Rule Extraction Algorithm of Ordered Decision Information System

MO Jing-lan
Department of Mathematics and Physics, Liuzhou Institute of Technology, Liuzhou 545616, Guangxi, China
100001178@gxust.edu.cn  E-mail: jinglan022@163.com

Abstract. In ordered decision information system, an improved LEM2 algorithm (DRI-LEM2) is proposed based on the generalized decision function, aiming at the low efficiency and low quality of LEM2 algorithm in extracting rules. The algorithm computes the candidate attribution-value pair set according to the generalized decision function, and removes redundant attribution-value pair from the candidate attribution-value pair set to gradually reduce the size of the property-value pair set. Through UCI experiments, it is proved that the improved LEM2 algorithm (DRI-LEM2) based on dominance relations can effectively improve the efficiency and quality of rules extraction.

1. Introduction

Rule extraction is a core problem in rough set theory [1-2]. Therefore, how to mine the knowledge hidden in various complex information systems and extract rules that are useful to decision makers are of great importance. At present, the research methods of decision rule extraction in classical rough set theory have been relatively perfect, but the research of rule extraction in order decision information system and complex order decision information system is not perfect. LERS (Learning from Examples Based on Rough Set)[3] is a data mining system developed by Grzymala-Busse, which is a rule extraction algorithm based on rough set theory. LEM2 is one of the core data extraction algorithms of the system. LEM2 algorithm is a very effective and widely used rule extraction algorithm based on Rough Sets theory. It is the core algorithm of the LERS (Learning from Examples based on Rough Sets) system. This algorithm is studied from the perspective of attribution-value pair of information system. When the coverage of attribution-value pair of upper and lower approximation set of information system is satisfied, the deterministic rules can be obtained from the lower approximation set and the uncertainty rules can be obtained from the upper approximation set.

Although LEM2 algorithm is able to obtain decision rules well, it also has the following limitations : (1) the acquisition rules of this algorithm are based on the condition that the upper and lower approximation sets are completely covered, and the rule extraction algorithm is too strict. As a result, some rules acquired have redundant attributes, and the algorithm efficiency needs to be improved [4-5]. (2) When LEM2 algorithm selects attribute-value pairs in the loop while, some unnecessary attribute-value pairs are selected due to redundancy caused by rough selection conditions, and unnecessary attribute-value pairs must be deleted by reverse elimination, so the efficiency of the algorithm needs to be improved [6]. (3) The LEM2 algorithm is based on the condition that the upper and lower approximation sets are completely covered, so it is not suitable for the extraction of upper and lower approximation set rules for calculation of a single object. It cannot guarantee that all objects
are covered, and the algorithm may fall into an infinite loop [6]. (4) Attribute-value pair of LEM2 algorithm is blind search, and multiple traversal of multiple loops leads to low algorithm efficiency [7].

Due to the limitations of LEM2 algorithm, many scholars have studied and improved the algorithm [8-14], mainly to obtain more deterministic rules by improving the calculation of upper and lower approximation sets, without much improvement in the algorithm itself. Xu Yi et al. improved the end conditions of the algorithm in literature [6] and effectively solved the LEM2 algorithm's problem of extracting rules from approximate sets that are not suitable for calculation of a single object, but the approximate sets that are only suitable for calculation of a single object have certain limitations. In literature [14], Ji Xia proposed an improved LEM2 rule extraction algorithm based on generalized decision function to improve the efficiency of LEM2 algorithm in extracting rules, but not enough deterministic rules were obtained. Many scholars had proposed various improved LEM2 algorithms [8-14], which extend the rough set model based on tolerance relation, similarity relation and characteristic relation, etc. None of these algorithms take into account the priority of attributes (attributes based on dominance relation), therefore, the LEM2 rule extraction of ordered decision information system is studied by using the theory of dominance rough set.

2. Ease of use

Definition 19 Let \( S = (U, AT, V, f) \) is an information system, Where \( U \) is a non-empty finite set of objects, called on the field; \( AT \) is the set of non-empty finite attributes; \( V = \bigcup_\alpha \forall A \in AT \) is the set of property values, and \( V_a \) is the range of the attribute \( a \), \( f: U \times AT \rightarrow V \) is an information function. For \( A \) given object \( x \), assign \( a \) value to the object \( x \) under the attribute \( a \).

