A novel method for attribute reduction of covering decision systems

Changzhong Wang a,*, Qiang He b, Degang Chen c, Qinghua Hu d

a Department of Mathematics, Bohai University, Jinzhou 121000, PR China
b Department of Mathematics and Computer Science, Hebei University, Baoding 071002, PR China
c Department of Mathematics and Physics, North China Electric Power University, Beijing 102206, PR China
d School of Computer Science and Technology, Tianjin University, Tianjin 300072, PR China

Article history:
Received 17 April 2011
Received in revised form 6 August 2013
Accepted 24 August 2013
Available online 2 September 2013

Keywords:
Covering rough set
Attribute reduction
Discernibility matrix
Consistent covering decision system
Inconsistent covering decision system

Abstract
Attribute reduction has become an important step in pattern recognition and machine learning tasks. Covering rough sets, as a generalization of classical rough sets, have attracted wide attention in both theory and application. This paper provides a novel method for attribute reduction based on covering rough sets. We review the concepts of consistent and inconsistent covering decision systems and their reducts and we develop a judgment theorem and a discernibility matrix for each type of covering decision system. Furthermore, we present some basic structural properties of attribute reduction with covering rough sets. Based on a discernibility matrix, we develop a heuristic algorithm to find a subset of attributes that approximate a minimal reduct. Finally, the experimental results for UCI data sets show that the proposed reduction approach is an effective technique for addressing numerical and categorical data and is more efficient than the method presented in the paper [D.G. Chen, C.Z. Wang, Q.H. Hu, A new approach to attribute reduction of consistent and inconsistent covering decision systems with covering rough sets, Information Sciences 177(17) (2007) 3500–3518].

© 2013 Elsevier Inc. All rights reserved.

1. Introduction

Attribute reduction plays an important role in pattern recognition and machine learning. Classical rough set theory, as proposed by Pawlak [31], has been used as a mathematical tool to conceptualize and analyze various types of imprecise and uncertain data. It can be a useful tool for studying attribute reduction in information systems. The main goal of attribute reduction is to remove redundant information in a data set, so that correct decisions can quickly be made while preserving or even improving classification ability. Rough set theory has become a popular mathematical framework for feature selection, rule extraction, data mining and knowledge discovery [5–23,26–38,44,46,51–53].

Originally, rough set theory was based on equivalence relations, which can partition the objects of a universe into mutually exclusive equivalence classes. Objects in the same equivalence class are indiscernible. Equivalence classes are also called elementary information granules; an arbitrary subset of the universe can be approximated by elementary information granules. Rough set-based data analysis starts from a data table, called an information system, which contains objects that are described by a finite set of categorical attributes [17–22,28–38,44,51,58–60].

However, in many real-world situations, the databases are not suitable for being handled by classical rough sets [1–7,10–27,39–43,45–58,61–68]. For example, some objects in a database have multiple attribute values. If we consider all of the
attribute values for a multi-valued attribute, we create a covering of the universe instead of a partition. Some illustrative examples of this type of database are given by Chen and Wang [5,45]. In addition, classical rough sets work only for categorical data and cannot be directly applied to reducing numerical attributes. Numerical data need to be discretized before attribute reduction, and this causes information loss [14–16]. For this reason, similarity relation rough sets [43], dominance rough sets [12,13,41], and neighborhood rough sets [14,15,55] were developed. These models induce a covering of a universe instead of a partition. Zakowski [64] used coverings of a universe to introduce the notion of covering rough sets. Later, many authors conducted studies on the properties of covering approximation operators [1–5,25,40,45,48–50,57,63–68]. Bonikowski et al. [1] mainly studied the structures of the coverings. Most authors studied some properties of the upper and lower approximations of covering rough sets and examined some axioms that are satisfied by Pawlak rough sets [25,48,63–65,67,68]. However, few people employ covering rough sets to conduct research on attribute reduction. Zhu and Wang [66] investigated the problem of reducing covering elements from a covering. Their objective was to remove excessive covering elements in a covering, under the condition that the approximations of an arbitrary subset were kept unchanged. A pioneering study on attribute reduction with covering rough sets was conducted by Chen and Wang [5]. Formal concepts of attribute reduction based on covering rough sets were introduced, and some approaches to computing all reducts using approximations of covering sets and examined some axioms that are satisfied by Pawlak rough sets [25,48,63–65,67,68]. However, few people employ covering rough sets to conduct research on attribute reduction. Zhu and Wang [66] investigated the problem of reducing covering elements from a covering. Their objective was to remove excessive covering elements in a covering, under the condition that the approximations of an arbitrary subset were kept unchanged. A pioneering study on attribute reduction with covering rough sets was conducted by Chen and Wang [5]. Formal concepts of attribute reduction based on covering rough sets were introduced, and some approaches to computing all reducts using a discernibility matrix were developed. However, these approaches are very complex and time-consuming due to their poor design.

In this paper, we present a novel method for the attribute reduction of decision systems based on covering rough sets. We review some concepts related to attribute reductions with covering rough sets, present some theorems describing attribute reductions, and then construct discernibility matrices in covering decision systems. Compared with the attribute reduction method presented in [5], the computational complexity of the proposed method is greatly reduced. Finally, based on a discernibility matrix, we develop a heuristic algorithm to find a minimal subset of attributes that approximate an optimal reduct. The experimental results show that the proposed reduction method can effectively handle data and is more efficient than the reduction method presented in [5].

This paper is organized as follows. In Section 2, we recall some basic concepts related to traditional and covering rough sets. In Section 3, we develop a new method of attribute reduction in consistent covering decision systems and examine some basic properties of attribute reduction with covering rough sets. In Section 4, we discuss a theory for reducing the conditional attributes of inconsistent covering decision systems. In Section 5, we present some experiments on some public data sets.

2. Preliminaries

First, we review some basic concepts related to classical rough sets that can be found in [30–32].

An information system is a pair \((U,A)\), where \(U\) is a nonempty set of samples \(\{x_1, x_2, \ldots, x_n\}\), called a universe or a sample space and \(A\) is a nonempty set of attributes or features. For any subset \(B \subseteq A\), we can define an equivalence relation as follows:

\[
\text{Ind}(B) = \{ (x, y) \in U \times U : a(x) = a(y), \forall a \in B \}
\]

Clearly, \(\text{Ind}(B) = \cap_{a \in B} \text{Ind}(\{a\})\). We denote the equivalence class of \(x\) with respect to \(\text{Ind}(B)\) as \([x]_B\). For \(X \subseteq U\), the lower and upper approximations of \(X\) with respect to \(\text{Ind}(B)\) are defined as follows:

\[
\underline{B}(X) = \{ x \in U : [x]_B \subseteq X \}, \quad \overline{B}(X) = \{ x \in U : [x]_B \cap X \neq \emptyset \}
\]

By \(M(U,A)\), we denote a \(n \times m\) matrix \((c_{ij})\), called the discernibility matrix of \((U,A)\), such that \(c_{ij} = \{a \in A : a(x_i) \neq a(x_j)\}\) for \(i, j = 1, 2, \ldots, n\). A discernibility function \(f(U,A)\) of an information system \((U,A)\) is a Boolean function of Boolean variables \(\overline{c_{1}}, \ldots, \overline{c_{m}}\) corresponding to the attributes \(a_1, \ldots, a_m\) and defined as follows:

\[
f(A) | \overline{c_{1}}, \ldots, \overline{c_{m}} = \land \{ \lor(c_{ij}) : 1 \leq j < i \leq n \}
\]

where \(\lor(c_{ij})\) is the disjunction operation on \(c_{ij}\).

A decision system is a pair \((U,A,D)\), where \(D\) is called a decision attribute and the elements in \(A\) are called conditional attributes. An attribute \(a \in B \subseteq A\) is called relatively dispensable in \(B\) if \(\text{Pos}_B(D) = \text{Pos}_{B-B} (D)\); otherwise, it is said to be relatively indispensable in \(B\). If each attribute in \(B\) is relatively dispensable in \(B\), \(B\) is said to be relatively independent in \((U,A,D)\). A subset \(B \subseteq A\) is called a relative reduct in \((U,A,D)\) if \(B\) is relatively independent in \((U,A,D)\) and \(\text{Pos}_B(D) = \text{Pos}_A(D)\). The set of all relatively dispensable attributes in \(A\) is called the relative core of \((U,A,D)\).

Classical rough sets can only be used to address discrete data. Rough sets have been extended to address complex data with discrete and numerical features. Covering rough set theory is a natural extension of classical rough sets and can effectively handle continuous data. Next, we recall some basic notions related to covering rough sets that can be found in [5,45].

**Definition 2.1.** Let \(U\) be a universe of discourse and \(C\) be a family of subsets of \(U\). Then, \(C\) is called a covering of \(U\) if no subset in \(C\) is empty and \(\cup C = U\).
Definition 2.2. Let $\mathcal{C} = \{K_1, K_2, \ldots, K_n\}$ be a covering of $U$. For every $x \in U$, let $C_x = \cap \{K_j : K_j \subseteq C, x \in K_j\}$. $\text{Cov}(C) = \{C_x : x \in U\}$ is then also a covering of $U$, and we call it the induced covering of $C$.

For every $x \in U$, $C_x$ is the minimal set including $x$ in $\text{Cov}(C)$. Each element in $\text{Cov}(C)$ cannot be written as the union of other elements in $\text{Cov}(C)$. $\text{Cov}(C) = C$ if and only if $C$ is a partition. For every $x, y \in U$, if $y \in C_x$ then $C_x \supseteq C_y$; so if $y \in C_x$ and $x \in C_y$, then $C_x = C_y$.

