# 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
e-mail: bazan@univ.rzeszow.pl
[2] Institute of Mathematics, Warsaw University
Banacha 2, 02-097, Warsaw, Poland
e-mail: szczuka@mimuw.edu.pl
[3] Polish-Japanese Institute of Information Technology
Koszykowa 86, 02-008, Warsaw, Poland
e-mail: 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. That was the idea behind creation of Rough Set Exploration System, further referred as RSES for short. Also, by providing a flexible, functional and freely available software system we want to simplify the first steps into rough sets. We hope that it will help students and researchers to make a good use of these methods.

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. The RSESlib - library of tools for rough set computations was successfully used for data analysis with encouraging effects. Comparison with other classification systems (see [12, 2]) proves its value. The RSESlib was also used in construction of computational kernel of ROSETTA - an advanced system for data analysis (see [21]).

The first version of Rough Set Exploration System (RSES v. 1.0) in its current incarnation have been introduced approximately two years ago (see [5]). 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 added to the library follow the current state of our research in classification methods originating in rough sets theory. 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 Graphical User Interface (RSES GUI) is constructed with use of the SWING library. The migration to Java simplify some development operations and, ultimately, leads to improved flexibility of the product. It also allows the future migration of RSES software to operating systems other than Windows.

## 2 Basic notions

In order to provide clear description further in the paper and avoid any misunderstandings we bring here some essential definitions from Rough Set theory.

The structure of data that is central point of our work is represented in the form of *information system* [13] 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 $\mathbf{A} = (U, A \cup \{d\})$ where the major feature that is different from the information system is the 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* (sometimes *decision attribute*). 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 $V_d = \{d_1, ..., d_{rank(d)}\}$

For any subset of attributes $B \subset A$ *indiscernibility relation* $IND(B)$ is defined as follows:

$$xIND(B)y \Leftrightarrow \forall_{a \in B} a(x) = a(y) \tag{1}$$

where $x, y \in U$.

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 the notion of *decision reduct* is handy. The decision reduct is a the 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

$$(a_{i_1} = v_1) \wedge ... \wedge (a_{i_k} = v_k) \Rightarrow d = v_d \tag{2}$$

where $1 \leq i_1 < ... < i_k \leq m$, $v_i \in V_{a_i}$. Atomic subformulae $(a_{i_1} = v_1)$ 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. *Support* denoted as $Supp_{\mathbf{A}}(r)$ is equal to the number of objects from $\mathbf{A}$ for which rule $r$ applies correctly i.e. premise of rule is satisfied and the decision given by rule is similar to the one preset in decision table. $Match_{\mathbf{A}}(r)$ is the number of objects in $\mathbf{A}$ for which rule $r$ applies in general. Analogously the notion of matching set for a rule or collection of rules may be introduced (see [2], [4]).

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 [9]).

*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 [8] for more details).

## 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 the files that can be read or written are regular ASCII text files. They particular sub-types can be distinguished by reviewing the contents or identifying file extensions (if used).

The mayor 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" attributes. Such virtual attributes are calculated during operation of the system. There may be for example derived as a linear combinations of other attributes – the result of application of the methods discussed further in the article.

The file format used to store decision tables includes header where the user specifies size of the table (number of columns and rows), name and type of attribute. 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. That makes possible continuation of experiment starting at the very point where previous action was terminated. The project layout together with underlying data structures is stored using dedicated, optimised binary file format.

## 4 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 in order to make basic manipulation with objects easier.

The algorithms for performing Rough Set theory based operations on data constitute the second, most essential kind of tools implemented inside RSES. To give the idea what apparatus is given to the user we describe shortly the most important algorithms.

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, however such operation may be time-consuming due to computational complexity of such task (see [14]). Therefore approximate and heuristic solutions such as genetic, covering and Jonhson algorithms were implemented (see [17], [6] for details). The methods for calculation of reducts allow setting initial conditions for number of reducts to be calculated. Basing on calculated reduct it is possible to calculate decision rules (see [4]). Procedures for rule calculation allow user to determine some crucial constrains 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 [3], [2]). In connection with algorithms for reduct/rule calculation appear the subclass of algorithms allowing shortening of rules and reducts with respect to different requirements (see [3]).

