A New Version of Rough Set Exploration System

Jan G. Bazan[1], Marcin S. Szczuka[2], and Jakub Wróblewski[3]

[1] Institute of Mathematics, University of Rzeszów
Rejtana 16A, 35-959 Rzeszów, Poland
bazan@univ.rzeszow.pl

[2] Institute of Mathematics, Warsaw University
Banacha 2, 02-097, Warsaw, Poland
szczuka@mimuw.edu.pl

[3] Polish-Japanese Institute of Information Technology
Koszykowa 86, 02-008, Warsaw, Poland
jakubw@mimuw.edu.pl

Abstract. We introduce a new version of the Rough Set Exploration System – a software tool featuring a library of methods and a graphical user interface supporting variety of rough-set-based computations. Methods, features and abilities of the implemented software are discussed and illustrated with a case study in data analysis.

1 Introduction

Research in decision support systems, classification algorithms in particular those concerned with application of rough sets requires experimental verification. To be able to make thorough, multi-directional practical investigations one have to possess an inventory of software tools that automatise basic operations, so it is possible to focus on the most essential matters. Several such software systems have been constructed by various researchers, see e.g. [10, vol. 2]. That was also the idea behind creation of Rough Set Exploration System (**RSES**).

First version of RSES and the library RSESlib was released several years ago. After several modifications, improvements and removal of detected bugs it was used in many applications. Comparison with other classification systems (see [9, 2]) proves its value. The RSESlib was also used in construction of computational kernel of ROSETTA — an advanced system for data analysis (see [15]).

The first version of Rough Set Exploration System (RSES v. 1.0) in its current incarnation was introduced approximately two years ago (see [4]). Present version (2.0) introduces several changes, improvements as well as new algorithms - the result of selected, recent research developments in the area of rough-set-based data analysis.

The RSES software and its computational kernel maintains all advantages of previous version. The algorithms from the previous version have been remastered to provide better flexibility and extended functionality. New algorithms

J.J. Alpigini et al. (Eds.): RSCTC 2002, LNAI 2475, pp. 397–404, 2002.

added to the library follow the current state of our research. Improved construction of the system allows further extensions and supports augmentation of RSES methods into different data analysis tools.

Another important change in terms of technical development is the re-implementation of the RSES core classes in Java^{TM} 2. Most of the computational procedures are now written in Java using its object-oriented paradigms. The migration to Java simplifies some development operations and, ultimately, leads to improved flexibility of the product permitting future migration of RSES software to operating systems other than Windows.

2 Basic Notions

The structure of data that is central point of our work is represented in the form of *information system* or, more precisely, the special case of information system called *decision table.*

Information system is a pair of the form $\mathbf{A} = (U, A)$ where U is a *universe* of *objects* and $A = \{a_1, ..., a_m\}$ is a set of *attributes* i.e. mappings of the form $a_i : U \to V_a$, where V_a is called *value set* of the attribute a_i. The decision table is also a pair of the form $\mathbf{A} = (U, A \cup \{d\})$ with distinguished attribute d. In case of decision table the attributes belonging to A are called *conditional attributes* or simply *conditions* while d is called *decision.* We will further assume that the set of decision values is finite. The i-th *decision class* is a set of objects $C_i = \{o \in U : d(o) = d_i\}$, where d_i is the i-th decision value taken from decision value set $V_d = \{d_1, ..., d_{|V_d|}\}$

For any subset of attributes $B \subset A$ *indiscernibility relation* $IND(B)$ for $x, y \in U$ is defined as follows:

$$x \, IND(B) \, y \Longleftrightarrow \forall_{a \in B} \, a(x) = a(y). \tag{1}$$

Having indiscernibility relation we may define the notion of reduct. $B \subset A$ is a *reduct* of information system if $IND(B) = IND(A)$ and no proper subset of B has this property. In case of decision tables *decision reduct* is a set $B \subset A$ of attributes such that it cannot be further reduced and $IND(B) \subset IND(d)$.

Decision rule is a formula of the form $(a_{i_1} = v_1) \wedge ... \wedge (a_{i_k} = v_k) \Rightarrow d = v_d$, where $1 \leq i_1 < ... < i_k \leq m$, $v_i \in V_{a_i}$. Atomic subformulas $(a_{i_1} = v_1)$ are called *conditions.* We say that rule r is *applicable* to object, or alternatively, the object *matches* rule, if its attribute values satisfy the premise of the rule. With the rule we can connect some numerical characteristics such as *matching* and *support* (see [2,3]).

