40	280	C	0.0385	0.0206	0.0491	0.1912	C
x_4	high	5	0.72	380	B	0.4231	0.1753	0.2454	0.3750	B
x_5	low	6	0.50	220	D	0.0385	0.2268	0.1104	0.0809	D
x_6	vhigh	14	1.30	480	C	0.6154	0.6392	0.6012	0.5588	C
~~x_7~~	~~mid~~	~~12~~	~~0.90~~	~~300~~	~~E~~	~~0.2308~~	~~0.5361~~	~~0.3558~~	~~0.2279~~	~~E~~
x_8	high	7	0.94	390	D	0.4231	0.2784	0.3804	0.3934	D

5. Matrix-based incremental algorithms for feature selection

From the analysis in the previous section, we determine that the original MNDCE will change when multiple objects are added to or deleted from a HODS. This condition directly causes a change in the significance of features, which may make the previous reduct invalid for the new HODS. Thus, we must recalculate a valid reduct when the objects vary in a HODS. In this section, we propose two matrix-based incremental feature selection algorithms based on the updated principle of MNDCE proposed in the previous section. To further improve the computational efficiency, we also construct a sequence of all candidate features in two incremental feature selection algorithms to accelerate the selection of feature subsets by referring to the principles of Corollary 3.

5.1. Matrix-based incremental feature selection algorithm while adding multiple objects

In this subsection, we first develop a matrix-based incremental feature selection algorithm while adding multiple objects (MIFSA). Then, we analyze the time complexity of the proposed algorithm. Lastly, an example is presented to demonstrate the computational process of MIFSA algorithm.

Algorithm 2 MIFSA algorithm

Input:
(1) An original normalized $H^{\succeq} = (U, C \cup \{d\}, V, f)$, where $U = \{x_1, x_2, \ldots, x_n\}$, data normalized parameters λ_1, λ_2 and distance threshold δ;
(2) The $U_{ad} = \{x_{n+1}, x_{n+2}, \ldots, x_{n+n'}\}$ is an added object set;
(3) The original reduct Red_U on U;
(4) The original neighborhood dominance relation matrices $\mathbb{R}_U^{\succeq C_\delta}$, $\mathbb{R}_U^{\succeq Red_\delta}$, $\mathbb{R}_U^{\succeq C_\delta \cup d}$, and $\mathbb{R}_U^{\succeq Red_\delta \cup d}$, and their diagonal matrices $\mathbb{D}_U^{\succeq C_\delta}$, $\mathbb{D}_U^{\succeq Red_\delta}$, $\mathbb{D}_U^{\succeq C_\delta \cup d}$, and $\mathbb{D}_U^{\succeq Red_\delta \cup d}$.
Output: A new reduct $Red_{U'}$ on $U \cup U_{ad}$.
1: Normalize the added object set U_{ad} by Eqs. (1) and (2), and integrate it into $H^{\succeq}$;
2: Initialize $B \leftarrow Red_U$, $U' \leftarrow U \cup U_{ad}$, $\mathbb{D}_{U'}^{\succeq C_\delta} \leftarrow \mathbb{D}_U^{\succeq C_\delta}$, $\mathbb{D}_{U'}^{\succeq C_\delta \cup d} \leftarrow \mathbb{D}_U^{\succeq C_\delta \cup d}$, $\mathbb{D}_{U'}^{\succeq B_\delta} \leftarrow \mathbb{D}_U^{\succeq B_\delta}$, and $\mathbb{D}_{U'}^{\succeq B_\delta \cup d} \leftarrow \mathbb{D}_U^{\succeq B_\delta \cup d}$;
3: Update the diagonal matrices $\mathbb{D}_{U'}^{\succeq C_\delta} \leftarrow \left[d_{(i,j)}{'}^{C_\delta}\right]_{(n+n') \times (n+n')}$, $\mathbb{D}_{U'}^{\succeq C_\delta \cup d} \leftarrow \left[d_{(i,j)}{'}^{C_\delta \cup d}\right]_{(n+n') \times (n+n')}$, $\mathbb{D}_{U'}^{\succeq B_\delta} \leftarrow \left[d_{(i,j)}{'}^{B_\delta}\right]_{(n+n') \times (n+n')}$, and $\mathbb{D}_{U'}^{\succeq B_\delta \cup d} \leftarrow \left[d_{(i,j)}{'}^{B_\delta \cup d}\right]_{(n+n') \times (n+n')}$ by Proposition 3;
4: Compute new MNDCEs $MNH^{\succeq}_{d|C_\delta}(U')$ and $MNH^{\succeq}_{d|B_\delta}(U')$ by Corollary 1;
5: **if** $MNH^{\succeq}_{d|B_\delta}(U') \leqslant MNH^{\succeq}_{d|C_\delta}(U')$ **then**
6: go to step 16;
7: **else**
8: go to step 10;
9: **end if**
10: For any $a \in (C - B)$, compute $Msig^{\succeq U'}_{outer}(a, B, d)$ by Corollary 3, then construct a descending sequence by $Msig^{\succeq U'}_{outer}(a, B, d)$, and record the results by $\left\{a'_1, a'_2, \ldots, a'_{|C-B|}\right\}$;
11: **while** $MDH^{\succeq}_{d|B}(U') > MDH^{\succeq}_{d|C}(U')$ **do**
12: **for** $h = 1$ to $|C - B|$ **do**
13 : select $B \leftarrow B \cup a'_h$ and compute $MNH^{\succeq}_{d|B_\delta}(U')$;
14: **end for**
15: **end while**
16: **for** each $a \in B$ **do**
17: compute $MNH^{\succeq}_{d|(B-a)_\delta}(U')$;
18: **if** $MNH^{\succeq}_{d|(B-a)_\delta}(U') \leqslant MNH^{\succeq}_{d|B_\delta}(U')$ **then**
19: $B \leftarrow B - a$;
20: **end if**
21: **end for**
22: $Red_{U'} \leftarrow B$;
23: **return** $Red_{U'}$;

The detailed explanation of the steps of Algorithm 2 and their time complexity are presented as follows. Step 1 normalizes the added object set, and its time complexity is $O(|C||U_{ad}|)$. Step 3 updates the diagonal matrices in an incremental manner via Proposition 3, and its time complexity is $O(|C||U_{ad}||U'|)$. Step 4 computes new MNDCEs by using Corollary 1. Steps 5-9 determine whether the new MNDCE under the original selected feature subset (i.e., original reduct) is not higher than that of under the entire feature set; if yes, then the original selected feature subset is kept unchanged. Steps 10-15 arrange the remaining features in descending order, and incrementally update the selected feature subset until Step 11 no longer holds, its time complexity is $O\left((|C|-|B|)|U'|^2\right)$. Steps 16-21 delete redundant features from the selected feature subset, and its time complexity is $O\left(|B|^2|U'|^2\right)$. Steps 22-23 output a final reduct. In summary, the time complexity of Algorithm 2 is $O\left(|C||U_{ad}|+|C||U_{ad}||U'|+(|C|-|B|)|U'|^2+|B|^2|U'|^2\right)$. The comparison of the time complexities of algorithms MHFS and MIFSA is provided in Table 4.

Notably, the time complexity of algorithm MIFSA is typically much less than that of algorithm MHFS, as shown in Table 4. The primary reason for this condition is that algorithm MHFS calculates a new reduct without prior information when multiple objects are added to the original HODS. By contrast, algorithm MIFSA uses previous knowledge to quickly calculate the new MNDCE by applying the updating principles. Then, it calculates a new reduct on the basis of the greedy search strategy. In real-life applications, the number of samples in a HODS is considerably higher than the number of features, i.e., $|U| \gg |C|$. Therefore, algorithm MIFSA exhibits a more significant time-saving effect on calculating reduct for large-scale data than algorithm MHFS.

Subsequently, we present an example to demonstrate the detailed steps for calculating a new reduct by using Algorithm 2 when multiple objects are added to a HODS.

Example 5. Continuing from Example 2, the known knowledge of the original HODS includes the reduct $Red_U=\{a_1,a_4\}$; the neighborhood dominance relation matrices $\mathbb{R}_U^{\succeq C_\delta}$, $\mathbb{R}_U^{\succeq Red_\delta}$, $\mathbb{R}_U^{\succeq C_\delta \cup d}$, and $\mathbb{R}_U^{\succeq Red_\delta \cup d}$; and their diagonal matrices $\mathbb{D}_U^{\succeq C_\delta}$, $\mathbb{D}_U^{\succeq Red_\delta}$, $\mathbb{D}_U^{\succeq C_\delta \cup d}$, and $\mathbb{D}_U^{\succeq Red_\delta \cup d}$.