This definition uses a covering to granulate a universe; each sample is associated with a neighborhood. This type of granulation mode defines a relation between samples. The relation has the following properties.

(1) Reflexivity: $\forall x \in U, x \in C_x$.
(2) Anti-symmetry: if $y \in C_x$ and $x \in C_y$, then $C_x = C_y$.
(3) Transitivity: $\forall x, y, z \in U$; if $x \in C_y$ and $y \in C_z$, then $x \in C_z$.

In fact, the relation induced by Definition 2.2 is a dominance relation and the elements in $\text{Cov}(C)$ are dominance classes of the dominance relation [50].

Definition 2.3. Let $\Delta = \{C_i : i = 1, 2, \ldots, m\}$ be a family of coverings of $U$. For every $x \in U$, let $\Delta_x = \cap \{(C_i)_x : (C_i)_x \in \text{Cov}(C_i), x \in (C_i)_x\}$. $\text{Cov}(\Delta) = \{\Delta_x : x \in U\}$ is then also a covering of $U$, and we call it the induced covering of $\Delta$.

Clearly $\Delta_x$ is the intersection of all coverings including $x$ in $\Delta$. So for every $x \in U$, $\Delta_x$ is the minimal set including $x$ in $\text{Cov}(\Delta)$. $\text{Cov}(\Delta)$ can be viewed as the intersection of coverings in $\Delta$. Every element in $\text{Cov}(\Delta)$ cannot be written as the union of other elements in $\text{Cov}(\Delta)$. If every covering in $\Delta$ is a partition, then $\text{Cov}(\Delta)$ is also a partition and $\Delta_x$ is the equivalence class that includes $x$. For every $x, y \in U$, if $y \in \Delta_x$, then $\Delta_x \supseteq \Delta_y$; so if $y \in \Delta_x$ and $x \in \Delta_y$, then $\Delta_x = \Delta_y$.

If a covering produces a dominance relation, a family of coverings can induce a set of dominance relations. Definition 2.3 presents a method for constructing dominance classes with a family of coverings [50]. In this sense, we can say that each element in $\text{Cov}(\Delta)$ is a dominance class. Based on Definitions 2.2 and 2.3, discrete or continuous data can be granulated into elementary information granules—dominance classes.

For any $X \subseteq U$, the lower and upper approximations of $X$ with respect to $\Delta$ are defined as follows:

$$\Delta(X) = \{x \in U : \Delta_x \subseteq X\}, \quad \tilde{\Delta}(X) = \{x \in U : \Delta_x \cap X \neq \emptyset\}.$$  

The pair approximation operators are dual to each other. The readers can refer to [55] for a discussion of the operators' important properties. The positive, negative and boundary domains of $X$ with respect to $\Delta$ are computed using the following formulas, respectively:

$$\text{POS}_\Delta(X) = \Delta(X), \quad \text{NEG}_\Delta(X) = U - \tilde{\Delta}(X), \quad \text{BN}_\Delta(X) = \tilde{\Delta}(X) - \Delta(X).$$

3. Attribute reduction in consistent covering decision systems

Attribute reduction is an important application of rough set theory. However, in practical applications, a large number of databases cannot be directly handled by classical rough sets. For this reason, similarity relation rough sets, dominance rough sets, and neighborhood rough sets are widely used. These models all induce coverings of a universe instead of partitions and can thus be categorized into covering rough sets. In the following, we present an illustrative example to further show which types of data sets are suitable for being handled by covering rough sets.

Example 3.1. Table 1 shows a petroleum exploration data set, where 'No' represents the number of wells, 'c1' represents the average amplitude, 'c2' represents the semi-attenuation time, 'c3' represents the instantaneous frequency, 'c4' represents the Root Mean Square (RMS) amplitude, 'D' represents the well type, '1' stands for a petroleum well, '2' stands for a dry well.

All of the conditional attributes in Table 1 are numerical. According to classical rough sets, these conditional attributes need to be discretized before attribute reduction. However, discretization can cause information loss. Here, we directly address the data set with covering rough sets. First, we employ two granulation methods to generate elementary covering information granules. Let $U = \{x_i : i = 1, 2, \ldots, 7\}$ be the set of well samples.

Table 1

<table>
<thead>
<tr>
<th>No</th>
<th>c1</th>
<th>c2</th>
<th>c3</th>
<th>c4</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>85.4051</td>
<td>54.9076</td>
<td>27.0976</td>
<td>61.2443</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>66.4600</td>
<td>39.0294</td>
<td>25.2136</td>
<td>53.5892</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>109.7228</td>
<td>34.7891</td>
<td>39.0904</td>
<td>69.2056</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>66.8497</td>
<td>63.5205</td>
<td>24.2548</td>
<td>48.1985</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>55.3513</td>
<td>66.4574</td>
<td>32.5181</td>
<td>39.7762</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>40.1296</td>
<td>77.9718</td>
<td>33.1104</td>
<td>24.6535</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>48.6229</td>
<td>35.0866</td>
<td>30.1238</td>
<td>35.4501</td>
<td>2</td>
</tr>
</tbody>
</table>
(1) For each conditional attribute \( c_i (i = 1, 2, 3, 4) \), we associate a neighborhood with each sample \( x_k (k = 1, 2, \ldots, 7) \), called a covering element of \( x_k \). For example, a covering element of \( x_9 \) with respect to \( c_i \) can be defined as \( (C_{9,i}) = \{ x : d(x_9, x) \leq \varepsilon \} \), where \( d(\cdot, \cdot) \) is a distance function and \( \varepsilon > 0 \) is a specified threshold. Obviously, \( C_i = \{ (R_i)(x_k) : k = 1, 2, \ldots, 7 \} \) is a covering of \( U \), and \( \Delta = \{ C_1, C_2, C_3, C_4 \} \) is a family of coverings of \( U \). By Definition 2.2, we can obtain a set of dominance classes for each conditional attribute. Then, by Definition 2.3, we can obtain a set of elementary information granules—dominance classes generated by all conditional attributes. In some sense, dominance classes reflect dominance relationships among the well samples in the classification.

(2) By sorting the well samples according to their values, we define an order relation \( R_i \) for each conditional attribute \( c_i \). For example, we can define a relation \( R_i \) for \( c_i \) such that the relationship between well samples is that \( x_5 > x_6 > x_8 > x_7 > x_4 > x_9 > x_3 > x_1 > x_2 \). It is easily observed that \( R_i \) is a special dominance relation. We can thus compute the dominance class of each well sample related to \( R_i \) as follows: \( (R_i)(x_k) = \{ x_5, x_6, x_8, x_7, x_4, x_9, x_3, x_1, x_2 \} \). Thus, \( C_i \) is a special dominance relation. We can thus compute the dominance class of each well sample related to \( R_i \) as follows: \( (R_i)(x_k) = \{ x_5, x_6, x_8, x_7, x_4, x_9, x_3, x_1, x_2 \} \).

With the above-mentioned granulation mechanisms of data sets, an attribute reduction approach was proposed to compute all of the reducts in [5]. However, this approach is very complex and time-consuming due to its poor design. Covering rough sets can also be used to address data sets with numerical and symbolic attributes. We discuss covering rough sets using real data sets in Section 5. In this section, we propose a new model of attribute reduction with covering rough sets in consistent decision systems.

Based on the above observations, if we apply covering rough sets to numerical data sets, each conditional attribute can induce a covering rather than a partition. Suppose \( U \) is a nonempty set of samples \( \{ x_1, x_2, \ldots, x_m \} \), \( \Delta = \{ C_i : i = 1, 2, \ldots, m \} \) is a family of coverings generated by condition attributes, \( D \) is a decision attribute, and \( U/D \) is the decision partition of \( U \). We call the triple \((U, \Delta, D)\) a covering decision system. If for any object \( x \), there exists \( D_* \in U/D \) such that \( \Delta_* \subset D_* \) and there is at least one \( \Delta_* \) in \( U/D \), then \((U, \Delta, D)\) is called a consistent covering decision system, denoted as \( \text{Cov}(\Delta) \leqslant U/D \). Otherwise, \((U, \Delta, D)\) is called an inconsistent covering decision system. Let \( D = \{ d \} \). The decision function is defined as \( d : U \rightarrow V_d \), where \( V_d = \{ 1, 2, \ldots, r \} \) is the decision domain of \( d \). For any \( x \in U \), the image of set \( \Delta_* \) under function \( d \) is denoted as \( d(\Delta_*) \). If \((U, \Delta, D)\) is a consistent covering decision system, then \( d(\Delta_*) \) is a singleton and \( d(\Delta_*) = \{ d(x) \} \) holds.

**Theorem 3.1.** Let \((U, \Delta, D)\) be a consistent covering decision system, \( P \subseteq \Delta \), then

1. \( \text{Cov}(P) = \text{Cov}(\Delta) \Leftrightarrow \Delta_* = P_* \forall x \in U \).
2. For \( \forall x, y \in U, y \neq \Delta_* \) if and only if there is at least a covering \( C_i \in \Delta \) such that \( y \notin (C_i)_* \).
3. \( \text{Cov}(P) = \text{Cov}(\Delta) \) if and only if for \( \forall x, y \in U \), if \( y \neq \Delta_* \), then \( y \neq P_* \).