By *cut* for an attribute $a_i \in A$, such that V_{a_i} is an ordered set we will denote a value $c \in V_{a_i}$. With the use of cut we may replace original attribute a_i with new, binary attribute which tells as whether actual attribute value for an object is greater or lower than c (more in [7]).

Template of $\mathbf{A}$ is a propositional formula $\bigwedge(a_i = v_i)$ where $a_i \in A$ and $v_i \in V_{a_i}$. A generalised template is the formula of the form $\bigwedge(a_i \in T_i)$ where $T_i \subset V_{a_i}$. An object *satisfies* (matches) a template if for every attribute a_i

occurring in the template the value of this attribute on considered object is equal to v_i (belongs to T_i in case of generalised template). The template induces in natural way the split of original information system into two distinct subtables. One of those subtables contains objects that satisfy the template, the other those that do not. Decomposition tree is a binary tree, whose every internal node is labelled by some template and external node (leaf) is associated with a set of objects matching all templates in a path from the root to a given leaf (see [7]).

3 Contents of RSES v. 2.0

3.1 Input/Output Formats

During operation certain functions belonging to RSES may read and write information to/from files. Most of these files are regular ASCII files.

The major change from the previous RSES versions is the format used for representing the basic data entity i.e. the decision table. There is no longer limitation on the type of attribute values. The new file format permits attributes to be represented with use of integer, floating point number or symbolic (text) value. There is also a possibility of using the "virtual", calculated during operation of the system. There may be for example derived as a linear combinations of other attributes as discussed further in the article. The file format used to store decision tables includes header where the user specifies size of the table, name and type of attributes. The effect of such a specification is visible to the user, as attribute names are used for displaying data tables in the RSES GUI.

The user is also given an option of saving the whole workspace (project) in a single file. The project layout together with underlying data structures is stored using dedicated, optimised binary file format.

3.2 The Algorithms

The algorithms that have been implemented in the RSES fall into two general categories.

First category gathers the algorithms aimed at management and edition of data structures. Functions allowing upload and download of data as well as derived structures, procedures for adding/removing objects, setting decision attributes, calculating statistical information about data and others are provided.

The algorithms for performing Rough Set theory based operations on data constitute the second, most essential kind of tools implemented inside RSES. Most important of them are:

Reduction algorithms i.e. algorithms allowing calculation of the collections of reducts for a given information system (decision table). The exhaustive algorithm for calculation of all reducts is present along with implementations of approximate and heuristic solutions such as genetic, covering and Johnson algorithms (see [12,5]).

Rule induction algorithms. Based on the calculated reduct it is possible to calculate decision rules (see [3]). Procedures for rule calculation allow user to determine some crucial constraints for the set of decision rules such as required accuracy, coverage and so on. Rules received are accompanied with several coefficients that are further used while the rules are being applied to the set of objects (see [2]).

Discretisation algorithms. Discretisation permits discovery cuts for attributes. By this process initial decision table is converted to one described with simplified, symbolic attributes; one that is less complex and contains the same information w.r.t. discernibility of objects. More in [7,2].

Algorithms for generation of new attributes – Recently added and discussed in detail further in the paper.

Template generation algorithms provide means for calculation of templates and generalised templates. Placed side by side with template generation are the procedures for inducing table decomposition trees. Details in [8].