(1) We perform Step 1. The added object set $U_{ad}=\{x_9,x_{10}\}$ is normalized and integrated into the original normalized HODS. The results are provided at the right side of Table 2.
(2) We perform Steps 2-3. Let $B \leftarrow Red_U, U' \leftarrow U \cup U_{ad}$. Then, the diagonal matrices are updated via Proposition 3 as $\mathbb{D}_{U'}^{\succeq C_\delta}, \mathbb{D}_{U'}^{\succeq C_\delta \cup d}, \mathbb{D}_{U'}^{\succeq B_\delta}$, and $\mathbb{D}_{U'}^{\succeq B_\delta \cup d}$.
(3) We perform Step 4. The two new MNDCEs can be calculated using Corollary 1 as $MNH^{\succeq}_{d|C_\delta}(U')=0.3708$ and $MNH^{\succeq}_{d|B_\delta}(U')=0.3463$.
(4) We perform Steps 5-9. Given that $MNH^{\succeq}_{d|C_\delta}(U')>MNH^{\succeq}_{d|B_\delta}(U')$, proceed to Step 16.

Table 4
The comparison of the time complexity of algorithms MHFS and MIFSA.

Algorithm	MHFS	MIFSA
Time complexity	$O\left(\lvert C\rvert\lvert U'\rvert+\lvert C\rvert\lvert U'\rvert^2+\lvert C\rvert^2\lvert U'\rvert^2+\lvert C\rvert^2\lvert U'\rvert^2+\lvert B\rvert^2\lvert U'\rvert^2\right)$	$O\left(\lvert C\rvert\lvert U_{ad}\rvert+\lvert C\rvert\lvert U_{ad}\rvert\lvert U'\rvert+(\lvert C\rvert-\lvert B\rvert)\lvert U'\rvert^2+\lvert B\rvert^2\lvert U'\rvert^2\right)$

Table 5
The comparison of the time complexity of algorithms MHFS and MIFSD.

Algorithm	MHFS	MIFSD
Time complexity	$O\left(\lvert C\rvert\lvert U'\rvert+\lvert C\rvert\lvert U'\rvert^2+\lvert C\rvert^2\lvert U'\rvert^2+\lvert C\rvert^2\lvert U'\rvert^2+\lvert B\rvert^2\lvert U'\rvert^2\right)$	$O\left(\lvert U_{de}\rvert\lvert U\rvert+(\lvert C\rvert-\lvert B\rvert)\lvert U'\rvert^2+\lvert B\rvert^2\lvert U'\rvert^2\right)$

Table 6
The description of data sets.

No.	Data sets	Abbreviation	Samples	Features			Classes
				Numerical	Categorical	Total	
1	Iris	Iris	150	4	0	4	3
2	Wine	Wine	178	13	0	13	3
3	Leaf	Leaf	340	15	0	15	30
4	Libras Movement	Libras	360	90	0	90	15
5	Robot Execution Failures	Robot	463	90	0	90	16
6	Mice Protein Expression	Mice	1077	68	0	68	8
7	Car	Car	1728	0	6	6	4
8	Postoperative	Post	87	1	7	8	3
9	Hepatitis	Hepa	155	6	13	19	2
10	Australian	Aust	690	6	8	14	2
11	Artificial Data 1	AD1	5473	7	3	10	5
12	Artificial Data 2	AD2	6436	7	3	10	5

(5) We perform Steps 16-21. For any $a \in B, MNH^{\succeq}_{d|(B-a_1)_\delta}(U') = 0.3445$ and $MNH^{\succeq}_{d|(B-a_4)_\delta}(U') = 0.4562$ are calculated. Given that $MNH^{\succeq}_{d|(B-a_1)_\delta}(U') < MDH^{\succeq}_{d|B}(U'), a_1$ is a redundant feature, and thus, the final selected feature subset is $B = \{a_4\}$.

(6) We perform Steps 22-23. The final reduct $Red_{U'} = \{a_4\}$ is the output.

5.2. Matrix-based incremental feature selection algorithm while deleting multiple objects

This subsection first introduces a matrix-based incremental feature selection algorithm while deleting multiple objects (MIFSD). Then, the time complexity of the proposed algorithm is analyzed. Lastly, we demonstrate the process of the proposed algorithm by providing an example.

Algorithm 3 MIFSD algorithm

Input:
(1) An original normalized $H^{\succeq} = (U, C \cup \{d\}, V, f)$, where $U = \{x_1, x_2, \ldots, x_n\}, U_{de} = \{x_{q_1}, x_{q_2}, \ldots, x_{q_{n'}}\}$ is a deleted object set, distance threshold δ;
(2) The original reduct Red_U on U;
(3) The original neighborhood dominance relation matrices $\mathbb{R}^{\succeq C_\delta}_U$, $\mathbb{R}^{\succeq Red_\delta}_U$, $\mathbb{R}^{\succeq C_\delta \cup d}_U$, and $\mathbb{R}^{\succeq Red_\delta \cup d}_U$, and their diagonal matrices $\mathbb{D}^{\succeq C_\delta}_U$, $\mathbb{D}^{\succeq Red_\delta}_U$, $\mathbb{D}^{\succeq C_\delta \cup d}_U$, and $\mathbb{D}^{\succeq Red_\delta \cup d}_U$.
Output: A new reduct $Red_{U'}$ on $U - U_{de}$.
1: Delete the object set U_{de} from the original normalized HODS;
2: Initialize $B \leftarrow Red_U, U' \leftarrow U - U_{de}, \mathbb{D}^{\succeq C_\delta}_{U'} \leftarrow \mathbb{D}^{\succeq C_\delta}_U$, $\mathbb{D}^{\succeq C_\delta \cup d}_{U'} \leftarrow \mathbb{D}^{\succeq C_\delta \cup d}_U, \mathbb{D}^{\succeq B_\delta}_{U'} \leftarrow \mathbb{D}^{\succeq B_\delta}_U$, and $\mathbb{D}^{\succeq B_\delta \cup d}_{U'} \leftarrow \mathbb{D}^{\succeq B_\delta \cup d}_U$;
3: Update the diagonal matrices $\mathbb{D}^{\succeq C_\delta}_{U'} \leftarrow [d_{(i,j)}{'}^{C_\delta}]_{(n-n')\times(n-n')}, \mathbb{D}^{\succeq C_\delta \cup d}_{U'} \leftarrow [d_{(i,j)}{'}^{C_\delta \cup d}]_{(n-n')\times(n-n')}, \mathbb{D}^{\succeq B_\delta}_{U'} \leftarrow [d_{(i,j)}{'}^{B_\delta}]_{(n-n')\times(n-n')}$, and $\mathbb{D}^{\succeq B_\delta \cup d}_{U'} \leftarrow [d_{(i,j)}{'}^{B_\delta \cup d}]_{(n-n')\times(n-n')}$ by Proposition 4;
4: Compute new MNDCEs $MNH^{\succeq}_{d|C_\delta}(U')$ and $MNH^{\succeq}_{d|B_\delta}(U')$ by Corollary 1;
5: **if** $MNH^{\succeq}_{d|B_\delta}(U') \leqslant MNH^{\succeq}_{d|C_\delta}(U')$ **then**
6: go to step 16;
7: **else**
8: go to step 10;
9: **end if**
10: For any $a \in (C - B)$, compute $Msig^{\succeq U'}_{outer}(a, B, d)$ by Corollary 3, then construct a descending sequence by $Msig^{\succeq U'}_{outer}(a, B, d)$, and record the results by $\{a'_1, a'_2, \ldots, a'_{|C-B|}\}$;
11: **while** $MNH^{\succeq}_{d|B_\delta}(U') > MNH^{\succeq}_{d|C_\delta}(U')$ **do**
12: **for** $h = 1$ to $|C - B|$ **do**
13: select $B \leftarrow B \cup a'_h$ and compute $MNH^{\succeq}_{d|B_\delta}(U')$;
14: **end for**
15: **endwhile**
16: **for** each $a \in B$ **do**
17: compute $MNH^{\succeq}_{d|(B-a)_\delta}(U')$;
18: **if** $MNH^{\succeq}_{d|(B-a)_\delta}(U') \leqslant MNH^{\succeq}_{d|B_\delta}(U')$ **then**
19: $B \leftarrow B - a$;
20: **end if**
21: **end for**
22: $Red_{U'} \leftarrow B$;
23: **return** $Red_{U'}$;

The detailed explanation of the steps of Algorithm 3 and its time complexity are provided as follows. Step 3 updates the diagonal matrices in an incremental manner via Proposition 4, and its time complexity is $O(|U_{de}||U|)$. Step 4 computes new MNDCEs by using Corollary 1. Steps 5-9 determine whether the new MNDCE under the original selected feature subset (i.e., the original reduct) is not higher than that of under the entire feature set; if yes, then the original selected feature subset is kept unchanged. Steps 10-15 arrange the remaining features in descending order and incrementally update the selected feature subset until Step 10 no longer holds; its time complexity is $O\left((|C| - |B|)|U'|^2\right)$. Steps 16-21 delete redundant features from the selected feature subset, and its time complexity is $O\left(|B|^2|U'|^2\right)$. Steps 22-23 output a final reduct. In summary, the time complexity of Algorithm 3 is $O\left(|U_{de}||U| + (|C| - |B|)|U'|^2 + |B|^2|U'|^2\right)$. The comparison of the time complexities of algorithms MHFS and MIFSD is presented in Table 5.

