Research Article
On Distribution Reduction and Algorithm Implementation in Inconsistent Ordered Information Systems

Yanqin Zhang
School of Economics, Xuzhou Institute of Technology, Xuzhou, Jiangsu 221008, China

Correspondence should be addressed to Yanqin Zhang; zhiyuan1111@163.com

Received 19 May 2014; Accepted 11 August 2014; Published 28 August 2014

Academic Editor: Weihua Xu

Copyright © 2014 Yanqin Zhang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As one part of our work in ordered information systems, distribution reduction is studied in inconsistent ordered information systems (OISs). Some important properties on distribution reduction are studied and discussed. The dominance matrix is restated for reduction acquisition in dominance relations based information systems. Matrix algorithm for distribution reduction acquisition is stepped. And program is implemented by the algorithm. The approach provides an effective tool for the theoretical research and the applications for ordered information systems in practices. For more detailed and valid illustrations, cases are employed to explain and verify the algorithm and the program which shows the effectiveness of the algorithm in complicated information systems.

1. Introduction

In Pawlak’s original rough set theory [1], partition or equivalence (indiscernibility) is an important and primitive concept. However, partition or equivalence relation, as the indiscernibility relation in Pawlak’s original rough set theory, is still restrictive for many applications. To address this issue, several interesting and meaningful extensions to equivalence relation have been proposed in the past, such as neighborhood operators [2], tolerance relations [3], and others [4–10]. Moreover, the original rough set theory does not consider attributes with preference ordered domain, that is, criteria. In many real life practices, we often face problems in which the ordering of properties of the considered attributes plays a crucial role. One such type of problem is the ordering of objects. For this reason, Greco et al. proposed an extension rough set theory, called the dominance based rough set approach (DRSA), to take into account the ordering properties of criteria [11–16]. This innovation is mainly based on substitution of the indiscernibility relation by a dominance relation. Moreover, Greco et al. characterizes the DRSA and decision rules induced from rough approximations, while the usefulness of the DRSA and its advantages over the CRS (classical rough set approach) are presented [11–16]. In DRSA, condition attributes are criteria and classes are preference ordered. Several studies have been made about properties and algorithmic implementations of DRSA [10, 17–19].

Nevertheless, only a limited number of methods using DRSA to acquire knowledge in inconsistent ordered information systems have been proposed and studied. Pioneering work on inconsistent ordered information systems with the DRSA has been proposed by Greco et al. [11–16], but they did not clearly point out the semantic explanation of unknown values. Shao and Zhang [20] further proposed an extension of the dominance relation in incomplete ordered information systems. Their work was established on the basis of the assumption that all unknown values are lost. Despite this, they did not mention the underlying concept of attribute reduction in inconsistent ordered decision system but they mentioned an approach to attribute reduction in consistent ordered information systems. Therefore, the purpose of this paper is to develop approaches to attribute reductions in inconsistent ordered information systems (IOIS). In this paper, theories and approaches of distribution reduction are investigated in inconsistent ordered information systems. Furthermore, algorithm of matrix computation of distribution reduction is introduced, from which we can provide a new approach to attributes reductions in inconsistent ordered information systems.
The rest of this paper is organized as follows. Some preliminary concepts are briefly recalled in Section 2. In Section 3, theories and approaches of distribution reduction are investigated in IOIS. In Section 4, we restate the definition of dominance matrix in ordered information systems and step the matrix algorithm for distribution reduction acquisition. Preparations are implemented to place the algorithm and the program is designed. The algorithm and the corresponding program we designed can provide a tool to theoretical research and applications of criterion based algorithm and the program are effective in complicated information system. Cases are employed to illustrate the theoretical research and applications of criterion based algorithm and program we designed can provide a tool to theoretical research and applications of criterion based algorithm and program are effective in complicated information system. Furthermore conclusions on what we study in this paper are drawn to understand this paper briefly.

2. Ordered Information Systems

The following recalls necessary concepts and preliminaries required in the sequel of our work. Detailed description of the theory can be found in [11-16].

An information system with decisions is an ordered quadruple \( \mathcal{I} = (U, A \cup D, F, G) \), where \( U = \{x_1, x_2, \ldots, x_n\} \) is a nonempty finite set of objects; \( A \cup D \) is a nonempty finite attributes set; \( A = \{a_1, a_2, \ldots, a_p\} \) denotes the set of condition attributes; \( D = \{d_1, d_2, \ldots, d_q\} \) denotes the set of decision attributes, \( A \cap D = \emptyset; F = \{f_k : U \rightarrow V_k, k \leq p\} \), \( f_k(x) \) is the value of \( d_k \) on \( x \in U \); \( V_k \) is the domain of \( d_k \); \( A, D \) and \( G = \{g_l : U \rightarrow V_l, l \leq q\} \), \( g_l(x) \) is the value of \( d_l \) on \( x \in U \); \( V_l \) is the domain of \( d_l \). In an information system, if the domain of an attribute is ordered according to a decreasing or increasing preference, then the attribute is a criterion. An information system is called an ordered information system (OIS) if all condition attributes are criterions.