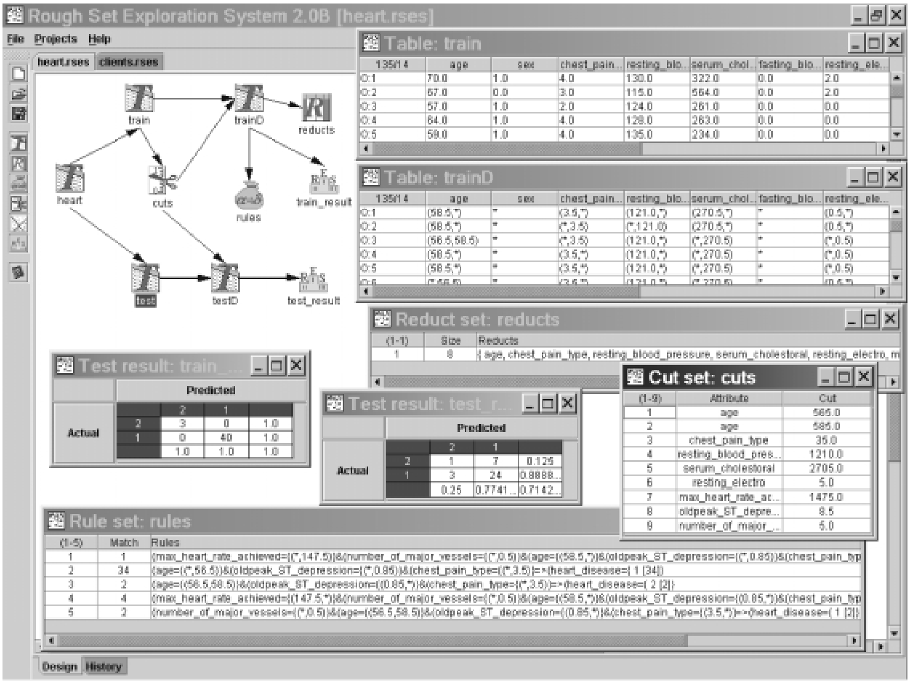
Classification algorithms used determine decision value for objects with use of decision rules and/or templates. Discussion in [8,3,2].

3.3 The RSES GUI

To simplify the use of RSES algorithms and make it more intuitive a graphical user interface was constructed. It is directed towards ease of use and visual representation of workflow. Project interface window (see Fig. 1) consists of two parts. The visible part is the project workspace where icons representing objects occurring during our computation are presented. Behind the project window there is history window dedicated to messages, status reports, errors and warnings produced during operations. The history window is reachable via tab on the bottom part of the interface window.

It was designers intention to simplify the operations on data within project. Therefore, the entities appearing in the process of rough set based computation are represented in the form of icons placed in the upper part of workplace. Such an icon is created every time the data (table, reducts, rules,...) is loaded from the file. User can also place an empty object in the workplace and further fill it with results of operation performed on other objects. The objects that may exist in the workplace are: decision table, collection of reducts, set of rules, decomposition tree, set of cuts, set of new attributes and collection of results. Every object appearing in the project have a set of actions connected with it. By right-clicking on the object the user invokes a context menu for that object. It is also possible to call the necessary action from general pull-down program menu in the main window. Menu choices allow to view and edit objects as well as make them input to some computation. In many cases choice of some option from context menu will cause a new dialog box to open. In this dialog box user can set values of coefficients used in desired calculation. If the operation performed on the object leads to creation of new object or modification of existing one then such a new object is connected with edge originating in object(s) which contributed to its current state. Setting of arrows connecting icons in the workspace changes dynamically as new operations are being performed.

There is also a possibility of working with multiple projects at the same time. In such a case all project windows are placed in the GUI and accessible via tabs in the upper part of main window.



135/14	age	sex	chest_pain...	resting_blo...	serum_chol...	fasting_blo...	resting_ele...
O:1	70.0	1.0	4.0	130.0	322.0	0.0	2.0
O:2	67.0	0.0	3.0	115.0	564.0	0.0	2.0
O:3	57.0	1.0	2.0	124.0	261.0	0.0	0.0
O:4	64.0	1.0	4.0	128.0	263.0	0.0	0.0
O:5	58.0	1.0	4.0	135.0	234.0	0.0	0.0

135/14	age	sex	chest_pain...	resting_blo...	serum_chol...	fasting_blo...	resting_ele...
O:1	(58.5,*)	*	(3.5,*)	(121.0,*)	(270.5,*)	*	(0.5,*)
O:2	(58.5,*)	*	(*,3.5)	(*,121.0)	(270.5,*)	*	(0.5,*)
O:3	(56.5,58.5)	*	(*,3.5)	(121.0,*)	(*,270.5)	*	(*,0.5)
O:4	(58.5,*)	*	(3.5,*)	(121.0,*)	(*,270.5)	*	(*,0.5)
O:5	(58.5,*)	*	(3.5,*)	(121.0,*)	(*,270.5)	*	(*,0.5)
O:6	(*,56.5)	*	(3.5,*)	(121.0,*)	(*,270.5)	*	(0.5,*)

(1-1)	Size	Reducts
1	8	{ age, chest_pain_type, resting_blood_pressure, serum_cholestoral, resting_electro, m

(1-9)	Attribute	Cut
1	age	565.0
2	age	585.0
3	chest_pain_type	35.0
4	resting_blood_pres...	1210.0
5	serum_cholestoral	2705.0
6	resting_electro	5.0
7	max_heart_rate_ac...	1475.0
8	oldpeak_ST_depre...	8.5
9	number_of_major_...	5.0

	Predicted			
Actual		2	1	
	2	3	0	1.0
	1	0	40	1.0
		1.0	1.0	1.0

	Predicted			
Actual		2	1	
	2	1	7	0.125
	1	3	24	0.8888...
		0.25	0.7741...	0.7142...