Table 5 clearly shows that the time complexity of algorithm MIFSD is considerably lower than that of algorithm MHFS. The primary reason for this condition is that algorithm MIFSD uses the previous knowledge when calculating a new reduct. By contrast, algorithm MHFS calculates a new reduct without previous knowledge. Therefore, the use of algorithm MHFS is highly time-consuming for computing a new reduct.

Subsequently, an example is presented to demonstrate the process of calculating a new reduct by using Algorithm 3 when multiple objects are deleted from a HODS.

Example 6. Continuing from Example 2, the known knowledge of the original HODS includes reduct $Red_U = \{a_1, a_4\}$; neighborhood dominance relation matrices $\mathbb{R}_U^{\succeq C_\delta}$, $\mathbb{R}_U^{\succeq Red_\delta}$, $\mathbb{R}_U^{\succeq C_\delta \cup d}$, and $\mathbb{R}_U^{\succeq Red_\delta \cup d}$; and their diagonal matrices $\mathbb{D}_U^{\succeq C_\delta}$, $\mathbb{D}_U^{\succeq Red_\delta}$, $\mathbb{D}_U^{\succeq C_\delta \cup d}$, and $\mathbb{D}_U^{\succeq Red_\delta \cup d}$.

(1) We perform Step 1. The object set $U_{de} = \{x_2, x_7\}$ is deleted from Table 1. The results are provided in Table 3.
(2) We perform Steps 2-3. Let $B \leftarrow Red_U, U' \leftarrow U - U_{de}$. Then, the diagonal matrices are updated via Proposition 4 as $\mathbb{D}_{U'}^{\succeq C_\delta}, \mathbb{D}_{U'}^{\succeq C_\delta \cup d}$, $\mathbb{D}_{U'}^{\succeq B_\delta}$, and $\mathbb{D}_{U'}^{\succeq B_\delta \cup d}$.
(3) We perform Step 4. The two new MNDCEs can be calculated using Corollary 1 as $MNH_{d|C_\delta}^{\succeq}(U') = 0.2895$ and $MNH_{d|B_\delta}^{\succeq}(U') = 0.2895$.
(4) We perform Steps 5-9. Given that $MNH_{d|C_\delta}^{\succeq}(U') = MNH_{d|B_\delta}^{\succeq}(U')$, then proceed to Step 16.
(5) We perform Steps 16-21. For any $a \in B, MNH_{d|(B-a_1)_\delta}^{\succeq}(U') = 0.3870$ and $MNH_{d|(B-a_4)_\delta}^{\succeq}(U') = 0.2895$ are calculated. Given that $MNH_{d|(B-a_4)_\delta}^{\succeq}(U') = MDH_{d|B}^{\succeq}(U'), a_4$ is a redundant feature, and thus, the final selected feature subset is $B = \{a_1\}$.
(6) We perform Steps 22-23. The final reduct $Red_{U'} = \{a_1\}$ is the output.

6. Experimental results and analysis

In this section, we demonstrate the effectiveness and efficiency of the proposed incremental algorithms by performing a series of experiments. We downloaded ten data sets from UC Irvine machine learning repository, including six numerical data sets, one categorical data set, and three heterogeneous data sets. To evaluate the proposed incremental algorithms on larger data sets, two heterogeneous artificial data sets AD1 and AD2 are also provided. The summary of the twelve selected data sets is provided in Table 6. To ensure repeatability of the experiment, relevant data sets can be downloaded from the GitHub homepage [1]. In this work, all algorithms are coded in Java language and run on a computer with 3.20 GHz CPU Intel(R) Core(TM) i7-8700, 16.0 GB memory, and 64-bit Windows 10 operation system.

Some of the raw data sets in Table 6 cannot be used directly in the experiments. Hence, we preprocess these data sets, and the detailed steps are described as follows. First, for data sets with few missing values, such as Mice Protein Expression and Postoperative, we deleted the objects with missing values. However, the data set Hepatitis contains a large number of missing values, and thus, we replace these value with the average of their value domain. Second, we replace symbols with integers for the value domain of categorical features. For example, in the data set Australian, the domain of the categorical feature A1 contains the symbols p, g, and gg, their relation is defined as $p < g < gg$. We naturally define the substitution rules for these symbols as $p = 1, g = 2$, and $gg = 3$. Evidently, $1 < 2 < 3$ conforms to the ranking rules for raw data. In this manner, the information contained in the raw data sets is not changed or lost.

6.1. Performance evaluations of algorithm MIFSA when adding multiple objects

In this subsection, we evaluate the performance of algorithm MIFSA in terms of effectiveness and efficiency. In terms of effectiveness, we compare algorithms MIFSA and MHFS from two aspects: reduct size and its classification accuracy. In terms of efficiency, we compare algorithms MIFSA and MHFS from two aspects: computational time and speed-up ratio. The specific experimental design is described as follows.

6.1.1. Effectiveness evaluations

This subsection compares the effectiveness of algorithms MIFSA and MHFS. We randomly select 50% of the objects from each data set in Table 6 as the original object set. The remaining 50% of the objects are regarded as the added objects. The two types of experiments described below are conducted.

(1) Comparison of the algorithms MIFSA and MHFS in terms of reduct size
For each data set in Table 6, algorithms MIFSA and MHFS are used to calculate a new reduct when the remaining 50% of the objects are added to the original 50% object set. The experimental results are presented in Table 7, which lists the number of features in the reduct (NFR), reduct, feature reduction rate (FRR) ($FRR = (|C| - NFR)/|C|$; C is the raw feature set), computational time, and time reduction rate (TRR) ($TRR = (T_{MHFS} - T_{MIFSA})/T_{MHFS}; T_*$ is the running time of

[1] https://github.com/binbinsang/Experimental-data-sets.git

Table 7
The comparison of reducts of algorithms MIFSA and MHFS.

Data sets	MHFS				MIFSA				TRR (%)
	NFR	Reduct	FRR (%)	Times (ms)	NFR	Reduct	FRR (%)	Times (ms)	
Iris	2	3,4	50.0	307	2	3,4	50.0	26	91.5
Wine	9	2,4,5,6,8,9,10,11,12	30.8	449	8	2,4,5,8,9,10,11,12	38.5	99	78.0
Leaf	5	2,5,6,9,12	66.7	977	5	2,5,6,9,10	66.7	154	84.2
Libras	12	2,7,27,33,53,54,60,70,77,80,85,90	86.7	14792	12	1,6,19,30,33,50,57,64,75,80,89,90	86.7	1805	87.8
Robot	5	1,39,40,42,85	94.4	14285	6	1,2,16,31,45,72	93.3	1294	90.9
Mice	14	1,6,29,30,34,40,41,44,46,48,49,55,57,68	79.4	69068	14	29,30,34,40,41,44,45,46,48,49,54,57,65,68	79.4	11396	83.5
Car	5	1,2,4,5,6	16.7	2442	5	1,2,4,5,6	16.7	1612	34.0
Post	2	6,7	75.0	322	3	2,3,6	62.5	16	95.0
Hepa	3	1,5,8	84.2	452	3	1,5,8	84.2	62	86.3
Aust	2	8,14	85.7	1762	2	8,14	85.7	380	78.4
AD1	5	1,4,5,7,10	50.0	29523	4	1,4,5,7	60.0	15425	47.8
AD2	5	1,4,5,7,10	50.0	49501	4	1,4,5,7	60.0	18041	63.6
Average	5.75	–	64.1	15323	5.67	–	65.3	4193	76.8

algorithm $*$).
[Table 7](#) shows that the NFR and reduct generated by the algorithms MIFSA and MHFS are extremely close. In particular, NFR and reduct are identical in data sets Iris, Car, Hepa, and Aust. Moreover, the FRR of the two algorithms is extremely close, with both reaching more than 60%. However, the running time of algorithm MIFSA is considerably shorter than that of algorithm MHFS. The TRR of most data sets is above 50%, particularly those of data sets Iris, Robot, and Post (i.e., all above 90%). The average TRR reaches 76.8%, indicating that algorithm MIFSA remarkably reduces the time cost of reduct computing.

