ROUGH SET THEORY FOR DISCRETIZATION BASED ON BOOLEAN REASONING AND GENETIC ALGORITHM

Srilatha Chebrolu

Department of Computer Science and Engineering,

NIT Warangal, AP, India

Dr. Sriram G Sanjeevi

Associate Professor,

Department of Computer Science and Engineering,

NIT Warangal, AP, India

Abstract - Real world datasets may be continuous. Many data analysis algorithms work efficiently on discrete data while some other algorithms work only on discrete data. Thus the continuous datasets are discretized as a pre-process step to knowledge acquisition. Attribute discretization is the process of reducing the domain of a continuous attribute with irreducible and optimal set of cuts, while preserving the consistency of the dataset classification. In this paper, we use discernibility relations of Rough Set Theory (RST) and propose a 2-step discretization process, where the set of cuts returned from *MD-Heuristics* approach are further reduced using Genetic Algorithm (GA). Experiments on datasets from UCI Machine Learning Repository show that the proposed discretization process is efficient in finding a consistent and irreducible set of cuts.

Keywords - Discretization, Genetic Algorithm, Rough Set Theory

1. Introduction

Any real world data can be represented and stored in the form of an information table, also known as decision table. All rows of the decision table called objects or examples make up knowledge, and are described by a

set of properties called attributes. Analysis of decision table, to extract patterns and for classification of objects, is an important task in data mining, knowledge discovery, decision analysis, machine learning and pattern recognition. Quality of data analysis algorithms on decision tables with continuous attributes is good, when attribute domain is small, as some algorithms work only on discrete data, while some other algorithms work efficiently on discrete data. The process of reducing the domain of continuous attributes is called discretization, and is achieved by replacing the domain of the continuous attributes with a finite number of discrete intervals. Various discretization approaches have been described in literature [2], [3], [8], [6]. Discretization of continuous attributes is shown to be a NP-hard problem in [9], [10], [7] by characterizing the computational complexity of the problem in terms of RST discernibility relations. Heuristic discretization methods based on RST discernibility relations and boolean reasoning are well studied in [16], [12] giving a suboptimal solution. We propose a *2-step discretization approach*, using RST discernibility relations, MD-Heuristics approach and Genetic Algorithm. Discretization can also be achieved using GA alone, but the search space will be huge for large datasets and more time will be taken to find optimal set of cuts [1]. Thus we are reducing the GA search space by considering only the set of cuts returned from MD-Heuristics. In this discretization process, all the superfluous cuts from MD-heuristics will be reduced using GA, thus consistent and irreducible set of cuts are identified with in less time.

The rest of the paper is organized as follows: Section 2 introduces basic concepts of *RST*, Section 3 describes the concepts of discretization and discernibility matrix discretization approach and *MD-Heuristics approach*, Section 4 describes the proposed *2-step discretization approach*, Section 5 describes experimental results on different datasets and Section 6 describes the conclusion of the paper.

2. Rough Sets

RST was introduced by Pawlak in 1982 [17], [14], [13], a mathematical methodology in data analysis, to handle uncertain information in data sets. RST carries through challenging tasks like attribute reduction, attribute discretization, identifying patterns in data, computation of attribute relevance and dataset characterization. Some of the applications of RST include machine learning, data mining, decision analysis, pattern recognition and knowledge discovery [11].

Decision table is defined as a 5 tuple $S = (U, C, d, V, f)$, where U is the universe of all objects $\{o_1, o_2, o_3, \dots\}$, C is the set of all conditional attributes $\{c_1, c_2, c_3, \dots\}$, d is the decision attribute, $d \notin C$, d decides the class of an object, let $\{x_1, x_2, x_3, \dots\}$ be distinct decisions in S, $V = \cup_{c \in C} V_c$, where V_c is the domain of the conditional attribute c, $f: U \times C \to V$ is a mapping function, where $f(o, c)$ represents a value for object o on attribute c in the domain V_c.

