RSESLIB 3: Rough Set and Machine Learning Open Source in Java

Agenda

- Overview
- Library contents
- Modular architecture
- Tools for Rseslib 3
- Projects using Rseslib 3
- Contributors

Rseslib 3: Motivation

- Deliver library of rough set methods in Java
  - Open source
  - Easily extensible
  - Easily modifiable
- Speed-up research & development of new machine learning algorithms
  - Reduce development effort
  - Additive implementation
    - Increase reusability of code
    - Increase inheritance of available algorithms
  - Code organization
- Speed-up experiments
  - Multi-platform executables – Java
  - Grid Computing / Network of Workstations
- Didactic framework
  - Research of new algorithms
  - Applications

Rseslib 3: Overview

- 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
- Parallel / distributed experiments

Library Content

- Transformation
- **Discretization**
- Missing value completion
- Filtering
- Sampling
- Clustering
- Sorting
- **Discernibility matrix computation**
- **Reduct calculation**
- **Rule induction**
- Metric induction
- Principal Component Analysis (PCA)
- Boolean reasoning
- Genetic algorithm scheme
- **Classification** and classifier evaluation

Data formats

- ARFF (Weka)
- CSV + Rseslib header
  - header file apart
  - header and data in one file
- RSES 2.x

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

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

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

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

- Local reducts

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

- 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
- Remove redundant attributes from R

Partial Reducts (H.S. Nguyen, D. Ślęzak 1999)

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

Rule induction algorithms

- From global reducts
- From local reducts
- AQ15

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

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

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

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

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

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*

AQ15 rule induction algorithm (Michalski at al. 1986)

- 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: Unique Implementations

- Rough Set Rule Classifier
- K Nearest Neighbors / RIONA
- K Nearest Neighbors with Local Metric Induction

Classification: Classics

- Decision tree C4.5 (Quinlan)
- Rule Classifier AQ15 (Michalski et al)
- Neural Network
- Naive Bayes
- Support Vector Machine
- PCA classifier
- Local PCA classifier
- Metaclassifiers
  - Bagging
  - AdaBoost

Rough Set Rule Classifier

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

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

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

K Nearest Neighbors

- Metrics working for data with both numerical and symbolic attributes
- Weighting attributes in metrics
- Fast indexing-based nearest neighbors search
- Number k of nearest neighbors optimized automatically
- Distance-dependent voting for decision by nearest neighbors
- Mode to work as RIONA algorithm
- For details and experimental evaluation see:
  - Wojna A., Analogy-Based Reasoning in Classifier Construction (phd thesis)

RIONA – Rule Induction with Optimal Neighborhood Algorithm (Góra, Wojna)

- Combines rule induction with k nearest neighbors
  - Only neighbors matching any consistent decision rule covering the classified object vote for decision
- Performs efficiently by
  - Utilizing the fact that decision support for classification can be calculated without explicit computation of rules
  - Restricting decision voting to nearest neighbors

K Nearest Neighbors with Local Metric Induction (Skowron, Wojna, 2004)

- Selects a large set S of nearest neighbors using global metric M
- Uses S to induce local metric M(S)
- Selects the decision using k nearest neighbors from the set S with respect to the local metric M(S)

Other algorithms

- **Transformations:** missing value completion (non-invasive data imputation by Gediga and Duentsch), attribute selection, numerical attribute scaling, new attributes (radial, linear and arithmetic transformations)
- **Filtering:** missing values filter, Wilson's editing, Minimal Consistent Subset (MSC) by Dasarathy, universal boolean function based filter
- **Sampling:** with repetitions, without repetitions, with given class distribution
- **Clustering:** k approximate centers algorithm
- **Sorting:** attribute value related, distance related
- **Metric induction:** Hamming and Value Difference Metric (VDM) for nominal attributes, city-block Manhattan, Interpolated Value Difference Metric (IVDM) and Density-Based Value Difference Metric (DBVDM) for numerical attributes, attribute weighting (distance-based, accuracy-based, perceptron)
- **Principal Component Analysis (PCA):** OjaRLS algorithm
- **Boolean reasoning:** two different algorithms generating prime implicant from CNF boolean formula
- **Genetic algorithm scheme:** user provides cross-over operation, mutation operation and fitness function only
- **Classifier evaluation:** single train-and-classify test, cross-validation, multiple test with random train-and-classify split, multiple cross-validation (all types of tests can be executed on many classifiers)