Definition 29 Let \( S = (U, AT \cup D, V, f) \) is an ordered decision information system, Where \( U \) is a non-empty finite set of objects, called on the field; \( AT \cap D = \emptyset \), \( AT \) is the set of ordered condition attributes, \( D \) is the set of ordered decision attributes, the sets \( V_{AT} \) and \( V_D \) are all fully ordered. When the ordered decision information system contains only the ordered condition attribute set, it is called the ordered information system.

Definition 39 Given that an information system is also quaternion \( S = (U, A, V, f) \), let \( t = (a, v) \) represent an attribute-value pair, when \( a \in A \), \( v \in V_a \) represents the set of objects satisfying the attribute-value pair, and if \( S \) is A complete information system, then \( t = \{ x \in U | a(x) = v \} \).

Definition 49 Let \( S = (U, AT \cup D, V, f) \) is an ordered decision information system, where \( U \) is a non-empty finite set of objects, called on the field. \( C \) is the set of ordered condition attributes, \( D \) is the set of ordered decision attributes, \( AT = C \cup D \), \( \forall A \subseteq C \), the dominance relation determined by attribute set \( A \) is \( R^*_A = \{ (x, y) \in U \times U | \forall a \in A, f(x, a) \geq f(y, a) \} \), when \( \forall x, y \in U \), \( (x, y) \in R^*_A \), means that \( x \) is at least as good as \( y \), that \( x \) is not worse than \( x \), in terms of \( x \) and \( x \), \( x \) and \( x \) means that \( x \) is no better than \( x \).

The dominant relation determined by set \( D \) is \( R^*_D = \{ (x, y) \in U \times U | \forall d \in D, f(x, d) \geq f(y, d) \} \).

There are:
\[
[x]_A^+ = \{ x \in U | (x, x) \in R^*_A \} = \{ x \in U | \forall x, y \in U, f(x, a) \geq f(y, a) \}, [x]_D^+ = \{ x \in U | (x, x) \in R^*_D \} = \{ x \in U | \forall d \in D, f(x, d) \geq f(y, d) \}.
\]

In ordered decision information system, the decision attribute set \( D \) constitutes the division \( CL = \{ C_l, k, C_l \} \) in the argument domain. For \( \forall r, s \in [1, 2, L, n] \), if \( r > s \), the object in \( C_l \) is superior to the object in \( C_l \). Let \( C_l^r = \bigcup_{k \leq r} C_l, C_l^r = \bigcup_{k \leq r} C_l \), among them, \( t = 1, 2, L, n \), \( x \in C_l^r \) means that \( x \) belongs to at least part of decision class \( C_l \), \( x \in C_l^r \) means that \( x \) belongs to at most part of decision class \( C_l \).

Definition 59 Let \( S = (U, AT \cup D, V, f) \) is an ordered decision information system, where \( U \) is a non-empty finite set of objects, called on the field. \( C \) is the set of ordered condition attributes, \( D \) is the set of ordered decision attributes, \( AT = C \cup D \), \( R \) is the dominant relation defined on \( U \), \( X \subseteq U \), \( X \) is defined as the lower approximation and upper approximation of \( R^*_A \) in the dominant relation as follows:
3. LEM2 algorithm based on dominant relation

**Definition 6** Let \( S = (U, AT \cup D, V, f) \) be an ordered decision information system, for \( \forall A \subseteq AT \), \( R_i^A \) is a dominant relation of dominant decision information system, \( [x]_i^A = \{ y \in U \mid (y, x) \in R_i^A \} \), let's define the function \( \partial_i^A : U \rightarrow V \) to be \( \partial_i^A(x) = \min_{y \in [x]_i^A} f(y, d) \).

**Definition 7** Let \( S = (U, AT \cup D, V, f) \) be an ordered decision information system, let \( t = (a, v) \) be a property-value pair, among them \( a \in AT, v \in V \), \([t]\) is the set of all objects satisfying the attribute-value pair \( t = (a, v) \), i.e \( \{ x \in U \mid f(x, a) \geq v \} \).