**Proof.** Straightforward.

**Theorem 3.2.** Let \((U, \Delta, D)\) be a consistent covering decision system, \( C_i \in \Delta \). If \( \text{Cov}(\Delta - \{ C_i \}) \leqslant U/D \), then \( C_i \) is called indispensable relative to \( D \); otherwise, \( C_i \) is called indispensable relative to \( D \) in \( \Delta \). Let \( P \subseteq \Delta \); if \( \text{Cov}(P) \leqslant U/D \) and \( \text{Cov}(P - \{ C_i \}) \leqslant U/D \) for any \( C_i \subseteq P \), then \( P \) is called a reduct of \( \Delta \) relative to \( D \). The collection of all indispensable elements in \( \Delta \) is called the core of \( \Delta \) relative to \( D \) and is denoted as \( \text{Core}_D(\Delta) \).

**Theorem 3.3.** Suppose \( \text{Cov}(\Delta) \leqslant U/D \), \( C \in \Delta \). Then, \( C \) is indispensable relative to \( D \) if and only if there is at least one pair of \( x_i, x_j \in U \) satisfying \( d(x_i) \neq d(x_j) \) such that \( x_i \notin \Delta_* \) if and only if \( x_j \notin (\Delta - \{ C_i \})_* \).

**Proof.** \( \Rightarrow \) If \( \text{Cov}(\Delta - \{ C_i \}) \leqslant U/D \), then true, but by the definition of \((U, \Delta, D)\) there is an \( x_i \) such that \( (\Delta - \{ C_i \})_* \leqslant [x_i]_D \) is not true. Thus, \( \exists x_j \in U \), such that \( x_j \in (\Delta - \{ C_i \})_* \) and \( x_j \notin [x_i]_D \), which implies that \( d(x_i) \neq d(x_j) \). Hence, there is at least a pair of \( x_i, x_j \in U \) satisfying \( d(x_i) \neq d(x_j) \) such that \( x_i \notin \Delta_* \Rightarrow x_j \in (\Delta - \{ C_i \})_* \).

\( \Leftarrow \) Because \( d(x_i) \neq d(x_j) \) and \( \text{Cov}(\Delta) \leqslant U/D \), it follows that \( \Delta_* \subseteq [x_i]_D \neq [x_j]_D \neq \Delta_* \). By the reflexivity of \( \Delta_* \), we have \( x_j \neq \Delta_* \). Because \( x_j \neq \Delta_* \Rightarrow x_j \in (\Delta - \{ C_i \})_* \), it follows that \( (\Delta - \{ C_i \})_* \leqslant [x_j]_D \) is not true. By the definition of \((U, \Delta, D)\), we know that \( C \) is indispensable relative to \( D \).
Theorem 3.3 shows that a covering C in a consistent covering decision system is indispensible if and only if there is at least one pair of objects \( x_i \) and \( x_j \) with different decisions such that \( x_j \) does not belong to \( \Delta_i \), instead belongs to \( \{ \Delta - \{ C \} \}_{x_i} \). This finding implies that C is a sole covering that meets \( x_j \notin C_{x_i} \).

**Theorem 3.4 (Judgment theorem of attribute reduction).** Suppose that \( Cov(\Delta) \leq U/D, \ P \subseteq \Delta \). Then, \( Cov(P) \leq U/D \) if and only if \( x_j \neq P_x \) for any \( x_i, x_j \in U \), satisfying \( d(x_i) \neq d(x_j) \).

**Proof.** Because \( Cov(P) \leq U/D \), it follows that \( P_x \subseteq [x_i]_D \) for any \( x_i \in U \). We also have \( P_x \subseteq [x_i]_D \) for any \( x_j \in U \) satisfying \( [x_i]_D \cap [x_j]_D = \emptyset \). This means that \( x_i \neq P_x \). Because \( \Delta_i \subseteq P_x \) and \( \Delta_j \subseteq P_x \), we have \( d(\Delta_i) \neq d(\Delta_j) \), which implies \( d(x_i) \neq d(x_j) \) by the reflexivity of \( \Delta_i \) and \( \Delta_j \). That is, we have \( x_j \neq P_x \) for any \( x_i, x_j \in U \) satisfying \( d(x_i) \neq d(x_j) \).

**Theorem 3.5** is not true. It follows from the definition of a consistent covering decision system that there must exist \( \delta_{x_i} \in U \) such that \( P_{x_i} \subseteq [x_i]_D \) is not true. This means that there is \( x_i \in U \) such that \( x_i \in P_{x_i} \), and \( [x_i]_D \cap [x_i]_D = \emptyset \), which implies \( d(x_i) \neq d(x_i) \). Thus, \( x_i \in P_{x_i} \) but \( x_i \) and \( x_j \) satisfy \( d(x_i) \neq d(x_i) \). This contradicts the fact that \( x_j \neq P_x \) for any \( x_i, x_j \in U \) satisfying \( d(x_i) \neq d(x_j) \). Hence, \( Cov(P) \leq U/D \).

The objective of relative reduction is to find a minimal subset of \( \Delta \) that ensures that the original relationship between every two objects with different decisions is invariant. For any two objects with different decisions, we can thus find at least one condition attribute (covering) to distinguish them. To search for all reducts in a consistent covering decision system, we define its discernibility matrix according to **Theorems 3.1(2) and 3.4** as follows:

**Definition 3.5.** Let \( (U, \Lambda, D = \{d\}) \) be a consistent covering decision system, where \( U = \{x_1, x_2, \ldots, x_n\} \) and \( \Lambda = \{C_i: i = 1, 2, \ldots, m\} \). By \( M(U, \Lambda, D) \), we denote a \( n \times n \) matrix \( (c_{ij}) \), called the discernibility matrix of \( (U, \Lambda, D) \), which is defined as

\[
\begin{align*}
C_{ij} = \begin{cases}
\{ C \in \Lambda : x_j \notin C_x \}, & d(x_i) \neq d(x_j) \\
\Lambda, & d(x_i) = d(x_j)
\end{cases}
\end{align*}
\]

for \( x_i, x_j \in U \).

To keep the classification of objects unchanged, any two objects must be distinguished by some attributes if they have different decisions. This idea is consistent with the viewpoint of classical rough sets. In addition, covering and classical rough sets bear some formal resemblance to discernibility matrices. Thus, the proposed reduction method is a generalization of classical rough sets.

In [5], a method based on a discernibility matrix was proposed for computing all relative reducts. To compare it with the proposed method in this paper, we present the reduction method in [5] as follows.

**Definition 3.6.** Let \( (U, \Lambda, D = \{d\}) \) be a consistent covering decision system, where \( U = \{x_1, x_2, \ldots, x_n\} \) and \( \Lambda = \{C_i: i = 1, 2, \ldots, m\} \). By \( M(U, \Lambda, D) \), we denote a \( n \times n \) matrix \( \hat{(c_{ij})} \), called the discernibility matrix of \( (U, \Lambda, D) \), which is defined as

\[
\begin{align*}
C^\prime_{ij} = \begin{cases}
\{ C \in \Lambda : (C_x \subset C_y) \wedge (C_x \subset C_x) \} \cup \{ C \in \Lambda : (C_x \subset C_y) \wedge (C_y \subset C_x) \}, & d(\Lambda_x) \neq d(\Lambda_y) \\
\Lambda, & d(\Lambda_x) = d(\Lambda_y)
\end{cases}
\end{align*}
\]

for \( x_i, x_j \in U \).

By observing Definitions 3.5 and 3.6, we find the time complexities in computing neighborhoods of samples are the same and that the main difference lies in the following two points:

1. In Definition 3.5, we only need to compute \( C_x \) for any covering \( C \in \Lambda \) and an arbitrary object \( x \in U \), while both \( C_x \) and \( \Lambda_x \) need to be computed in Definition 3.6.
2. The computational complexities of the two discernibility matrices are different. In Definition 3.6, to distinguish any two objects with different decisions, we need \( m^2 \) comparisons of their neighborhoods. The time complexity is thus \( O(m^2 \cdot n^2) \). However, we need \( m \) comparisons for the two objects in Definition 3.5. Hence, the time complexity in Definition 3.5 is \( O(m \cdot n^2) \). Therefore, the proposed method of reduction is simpler than the method presented in [5].

In fact, the reducts obtained by these two discernibility matrices are the same. Next, we provide a theoretical explanation of this point.

It is easily observed that the discernibility matrix in Definition 3.6 is symmetric and the discernibility matrix in Definition 3.5 is not. For any \( x_i \subset U \), because \( (U, \Lambda, D = \{d\}) \) is a consistent covering decision system, we know that \( d(x_i) \neq d(x_j) \) is equivalent to \( d(x_i) \neq d(x_j) \). Let \( c_{ij} = c_{ij} \cup c_{ji} \); we need to show that \( c_{ij} = c_{ij} \).

For any \( x_i \in c_{ij} \), it follows from Definition 3.5 that \( x_i \neq C_x \) or \( x_i \neq C_x \). This means that \( C_x \subset C_x \) or \( C_x \subset C_x \) by Definition 2.2, which implies \( C \in C_{ij} \) by Definition 3.6. Hence, we have that \( C_{ij} \subseteq C_{ij} \).