Assume that the domain of a criterion \( a \in A \) is completely preordered by an outranking relation \( \succeq \); then \( x \succeq y \) means that \( x \) is at least as good as \( y \) with respect to criterion \( a \). And we can say that \( x \) dominates \( y \). In the following, without any loss of generality, we consider condition and decision criterions having a numerical domain; that is, \( V_a \subseteq \mathbb{R} \) (\( \mathbb{R} \) denotes the set of real numbers).

We define \( x \succeq y \) by \( f(x, a) \geq f(y, a) \) according to increasing preference, where \( a \in A \) and \( x, y \in U \). For a subset of attributes \( B \subseteq A \), \( x \succeq_B y \) means that \( x \succeq a y \) for any \( a \in B \). That is to say \( x \) dominates \( y \) with respect to all attributes in \( B \). Furthermore, we denote \( x \succ_B y \) by \( x \succ_B y \). In general, we indicate an ordered information system with decision by \( \mathcal{I} = (U, A \cup D, F, G) \). Thus the following definition can be obtained.

Let \( \mathcal{I} = (U, A \cup D, F, G) \) be an ordered information system with decisions, for \( B \subseteq A \); denote

\[
R_B^\succeq = \{ (x_j, x_i) \in U \times U | f_j(x_j) \geq f_i(x_i), \forall a_i \in B \} ;
\]

\[
R_D^\succeq = \{ (x_j, x_i) \in U \times U | g_m(x_j) \geq g_m(x_i), \forall d_m \in D \} .
\]

(1)

\( R_B^\succeq \) and \( R_D^\succeq \) are called dominance relations of information system \( \mathcal{I} \).

Table 1: An ordered information system.

| \( x_1 \) | \( a_1 \) | \( a_2 \) | \( a_3 \) | \( d \) |
|---|---|---|---|---|
| 1 | 2 | 1 | 3 |
| 2 | 3 | 2 | 2 |
| 3 | 1 | 1 | 1 |
| 4 | 2 | 3 | 2 |
| 5 | 3 | 3 | 2 |
| 6 | 3 | 2 | 1 |

If we denote

\[
[x_j]_{B}^\succeq = \{ x_j \in U | (x_j, x_i) \in R_B^\succeq \} ;
\]

\[
[x_i]_{D}^\succeq = \{ x_i \in U | (x_j, x_i) \in R_D^\succeq \} ,
\]

then the following properties of a dominance relation are trivial.

Let \( R_A^\succeq \) be a dominance relation. The following properties hold.

1. \( R_A^\succeq \) is reflexive and transitive, but not symmetric, so it is not an equivalence relation.
2. If \( B \subseteq A \), then \( R_A^\succeq \subseteq R_B^\succeq .
3. If \( B \subseteq A \), then \( [x_i]_{A}^\succeq \subseteq [x_i]_{B}^\succeq .
4. If \( x_j \in [x_i]_{A}^\succeq \), then \( [x_i]_{A}^\succeq \subseteq [x_j]_{A}^\succeq \) and \( [x_j]_{A}^\succeq = \cup [x_i]_{A}^\succeq \).
5. \( [x_i]_{A}^\succeq = [x_j]_{A}^\succeq \) if and only if \( f(x_i, a) = f(x_j, a) \) \((\forall a \in A)\).
6. \( \mathcal{I} = \cup [x_i]_{A}^\succeq \) constitute a covering of \( U \).

For any subset \( X \) of \( U \), and \( A \) of \( \mathcal{I} \), define

\[
R_A^\succeq (X) = \{ x \in U | [x]_{A}^\succeq \subseteq X \} ,
\]

(3)

\[
R_A^\succeq (X) = \{ x \in U | [x]_{A}^\succeq \cap X \neq \emptyset \} .
\]

\( R_A^\succeq (X) \) and \( R_A^\succeq (X) \) are said to be the lower and upper approximations of \( X \) with respect to a dominance relation \( R_A^\succeq \). And the approximations have also some properties which are similar to those of Pawlak approximation spaces.

For an ordered information system with decisions \( \mathcal{I} = (U, A \cup D, F, G) \), if \( R_A^\succeq \subseteq R_B^\succeq \), then this information system is consistent, otherwise, this information system is inconsistent (IOIS).

Example 1. An ordered information system is given in Table 1.

From the table, we have

\[
[x_1]_{A}^\succeq = \{ x_1, x_2, x_3, x_5, x_6 \} ;
\]

\[
[x_2]_{A}^\succeq = \{ x_2, x_5, x_6 \} ;
\]

\[
[x_3]_{A}^\succeq = \{ x_2, x_3, x_4, x_5, x_6 \} ;
\]

\[
[x_4]_{A}^\succeq = \{ x_4, x_6 \} ;
\]
\[ [x_5]^c_A = \{ x_5 \} ; \quad [x_6]^c_A = \{ x_6 \} ; \]
\[ [x_1]^c_d = [x_2]^c_d = \{ x_1, x_3 \} ; \]
\[ [x_2]^c_d = [x_4]^c_d = \{ x_1, x_2, x_4, x_5 \} ; \]
\[ [x_3]^c_d = [x_6]^c_d = \{ x_1, x_2, x_3, x_4, x_5, x_6 \} . \]

Definition 2. Let \( \gamma \) where
\[ B \subseteq A \] and \( d \) respectively. For \( B \subseteq A \), we say that \( B \) is a distribution consistent set if \( \gamma_B(x) = \mu_B(x) \) for all \( x \in U \), we say that \( B \) is a distribution consistent set, and no proper subset of \( B \) is a distribution consistent set, then \( B \) is called a distribution consistent reduction of \( \mathcal{F} \).

