Fundamenta Informaticae 34 (2014) 1–10 1
IOS Press

Ensembles of classifiers based on approximate reducts*

Jakub Wróblewski
Polish-Japanese Institute of Information Technology
and
Institute of Mathematics, Warsaw University
Banacha 2, 02-097 Warsaw, Poland
e-mail: jakubw@jakubw.pl
http://www.jakubw.pl/about/

Abstract. The problem of improving rough set based expert systems by modifying a notion of reduct is discussed. The notion of approximate reduct is introduced, as well as some proposals of quality measure for such a reduct. The complete classifying system based on approximate reducts is presented and discussed. It is proved that the problem of finding optimal set of classifying agents based on approximate reducts is NP-hard; the genetic algorithm is applied to find the suboptimal set. Experimental results show that the classifying system is effective and relatively fast.

Keywords: Rough sets, approximate reduct, voting.

1. Introduction

Rough set expert systems are based on the notion of a *reduct* [13], [14] – a minimal subset of attributes which is sufficient to discern between objects with different decision values. A set of short reducts can be used to generate rules [2]. The problem of the minimum reducts generation (i.e. reducts with small number of attributes) is NP-hard, but approximate algorithms (like the genetic one described in [16], [10] and implemented successfully in [12]) can be used to obtain almost minimal reducts in reasonable time. On the other hand, rules generated by reducts are

*This work was supported by grant 8T11C02519 of Polish State Committee for Scientific Research. "Primary tumor" and "breast cancer" domains were obtained from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia. Thanks go to M. Zwitter and M. Soklic for providing the data.

often too specific and cannot classify new objects. Another types of reducts have been considered to improve efficiency on new objects (e.g. dynamic reducts [3], reducts optimized by number of generated rules [19], reducts based on variable precision models [23], ε-approximated reducts [15]). One of the methods is to calculate reducts basing on single objects (local reducts, [19]); results are good, but calculation time may be long, even when parallel algorithm is used [18].

In our approach classification results are improved by selecting optimal set (ensemble [5]) of classifying rough set based agents. This is a special case of the *wrapper approach* to the problem of feature selection [22], [4]: an optimal subset of features (approximate reduct) is found for each agent, then an optimal subset of subsets (ensemble) of features is selected.

The notions of *approximate reduct* and *classifying agent* are introduced in the next sections. The methods for the fast approximate reducts generation and evolutionary process (based on genetic algorithm) of the expert system tuning are presented. The main result of the paper is Theorem 3.1 (problem of the optimal selection of classifying agents based on approximate reducts is NP-hard), and experimental results presented in Section 4.

The paper is an extended and revised version of [21] presented on CS&P Workshop, Berlin 2000.

2. Approximate reducts

Let $\mathbb{A} = (U, A \cup \{d\})$ be an *information system* (see [14]), where U – set of objects, A – set of attributes, d – decision; every $a_i \in A$ is a function $a_i : U \to V_i$; $d : U \to V_d$.

Every subset of attributes $R \subset A$ defines indiscernibility relation [13] on U, i.e. two objects are in relation iff values of attributes from R are equal. Hence, R defines a partition of U. Every equivalence class $[u]_R$ of this relation generates one (generalized) decision rule:

$$r_i = (a_{i_1} = v_{i_1} \wedge ... \wedge a_{i_{|R|}} = v_{i_{|R|}} \Rightarrow d = (\mu_{i_1}, ..., \mu_{i_n}))$$

where $A = \{a_1, ..., a_{|A|}\}$ is the set of attributes, n is a number of decision classes $D_1, ..., D_n \subseteq U$, $a_{i_k}(u) = v_{i_k} \in V_k$, $(\mu_{i_1}, ..., \mu_{i_n})$ is a probability distribution of decision value, based on the rough membership function of the equivalence class in decision class: $\mu_{i_j} = \frac{|[u]_R \cap D_j|}{|[u]_R|} \in [0, 1]$.

When one have to take into account only one decision value, the maximal one can be chosen:

$$r_i = (a_{i_1} = v_{i_1} \wedge ... \wedge a_{i_{|R|}} = v_{i_{|R|}} \Rightarrow d = d_j)$$

where j is such that $\mu_{i_j} = \max(\mu_{i_1}, ..., \mu_{i_n})$.