(1-5)	Match	Rules
1	1	(max_heart_rate_achieved=((*,147.5))&(number_of_major_vessels=((*,0.5))&(age=((58.5,*))&(oldpeak_ST_depression=((*,0.85))&(chest_pain_typ
2	34	(age=((*,56.5))&(oldpeak_ST_depression=((*,0.85))&(chest_pain_type=((*,3.5))=>(heart_disease={ 1 [34]})
3	2	(age=((56.5,58.5))&(oldpeak_ST_depression=((0.85,*))&(chest_pain_type=((*,3.5))=>(heart_disease={ 2 [2]})
4	4	(max_heart_rate_achieved=((147.5,*))&(number_of_major_vessels=((*,0.5))&(age=((58.5,*))&(oldpeak_ST_depression=((0.85,*))&(chest_pain_typ
5	2	(number_of_major_vessels=((*,0.5))&(age=((56.5,58.5))&(oldpeak_ST_depression=((0.85,*))&(chest_pain_type=((3.5,*))=>(heart_disease={ 1 [2]})

Fig. 1. The project interface window

3.4 New Features

In the current version several new methods have been added or extended. The entirely new feature is the possibility for generating new attributes as linear combinations of existing ones (Fig. 2) as discussed in the section 4.1. Another significant addition is the incorporation of the decision rule generation algorithm LEM2 based on the original concept formulated by J. Grzymała-Busse in [6].

The new features are also directly visible to the user when it comes to interaction with RSES GUI. As already mentioned, the central data structure - decision table have been re-designed. Visible result is that the attribute names in the column headers are displayed (see Fig. 1). Other significant new features are:

- Presentation of results in the form of confusion matrix.
- Presentation of decision rules using original names of attributes, which improves readability. Also, each rule is accompanied with the number of objects from different decision classes that are matched by the rule.
- Classification of new (previously unseen) cases. The program can either compare its predictions to the desired values or add one more decision column containing the predicted decision values.

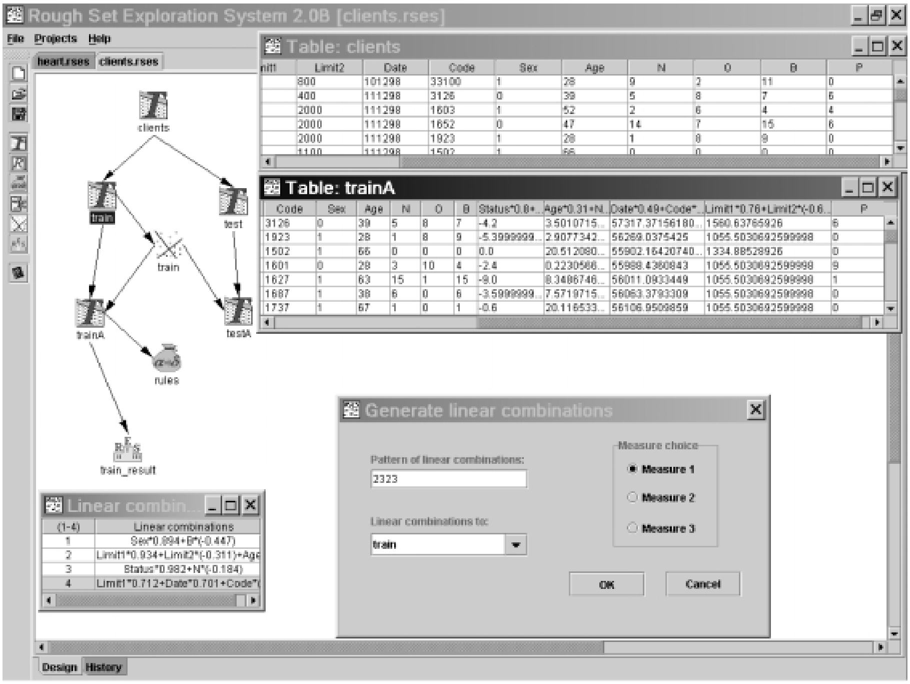


Table: clients

nit1	Limit2	Date	Code	Sex	Age	N	O	B	P
	800	101298	33100	1	28	9	2	11	0
	400	111298	3126	0	39	5	8	7	6
	2000	111298	1603	1	52	2	6	4	4
	2000	111298	1652	0	47	14	7	15	6
	2000	111298	1923	1	28	1	8	9	0
	1100	111298	1502	1	66	0	0	0	0