Definition 5. Let \( \mathcal{F}^c = (U, A \cup D, F, G) \) be an inconsistent information system. If \( \gamma^c_B(x) = \gamma^c_A(x) \) for all \( x \in U \), we say that \( B \) is a maximum distribution consistent set of \( \mathcal{F}^c \). If \( B \) is a maximum distribution consistent set, and no proper subset of \( B \) is a maximum distribution consistent set, then \( B \) is called a maximum distribution consistent reduction of \( \mathcal{F}^c \).

Example 6. For the system in Table 1, if we denote
\[ D_1 = [x_1]^c_d = [x_5]^c_d, \]
\[ D_2 = [x_2]^c_d = [x_4]^c_d, \]
\[ D_3 = [x_3]^c_d = [x_6]^c_d, \]
then we can have
\[ \mu_A^c(x_1) = \left( \frac{1}{3}, \frac{1}{2}, \frac{2}{3} \right); \]
\[ \mu_A^c(x_2) = \left( \frac{1}{6}, \frac{1}{3}, \frac{1}{2} \right); \]
\[ \mu_A^c(x_3) = \left( \frac{1}{6}, \frac{1}{2}, \frac{5}{6} \right); \]
\[ \mu_A^c(x_4) = \left( 0, \frac{1}{6}, \frac{1}{3} \right); \]
\[ \mu_A^c(x_5) = \left( \frac{1}{6}, \frac{1}{6}, \frac{1}{2} \right); \]
\[ \mu_A^c(x_6) = \left( 0, 0, \frac{1}{6} \right), \]
\[ \gamma_A^c(x_1) = \frac{2}{3}; \quad \gamma_A^c(x_2) = \frac{1}{2}; \quad \gamma_A^c(x_3) = \frac{5}{6}; \]
\[ \gamma_A^c(x_4) = \frac{1}{3}; \quad \gamma_A^c(x_5) = \frac{1}{6}; \quad \gamma_A^c(x_6) = \frac{1}{6}. \]

When \( B = \{a_2, a_3\} \), it can be easily checked that \( [x_1]^c_A = [x_1]^c_B \), for all \( x \in U \), so that \( \mu_A^c(x) = \mu_B^c(x) \) and \( \gamma_A^c(x) = \gamma_B^c(x) \) are true and \( B = \{a_2, a_3\} \) is a distribution consistent set of \( \mathcal{F}^c \). Furthermore, we can examine that \( \{a_2\} \) and \( \{a_3\} \) are not consistent sets of \( \mathcal{F}^c \). That is to say \( B = \{a_2, a_3\} \) is a distribution reduction and is a maximum distribution reduction of \( \mathcal{F}^c \).

Moreover, it can easily be calculated that \( B' = \{a_1, a_3\} \) and \( B'' = \{a_1, a_2\} \) are not distribution consistent sets of \( \mathcal{F}^c \). Thus there exist only one distribution reduction and maximum distribution reduction of \( \mathcal{F}^c \) in the system of Table 1, which are \( \{a_2, a_3\} \).
The distribution consistent set and the maximum distribution consistent set are related in the following theorem.

**Theorem 7.** Let \( \mathcal{F}^z = (U, A \cup D, F, G) \) be an ordered information system and \( B \subseteq A \) is a distribution consistent set of \( \mathcal{F}^z \) if and only if \( B \) is a maximum distribution consistent set of \( \mathcal{F}^z \).

**Proof.** It can be proved immediately from corresponding definitions and properties. From the definitions of distribution and maximum distribution consistent set, the key results of the implication is that \( [x]_B^z = [x]_A^z \) always holds for any \( x \in U \) while \( B \) is a distribution consistent set or maximum distribution consistent set. Thus, the theorem can be acquired immediately.

**Theorem 8.** Let \( \mathcal{F}^z = (U, A \cup D, F, G) \) be an ordered information system.

**Proof.** We will prove \( P \Rightarrow Q \).

Then we have \( P \Rightarrow Q \).

The distribution consistent set requires that the classification ability of the consistent remains the same with the original data table. That is, \( B \subseteq A \), which is a distribution consistent set of \( A \), must satisfy the fact that \( [x]_B^z = [x]_A^z \) holds for any \( x \in U \). This is very strict and other reductions studied in [21] may not reach this special condition.

### 4. Matrix Algorithm for Distribution Reduction Acquisition in Inconsistent Ordered Information Systems

In this section, the dominance matrices will be put as a restatement and matrices will be employed to realize the calculation of distribution reductions.