Constructing a data model (set of rules) is always a tradeoff between model generality and accuracy. To maintain the balance between these two aspects, the classification algorithm should be parameterized and optimal values of parameters should be found for a given data set. Optimization process should be based on a quality measure function estimating both generality and accuracy of the model.

The classification algorithm described in this paper is parameterized using a parameter α: for a given value $\alpha \in [0, 1]$ we will remove (filter out) rules for which $\mu_{i_j} < \alpha$. Let $Rul_{R,\alpha} = \{r_i :$

$\max(\mu_{i_1}, ..., \mu_{i_n}) \geq \alpha\}$ denotes a set of decision rules based on $R \subseteq A$, filtered using a constant α.

Definition 2.1. Let a method of generating a classification algorithm (CA based on $Rul_{R,\alpha}$) for any set of attributes be given. **Quality measure** Q of a subset of attributes A for a given testing decision table $\mathbb{A}$ is a function $Q_{\mathbb{A}} : 2^A \to \mathbb{R}$ (we will omit $\mathbb{A}$ in subscript) such that:

1. Values of $Q(R)$ depends only on CA and testing table $\mathbb{A}$.
2. Let $A_1 \subset A$, $A_2 \subset A$ such that any object recognized properly by CA based on A_1 is recognized properly by CA based on A_2 too, and that any object recognized by any of these CA is recognized properly. Then quality measure Q should satisfy: $Q(A_1) \leq Q(A_2)$ (i.e. Q is monotonous due to a set of objects classified properly).

The quality measure should take into account two aspects: a degree the subset is a reduct and its ability to generate good rule sets. In general, there is no limitation on form of the function Q. Some practical examples are presented below.

Definition 2.2. An **approximate reduct** with respect to a quality measure function Q and parameter value $\alpha \in [0, 1]$ is a subset $R \subseteq A$ such that:

a) $Q(R)$ is based on classification algorithm containing a set of rules $Rul_{R,\alpha}$

b) $\forall\ R' \subset R,\ R' \neq R,\ \ Q(R') < Q(R)$

c) $\forall\ R'' \supset R,\ R'' \neq R,\ Q(R'') \leq Q(R)$

i.e. R is a local maximum of measure function Q.

Unfortunately, optimal parameter α can be found only by experiments; for some testing databases the best results were obtained for $\alpha = 0$, whereas in another case for $\alpha = 0.9$.

Consider the following quality measure:

$Q(R) = |A| - |R|$, when R is a reduct; $Q(R) = 0$ otherwise.

It is easy to see, that for the above quality measure, definition 2.2 is equivalent of a definition of classical reduct; moreover, the quality values of short reducts are higher than those of long ones. Unfortunately, the systems based on classical short reducts are often too specific (there are many not recognized objects).

To obtain more effective set of rules we use another quality measure, originally used to evaluate new features in databases [20]:

Definition 2.3. Predictive quality measure of subsets R of attributes is defined as:

$$Q_{\mathbb{A}}(R) = \sqrt[n]{\prod_{i=1}^{n} \frac{P(\{r_1...r_k\}, \mathbb{A}, i)}{P(\{r_1...r_k\}, \mathbb{A}, i) + N(\{r_1...r_k\}, \mathbb{A}, i)}} \times P_{cov} \tag{1}$$

$$P_{cov} = 1 - \prod_{i=1}^{k} (1 - \frac{n(r_i) - 1}{|U| - 1}) \tag{2}$$

where $\mathbb{A} = (U, A, d)$ – training decision table, k – number of rules generated by R, $n(r_i)$ – number of training objects classified properly by rule r_i, $P(\{r_1...r_k\}, \mathbb{A}, i)$ – number of properly classified objects belonging to i-th decision class, $N(\{r_1...r_k\}, \mathbb{A}, i)$ – number of objects belonging to i-th decision class but classified to another one, n – number of decision classes.

