D2.1.3 The ROSETTA Rough Set Software System

Jan Komorowski[1,3], *Aleksander Øhrn*[1], *Andrzej Skowron*[2]

[1] Department of Computer and Information Science
Norwegian University of Science and Technology (NTNU)
7491 Trondheim, Norway
{janko, aleks}@idi.ntnu.no
[2] Institute of Mathematics, Warsaw University
02-097 Warsaw, Poland
skowron@mimuw.edu.pl
[3] Polish-Japanese Institute of Information Technology
Koszykowa 86, 02-008 Warsaw, Poland

Abstract

Research in rough sets [15,16] has resulted in a number of software tools for data mining and knowledge discovery from databases (KDD). Among many of these tools the ROSETTA system [27–29] is probably one of the most complete software environment for rough set operations.

In ROSETTA, the experimental nature of inducing classifiers from data is explicitly maintained by organizing the workspace in a tree structure that displays how input and output data relate to each other. ROSETTA supports the overall KDD process: From browsing and preprocessing of the data, to reduct computation and rule synthesis, to validation and analysis of the generated rules. Learning may be both supervised (resulting in if-then rules) or unsupervised (resulting in general patterns), and input data may be categorical, numerical or both. ROSETTA is not tied up to any particular application domain, and has been put to use for a variety of tasks.

ROSETTA is a cooperative effort between researchers at NTNU in Norway and Warsaw University in Poland, and is available on the World Wide Web [20]. The system runs under Windows NT/98/95/2000.

D2.1.3.1 Introduction

Rough sets offer an interesting and successful approach to data mining and knowledge discovery. Since the original work of Pawlak [15, 16] there has been a systematic growth in theoretical and applied research that relies on this formalism. The theoretical developments have been accompanied by various implementations of rough set tools. There exist today several experimental research systems that support different aspects of developing rough set models. Many prosperous applications using those systems have been reported in the literature.

This note starts with a list of all rough set software systems that are known to us. We then describe the ROSETTA system. It is assumed that the reader is familiar with basic notions of rough set theory [17] and Boolean reasoning [3]. A tutorial introduction to rough sets can be found in Section B6 of this Handbook.

D2.1.3.2 Software Systems for Rough Sets

The authors are aware of the following software systems for rough sets. The reader can find more details about these systems in [18,19], or by contacting the respective authors directly.

- Datalogic/R
  http://www.reduct.com/
- Grobian (Roughian)
  I.Duentsch@ulst.ac.uk, ggediga@luce.psycho.Uni-Osnabrueck.de
- KDD-R
  ziarko@cs.uregina.ca
- LERS
  jerzy@eecs.ukans.edu
- PRIMEROSE
  tsumoto@computer.org
- ProbRough
  {zpiasta, lenarcik}@sabat.tu.kielce.pl
- ROSETTA
  http://www.idi.ntnu.no/~aleks/rosetta/
- Rough Family
  Roman.Slowinski@cs.put.poznan.pl, Jerzy.Stefanowski@cs.put.poznan.pl
- RSDM
  {cfbaizan, emenasalvas}@fi.upm.es
- RoughFuzzyLab
  rswiniar@saturn.sdsu.edu
- RSL
  ftp://ftp.ii.pw.edu.pl/pub/Rough/
- TAS
  zsuraj@univ.rzeszow.pl
- Trance
  wojtek@cs.vu.nl

D2.1.3.3 Background Information

The ROSETTA system [27–29] has been developed by two groups: Knowledge Systems Group at NTNU, Norway [8], and Group of Logic, Warsaw University, Poland [6] under the guidance of, respectively, Jan Komorowski and Andrzej Skowron.

In Norway, the main design and programming effort of the graphical user interface (GUI) and the kernel architecture has been undertaken by Aleksander Øhrn. Several other people contributed, including (in alphabetical order) Merete Hvalshagen, Jørn Nygjerd, Daniel Remmem, Knut Magne Risvik, Ivan Uthus, Staal Vinterbo and Thomas Ågotnes.

In Poland, the original work on a library of rough set algorithms, the RSES library, has been done (in alphabetical order) by Jan Bazan, Agnieszka Chądzyńska, Adam Cykier, Sinh Hoa Nguyen, Son Hung Nguyen, Piotr Synak, Marcin Szczuka, Dominik Ślęzak and Jakub Wróblewski.