Conversely, if \( C \in C_{ij} \), by Definition 3.6, we have that \( C_x \subset C_x \) and \( C_x \subset C_x \). It follows from Definition 2.2 that \( x_i \neq C_x \) and \( x_i \neq C_x \), which implies that \( C \in C_{ij} \) by Definition 3.5. If \( C_x \subset C_x \), it follows from Definition 3.6 that \( (C_x)_{x_i} \subset (C_x)_{x_i} \) and \( (C_x)_{x_i} \subset (C_x)_{x_i} \). By Definition 2.2, we have that \( x_i \neq (C_x)_{x_i} \) and \( x_i \neq (C_x)_{x_i} \). From Definition 3.5, this means that \( C \in C_{ij} \) and \( C \in C_{ij} \).
\[ c_{ij} \in c_{ij}' \] Hence, \( C_i \land C_j \in c_{ij}' \), and it follows that \( c_{ij}' = c_{ij} \). We can thus say that these two discernibility matrices are equivalent and that the reductions obtained by them are the same.

The following theorems examine some structural properties of attribute reduction in a consistent covering decision system. The complexities of the following theorems are lower than the corresponding theorems in [5].

**Theorem 3.7.** Let \( (U, \Delta, D) \) be a consistent covering decision system.

1. Let \( P \subseteq \Delta \); then \( \text{Cov}(P) \subseteq U/D \) if and only if \( P \cap c_{ij} \neq \emptyset \) for every \( i, j \leq n \).
2. \( \text{Core}_D(\Delta) = \{ C \in \Delta : c_{ij} = \{ C \}, i, j \leq n \} \).

**Proof.**

1. \( \Rightarrow \) For any \( x_i, x_j \in U \):
   
   If \( d(x_i) = d(x_j) \), then by Definition 3.5, we have \( c_{ij} = \Delta \). Hence, \( P \cap c_{ij} \neq \emptyset \).
   
   If \( d(x_i) \neq d(x_j) \), then by Theorem 3.1(2), there must exist \( C \in P \) such that \( y \notin C \). By Definition 3.5, it follows that \( C \in c_{ij} \), which implies \( P \cap c_{ij} \neq \emptyset \).

2. \( \Leftarrow \) If \( P \cap c_{ij} \neq \emptyset \) for any \( c_{ij} \neq \emptyset \), we suppose that \( C \in P \cap c_{ij} \). If \( d(x_i) \neq d(x_j) \), then by Definition 3.5, we have \( x_i \neq x_j \). By Theorem 3.4, we have \( \text{Cov}(P) \subseteq U/D \).

Corollary 3.8. **Let** \( P \subseteq \Delta \), \( P \) is a relative reduct of \( \Delta \) if and only if it is a minimal set satisfying \( P \cap c_{ij} \neq \emptyset \), \( i, j = 1,\ldots,n \).

**Example 3.2 [5]**. Suppose \( U = \{ x_1,\ldots,x_9 \} \), \( \Delta = \{ C_i : i = 1,\ldots,4 \} \), and

\[
\begin{align*}
C_1 &= \{ \{ x_1, x_2, x_3, x_5, x_7, x_8 \}, \{ x_2, x_3, x_5, x_6, x_9, x_8 \} \}, \\
C_2 &= \{ \{ x_1, x_2, x_3, x_4, x_5, x_6 \}, \{ x_4, x_5, x_6, x_7, x_8, x_9 \} \}, \\
C_3 &= \{ \{ x_1, x_2, x_3 \}, \{ x_4, x_5, x_7, x_8, x_9 \}, \{ x_7, x_8, x_9 \} \}, \\
C_4 &= \{ \{ x_1, x_2, x_4, x_5 \}, \{ x_2, x_3, x_5, x_6 \}, \{ x_4, x_5, x_7, x_8, x_9 \}, \{ x_5, x_6, x_8, x_9 \} \}, \\
U/D &= \{ \{ x_1, x_2, x_3 \}, \{ x_4, x_5, x_6 \}, \{ x_7, x_8, x_9 \} \}
\end{align*}
\]

By Definition 3.5 we can compute the discernibility matrix of \( (U, \Delta, D) \) as follows:

\[
\begin{array}{cccccccc}
\Delta & \Delta & \Delta & \{ C_3 \} & \{ C_3 \} & \{ C_1, C_3, C_4 \} & \{ C_2, C_3, C_4 \} & \{ C_2, C_3, C_4 \} & \Delta \\
\Delta & \Delta & \Delta & \{ C_1, C_3, C_4 \} & \{ C_3 \} & \{ C_1, C_3, C_4 \} & \{ C_1, C_3, C_4 \} & \{ C_2, C_3 \} & \Delta \\
\Delta & \Delta & \Delta & \{ C_1, C_3, C_4 \} & \{ C_3 \} & \{ C_1, C_3, C_4 \} & \{ C_1, C_3, C_4 \} & \{ C_2, C_3 \} & \Delta \\
\{ C_2, C_3, C_4 \} & \{ C_2, C_3, C_4 \} & \Delta & \Delta & \Delta & \{ C_2, C_3 \} & \{ C_2, C_3 \} & \{ C_2, C_3 \} & \Delta \\
\{ C_2, C_3, C_4 \} & \{ C_2, C_3, C_4 \} & \Delta & \Delta & \Delta & \{ C_1, C_2, C_4 \} & \{ C_1, C_2, C_4 \} & \{ C_1, C_2, C_4 \} & \Delta \\
\{ C_2, C_3, C_4 \} & \{ C_2, C_3, C_4 \} & \Delta & \Delta & \Delta & \{ C_1, C_2, C_4 \} & \{ C_1, C_2, C_4 \} & \{ C_1, C_2, C_4 \} & \Delta \\
\{ C_2, C_3, C_4 \} & \{ C_2, C_3, C_4 \} & \Delta & \Delta & \Delta & \{ C_1, C_2, C_4 \} & \{ C_1, C_2, C_4 \} & \{ C_1, C_2, C_4 \} & \Delta \\
\{ C_2, C_3, C_4 \} & \{ C_2, C_3, C_4 \} & \Delta & \Delta & \Delta & \{ C_1, C_2, C_4 \} & \{ C_1, C_2, C_4 \} & \{ C_1, C_2, C_4 \} & \Delta \\
\end{array}
\]

and
\[ f(U, \Delta) = \bigwedge \{ \bigvee (c_i) : 1 \leq j < i \leq 9 \} \]
\[ = C_1 \land (C_1 \lor C_2 \lor C_4) \land (C_2 \lor C_3) \land (C_2 \lor C_4) \land (C_1 \lor C_2 \lor C_4) = C_1 \land (C_2 \lor C_4) \]
\[ = (C_2 \land C_3) \lor (C_1 \land C_4) \]

Thus, \( \text{Red}(A) = \{ (C_3, C_4), (C_2, C_3) \} \), \( \text{Core}_0(A) = \{ C_3 \} \).

In comparison with the discernibility matrix of \((U, \Delta, D)\) presented in [5], the discernibility matrix constructed in this paper is simpler and the computational complexity is greatly reduced. However, the reduces that it obtained are the same as those obtained in the method used in [5], so we can say that the proposed method is a more efficient way to compute all of the reducts than the method presented in [5].

4. Attribute reduction in inconsistent covering decision systems

In consistent covering decision systems, each object has a neighborhood in which all objects have the same decision as itself. In many practical problems, this situation does not always exist. There are also some objects in a neighborhood whose decisions are not consistent. For this reason, consistent covering decision systems need to be generalized to more general decision systems: inconsistent covering decision systems. In this section, we study attribute reduction for this type of decision system.

Suppose that \( \Delta = \{ C_i : i = 1, 2, \ldots, m \} \) is a family of coverings of \( U, D \) is a decision attribute, \( d: U \to V_d \) is the decision function, and \( (U, \Delta, D = \{ d \}) \) is an inconsistent covering decision system. The positive domain of \( D \) relative to \( \Delta \) is defined as \( \text{Pos}_\Delta(D) = \bigcup_{x \in U} \Delta(x) \). For an attribute \( C_i \in \Delta \), if \( \text{Pos}_{\Delta \setminus \{ C_i \}}(D) \neq \text{Pos}_{\Delta \setminus \{ C_i \}}(D) \), then \( C_i \) is called superfluous relative to \( D \) in \( \Delta \); otherwise, \( C_i \) is called indispensable in \( \Delta \). Let \( P \) be a subset of \( \Delta \). If each element in \( P \) is indispensable in \( \text{Pos}_\Delta(D) = \text{Pos}_{\Delta \setminus \{ C \}}(D) \), then \( P \) is called a reduct of \( \Delta \) relative to \( D \). The set of all indispensable elements relative to \( D \) in \( \Delta \) is called the relative core of \( \Delta \) and is denoted as \( \text{Core}_0(\Delta) \).

From the definition of positive domain, we can easily obtain the following properties.

**Theorem 4.1.** Let \((U, \Delta, D = \{ d \})\) be an inconsistent covering decision system. Then,

1. For any \( x_i \in U \), if \( x_i \in \text{Pos}_\Delta(D) \), then \( \Delta_x \subseteq \{ x_i \} \); if \( x_i \notin \text{Pos}_\Delta(D) \), then for all \( x_k \in U \), \( \Delta_x \subseteq \{ x_i \} \) is not true.
2. For any \( P \subseteq \Delta \), \( \text{Pos}_{\Delta \setminus \{ C \}}(P) = \text{Pos}_{\Delta \setminus \{ C \}}(D) \) if and only if for all \( x_i \in U \), \( \Delta_x \subseteq \{ x_i \} \) if and only if \( P_x \subseteq \{ x_i \} \).