(2) Comparison of algorithms MIFSA and MHFS in terms of classification accuracy
We then compare the classification performance of the reduct generated using MIFSA, the reducts generated using MHFS, and the raw feature set. On the basis of 10-fold cross-validation, we use the classifiers ordinal class classifier (OCC), J48, and random tree (RT) to test the classification accuracy of the raw feature set and the selected reducts in Table 7. The algorithms of the these three classifiers are implemented in Weka [10]. The OCC is a meta-classifier that enables the application of conventional classification techniques to ordinal class problems [9]. The J48 classifier is an implementation of the C4.5 decision tree algorithm. It classifies a new instance via a decision tree, which is generated by evaluating the information gain ratio of attributes [21]. Multiple decision trees are built in RT, and these trees are used in the classification task. Attributes are selected randomly when building each decision tree, and attributes with maximum information gain are selected as the split node to construct a decision tree. For the three aforementioned classifiers, we use the ratio of correct classification of instances to evaluate classification performance. The experimental results are recorded in Table 8.

As indicated in Table 8, the classification accuracy of the reducts generated using algorithms MHFS and MIFSA in most data sets is extremely close to or slightly higher than that of the raw feature set. This finding proves that the proposed feature selection strategy based on neighborhood dominance relation can accurately delete redundant features in a HODS and main-

Table 8
The comparison of algorithms MIFSA and MHFS on classification accuracy (%).

Data sets	OCC			J48			TR		
	Raw	MHFS	MIFSA	Raw	MHFS	MIFSA	Raw	MHFS	MIFSA
Iris	94.00	95.33	**95.33**	96.00	96.00	**96.00**	92.00	93.33	**93.33**
Wine	89.89	89.89	**90.45**	94.38	94.38	**94.38**	91.01	91.57	**91.57**
Leaf	37.94	40.00	**41.18**	59.70	**60.59**	60.29	59.12	59.71	**61.47**
Libras	51.94	48.06	**54.16**	70.28	65.27	**70.83**	66.11	61.39	**68.88**
Robot	**44.71**	44.06	38.23	**47.52**	42.11	44.50	31.32	31.75	**31.75**
Mice	**77.62**	76.04	76.13	84.68	84.68	**85.52**	81.15	83.57	**85.42**
Car	92.19	92.19	**92.19**	92.36	93.22	**93.22**	83.97	94.21	**94.21**
Post	67.82	68.97	**68.97**	66.67	68.97	**68.97**	59.77	**73.56**	68.97
Hepa	60.00	66.45	**66.45**	60.00	66.45	**66.45**	58.71	66.45	**66.45**
Aust	85.22	85.50	**85.50**	85.22	85.51	**85.51**	81.30	83.33	**83.33**
AD1	96.67	96.82	**96.89**	96.88	97.22	**97.24**	96.18	**96.77**	96.46
AD2	97.14	97.33	**97.42**	97.33	**97.51**	97.41	**97.56**	97.48	97.27
Average	74.60	75.05	**75.24**	79.25	79.33	**80.03**	74.85	77.76	**78.26**

tain or even improve the classification accuracy of data sets. Moreover, the classification accuracy of the reducts generated using algorithm MIFSA in the three classifiers is extremely close or even slightly higher than that of the reducts generated using algorithm MHFS for most data sets, as fully illustrated by their average values. Notably, for the data set Libras, the reduct produced by algorithm MIFSA is significantly higher in terms of classification accuracy than that produced by algorithm MHFS in each classifier. Hence, Table 8 indicates that the reduct generated by algorithm MIFSA is feasible.

6.1.2. Efficiency evaluations

In this subsection, we evaluate the efficiency of algorithm MIFSA by comparing the computational time and speed-up ratio of algorithms MIFSA and MHFS. For each data set in Table 6, 50% of the objects are randomly selected as the original object set. Then, different test sets are constructed by adding varying proportions of objects from the remaining 50% objects to the original object set, i.e., 10%, 20%, 30%, 40%, and 50% of the objects from the remaining 50% objects are added to the original object set. Therefore, we can obtain five new data sets with different sizes in the proportion to 60%, 70%, 80%, 90%, and 100% of the data sets.

(1) Comparison of algorithms MIFSA and MHFS in terms of computational time

Each data set obtains five new data sets with different sizes. These data sets are used to determine the computational time of algorithms MIFSA and MHFS. Fig. 2 shows the detailed change trend lines of the two algorithms with the increasing size of different data sets. The abscissa represents the size of added data sets, and the ordinate represents the computational time value. The computational time of algorithm MHFS is depicted by lines with hollow dots, and that of algorithm MIFSA is depicted by lines with solid dots.

As shown in Fig. 2, the computational time of algorithms MHFS and MIFSA increases as the size of the incremental object set increases. Each sub-figure indicates that the computational time of algorithm MIFSA is significantly shorter than that of algorithm MHFS. Moreover, as the proportion of the added object set increases, the growth trend of the time consumed by MIFSA is slower than that by MHFS. For the AD1 and AD2 data sets, the computational time of algo-

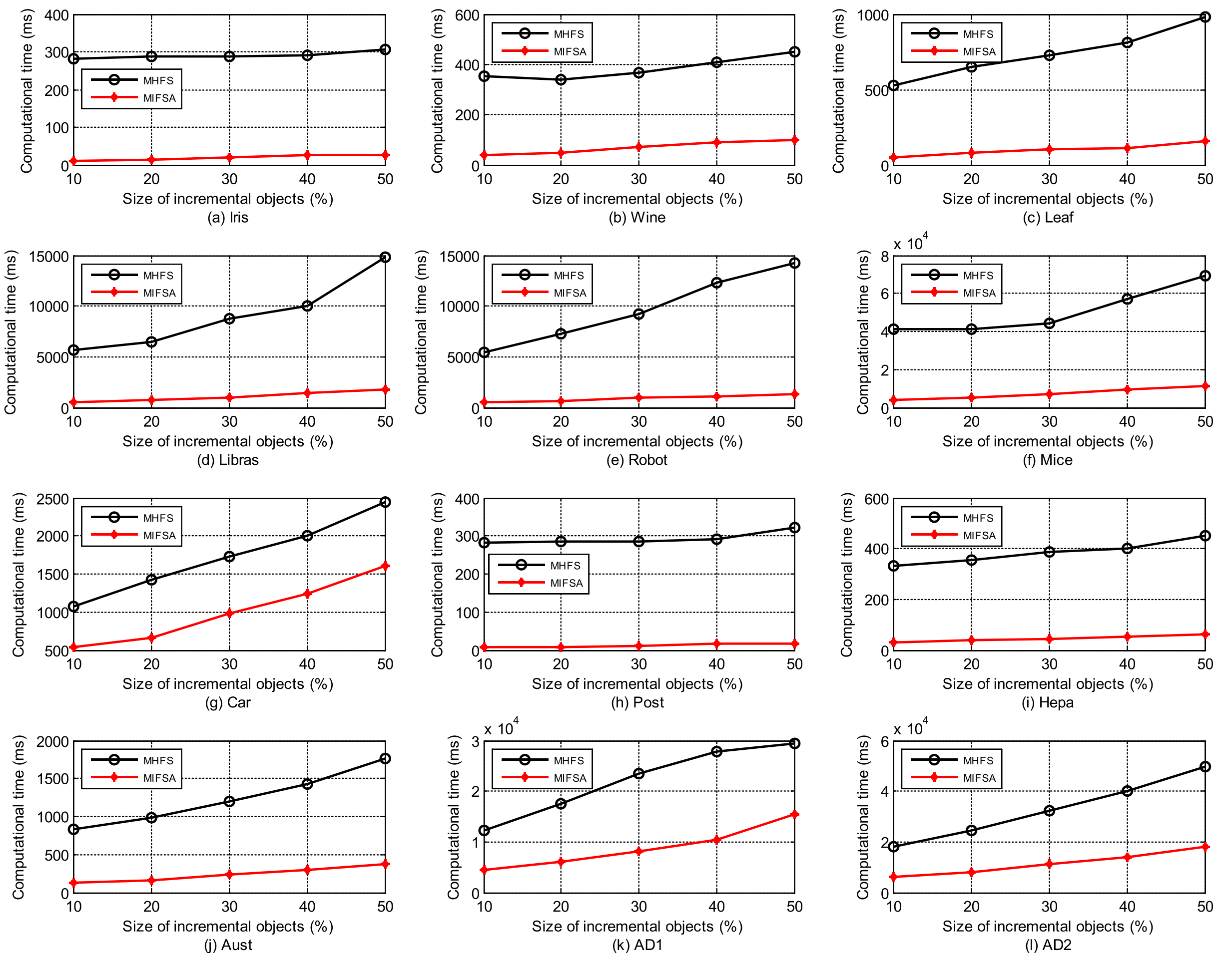

Fig. 2. The comparison of computational time between algorithms MHFS and MIFSA versus different ratio size of adding objects.

rithm MIFSA is approximately 50% that of algorithm MHFS. Hence, we can conclude from Fig. 2 that algorithm MIFSA can calculate the reduct within a shorter time when adding multiple objects, and the effect of saving time is highly evident.