**Definition 9.** Let \( \mathcal{F}^z = (U, A \cup D, F, G) \) be an ordered information system, and \( B \subseteq A \). Denote

\[
M_B = (m_{ij})_{n \times n}, \quad \text{where} \quad m_{ij} = \begin{cases} 1, & x_j \in [x_i]_{B}^z, \\ 0, & \text{otherwise}. \end{cases}
\]

The matrix \( M_B \) is called dominance matrix of attributes set \( B \subseteq A \). If \(|B| = l\), we say that the order of \( M_B \) is \( l \).

**Definition 10.** Let \( \mathcal{F}^z = (U, A \cup D, F, G) \) be an ordered information system and \( M_B \) and \( M_C \) be dominance matrices of attributes sets \( B, C \subseteq A \). The intersection of \( M_B \) and \( M_C \) is defined by

\[
M_B \cap M_C = (m_{ij})_{n \times n} \cap (m'_{ij})_{n \times n} = (\min\{m_{ij}, m'_{ij}\})_{n \times n}.
\]

The intersection defined above can be implemented by the operator “*" in Matlab platform, \( M_B \cap M_C = M_B \ast M_C \), that is, the product of elements in corresponding positions. Then the following properties are obvious.

**Proposition 11.** Let \( M_B, M_C \) be dominance matrices of attributes sets \( B, C \subseteq A \); the following results always hold.

1. \( m_{ij} = 1 \).
2. \( M_B \ast M_C = M_B \cap M_C \).

From the above, we can see that a dominance relation of objects has one-one correspondence to a dominance matrix. The combination of dominance relations can be realized by the corresponding matrices and the dominance relations can be compared by the corresponding matrices from the following definitions.

**Definition 12.** Let \( A = (\alpha_1, \alpha_2, \ldots, \alpha_n)^T \) and \( B = (\beta_1, \beta_2, \ldots, \beta_n)^T \) be matrices with \( n \times n \) dimensions and \( \alpha_i \) and \( \beta_i \) row vectors, respectively. If \( \alpha_i \leq \beta_i \) holds, for any \( i \leq n \), we say that \( M_A \) is less than \( M_B \) and it is denoted by \( M_A \leq M_B \).

By the definitions, dominance matrices have the following properties straightly.

**Proposition 13.** Let \( \mathcal{F}^z = (U, A \cup D, F, G) \) be an ordered information system and \( B \subseteq A \). The dominance matrices with respect to \( A \) and \( B \) are, respectively, \( M_A \) and \( M_B \). Then \( M_A \leq M_B \).

In the following, we give the preparation of matrix computation for distribution reductions in ordered information systems.

**Proposition 14.** Let \( \mathcal{F}^z = (U, A \cup D, F, G) \) be an ordered information system and \( U = \{x_1, x_2, \ldots, x_n\} \) and \( A = \{a_1, a_2, \ldots, a_m\} \). Then

\[
M_A = \bigcap_{i=1}^{p} M_{[a_i]} = \begin{pmatrix} a_{i1} & a_{i2} & \cdots & a_{im} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}
\]

and any vector \( \alpha_i = (a_{i1}, a_{i2}, \ldots, a_{im}) \) represents the dominance class of object \( x_i \) by the values 0 and 1, where 0 means the object not included in the class and 1 means the object included in the class.
Input: An inconsistent ordered information system $\mathcal{F}^x = (U, A \cup D, V, f)$, where $U = \{x_1, x_2, \ldots, x_n\}$ and $A = \{a_1, a_2, \ldots, a_p\}$.
Output: All distribution reductions of $\mathcal{F}^x$.

Step 1. Load the ordered information system and simplify the system by combining the objects with same values of every attribute.

Step 2. Classify by every single criterion and store then in separate matrices

$$M_n = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ d_{11} & d_{12} & \cdots & d_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix},$$

$$M_d = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ d_{11} & d_{12} & \cdots & d_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

Step 3. Check the consistence of the information system

$$M_A = \bigcap_{i=1}^p M_m = M_{a_1} \ast M_{a_2} \ast \cdots \ast M_{a_p} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}.$$

where $\ast$ is the operator in Matlab platform. If $M_A \leq M_d$, the system is consistent, terminate the algorithm. Else the system is inconsistent, go to the next step.

Step 4. Acquire the consistent set. Let $B = \{b_1, b_2, \ldots, b_m\} \subseteq A$

$$M_B = \bigcap_{i=1}^m B_i = B_1 \ast B_2 \ast \cdots \ast B_m = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{pmatrix}.$$

If $M_B = M_A$, $B$ is a consistent set, store the set into the temporary storage cell. Else fetch another subset of $A$ and repeat this step. Calculate till all subsets of $A$ are verified, then go to the next step.

Step 5. Sort the consistent sets in the storage cell and find out the minimum consistent sets which are just the reductions. Output all reductions and terminate the algorithm.

\begin{algorithm}
\begin{enumerate}
    \item Load the ordered information system and simplify the system by combining the objects with same values of every attribute.
    \item Classify by every single criterion and store then in separate matrices
    \item Check the consistence of the information system
    \item Acquire the consistent set. Let $B = \{b_1, b_2, \ldots, b_m\} \subseteq A$
    \item Sort the consistent sets in the storage cell and find out the minimum consistent sets which are just the reductions.
\end{enumerate}
\end{algorithm}

**Theorem 15.** Let $\mathcal{F}^x = (U, A \cup D, F, G)$ be an ordered information system and $B \subseteq A$. $B$ is a consistent set if and only if $M_B = M_A$.