**Theorem 4.1** shows that if a sample belongs to the positive domain, then all samples in its neighborhood belong to the positive domain. On the contrary, if a sample does not belong to the positive region, the samples in its neighborhood are not pure. **Theorem 4.1** (2) and (3) show that a subset of attributes can keep the positive domain of a decision invariant if and only if it needs to ensure the positive domain of each decision equivalence class is invariant. The following theorem presents a sufficient and necessary condition that enables one to judge whether a conditional attribute (covering) is superfluous.

**Theorem 4.2.** Let \((U, \Delta, D = \{ d \})\) be an inconsistent covering decision system, \( C \in \Delta \). Then, \( \text{Pos}_{\Delta \setminus \{ C \}}(D) \neq \text{Pos}_{\Delta \setminus \{ C \}}(D) \) if and only if there is at least one pair of \( x_i, x_j \in U \) such that if they satisfy one of the following conditions, then \( x_j \notin \Delta_x \Rightarrow x_j \in \{ \Delta \setminus \{ C \} \} \).

1. \( x_i \in \text{Pos}_\Delta(D) \) and \( x_j \notin \text{Pos}_\Delta(D) \);
2. \( x_i, x_j \in \text{Pos}_\Delta(D) \) and \( \{ x_i \} \cap \{ x_j \} = \emptyset \).

**Proof.** We denote \( P = \Delta \setminus \{ C \} \).

\( \Leftrightarrow \) Assume that there is a pair of \( x_i, x_j \in U \) satisfying \( x_i \in \text{Pos}_\Delta(D) \) and \( x_j \notin \text{Pos}_\Delta(D) \). Then by **Theorem 4.1** (1), we have that \( \Delta_x \subseteq \{ x_i \} \) holds and that \( \Delta_x \subseteq \{ x_i \} \) is not true, which implies \( x_j \notin \Delta_x \) by the reflexivity of \( \Delta_x \). Because \( x_j \notin \Delta_x \Rightarrow x_j \in P_x \), it follows from the transitivity of \( P_x \) that \( P_x \subseteq P_x \). Assume that \( P_x \subseteq \{ x_i \} \). By \( \Delta_x \subseteq P_x \), we have \( \Delta_x \subseteq \{ x_i \} \), which implies \( x_j \notin \text{Pos}_\Delta(D) \). This is a contradiction. It follows that \( P_x \subseteq \{ x_i \} \) is not true. Thus \( x_i \in \{ \Delta_x \cap \{ x_i \} \} \) and \( x_j \notin \{ \Delta_x \cap \{ x_i \} \} \), which implies \( \text{Pos}_{\Delta \setminus \{ C \}}(D) \neq \text{Pos}_{\Delta \setminus \{ C \}}(D) \).

\( \Rightarrow \) Assume that \( \Delta_x \subseteq \{ x_i \} \), \( \Delta_x \subseteq \{ x_i \} \), and \( \Delta_x \subseteq \{ x_i \} \), which implies \( x_j \notin \Delta_x \) by the reflexivity of \( \Delta_x \). Because \( x_j \notin \Delta_x \Rightarrow x_j \in P_x \), it follows that \( P_x \subseteq \{ x_i \} \) is not true. Thus \( x_i \in \{ \Delta_x \cap \{ x_i \} \} \) and \( x_j \notin \{ \Delta_x \cap \{ x_i \} \} \), which implies \( \text{Pos}_{\Delta \setminus \{ C \}}(D) \neq \text{Pos}_{\Delta \setminus \{ C \}}(D) \).

If \( \Delta_x \subseteq \{ x_i \} \), then \( x_i, x_j \in \text{Pos}_\Delta(D) \) and \( \{ x_i \} \cap \{ x_j \} = \emptyset \), which implies that \( x_i, x_j \) satisfy condition (2). If \( \Delta_x \subseteq \{ x_i \} \) is not true, then \( x_i \in \text{Pos}_\Delta(D) \) and \( x_j \notin \text{Pos}_\Delta(D) \), which implies that \( x_i, x_j \) satisfy the condition (1). No matter the type of situation in which \( x_i, x_j \) meet, we always have that \( x_i \notin \Delta_x \) and \( x_j \notin P_x \), i.e., \( x_j \notin \Delta_x \Rightarrow x_j \in \{ \Delta \setminus \{ C \} \} \).
The first condition means that one sample belongs to the positive domain of the decision and that the other does not. The second item means that both samples belong to the positive domain but have different decisions. We call the two conditions covering reduction conditions. An indispensable attribute must be retained in reducts; otherwise, the positive domain of decision will change. An attribute is indispensable if and only if there is at least one pair of samples meeting the covering reduction conditions and this pair can only be distinguished by this attribute. Generally speaking, we always find all indispensable attributes first when searching for a reduct; doing so can guarantee the quality of the selected reducts and the efficiency of the searches for reducts.

**Theorem 4.3** (Judgment theorem of attribute reduction). Suppose that \((U, \Lambda, D = \{d\})\) is an inconsistent covering decision system, \(P \subseteq \Lambda\). Then, \(Pos_\Delta(D) = Pos_\Lambda(D)\) if and only if for any \(x_i, x_j \in U\), no matter what case, they meet in the following conditions. Then, \(x_j \not\in \Delta_x \Rightarrow x_j \not\in P_x\).

1. \(x_i \in Pos_\Delta(D)\) and \(x_j \not\in Pos_\Delta(D)\);
2. \(x_i, x_j \in Pos_\Delta(D)\) and \([x]_D \cap [x]_D = \emptyset\).

**Proof.** \(\Leftarrow\) For any \(x_i, x_j \in U\), if \(x_i \in Pos_\Lambda(D)\) and \(x_j \not\in Pos_\Lambda(D)\), then by **Theorem 4.1** (1), we have that \(\Delta_x \subseteq [x]_D\) holds and that \(\Delta_x \subseteq [x]_D\) is not true for any \(x_k \in U\). Assume that \(x_j \in \Delta_x\); then, \(\Delta_x \subseteq \Delta_x\) by the transitivity of \(\Delta_x\). Thus, \(\Delta_x \subseteq [x]_D\), which implies that \(x_j \not\in Pos_\Lambda(D)\). This is a contradiction. Hence, \(x_j \not\in \Delta_x\). If \(x_i, x_j \in Pos_\Lambda(D)\) and \([x]_D \cap [x]_D = \emptyset\), then by **Theorem 4.1** (1), we have that \(\Delta_x \subseteq [x]_D\) and \(\Delta_x \subseteq [x]_D\), which implies \(x_j \not\in \Delta_x\). Because \(x_j \not\in \Delta_x \Rightarrow x_j \not\in P_x\), it follows that \(P_x \subseteq [x]_D\). This means that \(\Delta_x \subseteq [x]_D \Rightarrow P_x \subseteq [x]_D\), which implies \(\Delta_x \subseteq [x]_D \Rightarrow P_x \subseteq [x]_D\) by \(\Delta_x \subseteq P_x\). By **Theorem 4.1** (3) we have \(Pos_\Lambda(D) = Pos_\Lambda(D)\).

\(\Rightarrow\) For any \(x, y \in U\), if \(x \in Pos_\Lambda(D)\) and \(x_j \not\in Pos_\Lambda(D)\), then by **Theorem 4.1** (1), we have that \(\Delta_x \subseteq [x]_D\) holds and that \(\Delta_x \subseteq [x]_D\) is not true for any \(x_k \in U\). Assume that \(x_j \in \Delta_x\); then, \(\Delta_x \subseteq \Delta_x\) by the transitivity of \(\Delta_x\). Thus, \(\Delta_x \subseteq [x]_D\), it is a contradiction. Hence, \(x_j \not\in \Delta_x\). Because \(Pos_\Lambda(D) = Pos_\Lambda(D)\), it follows from **Theorem 4.1** (3) that \(\Delta_x \subseteq [x]_D \iff P_x \subseteq [x]_D\). Combined with the fact that \(\Delta_x \subseteq [x]_D\) is not true for any \(x_k \in U\), it follows that \(P_x \subseteq [x]_D\) is not true for any \(x_k \in U\). Assume that \(x_j \in P_x\). Then, \(P_x \subseteq P_x\) by the transitivity of \(P_x\). Thus, \(P_x \subseteq [x]_D\), which is a contradiction. Hence, \(x_j \not\in P_x\), i.e., \(x_j \not\in \Delta_x \Rightarrow x_j \not\in P_x\).

If \(x_i, x_j \in Pos_\Delta(D)\) and \([x]_D \cap [x]_D = \emptyset\), then by **Theorem 4.1** (1), we have that \(\Delta_x \subseteq [x]_D\) and \(\Delta_x \subseteq [x]_D\), which implies \(x_j \not\in \Delta_x\). Because \(Pos_\Delta(D) = Pos_\Delta(D)\), it follows from **Theorem 4.1** (3) that \(\Delta_x \subseteq [x]_D \iff P_x \subseteq [x]_D\) and \(\Delta_x \subseteq [x]_D \iff P_x \subseteq [x]_D\), which implies that \(x_j \not\in P_x\), i.e., \(x_j \not\in \Delta_x \Rightarrow x_j \not\in P_x\). \(\square\)

This theorem is the basis for constructing a discernibility matrix. From **Theorems 3.1** (2) and 4.3, we have the following definition.