A public version of ROSETTA is available [20], and a reference manual containing a detailed description of the features of the current system can be found at the same location. Readers of the manual should consult [27] for an overview of the background theory and references therein.

ROSETTA runs on 32-bit Windows platforms, both with a GUI and as a command-line utility. It is simple to port ROSETTA to other platforms, albeit not with the GUI. A screenshot of the ROSETTA GUI can be found in Figure 1.

Although the system will run on small configurations, any serious application will benefit from large RAM and fast CPU. Also, since ROSETTA keeps its working data in main memory, machines with low memory may experience degraded performance on very large datasets. For reference, the widest decision tables that we have processed had around 2500 attributes, and the largest number of objects was about 15000.

D2.1.3.4 Design Issues

The ROSETTA computational kernel is a general C++ class library, and offers an advantageous code base for researchers to quickly assemble and try out new algorithms and ideas [21]. Extensibility and flexibility have been chief parameters in its design and construction. By applying suitable object-oriented design patterns, the kernel is maintainable and highly modular, and completely independent of the GUI.

However, a collection of clever algorithms does not in itself make a complete system. In order to comprise a fully usable KDD tool, the algorithms need to be set in an environment such that models can be developed in an interactive manner, data items are organized, and experiments may be conducted in a setting that allows backtracking. These considerations were all made explicit when designing the ROSETTA GUI.

D2.1.3.5 Input Data and Background Knowledge

Basic input to ROSETTA are flat data tables describing Pawlak information systems. An information system is a single table, either physically or as a logical view across several underlying tables. Several tables can be present in the system simultaneously. Tabular data may be imported from a wide range of data sources by means of the Open Database Connectivity (ODBC) interface. Possible data sources include spreadsheets, relational database management systems or plain ASCII files, depending on which ODBC drivers the user has installed on his/her system. Data dictionaries are automatically constructed from the imported data, but hand-crafted dictionaries can also be imported. Dictionaries contain meta-data such as attribute names, types, measurement units, coding schemes, etc. The user may manually control how raw data is preprocessed, and embed background knowledge in that way. For instance, in a medical setting, known medically relevant cut-off values may be specified during discretization of real-valued attributes.

ROSETTA also has the option of employing user-defined notions of discernibility on a per attribute basis, e.g., in the reduct computation process. If the domain of an attribute defines a partial order, this knowledge may be input to the system and appropriate discernibility considerations will be made. This feature also enables the user to specify how missing values are to be treated, since missing values may indicate different things for different attributes.

If cost information is available, some of the heuristics for reduct computation can be equipped with a bias for computing "inexpensive" attribute subsets.

D2.1.3.6 System Output

The main output from ROSETTA that may convey discovered knowledge consists of if-then classification rules, or minimal object descriptions or general patterns, along with several numerical factors associated with these. Such induced rules or patterns may be either as specific as possible, or approximate up to a degree specified by the user. These structures may subsequently be put to use inside the ROSETTA system itself, or exported and applied elsewhere.

Most structures in use by the ROSETTA system such as decision tables, reduct sets, rule sets, indiscernibility graphs and discernibility functions can be exported to ASCII files. This opens up a link to other systems that support manipulation and visualization of structures in ways not currently supported by ROSETTA. For example, tables can be exported to MATLAB as matrices, and rule sets and tables can be exported to Prolog as sets of clauses and facts. Rule sets can also be exported as C++ source code.

Optional output from computational processes include detailed log files containing information about performance and/or costs. Such files are often directly plottable by external programs.

D2.1.3.7 Supported Data Mining Tasks

Data mining is understood as a component of the overall KDD process, and is in the rough set framework defined to be the computation of reducts and their postprocessing, as well as the synthesis of rules (from the reducts) and their postprocessing.

ROSETTA supports computation of both proper and approximate reducts [2,17,22], and reducts may be relative to an object or relative to a full table. Furthermore, reducts may be relative to a decision attribute or not. Since computing minimal reducts is NP-hard, efficient heuristics such as, e.g., genetic algorithms for searching for individual reducts are included [25,26].

