Rough Sets in Data Mining and Databases: Foundations and Applications

Dominik Ślęzak
Arkadiusz Wojna
Rough set systems

- LERS
- Rosetta
- Rose
- RSES
LERS – Learning from Examples based on Rough Sets

- Computes lower and upper approximations
- Computes two set of rules: certain and possible
- Classifies using the set of induced rules

Applications:
- medical decision making on board the International Space Station
- enhancing facility compliance under Sections 311, 312, and 313 of Title III. the Emergency Planning and Community Right to Know

https://people.eecs.ku.edu/~jerzygb/LERS.html
Rosetta

- GUI based on Rseslib ver. 1
- Analyzing tabular data
- Supports the overall data mining and knowledge discovery process
- Computes exact and approximate reducts
- Generates if-then rules from computed reducts

http://bioinf.icm.uu.se/rosetta
ROSE – Rough Set Data Explorer

- Data processing, including discretization
- Rough set based analysis of data
- Computes core and reducts
- Computes decision rules from rough approximations
- Applies rules to classification
- Includes variable precision rough set model
RSES – Rough Set Exploration System

- GUI based on Rseslib ver. 2
- Discretizes data
- Computes reducts
- Generates decision rules from reducts
- Classifies data using decision rules

http://logic.mimuw.edu.pl/~rses
Rough set open source

- Richard Jensen's programs
- Modlem
- RoughSets
- NRough
- Rseslib 3
Richard Jensen’s programs

- FRFS2: fuzzy-rough feature selection based on fuzzy similarity relations
- RSAR: rough set attribute reduction via QuickReduct
- EBR: entropy-based attribute reduction
- AntRSAR: searching for reducts using ant colony optimization
- GenRSAR: searching for reducts using genetic algorithm
- SimRSAR: searching for reducts using simulated annealing
- Some attribute reduction methods ported to Weka
- [http://users.aber.ac.uk/rkj/book/programs.php](http://users.aber.ac.uk/rkj/book/programs.php)
Modlem

- Rule induction using sequential covering algorithm
- Handles numerical attributes without discretization
- Available as Weka package (classification method)
- [https://sourceforge.net/projects/modlem](https://sourceforge.net/projects/modlem)
RoughSets

- R as programming language
- Rough set and fuzzy rough models and methods
- Implements
  - indiscernibility relations
  - lower/upper approximations
  - positive region
  - discernibility matrix
- Discretizations
- Feature selection
- Instance selection
- Rule induction
- Prediction/classification
- https://github.com/janusza/RoughSets
NRough

- C# as programming language
- Algorithms for
  - decision reducts
  - bireducts
  - decision reduct ensembles
  - rule induction
- Feature selection
- Classification
- http://www.nrough.net
Rseslib 3

- Java Library providing API
- Open Source (GNU GPL) available at GitHub
- Collection of Rough Set and other Machine Learning algorithms
- Modular component-based architecture
- Easy-to-reuse data representations and methods
- Easy-to-substitute components
- Available in Weka
- Graphical Interface
- http://rseslib.mimuw.edu.pl
Modularity example: rough set methods

- Rough Set Classifier
- AQ15 Rules
- Reduct Rules
- Reducts
- Discernibility matrix
- Logic
  - Prime implicants algorithm 1
  - Prime implicants algorithm 2

- Discretization
  - 1R
  - MD
  - ChiMerge

- Rules
  - AllGlobal
  - AllLocal
  - Johnson
  - Partial

Rough Sets in Data Mining and Databases: Foundations and Applications
Dominik Ślęzak, Arkadiusz Wojna
Discretizations

- Equal Width
- Equal Frequency
- 1R (Holte, 1993)
- Entropy Minimization Static (Fayyad, Irani, 1993)
- Entropy Minimization Dynamic (Fayyad, Irani, 1993)
- Chi Merge (Kerber, 1992)
- Maximal Discernibility Heuristic Global (H.S. Nguyen, 1995)
- Maximal Discernibility Heuristic Local (H.S. Nguyen, 1995)
Discretization: Entropy Minimization (top-down)

\[ Ent(S) = - \sum_{i=1}^{k} \frac{P(C_i,S)}{|S|} \log \left( \frac{P(C_i,S)}{|S|} \right) \]

