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
- http://idss.cs.put.poznan.pl/site/rose.html

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

Modlem

- Rule induction using sequential covering algorithm
- Handles numerical attributes without discretization
- Available as Weka package (classification method)
- 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

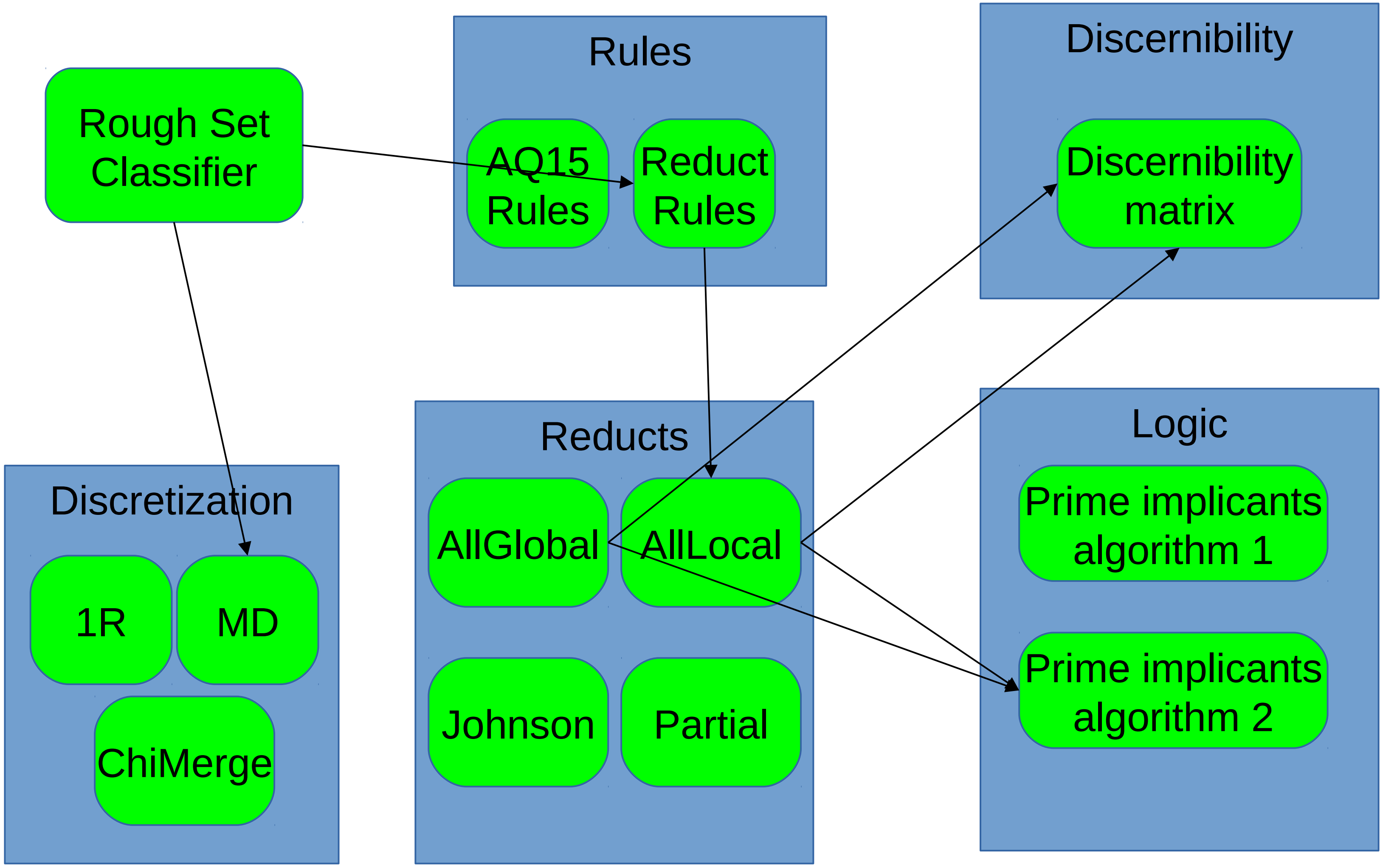
NRough

- 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



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:

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

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 S1 and S2 with the value v maximizing:

$$\left|\left(x,y\right)\in S_1\times S_2 : dec\left(x\right)\neq dec\left(y\right)\right|$$

