Rough Set Approaches to Incomplete Information Systems

Presenter: Zhen Zhao

Outline

- Motivation
- Review of Rough Set Theory
- Modeling incomplete information systems
- Some Rough Set approaches to incomplete information systems
- Conclusion
- Reference
Motivation

- Rough Set Theory, developed by Z. Pawlak, has made a great success in dealing with inexact, uncertain or vague knowledge in the area of Artificial Intelligence, Machine Learning, etc...
- However, it is based on complete information systems
- In practice, because of errors in data measuring, limitation of comprehension of data, limitation of acquiring data, etc, it is always impossible for us to obtain complete information, instead, incomplete information with missing values often occur when acquiring knowledge
- Then, how can we extend Rough Set Theory to cope with Incomplete Information Systems?

Review of Rough Set Theory

- Rough Set Theory was proposed by Z. Pawlak in 1982
- It classifies objects using upper-approximation and lower-approximation defined on an indiscernibility relation, which is a kind of equivalence relation
- Definition of rough set:
  - suppose we are given knowledge base \( K = (U, R) \), with each subset \( X \subseteq U \) and an equivalence relation \( R \in \text{IND}(K) \), we associate two subsets:
    - called the \textbf{R-lower} and \textbf{R-upper approximation} of \( X \) respectively.
  - There are some more denotations important in rough set:
Modeling incomplete information systems

- Information system (IS) is a triplet $\zeta = (O, AT, f)$, where
  - $O$ is a non-empty finite set of objects
  - $AT$ is a non-empty finite set of attributes, such that
    - for any $a \in AT$, $V_a$ is called the domain of an attribute $a$
  - $f_a : O \rightarrow V_a$ for any $a \in AT$, where $V_a$ is called the domain of an attribute $a$
  - $\text{Inf}(x) = \{(a, f_a(x)) | a \in AT\}$ is called an information vector of $x$

- Any attribute domain $V_a$ may contain special symbol “*” to indicate that the value of an attribute is unknown

- Any domain value different from “*” will be called regular

- A system which contains only regular values of all the attributes for all objects is called complete, otherwise, incomplete

- $\zeta' = (O', AT', f')$ is called an extension of $\zeta$ iff
  - $O' = O$
  - $AT' = AT$
  - $f(x) = f'(x)$ implies that $f'(x) = f(x)$ for any $a \in AT$ and $x \in O$

---

Modeling incomplete information systems (cont’)

- $\zeta'$ is a completion of $\zeta$ iff
  - $\zeta'$ is a complete information system, and
  - is an extension of $\zeta$

- $\text{EXTN}(\zeta)$: the set of all extensions of the system $\zeta$

- $\text{COMP}(\zeta)$: the set of all completions of the system $\zeta$

- $\zeta'$ is called $x$-extension of $\zeta$ iff
  - $\zeta' \in \text{EXTN}(\zeta)$
  - for any $y \in O \setminus \{x\}$: $\text{Inf}(y) = \text{Inf}'(y)$
  - for any $a \in AT$: $f'(x)$ is regular

- $\text{C}\text{EXTN}(\zeta, x)$: the set of all $x$-extensions of $\zeta$

- any attribute-value pair $(a, v)$, where $a \in AT$, $v \in V_a$ will be called an atomic property

- any atomic property or its conjunction will be called descriptor

- $\{|(a,v)|\}$: the set of objects possessing the atomic property $(a,v)$
Modeling incomplete information systems (cont’)

Example of extensions of $\zeta$

---

Table 1
Exemplary incomplete information system $\mathcal{V}$

<table>
<thead>
<tr>
<th>$\ell$</th>
<th>$u$</th>
<th>$v$</th>
<th>$c$</th>
<th>$d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>*</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>*</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>*</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

---

Fig. 1: $\mathcal{U} \in \text{EXTN}(\mathcal{V})$, $\mathcal{U}^* \in \text{CEXTN}(\mathcal{V}, 2)$, where $\mathcal{V}$ is the system from Table 1.
Modeling incomplete information systems (cont’)