**Definition 4.4.** Let \((U, \Lambda, D = \{d\})\) be an inconsistent covering decision system, where \(U = \{x_1, x_2, \ldots, x_n\}\) and \(\Lambda = \{C_i: i = 1,2, \ldots, m\}\). By \(M(U, \Lambda, D)\), we denote an \(n \times n\) matrix \((c_{ij})\), which is called the discernibility matrix of \((U, \Lambda, D)\), such that if any \(x_i, x_j \in U\) satisfy one of the following conditions, then \(c_{ij} = \{C \in \Delta: x_j \not\in C\}\); otherwise, \(c_{ij} = \Delta\).

1. \(x_i \in Pos_\Delta(D)\) and \(x_j \not\in Pos_\Delta(D)\);
2. \(x_i, x_j \in Pos_\Delta(D)\) and \([x]_D \cap [x]_D = \emptyset\).

**Definition 4.4** shows that we have to distinguish each pair of samples that meet the above two conditions by some attributes. This discrimination ensures that the positive domain of the decision is invariant. The concept of constructing a discernibility matrix is similar to that used in classical rough sets.

In [5], a reduction method based on discernibility matrix was also proposed to compute all reducts. To compare these methods, the reduction method proposed in [5] is presented as follows.

**Definition 4.5.** Let \((U, \Lambda, D = \{d\})\) be an inconsistent covering decision system, where \(U = \{x_1, x_2, \ldots, x_n\}\) and \(\Lambda = \{C_i: i = 1,2, \ldots, m\}\). By \(M(U, \Lambda, D)\), we denote an \(n \times n\) matrix \((c_{ij})\), which is called the discernibility matrix of \((U, \Lambda, D)\), such that if \(x_i, x_j \in U\) satisfy

1. \(\Delta_x \subseteq Pos_\Delta(D)\) and \(\Delta_y \not\subseteq Pos_\Delta(D)\), then
   \[
   c_{ij} = \left\{ \begin{array}{ll}
   \{C \in \Delta: \Delta_x \subseteq C \} & \text{if} \ \Delta_x \subseteq \Delta_y \\
   \{C \in \Delta: \Delta_y \not\subseteq C \} & \text{if} \ \Delta_y \not\subseteq \Delta_x
   \end{array} \right.
   \]
2. \(\Delta_x, \Delta_y \subseteq Pos_\Delta(D)\) and \([x]_D \cap [x]_D = \emptyset\), then
\[ c_j = \{ C \in \Lambda : (C_x \not\subset C_y) \land (C_y \not\subset C_x) \} \cup \{ C_i \land C_i : (C_{x_i} \subset C_y) \land (C_{y_i} \subset C_x) \} \]

(3) otherwise \( c_j = \Lambda \).

By the fact that \( = \) for any covering \( \Lambda \), which implies that \( \) meet one of the covering reduction conditions. Thus, \( \) and \( \) meet, and \( \). It follows from Theorem 4.3 that \( \). This means that only \( \) and \( \) meet, we have that \( \).

Theorem 4.6. Let \((U, \Lambda, D = \{d\})\) be an inconsistent covering decision system. Then,

1. For any \( P \subseteq \Lambda \), \( P \cap c_j \neq \emptyset \) for any \( i, j \leq n \) if and only if \( \text{Pos}_P(D) = \text{Pos}_{\Lambda}(D) \).
2. \( \text{Core}_{\Lambda}(\Lambda) = \{ C \in \Lambda : c_j = \{ C \} \} \).

Proof. (1) \( \Rightarrow \) If \( x_j \) and \( x_i \), \( \) do not meet conditions (1) and (2) in Definition 4.4, then \( c_j = \Lambda \). Hence, \( P \cap c_j \neq \emptyset \). In the following, we prove the other cases:

Assume that there exist \( i_0, j_0 \leq n \) such that \( P \cap c_{i_0} = \emptyset \) and \( \emptyset \).

By Definition 4.4 we have \( x_{i_0} \in C_{x_i} \) for \( \forall C \in P \), which implies \( x_{i_0} \in P_{x_j} \). Because \( \text{Pos}_P(D) = \text{Pos}_{\Lambda}(D) \), it follows from Theorem 3.3 that \( x_{i_0} \in P_{x_j} \) for any \( C \in \Lambda \). Hence, \( c_{i_0} = \emptyset \), which is a contradiction to the assumption.

\( \Rightarrow \) Because \( P \cap c_j \neq \emptyset \) for any \( c_j \neq \emptyset \), we assume that \( C \in P \cap c_j \). No matter the conditions of Definition 4.4 in which \( x_i \) and \( x_j \) meet, we have that \( x_j \notin C_x \Rightarrow x_j \notin C_x \Rightarrow x_j \notin P_x \). It follows from Theorem 4.3 that \( \text{Pos}_P(D) = \text{Pos}_{\Lambda}(D) \).

(2) If \( C \in \text{Core}_{\Lambda}(\Lambda) \), then \( \text{Pos}_{\Lambda}(D) = \text{Pos}_{\Lambda}(\{C\})(D) \). By Theorem 4.2, there is at least one pair of \( x_i, x_j \in U \) meeting one of the covering relation conditions such that \( x_j \notin C_x \Rightarrow x_j \notin \{ \Lambda - \{C\} \} x_i \). Combined with \( \{ \Lambda - \{C\} \} x_i = \bigcap_{C \subseteq \Lambda - \{C\}} C_x \), it follows that \( x_j \in C_{x_i} \) for any covering \( C \in \Lambda - \{C\} \). Assume that \( x_j \in C_x \). By the fact that \( \Lambda_x = \{ \Lambda - \{C\} \} x_i \cap C_{x_i} \), it follows that \( x_j \in C_{x_i} \), which is a contradiction. Thus, \( x_j \notin C_x \). This means that only \( C \in \Lambda \) satisfies \( x_j \notin C_x \). Thus, \( c_j = \{ C \} \), which implies that \( C \in \{ C : C_j = \{ C \} \} \). Hence, \( \text{Core}_{\Lambda}(\Lambda) \subseteq \{ C \in \Lambda : c_j = \{ C \} \} \).

Conversely, if \( c_j = \{ C \} \) for \( x_i, x_j \in U \), it follows from Definition 4.4 that \( x_j \) and \( x_j \) meet one of the covering relation conditions and \( x_j \notin C_x \), which implies that \( x_j \in \{ \Lambda - \{C\} \} x_i \). It follows from Theorem 4.2 that \( C \) is indispensable relative to \( D \), i.e., \( C \in \text{Core}_{\Lambda}(\Lambda) \). Thus \( \text{Core}_{\Lambda}(\Lambda) \supseteq \{ C : C_j = \{ C \} \} \). Therefore, \( \text{Core}_{\Lambda}(\Lambda) = \{ C \in \Lambda : c_j = \{ C \} \} \). \( \square \)

Corollary 4.7. Suppose that \( P \subseteq \Lambda \). \( P \) is thus a reduct of \( \Lambda \) if and only if it is a minimal subset of \( \Lambda \) satisfying \( P \cap c_j \neq \emptyset \), for \( i, j \leq n \).

Suppose that \( \Lambda = \{ C_1, C_2, \ldots, C_n \} \) is a family of coverings on \( U \) and that the corresponding Boolean variable \( \text{C}_i \) (\( i \leq n \)) is defined for each covering \( C_\text{i} \leq n \). \( f(U, \Lambda, D) \) is a function on \( (U, \Lambda, D) \) and is defined as

\[ f(U, \Lambda, D)(C_1, C_2, \ldots, C_n) = \land (\lor (c_j)) \quad (i, j \leq n, c_j \neq \emptyset) \]

Then, \( f(U, \Lambda, D) \) is a Boolean function of \( (U, \Lambda, D) \) and is called a discernibility function or discernibility formula of \( (U, \Lambda, D) \), where \( (c_j) \) represents a disjunction operation among the elements in \( c_j \). By the discernibility function, we have the following theorem to compute all reducts.

The following example is employed to compare the reduction method proposed in this section with the method presented in [5].

Example 4.8 5. We now consider a house evaluation problem. Suppose that \( U = \{ x_1, x_2, \ldots, x_{10} \} \) is a set of nine houses, \( E = \{ \text{price; color; structure; surroundings} \} \) is a set of attributes, the values of “price” are \( \{ \text{high; middle; low} \} \), the values of “color” are \( \{ \text{good; bad} \} \), the values of “structure” are \( \{ \text{reasonable; ordinary; poor} \} \), and the values of “surroundings” are \( \{ \text{quiet; a little noisy; noisy; quite noisy} \} \).

Price: \( C_1 = \{ x_1, x_2, x_3, x_4, x_6, x_7, x_8, x_9, x_{10} \}, x_3, x_4, x_6, x_7 \}, \{ x_3, x_4, x_5, x_6, x_7 \} \} \).

Structure: \( C_2 = \{ x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10} \}, x_9, x_{10} \}, \{ x_3, x_4, x_5, x_6, x_7 \} \) \).

Color: \( C_3 = \{ x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10} \}, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10} \} \).

Surroundings: \( C_4 = \{ x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10} \}, x_9, x_{10} \}, \{ x_8, x_9, x_{10} \} \).