If \( S \) is an incomplete ordered decision information system, then \([t] = \{ x \in U \mid f(x, a) \geq v \lor f(x, a) = v \} \).

In this paper, the traditional LEM2 algorithm is extended to the ordered information system, and the LEM2 rule acquisition algorithm based on the dominant relation is proposed. The expanded algorithm is referred to as DR-LEM2.

**DR-LEM2 algorithm** is described as follows:

**Input:** a lower approximation set \( B \) of the ordered decision information system.

**Output:** Overrides the attribute-value pair set \( Q \) of the lower approximation set \( B \).

**Begin**

// A collection of objects that evaluates property value pairs \([t]\)

\[ G := B \]
\[ Q := \emptyset \]

while \( G \neq \emptyset \)

begin

\[ T := \emptyset \]
\[ T(G) := \{ t \mid [t] \cap G \neq \emptyset \} ; \]

while \( T = \emptyset \) or \( [T] \not\subseteq B \)

begin

select a pair \( t \in T(G) \) with the highest attribute priority; if a tie occurs, select a pair \( t \in T(G) \)

such that \( |[t] \cap G| \) is maximum; if another tie occurs, select a pair \( t \in T(G) \) with the smallest

cardinality of \([t]\); if a further tie occurs, select first pair;

\[ T := T \cup \{ t \} ; \]
\[ G := [t] \cap G ; \]
\[ T(G) := \{ t \mid [t] \cap G \neq \emptyset \} ; \]
\[ T(G) := T(G) - T ; \]
\end{while}

for each \( t \) in \( T \) do

if \( [T - \{ t \}] \subseteq B \) then \( T := T - \{ t \} ; \)
\end{for}

\[ Q := Q \cap \{ T \} ; \]
\[ G := B - \bigcup_{T \in Q} \{ T \} ; \]

end { while }

for each \( T \) in \( Q \) do

...
if $\bigcup_{s \in Q \setminus \{T\}} [s] = B$ then $Q := Q \setminus \{T\}$.

end \ procedure.

Here is an example to illustrate the limitations of the DR-LEM2 algorithm.

Sample 1: An ordered decision information system is given in Table 1, in which condition attribute set \( AT = \{a_1, a_2, a_3\} \) and decision attribute set \( D = \{d\} \).

| \( U \) | \( a_1 \) | \( a_2 \) | \( a_3 \) | \( d \) |
|-------|-------|-------|-------|-------|
| \( x_1 \) | 6     | 2     | 3     | 3     |
| \( x_2 \) | 5     | 3     | 4     | 3     |
| \( x_3 \) | 3     | 3     | 3     | 2     |
| \( x_4 \) | 3     | 4     | 1     | 1     |
| \( x_5 \) | 2     | 2     | 4     | 1     |
| \( x_6 \) | 4     | 2     | 3     | 1     |

According to definitions 4 and 5:

\[ \begin{align*}
[x_1]^+_d &= \{x_1\}; & [x_2]^+_d &= \{x_2\}; & [x_3]^+_d &= \{x_3\}; & [x_4]^+_d &= \{x_1, x_2, x_3\}; \\
C_{[T]} &= \{x_1, x_2, x_3\}; & C_{[T]}^+ &= \{x_1, x_2, x_3\}; & C_{[T]}^- &= \{x_1, x_2, x_3\}; & C_{[T]}^0 &= \{x_1, x_2, x_3\}; \\
R_{[T]}^0 (C_{[T]}^0) &= \{x_1, x_2, x_3, x_4, x_5, x_6\}; & R_{[T]}^1 (C_{[T]}^0) &= \{x_1, x_2, x_3\}; & R_{[T]}^2 (C_{[T]}^0) &= \{x_1, x_2\}. 
\end{align*} \]

