Data Decomposition and Decision Rule Joining for Classification of Data with Missing Values

Rafał Latkowski[1] and Michał Mikołajczyk[2]

[1] Warsaw University, Institute of Computer Science,
ul. Banacha 2, 02-097 Warszawa, Poland,
R.Latkowski@mimuw.edu.pl
[2] Warsaw University, Institute of Mathematics,
ul. Banacha 2, 02-097 Warszawa, Poland,
M.Mikolajczyk@mimuw.edu.pl

Abstract. In this paper we present a new approach to handling incomplete information and classifier complexity reduction. We describe a method, called D^3RJ, that performs data decomposition and decision rule joining to avoid the necessity of reasoning with missing attribute values. In the consequence more complex reasoning process is needed than in the case of known algorithms for induction of decision rules. The original incomplete data table is decomposed into sub-tables without missing values. Next, methods for induction of decision rules are applied to these sets. Finally, an algorithm for decision rule joining is used to obtain the final rule set from partial rule sets. Using D^3RJ method it is possible to obtain smaller set of rules and next better classification accuracy than standard decision rule induction methods. We provide an empirical evaluation of the D^3RJ method accuracy and model size on data with missing values of natural origin.

1 Introduction

Rough Set theory, proposed by Pawlak in 1982, creates a framework for handling the imprecise and incomplete data in information systems. However, in standard formalization it is not addressed to the problem of missing attribute values. Some methods for reasoning with missing attribute values were proposed by Grzymała-Busse, Stefanowski, Skowron, Słowiński, Kryszkiewicz and many others. Current findings on Granular Computing, Approximated Reasoning Schemes and Rough-Mereology (see, e.g., [9]) inspired research on new methods for handling incomplete information as well as better understanding of classifier and knowledge description complexity. In this paper we describe two of issues: reasoning under missing attribute values and reduction of induced concept description. A concatenation of solutions for problems related to these issues results in high quality classifier induction method, called D³RJ.

The D^3RJ method is based on data decomposition and decision rule joining. The aim of this method is to avoid the necessity of reasoning with missing attribute values and to achieve better classification accuracy at the reduced

classification time. The D³RJ method is based on more complex reasoning process, comparing the case of typical algorithms for induction of decision rules. The original incomplete data table is decomposed into data sub-tables without missing values. This is done using total templates that represent information granules describing the resulting data subset. Next, methods for induction of decision rules are applied to these sets. The standard decision rule induction methods are used here. In this way the knowledge hidden in data is extracted and synthesized in form of decision rules, that can also be perceived as information granules. Finally, an algorithm for decision rule joining is used to obtain classifier consisting of generalized rules built from previously induced decision rules. This final phase realizes an additional step of knowledge synthesization and can be perceived as transformation of simpler granules into the more complex ones. The D³RJ method makes is possible to obtain smaller set of rules and to achieve better classification accuracy than standard decision rule induction methods known from literature.

In the following section we introduce some necessary formal concepts. In Section 3 overview of the D³RJ method is provided. Section 4 describes the data decomposition phase. Next, the description of rule induction is provided. Section 6 describes the decision rule joining. Section 7 contain the empirical evaluation of the D³RJ method. The final section presents some conclusions and remarks.

2   Preliminaries

2.1   Decision Tables

For the classification and the concept approximation problems we consider data represented in *information systems* called also *information tables* due to its natural tabular representation (see, e.g., [10, 3]). A *decision system* (*decision table*) is an information system with a distinguished attribute called decision (see, e.g., [10, 3]). The standard definition assumes that all objects have a complete description. In a real world data, however, frequently not all attribute values are known. To be able to deal with missing attribute values we have to extend the definition of a decision table.

Definition 1. *A decision table with missing attribute values* $\mathbb{A} = (U, A, \{d\})$ *is a pair, where U is a non-empty finite set of objects called the universe and A is a non-empty set of attributes such that $a_i \in A$, $a_i : U \to V_i^*$, where $V_i^* = V_i \cup \{*\}$ and $* \notin V_i$, are conditional attributes and $d : U \to V_d$ is a special attribute called decision.*

The special symbol "$*$" denotes absence of the regular attribute value and if $a_i(x) = *$ we say that a_i is not defined on x. If all attribute values are known, the definition of the decision table with missing attribute values is equivalent to the definition of the decision table. From now on we will call decision tables with missing attribute values just decision tables, for short.

2.2 Total Templates

To discover knowledge hidden in data we should search for patterns of regularities in decision tables. A standard tool for describing a data regularities are *templates* (cf. [7, 8]). The concept of template requires some modifications to be applicable in the incomplete decision table decomposition.