- Examples of all completions of \( \zeta \)

Some Rough Set approaches to incomplete information systems

- Three Rough Set approaches to incomplete information systems
  - based on tolerance relation
  - based on non-symmetric similarity relation
  - based on valued tolerance relation
Tolerance relation

- Let $\zeta = (O, AT, f)$, a subset of attributes $A \subseteq AT$ determines a binary indiscernibility relation $IND(A)$:
  \[ IND(A) = \{(x, y) \in O \times O \mid \forall a \in A, f_a(x) = f_a(y)\} \]

- $IND(A)$ is an equivalence relation

- let us denote by $I_A(x)$ the set of objects $\{y \in O \mid (x, y) \in IND(A)\}$, the set of objects indiscernible with regard to their description in the system

- however, if the system is complete, these objects might possibly have different properties in reality

- thus, we define another relation, similarity relation, $SIM(A)$:
  \[ SIM(A) = \{(x, y) \in O \times O \mid \forall a \in A, f_a(x) = f_a(y) \text{ or } f_a(y) = *\} \]

- two objects are similar if they might have the same properties in reality

- let us denote by $S_A(x)$ the set of possibly indiscernible objects
  \[ \{y \in O \mid (x, y) \in SIM(A)\} \]

---

Tolerance relation (cont’)

- We can now refine the lower and upper approximation for incomplete information systems
  \[ \underline{\text{L}} \text{ower approximation of } X, \text{ iff } \underline{\text{L}} \text{ower } \]
  \[ \underline{\text{L}} \text{ower approximation of } X = \{ x \in O \mid S(x) \subseteq X \} \]

- upper approximation of $X$, iff $\overline{\text{U}} \text{pper }$
  \[ \overline{\text{U}} \text{pper approximation of } X = \{ x \in O \mid S(x) \cap X \neq \emptyset \} \]

- decision table (DT) is an information system $\zeta = \{O, AT \cup \{d\}, f\}$, where $d \in AT$ and $* \notin V_d$, $d$ is an attribute called the decision, any $a \in AT$ is called condition.

- decision rules discover the knowledge hidden in the decision table and is in the form of $t \rightarrow s$, where $t = \land(c, v), c \in A, v \in V_d \setminus \{\ast\}$, and $s = \lor(d, w), w \in V_d$

- A rule with a single decision value in the decision part will be called definite, otherwise, non definite
Tolerance relation (cont’)

Some definitions in complete decision tables:

- A decision rule is called **certain** in \( \xi \) iff \( t \rightarrow s \) is definite and \( ||t|| \subseteq ||s|| \)
- A decision rule is called **optimal certain** in \( \xi \) iff it is certain and no other rule constructed from a proper subset of atomic properties occurring in \( t \) is certain in \( \xi \)
- To determine the condition part of an optimal decision rule, we define that assuming \( A \subseteq AT, x \in O \) and \( I_{\alpha}(x) \subseteq \mathcal{I}(x) \), the set \( A \) is a **certain reduct** in \( \xi \) iff \( A \) is a minimal set such that:

\[
I(x) \subseteq \mathcal{I}(x)
\]

- **discernibility function:**

\[
\alpha(x, y) = \{ a \in AT \mid (x, y) \notin \text{SIM}(\{a\}) \}
\]

\[
\sum \alpha(x, y), \text{ denotes a Boolean expression which is equal to 1, if } \alpha(x, y) = \emptyset \text{ otherwise, let it be a disjunction of variables corresponding to attributes contained in } \alpha(x, y)
\]

---

Tolerance relation (cont’)

Some definitions in complete decision tables (cont’):

- Let \( x \in O \) and \( I_{\alpha}(x) \subseteq \mathcal{I}(x) \), \( \Delta(x) \) is a certain \( x \)-discernibility function iff

\[
\Delta(x) = \prod_{x \in O} \sum \alpha(x, y), \text{ where } Y := O \setminus I_{\alpha}(x).
\]