According to definitions 7:

\[ \begin{align*}
[t_1] &= [(a_1, 6)] = \{x_1\}; & [t_2] &= [(a_2, 5)] = \{x_1, x_2\}; & [t_3] &= [(a_3, 4)] = \{x_1, x_2, x_3, x_6\}; & [t_4] &= [(a_3, 1)] = \{x_1, x_2, x_3, x_4, x_6\}; \\
[t_5] &= [(a_2, 4)] = \{x_1, x_2, x_3, x_6\}; & [t_6] &= [(a_2, 3)] = \{x_1, x_2, x_4, x_6\}; & [t_7] &= [(a_1, 2)] = \{x_1, x_2, x_3, x_4, x_6\}; \\
[t_8] &= [(a_1, 4)] = \{x_1, x_2, x_3\}; & [t_9] &= [(a_1, 1)] = \{x_1, x_2, x_3, x_6\}. 
\end{align*} \]

According to DR-LEM2 algorithm, first extract the determination rule from \( R_{[T]}^2 (C_{[T]}^0) = \{x_1, x_2, x_3\} \), then \( B = \{x_1, x_2, x_3\} \), the algorithm starts with \( G = B = \{x_1, x_2, x_3\} \), inside loop “While”, first select \( t_2, T = t_2 \), due to the \( [T] = \{x_1, x_2\} \subseteq B \), at the end of the inner loop, i get a definite rule if \( f(x, a_2) \geq 3 \), then \( x \in C_{[T]}^2 \). Since then \( G = B - \bigcup_{T} [T] = \emptyset \), the algorithm ends.

Next, extract the determination rule from \( R_{[T]}^2 (C_{[T]}^0) = \{x_1, x_2, x_3\} \), then \( B = \{x_1, x_2, x_3\} \), inside loop “While”, since \( t_4 \) and \( t_{10} \) have the same priority, \( t_4 \) and \( t_{10} \) are first selected. Due to the \( [T] = \{x_1, x_2, x_3, x_4, x_6\} \cap \{x_1, x_2, x_4, x_6\} \cap \{x_1, x_2, x_5, x_6\} \cap \{x_1, x_2, x_4, x_5, x_6\} \cap \{x_1, x_2, x_3, x_4, x_6\} \cap \{x_1, x_2, x_3, x_5, x_6\} \cap \{x_1, x_2, x_3, x_4, x_5, x_6\} \cap \{x_1, x_2, x_3, x_4, x_5, x_6\} \subseteq B \), and then we go on to choose \( t_5, T = t_5 \), \( t_3 \) and \( t_{11} \), \( T = t_5 \cup t_{10} \cup t_3 \cup t_{11} \cup [T] = \{x_1, x_2, x_3, x_4, x_6\} \cap \{x_1, x_2, x_3, x_4, x_5, x_6\} \cap \{x_1, x_2, x_3, x_4, x_5, x_6\} \cap \{x_1, x_2, x_3, x_4, x_5, x_6\} \cap \{x_1, x_2, x_3, x_4, x_5, x_6\} \cap \{x_1, x_2, x_3, x_4, x_5, x_6\} \cap \{x_1, x_2, x_3, x_4, x_5, x_6\} \cap \{x_1, x_2, x_3, x_4, x_5, x_6\} \subseteq B \), and then we go on to choose \( t_4 \), \( T = t_4 \cup t_5 \cup t_6 \cup t_7 \cup t_8 \cup t_9 \cup [T] = \{x_1, x_2, x_3, x_4, x_6\} \cap \{x_1, x_2, x_3, x_4, x_5, x_6\} \cap \{x_1, x_2, x_3, x_4, x_5, x_6\} \cap \{x_1, x_2, x_3, x_4, x_5, x_6\} \cap \{x_1, x_2, x_3, x_4, x_5, x_6\} \cap \{x_1, x_2, x_3, x_4, x_5, x_6\} \cap \{x_1, x_2, x_3, x_4, x_5, x_6\} \cap \{x_1, x_2, x_3, x_4, x_5, x_6\} \cap \{x_1, x_2, x_3, x_4, x_5, x_6\} \subseteq B \), the inner loop ends. In the 7 property-value pairs selected, it is obvious that \( t_4 \), \( t_5 \) and \( t_7 \) are redundant, after the reverse elimination of \( T = t_5 \cup t_7 \), a determination rule if \( f(x, a_1) \geq 4 \land f(x, a_2) \geq 3 \), then \( x \in C_{[T]}^2 \) is obtained. Since then, \( G = B - \bigcup_{T} [T] = \{x_1, x_2\} \), when you
extract a rule from it, inside loop “While”, select \( t \) first, \( T = t \), because, \([T] = \{x_i\} \subseteq G = B\), at the end of the inner loop, we get a definite rule if \( f(x, a_i) \geq 6\), then \( x \in C_i^{x_{+}} \). Since then, \( G := B - \bigcup_{t \in Q} [T] = \{x_i\}\), at this point, since \( T(G) = \{t | [t] \cap G \neq \emptyset\} \) cannot be satisfied all the time, the algorithm ends.