**Proof.** As is known, $[x]_B^x \subseteq [x]_A^x$ holds since $B \subseteq A$.

$(\Rightarrow)$ For $B$ is a distribution consistent set, one can have $\mu_B = \mu_A$. Then, for any $x$ and $D_j$, we have $|D_j \cap [x]_B^x| = |D_j \cap [x]_A^x|$. Since $[x]_B^x \subseteq [x]_A^x$, it is obvious that $[x]_B^x = [x]_B^x$. That is, the row vectors in $M_B$ and $M_A$ are correspondingly same. Then $M_B = M_A$.

$(\Leftarrow)$ Since $M_B = M_A$, we can easily obtain that $[x]_A^x = [x]_B^x$ holds for any $x$ and $D_j$. Then $|D_j \cap [x]_A^x| = |D_j \cap [x]_B^x|$ holds for any $x$ and $D_j$. We can obtain that $\mu_B^x(x) = \mu_A^x(x)$ holds for any $x$. That is, $B$ is a distribution consistent set.

To acquire reductions in inconsistent ordered information system, the matrices can be the only forms of storage in computing. And we illustrate the progress to calculate the reductions as shown in Algorithm 1.

The algorithm and the distribution reduction allow us to calculate reductions which keep the classification ability the same with the original system in a brief way. And we do not need to acquire every approximation of the decisions. It shortens the computing time and provides an effective tool for knowledge acquisition in information systems. We analyze the time complexity of Algorithm 1 step by step.

The time complexity to simplify the original information system is $n^2_t$ for any two objects being compared and is denoted by $t_1 = n^2_t$. Since $|U| = n$, $|A| = m$, and $|D| = 1$, the time complexities to be classified by condition attributes and decision $D$ are, respectively, $t_2 = |U|^2 \times |A|$ and $t_3 = |U|^2$.
Inconsistent

Consistent

Begin:

Input data table
and simplify it.

Classify by every single criterion and store them in
separate matrices.

Is the system consistent?

Consistent

Temporary storage of consistent sets.

Sort and output
all reductions.

End:

Terminate the program.

Figure 1: The flow chart of Algorithm 1.

For decision classes being merged by comparing classes of
any two objects, the time complexity is \( t_4 = |U|^2 \). Now the
consistency of the information system needs to be checked
by comparing the condition class and decision class of any
object. If the information system is consistent, the time com-
plexity to check consistency is \(|U|\). If the information system
is inconsistent, the time complexity to check consistency is
less than \(|U|\). Thus, the time complexity to check consistency
is no more than \(|U|\); that is, it is presented as \( t_5 \leq |U| \). Then,
the possible and compatible distribution functions can be
calculated and the time complexity is \( t_6 = 2r \times |U| \). The time
complexity to calculate each of these two functions is \( r \times |U| \)
and is denoted by \( t_6^P = t_6^C = r \times |U| \). The analysis to Step 1
is finished.

For Step 2, the time complexity to calculate possible
and compatible distribution decision matrices, respectively,
is denoted by \( t_7^P = t_7^C = |U|^2 \). Thus, the time complexity to
calculate distribution decision matrices is \( t_7 = 2|U|^2 \). The
time complexity of Step 2 is completed.

The first two steps are preparations to calculate reduc-
tions. The next Step 3 to Step 5 are the steps which run
the operations. There are \( C_m^i = m \) subsets \( \{a_i\} \) and the
dominance matrices are with dimensions \( n \times n \). In addition,
the representation \( C_m^i \) is the combinatorial number which
means the number of selections to chose \( i \) elements from \( m \)
one. We consider that the judgement of a vector if it is zero
runs one operation and the comparison of two vectors runs
according to the dimension of the vectors. Therefore, the time
complexities to compare \( M^P_{i,j} \) and \( M^C_{i,j} \) with \( M_{(a_j)} \), respectively,
are \(|U|^2 \). And the time complexity to compare every line
vector of \( M_{(a_j)} \) with zero is \(|U| \). The possible and compatible
distribution matrices are obtained by reassignment values
\( n \) times. And the time complexities to process possible and
compatible distribution matrices, respectively, are both \( n \).
Then, we have that the total time complexity of Step 3
is \( t_8 = C_m^1 \times (3|U|^2 + 3|U|) \). The judgement in Step 4 just need to
run according to the number of \( \{a_j\} \) and the time complexity
is \( t_9 = 2C_m^1 \).