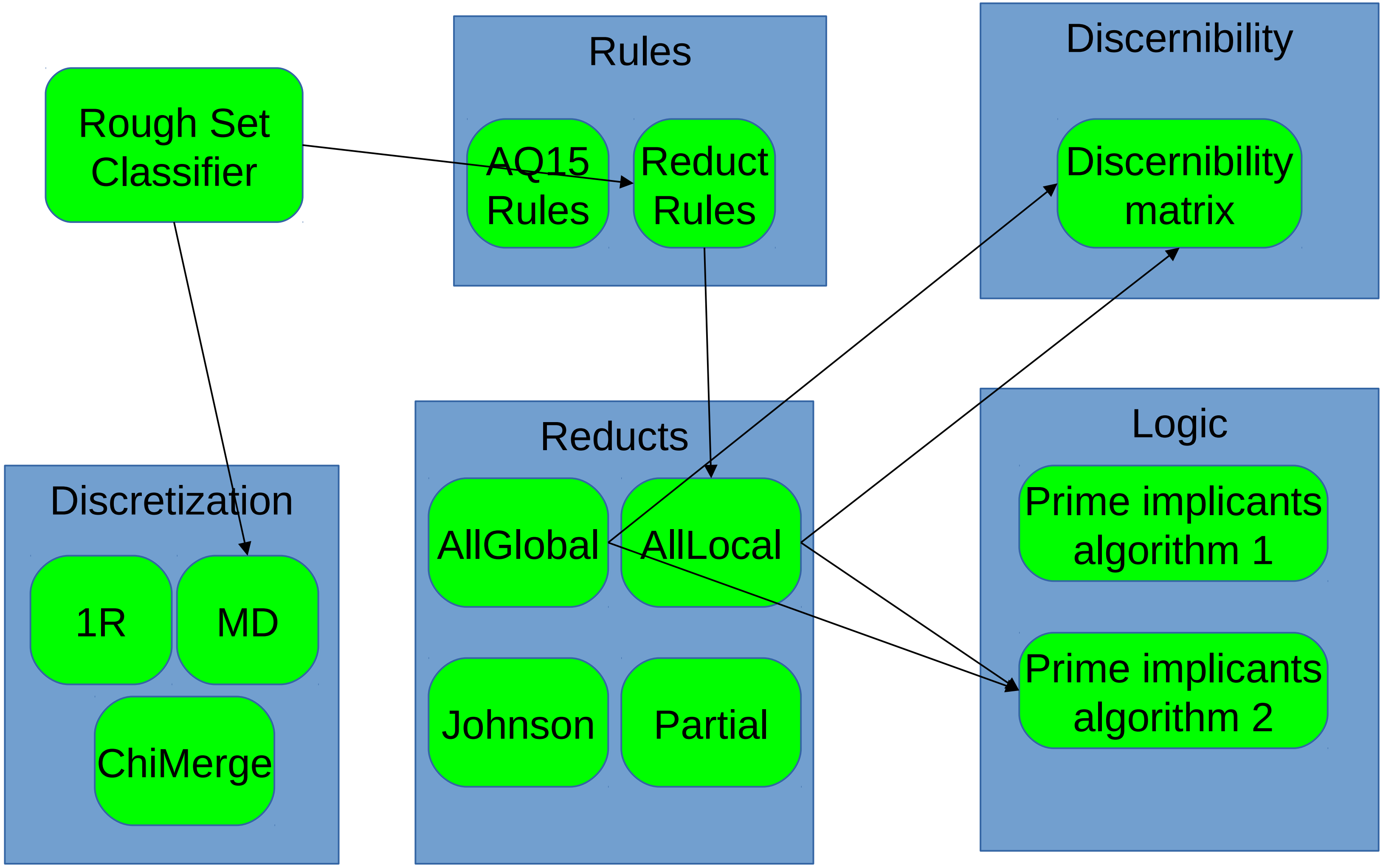
Modularity

- Modules
- Interfaces
- Isolated elementary mathematical objects
- Isolated processing algorithms