Minimize:

\[ E(a,v,S) = \frac{|S_1|}{|S|} \cdot Ent(S_1) + \frac{|S_2|}{|S|} \cdot Ent(S_2) \]

S - data set
C_i – decision class
P(C_i,S) – number of records from decision class C_i in S
S_1, S_2 – partition of S split by a value v on an attribute a
Discretization: ChiMerge (bottom-up)

Merge the neighbouring pair of intervals with minimal:

\[
\chi^2(S_1, S_2) = \sum_{i=1}^{k} \frac{\left( P(C_i, S_1) - E(C_i, S_1) \right)^2}{E(C_i, S_1)} + \sum_{i=1}^{k} \frac{\left( P(C_i, S_2) - E(C_i, S_2) \right)^2}{E(C_i, S_2)}
\]

\(S_1, S_2\) - data sets from neighbouring intervals
\(C_i\) – decision class
\(P(C_i, S)\) – number of records from decision class \(C_i\) in \(S\)
\(E(C_i, S)\) – expected number of records from decision class \(C_i\) in \(S\)
Discretization: Maximal Discernibility (top-down)

Split a data set $S$ into $S_1$ and $S_2$ with the value $v$ maximizing:

$$\left| \left\{ (x, y) \in S_1 \times S_2 : \text{dec}(x) \neq \text{dec}(y) \right\} \right|$$
Discernibility matrix: all pairs

\[ M^{all}(x, y) = \{ a_i \in A : x_i \neq y_i \} \]
Discernibility matrix: pairs with different decisions

\[ M^{\text{dec}}(x,y) = \begin{cases} a_i \in A : x_i \neq y_i & \text{if } \text{dec}(x) \neq \text{dec}(y) \\ \emptyset & \text{if } \text{dec}(x) = \text{dec}(y) \end{cases} \]

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td></td>
<td>bc</td>
<td></td>
<td>ac</td>
</tr>
<tr>
<td>x2</td>
<td>bc</td>
<td></td>
<td>abc</td>
<td></td>
</tr>
<tr>
<td>x3</td>
<td>abc</td>
<td></td>
<td>b</td>
<td></td>
</tr>
<tr>
<td>x4</td>
<td>ac</td>
<td></td>
<td>b</td>
<td></td>
</tr>
</tbody>
</table>
Discernibility matrix: pairs with different generalized decision

\[ M_{\text{gen}}(x, y) = \begin{cases} \{ a_i \in A : x_i \neq y_i \} & \text{if } \partial(x) \neq \partial(y) \\ \emptyset & \text{if } \partial(x) = \partial(y) \end{cases} \]

\[ \partial(x) = \left\{ d \in V_{\text{dec}} : \exists y \in U : \forall a_i \in A : x_i = y_i \land \text{dec}(y) = d \right\} \]
Discernibility matrix: pairs with different both decisions

\[ M^{both}(x, y) = \begin{cases} \{ a_i \in A : x_i \neq y_i \} & \text{if } \text{dec}(x) \neq \text{dec}(y) \land \partial(x) \neq \partial(y) \\ \emptyset & \text{if } \text{dec}(x) = \text{dec}(y) \lor \partial(x) = \partial(y) \end{cases} \]

\[ \partial(x) = \left\{ d \in V_{\text{dec}} : \exists y \in U : \forall a_i \in A : x_i = y_i \land \text{dec}(y) = d \right\} \]
Discernibility matrix: handling incomplete data (missing values)

- **Missing value is a different value**
  \[ a_i \notin M(x, y) \iff x_i = y_i \lor (x_i = ? \land y_i = ?) \]

- **Symmetric similarity**
  \[ a_i \notin M(x, y) \iff x_i = y_i \lor x_i = ? \lor y_i = ? \]

- **Nonsymmetric similarity**
  \[ a_i \notin M(x, y) \iff (x_i = y_i \land y_i \neq ?) \lor x_i = ? \]
Reduct Algorithms

- All Global
- All Local
- One Johnson
- All Johnson
- Partial Global
- Partial Local
All Reducts (Skowron 1993)