As mentioned, the reducts computed by ROSETTA do not necessarily have to be reducts in the strict sense, but can, if specified by the user, be approximations of reducts. Reduct approximations [2,22] are generally believed to be more tolerant to noise and other data impurities. Computation of dynamic reducts [2] is an example of an approximation option that is implemented in ROSETTA.

Postprocessing of reduct and rule sets typically amounts to filtering operations. Such collections of structures may be filtered down to more manageable sizes according to a wide variety of criteria.

D2.1.3.8 Support for Task and Method

With respect to supervised learning, ROSETTA can be used to induce classification rules. However, inducing a classifier from raw data is usually composed of several steps that effectively define a pipeline. For instance, discretization may have to be performed before reducts are computed, before rules in turn are generated from the reducts. At each of these steps, choices may have to be made as to which of several candidate algorithms to apply. And different choices at each step define alternative branchings of the induction pipeline. ROSETTA offers a variety of algorithm alternatives at each step, easily selectable through pop-up menus. Furthermore, ROSETTA offers a GUI environment in which the resulting structures in the pipeline branchings can be organized.

D2.1.3.9 Support of the KDD process

KDD is understood to constitute the full process from initial target data selection and preprocessing, to validation and interpretation of the induced models [4]. ROSETTA supports the entire KDD process both in terms of an appropriately designed GUI, and in its offering of relevant algorithms.

The ROSETTA GUI is fully object-oriented in the sense that all structural instances are represented as individual icons, each with their own set of operations that can be performed on them. The icons are organized in a project tree where the interrelationship between the icons in the tree immediately reflect how they relate to each other. This aids in data navigation. As a data analysis project is carried out, the tree gets automatically updated and annotated. Annotations help automate the process of generating session logs and project documentation.

Initial browsing and target data selection is done in intuitive, object-oriented grid environments, using data dictionaries to allow communication with the user in terms from the modelling domain. Attributes and objects in a table can be masked out or removed altogether before the table is passed on in the KDD pipeline.

Currently, completion and discretization are the main two preprocessing options offered by ROSETTA. Completion consists of heuristics for eliminating missing values, while discretization consists of converting real-valued attributes into interval-attributes. Discretization problems and symbolic value partition problems are of high computational complexity, i.e., NP-complete or NP-hard. The Warsaw group has made significant contributions to the development of practical heuristics for these problems, e.g., [10–14]. Several heuristics for completion and discretization are implemented in ROSETTA.

Support for the data mining component of KDD is described elsewhere in this document. Rules produced by this step may be inspected and interpreted in grid environments using terms from the modelling domain, and may also be applied to new cases in order to assess their predictive capabilities. ROSETTA offers support for cross-validation as well as for simple train-test splits, and allows quantities derived from confusion matrices and ROC curves [7] to be used as performance measures. A handful statistical tests for hypothesis testing are also implemented.

Although the ROSETTA GUI offers an intuitive environment for interactive model construction, the system also has support for automating the steps in the KDD pipeline. ROSETTA is equipped with a simple scripting language that enables structural objects to "flow" through a sequence of algorithms. Long KDD pipelines can thus easily be assembled and executed without the need for any intervention.

D2.1.3.10 Visualization

ROSETTA does not currently offer any means of visualizing data or models graphically. Instead, there is a set of export routines so that the user can do this outside of the system. For example, the indiscernibility relation can be exported to a format recognized by the GraphViz suite of graph visualization programs [5]. Such graphs can be used for clustering and unsupervised learning.

D2.1.3.11 Main Applications

ROSETTA is a general purpose tool for data mining and KDD within the rough set framework, and is not geared towards any particular application area. Hence, any area in which the fundamental task of constructing a mapping from one domain to another (or forming a minimal description of a domain) comes up, is a potential candidate application area.

Users having downloaded the system report using ROSETTA in fields such diverse as power electronics, analysis of medical data, satellite control, software engineering, finance, public policy generation, medical ethics, history of science, real-time decision-making, anthropology, selection of controller gains, environmental modelling and diagnosis of rotating machinery. For detailed references, see [27].