Discretisation algorithms allow to find cuts for attributes. In this way initial decision table is converted to one described with less complex, binary attribute without lose of information about discernibility of objects. Foundations for discretisation algorithms preserving information are given in [11], [9], [2].

Recently added algorithms for generation of new attributes are 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 of that algorithms may be reviewed in [8] and [10].

Classification algorithms used for establishing decision value with use of decision rules and/or templates. Operations for voting among rules with use of two different schemes fall into this category. Thorough discussion of such algorithms can be found in [3], [10], [4] and [2].

### 4.1 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 Figure 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

**Fig. 1.** The project interface window - FIGURE NOT AVALIABLE

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 (linear-combination-based) attributes and collection of results. Every object (icon) 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, in particular, designate the variable which will store the results of invoked operation. 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 (or objects) which contributed to its current state. Setting of arrows connecting icons in the workspace changes dynamically as new operations are being performed.

The entire project can be saved to file on disk to preserve results and information about current setting of coefficients. That also allows to re-create the entire work on other computer or with other data. There is also a possibility of working with multiple projects at the same time. In such a case the all project windows are placed in the GUI and accessible via tabs in the upper part of main window.

### 4.2 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). This approach is discussed in detail in the next section of this paper. Another significant addition is the incorporation of the decision rule generation algorithm LEM 2. Our implementation of LEM is based on the original concept formulated by Jerzy Grzymała-Busse in [7].

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 of this change is the presence of attribute names in the column headers when the table is displayed (see Figure 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. This derives from use of *generalised decision* during calculations.
- Classification of new (previously unseen) cases. In case the test data have the decision column already, the program can either compare its predictions to the desired values or add one more decision column containing the predicted decision values. In case of new cases (without decision) the predictions given by RSES may be stored in newly created, additional column (decision column).

**Fig. 2.** New attribute generation - controls - FIGURE NOT AVALIABLE

## 5 Case study

### 5.1 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 attributes, $|B| = m$, and let $\alpha = (\alpha_1, ..., \alpha_m) \in \mathbf{R}^m$ be a vector of coefficients. Let $h : U \to \mathbf{R}$ be a function defined as:

$$h(u) = \alpha_1 b_1(u) + ... + \alpha_m b_m(u) \tag{3}$$

Usefulness of new attribute defined as $\overline{a}(u) = h(u)$ depends on proper selection of parameters $B$ and $\alpha$. 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 [15, 16] and [19]. Let $L$ be a straight line in $\mathbf{R}^m$ defined by given linear combination $h$. The general idea of these measures is given below.

- 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. This 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. [18, 19]).

### 5.2 Experimental Results

Two databases from repository [22] 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 was 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 was 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 [17] for details) 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 reducts 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 was 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 take into account that there is 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 reported in [12]. However, the result on sat_image is worse than one obtained using k-NN on feature subsets (91.5%, see [1]).

| Table name | Result (k-NN) | Result (local reducts) | No. of rules |
|---|---|---|---|
| sat_image | 90.60% | 81.30% | 5156 |
| extended, $Q_1$ | 90.30% | 79.50% | 3405 |
| extended, $Q_{mod}$ | **91.05%** | 82.40% | 1867 |
| new attributes, $Q_1$ | 81.65% | 64.50% | 445 |
| new attributes, $Q_{mod}$ | 84.30% | 76.60% | 475 |
| letter_recognition | 95.64% | 79.64% | 21410 |
| extended, $Q_1$ | 92.00% | 81.64% | 17587 |
| extended, $Q_{mod}$ | **95.90%** | 79.74% | 15506 |
| new attributes, $Q_1$ | 50.40% | 45.40% | 1765 |
| new attributes, $Q_{mod}$ | 67.80% | 70.84% | 4569 |

**Table 1.** Classification efficiency on the test data

## 6 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 [20].

## 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 was supported by grant 8T11C02519 from Polish State Committee for Scientific Research. M. Szczuka is also supported by the Wallenberg Foundation in frame of the WITAS project.