(2) Evaluation of the efficiency of algorithm MIFSA in terms of speed-up ratio
In this subsection, we demonstrate the efficiency of algorithm MIFSA from the aspect of speed-up ratio. We compute the speed-up ratio of each data set on the basis of the results shown in Fig. 2. The experimental results are presented in Fig. 3, where the abscissa denotes the size of the added data sets and the ordinate denotes the value of the speed-up ratio.

As shown in Fig. 3, algorithm MIFSA is at least nearly two times faster than algorithm MHFS on all the data sets. Notably, on some data sets, such as Post, Iris, and Robot, algorithm MIFSA is even ten times faster than algorithm MHFS. The experimental results prove again that algorithm MIFSA exhibits better performance than algorithm MHFS in computing a feasible reduct.

6.1.3. Summary

From the comparisons of effectiveness and efficiency between algorithms MIFSA and MHFS, a conclusion can be drawn that algorithm MIFSA is better than algorithm MHFS. When adding multiple objects to a HODS, we study the update mechanism of the neighborhood dominance relation matrix and its diagonal matrix that are the basis for calculating MNDCE (see Corollary 1). Therefore, we can efficiently update MNDCE by improving its computational efficiency, and consequently, efficiently obtain the reduct of a HODS. Algorithm MIFSA uses previous knowledge, and thus, avoids recalculation. By contrast, algorithm MHFS retrains a changed data set, which does not use knowledge generated in the original data set, and performs numerous repeated calculations. The computational time required to obtain a feasible reduct by algorithm MIFSA is considerably shorter than that required by algorithm MHFS. Hence, algorithm MIFSA can effectively obtain a feasible reduct without reducing the classification accuracy of the raw feature set.

6.2. Performance evaluations of algorithm MIFSD when deleting multiple objects

In this subsection, we compare the effectiveness of algorithms MIFSD and MHFS from two aspects: reduct size and its classification accuracy. The efficiency of algorithms MIFSD and MHFS is compared in terms of computational time and speed-up ratio. The specific experimental design is described as follows.

6.2.1. Effectiveness evaluations

In this subsection, we compare the effectiveness of algorithms MIFSD and MHFS. For each data set in Table 6, we randomly select the 50% objects as deleted objects. The details of the experiments are given as follows.

(1) Comparison of algorithms MIFSD and MHFS in terms of reduct size
We compare the effectiveness of algorithms MIFSD and MHFS in this subsection. For each data set in Table 6, we randomly select 50% of the objects as deleted objects. The details of the experiments are presented as follows.
Table 9 shows that the NFR and reduct generated by algorithms MHFS and MIFSD are extremely close. In fact, NFR and reduct are identical in some data sets, such as Iris, Wine, Car, Aust, AD1, and AD2. Moreover, the FRR of the two algo-

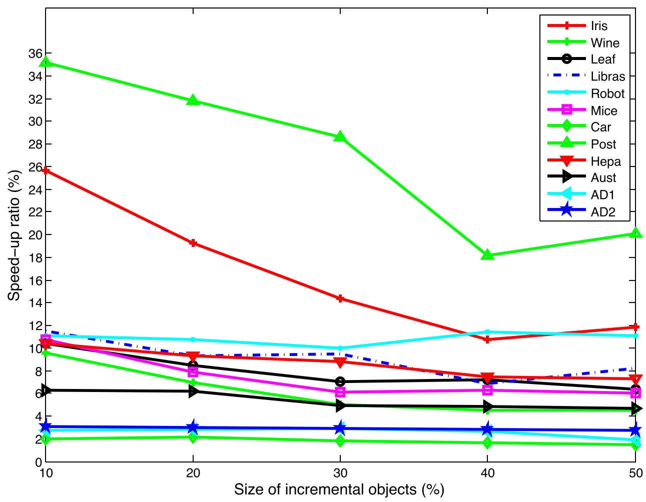

Fig. 3. Speed-up ratio of algorithm MIFSA.

rithms is extremely close, with most data sets reaching more than 60%. This finding indicates that the proposed feature selection strategies based on neighborhood dominance relation are effective, and they can remove most redundant features. However, the running time of algorithm MIFSD is considerably shorter than that of algorithm MHFS. The TRR of most data sets is above 90%. In particular, the values of TRR are all above 95% and the average value of TRR reaches 89.8% in the data sets Iris, Wine, Leaf, Post, and Hepa. This result indicates that algorithm MIFSD can compute a reduct within a considerably shorter time.

(2) Comparison of algorithms MIFSD and MHFS in terms of classification accuracy
Here, we compare the classification performance of the reducts generated using algorithms MIFSD, reducts generated using algorithms MHFS, and the raw feature set. On the basis of a 10-fold cross-validation, we use the classifiers OCC, J48, and RT in Weka [10] to calculate the classification accuracy of the raw feature set and the selected reducts in Table 9. The experimental results are recorded in Table 10.

As shown in Table 10, the classification accuracy of the reducts generated using algorithms MHFS and MIFSD in most data sets is extremely close to or slightly higher than that of the raw feature set. This finding proves that the neighborhood dominance relation-based feature selection strategy can delete redundant features in a HODS and maintain or even improve the classification accuracy of data sets. Moreover, the classification accuracy of the reducts generated using algorithm MIFSD in the three classifiers is extremely close to or even slightly higher than that of the reducts generated using algorithm MHFS for most data sets, as fully illustrate by their average values. Thus, the experimental results indicates that algorithm MIFSD can generate a feasible reduct.

6.2.2. Efficiency evaluations

In this subsection, we compare algorithms MIFSD and MHFS in terms of computational time and speed-up ratio. For each data set in Table 6, different test sets are constructed by deleting varying proportion of objects, i.e., 10%, 20%, 30%, 40%, and 50% of the original objects. Therefore, we can obtain five new data sets with different sizes, which contain 90%, 80%, 70%, 60%, and 50% of the objects from the original data set to prepare for the subsequent experiments.

Table 9
The comparison of reducts of algorithms MIFSD and MHFS.

Data sets	MHFS				MIFSD				TRR (%)
	NFR	Reduct	FRR (%)	Times (ms)	NFR	Reduct	FRR (%)	Times (ms)	
Iris	1	3	75.0	281	1	3	75.0	1	99.6
Wine	5	4,8,10,11,12	61.5	332	5	4,8,10,11,12	61.5	8	97.6
Leaf	4	5,6,9,12	73.3	491	4	2,5,9,12	73.3	21	95.7
Libras	9	1,6,19,33,42,64,75,80,90	90.0	3548	11	2,7,27,33,53,54,60,70,80,85,90	87.8	392	89.0
Robot	5	1,2,16,45,72	94.4	3975	4	39,40,42,85	95.6	279	93.0
Mice	13	29,30,34,40,41,44,45,46,49,54,57,65,68	80.9	32013	10	1,30,34,40,41,44,46,49,57,68	85.3	2794	91.2
Car	5	1,2,4,5,6	16.7	820	5	1,2,4,5,6	16.7	150	81.7
Post	2	3,6	75.0	283	2	6,7	75.0	1	99.6
Hepa	1	1	94.7	320	1	8	94.7	4	98.8
Aust	3	1,8,12	78.6	629	3	1,8,12	78.6	54	91.4
AD1	3	1,4,7	70.0	8797	3	1,4,7	70.0	2729	69.0
AD2	3	1,4,7	70.0	11806	3	1,4,7	70.0	3462	70.7
Average	4.50	–	73.3	5275	4.33	–	73.6	825	89.8

Table 10
The comparison of algorithms MIFSD and MHFS on classification accuracy (%).

