Generation of Decision Rules from Nondeterministic Decision Table based on Rough Sets Theory

Pavel Jirava (Institut of System Engineering and Informatics, Faculty of Economic and Administration University of Pardubice, Czech Republic) - E-mail: pavel.jirava@upce.cz
Jiří Křupka (Institut of System Engineering and Informatics, Faculty of Economic and Administration University of Pardubice, Czech Republic) - E-mail: jiri.krupka@upce.cz

A process of gaining decision rules from nondeterministic decision table with use of Rough sets theory is presented in this paper. Explored data in decision tables were collected in our previous research on information systems1. To deal with given problem, we reflect different approaches and algorithms for generating minimal decision rules. Further is introduced original tool used for computation, which was designed in MATLAB.

Key-words: data analysis, nondeterministic decision table, rough sets, decision rules

[1] JIRAVA, P., KŘUPKA, J. Rough Sets and Evaluation of Information System. *Proc. of the 11th Int. Conference on Soft Computing: Mendel 2005*, Brno, 2005, pp.178-183. ISBN: 80-214-2961-5.

1. Introduction

In presented paper, we introduce outputs from our work in the area of data mining and data analysis. We use here tools of Rough sets theory for generation of decision rules from data collected in decision tables. The main goal of the rough sets (RSs) analysis is to synthesize approximation of concepts from the acquired data [1,2]. Every object we explore we associate with some information (data). Objects characterized by the same data are indiscernible in view of the available information about them. The indiscernibility relation generated in this way is the mathematical basis of RSs theory [3].

The assumption that objects can be seen only through the information available about them leads to the view that knowledge has granular structure. Thus some objects appear as similar and undiscerned. Therefore in rough set theory we assume that any vague concept is replaced by a pair of precise concepts – the lower and the upper approximation of the vague concept. The lower approximation consists of all objects which surely belong to the concept and upper approximation of all objects which possibly belong to the concept. And the difference between the upper and lower approximation is called the boundary region. The approximations are two basic operations in RSs theory [3]. Suppose we are given two finite and non empty sets *U* and *A*, *U* is called the universe and *A* is a set of attributes. With attributes $a \in A$ we associate a set V_a (value set) called the domain of a. Any subset *B* of *A* determines a binary relation *I(B)* on *U* which will be called an indiscernibility relation. Indiscernibility relation (denoted *IND*) is defined [2] :

$$IND(B) = \{(x, y) \in U^2 \mathrm{I} \forall a \in Ba(x) = a(y)\}, \qquad (1)$$

where *IND(B)* is an equivalence relation and is called *B*-indiscernibility relation. If $(x, y) \in IND(B)$ then x and y are *B*- indiscernible (indiscernible from each other by attributes from *B*). The equivalence classes of the *B*-indiscernibility relation will be denoted *B(x).*

The indiscernibility relation will be used now to define basic concept of rough sets theory.

Let *A*= *(U, A)* be an information system and let $B \subseteq A$ *and* $X \subseteq U$. We can approximate *X* using only the information contained in *B* by constructing $\underline{B}(X)$ (lower approximation) and $\overline{B}(X)$ (upper approximation) of *X* on the following way:

$$\underline{B}(X) = \{ x \in U : B(x) \subseteq X \} \text{ and } \overline{B}(X) = \{ x \in U : B(x) \cap X \neq \emptyset \}. \qquad (2)$$

The objects in $\underline{B}(X)$ can be with certainly classified as a members of *X* on the basis of knowledge in *B* and the objects in $\overline{B}(X)$ are classified as possible members of *X* on the basis of knowledge in *B*. The set

$$BN_B(X) = \overline{B}(X) - \underline{B}(X) \qquad (3)$$

is called the boundary region of *X* and thus consist of those objects that we cannot decisively classify into *X* on the basis of knowledge *B*. If the boundary region is empty, then the set *X* is crisp with respect to *B*. If the boundary region is not empty, the set *X* is rough with respect to *B*.