Example of generating optimal certain rules for complete information system

<table>
<thead>
<tr>
<th>Table 2: Incomplete car table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
</tbody>
</table>
Let us define the function \( \tilde{d}(x) : O \rightarrow \wp(V_x), A \subseteq AT \) as follows:
\[
\tilde{d}(x) = \{ f(y) \mid y \in S_d(x) \}
\]
\( \tilde{d} \) is called the \textit{generalized decision} in DT
\( \tilde{d}(x), x \in O \) determines to which decision classes the object \( x \) may be classified to according to the available information on \( x \)

Table 3 below is a completion of table 2

<table>
<thead>
<tr>
<th>Cat</th>
<th>Price</th>
<th>MakeType</th>
<th>Size</th>
<th>Mark Speed</th>
<th>ifd</th>
<th>cfA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>high</td>
<td>low</td>
<td>full</td>
<td>good</td>
<td>;good</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>low</td>
<td>low</td>
<td>full</td>
<td>good</td>
<td>;good</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>high</td>
<td>compact</td>
<td>low</td>
<td>poor</td>
<td>;good</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>good</td>
<td>;good, excellent</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>excellent</td>
<td>;good, excellent</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>good</td>
<td>;good</td>
<td></td>
</tr>
</tbody>
</table>

Certain rules are supported only by objects with single value generalized decisions, so we use object 1, 2, 3, 6 to generate optimal certain rules

- for object 1, \( I_d(1) = \{(d, \text{good})\} = \{1, 2, 4, 6\} \)
  - thus, \( Y = O \setminus I_d(1) = \{3, 5\}, \Lambda(1) = \sum a(1, 3) \Lambda(5) = (M \vee S) \land (X) = MX \land SY \)
  - there are two certain 1-reduct \( \{M, X\} \) and \( \{S, X\} \), therefore, object 1 supports two optimal certain rules:
    \( (M, \text{low}) \land (X, \text{low}) \rightarrow (d, \text{good}) \)
    \( (S, \text{full}) \land (X, \text{low}) \rightarrow (d, \text{good}) \)
- similarly, we can find optimal rules supported by objects 2, 3 and 6

Some definitions in incomplete decision tables:

- A decision rule \( t \rightarrow s \) is called \textit{certain} in \( \zeta \) iff it is certain in every completion of \( \zeta \)
- A decision rule is called \textit{optimal certain} in \( \zeta \) iff it is certain and no other rule constructed from a proper subset of atomic properties occurring in \( t \) is certain in \( \zeta \)
Tolerance relation (cont')

- Some definitions in incomplete decision tables (cont'):
  - **Proposition 1:**
    \( t \rightarrow s \) is certain in \( \xi \) if it is certain in every completion of \( \xi \) in which \( \| t \| = \emptyset \)
  - **Property 1:**
    \( \xi \) be an \( x \)-extension of \( \xi \), \( x \) supports a certain rule in \( \xi \)
    \[ \text{iff } S_{\xi'}(x) \subseteq I_{\| x \|(x)} \text{ in } \xi \]
    - Assuming that \( A \subseteq AT, x \in O, \xi \) is an \( x \)-extension of \( \xi \) and \( S_{\xi'}(x) \subseteq I_{\| x \|(x)} \)
      the set \( A \) is a certain \( x \)-reduct in \( \xi \) iff \( A \) is a minimal set such that
      \[ L_{\| x \|}(x) \subseteq L_{\| x \|}(x) \]
      in each completion \( \xi \) of \( \xi \)
    - Let \( x \in O \), \( \xi \) be an \( x \)-extension of \( \xi \), and \( S_{\xi'}(x) \subseteq I_{\| x \|(x)} \), \( \Delta'(x) \) is a certain \( x \)-discernibility function iff
      \[ \Delta'(x) = \prod_{\xi \in \text{COMPL}(\xi), x \in Y} \sum_{y \in Y} \alpha'(x,y) \text{, where } Y = O \setminus L_{\| x \|}(x) \]