Modularity: mathematical objects

- **Basic:** attribute, data header, data object, boolean data object, numbered data object, data table, nominal attribute histogram, numeric attribute histogram, decision distribution
- **Boolean functions/operators:** attribute equality, attribute interval, attribute value subset, binary discrimination, metric cube, negation, conjunction, disjunction
- **Real functions/operators:** scaler, perceptron, radius, multiplication, addition
- **Integer functions:** discrimination (discretization, 3-value cut)
- **Decision distribution functions:** nominal to dec distr, numeric to vicinity-based dec distr, numeric to interpolated dec distr
- **Vector space:** vector, linear subspace, PCA subspace, vector function
- **Linear order**
- **Indiscernibility relations**
- **Rules:** universal boolean function rule, equality descriptors rule, partial matching rule
- **Reducts**
- **Metrics:** City + Hamming, City + VDM, IVDM, DBVDM, metric-based indexing tree
- **Probability:** guassian kernel function, hypercube kernel function, m-estimate

Modularity: example



Modularity: examples

- Attribute weighting in metric
  - Perceptron as one of weighting methods
- Estimate of value probability at given decision
  - Probability defined by k nearest neighbours

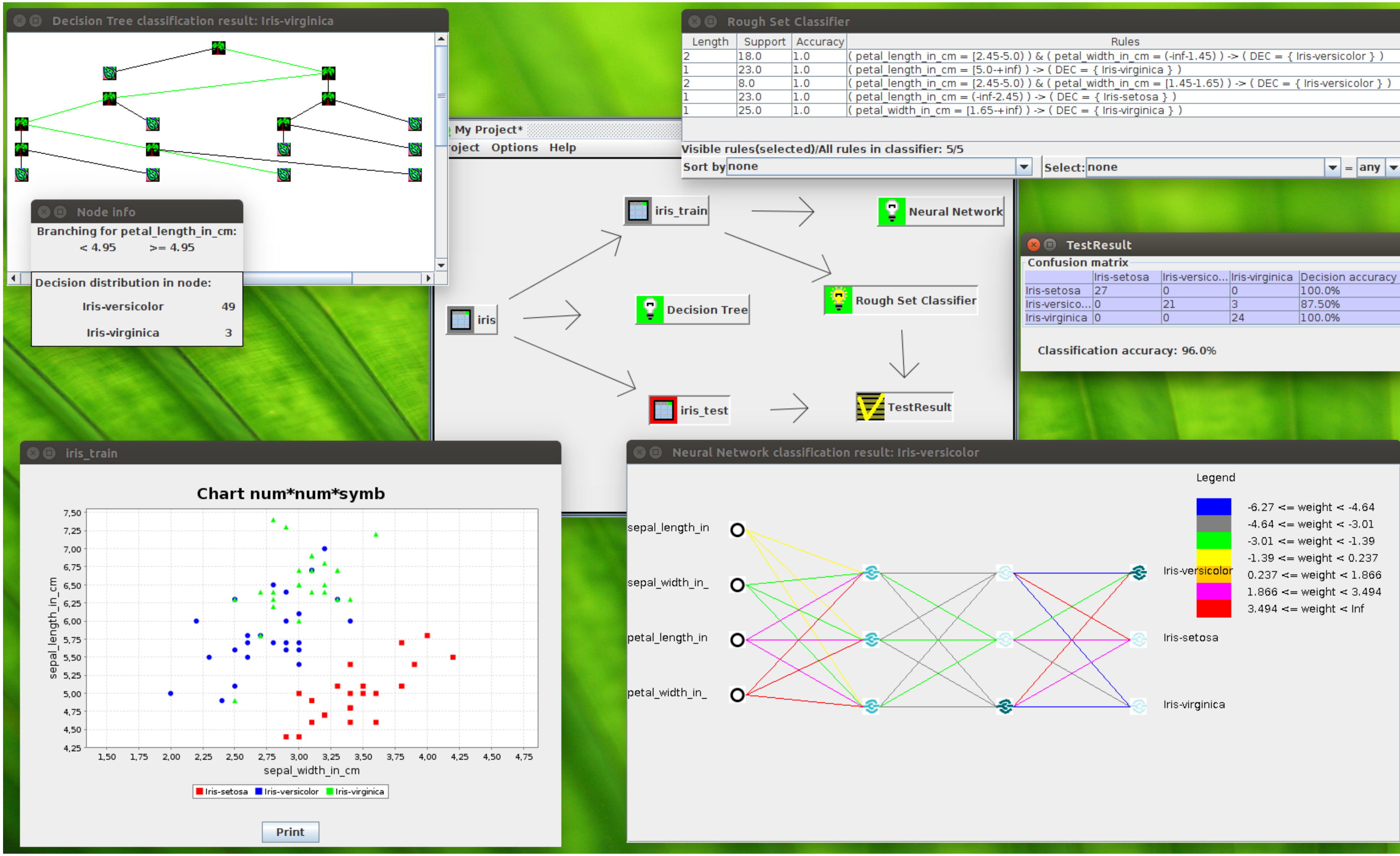
Tools for Rseslib 3

- Weka
- QMAK
- Simple Grid Manager

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 Neighbors / RIONA
  - K Nearest Neighbors with Local Metric Induction