Rough sets are defined by approximations and have properties defined in [2,3,5]. We can define four basic classes of RSs (four categories of vagueness - *X* is roughly *B*-definable; *X* is internally or externally or totally *B*-undefinable) [2] on the following way:

The *X* is roughly *B*-definable, if

$$\underline{B}(X) \neq \emptyset \ and \ \overline{B}(X) \neq U. \qquad (4)$$

This means that we are able to decide for some elements of *U* whether they belong to *X* or *–X*, using *B*.

The *X* is internally *B*-undefinable, if

$$\underline{B}(X) = \varnothing \ \ and \ \ \overline{B}(X) \neq U \, . \tag{5}$$

This means that we are able to decide whether some elements of *U* belong to *–X*, but we are unable to decide for any element of *U*, whether it belongs to *X* or not, using *B*.

The *X* is externally *B*-undefinable, if

$$\underline{B}(X) \neq \varnothing \ \ and \ \ \overline{B}(X) = U \, . \tag{6}$$

This means that we are able to decide for some elements of *U* whether they belong to *X*, but we are unable to decide, for any element of *U* whether it belongs to *–X* or not, using *B*.

The *X* is totally *B*-undefinable, if

$$\underline{B}(X) = \varnothing \ \ and \ \ \overline{B}(X) = U \, . \tag{7}$$

This means that we are unable to decide for any element of *U* whether it belongs to *X* or *–X*, using *B*.

2. Model Definition

Processed data were represented as a table. Every column represents an attribute that can be measured for each object. The attribute may be also supplied by a human expert or user. Each row represents a case or generally an object. This table is called an information system [2]. More formally, the information system *IS* is the 4-tuple

$$IS=(U, A, V_a, f), \tag{8}$$

where *U* is a finite sets of objectives (universe), $A= \{a_1, a_2,...,a_m\}$ is a finite set of attributes, V_a is the domain of the attribute *a*, $V = \mathrm{U}_{a\in A} V_a$ and $f\colon U\times A \rightarrow V_a$ is a total function such that $f(x,a) \in V_a$ for each $a \in A$, $x \in U$, called information function [5].

We evaluated data from two information systems (the information system STAG and the library information system Daimon Opac1.6.0- OPAC), which runs on university intranet[2]. First of them is system STAG. Its main goal of the database system is to provide an organizational and administrative support to this system of study for students, staff, departments and faculties. The main goals of the second of them are online book reservation, retrieval catalogue and library services, quick searching in library database and monitoring reader's accounts. Data processing and rules generation in experimental part is described by the procedures in the Fig. 1.

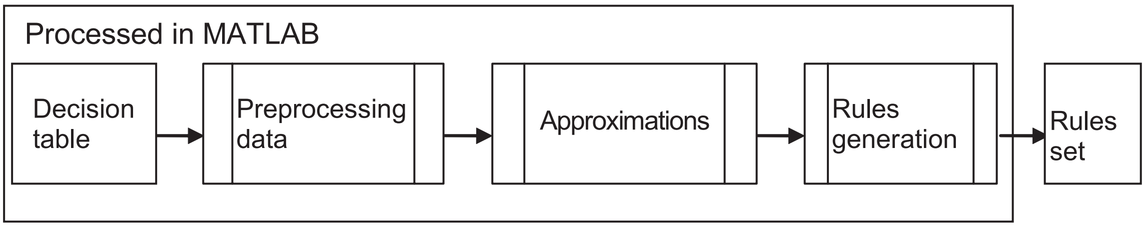


Fig. 1 Algorithm of rules generation

Generation of Decision Rules

So far many methods have been developed to generate decision rules from decision table [10,13]. We want to refer about some of them now. Method to generate decision rules

[2] University of Pardubice

using the concept of dynamic reducts (based on RSs) was presented in 1996 [7]. Methods based on reducts allow us to compute descriptions of all decision classes in the form of decision rules, for a given decision table D_s. In search for robust decision algorithms this method seems to be very good, how confirm experiments (for example in [8,7]). For a decision table $D_s = (U,Q,d)$, any system $B=(U',Q,d)$ such that $U' \subseteq U$ we call a subtable of A. By $P(A_d)$ we denote the set of all subtables of A_d. For $F \subseteq P(A_d)$ by $DR(A,F)$ we denote the set $RED(Ad) \cap \bigcap_{B \in F} RED(B)$.