Data sets	OCC			J48			TR		
	Raw	MHFS	MIFSD	Raw	MHFS	MIFSD	Raw	MHFS	MIFSD
Iris	97.33	100	**100**	97.33	100	**100**	100	100	**100**
Wine	80.90	92.13	**92.13**	**94.38**	91.01	91.01	**92.13**	88.76	88.76
Leaf	**26.47**	17.06	22.94	51.76	40.00	**52.94**	**58.24**	41.76	52.94
Libras	**44.44**	32.22	31.11	51.11	52.22	**52.78**	57.22	54.44	**57.56**
Robot	31.17	**37.66**	33.33	**43.29**	42.86	42.86	32.03	**34.63**	32.03
Mice	**70.07**	66.54	66.80	75.84	74.72	**78.10**	71.56	73.79	**79.44**
Car	96.19	97.23	**97.23**	96.19	97.23	**97.23**	92.96	96.19	**96.19**
Post	81.82	81.82	**81.82**	81.82	81.82	**81.82**	63.64	81.82	**81.82**
Hepa	97.44	97.44	**97.44**	97.44	97.44	**97.44**	97.44	97.44	**97.44**
Aust	82.32	84.93	**85.51**	82.32	84.93	**85.51**	82.90	84.64	**84.64**
AD1	96.02	96.13	**96.13**	**96.60**	96.42	96.42	**96.19**	95.10	95.10
AD2	96.50	96.53	**96.53**	**97.00**	96.47	96.47	**96.41**	95.18	95.18
Average	75.06	74.97	**75.08**	80.42	79.59	**81.05**	78.39	78.65	**80.09**

(1) Comparison of algorithms MIFSD and MHFS in terms of computational time
Multiple objects are proportionally deleted from the original data set for each data set listed in Table 6. The detailed change trend lines of the two algorithms with increasing size of data sets are shown in Fig. 4, where the abscissa represents the size of the deleted data sets and the ordinate represents the computational time in each sub-figure. The computational time of algorithm MHFS is depicted as lines with hollow dots, and that of algorithm MIFSD is depicted as lines with solid dots.
Fig. 4 clearly shows that the computational time of algorithms MIFSD and MHFS decreases as the size of the deleted object set increases. From each sub-figure, the computational time of algorithm MIFSD is significantly shorter than that of algorithm MHFS. This finding indicates that algorithm MIFSD can compute a reduct within a considerably shorter time.

(2) Evaluation of the efficiency of algorithm MIFSD in terms of speed-up ratio
The efficiency of algorithm MIFSD is illustrated in term of the speed-up ratio. The experimental results are presented in Fig. 5, where the abscissa represents the size of the deleted data sets and the ordinate represents the speed-up ratio. We can observe from this figure that for the data sets Iris, Post, and hepa, the time taken by algorithm MHFS considerably exceeds that of algorithm MIFSD. Hence, we only shown the speed-up ratio of the remaining nine data sets in Fig. 5.

Fig. 5 shows that algorithm MIFSD is at least nearly four times faster than algorithm MHFS for all the data sets. Moreover, when data sizes is reduced, the speed-up of most data sets remains stable. Furthermore, we notice that algorithm MIFSD is approximately 14 times faster than algorithm MHFS on average. The experimental results verify that algorithm MIFSD exhibits better efficiency in computing the reduct in a HODS than algorithm MHFS.

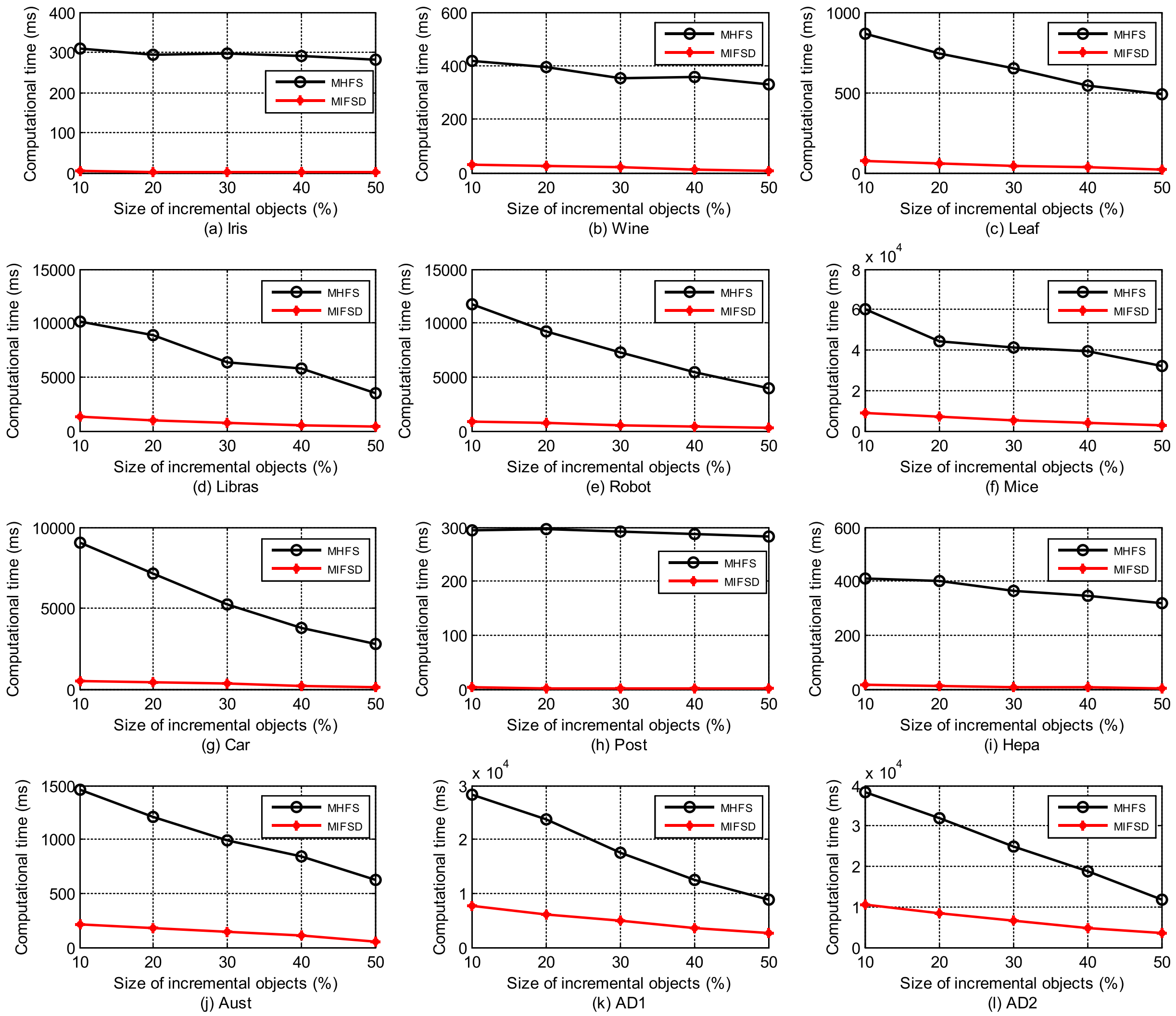

Fig. 4. The comparison of computational time between algorithms MHFS and MIFSD versus different ratio size of deleting objects.

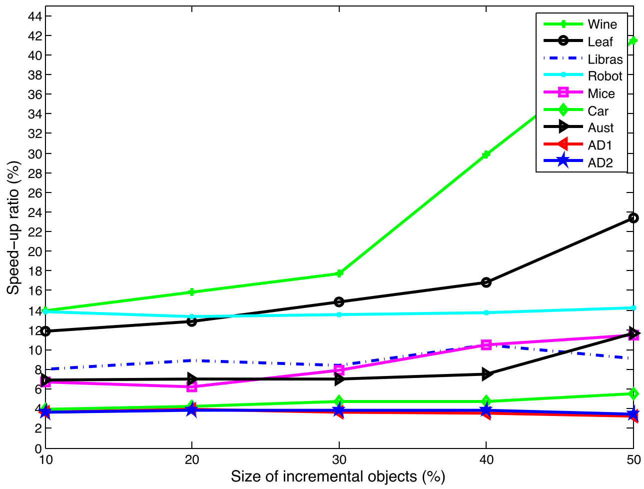

Fig. 5. Speed-up ratio of algorithm MIFSD.