The principal idea of RST is indiscernibility relation I_A. Object o_i is said to be indiscernible from object o_j, if they have same values for all attributes in A. I_A is also known as equivalence relation as it satisfies reflexive, symmetric and transitive properties.

$$I_A = \{(o_i, o_j) \in U \times U \mid f(o_i, a) = f(o_j, a), \ \forall \ a \in A, A \subseteq C\}$$

Equivalence class of an object o_i, $[o_i]_{I_A}$ is defined as the set of objects those that are indiscernible from o_i. Decision class of a decision x_i is defined as, the set of all objects with x_i as their decision. Decision class of x_i is denoted by X_i,

$$X_i = \{ oj \mid f(oj, d) = x_i, \forall\ oj \in U \}$$

The set of all decision classes, D partitions U.

$$D = \{X_1, X_2, X_3, \ldots\}, \quad U = \bigcup X_i$$

Rough set of a decision class X_i on any subset of conditional attributes $A \subseteq C$, is defined by a lower approximation and an upper approximation. Lower approximation of X_i on A is defined as, the set of objects that are certainly belonging to decision class X_i. Upper approximation of X_i on A is defined as, the set of objects that may belong to decision class X_i.

$$\underline{apr}^A(X_i) = \bigcup \left\{ [o_j]_{I_A} \mid [o_j]_{I_A} \subseteq X_i \right\}$$

$$\overline{apr^A}(X_i) = \bigcup \left\{ [o_j]_{I_A} \mid [o_j]_{I_A} \cap X_i \neq \phi \right\}$$

Consistency of a decision table is defined in terms of a generalized decision function. A generalized decision function of an object o_i on a set of conditional attributes A, $A \subseteq C$, is defined as the set of decisions of all the objects in the equivalence class of o_i.

$$\partial : U \to 2^{V_d} \quad \text{and} \quad \partial_A(o_i) = \left\{ f(o_j, d) \mid o_j \in [o_i]_{I_A} \right\} \text{ where}$$

$$V_d = \bigcup f(o_i, d) = \{x_1, x_2, x_3, \ldots\}$$

Decision table is said to be consistent, if the cardinality of ∂_C is 1 for all the objects in U.

$$S = \begin{cases} consistent, & |\partial_C(o_i)| = 1, \forall\ o_i \in U \\ inconsistent, & otherwise \end{cases}$$

3. Discretization of continuous attributes

Let S be a consistent decision table and let $V_c = [l_c,\ r_c) \subset \Re$, where $c \in C$, $\Re$ is the set of real numbers and $l_c < r_c$. Any pair (c, v) is called a cut on V_c, where $v \in V_c$.

Definition 1[12]. The set of basic cuts on an attribute $c \in C$, denoted by B_c, is defined as

$$B_c = \left\{ \left(c, \frac{(v_1^c + v_2^c)}{2}\right), \left(c, \frac{(v_2^c + v_3^c)}{2}\right), \ldots, \left(c, \frac{(v_{k-1}^c + v_k^c)}{2}\right) \right\}$$

where $v_1^c < v_2^c < \ldots < v_{k_c}^c$ is a sequence of continuous values defined by c and

$$\bigcup f(o_i, c) = \{v_1^c, v_2^c, \ldots, v_{k_c}^c\}$$

Let B be the set of all basic cuts defined on all conditional attributes

$$B = \bigcup_{\{c \in C\}} B_c$$

Definition 2[12]. A new decision system $P - discretization\ of\ S$, is defined as a 6 tuple $S^P = (U, C, d, P, V^P, f^P)$, where P is the set of cuts

$$P = \bigcup_{c \in C} \{P_c | P_c = \{p_1^c, p_2^c, \dots, p_k^c\}, p_1^c < p_2^c < \dots < p_k^c\}$$

$$f^P(o,c) = \begin{cases} 0, & f(o,c) < p_1^c \\ i, & f(o,c) \in [p_i^c, p_{i+1}^c), 1 \le i \le k-1 \\ k+1, & f(o,c) > p_k^c \end{cases}$$

Quality of $P - discretization\ of\ S$ is defined as the ratio of the number of all objects in lower approximation to the total number of objects in U.

$$\gamma = \frac{apr^P(D)}{|U|}, \qquad where\ \underline{apr^P}(D) = \bigcup_{X_i \in D} \underline{apr^A}(X_i)$$

Definition 3[12]. A set of cuts P, is called $S - consistent$ if $\partial = \partial^P$, where ∂ and ∂^P are the generalized definition functions of S and S^P respectively.

Definition 4[12]. A set of cuts P, is called $S - irreducible$ if P is $S - consistent$ and for any P', $P' \subset P$, P' is not $S - consistent$.