Any element of $DR(A,F)$ is called an F-dynamic reduct of A.

For $\varepsilon \in [0,1]$ we use notion (F,ε)-dynamic reduct. Set $DR(A,F)$ of all (F,ε)-dynamic reduct is defined:

$$DR_\varepsilon(A,F) = \{ C \in RED(A_d): \ \frac{card(\{B \in F : C \in RED(B)\})}{card(F)} \geq 1 - \varepsilon \}, \tag{9}$$

for $C \in RED(A_d)$ the number: $\dfrac{card(\{B \in F : C \in RED(B)\})}{card(F)}$. (10)

is termed the stability coefficient of the reduct C, relative to F [6,7,8].

Another approach for creating a set of minimal rules proposed in [9,11] is employing Apriory algorithm. Some approaches based on the Apriori algorithm, in a post-processing stage, allow us to reduce decision rules obtained by means of RSs. They assume that a positive region has already been calculated, so that a set of possible redundant rules are available. The set of dependencies obtained by applying the Apriori helps us to discover irrelevant attribute values [11].

Another interesting approach to the attributes reduction and generating minimal decision rules set is described in [9]. Authors used here RSs and neural networks as two common techniques applied to this problem. This hybrid approach of rough sets and neural networks for mining classification rules consists of three major phases: - attribute reduction by rough sets; the further reduction of decision table by neural networks and rule extraction from decision table by rough sets.

Coverage and certainty

Every decision rule is an implication if Φ then Ψ , where Φ is condition and Ψ is decision; Φ and Ψ are logical formulas created from attributes values and described some properties of facts. With every decision rule we associate two conditional probabilities: the certainty factor (*CeF*) and coverage factor (*CoF*) [4], where:

$$CeF(\Psi I \Phi) = \frac{number\ of\ all\ cases\ satisfying\ \Phi\ and\ \Psi}{number\ of\ all\ cases\ satisfying\ \Phi}, \tag{11}$$

$$CoF(\Phi I \Psi) = \frac{number\ of\ all\ cases\ satisfying\ \Phi\ and\ \Psi}{number\ of\ all\ cases\ satisfying\ \Psi}. \tag{12}$$

If CeF=1 then the rule is called certain and if 0< CeF <1 then the rule is called uncertain.

3. Results

For computation were used two attributes from collected data : the 1st attribute (a_1) “amount of investment resources” with scope low, middle and high; the 2nd one (a_2)

"graphical interface is friendly" with scope yes, no; The 3rd decision attribute (*D*) "implementation of IS" with scope yes, no (see Table 1).

The set of all deployment decision can be described approximately as noted below.

$$\textit{Rule } X_n\textit{: If } (a_1 \textit{ is } Va_1) \textit{ and } (a_2 \textit{ is } Va_2) \quad \textit{then } (D \textit{ is } V_d), \qquad (13)$$

where a_1, a_2 are attributes, Va_1 and Va_2 are values of attributes, *D* is decision and V_d is value of decision.

Data used in experimental part were in simple tabulator delimited format as shown in the Table 1. The 1st column represents attribute a_1 , 2nd is attribute a_2 and the last column represents decision attribute *D.* Then were data loaded into MATLAB as shown in Fig. 2.

The input of Graphic User Interface (GUI) is a knowledge representation system *IS = (U, A)*. The information system is formatted as a tabulator delimited table. GUI output is s sorted reduced decision table with computed approximations.