Since we just need to compute the intersection of nonzero
1st order possible (or compatible) distribution matrices, the
| (U, C ∪ {d}) | a_1 | a_2 | a_3 | a_4 | a_5 | a_6 | a_7 | a_8 | a_9 | d |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|
| x_1:        | African giant pouch rat | 1   | 6.6 | 6.3 | 2   | 8.3 | 4.5 | 42  | 3   | 1 | 3 |
| x_2:        | Asian elephant         | 2547 | 4603 | 2.1 | 1.8 | 3.9 | 69  | 624 | 3   | 5 | 4 |
| x_3:        | Baboon                 | 10.55 | 179.5 | 9.1 | 0.7 | 9.8 | 27  | 180 | 4   | 4 | 4 |
| x_4:        | Big brown bat          | 0.023 | 0.3 | 15.8 | 3.9 | 19.7 | 19  | 35  | 1   | 1 | 1 |
| x_5:        | Brazilian tapir        | 160  | 169  | 5.2 | 1   | 6.2 | 30.4 | 392 | 4   | 5 | 4 |
| x_6:        | Cat                    | 3.3  | 25.6 | 10.9 | 3.6 | 14.5 | 28  | 63  | 1   | 2 | 1 |
| x_7:        | Chimpanzee             | 52.16 | 440  | 8.3 | 1.4 | 9.7 | 50  | 230 | 1   | 1 | 1 |
| x_8:        | Chinchilla             | 0.425 | 6.4  | 11  | 1.5 | 12.5 | 7   | 112 | 5   | 4 | 4 |
| x_9:        | Cow                    | 465  | 423  | 3.2 | 0.7 | 3.9 | 30  | 281 | 5   | 5 | 5 |
| x_10:       | Eastern American mole  | 0.075 | 1.2  | 6.3 | 2.1 | 8.4 | 3.5 | 42  | 1   | 1 | 1 |
| x_11:       | Echidna                | 3    | 25   | 8.6 | 0   | 8.6 | 50  | 28  | 2   | 2 | 2 |
| x_12:       | European hedgehog      | 0.785 | 3.5  | 6.6 | 4.1 | 10.7 | 6   | 42  | 2   | 2 | 2 |
| x_13:       | Galago                 | 0.2  | 5    | 9.5 | 1.2 | 10.7 | 10.4 | 120 | 2   | 2 | 2 |
| x_14:       | Goat                   | 27.66 | 115  | 10.9 | 0.7 | 3.8 | 20  | 148 | 5   | 5 | 5 |
| x_15:       | Golden hamster         | 0.12 | 1    | 11  | 3.4 | 14.4 | 3.9 | 16  | 3   | 1 | 2 |
| x_16:       | Gray seal              | 85   | 325  | 4.7 | 1.5 | 6.2 | 41  | 310 | 1   | 3 | 1 |
| x_17:       | Ground squirrel        | 0.101 | 4   | 10.4 | 3.4 | 13.8 | 9   | 28  | 5   | 1 | 3 |
| x_18:       | Guinea pig             | 1.04 | 5.5  | 7.4 | 0.8 | 8.2 | 7.6 | 68  | 5   | 3 | 4 |
| x_19:       | Horse                  | 521  | 655  | 2.1 | 0.8 | 2.9 | 46  | 336 | 5   | 5 | 5 |
| x_20:       | Lesser short-tailed shrew | 0.005 | 0.14 | 7.7 | 1.4 | 9.1 | 2.6 | 21.5 | 5 | 2 | 4 |
| x_21:       | Little brown bat       | 0.01 | 0.25 | 17.9 | 2   | 19.9 | 24  | 50  | 1   | 1 | 1 |
| x_22:       | Man                    | 62   | 1320 | 6.1 | 1.9 | 8   | 100 | 267 | 1   | 1 | 1 |
| x_23:       | Mouse                  | 0.023 | 0.4  | 11.9 | 1.3 | 13.2 | 3.2 | 19  | 4   | 1 | 3 |
| x_24:       | Musk shrew             | 0.048 | 0.33 | 10.8 | 2   | 12.8 | 2   | 30  | 4   | 1 | 3 |
| x_25:       | N. American opossum    | 1.7  | 6.3  | 13.8 | 5.6 | 19.4 | 5   | 12  | 2   | 1 | 1 |
| x_26:       | Nine-banded armadillo  | 3.5  | 10.8 | 14.3 | 3.1 | 17.4 | 6.5 | 120 | 2   | 1 | 1 |
| x_27:       | Owl monkey             | 0.48 | 15.5 | 15.2 | 1.8 | 17   | 12  | 140 | 2   | 2 | 2 |
| x_28:       | Patas monkey           | 10   | 115  | 10  | 0.9 | 10.9 | 20.2 | 170 | 4   | 4 | 4 |
| x_29:       | Phanlanger             | 1.62 | 11.4 | 11.9 | 1.8 | 13.7 | 13  | 17  | 2   | 1 | 2 |
| x_30:       | Pig                    | 192  | 180  | 6.5 | 1.9 | 8.4 | 27  | 115 | 4   | 4 | 4 |
| x_31:       | Rabbit                 | 2.5  | 12.1 | 7.5 | 0.9 | 8.4 | 18  | 31  | 5   | 5 | 5 |
| x_32:       | Rat                    | 0.28 | 1.9  | 10.6 | 2.6 | 13.2 | 4.7 | 21  | 3   | 1 | 3 |
| x_33:       | Red fox                | 4.235 | 50.4 | 7.4 | 2.4 | 9.8 | 9.8 | 52  | 1   | 1 | 1 |
| x_34:       | Rhesus monkey          | 6.8  | 179  | 8.4 | 1.2 | 9.6 | 29  | 164 | 2   | 3 | 2 |
| x_35:       | Rock hyrax (Hetero.b)  | 0.75 | 12.3 | 5.7 | 0.9 | 6.6 | 7   | 225 | 3   | 2 | 3 |
| x_36:       | Rock hyrax (Procavia hab) | 3.6  | 21   | 4.9  | 0.5 | 5.4 | 6   | 225 | 3   | 2 | 2 |
| x_37:       | Sheep                  | 55.5 | 175  | 3.2 | 0.6 | 3.8 | 20  | 151 | 5   | 5 | 5 |
| x_38:       | Tenrec                 | 0.9  | 2.6  | 11  | 2.3 | 13.3 | 4.5 | 60  | 2   | 1 | 2 |
| x_39:       | Tree hyrax             | 2    | 12.3 | 4.9 | 0.5 | 5.4 | 7.5 | 200 | 3   | 1 | 3 |
| x_40:       | Tree shrew             | 0.104 | 2.5  | 13.2 | 2.6 | 15.8 | 2.3 | 46  | 3   | 2 | 2 |
| x_41:       | Vervet                 | 4.19 | 58   | 9.7  | 0.6 | 10.3 | 24  | 210 | 4   | 3 | 4 |
| x_42:       | Water opossum          | 3.5  | 3.9  | 12.8 | 6.6 | 19.4 | 3   | 14  | 2   | 1 | 1 |
5. Experimental Computing and Case Study