## References

1. Bay, S.D.: Combining Nearest Neighbor Classifiers Through Multiple Feature Subsets. In: Proc. of the International Conference of the Machine Learning. Morgan Kaufmann, Madison,1998
2. Bazan J., A Comparison of Dynamic and non-Dynamic Rough Set Methods for Extracting Laws from Decision Tables, In: Skowron A., Polkowski L.(ed.), Rough Sets in Knowledge Discovery 1, Physica Verlag, Heidelberg, 1998, pp. 321-365
3. Bazan J., Approximate Reasoning Methods for Synthesis of Decision Algorithms (in Polish), Ph. D. Thesis, Department of Math., Comp. Sci. and Mechanics, Warsaw University, Warsaw, 1998
4. 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: New Developments in Knowledge Discovery in Information Systems. Physica-Verlag, 2000 pp. 49–88.
5. Bazan J., Szczuka M. RSES and RSESlib - A Collection of Tools for Rough Set Computations, Proceeding of RSCTC'2000, LNAI 2005, Springer Verlag, Berlin, 2001

6. Garey M., Johnson D., Computers and Intarctability: A Guide to the Theory of NP-completness, W.H. Freeman&Co., San Francisco, 1998, (twentieth print)
7. Grzymała-Busse J., A New Version of the Rule Induction System LERS Fundamenta Informaticae, Vol. 31(1), 1997, pp. 27–39
8. Nguyen Sinh Hoa, Data Regularity Analysis and Applications in Data Mining. Ph. D. Thesis, Department of Math., Comp. Sci. and Mechanics, Warsaw University, Warsaw, 1999
9. Nguyen Sinh Hoa, Nguyen Hung Son, Discretization Methods in Data Mining, In: Skowron A., Polkowski L.(ed.), Rough Sets in Knowledge Discovery 1, Physica Verlag, Heidelberg, 1998, pp. 451-482
10. Hoa S. Nguyen, A. Skowron and P. Synak, Discovery of Data Patterns with Applications to Decomposition and Classfification Problems. In: L. Polkowski and A. Skowron (eds.), Rough Sets in Knowledge Discovery 2, Physica-Verlag, Heidelberg, 1998, pp. 55-97.
11. Nguyen Hung Son, Discretization of Real Value Attributes. Boolean Reasoning Approach. Ph. D. Thesis, Department of Math., Comp. Sci. and Mechanics, Warsaw University, Warsaw, 1997
12. Michie D., Spiegelhalter D. J., Taylor C. C., Machine Learning, Neural and Statistical Classification, Ellis Horwood, London, 1994
13. Pawlak Z., Rough Sets: Theoretical Aspects of Reasoning about Data, Kluwer, Dordrecht, 1991
14. Rauszer C., Skowron A., The Discernibility Matrices and Functions in Information Systems, In: Słowiński R. (ed.), Intelligent Decision Support, Kluwer, Dordrecht, 1992
15. Ślęzak D., Wróblewski J., Classification Algorithms Based on Linear Combinations of Features. In: Proc. of PKDD'99. Praha, Czech Republic, LNAI 1704, Springer Verlag, Berlin, 1999, pp. 548–553.
16. Ślęzak D., Synak P., Wieczorkowska A., Wróblewski J., KDD-based Approach to Musical Instrument Sound Recognition, Proceedings of the $9^th$ International Conference IPMU 2002, Annecy, France, 2002
17. Wróblewski J., Covering with Reducts - A Fast Algorithm for Rule Generation, Proceeding of RSCTC'98, LNAI 1424, Springer Verlag, Berlin, 1998, pp. 402-407
18. Wróblewski J.: Ensembles of classifiers based on approximate reducts, Fundamenta Informaticae **47** (3,4), IOS Press (2001) 351–360.
19. Wróblewski, J.: Adaptive Methods of Object Classification. Ph. D. Thesis, Department of Math., Comp. Sci. and Mechanics, Warsaw University, Warsaw, 2001
20. Bazan J., Szczuka M., The RSES Homepage, `http://alfa.mimuw.edu.pl/~rses`
21. Ørn A., The ROSETTA Homepage, `http://www.idi.ntnu.no/~aleks/rosetta`
22. Bay, S. D. , The UCI ML Archive, `http://www.ics.uci.edu/ml`