A set of decision rules is generated from this table. Every row in the table is associated with one decision rule. The decision table may include inconsistent rules.

The set of all decision rules for our table can be described approximately as noted below where *deployment* it means *D*:

Rule 1:If (a_1 is low) and (a_2 is friendly) then (*deployment* is yes),
Rule 2:If (a_1 is low) and (a_2 is not friendly) then (*deployment* is yes),
Rule 3:If (a_1 is middle) and (a_2 is friendly) then (*deployment* is yes),
Rule 4:If (a_1 is middle) and (a_2 is not friendly) then (deployment is yes),
Rule 5:If (a_1 is high) and (a_2 is friendly) then (*deployment* is yes),
Rule 6:If (a_1 is high) and (a_2 is not friendly) then (*deployment* is yes),
Rule 7:If (a_1 is low) and (a_2 is friendly) then (*deployment* is no),
Rule 8:If (a_1 is low) and (a_2 is not friendly) then (*deployment* is no),
Rule 9:If (a_1 is middle) and (a_2 is friendly) then (*deployment* is no),
Rule 10: If (a_1 is middle) and (a_2 is not friendly) then (*deployment* is no),
Rule 11: If (a_1 is high) and (a_2 is friendly) then (*deployment* is no),
Rule 12: If (a_1 is high) and (a_2 is not friendly) then (*deployment* is no).

Table 1 Input data

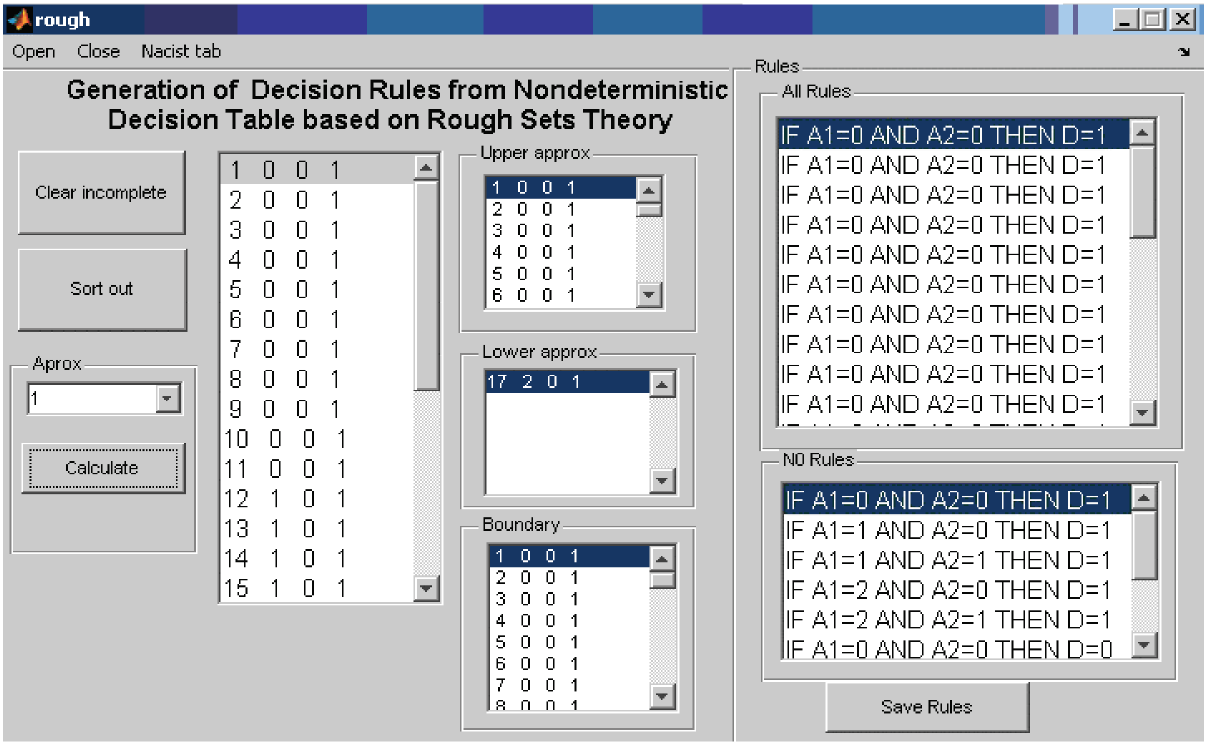
a_1	a_2	*D*
0	0	1
0	0	1
0	0	1
0	0	1
0	0	1
0	0	1
0	0	1
0	0	1
0	0	1
0	0	1
0	0	1
1	0	1
1	0	1