---

Tolerance relation (cont')

- Some definitions in incomplete decision tables (cont'):
  - **Proposition 2:**
    Let \( x \in O \), \( \xi \) be an \( x \)-extension of \( \xi \) and \( S_{\xi'}(x) \subseteq I_{\| x \|(x)} \), then
    \[ \Delta'(x) = \prod_{\xi \in \text{COMPL}(\xi), x \in Y} \sum_{y \in Y} \alpha'(x,y) \text{, where } Y = O \setminus L_{\| x \|}(x) \]

- Example of generating optimal certain rules for incomplete information system
  - Consider table 2, object 5 will be used to generate rules
  - There are altogether four possible complete AT-descriptors of object 5, table 4 shows one of 5-extension of table 2, and we talk about this case only
    \[ S_{\xi'}(5) = [5] \text{ and } I_{\| 5 \|}(5) = \| \{ d, \text{excellent} \} \| = [\{ 5 \}], \text{ so } S_{\xi'}(5) \subseteq I_{\| 5 \|}(5) \]
    from property 1, we know that object 5 supports some certain rules in \( \xi \).
    \[ Y = O \setminus L_{\| 5 \|}(5) = \{ 1, 2, 3, 4, 6 \}; \Delta'(5) = \sum \alpha'(5,1) \land \sum \alpha'(5,2) \land \sum \alpha'(5,3) \land \sum \alpha'(5,4) \land \sum \alpha'(5,6) = (P \setminus X) \land (X) \land (S \setminus X) \land (P) \land (M) = PMX \]
Tolerance relation (cont')

- Example of generating optimal certain rules for incomplete information system (cont')
  - There is only one certain 5-reduct in $\xi^e$, so there's only one optimal certain rules supported by object 5 in $\xi^e$, that is $(P, \text{low}) \land (M, \text{low}) \land (X, \text{high}) \rightarrow (d, \text{excellent})$
  - Similarly, we can compute the optimal certain rules supported by object 5 for the rest cases.

<table>
<thead>
<tr>
<th>Table 4</th>
<th>A review of Table 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car</td>
<td>Price</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>1</td>
<td>high</td>
</tr>
<tr>
<td>2</td>
<td>low</td>
</tr>
<tr>
<td>3</td>
<td>*</td>
</tr>
<tr>
<td>4</td>
<td>high</td>
</tr>
<tr>
<td>5</td>
<td>low</td>
</tr>
<tr>
<td>6</td>
<td>low</td>
</tr>
</tbody>
</table>

Non-symmetric similarity relation

- In the previous approach, "*", null value, indicates that the value of the attribute does exist, it's just "missing" for some reasons.
- In this approach, we treat such values as non-existing, and unknown values are not allowed to be compared.
- Given an information table $IT=(A,C)$ and a subset of attributes $B \subseteq C$, a similarity relation $S$ is defined as follows:
  $\forall x, y \ S(x, y) \Leftrightarrow \forall c_j \in B : c_j(x) \neq * , c_j(y) = c_j(y)$
- For any object $x \in A$, two sets can be defined:
  - $R(x) = \{ y \in A | S(y, x) \}$: the set of objects similar to $x$
  - $R^{-1}(x) = \{ y \in A | S(x, y) \}$: the set of objects to which $x$ is similar.
- Based on these two sets, we can define the lower and upper approximation of a set $\Phi$
  - $\Phi^L = \{ x \in A | R^{-1}(x) \subseteq \Phi \}$: the lower approximation of $\Phi$
  - $\Phi^U = \cup \{ R(x) | x \in \Phi \}$: the upper approximation of $\Phi$
Non-symmetric similarity relation (cont’)

Theorem

Given an information table IT = (A,C) and a set \( \Phi \), the upper and lower approximations of \( \Phi \) obtained using a non symmetric similarity relation are a refinement of the ones obtained using a tolerance relation.