Experiments show, that Q provides good estimation of final classification algorithm quality on a real data. Approximate reducts optimized by Q have better forecasting capabilities than e.g. short ones. One uses short reducts because rules based on few attributes should be more general than others (minimum description length principle). On the other hand, predictive quality measure estimates an effectiveness of expert system on unknown testing data table, so reducts (and respective sets of rules) are optimized "more directly" [8].

A connection between approximate reducts and classical ones is given by the following fact:

Theorem 2.1. *If R is a reduct (in classical sense – see [13], [14]), then R satisfies condition c) in approximate reduct definition (with respect to predictive quality measure).*

We have used a simple heuristics to generate approximate reducts. First, a random permutation σ of attributes is generated. Then, according to this permutation, attributes are added to a subset R and its quality is calculated. Typically, quality value is low for small subsets, and increases when the next attribute is added. When quality starts to decrease, next phase begins. Each attribute from R is replaced one by one and a quality measure is calculated – an attribute causing the highest quality increase is replaced from R. The algorithm stops when local optimum is achieved (the result is not always an approximate reduct; the algorithm just approximates it). If two subsets have the same quality measure, the shorter one (by means of number of attributes) is taken (a corollary of definition 2.2).

Note, that ordering σ of attributes together with R generates an ordering τ of set of objects U: objects are sorted by attributes' values. Two additional techniques were used to improve final classification quality. First, a generalization method for rules was used: every three adjacent (by means of objects ordering τ) rules was analyzed and some of them was joined (generalized) if system quality was higher after this operation. Second, a method of unrecognized objects classification was introduced: when a new object does not math any rule but we have to classify it anyway[1], we can use *upper approximation* [13] of rule set. In this case we find the closest rules (by means of objects ordering τ) and use them to determine a decision value. Hence, the complete classifying system based on an approximate reduct R should contain not only a set of rules generated by R, but also the ordering τ.

Definition 2.4. By **classifying agent based on approximate reduct** we will denote a triple $(Rul, \mathbb{A}, \tau)$, where Rul is a set of decision rules generated by R, based on values of attributes of a training information system $\mathbb{A}$, τ – a permutation of objects of $\mathbb{A}$.

[1] in some applications, e.g. vehicle control, it is crucial to classify all objects, even suboptimally.

3. Optimal set of classifying agents

Some other reduct-based systems [12] generate several well-optimized reducts (e.g. using genetic algorithm) and use all of them to create rule set. In some approaches [1] the rule set is then filtered (e.g. by genetic algorithm), which is very time-consuming due to large number of rules in real-life data. Our strategy is different: generate many reducts (classifying agents) using fast approximation heuristics (these reducts may not be optimal), then construct a classification system by selecting optimal subset of them. If some agents will be worse than others, they simply will not be used in the final system. On the other hand, even very poor (when evaluated separately) agent may become valuable e.g. because it can classify some objects which are hard to recognize by other agents in a team.

We will optimize a set of agents due to classification results on a testing set. Let T denote a set of agents (based on approximate reducts), $P(T, \mathbb{A})$ – a set of properly classified objects from a testing table $\mathbb{A}$, $N(T, \mathbb{A})$ – a set of misclassified objects from $\mathbb{A}$. By a **team quality function** S we will denote any function such that for any $T_1, T_2, \mathbb{A}$:

$$P(T_1, \mathbb{A}) > P(T_2, \mathbb{A}) \ \wedge \ N(T_1, \mathbb{A}) = N(T_2, \mathbb{A}) \ \Rightarrow \ S(T_1) > S(T_2) \tag{3}$$

$$P(T_1, \mathbb{A}) = P(T_2, \mathbb{A}) \ \wedge \ N(T_1, \mathbb{A}) = N(T_2, \mathbb{A}) \ \Rightarrow \ (S(T_1) < S(T_2) \Longleftrightarrow |T_1| > |T_2|) \tag{4}$$

A voting method Φ is used if classification results are different for different agents. Let $v_1, ..., v_k$ be a set of agents' classification results for an object, $v_i \in V_d \cup \{\emptyset\}$. We will assume the following property of Φ:

$$(\forall_i \ v_i = v \vee v_i = \emptyset) \wedge (\exists_i \ v_i = v) \implies \Phi(v_1, ..., v_k) = v \tag{5}$$