1	0	1
1	1	1
2	0	1
2	1	1
2	1	1
0	0	0
0	0	0
0	1	0
0	1	0
0	1	0
1	0	0
1	0	0
1	1	0
2	1	0

Next step is computation of coverage and certainty factors. Finally, for particular information systems, we can draw following conclusions based on coverage and certainty factors and decision rules:

- for IS-STAG - low investment resources and friendly graphical interface caused positive decision (deployment = yes) in 50 % of the causes; middle investment resources and friendly graphical interface caused positive decision (deployment = yes) in 29 % of the causes; high investment resources and friendly graphical interface caused positive decision (deployment = yes) in 80 % of the causes; 42 % positive decisions occurred when investment resources are low and graphical interface is friendly; 8 % positive decisions occurred when investment resources are low and graphical interface is unfriendly; 17 % positive decisions occurred when investment resources are middle and graphical interface is friendly; 33% positive decisions occurred when investment resources are high and graphical interface is friendly; otherwise 92 % positive decisions occurred when graphical interface is friendly;
- for IS OPAC - low investment resources and friendly graphical interface caused positive decision (deployment = yes) in 85 % of the causes; middle investment resources and friendly graphical interface caused positive decision (deployment = yes) in 67 % of the causes; high investment resources and friendly graphical interface caused always positive decision (deployment = yes); 58 % positive decisions occurred when investment resources are low and graphical interface is friendly; 21 % positive decisions occurred when investment resources are middle and graphical interface is friendly; 5 % positive decisions occurred when investment resources are middle and graphical interface is unfriendly; 5% positive decisions occurred when investment resources are high and graphical interface is friendly; 11% positive decisions occurred when investment resources are high and graphical interface is unfriendly; otherwise 84 % positive decisions occurred when graphical interface is friendly.

We can see that for informants is crucial attribute graphical interface. In the first case (IS STAG) 92% and in the second (IS OPAC) 84% positive decisions occurred, when graphical interface is friendly



```
1  0  0  1
2  0  0  1
3  0  0  1
4  0  0  1
5  0  0  1
6  0  0  1
7  0  0  1
8  0  0  1
9  0  0  1
10  0  0  1
11  0  0  1
12  1  0  1
13  1  0  1
14  1  0  1
15  1  0  1
```

Upper approx

```
1  0  0  1
2  0  0  1
3  0  0  1
4  0  0  1
5  0  0  1
6  0  0  1
```

Lower approx

```
17  2  0  1
```

Boundary

```
1  0  0  1
2  0  0  1
3  0  0  1
4  0  0  1
5  0  0  1
6  0  0  1
7  0  0  1
8  0  0  1
```

All Rules

```
IF A1=0 AND A2=0 THEN D=1
IF A1=0 AND A2=0 THEN D=1
IF A1=0 AND A2=0 THEN D=1
IF A1=0 AND A2=0 THEN D=1
IF A1=0 AND A2=0 THEN D=1
IF A1=0 AND A2=0 THEN D=1
IF A1=0 AND A2=0 THEN D=1
IF A1=0 AND A2=0 THEN D=1
IF A1=0 AND A2=0 THEN D=1
IF A1=0 AND A2=0 THEN D=1
```