- Data Table $\rightarrow$ Discernibility Matrix $\rightarrow$ Prime Implicants $\rightarrow$ Reducts

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>bc</td>
<td>ac</td>
<td></td>
</tr>
<tr>
<td>x2</td>
<td>bc</td>
<td>abc</td>
<td></td>
</tr>
<tr>
<td>x3</td>
<td>abc</td>
<td>b</td>
<td></td>
</tr>
<tr>
<td>x4</td>
<td>ac</td>
<td>b</td>
<td></td>
</tr>
</tbody>
</table>

Prime Implicants
$\Phi \rightarrow \psi$
CNF $\rightarrow$ DNF

- Global reducts

$\left( b \lor c \right) \land \left( a \lor b \lor c \right) \land \left( a \lor c \right) \land \left( b \right) \Rightarrow \left\{ a, b \right\}, \left\{ b, c \right\}$

- Local reducts

$x 1: \left( b \lor c \right) \land \left( a \lor c \right) \Rightarrow \left\{ a, b \right\}, \left\{ c \right\}$

- Advanced algorithm finding prime implicants
Johnson Reduct

- Repeat
  - Find most frequent attribute $a$ in discernibility matrix
  - Remove all fields with $a$ from discernibility matrix
  - Add $a$ to $R$
- until discernibility matrix is empty
Partial Reducts
(Moshkov, Piliszczuk, Zielosko 2008)

\[
\begin{array}{cccc}
\text{a} & \text{b} & \text{c} & \text{dec} \\
1 & 2 & 3 & 1 \\
1 & 3 & 4 & 2 \\
2 & 1 & 1 & 1 \\
2 & 2 & 1 & 2 \\
\end{array}
\]

\[
\begin{array}{cccc}
\text{x1} & \text{x2} & \text{x3} & \text{x4} \\
\hline
\text{x1} & \text{bc} & \text{ac} & \\
\text{x2} & \text{bc} & \text{abc} & \\
\text{x3} & \text{abc} & \text{b} & \\
\text{x4} & \text{ac} & \text{b} & \\
\end{array}
\]

R is an $\alpha$-reduct if:
discerns $\geq (1 - \alpha)$ of non-empty fields of discernibility matrix
none subset of $R$ satisfies the above property

\{b\} is 0.25-reduct but is not 0.2-reduct
\{a,c\} is not 0.25-reduct because \{c\} is 0.25-reduct
## Reduct computation time (sec.)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Attrs</th>
<th>Objects</th>
<th>All global</th>
<th>All local</th>
<th>Global partial</th>
<th>Local partial</th>
</tr>
</thead>
<tbody>
<tr>
<td>segment</td>
<td>19</td>
<td>1540</td>
<td>0.6</td>
<td>0.9</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>chess</td>
<td>36</td>
<td>2131</td>
<td>4.1</td>
<td>66.1</td>
<td>0.2</td>
<td>0.4</td>
</tr>
<tr>
<td>mushroom</td>
<td>22</td>
<td>5416</td>
<td>2.9</td>
<td>4.9</td>
<td>0.8</td>
<td>1.5</td>
</tr>
<tr>
<td>pendigit</td>
<td>16</td>
<td>7494</td>
<td>10.4</td>
<td>23.2</td>
<td>2.2</td>
<td>4.3</td>
</tr>
<tr>
<td>nursery</td>
<td>8</td>
<td>8640</td>
<td>6.5</td>
<td>6.7</td>
<td>1.5</td>
<td>2.8</td>
</tr>
<tr>
<td>letter</td>
<td>16</td>
<td>15000</td>
<td>44.6</td>
<td>179.7</td>
<td>9.7</td>
<td>20.5</td>
</tr>
<tr>
<td>adult</td>
<td>13</td>
<td>30162</td>
<td>62.1</td>
<td>70.1</td>
<td>18.0</td>
<td>33.0</td>
</tr>
<tr>
<td>shuttle</td>
<td>9</td>
<td>43500</td>
<td>91.8</td>
<td>92.5</td>
<td>22.7</td>
<td>48.4</td>
</tr>
<tr>
<td>covtype</td>
<td>12</td>
<td>387342</td>
<td>8591.9</td>
<td>8859.0</td>
<td>903.7</td>
<td>7173.7</td>
</tr>
</tbody>
</table>
Decision rules from global reducts