Acknowledgments

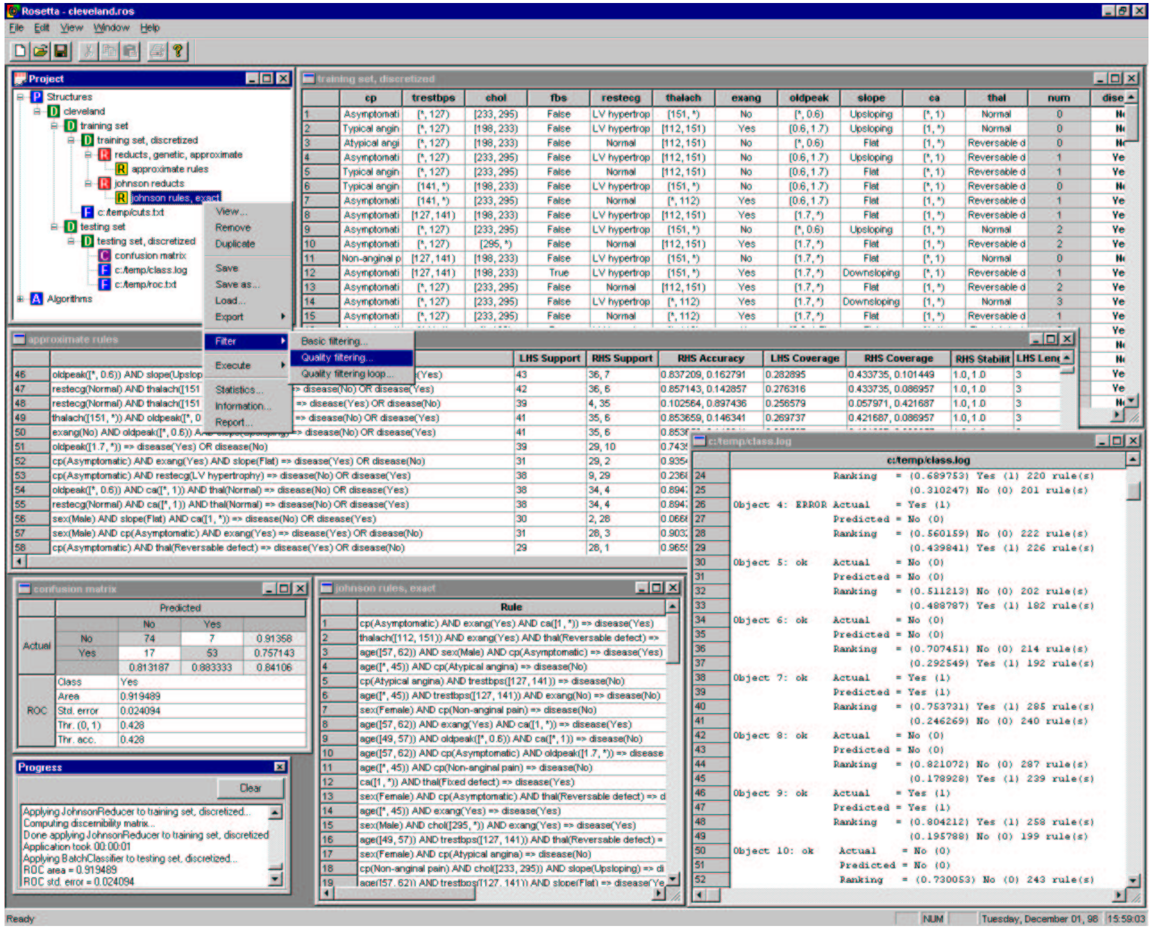
Numerous people have been involved in the ROSETTA project. We can only express our general thanks to all colleagues, collaborators, reviewers, students, friends, wives and funding agencies who have contributed to ROSETTA in very many direct and indirect ways.