N0 Rules

```
IF A1=0 AND A2=0 THEN D=1
IF A1=1 AND A2=0 THEN D=1
IF A1=1 AND A2=1 THEN D=1
IF A1=2 AND A2=0 THEN D=1
IF A1=2 AND A2=1 THEN D=1
IF A1=0 AND A2=0 THEN D=0
```

Fig. 2 GUI constructed for computations

4. Conclusion

Several approaches for generating decision rules from decision tables were proposed up to the present day. We have presented here basic ideas of approach based on rough set theory. Some of the tools, which are based on rough sets, were used in our original application designed in MATLAB. We suppose that our future work will trend to a complementarity between different methods (e.g. rough sets and neural networks) and to creation of hybrid approaches utilizing these methods.

5. Acknowledgement

This paper has been prepared as a part of the institutional projects number FG/FES/2006/31 and FG/FES/462026.

References

1. PAWLAK, Z. Rough sets. In: *Int. J. of Information and Computer Sciences*, *11, 5.* 1982, pp.341-356.
2. KOMOROWSKI, J., POLKOWSKI, L., SKOWRON, A. Rough sets: a tutorial. In: *S.K. Pal and A. Skowron, editors, Rough-Fuzzy Hybridization: A New Method for Decision Making*, Springer-Verlag, Singapore, 1998.
3. PAWLAK, Z. Rough set elements. In: *Rough Sets in Knowledge Discovery I – Methodology and Applications*, Physica Verlag, Heidelberg, 1998, pp.10-31.
4. PAWLAK, Z. A Primer on Rough Sets: A New Approach to Drawing Conclusions from Data. In: *Cardozo Law Review,* Volume 22, Issue 5-6, July, 2001, pp.1407-1415.

5. KŘUPKA, J. Rough Sets Theory in Decision Analysis. In: *Scientific papers of the University of Pardubice, Series D 9/2004*. Pardubice: Univerzita Pardubice, 2004, pp.93-99. ISSN 1211-555X.
6. POLKOWSKI, L. *Rough Sets - Mathematical foundations*, Physica-Verlag A Springer-Verlag company, Heidelberg, 2002, ISBN : 3-7908-1510-1
7. BAZAN, J.G. Dynamic reducts and statistical inference. In: *5-th Int. Conf.Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU'96*. Granada, Spain, 1996, pp.1147-1151.
8. BAZAN, J., NGUYEN, H.S., NGUYEN, S.H., SYNAK, P., WRÓBLEWSKI J. Rough Set Algorithms in Classification Problem. In: *Rough Set Methods and Applications. L. Polkowski, S. Tsumoto and T.Y. Lin, editors.* Physica -Verlag, Heidelberg, New York 2000, pp. 49 - 88.
9. KRYSZKIEWICZ, M. String rules in large databases. In*: 7-th Int. Conf. Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU98*, 1998, La Sorbonne, Paris, Vol.2, pp.1520-1527.
10. SAKAI, H., NAKATA, M. On Possible Rules and Apriori Algorithm in Non-deterministic Information Systems. In*: Rough Sets and Current Trends in Computing RSCTC 2006*, Kobe, Japan, November 6-8, 2006, ISBN: 3-540-47693-8
11. FERNANDEZ, M., MENASALVAS, E.,, MARBÁN, O., PENA, J.M., MILLÁN, S. Minimal Decision Rules Based On the Apriori Algorithm. In: *International J. Appl. Math. Comput. Sci.,* 2001, Vol.11, No.3, pp. 691-704.
12. LI, R., ZHENG-OU, W. Mining classification rules using rough sets and neural networks. In: *European Journal of Operational Research*, Elsevier Science Inc., Amsterdam, number 2, volume 157, pages: 439-448, 2004, ISSN: 0377-2217
13. KUDO, Y., MURAI, T. A method of Generating Decision Rules in Object Oriented Rough Set Models. In*: Rough Sets and Current Trends in Computing RSCTC 2006*, Kobe, Japan, November 6-8, 2006, ISBN: 3-540-47693-8