Based on the analysis of the above solution results, it is shown that DR-LEM2 algorithm has the following limitations:

1. The selection of attribute value pairs in the inner loop “while” statement is blind. Every time the candidate set of attribute value pairs is selected, all attribute value pairs must be traversed. The multiple traversal of multiple loops leads to low efficiency of DR-LEM2 algorithm.
2. It is possible to select redundant attribute value pairs in the inner loop “while” statement, and these redundant attribute value pairs are finally deleted by reverse elimination, which greatly reduces the efficiency of the algorithm.
3. The captured rule is not the simplest rule or leaves out potential rules.

Aiming at the limitation of DR-LEM2 algorithm, this paper improves LEM2 algorithm based on dominance relations.

**Definition 8** A collection of \( N = [T] \), where \( T \) is defined in the same way as DR-LEM2. The redundant attribute value pair \( R \) is defined as

\[
R = \begin{cases} 
\{t \in T(G) | N \subseteq [t] \}, & N \neq \emptyset \\
\emptyset, & N = \emptyset
\end{cases}
\]

After each selection of attribute value pair from the candidate attribute value pair set, \( T(G) = T(G) - T - R \) is performed, that is, the candidate attribute value pair set not only deletes the selected attribute value pair, but also deletes the redundant attribute value pair set \( R \) to avoid the reverse elimination step and improve the efficiency of the algorithm.

4. **Improved LEM2 Algorithm Based on Advantageous Relationship (DRI-LEM2 for short)**

The improvement of LEM2 algorithm based on dominance relations in this paper mainly includes the following four aspects:

1. Define the decision advantage function and change the “while” loop in DR-LEM2 algorithm to blindly select attribute value pairs to heuristic search.
2. Redundant attribute value pairs are removed directly after the attribute value pair is selected in the “while” loop of DR-LEM2 algorithm.
3. The determiner rule for the single condition property is printed directly without the reverse elimination step following the “while” loop.
4. The algorithm takes the intersecting attribute value pairs in the candidate attribute value set \( T(G) \) as the settlement condition of the algorithm.

The DRI-LEM2 algorithm is described as follows:

**Input:** A lower approximation set \( B \) of the ordered decision information system.

**Output:** Overrides the attribute-value pair set \( Q \) of the lower approximation set \( B \).

**Begin**

\[ G := B \]
\[ Q := \emptyset \]

while \( G \neq \emptyset \) begin

\[ T := \emptyset ; \]
\[ T(G) := \{t | [t] \cap G \neq \emptyset\} ; \]
\[ t-\text{Calculate}() ; \]

/A collection of objects that evaluates property value pairs \([t]\)
\[ \partial_{AR}^2 ([t]) \text{-Calculate}(); \text{The dominant decision function for calculating attribute value pairs} \]

while \( \{t \in T(G) | \partial_{AR}^2 ([t]) \geq \partial_{AR}^2 (B) \} \cup \{(t,t_i) | t_i \in T(G), [t_i] \cap [t] \subseteq G \} \neq \emptyset \)

while \( T = \emptyset \) and \( [T] \not\subseteq B \)

begin
attrpair_select(); //Select the property value pairs, in the following order of priority

1) \( \max \left( \frac{[t] \cap G}{[t]} \right) \); 