Definition 5[12]. A set of cuts P, is called $S - optimal$ if P is $S - consistent$ and for any $S - consistent$ set of cuts P', $|P| \le |P'|$

Let $S^* = \{U^*, B, f^*\}$ be an information table, where U^* be the set of pairs (i, j), such that $i < j$ and $f(o_i, d) \ne f(o_j, d)$, B is the set of all basic cuts on S, f^* is a mapping function $f^*: U^* \times B \to \{0\ or\ 1\}$,

$$f^*((i,j), p_r^c) = \begin{cases} 1, & f(o_i, c) < p_r^c \le f(o_j, c)\ or\ f(o_j, c) < p_r^c \le f(o_i, c) \\ 0, & otherwise \end{cases}$$

where $c \in C$, $p_r^c \in B_c$. The set of $S - consistent$, $S - irreducible$ and $S - optimal$ cuts can be generated from S^*.

3.1 Discernibility matrix approach to discretization

Discernibility matrix is introduced by Skowron and Rauszer [15]. Let $M = \{R^*, C^*\}$ be a discernibility matrix. Rows of the matrix R^*, are the set of pairs (i, j), such that $i < j$ and $f(o_i, d) \ne f(o_j, d)$, columns of the matrix C^*, are the intervals $[v_k^c, v_{k+1}^c), \forall c \in C$ and $1 \le k < |V_c|$. Element of the matrix M is defined as,

$$f(x) = \begin{cases} 1, & x[v_k^c, v_{k+1}^c) \subseteq \left[min\left(f(o_i,c), f(o_j,c)\right), max\left(f(o_i,c), f(o_j,c)\right)\right) \\ 0, & otherwise \end{cases}$$

The discernibility function, is defined as $f(M) = \wedge\{\vee c_{ij}\}$.

The prime implicant of $f(M)$, is of the form $\left\{\left[v_{k_1}^{c_1}, v_{k_1+1}^{c_1}\right), \dots, \left[v_{k_r}^{c_r}, v_{k_r+1}^{c_r}\right)\right\}$. These prime implicants defines set of cuts of the form,

$$P = \left\{\left(c_1, \frac{\left(v_{k_1}^{c_1}, v_{k_1+1}^{c_1}\right)}{2}\right), \dots, \left(c_r, \frac{v_{k_r}^{c_r}, v_{k_r+1}^{c_r}}{2}\right)\right\}$$

P is $S - consistent$ and $S - irreducible$ set of cuts, and the minimal of all such P's is the $S - optimal$ set of cuts. Searching for $S - optimal$ set of cuts is a NP-hard problem. Efficient heuristics will help in identifying optimal set of cuts. The next section describes MD-heuristics, a heuristic based approach to find reasonable set of cuts.

3.2 MD-heuristics approach to discretization

Using MD-heuristics, the best set of cuts can be found in $O(|U||C|)$ steps, with $O(|U||C|)$ memory usage [12]. Consider the information table S^*. In this approach the column in S^*, with maximum number of 1's is added to the set of cuts, then that column is deleted from S^*, together with all rows with 1 in that column. This process is repeated till S^* is empty. The best set of cuts obtained using this approach might not be a $S - optimal$ set of cuts, it may include superfluous cuts which have to be removed. The next section describes a 2-step discretization approach to generate an $S - consistent$ and $S - irreducible$ set of cuts. This approach is based on the set of cuts returned from MD-heuristics and Genetic Algorithm.

4. 2-step Discretization approach

In this approach, all the superfluous cuts generated by MD-heuristics will be reduced using GA. 2-step Discretization approach is as shown in Figure 1.
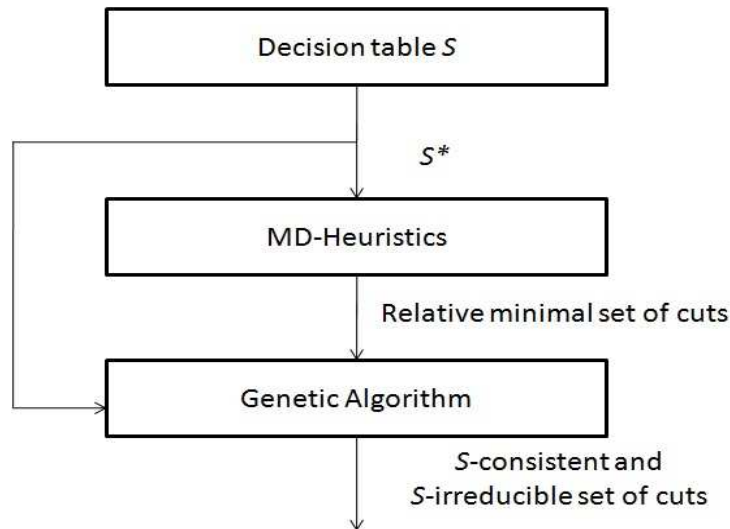
Fig.1. 2-Step discretization approach