We design programs and employ two cases to demonstrate the effectiveness of the method in the last section. This experimental computing program is running on a personal computer with the following hardware and software configuration. The configuration of the computer is a bit low but the program runs well and fast. It also shows the advantage of Algorithm 1 and the corresponding computing program (see Table 2).

An inconsistent ordered information system on animals sleep is presented in Table 3.

The information system is denoted by \( \mathcal{F} = (U, A \cup \{d\}, V, f) \), where \( A \) is the condition attribute set and \( d \) is the single dominance decision. There are 42 objects which represent the species of animals and 10 attributes with numerical values in the ordered information system. The animals’ names are shown in Table 3 and the interpretations of the attributes will be listed as follows. The interpretations and the units of attributes are represented as shown in Table 4.

By the experimental computing program, the distribution reductions of the system can be calculated and they are represented in the following. The operating time to compute this case is 0.158581 seconds.

The distribution reductions are

\[
\{a_1, a_3, a_4, a_6, a_7, a_8, a_9\}, \\
\{a_2, a_3, a_4, a_6, a_7, a_8, a_9\}, \\
\{a_1, a_2, a_3, a_4, a_6, a_7, a_8, a_9\},
\]

And it can be verified by taking the computer as an assistant that the above sets are reductions of the data table. Detailed progress of the verifying are not arranged here. From the results, we can easily see that the reductions studied in this paper are different from the ones approached in [25], since these reductions are \( \{a_3, a_4, a_6, a_7\}, \{a_4, a_5, a_6, a_7\}, \{a_5, a_6, a_7\}, \) and \( \{a_1, a_2, a_3, a_4\} \). They are different kinds of reductions in ordered information systems and can adapt to different needs in practices. From the definition of different reductions, we can also easily obtain that possible and compatible reductions are usually subsets of distribution reduction. This is not strict and should be studied and verified separately and theoretically. And the work may be taken into account as one part of the future studies in our work.

Finally, we take other inconsistent ordered information system to acquire the distribution reduction, respectively. And the descriptions on the data tables are listed in Table 5.

From the results in Table 5, we can obtain that the algorithm and the program we studied in this paper could be effective and useful to acquire distribution reductions in practice. The numbers of objects and attributes can increase the computing time. But the matrices storage has the ability to shorten the memory and computing time. And it can be helpful in research theoretically and it is applicable.

6. Conclusions

As is known, many information systems are data tables considering criteria for various factors in practise. Therefore, it is meaningful to study the attribute reductions in inconsistent information system on the basis of dominance relations. In this paper, distribution reduction is restated in inconsistent ordered information systems. Some properties and theorems are studied and discussed. A fact is certified that the distribution reduction is equivalent to the maximum distribution reduction in ordered information systems. Theorems on distribution reduction are implemented to create preparations for reduction acquisition and the dominance matrix is also restated to acquire distribution reductions in criterion based information systems. The matrix algorithm for distribution reduction acquisition is stepped and programmed. The algorithm can provide an approach and the program can be effective for theoretical research on knowledge reductions in criterion based inconsistent information systems. Dominance matrices are the only relied parameters which need to be considered without others such as approximations and subinformation systems being brought in. Furthermore, cases are employed to illustrate the validity of the matrix method and the program, which shows that the effectiveness of the algorithm in complicated information systems.
Table 5: Descriptions on the calculations.

| Data name   | Values | Objects | Conditions | Decisions | Reductions | Time          | Operations |
|-------------|--------|---------|------------|-----------|------------|---------------|------------|
| Body fat    | Real   | 252     | 14         | 1         | 11         | 36.43723 s   | 10         |
| Glass       | Real   | 213     | 9          | 1         | 7          | 2.04624 s    | 10         |
| Animal sleep| Real   | 42      | 9          | 1         | 5          | 0.13153 s    | 10         |

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (no. 61100116), the Natural Science Foundation Project of Jiangshu Province (no. BK2011492), and the Youth Foundation of Xuzhou Institute of Technology (no. xxy2011201).