Theorem 3.1. *Let $\alpha > 0$, reduct quality Q and team quality S measures be given. Problem of selecting S-optimal subset of classifying agents based on approximate reducts is NP-hard.*

Proof:
We will show, that any minimal row covering problem for binary matrices (MATRIXCOVER, see [6]) can be solved by selecting optimal subset of classifying agents in a case of special data table. Let $\mathbf{B} = \{b_{ij}\}$ be a binary matrix of size $n \times m$. Our goal is to find a minimal set of columns such that in every row there is at least one "1" in a selected column. Without loss of generality we will assume that the matrix is large enough, i.e.:

$$m > \frac{2}{\alpha} - 1 \tag{6}$$

A special information system $\mathbb{A} = (U, A, d)$ is constructed in the following way. Every row in matrix $\mathbf{B}$ corresponds to a pair of objects in U; an additional set of $2m^2$ objects is used. Every column in matrix $\mathbf{B}$ corresponds to one attribute in $\mathbb{A}$. So $|A| = n$, $|U| = 2m(1 + m)$. Decision is a column of values $V_d = \{0, ..., m - 1\}$. Attributes and decision values for $\mathbb{A}$, for the first $2m$ objects:

$a_i(u_{2j-1}) = b_{ij}(j+1)$,
$a_i(u_{2j}) = b_{ij}j + 1$,
$d(u_{2j}) = d(u_{2j-1}) = j - 1$, where $j = 1...m$.

Attributes and decision values for $\mathbb{A}$, for the next $2m^2$ objects:

$a_i(u_{2m+2j-1}) = 0$,
$a_i(u_{2m+2j}) = 1$,
$d(u_{2m+2j}) = d(u_{2m+2j-1}) = (j-1)\ mod\ m$, where $j = 1...m^2$ (see example – Table 1.).

1	0	**1**	0
1	0	0	**1**
0	0	**1**	0
0	0	**1**	**1**

$\longrightarrow$

a_1	a_2	a_3	a_4	d
2	0	**2**	0	0
2	1	**2**	1	0
3	0	0	**3**	1
3	1	1	**3**	1
0	0	**4**	0	2
1	1	**4**	1	2
0	0	**5**	**5**	3
1	1	**5**	**5**	3
0	0	0	0	0
1	1	1	1	0
	⋮			⋮
0	0	0	0	3
1	1	1	1	3
	⋮			⋮
0	0	0	0	0
1	1	1	1	0
	⋮			⋮
0	0	0	0	3
1	1	1	1	3

Table 1. Matrix **B** and information system $\mathbb{A}$.

We will prove, that the set of approximate reducts (for a given parameter value α) is composed of sets of the form $R_k = \{a_k\}$. Consider R_1 in the example presented above – it generates the following rules:

$$r_1 = (a_1 = 2 \Rightarrow d = 0)$$
$$r_2 = (a_1 = 3 \Rightarrow d = 1)$$
$$r_3 = \left(a_1 = 0 \Rightarrow d = \left\{\tfrac{1}{6}, \tfrac{1}{6}, \tfrac{2}{6}, \tfrac{2}{6}\right\}\right)$$
$$r_4 = \left(a_1 = 1 \Rightarrow d = \left\{\tfrac{1}{6}, \tfrac{1}{6}, \tfrac{2}{6}, \tfrac{2}{6}\right\}\right)$$

Rules r_3 and r_4 will be removed because a support of the most supported decision class is lower than α (by assumption 6: $\alpha > \frac{2}{5}$). This is a general rule for R_k family: only rules corresponding to $b_{ij} = 1$ will be taken into account. Any rule with attribute value 0 or 1 will be filtered out due to a special form of the second part of decision table: every decision class is supported by at least m objects matching the rule, thus at least $(m-1)m$ objects support decision classes other than the best one. On the other hand, at most $m + m^2$ objects match the rule, thus a value of the rule's support can be upper-bounded by:

$$1 - \frac{(m-1)m}{m(m+1)} = \frac{2}{m+1} < \alpha$$

because, by assumption 6, $m > \frac{2}{\alpha} - 1$. Hence all rules corresponding to $b_{ij} = 0$ are filtered out.