QMAK: interacting with and visualizing classifiers

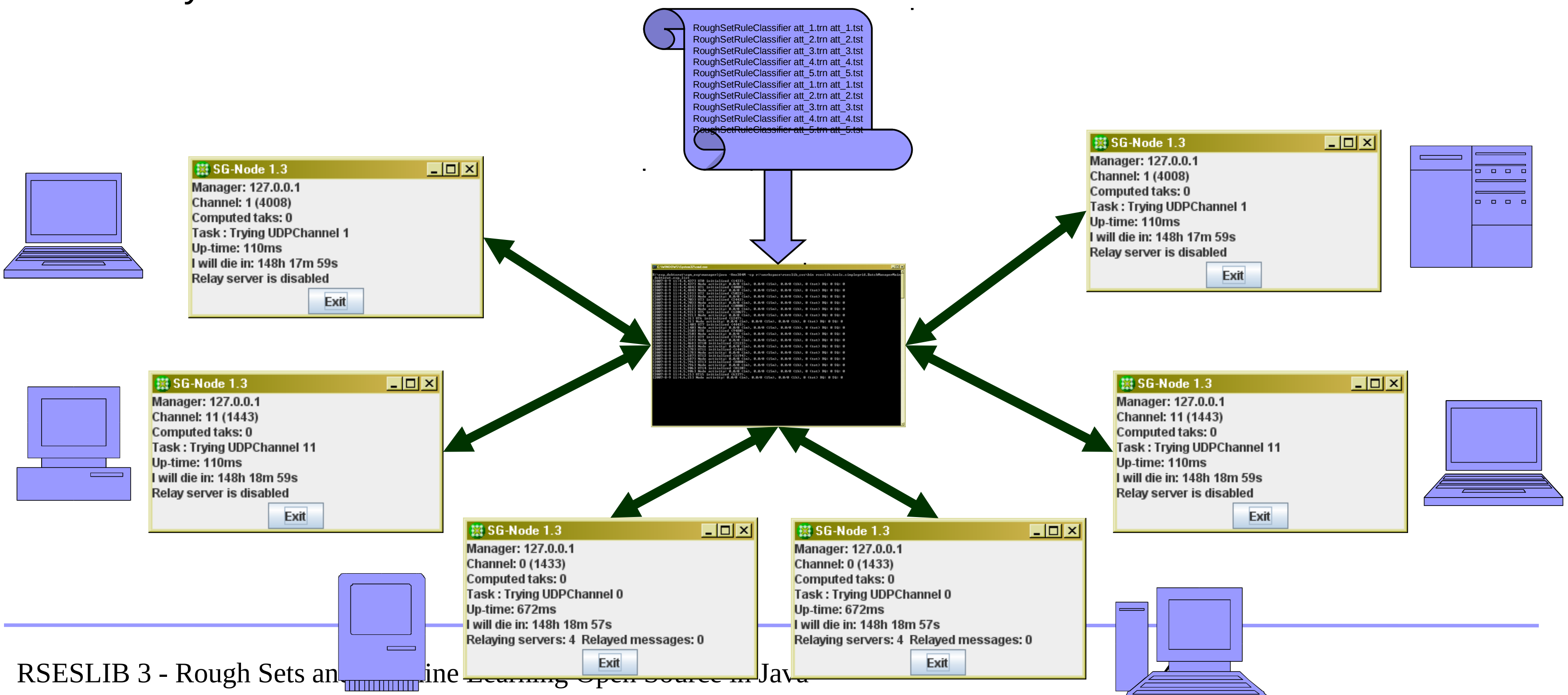


QMAK functionality

- Visualization of
  - data
  - classifiers
  - single object classification
- Interactive classifier modification
- Presentation of misclassified objects
- Comparing classifiers with tests
  - multiple cross-validation
  - multiple random split
- Classifiers with visualization implemented by users
  - can be added using menu or in the configuration file
  - do not require changes in Qmak
- Watch 5-minute demo of Qmak:
  - http://rseslib.mimuw.edu.pl/qmak

Simple Grid Manager

- Train-and-test experiments with Rseslib classifiers
- Ad-hoc cluster creation
- Resuming failed jobs
- Skipping completed jobs in case of restart
- Robust communication: working in non-reliable networks
- Many clients on one machine utilizes multicore CPU



Projects using Rseslib 3

- TunedIT
  - system for automated evaluation, benchmarking and comparison of data mining and machine learning algorithms
- Debellor
  - framework for scalable data mining and machine learning with data streaming
- Mahout-extensions