Discernibility matrix: all pairs

$$M^{all}(x,y) = \{a_i \in A : x_i \neq y_i\}$$

a	b	c	dec
1	2	3	1
1	3	4	2
2	1	1	1
2	2	1	2

	x1	x2	x3	x4
x1		bc	abc	ac
x2	bc		abc	abc
x3	abc	abc		b
x4	ac	abc	b	

Discernibility matrix: pairs with different decisions

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

a	b	c	dec
1	2	3	1
1	3	4	2
2	1	1	1
2	2	1	2

	x1	x2	x3	x4
x1		bc		ac
x2	bc		abc	
x3		abc		b
x4	ac		b	

Discernibility matrix: pairs with different generalized decision

$$M^{gen}(x,y) = \begin{cases} \{a_i \in A : x_i \neq y_i\} & if\ \partial(x) \neq \partial(y) \\ \emptyset & if\ \partial(x) = \partial(y) \end{cases}$$

$$\partial(x) = \{d \in V_{dec} : \exists\, y \in U : \forall\, a_i \in A : x_i = y_i \wedge dec(y) = d\}$$

Discernibility matrix: pairs with different both decisions

$$M^{both}(x,y) = \begin{cases} \{a_i \in A : x_i \neq y_i\} & \text{if } \mathrm{dec}(x) \neq dec(y) \wedge \partial(x) \neq \partial(y) \\ \emptyset & \text{if } \mathrm{dec}(x) = dec(y) \vee \partial(x) = \partial(y) \end{cases}$$

$$\partial(x) = \{d \in V_{dec} : \exists\, y \in U : \forall\, a_i \in A : x_i = y_i \wedge dec(y) = d\}$$

Discernibility matrix: handling incomplete data (missing values)

- Missing value is a different value

$$a_i \notin M(x,y) \Leftrightarrow x_i = y_i \vee (x_i = ? \wedge y_i = ?)$$

- Symmetric similiarity

$$a_i \notin M(x,y) \Leftrightarrow x_i = y_i \vee x_i = ? \vee y_i = ?$$

- Nonsymmetric similarity

$$a_i \notin M(x,y) \Leftrightarrow (x_i = y_i \wedge y_i \neq ?) \vee x_i = ?$$

Reduct Algorithms

- All Global
- All Local
- One Johnson
- All Johnson
- Partial Global
- Partial Local

All Reducts (Skowron 1993)

- Data Table → Discernibility Matrix → Prime Implicants → Reducts

a	b	c	dec
1	2	3	1
1	3	4	2
2	1	1	1
2	2	1	2

	x1	x2	x3	x4
x1		bc		ac
x2	bc		abc	
x3		abc		b
x4	ac		b	

Prime Implicants
Φ → ψ
CNF →DNF

{a, b}
{b, c}

- Global reducts

$$\left(b\vee c\right)\wedge\left(a\vee b\vee c\right)\wedge\left(a\vee c\right)\wedge\left(b\right)\Rightarrow\left\{a,b\right\},\left\{b,c\right\}$$

- Local reducts

$$x1:\left(b\vee c\right)\wedge\left(a\vee c\right)\Rightarrow\left\{a,b\right\},\left\{c\right\}$$

- 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)

a	b	c	dec
1	2	3	1
1	3	4	2
2	1	1	1
2	2	1	2

	x1	x2	x3	x4
x1		bc		ac
x2	bc		abc	
x3		abc		b
x4	ac		b	

R is an α-reduct if:

discerns ≥ (1 – α) 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.)

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

Decision rules from global reducts

$$a_{i_1} = v_1 \wedge \ldots \wedge a_{i_p} = v_p \Rightarrow (p_1, \ldots, p_m)$$

$$p_j = \frac{\left|\left\{x \in U : x_{i_1} = v_1 \wedge \ldots \wedge x_{i_p} = v_p \wedge dec(x) = d_j\right\}\right|}{\left|\left\{x \in U : x_{i_1} = v_1 \wedge \ldots \wedge x_{i_p} = v_p\right\}\right|}$$

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

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

GR – a set of global reducts
U – data set used to compute reducts

Decision rules from local reducts

$$a_{i_1} = v_1 \wedge \ldots \wedge a_{i_p} = v_p \Rightarrow (p_1, \ldots, p_m) \qquad p_j = \frac{\left|\left\{x \in U : x_{i_1} = v_1 \wedge \ldots \wedge x_{i_p} = v_p \wedge dec(x) = d_j\right\}\right|}{\left|\left\{x \in U : x_{i_1} = v_1 \wedge \ldots \wedge x_{i_p} = v_p\right\}\right|}$$

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

$$Rules(LR) = \left\{ t \Rightarrow (p_1, \ldots, p_m) : t \in Templates(LR) \right\}$$

LR:U–>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:

$$vote_j(x) = \sum_{t \Rightarrow (p_1, \ldots, p_m) \in Rules : x\, matches\, t} p_j \cdot support(t \Rightarrow (p_1, \ldots, p_m))$$

$$dec_{roughset}(x) = \max_{d_j \in V_{dec}} vote_j(x)$$

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 ≠ 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 consistenst 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