Let us consider any subset R being an extension of R_i. Note that the set of decision rules based on R is an extension of set of rules based on R_i: some rules are not changed (in a sense of a set of supporting objects), other are divided into two or more subrules. Namely, every rule corresponding to $b_{kj} = 0$ (where j is a number of attribute present in R, $j \neq i$) will be divided into two rules, both with support 1. E.g. let $R = \{a_1, a_2\}$ for the information system $\mathbb{A}$ presented in table 1; in this case a rule based on R_1:

$$r_1 = (a_1 = 2 \Rightarrow d = 0)$$

will be divided into:

$$r_1^1 = (a_1 = 2 \wedge a_2 = 0 \Rightarrow d = 1)$$
$$r_1^2 = (a_1 = 2 \wedge a_2 = 1 \Rightarrow d = 1)$$

These rules will be filtered out before quality measure $Q(R)$ is calculated. In general, given a set of rules based on $R = \{a_{i_1}, ..., a_{i_t}\}$, the support of these rules is larger than one only when $b_{i_1 j} = ... = b_{i_t j} = 1$. A set of rules (filtered) based on R is then an intersection of sets of rules based on R_{i_1}, ... , R_{i_t}, hence (as all these rules are deterministic) $Q(R) \leq Q(R_{i_1})$, ... , $Q(R) \leq Q(R_{i_t})$. But, in this case, we will rather prefer a set containing one attribute (corollary of definition 2.2). Thus, the set of approximate reducts for this information system contains sets R_1,..., R_n only.

Let $T = \{R_{i_1}, ..., R_{i_t}\}$ be an arbitrary set of one-column reducts, let $B(T)$ be a set of columns $\{i_1, ..., i_t\}$ of $\mathbf{B}$. Now we will prove, that optimal set of reducts (agents) $\hat{T} \subseteq \{R_1, ..., R_n\}$ (for any quality measure S based on classification results on $\mathbb{A}$, satisfying conditions 3 and 4) corresponds to the minimal set of columns covering $\mathbf{B}$. Note, that for any T a set of rules based on it will contain no rules for attributes' values 0 or 1 (because they are filtered out – see above). Therefore no objects from the second part of $\mathbb{A}$ (object number $2m + 1$ and above) will be recognized properly. On the other hand, for $j \leq 2m$:

$$o_j \in P(T, \mathbb{A}) \Longleftrightarrow \exists_{R_i \in T}\, b_{ij} = 1$$

where $P(T, \mathbb{A})$ is a set of objects properly classified by set of agents T. This result does not depend on voting method (assuming 5), because any agent either assigns proper decision class to an object, or does not recognize it at all. Hence:

$$P(T, \mathbb{A}) = \{o_1, ..., o_{2m}\} \Longleftrightarrow B(T) \textit{ is a covering of } \mathbf{B}$$

Now let T_1 and T_2 will be two subsets of $\{R_1, ..., R_n\}$ (classifying agents), corresponding to two coverings of $\mathbf{B}$. As shown above, both of them recognizes properly the same set of $2m$ objects from $\mathbb{A}$. By assumption 4 we have:

$$S(T_1) < S(T_2) \Longleftrightarrow |T_1| > |T_2|$$

hence the optimal $\hat{T}$ is a subset of minimal cardinality, which corresponds to minimal covering $B(\hat{T})$ of matrix $\mathbf{B}$.

It was shown, that for any binary matrix $\mathbf{B}$ of size $n \times m$ we can construct an information system and a set of classifying agents (approximate reducts with parameter value α) such that finding an optimal set of agents gives a minimal column covering of $\mathbf{B}$. This construction is polynomial since $\mathbb{A}$ has $O(n)$ columns and $O(m^2)$ objects (assuming $m > \frac{2}{\alpha} - 1$). As MATRIX-COVER is NP-hard, the problem of finding optimal set of agents is NP-hard too. □

4. Results of experiments

Due to its NP-hardness, the problem of selection of optimal set of agents cannot be solved exactly in reasonable time. A genetic algorithm [7] was used in our experiments to choose the best team of agents in approximate way. Chromosomes (sets of agents) were evaluated by their effectiveness on a separate testing data set (selected randomly at the beginning of training process). Such a classification process is often very time-consuming (in general, time is proportional to [number of rules]×[number of testing objects]), but in our case, because of reduct-based rule generation, a fast $O(|A| \times |U| \log |U|)$ testing algorithm can be used:

1. Let R_i be an approximate reduct, U – a set of training objects, U_1 – a set of testing objects.
2. Sort U and U_1 according to R_i.
3. For each $u_j \in U_1$ find a proper decision rule r_k by selecting a set of objects in U identical on R_i with u_j.
4. Repeat from 2. with the next reduct R_{i+1}.
5. Calculate final decision values for objects from U_1 using a voting technique.