4.1 Genetic Algorithm

GA provides a methodology to solve optimization problems. GA is motivated by biological evolution [5] and is stochastic in searching a huge search space. The proposed GA starts with the best set of cuts, P generated by MD-heuristics approach, superfluous cuts are identified in P and are discarded through GA iterations and finally $S - consistent$ and $S - irreducible$ set of cuts are generated. If there are no superfluous cuts in P, then GA will return the same set of cuts P.

4.1.1 Representation

Let P be the set of cuts returned from *MD-heuristics approach*. Each candidate set of cuts, is represented as a chromosome by a bit string of length $|P|$. A 1 in the bit string at position i represents, the cut p_i is present in the candidate set of cuts and a 0 in the bit string at position i represents the cut p_i is not present in the candidate set of cuts.

4.1.2 Initial Population

Initial population includes the chromosome representation of P, i.e., a bit string of all 1's of length $|P|$ along with randomly generated chromosomes. Size of population is set to 20.

4.1.3 Fitness Function

Fitness function is designed in such a way to discard any superfluous cuts from P, while being $S - consistent$. Consider the information system S^*. Fitness function of a candidate set of cuts P' is denoted by $f(ch_{P'})$.

$$f(ch_{P'}) = \begin{cases} \frac{|B|-|P'|}{|B|}, & \bigwedge\{\bigvee f^*((i,j),p_i)\} = 1 \ \forall p_i \in P' \\ 0, & otherwise \end{cases} \quad (1)$$

where B is the basic set of cuts and $|P'|$ is the total number of 1's in the chromosome $ch_{P'}$

4.1.4 Proof of Fitness Function

The correctness of fitness function for attribute discretization, defined in Equation 1 is proved in Theorem 1.

Theorem 1. Candidate set of cuts with maximum fitness value is $S - consistent$ and $S - irreducible$.

Proof. A chromosome will get a positive fitness value, only if set of cuts defined by the chromosome is $S - consistent$, otherwise fitness value will be zero.

Let P'_1 and P'_2 be two $S - consistent$ chromosomes with fitness values $f(ch_{P'_1})$ and $f(ch_{P'_2})$ respectively and let

$$f(ch_{P'_1}) < f(ch_{P'_2})$$

$$\Rightarrow \frac{|B| - |P'_1|}{|B|} < \frac{|B| - |P'_2|}{|B|}$$

$$\Rightarrow |P'_1| > |P'_2|$$

P'_1 has less number of cuts and greater fitness value than P'_2. Therefore the candidate with maximum fitness value will be $S - consistent$ and with no superfluous cuts in it i.e., $S - irreducible$.

4.1.5 Algorithm

The GA with the proposed fitness function is as shown in Algorithm 1. The algorithm is run with the GA parameter settings as shown in Table 1.

Table 1. GA parameter settings

Parameter name	Parameter value
Population size	20
Population type	Bit String
Creation function	Uniform
Scaling function	Rank
Selection function	Stochastic uniform

Elite count	2
Crossover fraction	0.8
Crossover function	Scattered
Mutation rate	0.01
Stopping criteria	Avg. change in fitness value $< 10^{-6}$

Algorithm 1: Algorithm to find $S - consistent$ and $S - irreducible$ set of cuts using GA

Input: Population type, population size, creation function, scaling function, selection function, elite count, crossover rate, fitness function, crossover function, mutation function, iteration count

Output: $S - consistent$ and $S - irreducible$ set of cuts

Begin

1: Search space of GA is $2^n - 1$ possible candidate set of cuts, where n is the number of cuts returned by MD-heuristics.

2: Include the set of cuts returned by MD-heuristics in Initial population.

3: Use uniform creation function and generate the rest of the initial population of given population size.

4: Evaluate fitness value, for each candidate set of cuts by using the fitness function given in Equation (1).