6.2.3. Summary

We can draw the conclusion that the incremental algorithm outperforms algorithm MHFS by evaluating the effectiveness and efficiency of algorithm MIFSD. The incremental mechanism is fully investigated by obtaining the neighborhood dominance relation matrix and its diagonal matrix when multiple objects are deleted from a HODS. Similar to the case of adding multiple objects, quickly updating the neighborhood dominance relation matrix and its diagonal matrix can improve the efficiency of calculating MNDCE, enhancing the efficiency of obtaining the reduct of a HODS. Algorithm MIFSD obtains a new reduct on the basis of previous knowledge, avoiding recalculation. By contrast, algorithm MHFS is used to retrain the changed data set as a new one and does not use knowledge generated from the original data set. Thus, algorithm MHFS performs numerous repeated calculations. The computational time required to obtain a feasible reduct by algorithm MIFSD is considerably shorter than that required by algorithm MHFS. Hence, a feasible reduct can be obtained more efficiently by algorithm MIFSD than by algorithm MHFS.

7. Conclusion

DNRS considers the degree of preference on the basis of DRSA, which can robustly handle MCDM of heterogeneous data. This study investigated incremental heterogeneous feature selection approaches for dynamic ordered data sets in DNRS framework. The incremental feature selection algorithms MIFSA and MIFSD proposed in this work are composed of two aspects: non-monotonic feature selection strategy and the updating mechanisms of MNDCE. Experiments are conducted to compare the effectiveness and efficiency of the proposed incremental algorithms with a non-incremental algorithm. The experimental results show that the proposed algorithms can quickly obtain an effective reduct from dynamic heterogeneous ordered data sets. However, the criteria may be fuzzy in MCDM. DNRS does not consider the fuzziness of the criteria, motivating our further research. In the future, we will study fuzzy dominance relation based neighborhood rough set and develop an incremental feature selection algorithm based on it.

CRediT authorship contribution statement

Binbin Sang: Methodology, Validation, Writing - original draft, Writing - review & editing, Software, Data curation. **Hongmei Chen:** Conceptualization, Resources, Visualization, Supervision, Project administration, Funding acquisition. **Tianrui Li:** Resources, Supervision, Funding acquisition. **Weihua Xu:** Resources, Supervision. **Hong Yu:** Resources.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (Nos. 61976182, 61572406, 61573292, 61602327, 61876157, 61976245), Key Program for International S&T Cooperation of Sichuan Province (No.

2019YFH0097), Applied Basic Research Programs of Science and Technology Department of Sichuan Province (No. 2019YJ0084).

Appendix A

In this part, we list the corresponding explanations of the symbols and abbreviations used in this paper in Tables 11 and 12, respectively.

Table 11
The description of symbols.

U	A non-empty finite set of objects	$\underline{N_{P_\delta}}\left(Cl_t^{\succeq}\right)/\underline{N_{P_\delta}}\left(Cl_t^{\preceq}\right)$	The lower approximations of $Cl_t^{\succeq}/Cl_t^{\preceq}$ on P
C	A conditional feature set	$NH^{\succeq}_{d\|P_\delta}(U)$	The neighborhood dominance conditional entropy of P to d
P/B	A conditional feature subset	$sig^{\succeq U}_{inner}(a,P,d)$	The inner significance of a in P
d	A decision feature	$sig^{\succeq U}_{outer}(a,P,d)$	The outer significance of a to P
$H^{\succeq}$	A heterogeneous ordered decision system	$\mathbb{R}_U^{\succeq P_\delta}$	The neighborhood dominance relation matrix on P
λ_1, λ_2	Two given parameters ($0 < \lambda_1 < 1$, $\lambda_2 > 1$)	$\mathbb{R}_U^{\succeq P_\delta}$	The neighborhood dominance relation matrix on P
$N^{\succeq}_{P_\delta}$	The neighborhood dominance relation on P	$r^{P_\delta}_{(i,j)}$	The characteristic function of the $\mathbb{R}_U^{\succeq P_\delta}$
$\hat{d}_P(x,y)$	The distance between x and y on P	$\mathbb{D}_U^{\succeq P_\delta}$	The diagonal matrix matrix of the $\mathbb{R}_U^{\succeq P_\delta}$
δ	A given threshold $\delta \in [0,1]$	$d^{P_\delta}_{(i,j)}$	The characteristic function of the $\mathbb{D}_U^{\succeq P_\delta}$
$N_d^{\succeq}$	The dominance relation on d	$\left(\mathbb{D}_U^{\succeq P_\delta}\right)^{-1}$	The inverse matrix of the $\mathbb{D}_U^{\succeq P_\delta}$
$N^{+}_{P_\delta}(x)/N^{-}_{P_\delta}(x)$	The neighborhood dominating/dominated set of x on P	$MNH^{\succeq}_{d\|P_\delta}(U)$	The matrix-based neighborhood dominance conditional entropy
$N_d^{+}(x)/N_d^{-}(x)$	The dominating/dominated set of x on d	$Msig^{\succeq U}_{inner}(a,B,d)$	The matrix-based inner significance of a in B
Cl_t	A decision class	$Msig^{\succeq U}_{outer}(a,B,d)$	The matrix-based outer significance of a to B
$Cl_t^{\succeq}/Cl_t^{\preceq}$	The upward/downward union of decision class Cl_t	Red_U	A reduct on U
$\overline{N_{P_\delta}}\left(Cl_t^{\succeq}\right)/\overline{N_{P_\delta}}\left(Cl_t^{\preceq}\right)$	The upper approximations of $Cl_t^{\succeq}/Cl_t^{\preceq}$ on P		

Table 12
The description of abbreviations.

RST	Rough set theory	MIFSA	Matrix-based incremental feature selection algorithm while adding multiple objects
DRSA	Dominance-based rough set approach	MIFSD	Matrix-based incremental feature selection algorithm while deleting multiple objects
MCDM	Multi-criteria decision-making	NFR	Number of features in the reduct
HODS	Heterogeneous ordered decision system	FRR	Feature reduction rate
DNRS	Dominance-based neighborhood rough set	TRR	Time reduction rate
NDCE	Neighborhood dominance conditional entropy	OCC	Ordinal class classifier
MNDCE	Matrix-based neighborhood dominance conditional entropy	RT	Random tree
MHFS	Matrix-based heuristic feature selection algorithm	UCI	University of california at irvine

References

[1] Mingjie Cai, Guangming Lang, Hamido Fujita, Zhenyu Li, Tian Yang, Incremental approaches to updating reducts under dynamic covering granularity, Knowl.-Based Syst. 172 (2019) 130–140.
[2] Degang Chen, Yanyan Yang, Attribute reduction for heterogeneous data based on the combination of classical and fuzzy rough set models, IEEE Trans. Fuzzy Syst. 22 (5) (2014) 1325–1334.
[3] Degang Chen, Yanyan Yang, Ze Dong, An incremental algorithm for attribute reduction with variable precision rough sets, Appl. Soft Comput. 45 (2016) 129–149.
[4] Hongmei Chen, Tianrui Li, Yong Cai, Chuan Luo, Hamido Fujita, Parallel attribute reduction in dominance-based neighborhood rough set, Inf. Sci. 373 (2016) 351–368.
[5] Hongmei Chen, Tianrui Li, Da Ruan, Jianhui Lin, Chengxiang Hu, A rough-set-based incremental approach for updating approximations under dynamic maintenance environments, IEEE Trans. Knowl. Data Eng. 25 (2) (2013) 274–284.
[6] Davide Ciucci, Yiyu Yao, Synergy of granular computing, shadowed sets, and three-way decisions, Inf. Sci. 508 (2020) 422–425.
[7] Jianhua Dai, Qinghua Hu, Hu Hu, Debiao Huang, Neighbor inconsistent pair selection for attribute reduction by rough set approach, IEEE Trans. Fuzzy Syst. 26 (2) (2018) 937–950.
[8] Wensheng Du, Baoqing Hu, A fast heuristic attribute reduction approach to ordered decision systems, Eur. J. Oper. Res. 264 (2) (2018) 440–452.
[9] Eibe Frank, Mark Hall, A simple approach to ordinal classification, Lect. Notes Comput. Sci. 2167 (08) (2001) 145–156.
[10] Eibe Frank, Mark Hall, Geoffrey Holmes, Richard Kirkby, Bernhard Pfahringer, Ian H Written, Len Trigg, Weka-a machine learning workbench for data mining, Data Mining Knowl. Discovery Handbook. 2 (2010) 1269–1277.