Step 3. can be done in linear time, because any testing object u_j will match either the same rule r_k than u_{j-1} did, or a rule r_l, $l > k$. Thus the complexity of the algorithm is dominated by step 2. complexity $O(|A| \times |U| \log |U|)$, supposing $|U| \approx |U_1|$.

Results of experiments described by author in [15] suggest, that dependence between a number of agents and classification results is irregular: when a set of agents is large enough, new agents hardly increase (or even decrease) classification rate. The "critical size" of team is about 20-25 agents for medium size benchmark databases. Thus a set of 60 agents was used in our current experiments, whereas typically only 20-30 reducts out of initial number of 60 were used in the final classification algorithm.

Several well known benchmark databases published in StatLog project [9] were used in experiments. Table 2. presents results, including data size, calculation time and average error rate. All parameters of the algorithm were chosen experimentally: 60 reducts have been found, then the reducts were filtered by genetic algorithm (population: 60, evolution steps: about 300) and used in classification algorithm.

Data	Size (obj.×attrib.)	Time (s)	Error	StatLog rank
sat image	4435×37	223.0	0.129	5
letter	15000×17	1310.0	0.086	4
diabetes	768×9	5.5	0.267	12
breast cancer	286×10	2.3	0.272	–
primary tumor	339×18	25.5	0.606	–
Australian credit	690×15	5.9	0.137	2
vehicle	846×19	16.1	0.319	19
DNA splices	2000×181	47.0	0.047	2

Table 2. Experimental results(Celeron 400 MHz).

Calculation time includes classificator generation and testing on test table. If database does not contain separate test table, cross-validation method is used. Results are in many cases better than those obtained by classical methods (C4.5, k-NN, neural nets; the last column shows the rank of our method in StatLog [9] experiment, comparing with 24 other methods). This is worth noting that error rates presented in table are average of 10 experiments; in several experiments (due to their nondeterministic nature) results were significantly better – e.g. 0.126 for "sat image" data, 0.525 for "primary tumor" data, 0.125 for "Australian credit" data.

Relatively long calculation time for "letter" database is concerned with high number (26) of decision classes rather than with number of objects. We did not use databases smaller than about 200 objects because of low stability of results: about 25% of training objects are used as internal testing sample; when this sample is too small, genetic algorithm cannot optimize the set of agents well enough.

5. Conclusions

A classification system based on approximate reducts was presented. As shown in Section 3, problem of the optimal classifying agents selection is NP-hard, so the only way to construct such a set effectively is to use approximate adaptive technique (e.g. genetic algorithm based on system performance on testing data). The system described above proved to be effective and relatively fast on several benchmark data sets.

Experiments with various voting techniques as well as on incorporating voting parameters into genetic algorithm are in progress.

References

[1] Ågontes T., Komorowski J., Løken T.: "Taming Large Rule Models in Rough Set Approaches". *Proc. of PKDD'99, Prague, Czech Republic*. Springer-Verlag (LNAI 1704), Berlin Heidelberg 1999, 193–203.

[2] Bazan J., Skowron A., Synak P.: "Dynamic reducts as a tool for extracting laws from decision tables". *Proc. of the Symp. on Methodologies for Intelligent Systems, Charlotte, NC, October 16-19, 1994*, Lecture Notes in Artificial Intelligence 869, Springer-Verlag, Berlin 1994, 346–355, also in: *ICS Research Report* 43/94, Warsaw University of Technology.