2) \( \min ([t]) \);

\( T := T \cup \{t\} \);

\( G := [t] \cap G \);

\( T(G) := \{t | [t] \cap G \neq \emptyset\} \);

\( N := [T] \);

if \( N \neq \emptyset \) then \( R = \{t \in T(G) | N \subseteq [t] \} \)

else \( R = \emptyset \);

\( T(G) := T(G) - T - R \);

end \{ while \}

for each \( t \) in \( T \) do

if \([T - \{t\}] \subseteq B \) then \( T := T - \{t\} \);

\( Q := Q \cap \{T\} \);

\( G := B - \bigcup_{T \in Q} [T] \);

end \{ while \}

for each \( T \) in \( Q \) do

if \( \bigcup_{S \in Q, T} [S] = B \) then \( Q := Q - \{T\} \);

end \{ procedure \}.

5. Example and algorithm performance analysis

Table 2 Take the sequential decision information system in Table 1 as an example.

According to DR1-LEM2 algorithm, First extract the determination rule from \( R_{AR}^2 (C_{I_1}^2) = \{x_1, x_2\} \), then \( B = \{x_1, x_2\} \), the algorithm starts with \( G = B = \{x_1, x_2\} \), first, the dominant decision function \( \partial_{AR}^2 ([t]) \) of the attribute value pair is calculated, because \( \partial_{AR}^2 ([t_1]) = \partial_{AR}^2 (B), \partial_{AR}^2 ([t_2]) = \partial_{AR}^2 (B) \), so we have two definite rules if \( f(x,a_1) \geq 6 \) then \( x \in C_{I_1}^2 \) and if \( f(x,a_1) \geq 5 \), then \( x \in C_{I_1}^2 \). In the “While” loop, select \( t_1 \) first after deleting the attribute value pairs \( t_1 \) and \( t_2 \), \( T = t_1 \), due to \( [T] = \{x_1, x_2, x_3\} \not\subseteq B \), the remaining attribute value pairs \( t_1 \) and the redundant attribute value pairs \( t_4 \), \( t_5 \), \( t_6 \), \( t_9 \), \( t_{11} \), and I'm going to go ahead and select \( t_9 \), \( T = t_1 \cup t_9 \), due to the \( [T] = \{x_1, x_2, x_3\} \cap \{x_2, x_3\} = \{x_2\} \subseteq B \), I get a definite rule if \( f(x,a_1) \geq 4 \wedge f(x,a_1) \geq 4 \), then \( x \in C_{I_1}^2 \). Since then, \( G = B - \bigcup_{T \in Q} [T] = \{x_1\} \), no conditions are met, the algorithm ends.

Then extract the determination rule from \( R_{AR}^2 (C_{I_2}^2) = \{x_1, x_2, x_3\} \), then \( B = \{x_1, x_2, x_3\} \), The algorithm starts with \( G = B = \{x_1, x_2, x_3\} \), first, the dominant decision function \( \partial_{AR}^2 ([t]) \) of the attribute value pair is
calculated, because \( \tilde{\delta}_{ax}(t_l) = \tilde{\delta}_{ax}(B) \), \( \tilde{\delta}_{ax}(t_r) = \tilde{\delta}_{ax}(B) \), so we have two definite rules if \( f(x, a_i) \geq 6 \), then \( x \in C_{i2} \) and if \( f(x, a_i) \geq 5 \), then \( x \in C_{i2} \). After deleting the attribute value pairs \( t_i \) and \( t_r \), make \( T = \emptyset \), because \( \{t_i, t_r\} \subseteq T(G) \), enter the inside loop “While” again, since \( t_i \) and \( t_r \) have the same priority, \( t_i \) and \( t_r \) are selected first, \( T = t_i \cup t_r \), due to the \( |T| = |\{x, x, x, x\} \cap \{x, x, x, x\} = \{x\} \subseteq B| \), I get a definite rule if \( f(x, a_i) \geq 4 \wedge f(x, a_i) \geq 3 \), then \( x \in C_{i2} \).