References

1. A. Aamodt and J. Komorowski, editors. *Proc. Fifth Scandinavian Conference on Artificial Intelligence*, Trondheim, Norway. Volume 28 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 1995.
2. J. G. Bazan. A comparison of dynamic and non-dynamic rough set methods for extracting laws from decision tables. In Polkowski and Skowron [18], pp. 321–365.
3. F. M. Brown. *Boolean Reasoning: The Logic of Boolean Equations.* Kluwer Academic Publishers, 1990. ISBN 0-7923-9121-7.
4. U. Fayyad, G. Piatetsky-Shapiro and P. Smyth. The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11):27–34, 1996.
5. The GraphViz homepage http://www.research.att.com/sw/tools/graphviz. AT&T Research.
6. The Group of Logic homepage. http://alfa.mimuw.edu.pl/logic/.
7. J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143:29–36, 1982.
8. The Knowledge Systems Group homepage. http://www.idi.ntnu.no/grupper/KS-grp/.
9. J. Komorowski and J. Zytkow, editors. *Proc. First European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'97)*, Trondheim, Norway. Volume 1263 of *Lecture Notes in Artificial Intelligence*, Springer Verlag, 1997.
10. H. S. Nguyen. *Discretization of Real Valued Attributes – A Boolean Reasoning Approach.* PhD thesis, Warsaw University, 1997.
11. S. H. Nguyen and A. Skowron. Quantization of real valued attributes. In Wang [23], pp. 34–37.
12. H. S. Nguyen. From optimal hyperplanes to optimal decision trees. *Fundamenta Informaticae*, 34:145–174, 1998.
13. H. S. Nguyen and S. H. Nguyen. Discretization methods in data mining. In Polkowski and Skowron [18], pp. 451–482.
14. S. H. Nguyen and A. Skowron. Searching for relational patterns in data. In Komorowski and Zytkow [9], pp. 265–276.
15. Z. Pawlak. Information systems – Theoretical foundations. *Information Systems*, 6:205–218, 1981.
16. Z. Pawlak. Rough sets. *International Journal of Computer and Information Sciences*, 11(5):341–356, 1982.
17. Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data.* Kluwer Academic Publishers, 1991. ISBN 0-7923-1472-7.
18. L. Polkowski and A. Skowron, editors. *Rough Sets in Knowledge Discovery 1: Methodology and Applications.* Physica-Verlag, 1998. ISBN 3-7908-1119-X.
19. L. Polkowski and A. Skowron, editors. *Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems.* Physica-Verlag, 1998. ISBN 3-7908-1119-X.
20. The ROSETTA homepage. http://www.idi.ntnu.no/~aleks/rosetta/.
21. The ROSETTA C++ Library homepage. http://www.idi.ntnu.no/~aleks/thesis/source/.
22. A. Skowron. Synthesis of adaptive decision systems from experimental data. In Aamodt and Komorowski [1], pp. 220–238.
23. P. P. Wang, editor. *Proc. Second Joint Annual Conference on Information Sciences*, Wrightsville Beach, North Carolina, USA, 1995.

24. P. P. Wang, editor. *Proc. Third Joint Annual Conference on Information Sciences*, Durham, North Carolina, USA, 1997.
25. J. Wróblewski. Finding minimal reducts using genetic algorithms. In Wang [23], pp. 186–189.
26. J. Wróblewski. Genetic algorithms in decomposition and classification problems. In Polkowski and Skowron [19], pp. 472–492.
27. A. Øhrn. *Discernibility and Rough Sets in Medicine: Tools and Applications.* PhD thesis, Norwegian University of Science and Technology, December 1999. ISBN 82-7984-014-1. http://www.idi.ntnu.no/~aleks/thesis/.
28. A. Øhrn and J. Komorowski. ROSETTA: A rough set toolkit for analysis of data. In Wang [24], pp. 403–407.
29. A. Øhrn, J. Komorowski, A. Skowron and P. Synak. The design and implementation of a knowledge discovery toolkit based on rough sets: The ROSETTA system. In Polkowski and Skowron [18], pp. 376–399.