  - attribute selection extensions to Mahout
- DMEXL
  - data mining expression library facilitating development of concurrent data mining algorithms

Contributors

- Library
  Jan Bazan, Rafał Falkowski, Grzegorz Góra, Wiktor Gromniak, Marcin Jałmużna, Łukasz Kosson, Łukasz Kowalski, Michał Kurzydłowski, Rafał Latkowski, Łukasz Ligowski, Michał Mikołajczyk, Krzysztof Niemkiewicz, Dariusz Ogórek, Marcin Piliszczuk, Maciej Próchniak, Jakub Sakowicz, Sebastian Stawicki, Cezary Tkaczyk, Arkadiusz Wojna, Witold Wojtyra, Damian Wójcik, Beata Zielosko
- Graphical interface Qmak
  Katarzyna Jachim, Damian Mański, Michał Mański, Krzysztof Mroczek, Robert Piszczatowski, Maciej Próchniak, Tomasz Romańczuk, Piotr Skibiński, Marcin Staszczyk, Michał Szostakiewicz, Leszek Tur, Arkadiusz Wojna, Damian Wójcik, Maciej Zuchniak
- Simple Grid Manager
  Rafał Latkowski

Summary

- Ready to use Open Source Java Library
- Broad collection of Rough Set & Machine Learning algorithms
- Easy to use & implement own algorithms
- Mailing list
  - rseslib-users@googlegroups.com
- Visit the website:
  - http://rseslib.mimuw.edu.pl