5: Candidate set of cuts are sorted as per fitness value. Rank is assigned to each candidate basing on its position in the sorted list.

6: Repeat steps 7-10, until the average change in the fitness value is less than 10^{-6}

7: Generate offspring's by using scattered crossover function with crossover rate as 0.8. In scattered crossover function a random binary vector is created. Offspring's are generated by taking genes from first parent where the binary vector is 1 and genes from the second parent where the binary vector is 0. Include these offspring's in the next generation.

8: Apply mutation operator to the candidates in next generation, with mutation rate of 0.01.

9: Evaluate fitness value for each candidate set of cuts of the next generation.

10: Elite count is taken as 2, so top 2 fittest candidates are guaranteed to survive in the next generation.

11: Output the candidate set of cuts, that has the maximum fitness value from the current population.

5. Experiments

The proposed algorithm is evaluated on University of California, Irvine (UCI) Machine learning repository data sets *iris, glass, wine, liver-disorders* and *ecoli [4].* Table 2 describes these *5* data sets. All these datasets are continuous. Decimal point precision of attribute domain values is taken as 2.

Table 2. Description of the Data sets

Data set	$|U|$	$|C|$	$\# classes$	$|U^*|$	$|C^*|$
Iris	150	4	3	7500	119
Glass	214	9	6	16870	755
Wine	178	13	3	15667	1262
Liver-disorders	345	6	2	29000	322
Ecoli	336	7	8	41078	356

Information table S^* is constructed from these datasets. S^* is given as input to *MD-Heuristics* approach and best set of cuts are generated. These set of cuts are further reduced using the GA with the proposed chromosome representation and fitness function. GA is implemented using MATLAB's gatool. In MATLAB's *gatool* minimization of $-f(ch_{p'})$ is used to achieve maximization of $f(ch_{p'})$. Table 3 shows the number of $S-consistent$ and $S-irreducible$ cuts, set of cuts and time taken for execution.

Table 3. Experiment results of 2-step Discretization approach

Data set	$\# cuts$	Set of cuts	Run time (in seconds)
Iris	7	$\{(c_1, 6.05), (c_2, 2.25), (c_2, 3.05), (c_3, 2.45),$ $(c_3, 4.95), (c_4, 1.65), (c_4, 1.75)\}$	2.6
Glass	17	$\{(c_2, 12.56), (c_2, 13.22), (c_2, 13.37), (c_2, 14.04),$ $(c_3, 0.16), (c_3, 3.42), (c_3, 3.5), (c_3, 3.6),$ $(c_3, 3.82), (c_4, 1.24), (c_4, 1.42), (c_5, 71.54),$ $(c_5, 72.76), (c_5, 72.9), (c_7, 8.46), (c_7, 9.25),$ $(c_9, 0.02)\}$	7.2
Wine	16	$\{(c_1, 12.56), (c_1, 13.57), (c_2, 1.79), (c_2, 3.14),$ $(c_3, 2.41), (c_4, 15.55), (c_4, 19.55), (c_4, 22.25),$ $(c_5, 86.5), (c_5, 102.5), (c_6, 2.99), (c_9, 1.5),$ $(c_{10}, 5.29), (c_{11}, 1.06), (c_{12}, 2.95), (c_{13}, 673.5)\}$	3.6
Liver-disorders	20	$\{(c_1, 84.5), (c_1, 87.5), (c_1, 90.5), (c_1, 93.5),$ $(c_2, 57.5), (c_2, 65.5), (c_2, 74.5), (c_2, 82.5),$ $(c_2, 93.5), (c_3, 17.5), (c_3, 22.5), (c_3, 35.5),$ $(c_4, 18.5), (c_4, 21.5), (c_4, 24.5), (c_4, 44.0),$ $(c_5, 22.5), (c_5, 35.5), (c_5, 56.5), (c_6, 3.5)\}$	6.5
Ecoli	27	$\{(c_1, 0.18), (c_1, 0.38), (c_1, 0.46), (c_1, 0.5),$ $(c_1, 0.62), (c_1, 0.66), (c_1, 0.72), (c_1, 0.8),$	9.7

| | | $(c_2, 0.29), (c_2, 0.4), (c_2, 0.44), (c_2, 0.52),$
 $(c_2, 0.58), (c_2, 0.77), (c_2, 0.79), (c_5, 0.38),$
 $(c_5, 0.44), (c_5, 0.5), (c_5, 0.52), (c_5, 0.54),$
 $(c_5, 0.6), (c_5, 0.64), (c_6, 0.48), (c_6, 0.74),$
 $(c_6, 0.74), (c_7, 0.36), (c_7, 0.46)\}$ | |

2-step Discretization approach can be compared with the discretization approach described in [1]. Discretization in [1] is also done using GA with optimization strategies like elitism and penalty. Comparison of discretization execution times is shown in Table 4. Comparison results show that the proposed *2-step discretization approach* finds $S - consistent$ and $S - irreducible$ set of cuts in reduced time.