[3] Bazan J.: "A Comparison of Dynamic and non-Dynamic Rough Set Methods for Extracting Laws from Decision Tables". L. Polkowski, A. Skowron (eds.). *Rough Sets in Knowledge Discovery*. Physica Verlag, 1998.
[4] Cios K.J., Pedrycz W., Swiniarski R.W.: *Data Mining Methods for Knowledge Discovery*. Kluwer Academic Press, Boston, 1998.
[5] Dietterich T.: "Machine learning research: four current directions". *AI Magazine*, vol. **18** nr 4, 1997, 97–136.
[6] Garey M. R., Johnson D. S.: *Computers and Intractability, a Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, San Francisco, 1979.
[7] Goldberg D.E.: *GA in Search, Optimisation, and Machine Learning*. Addison-Wesley, 1989.
[8] Liu H., Motoda H.: *Feature selection for knowledge discovery and data mining*. Kluwer, Dordrecht, 1998.
[9] Michie D., Spiegelhalter D.J., Taylor C.C. (ed.): *Machine Learning, Neural and Statistical Classification*. Ellis Horwood Limited, 1994. Data avaliable at: http://www.ics.uci.edu/~mlearn/MLRepository.html.
[10] Nguyen S. H., Skowron A., Synak P., Wróblewski J.: "Knowledge Discovery in Databases: Rough Set Approach". *Proc. of The Seventh International Fuzzy Systems Association World Congress*, IFSA97, Prague, Czech Republic, 1997, vol. II, 204–209.
[11] Nguyen H. S., Nguyen S. H.: "Discretization Methods in Data Mining". L. Polkowski, A. Skowron (eds.). *Rough Sets in Knowledge Discovery*. Physica Verlag, 1998.
[12] Øhrn A., Komorowski J.: "Rosetta – A rough set toolkit for analysis of data". *Proc. of Third International Join Conference on Information Sciences (JCIS97)*, Durham, NC, USA, March 1 - 5, 3, 1997, 403–407.
[13] Pawlak Z.: *Rough sets: Theoretical aspects of reasoning about data*. Kluwer, Dordrecht 1991.
[14] Skowron A., Rauszer C.: "The Discernibility Matrices and Functions in Information Systems". R. Slowiński (ed.): *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory*. Kluwer, Dordrecht 1992, pp: 331–362.
[15] Ślęzak D., Wróblewski J., "Application of normalized decision measures to the new case classification". *Proc. of RSCTC-2000, Banff, Canada*, University of Regina, Regina, Saskatchewan 2000, 515–522.
[16] Wróblewski J.: "Finding minimal reducts using genetic algorithms". *Proc. of the Second Annual Join Conference on Information Sciences, September 28-October 1, 1995*, Wrightsville Beach, NC, 1995, 186–189. Also in: *ICS Research report 16/95*, Warsaw University of Technology.
[17] Wróblewski J.: "Theoretical Foundations of Order-Based Genetic Algorithms". *Fundamenta Informaticae*, **28 (3, 4)**, IOS Press, 1996, 423–430.
[18] Wróblewski J.: "A Parallel Algorithm for Knowledge Discovery System". *Proc. of PARELEC'98, Bialystok, Poland*. The Press Syndicate of the Technical University of Bialystok 1998, 228–230.
[19] Wróblewski J.: "Covering with reducts – a fast algorithm for rule generation". *Proc. of RSCTC'98, Warsaw, Poland*. Springer-Verlag (LNAI 1424), Berlin, Heidelberg 1998, 402–407.
[20] Wróblewski J.: "Analyzing relational databases using rough set based methods". *Proc. of IPMU 2000, Madrid, Spain*. Universidad Politecnica de Madrid, 2000, vol. 1, 256–262.
[21] Wróblewski J.: "Ensembles of classifiers based on approximate reducts". *Proc. of CS&P 2000 Workshop*, Informatik-Bericht Nr. 140, Humboldt-Universität zu Berlin, 2000, vol. 2, 355–362.
[22] Yang J., Honavar V.: "Feature subset selection using a genetic algorithm". *Proc. of the Second Annual Conference on Genetic Programming*, Morgan Kaufmann, 1997.
[23] Ziarko, W.: "Variable Precision Rough Set Model". *Journal of Computer and System Sciences* **40**, 1993, 39–59.