This article was processed using the LaTeX macro package with LMAMULT style



	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num	dise
1	Asymptomati	[*, 127)	[233, 295)	False	LV hypertrop	[151, *)	No	[*, 0.6)	Upsloping	[*, 1)	Normal	0	N
2	Typical angin	[*, 127)	[198, 233)	False	LV hypertrop	[112, 151)	Yes	[0.6, 1.7)	Upsloping	[1, *)	Normal	0	N
3	Atypical angi	[*, 127)	[198, 233)	False	Normal	[112, 151)	No	[*, 0.6)	Flat	[1, *)	Reversable d	0	N
4	Asymptomati	[*, 127)	[233, 295)	False	LV hypertrop	[112, 151)	No	[0.6, 1.7)	Upsloping	[*, 1)	Reversable d	1	Ye
5	Typical angin	[*, 127)	[233, 295)	False	Normal	[112, 151)	No	[0.6, 1.7)	Flat	[*, 1)	Reversable d	1	Ye
6	Typical angin	[141, *)	[198, 233)	False	LV hypertrop	[151, *)	No	[0.6, 1.7)	Flat	[*, 1)	Reversable d	0	N
7	Asymptomati	[141, *)	[233, 295)	False	Normal	[*, 112)	Yes	[0.6, 1.7)	Flat	[1, *)	Reversable d	1	Ye
8	Asymptomati	[127, 141)	[198, 233)	False	LV hypertrop	[112, 151)	Yes	[1.7, *)	Flat	[1, *)	Reversable d	1	Ye
9	Asymptomati	[*, 127)	[233, 295)	False	LV hypertrop	[151, *)	No	[*, 0.6)	Upsloping	[1, *)	Normal	2	Ye
10	Asymptomati	[*, 127)	[295, *)	False	Normal	[112, 151)	Yes	[1.7, *)	Flat	[1, *)	Reversable d	2	Ye
11	Non-anginal p	[127, 141)	[198, 233)	False	LV hypertrop	[151, *)	No	[1.7, *)	Flat	[*, 1)	Normal	0	N
12	Asymptomati	[127, 141)	[198, 233)	True	LV hypertrop	[151, *)	Yes	[1.7, *)	Downsloping	[*, 1)	Reversable d	1	Ye
13	Asymptomati	[*, 127)	[233, 295)	False	Normal	[112, 151)	Yes	[1.7, *)	Flat	[1, *)	Reversable d	2	Ye
14	Asymptomati	[*, 127)	[233, 295)	False	LV hypertrop	[*, 112)	Yes	[1.7, *)	Downsloping	[1, *)	Normal	3	Ye
15	Asymptomati	[*, 127)	[233, 295)	False	Normal	[*, 112)	Yes	[1.7, *)	Flat	[1, *)	Reversable d	1	Ye

		LHS Support	RHS Support	RHS Accuracy	LHS Coverage	RHS Coverage	RHS Stabilit	LHS Leng
46	oldpeak([*, 0.6)) AND slope(Upslop (Yes)	43	36, 7	0.837209, 0.162791	0.282895	0.433735, 0.101449	1.0, 1.0	3
47	restecg(Normal) AND thalach([151 => disease(No) OR disease(Yes)	42	36, 6	0.857143, 0.142857	0.276316	0.433735, 0.086957	1.0, 1.0	3
48	restecg(Normal) AND thalach([151 => disease(Yes) OR disease(No)	39	4, 35	0.102564, 0.897436	0.256579	0.057971, 0.421687	1.0, 1.0	3
49	thalach([151, *)) AND oldpeak([*, 0 => disease(No) OR disease(Yes)	41	35, 6	0.853659, 0.146341	0.269737	0.421687, 0.086957	1.0, 1.0	3
50	exang(No) AND oldpeak([*, 0.6)) A => disease(No) OR disease(Yes)	41	35, 6	0.853				
51	oldpeak([1.7, *)) => disease(Yes) OR disease(No)	39	29, 10	0.743				
52	cp(Asymptomatic) AND exang(Yes) AND slope(Flat) => disease(Yes) OR disease(No)	31	29, 2	0.935				
53	cp(Asymptomatic) AND restecg(LV hypertrophy) => disease(No) OR disease(Yes)	38	9, 29	0.236				
54	oldpeak([*, 0.6)) AND ca([*, 1)) AND thal(Normal) => disease(No) OR disease(Yes)	38	34, 4	0.894				
55	restecg(Normal) AND ca([*, 1)) AND thal(Normal) => disease(No) OR disease(Yes)	38	34, 4	0.894				
56	sex(Male) AND slope(Flat) AND ca([1, *)) => disease(No) OR disease(Yes)	30	2, 28	0.066				
57	sex(Male) AND cp(Asymptomatic) AND exang(Yes) => disease(Yes) OR disease(No)	31	28, 3	0.903				
58	cp(Asymptomatic) AND thal(Reversable defect) => disease(Yes) OR disease(No)	29	28, 1	0.965				