\[ a_{i_1} = v_1 \land \ldots \land a_{i_p} = v_p \Rightarrow (p_1, \ldots, p_m) \]

\[ p_j = \frac{\left| \{ x \in U : v_1 \land \ldots \land v_p \land dec(x) = d_j \} \right|}{\left| \{ x \in U : v_1 \land \ldots \land v_p \} \right|} \]

\[ Templates(GR) = \left\{ \bigwedge_{a_i \in R} a_i = x_i : R \in GR, x \in U \right\} \]

\[ Rules(GR) = \left\{ t \Rightarrow (p_1, \ldots, p_m) : t \in Templates(GR) \right\} \]

GR – a set of global reducts

U – data set used to compute reducts
Decision rules from local reducts

\[ a_{i_1} = v_{1} \land \ldots \land a_{i_p} = v_{p} \Rightarrow (p_1, \ldots, p_m) \]

\[ p_j = \frac{|\{x \in U : x_{i_1} = v_{1} \land \ldots \land x_{i_p} = v_{p} \land \text{dec}(x) = d_j\}|}{|\{x \in U : x_{i_1} = v_{1} \land \ldots \land x_{i_p} = v_{p}\}|} \]

\[
\text{Templates}(LR) = \left\{ \bigwedge_{a_i \in R} a_i = x_i : R \in LR(x), x \in U \right\}
\]

\[
\text{Rules}(LR) = \left\{ t \mapsto (p_1, \ldots, p_m) : t \in \text{Templates}(LR) \right\}
\]

\[ LR: U \rightarrow P(A) \] – algorithm computing local reducts given an object
\[ U \] – data set used to compute reducts
\[ A \] – a set of attributes describing \( U \)
Rough Set Rule Classifier

- Uses discretization
- Generates reducts and decision rules from reducts
- Classification:

\[
\text{vote}_j(x) = \sum_{t \Rightarrow (p_1, \ldots, p_m) \in \text{Rules} : x \text{ matches } t} p_j \cdot \text{support}(t \Rightarrow (p_1, \ldots, p_m))
\]

\[
\text{dec}_{\text{roughset}}(x) = \max_{d_j \in V_{\text{dec}}} \text{vote}_j(x)
\]
RIONA – Rule Induction with Optimal Neighbourhood Algorithm

- Combines rule induction with k nearest neighbours
- Works like rough set rule classifier using all global reducts but:
  - voting in classification is restricted to k nearest training objects
- Optimizes classification accuracy by searching for the best number k of nearest neighbours
- Performs efficiently by
  - Utilizing the fact that decision support for classification can be calculated without explicit computation of reducts
  - Implementing fast indexing-based nearest neighbours search
  - Restricting decision voting to nearest neighbours
AQ15 algorithm (Michalski 1986)

- Computes decision rules
- Uses $a = v$ and $a \neq v$ descriptors for symbolic attributes
- Uses the $a < v$ descriptor type for numerical attributes without discretization
- Implements covering algorithm, separate for each decision class
- Heuristic search for each rule:
  - from most general to more specific
  - driven by a selected training object
  - candidate rules are extended until they are consistent with the training set, the next rule is selected among final consistent candidate rules
- Classification: voting by the matching rules
Rseslib 3 in Weka

- Official registered package
  - Available in Weka Package Manager
  - requires Weka 3.8.0 or later

- 3 classifiers available now in Weka
  - Rough Set Rule Classifier
  - K Nearest Neighbours / RIONA
  - K Nearest Neighbours with Local Metric Induction
Qmak: graphical interface for Rseslib 3

- Visualization of
  - data
  - classifiers
  - classification
- Classifier modification (interactive)
- Classification of test data
  - shows misclassified objects
- Experiments
  - Cross-validation
  - Multiple cross-validation
  - Multiple random split
- New classifiers including visualization
  - can be added within GUI or in the configuration file
  - do not require changes in Qmak
Rough Sets in Data Mining and Databases: Foundations and Applications

Questions