The final decision \( D \) is

\[ U/D = \{ x_1, x_2, x_3, x_6 \} \} (\text{sale}), \{ x_4, x_5, x_7 \} \} (\text{further evaluation}), \{ x_8, x_9, x_{10} \} (\text{reject}) \].
The positive domain of $D$ relative to $\Delta$ is

$$\text{POS}_\Delta(D) = \bigcup_{X \in \text{POS}_D} \Delta(X) = \{x_1, x_2, x_3, x_6, x_{10}\}. $$

By Definition 4.5, we construct the discernibility matrix as follows:

$$\begin{bmatrix}
\Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\
\Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\
\{C_1, C_4\} & \{C_2, C_4\} & \{C_3, C_4\} & \Delta & \{C_2, C_3, C_4\} & \{C_2, C, 3, C_4\} & \{C_2, C_3, C_4\} & \Delta & \{C_2, C_3, C_4\} \\
\{C_1, C_3, C_4\} & \{C_1, C_3, C_4\} & \{C_1, C_3, C_4\} & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\
\Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\
\{C_2, C_4\} & \{C_2, C_3, C_4\} & \{C_2, C_3, C_4\} & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\
\{C_2, C_4\} & \{C_1, C_2, C_4\} & \{C_1, C_2, C_4\} & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\
\{C_2, C_4\} & \{C_2, C_3, C_4\} & \{C_2, C_3, C_4\} & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\
\end{bmatrix}$$

and

$$f(U, \Delta)(\overline{C_1}, \ldots, \overline{C_4}) = \wedge\{\vee(c_i) \mid 1 \leq i \leq 10\}$$

$$= C_2 \wedge (C_2 \vee C_4) \wedge (C_1 \vee C_4) \wedge (C_1 \vee C_3 \vee C_4) \wedge (C_2 \vee C_3 \vee C_4) \wedge (C_1 \vee C_2 \vee C_4) = C_2 \wedge (C_3 \vee C_4)$$

$$= (C_2 \wedge C_3) \vee (C_2 \wedge C_4).$$

Thus, $\text{RED}(\Delta, D) = \{(C_2, C_4), (C_2, C_3)\}$, $\text{Core}_0(\Delta) = \{C_2\}$. We can see that the discernibility matrix computed by the proposed method is simpler than the method presented in [5].

The results are also the same as those obtained in [5]. We can say that the proposed method is more efficient at computing all of the reducts than the method presented in [5].

5. Experimental analysis

Attribute reduction and feature selection have become the focus of pattern recognition and machine learning tasks. Selecting an optimal feature subset for classification learning is required for many practical problems. Classical rough sets are only applicable to selecting nominal features. Covering rough sets can be used to address more complex data, such as numerical and categorical data. In Sections 3 and 4, we present approaches for finding all reducts of conditional attributes based on covering rough sets, but the computational complexities increase exponentially with the number of samples. In real-world applications, it is not necessary to find all reducts. It is adequate to solve a practical problem by using a minimal reduct. For this reason, we develop a heuristic algorithm to find a minimal subset of features to approximate an optimal reduct. We call the heuristic algorithm the covering discernibility algorithm (CDA) and describe it as follows:

**Heuristic algorithm:** (Covering discernibility algorithm).

**Input:** A decision table $DT = (U, A, D)$ and Epsilon $\varepsilon$.\
**Output:** Reduct $\text{Red}$

**Step1:** $\text{Red} \leftarrow \{\}$, $\text{Core} \leftarrow \{\}$//Red and Core are the pools that contain the selected features and the features that belong to the core of a reduct, respectively.

**Step2:** For any attribute $a_k$ in an attribute set $A$, compute its discernibility matrix $M_k$ at a threshold $\varepsilon$ whose elements are 0 and 1. Here, 1 indicates that the corresponding objects are discernible, and 0 indicates that they are not.

**Step3:** Compute the discernibility matrices $\text{Total}_k\text{Discern}(A)$ and $\text{Additive}_k\text{Discern}(A)$ of the set $\{M_k: a_k \in A\}$ by disjunction and addition operation, respectively.

**Step4:** Find the core of the given decision table. For each $a_k \in A$, if the intersection of $\text{Additive}_k\text{Discern}(A)$ and $M_k$ is nonempty and some elements in $M_k \cap \text{Additive}_k\text{Discern}(A)$ are equal to 1, then put attribute $a_k$ into Core.

**Step5:** $\text{Red} \leftarrow \text{Core}$.

**Step6:** For each $b \in A - \text{Red}$, compute the discernibility matrix $\text{Total}_k\text{Discern}(\text{Red} \cup \{b\})$ of the set $\{M_k: a_k \in \text{Red} \cup \{b\}\}$. Select the attribute with maximum discernible power and put it into $\text{Red}$.

**Step7:** For each $c \in A - \text{Red}$, if the discernible power of $\text{Total}_k\text{Discern}(\text{Red} \cup \{c\})$ is equal to the discernible power of $\text{Total}_k\text{Discern}(A)$, then stop. Otherwise, go to step 6.

This algorithm can obtain a suboptimal reduct from a given decision table. To show that the proposed algorithm is an effective technique to handle complex data sets and is more efficient than the methods presented in [5], we employ eight datasets from the UCI Machine Learning Repository to verify the performance of the proposed algorithm. The information
from the eight datasets is described in Table 2. Of the eight data sets, five have numerical features: Glass, Ionosphere, Sonar, Wdbc and Wine; the remaining data sets have mixed features.

In the sequence of experiments, we compare the covering discernibility algorithm with both the reduction algorithm in [5] and the classical rough set method. Three indices are used in the comparison: (1) the number of selected attributes with different methods, (2) the running time of selecting a close to minimal reduct, and (3) the classification performances of attributes selected using different methods.

In the following analysis, we apply each of the three algorithms to search for attribute reducts on the eight datasets. To evaluate the quality of selected attributes, the SVM and 3NN learning algorithms are used to train classifiers and test classification accuracies based on 10-fold cross validation. As we know, the classical rough set method considers only categorical data. Hence, a fuzzy $c$-means clustering (FCM) technique is employed to discretize numerical data before reduction. The numeric attributes are discretized into four intervals. The parameter $c$ is used to control the size of the neighborhood of an object; it has a great impact on different data sets. Because classification performance varies with $c$, we select different values of $c$ for these datasets. The experimental results of these datasets are shown in Tables 3–6, where the covering discernibility algorithm is abbreviated by CDA, the method proposed in [5] by Covering, and the classical rough set method by FCM + PKRS.

From Tables 3, 5 and 6, we find that all three methods can effectively reduce the datasets’ attributes. The FCM + PKRS method selected the smallest number of attributes, and the classification accuracies of the reduced data are also lowest, most of which decrease a little compared with the original data. This result shows that discretization causes information loss. The reductions with the CDA and Covering methods produce higher classification accuracies and keep more attributes in reducts, which shows that there are still some useful attributes in the subsets deleted by FCM + PKRS.

It is notable that the CDA method has a slightly better classification performance than the Covering method in most cases, although the number of attributes selected by each method is comparable. Furthermore, in the majority of cases, the standard deviations of the classification accuracies produced by CDA are lower than those produced by the Covering method because CDA optimizes the Covering method, which is poorly designed. The superiority of the CDA method is also shown in

### Table 2
Description of data sets.

<table>
<thead>
<tr>
<th>No</th>
<th>Data sets</th>
<th>Samples</th>
<th>Numerical</th>
<th>Categorical</th>
<th>Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Credit</td>
<td>690</td>
<td>8</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Dermatology</td>
<td>366</td>
<td>1</td>
<td>33</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>Glass</td>
<td>214</td>
<td>9</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>Heart</td>
<td>270</td>
<td>5</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>Ionosphere</td>
<td>351</td>
<td>34</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>Sonar</td>
<td>208</td>
<td>60</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>Wdbc</td>
<td>569</td>
<td>30</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>Wine</td>
<td>178</td>
<td>13</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

### Table 3
Number of selected features with different reduction algorithms.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Raw data</th>
<th>CDA</th>
<th>Covering</th>
<th>FCM + PKRS</th>
<th>$c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit</td>
<td>15</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>0.050</td>
</tr>
<tr>
<td>Dermatology</td>
<td>34</td>
<td>21</td>
<td>18</td>
<td>8</td>
<td>0.375</td>
</tr>
<tr>
<td>Glass</td>
<td>9</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>0.275</td>
</tr>
<tr>
<td>Heart</td>
<td>13</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>0.125</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>34</td>
<td>11</td>
<td>9</td>
<td>7</td>
<td>0.150</td>
</tr>
<tr>
<td>Sonar</td>
<td>60</td>
<td>21</td>
<td>19</td>
<td>6</td>
<td>0.300</td>
</tr>
<tr>
<td>Wdbc</td>
<td>30</td>
<td>22</td>
<td>23</td>
<td>7</td>
<td>0.250</td>
</tr>
<tr>
<td>Wine</td>
<td>13</td>
<td>10</td>
<td>8</td>
<td>5</td>
<td>0.225</td>
</tr>
</tbody>
</table>