Table 4. Comparison of Run times

Data set	*2-step discretization* (in seconds)	Discretization approach described in [1] (in minutes)
Iris	2.6	2
Glass	7.2	5
Wine	3.6	4

Conclusion

In this paper we have proposed a *2-step discretization approach* to discretize continuous datasets. Discretization is achieved based on discernibility relations of Rough Set Theory. Instead of directly considering all the possible basic set of cuts for optimization using *Genetic Algorithm*, in *2-step discretization approach* we have taken the set of cuts returned from *MD-heuristics approach*, and have reduced the superfluous cuts among them, using GA. GA is applied by designing chromosome and fitness function that preserve object discernibility. Experimental results on UCI Machine Learning Repository datasets shows that the entire discretization process is completed in reduced time when compared to the discretization approach described in [1].

Acknowledgment

References

[1] C. Y. Chen, Z. G. Li, S. Y. Qiao and S. P. Wen, "Study on Discretization in Rough Set Based on Genetic Algorithm", Proceeding of the Second International Conference on Machine Learning and Cybernetics, Xi'an, China, pp. 1430-1434, 2003

[2] M.R. Chmielewski and J.W. Grzymala-Busse, "Global discretization of continuous attributes as preprocessing for machine learning", presented at Int. J. Approx. Reasoning, 1996, pp.319-331.

[3] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and Unsupervised Discretization of Continuous Features", in Proc. ICML, 1995, pp.194-202.

[4] A.Asuncion and D.Newman, UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/ MLRepository.html], Irvine, CA: University of California, Department of Information and Computer Science, 2010

[5] David E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley, 1989.

[6] L.A. Kurgan and K.J. Cios, "CAIM Discretization Algorithm", presented at IEEE Trans. Knowl. Data Eng., 2004, pp.145-153.

[7] Nguyen H.S. Nguyen, "Discretization Problem for Rough Sets Methods", in Proc. Rough Sets and Current Trends in Computing, 1998, pp.545-552.

[8] Nguyen, H.S., Nguyen, S.H.: Rough Sets in Knowledge Discovery, vol. 1, chap.
Discretization Methods in Data Mining, pp. 451{482. Physica Verlag, Heidelberg
(1998)

[9] H.S. Nguyen and A. Skowron, "Boolean Reasoning for Feature Extraction Problems", in Proc. ISMIS, 1997, pp.117-126.

[10] Nguyen H.S., Skowron A. Quantization of real values attributes, rough set and Boolean reasoning approaches: proc. of the 2nd Joint Annual Conference on Information Science[C], Wrightsville Beach:NC,1995:34-37.

[11] Han Zhenxiang, Zhang Qi and Wen Fushuan, A Survey on Rough Set Theory and Its Application, control theory and applications, vol. 16, pp. 153-157, Feb. 1999.

[12] Z. Pawlak, and A. Skowron, "Rough Sets and Boolean Reasoning," Information Sciences, vol. 177, pp. 41-73, 2007.

[13] Z. Pawlak, A. Skowron, Rough sets: some extensions, Information Sciences, 177(2007) 28-40.

[14] Z. Pawlak, A. Skowron, Rudiments of rough sets, Information Sciences, 177(2007) 3-27.

[15] A. Skowron, C. Rauszer, The discernibility matrices and functions in information systems, in: R. Slowinski (Ed.), Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory, Khlwer Academic Publishers, Dordrecht, 1992: 331-362.

[16] Roman W. Swiniarski and Andrzej Skowron, "Rough set methods in feature selection and recognition", Pattern Recognition Letters 24, 2003, pp.833-849.

[17] L.A. Pawlak, "Rough Sets," Communications of the ACM, vol.38, no.11, pp.89-135,Nov. 1995.