References

[1] Z. Pawlak, "Rough sets," International Journal of Computer and Information Sciences, vol. 11, no. 5, pp. 341–356, 1982.
[2] Y. Y. Yao, "Relational interpretations of neighborhood operators and rough set approximation operators," Information Sciences, vol. 113, no. 1–4, pp. 239–259, 1998.
[3] A. Skowron and J. Stepaniuk, "Tolerance approximation spaces," Fundamenta Informaticae, vol. 27, no. 2–3, pp. 245–253, 1996.
[4] Y. Leung, W. Wu, and W. Zhang, "Knowledge acquisition in incomplete information systems: a rough set approach," European Journal of Operational Research, vol. 168, no. 1, pp. 164–180, 2005.
[5] S. Im, Z. Raš, and H. Wasyłuk, "Action rule discovery from incomplete data," Knowledge and Information Systems, vol. 25, no. 1, pp. 21–33, 2010.
[6] K. Thangavel and A. Pethalakshmi, "Dimensionality reduction based on rough set theory: a review," Applied Soft Computing Journal, vol. 9, no. 1, pp. 1–12, 2009.
[7] H. Wang and S. Wang, "Mining incomplete survey data through classification," Knowledge and Information Systems, vol. 24, no. 2, pp. 221–233, 2010.
[8] W. Z. Wu, Y. Leung, and W. X. Zhang, "Connections between rough set theory and Dempster-Shafer theory of evidence," International Journal of General Systems, vol. 31, no. 4, pp. 405–430, 2002.
[9] W. Wu, M. Zhang, H. Li, and J. Mi, "Knowledge reduction in random information systems via Dempster-Shafer theory of evidence," Information Sciences, vol. 174, no. 3–4, pp. 143–164, 2005.
[10] W. Xu, Y. Li, and X. Liao, "Approaches to attribute reductions based on rough set and matrix computation in inconsistent ordered information systems," Knowledge-Based Systems, vol. 27, pp. 78–91, 2012.
[11] S. Greco, B. Matarazzo, and R. Slowinski, "Rough approximation of a preference relation by dominance relations," European Journal of Operational Research, vol. 117, no. 1, pp. 63–83, 1999.
[12] S. Greco, B. Matarazzo, and R. Slowinski, "A new rough set approach to multicriteria and multiattribute classification," in Rough Sets and Current Trends in Computing (RSCTC ’98), L. Polkowski and A. Skowron, Eds., vol. 1424 of Lecture Notes in Artificial Intelligence, pp. 60–67, Springer, Berlin, Germany, 1998.
[13] S. Greco, B. Matarazzo, and R. Slowinski, "A new rough sets approach to evaluation of bankruptcy risk," in Operational Tools in the Management of Financial Risks, X. Zopounidis, Ed., pp. 121–136, Kluwer, Dordrecht, The Netherlands, 1999.
[14] S. Greco, B. Matarazzo, and R. Slowinski, "Rough sets theory for multicriteria decision analysis," European Journal of Operational Research, vol. 129, no. 1, pp. 1–47, 2001.
[15] S. Greco, B. Matarazzo, and R. Slowinski, "Rough sets methodology for sorting problems in presence of multiple attributes and criteria," European Journal of Operational Research, vol. 138, no. 2, pp. 247–259, 2002.
[16] S. Greco, B. Matarazzo, and R. Slowinski, "Dominance-based rough Set approach as a proper way of handling graduality in rough set theory," in Transaction on Rough Sets VII, vol. 4400 of Lecture Notes in Computer Science, pp. 36–52, 2007.
[17] K. Dembczyński, R. Pindur, and R. Susmaga, "Generation of exhaustive set of rules within dominance-based rough set approach," Electronic Notes in Theoretical Computer Science, vol. 82, no. 4, pp. 99–110, 2003.
[18] K. Dembczyński, R. Pindur, and R. Susmaga, "Dominance-based rough set classifier without induction of decision rules," Electronic Notes in Theoretical Computer Science, vol. 82, no. 4, pp. 84–95, 2003.
[19] R. Susmaga, R. Slowiński, S. Greco, and B. Matarazzo, "Generation of reducts and rules in multi-attribute and multi-criteria classification," Control and Cybernetics, vol. 29, no. 4, pp. 968–988, 2000.
[20] M. W. Shao and W. X. Zhang, "Dominance relation and relus in an incomplete ordered information system," International Journal of Intelligent Systems, vol. 20, pp. 13–27, 2005.
[21] D. Yu, Q. Hu, and C. Wu, "Uncertainty measures for fuzzy relations and their applications," Applied Soft Computing Journal, vol. 7, no. 3, pp. 1135–1143, 2007.