Table: trainA

Code	Sex	Age	N	O	B	Status*0.8+...	Age*0.31+N...	Date*0.49+Code*...	Limit1*0.76+Limit2*(-0.6...	P
3126	0	39	5	8	7	-4.2	3.5010715...	57317.37156180...	1560.63765926	6
1923	1	28	1	8	9	-5.3999999...	2.9077342...	56269.0375425	1055.5030692599998	0
1502	1	66	0	0	0	0.0	20.512080...	55902.16420740...	1334.88528926	0
1601	0	28	3	10	4	-2.4	0.2230566...	55988.4360843	1055.5030692599998	9
1627	1	63	15	1	15	-9.0	8.3406746...	56011.0933449	1055.5030692599998	1
1687	1	38	6	0	6	-3.5999999...	7.5719715...	56063.3793309	1055.5030692599998	0
1737	1	67	1	0	1	-0.6	20.116533...	56106.9509859	1055.5030692599998	0

Linear combin...

(1-4)	Linear combinations
1	Sex*0.894+B*(-0.447)
2	Limit1*0.934+Limit2*(-0.311)+Age
3	Status*0.982+N*(-0.184)
4	Limit1*0.712+Date*0.701+Code*(

Generate linear combinations

Pattern of linear combinations: 2323

Linear combinations to: train

Measure choice
● Measure 1
○ Measure 2
○ Measure 3

OK Cancel

Fig. 2. New attribute generation - controls

4 Case Study

4.1 Generation of New Attributes

In our approach the original data set is extended by a number of new attributes defined as a linear combination of existing ones. Let $B = b_1, ..., b_m \subseteq A$ be a subset of numerical attributes, $|B| = m$, and let $\alpha = (\alpha_1, ..., \alpha_m) \in \mathbb{R}^m$ be a vector of coefficients. Let $h : U \to \mathbb{R}$ be a function defined as $h(u) = \alpha_1 b_1(u) + ... + \alpha_m b_m(u)$.

Usefulness of new attribute defined as $\overline{a}(u) = h(u)$ depends on proper selection of parameters B and α. The new attribute $\overline{a}$ is useful, when the model of data (e.g. decision rules) based on discretised values of $\overline{a}$ becomes more general (without loss of accuracy). Evolution strategy algorithm optimises $\overline{a}$ using quality functions described below. Three such quality measures are implemented in the current version of RSES. Detailed description can be found in [11,13]. Let L be a straight line in $\mathbb{R}^m$ defined by given linear combination h.

The *distance measure* is average (normalised) distance of objects from different decision classes in terms of $\overline{a}$ (i.e. projected onto L).

The *discernibility measure* takes into account two components: distance (as above) and average discernibility, defined as a sum of squares of cardinalities of decision-uniform intervals defined on L.

The *predictive measure* is an estimate of expected classifier's prediction quality when using only $\overline{a}$. It is constructed with use of some probabilistic methods for approximating the expected values of coverage and sensibility (ability to assign the objects to proper classes; cf. [13]).

4.2 Experimental Results

Two databases from repository [16] were used for experiments: sat_image database (4435 training and 2000 test objects, 36 attributes) and letter_recognition database (15000 training and 5000 test objects, 16 attributes). Four new attributes were generated for each table: two of them as a linear combination of two selected attributes, two other was created basing on three selected attributes (experiments show, that considering more than three attributes hardly improves results, while the computation time grows dramatically). Both the training and test table were extended by four new attributes; only the training tables, however, were used to choose the linear combinations.

Then, the newly created data sets were analysed using two data mining methods: k-NN (for k from 1 to 10; distances on all dimensions were normalised) and a rough set based analyser using local reducts (see [12]) which is implemented as part of RSES. Table 1 presents results of classification of test tables of the databases extended by new attributes as well as containing only these new ones. In the case of local-reduct-based method there is a number of decision rules presented in the last column.