[11] Salvatore Greco, Benedetto Matarazzo, Roman Slowinski, Rough sets theory for multicriteria decision analysis, Eur. J. Oper. Res. 129 (1) (2001) 1–47.
[12] Qinghua Hu, Xunjian Che, Lei Zhang, David Zhang, Maozu Guo, Daren Yu, Rank entropy based decision trees for monotonic classification, IEEE Trans. Knowl. Data Eng. 24 (11) (2012) 2052–2064.
[13] Qinghua Hu, Daren Yu, Jinfu Liu, Congxin Wu, Neighborhood rough set based heterogeneous feature subset selection, Inf. Sci. 178 (18) (2008) 3577–3594.
[14] Bing Huang, Weizhi Wu, Jinjiang Yan, Huaxiong Li, Xianzhong Zhou, Inclusion measure-based multi-granulation decision-theoretic rough sets in multi-scale intuitionistic fuzzy information tables, Inf. Sci. 507 (2020) 421–448.
[15] Qianqian Huang, Tianrui Li, Yanyong Huang, Xin Yang, Hamido Fujita, Dynamic dominance rough set approach for processing composite ordered data, Knowl.-Based Syst. 187 (2020) 1–17.
[16] Yanyong Huang, Tianrui Li, Chuan Luo, Hamido Fujita, Shijinn Horng, Matrix-based dynamic updating rough fuzzy approximations for data mining, Knowl.-Based Syst. 119 (2017) 273–283.
[17] Yunge Jing, Tianrui Li, Hamido Fujita, Zeng Yu, Bin Wang, An incremental attribute reduction approach based on knowledge granularity with a multi-granulation view, Inf. Sci. 411 (2017) 23–38.
[18] Guangming Lang, Duoqian Miao, Mingjie Cai, Zhifei Zhang, Incremental approaches for updating reducts in dynamic covering information systems, Knowl.-Based Syst. 134 (2017) 85–104.
[19] Jiye Liang, Feng Wang, Chuangyin Dang, Yuhua Qian, A group incremental approach to feature selection applying rough set technique, IEEE Trans. Knowl. Data Eng. 26 (2014) 294–308.
[20] Yaojin Lin, Yuwen Li, Chenxi Wang, Jinkun Chen, Attribute reduction for multi-label learning with fuzzy rough set, Knowl.-Based Syst. 152 (2018) 51–61.
[21] Ranjit Panigrahi, Samarjeet Borah, Rank allocation to j48 group of decision tree classifiers using binary and multiclass intrusion detection datasets, Procedia Computer Sci. 132 (2018) 323–332.
[22] Zdzisaw Pawlak, Rough sets, Int. J. Computer Inform. Sci. 11 (5) (1982) 341–356.
[23] Lech Polkowski, A note on 3-valued rough logic accepting decision rules, Fundamenta Informaticae. 61 (1) (2004) 37–45.
[24] Lech Polkowski, Rough-fuzzy-neurocomputing based on rough mereological calculus of granules, Int. J. Hybrid Intelligent Syst. 2 (2) (2005) 91–108.
[25] Lech Polkowski. Granulation of knowledge by tools of rough mereology. IEEE International Conference on Granular Computing, 8(2010)35–36.
[26] Polkowski Lech, Artiemjew Piotr, Granular computing in the frame of rough mereology. A case study: classification of data into decision categories by means of granular reflections of data, International Journal of Intelligent Systems. 26 (6) (2011) 555–571..
[27] Skowron Andrzej, Rough mereology: a new paradigm for approximate reasoning, Int. J. Approximate Reasoning. 15 (4) (1996) 333–365.
[28] Yuhua Qian, Honghong Cheng, Jieting Wang, Jiye Liang, Witold Pedrycz, Chuangyin Dang, Grouping granular structures in human granulation intelligence, Inf. Sci. 383 (2017) 150–169.
[29] Yuhua Qian, Chuangyin Dang, Jiye Liang, Dawei Tang, Set-valued ordered information systems, Inf. Sci. 179 (16) (2009) 2809–2832.
[30] Yuhua Qian, Jiye Liang, Combination entropy and combination granulation in rough set theory, Int. J. Uncertainty, Fuzziness Knowl. -Based Syst. 16 (2008) 179–193.
[31] Yuhua Qian, Hu Zhang, Feijiang Li, Qinghua Hu, Jiye Liang, Set-based granular computing: A lattice model, Int. J. Approximate Reasoning. 55 (3) (2014) 834–852.
[32] Claude Shannon, Warren Weaver, The mathematical theory of communication, Bell System Tech. J. 27 (3/4) (1948) 373–423.
[33] Yanhong She, Xiaoli He, Huixian Shi, Yuhua Qian, A multiple-valued logic approach for multigranulation rough set model, Int. J. Approximate Reasoning. 82 (2017) 270–284.
[34] Wenhao Shu, Wenbin Qian, Yonghong Xie, Incremental approaches for feature selection from dynamic data with the variation of multiple objects, Knowl.-Based Syst. 163 (2019) 320–331.
[35] Wenhao Shu, Hong Shen, Incremental feature selection based on rough set in dynamic incomplete data, Pattern Recogn. 47 (12) (2014) 3890–3906.
[36] Lin Sun, Xiaoyu Zhang, Yuhua Qian, Jiucheng Xu, Shiguang Zhang, Feature selection using neighborhood entropy-based uncertainty measures for gene expression data classification, Inf. Sci. 502 (2019) 18–41.
[37] Changzhong Wang, Mingwen Shao, Qiang He, Yuhua Qian, Yali Qi, Feature subset selection based on fuzzy neighborhood rough sets, Knowl.-Based Syst. 111 (2016) 173–179.
[38] Feng Wang, Jiye Liang, Chuangyin Dang, Attribute reduction for dynamic data sets, Applied Soft Comput. 13 (1) (2013) 676–689.
[39] Feng Wang, Jiye Liang, Yuhua Qian, Attribute reduction: a dimension incremental strategy, Knowl.-Based Syst. 39 (2013) 95–108.
[40] Shu Wang, Tianrui Li, Chuan Luo, Hongmei Chen, Hamido Fujita, Domain-wise approaches for updating approximations with multi-dimensional variation of ordered information systems, Inf. Sci. 478 (2019) 100–124.
[41] Wei Wei, Xiaoying Wu, Jiye Liang, Junbiao Cui, Yijun Sun, Discernibility matrix based incremental attribute reduction for dynamic data, Knowl.-Based Syst. 140 (2018) 142–157.
[42] Xin Yang, Tianrui Li, Dun Liu, Hamido Fujita, A multilevel neighborhood sequential decision approach of three-way granular computing, Inf. Sci. 538 (2020) 119–141.
[43] Yanyan Yang, Degang Chen, Hui Wang, Active sample selection based incremental algorithm for attribute reduction with rough sets, IEEE Trans. Fuzzy Syst. 25 (4) (2017) 825–838.
[44] Yanyan Yang, Degang Chen, Hui Wang, Eric C.C. Tsang, Deli Zhang, Fuzzy rough set based incremental attribute reduction from dynamic data with sample arriving, Fuzzy Sets Syst. 312 (2017) 66–86.
[45] Yiyu Yao, Three-way granular computing, rough sets, and formal concept analysis, Int. J. Approximate Reasoning. 116 (2020) 106–125.
[46] Jianhang Yu, Minghao Chen, Weihua Xu, Dynamic computing rough approximations approach to time-evolving information granule interval-valued ordered information system, Appl. Soft Comput. 60 (2017) 18–29.
[47] Zhong Yuan, Xianyong Zhang, Shan Feng, Hybrid data-driven outlier detection based on neighborhood information entropy and its developmental measures, Expert Syst. Appl. 112 (2018) 243–257.
[48] Anping Zeng, Tianrui Li, Dun Liu, Junbo Zhang, Hongmei Chen, A fuzzy rough set approach for incremental feature selection on hybrid information systems, Fuzzy Sets Syst. 258 (2015) 39–60.
[49] Xianyong Zhang, Jilin Yang, Lingyu Tang, Three-way class-specific attribute reducts from the information viewpoint, Inf. Sci. 507 (2020) 840–872.
[50] Xiao Zhang, Changlin Mei, Degang Chen, Jinhai Li, Feature selection in mixed data: A method using a novel fuzzy rough set-based information entropy, Pattern Recogn. 56 (2016) 1–15.