Example comparing two approaches

<table>
<thead>
<tr>
<th>A</th>
<th>a1</th>
<th>a2</th>
<th>a3</th>
<th>a4</th>
<th>a5</th>
<th>a6</th>
<th>a7</th>
<th>a8</th>
<th>a9</th>
<th>a10</th>
<th>a11</th>
<th>a12</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>*</td>
<td>2</td>
<td>2</td>
<td>*</td>
<td>3</td>
<td>1</td>
<td>*</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>C2</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>*</td>
<td>0</td>
<td>2</td>
<td>*</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>*</td>
<td>2</td>
<td>*</td>
<td>0</td>
<td>1</td>
<td>*</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>C4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>*</td>
<td>3</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C5</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ \Phi_C = \emptyset, \quad \Phi_C = \{a1, a2, a3, a4, a5, a7, a8, a9, a10, a11, a12\}, \]
\[ \Psi_C = \{a6\}, \quad \Psi_C = A \]
Non-symmetric similarity relation (cont')

Using the similarity relation, we obtain the following results:

\[ R^{-1}(a1) = \{a1\}, \quad R(a1) = \{a1, a11, a12\}, \]
\[ R^{-1}(a2) = \{a2, a3\}, \quad R(a2) = \{a2, a3\}, \]
\[ R^{-1}(a3) = \{a2, a3\}, \quad R(a3) = \{a2, a3\}, \]
\[ R^{-1}(a4) = \{a4, a5\}, \quad R(a4) = \{a4, a5, a11\}, \]
\[ R^{-1}(a5) = \{a4, a5\}, \quad R(a5) = \{a4, a5, a11\}, \]
\[ R^{-1}(a6) = \{a6\}, \quad R(a6) = \{a6\}, \]
\[ R^{-1}(a7) = \{a7, a9\}, \quad R(a7) = \{a7\}, \]
\[ R^{-1}(a8) = \{a8\}, \quad R(a8) = \{a8\}, \]
\[ R^{-1}(a9) = \{a9\}, \quad R(a9) = \{a7, a9, a11, a12\}, \]
\[ R^{-1}(a10) = \{a10\}, \quad R(a10) = \{a10\}, \]
\[ R^{-1}(a11) = \{a1, a4, a5, a7, a10, a11, a12\}, \quad R(a11) = \{a11\}, \]
\[ R^{-1}(a12) = \{a1, a9, a12\}, \quad R(a12) = \{a11, a12\}, \]

from which we can deduce that:

\[ \Phi = \{a1, a10\}, \quad \Phi = \{a1, a2, a3, a4, a5, a7, a10, a11, a12\}. \]
\[ \Psi = \{a6, a8, a9\}, \quad \Psi = \{a2, a3, a4, a5, a6, a7, a8, a9, a11, a12\}. \]

Valued tolerance relation

Review the previous two methods, given information table IT=(A,C), B \subseteq C

- Tolerance relation T
  \[ \forall x, y \in A \times A \quad T(x, y) \iff \forall c_j \in B \quad c_j(x) = c_j(y) \text{ or } c_j(x) = * \text{ or } c_j(y) = * \]
- Similarity relation S
  \[ \forall x, y \in A \times A \quad S(x, y) \iff \forall c_j \in B \quad c_j(x) = * \text{ or } c_j(x) = c_j(y) \]
- We now have T(a_{11}, a_{12}), T(a_{12}, a_{11}), S(a_{11}, a_{12}), S(a_{12}, a_{11}), how can we express the idea that
  - a_{12} is more similar to a_{11} than a_{11}
  - a_{11} is less similar to a_{12} than a_{12}

Thus, we want a kind of relation that captures such a similarity difference, that is, valued tolerance relation