Residual attribute value pairs \( t_i \), \( t_r \), and redundant attribute value pairs \( t_5 \), \( t_7 \), \( t_9 \), \( t_{11} \). To make \( T = \emptyset \), at this time \( \{t_i\} \not\subseteq B \), \( \{t_r\} \not\subseteq B \), so break out of the loop and the algorithm ends.

Compared with DRI-LEM2 algorithm, DRI-LEM2 algorithm reduces algorithm complexity mainly in two aspects. One is to reduce the calculation cost of determining rules of single-condition attribute. The second is to change the blind search of attribute value to heuristic search to save search cost.

Given a sequential decision information system \( S = (U, AT \cup \{d\}, V, f) \), let’s say that the number of approximate sets \( R^a_{\geq}(C_{i2}) \) is \( |R^a_{\geq}(C_{i2})| \). The range of each conditional attribute is \( |V_a| (a \in AT) \) in DRI-LEM2 algorithm, the determined rule set of the sequential decision information system goes through three “While” loops. For convenience, it is assumed that each lower approximation set only gets the determined rule once, and the algorithm complexity is \( O(|R^a_{\geq}(C_{i2})| + |\bigcap_{a \in AT} T(G)|) \), among them \( |R^a_{\geq}(C_{i2})| \leq d \), \( |\bigcap_{a \in AT} T(G)| \) is the time complexity of searching for \( T(G) \), and \( |\bigcap_{a \in AT} T(G)| \) is the worst time complexity of searching for the attribute value pairs with the maximum intersection of objects in the innermost loop “While”, \( |\bigcap_{a \in AT} T(G)| = |\bigcap_{a \in AT} T(G)| \), so the total time complexity of DRI-LEM2 algorithm is \( O(|R^a_{\geq}(C_{i2})| + |\bigcap_{a \in AT} T(G)|) \).

The following simulation experiment further verifies the effectiveness of DRI-LEM2 algorithm in the rule extraction of sequential decision information system. The experimental environment is PIV4, 512M memory, Matlab2016b, Win7 operating system, 2GB memory, 500GB hard disk. Experimental data set: The data in Table 1 is recorded as system 1. The no. 5 child data set under Multiple Features dataset from UIC database is recorded as system 2, and Arrhythmia is recorded as system 3. System 1 contains 6 objects and 4 attributes. System 2 contains 204 objects and 240 attributes; System 3 contains 425 objects and 235 attributes. Rules were extracted by LEM2 algorithm, DR-LEM2 algorithm and DRI-LEM2 algorithm respectively in the three data sets. The performance of the algorithm was compared from four aspects of extracting rule number, average rule length, extracting rule time and classification accuracy. The extraction results of rules from three algorithms for three data sets were shown in Table 2.

| Data set   | Rule number | Average number of attributes | Get rule time/s | Classification accuracy % |
|------------|-------------|-----------------------------|----------------|--------------------------|
| LEM2       | system 1    | 1                           | 5.56           | 94.16                    |
|            | system 2    | 45                          | 5.29           | 91.05                    |
|            | system 3    | 52                          | 6.35           | 87.23                    |
| DR-LEM2    | system 1    | 1                           | 1.12           | 92.34                    |
|            | system 2    | 39                          | 5.11           | 89.47                    |
|            | system 3    | 48                          | 6.07           | 84.78                    |
| DRI-LEM2   | system 1    | 1                           | 1.21           | 91.67                    |
|            | system 2    | 35                          | 4.92           | 88.69                    |
|            | system 3    | 45                          | 5.86           | 80.11                    |

From Table 2: LEM2 algorithm than DR - LEM2 algorithm and DRI - LEM2 algorithm, compared to the rules for getting a more, this is because the improved DR - LEM2 algorithm and DRI - LEM2 algorithm is based on the generalized decision function, and does not require attribute-value
containing object completely belongs to the upper and lower approximation set, thus can quickly covering the upper and lower approximation set, can extract rules for less. In system 1 and System 2, DR-LEM2 algorithm and DRI-LEM2 algorithm have the same classification performance as traditional LEM2 algorithm, but the extracted rules are more concise, smaller in scale and more efficient. In system 3, drl-LEM2 algorithm is significantly superior to the traditional LEM2 algorithm in terms of the average number of attributes, rule acquisition time and classification accuracy, which indicates the effectiveness of the improved DRI-LEM2 algorithm in extracting rules.