		Predicted		
		No	Yes	
Actual	No	74	7	0.91358
	Yes	17	53	0.757143
		0.813187	0.883333	0.84106
ROC	Class	Yes		
	Area	0.919489		
	Std. error	0.024094		
	Thr. (0, 1)	0.428		
	Thr. acc.	0.428		

	Rule
1	cp(Asymptomatic) AND exang(Yes) AND ca([1, *)) => disease(Yes)
2	thalach([112, 151)) AND exang(Yes) AND thal(Reversable defect) =>
3	age([57, 62)) AND sex(Male) AND cp(Asymptomatic) => disease(Yes)
4	age([*, 45)) AND cp(Atypical angina) => disease(No)
5	cp(Atypical angina) AND trestbps([127, 141)) => disease(No)
6	age([*, 45)) AND trestbps([127, 141)) AND exang(No) => disease(No)
7	sex(Female) AND cp(Non-anginal pain) => disease(No)
8	age([57, 62)) AND exang(Yes) AND ca([1, *)) => disease(Yes)
9	age([49, 57)) AND oldpeak([*, 0.6)) AND ca([*, 1)) => disease(No)
10	age([57, 62)) AND cp(Asymptomatic) AND oldpeak([1.7, *)) => disease
11	age([*, 45)) AND cp(Non-anginal pain) => disease(No)
12	ca([1, *)) AND thal(Fixed defect) => disease(Yes)
13	sex(Female) AND cp(Asymptomatic) AND thal(Reversable defect) => d
14	age([*, 45)) AND exang(Yes) => disease(Yes)
15	sex(Male) AND chol([295, *)) AND exang(Yes) => disease(Yes)
16	age([49, 57)) AND trestbps([127, 141)) AND thal(Reversable defect) =
17	sex(Female) AND cp(Atypical angina) => disease(No)
18	cp(Non-anginal pain) AND chol([233, 295)) AND slope(Upsloping) => di
19	age([57, 62)) AND trestbps([127, 141)) AND slope(Flat) => disease(Ye

```
Applying JohnsonReducer to training set, discretized...
Computing discernibility matrix...
Done applying JohnsonReducer to training set, discretized
Application took 00:00:01
Applying BatchClassifier to testing set, discretized...
ROC area = 0.919489
ROC std. error = 0.024094
```

c:/temp/class.log

```
24                      Ranking   = (0.689753) Yes (1) 220 rule(s)
25                                  (0.310247) No (0) 201 rule(s)
26  Object 4: ERROR Actual    = Yes (1)
27                  Predicted = No (0)
28                  Ranking   = (0.560159) No (0) 222 rule(s)
29                              (0.439841) Yes (1) 226 rule(s)
30  Object 5: ok    Actual    = No (0)
31                  Predicted = No (0)
32                  Ranking   = (0.511213) No (0) 202 rule(s)
33                              (0.488787) Yes (1) 182 rule(s)
34  Object 6: ok    Actual    = No (0)
35                  Predicted = No (0)
36                  Ranking   = (0.707451) No (0) 214 rule(s)
37                              (0.292549) Yes (1) 192 rule(s)
38  Object 7: ok    Actual    = Yes (1)
39                  Predicted = Yes (1)
40                  Ranking   = (0.753731) Yes (1) 285 rule(s)
41                              (0.246269) No (0) 240 rule(s)
42  Object 8: ok    Actual    = No (0)
43                  Predicted = No (0)
44                  Ranking   = (0.821072) No (0) 287 rule(s)
45                              (0.178928) Yes (1) 239 rule(s)
46  Object 9: ok    Actual    = Yes (1)
47                  Predicted = Yes (1)
48                  Ranking   = (0.804212) Yes (1) 258 rule(s)
49                              (0.195788) No (0) 199 rule(s)
50  Object 10: ok    Actual    = No (0)
51                   Predicted = No (0)
52                   Ranking   = (0.730053) No (0) 243 rule(s)
```

Fig. 1. An example ROSETTA workspace.