Given a set \( \Phi \) and another set \( Z \subseteq A \), we need to define the degree by which \( Z \) approximates from the top or from the bottom the set \( \Phi \).
Valued tolerance relation (cont’)

- Define some logical connectives
  - A **negation** is a function \( N : [0, 1] \rightarrow [0, 1] \)
    representation: \( N(x) = 1 - x \)
  - A **T-norm** is a continuous, non-decreasing function \( T : [0, 1]^2 \rightarrow [0, 1] \)
    representation: conjunction, \( T(x, y) = \min(x, y), T(x, y) = xy \)
  - A **T-conorm** is a continuous, non-decreasing function \( S : [0, 1]^2 \rightarrow [0, 1] \)
    representation: disjunction, \( S(x, y) = \max(x, y), S(x, y) = x + y - xy \)
  - I.e. \( S(x, y) = N(T(N(x), N(y))) \) \( \iff \) De Morgan’s Law

- \( I(x, y) \), the degree by which \( x \) may imply \( y \) is again a function \( I : [0, 1]^2 \rightarrow [0, 1] \)
  - \( I(x, y) = S(N(x), y) \) is equivalent to \( x \rightarrow y = \text{def} \neg x \lor y \)
  - \( x \leq y \iff I(x, y) = 1 \)

Valued tolerance relation (cont’)

- Definitions of upper and lower approximations:
  - Given a set \( Z \subseteq A \), a set \( \Phi \) and attributes \( B \subseteq C \)
    - \( Z = \Phi_B \iff \forall z \in Z, \Theta(z) \subseteq \Phi \)
    - \( Z = \phi^B \iff \forall z \in Z, \Theta(z) \cap \Phi = \emptyset \)
      where \( \Theta(z) \) is the “indiscernibility” class of element \( z \)

- We can functionally translate the above definition into the logic connectives defined before, that is,
  - \( \forall x \ \phi(x) = \text{def} \ T_x \phi(x) \)
  - \( \exists x \ \phi(x) = \text{def} \ S_x \phi(x) \)
  - \( \phi \subseteq \psi = \text{def} \ T_x(I(\mu_\phi(x), \mu_\psi(x))) \)
  - \( \phi \cap \psi = \emptyset = \text{def} \ \exists x \ \phi(x) \land \psi(x) \)
    \[ = \text{def} \ S_x(T(\mu_\phi(x), \mu_\psi(x))) \]
Valued tolerance relation (cont’)

Based on these translations, we can deduce

\[
\mu_{\hat{\Phi}}^B(Z) = T_{x \in \Theta(z)}(I(R(z, x), \hat{x})), \\
\mu_{\Phi}^B(Z) = T_{x \in \Theta(z)}(S_{x \in \Phi}(T(R(z, x), \hat{x}))),
\]

where

- \( \mu_{\hat{\Phi}}^B(Z) \) is the degree for set \( Z \) to be a lower approximation of \( \Phi \);
- \( \mu_{\Phi}^B(Z) \) is the degree for set \( Z \) to be an upper approximation of \( \Phi \);
- \( \Theta(z) \) is the tolerance class of element \( z \);
- \( T, S, I \) are the functions previously defined;
- \( R(z, x) \) is the membership degree of element \( x \) in the tolerance class of \( z \);
- \( \hat{x} \) is the membership degree of element \( x \) in the set \( \Phi (\hat{x} \in \{0, 1\}) \).

Conclusion

- Non-symmetric relation is more informative than tolerance relation, though it is not as safe as tolerance relation.
- Valued tolerance relation is in the middle between two other relations.
- There are usually two ways to deal with incomplete information systems:
  - Indirect method that transforms incomplete information systems to complete one, data reparation
  - Direct method that extends the classical rough set theory
    - The three approaches I have introduced here are all direct methods
- Find more applications using rough set theory to deal with incomplete information systems.
Reference


The end

Thank you for your attention!