### Table 4
Running time of the reduction with different algorithms.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>CDA</th>
<th>Covering</th>
<th>FCM + PKRS</th>
<th>$c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit</td>
<td>29.35 ± 0.13</td>
<td>120.14 ± 3.81</td>
<td>19.20 ± 2.65</td>
<td>0.050</td>
</tr>
<tr>
<td>Dermatology</td>
<td>11.60 ± 0.12</td>
<td>142.52 ± 10.59</td>
<td>24.36 ± 0.43</td>
<td>0.375</td>
</tr>
<tr>
<td>Glass</td>
<td>0.73 ± 0.02</td>
<td>1.64 ± 0.01</td>
<td>1.38 ± 0.29</td>
<td>0.275</td>
</tr>
<tr>
<td>Heart</td>
<td>1.53 ± 0.09</td>
<td>11.7 ± 1.54</td>
<td>0.43 ± 0.07</td>
<td>0.125</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>8.50 ± 0.15</td>
<td>79.59 ± 2.65</td>
<td>2.27 ± 0.08</td>
<td>0.150</td>
</tr>
<tr>
<td>Sonar</td>
<td>4.15 ± 0.13</td>
<td>51.65 ± 2.19</td>
<td>2.05 ± 0.06</td>
<td>0.300</td>
</tr>
<tr>
<td>Wdbc</td>
<td>41.73 ± 0.10</td>
<td>175.38 ± 11.98</td>
<td>12.24 ± 2.61</td>
<td>0.250</td>
</tr>
<tr>
<td>Wine</td>
<td>1.00 ± 0.07</td>
<td>7.65 ± 0.40</td>
<td>0.49 ± 0.07</td>
<td>0.225</td>
</tr>
</tbody>
</table>
Table 4, from which we can see that in most datasets, the CDA method greatly reduces the time consumption of the Covering method and that the standard deviations of running time have the similar trends. However, both the CDA and Covering methods spend more time than FCM + PKRS. Let us analyze the reasons in the following.

The procedures that the three methods use to find reducts are basically the same: (1) compute the discernibility matrix, (2) compute the core, and (3) compute the reducts. The difference lies in that the CDA and Covering methods need to use some time to compute the neighborhoods of samples. However, because the CDA method overcomes the design flaws of the Covering method, the efficiency of searching for reducts is greatly improved as showed in Table 4, which corresponds to the theoretical analysis in Sections 3 and 4.

Now, we conduct a series of experiments to find the optimal parameter \( \varepsilon \). For perfectly observing the performance variation with \( \varepsilon \), we set the value of \( \varepsilon \) to vary from 0 to 1 with a step of 0.025. In the following, we display only the experimental results of these datasets with SVM in Figs. 1–8. The experimental results obtained using 3NN are roughly consistent those obtained using SVM.

Figs. 1–8 show the numbers of selected features and classification accuracies varying with the threshold \( \varepsilon \). We find that there are similar trends in these curves. Generally speaking, with an increase of the value of \( \varepsilon \), the numbers of selected features rise and, correspondingly, the classification accuracies increase, arrive at a peak, and remain unchanged. Finally, the numbers of se-

Table 5
Comparison of the classification accuracies of reduced data with SVM.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Raw data</th>
<th>CDA</th>
<th>Covering</th>
<th>FCM + PKRS</th>
<th>( \varepsilon )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit</td>
<td>84.78 ± 4.44</td>
<td>85.31 ± 6.11</td>
<td>84.78 ± 5.65</td>
<td>82.90 ± 4.77</td>
<td>0.050</td>
</tr>
<tr>
<td>Dermatology</td>
<td>96.72 ± 2.13</td>
<td>97.57 ± 2.37</td>
<td>94.76 ± 2.98</td>
<td>76.05 ± 6.70</td>
<td>0.375</td>
</tr>
<tr>
<td>Glass</td>
<td>69.56 ± 9.21</td>
<td>71.39 ± 9.07</td>
<td>69.56 ± 9.21</td>
<td>67.83 ± 7.57</td>
<td>0.275</td>
</tr>
<tr>
<td>Heart</td>
<td>76.67 ± 8.38</td>
<td>81.26 ± 7.24</td>
<td>81.37 ± 9.73</td>
<td>76.30 ± 7.24</td>
<td>0.125</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>90.59 ± 4.69</td>
<td>92.87 ± 4.52</td>
<td>92.30 ± 4.49</td>
<td>90.32 ± 8.43</td>
<td>0.150</td>
</tr>
<tr>
<td>Sonar</td>
<td>83.93 ± 11.45</td>
<td>84.18 ± 3.81</td>
<td>83.22 ± 9.87</td>
<td>79.80 ± 8.34</td>
<td>0.300</td>
</tr>
<tr>
<td>Wdbc</td>
<td>96.31 ± 2.80</td>
<td>97.57 ± 2.03</td>
<td>97.19 ± 2.22</td>
<td>96.14 ± 2.59</td>
<td>0.250</td>
</tr>
<tr>
<td>Wine</td>
<td>97.15 ± 3.99</td>
<td>97.15 ± 3.99</td>
<td>96.04 ± 3.79</td>
<td>94.93 ± 4.88</td>
<td>0.225</td>
</tr>
</tbody>
</table>

Table 6
Comparison of the classification accuracies of reduced data with 3NN.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Raw data</th>
<th>CDA</th>
<th>Covering</th>
<th>FCM + PKRS</th>
<th>( \varepsilon )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit</td>
<td>84.49 ± 4.74</td>
<td>84.77 ± 5.95</td>
<td>85.22 ± 4.96</td>
<td>84.49 ± 5.64</td>
<td>0.050</td>
</tr>
<tr>
<td>Dermatology</td>
<td>96.49 ± 2.56</td>
<td>96.18 ± 3.42</td>
<td>95.56 ± 4.01</td>
<td>73.15 ± 6.64</td>
<td>0.375</td>
</tr>
<tr>
<td>Glass</td>
<td>67.03 ± 6.73</td>
<td>67.03 ± 5.93</td>
<td>67.03 ± 6.73</td>
<td>66.73 ± 0.67</td>
<td>0.275</td>
</tr>
<tr>
<td>Heart</td>
<td>77.41 ± 9.48</td>
<td>81.48 ± 8.46</td>
<td>81.83 ± 7.25</td>
<td>79.93 ± 9.44</td>
<td>0.125</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>85.74 ± 5.74</td>
<td>86.87 ± 7.78</td>
<td>91.16 ± 5.64</td>
<td>84.93 ± 7.40</td>
<td>0.150</td>
</tr>
<tr>
<td>Sonar</td>
<td>84.94 ± 7.93</td>
<td>86.04 ± 7.28</td>
<td>84.17 ± 11.06</td>
<td>77.32 ± 11.11</td>
<td>0.300</td>
</tr>
<tr>
<td>Wdbc</td>
<td>97.01 ± 2.19</td>
<td>97.34 ± 1.38</td>
<td>97.08 ± 2.34</td>
<td>95.96 ± 2.87</td>
<td>0.250</td>
</tr>
<tr>
<td>Wine</td>
<td>96.11 ± 4.57</td>
<td>96.04 ± 3.79</td>
<td>94.38 ± 3.71</td>
<td>93.75 ± 5.68</td>
<td>0.225</td>
</tr>
</tbody>
</table>
Fig. 2. Number of selected features and accuracy varying with the threshold $\varepsilon$ (dermatology).

Fig. 3. Number of selected features and accuracy varying with the threshold $\varepsilon$ (glass).

Fig. 4. Number of selected features and accuracy varying with the threshold $\varepsilon$ (heart).
Selected features and the accuracies drastically decrease after a threshold. These curves show that the classification performance is stable and can provide a selection of an appropriate subset of features. The optimal positions of classification performance are different between these curves. Here, we recommend that $\varepsilon$ should take values in the interval $[0.05,0.6]$. 

![Graph](image1)

**Fig. 5.** Number of selected features and accuracy varying with the threshold $\varepsilon$ (ionosphere).

![Graph](image2)

**Fig. 6.** Number of selected features and accuracy varying with the threshold $\varepsilon$ (sonar).

![Graph](image3)

**Fig. 7.** Number of selected features and accuracy varying with the threshold $\varepsilon$ (wdbc).
6. Conclusion and future work

Attribute reduction is an important step in classification learning. In this paper, we propose a novel method of attribute reduction using covering rough sets to handle datasets with discrete and numerical attributes. We construct the discernibility matrix to compute the reducts in decision systems and analyze the structural properties of attribute reduction. In comparison with the approaches for reduction presented in [5], the time complexity of the proposed reduction method is relatively low. Based on a discernibility matrix, we develop a heuristic algorithm to find a minimal subset of attributes that approximate an optimal reduct. Experiments using UCI datasets show that the proposed reduction approaches represent an effective technique to address complex datasets.

If each covering is a partition, then the proposed method for computing reducts coincides with the traditional rough sets presented in [44]. Thus, the proposed method is a direct generalization of the traditional reduction method. From a theoretical viewpoint, we can compute all of the reducts of a covering decision system. Our study on this topic plays the same important theoretical role as previous studies [44,46] in traditional and generalized rough sets.

As shown in the experimental analysis, the proposed heuristic algorithm can effectively address medium-sized datasets. However, the computational complexity of discernibility matrix increases exponentially with the numbers of samples and attributes. Therefore, it is necessary to develop methods that avoid the construction of a discernibility matrix to conduct attribute reduction and feature selection for large datasets in the future.

Acknowledgements

The authors thank the Language Editor and other anonymous reviewers for their valuable suggestions. This work was supported by the National Basic Research Program of China (2009CB219801-3), the National Natural Science Foundation of China under Grants 61070242, 71171080 and 61222210, the Program for Liaoning Excellent Talents in University under Grant (LR2012039), the Natural Science Foundation of Hebei Province (F2012201023), and the Scientific Research Project of Hebei University (09265631D-2).

References