6. Conclusion
In the ordered decision information system, aiming at the limitations of traditional LEM2 algorithm, it could be improved from the aspects of both the quality and efficiency of extracting rules. From the Angle of generalized decision function, an improved DRI-LEM2 algorithm based on dominance relation is proposed, and the improved algorithm obtains more concise rules and provides conditions for the quality of rules. Through detailed example analysis and comparison and UCI data experiment, it is proved that the improved DRI-LEM2 algorithm is more effective in extracting the rules of the ordered decision information system. On the premise of ensuring the quality and efficiency of rule extraction, how to extract as many deterministic rules as possible will be the main content of our future research.

Author
Corresponding author: Wei Bipeng.
Biography: Mo Jinglan was born in 1984. She is a associate professor. Her research interests include Rough Sets and Data Mining.

Acknowledgments
The author would like to thank the valuable suggestions from the anonymous reviewers and Professor. This research was supported by the basic ability improvement project of middle and young teachers in colleges and universities of Guangxi (No.2018KY0869 and 2019KY1098), the Natural Science Foundation of Guangxi Province of China (No.2019GXNSFAA245031).

References:
[1] Pawlak Z, rzymala-Busse J W, lowinski R, t al. Rough sets[J]. Communucations of the ACM, 1995, 38(11): 88-95.
[2] Pawlak Z, Skowron A. Rough sets: some extensions[J]. Information Science,2007,177(1):28-40.
[3] Grzymala-Busse J W. A new version of the rule induction system LE R S[J]. Fundamental Information, 1997, 31(1): 27-39.
[4] Stefanowski J. On rough set based approaches to induction of decision rules[C]// Proceedings of the International Conference on Rough Sets in Data Mining and Knowledge Discovery. Heidelberg: Springer-Verlag, 1998: 500-529.
[5] Yang H H, Wu C L. Rough sets to help medical diagnosis vidence from a Taiwan's clinic[J]. Expert Systems with Applications, 2009, 36(1): 9293-9298.
[6] Xu Yi, Li Long-shu, Li Xue-jun. Improved LEM2 rule induction algorithm[J]. System Engineering: Theory & Practice,2010, 30(10):1841-1849.
[7] Ji Xia, Li Long-shu, Qi Ping, et al. Fast algorithm for rule extraction based on heuristic LEM2 [J]. Journal of Chinese Computer Systems,2010, 31(11):2278-2281.
[8] Grzymala-Besse J W. MLEM2: a new algorithm for rule induction from imperfect data[C]// Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems. Annecy: ESIA-University de Savoie,2002:243-250.
[9] Grzymala-Besse J W. Data with missing attributes values: generalization of indiscernibility relation and rule induction[J]. Transactions on Rough Sets, 2004, 31(1):78-95.
[10] Kryszkiewicz M. Rough set approach to incomplete information system[J]. Information Sciences, 1998, 112(1): 39-49.
[11] Wang Guo-yin. Extension of rough set under incomplete information system[J]. Computer Research and Development, 2002, 39(10):1238-1243.
[12] Xu Xiao-dong, Shen Hui-zhang, Wang Zi-kai. A rule extraction algorithm based on asymmetrical similarity rough set[J]. Computer Simulation, 2008, 25(10):110-113.
[13] Xu Yi, Li Long-shu, Li Xue-jun. Improved LEM2 algorithm for incomplete information system[J]. Journal of South China University of Technology: Natural Science Edition, 2010, 38(11):104-109.
[14] Ji Xia, Li Long-shu, Xu Yi. Improved LEM2 rule induction algorithm based on generalized decision function [J]. Journal of South China University of Technology (Natural Science Edition), 2014, 5(42):143-148.
[15] Zhang Wenxiu, Qiu Guofang. Uncertain decision based on rough set[M]. Beijing: Qinghua University Press, 2005.