Results show that in case of both k-NN and rough sets based method a table extended with four additional attributes can be analysed more accurately (see Table 1). Moreover, even if only four additional attributes are taken into account, a classification can be done with a pretty good efficiency (e.g. 70.8% of correct answers in case of letter_recognition – this is good result if one recalls that there are 26 possible answers). Note that in these cases we have 4 attributes instead of 36 or 16 – this is a significant compression of information.

The best results obtained in case of both sat_image and letter_recognition database are better than the best results in [9]. However, the result on sat_image is worse than one obtained using k-NN on feature subsets (91.5%, see [1]).

Table 1. Classification efficiency on the test data

Table name	Result (k-NN)	Result (local reducts)	No. of rules
sat_image	90.60%	81.30%	5156
extended	**91.05%**	82.40%	1867
new attributes	84.30%	76.60%	475
letter_recognition	95.64%	79.64%	21410
extended	**95.90%**	79.74%	15506
new attributes	67.80%	70.84%	4569

5 Perspective

The RSES toolkit will further grow as new methods and algorithms emerge. More procedures are still coming from current state-of-the-art research. The article reflect the state of software tools at the moment of writing, i.e. beginning of April. For information on most recent developments visit the Website [14].

Acknowledgement. In the first place the special tribute should be paid to Professor Andrzej Skowron who, for many years, overlooks the pursuit of research that led to the creation of RSES. Development of our software is supported by Polish grant KBN 8T11C02519. M. Szczuka is also supported by the WITAS project of the Wallenberg Foundation.

References

1. Bay, S.D.: Combining Nearest Neighbor Classifiers Through Multiple Feature Subsets. In: Proc. of 15 ICML, Morgan Kaufmann, Madison, 1998
2. Bazan J., A Comparison of Dynamic and non-Dynamic Rough Set Methods for Extracting Laws from Decision Tables, In [10], vol. 1, pp. 321–365
3. Bazan, J.G., Nguyen, H.S., Nguyen, S.H, Synak, P., Wróblewski, J.: Rough Set Algorithms in Classification Problem. In: Polkowski, L., Tsumoto, S., Lin, T.Y. (eds), Rough Set Methods and Applications, Physica-Verlag, 2000 pp. 49–88.
4. Bazan J., Szczuka M. RSES and RSESlib – A Collection of Tools for Rough Set Computations, Proc. of RSCTC'2000, LNAI 2005, Springer Verlag, Berlin, 2001
5. Garey M., Johnson D., Computers and Intarctability: A Guide to the Theory of NP-completness, W.H. Freeman&Co., San Francisco, 1998, (twentieth print)
6. Grzymała-Busse J., A New Version of the Rule Induction System LERS Fundamenta Informaticae, Vol. 31(1), 1997, pp. 27–39
7. Nguyen Sinh Hoa, Nguyen Hung Son, Discretization Methods in Data Mining, In [10] vol.1, pp. 451–482
8. Hoa S. Nguyen, A. Skowron and P. Synak, Discovery of Data Patterns with Applications to Decomposition and Classfification Problems. In [10] vol.2, pp. 55–97.
9. Michie D., Spiegelhalter D. J., Taylor C. C., Machine Learning, Neural and Statistical Classification, Ellis Horwood, London, 1994
10. Skowron A., Polkowski L.(ed.), Rough Sets in Knowledge Discovery vol. 1 and 2, Physica Verlag, Heidelberg, 1998
11. Ślęzak D., Wróblewski J., Classification Algorithms Based on Linear Combinations of Features. In: Proc. of PKDD'99. LNAI 1704, Springer Verlag, Berlin, 1999, pp. 548–553.
12. Wróblewski J., Covering with Reducts – A Fast Algorithm for Rule Generation, Proceeding of RSCTC'98, LNAI 1424, Springer Verlag, Berlin, 1998, pp. 402–407
13. Wróblewski J.: Ensembles of classifiers based on approximate reducts, Fundamenta Informaticae **47** (3,4), IOS Press (2001) 351–360.
14. Bazan J., Szczuka M., The RSES Homepage, `http://alfa.mimuw.edu.pl/˜rses`
15. Ørn A., The ROSETTA Homepage, `http://www.idi.ntnu.no/˜aleks/rosetta`
16. Blake C.L., Merz C.J., UCI Repository of machine learning databases, Irvine, CA: University of California, 1998, `http://www.ics.uci.edu/˜mlearn`