Definition 2. *Let* $\mathbb{A} = (U, A, \{d\})$ *be a decision table and let* $a_i \neq *$ *be a* total descriptor. *An object* $u \in U$ *satisfies a total descriptor* $a_i \neq *$, *if the value of the attribute* $a_i \in A$ *on this object* u *is not missing in* $\mathbb{A}$, *otherwise the object* u *does not satisfy total descriptor.*

Definition 3. *Let* $\mathbb{A} = (U, A, \{d\})$ *be a decision table. Any conjunction of total descriptors* $(a_{k_1} \neq *) \wedge \ldots \wedge (a_{k_n} \neq *)$ *is called a* total template. *An object* $u \in U$ *satisfies total template* $(a_{k_1} \neq *) \wedge \ldots \wedge (a_{k_n} \neq *)$ *if the values of attributes* $a_{k_1}, \ldots, a_{k_n} \in A$ *on the object* u *are not missing in* $\mathbb{A}$.

Total templates are used to discover regular areas in data without missing values. On the basis of the total templates we can create a granule system in following way. We consider decision sub-tables $\mathbb{B} = (U_{\mathbb{B}}, B, \{d\})$ of the decision table $\mathbb{A}$, where $U_{\mathbb{B}} \subseteq U$ and $B \subseteq A$. A template t uniquely determines a granule $\mathcal{G}_t = \{\mathbb{B} = (U_{\mathbb{B}}, B, \{d\})\}$ consisting of such data tables $\mathbb{B}$ that all objects from $U_{\mathbb{B}}$ satisfies template t and all attributes $b \in B$ occur in descriptors of template t. In granule $\mathcal{G}_t$ exists the maximal decision table $\mathbb{B}_t = (U_{\mathbb{B}_t}, B_t, \{d\})$, such that for all $\mathbb{B}' = (U_{\mathbb{B}'}, B', \{d\}) \in \mathcal{G}_t$ the condition $U_{\mathbb{B}'} \subseteq U_{\mathbb{B}_t} \wedge B' \subseteq B_t$ is satisfied. Such maximal decision table has all attributes that occur in descriptors of template t and all objects from U that satisfy template t. We will use this fact later to present the data decomposition process in a formal and suitable for implementation way.

2.3 Decision Rules

Decision rules and methods for decision rule induction from decision data table without missing attribute values are well known in rough-sets (see, e.g., [10, 3]).

Definition 4. *Let* $\mathbb{A} = (U, A, \{d\})$ *be a decision table. The decision rule is a function* $\mathbb{R}: U \to V_d \cup \{?\}$, *where* $? \notin V_d$. *The decision rule consist of condition* α *and value of decision* $d^{\mathbb{R}} \in V_d$ *and can be also denoted in form of logical formula* $\alpha \Rightarrow d^{\mathbb{R}}$. *If the condition* α *is satisfied for an object* $x \in U$, *then the rule classifies* x *to the decision class* $d^{\mathbb{R}}$ ($\mathbb{R}(x) = d^{\mathbb{R}}$). *Otherwise, rule* $\mathbb{R}$ *for* x *is not applicable, which is expressed by the answer* $? \notin V_d$.

In above definition one decision rule describes a part of exactly one decision class (in mereological sense [9]). If several rules are satisfied for a given object, than voting methods have to be used to solve potential conflicts. The simplest approach assigns each rule exactly one vote. In more advanced approach the weights are assigned to decision rules to measure their strength in voting (e.g. using their quality).

Decision rule induction algorithms produce rules with conjunction of descriptors in the rule predecessor:

$$(a_{i_1}(x) \in r^{a_{i_1}} \wedge \cdots \wedge a_{i_n}(x) \in r^{a_{i_n}}) \Rightarrow d^{\mathbb{R}},$$

where $x \in U$, $a_{i_1}, \ldots, a_{i_n} \in A$, $r^{a_{i_j}} \subseteq V_{i_j}$. The classic algorithms induce decision rules with only one value in each descriptor, for example:

$$\mathbb{R}_1 : (a_1(x) = 1 \wedge a_3(x) = 4 \wedge a_7(x) = 2) \Rightarrow d^{\mathbb{R}}.$$

The D^{3}RJ method produces more general rules, where each descriptor can enclose subset of values. We call such rules the *generalized decision rules* (cf. [7, 13]).

The conditional part of a decision rule can be represented by ordered sequence of attribute value subsets $\{r^a\}_{a \in A}$ for any chosen liner order on A. For example, the decision rule $\mathbb{R}_1$, can be represented by:

$$\mathbb{R}_1 : (\{1\}, \emptyset, \{4\}, \emptyset, \emptyset, \emptyset, \{2\}) \Rightarrow d^{\mathbb{R}}.$$

The empty set denotes absence of condition for that attribute.

Definition 5. *Let* $\mathbb{A} = (U, A, \{d\})$ *be a decision table and let* $\mathbb{R}_1$, $\mathbb{R}_2$ *be generalized rules calculated from the decision table* $\mathbb{A}$. *Then we can define the distance function:*

$$\mathrm{dist}(\mathbb{R}_1, \mathbb{R}_2) = \begin{cases} \mathrm{card}(A)^2 & when \; d^{\mathbb{R}_1} \neq d^{\mathbb{R}_2} \\ \sum_{a \in A} d_a(r_1^a, r_2^a) & otherwise \end{cases}$$

where:

$$d_a(X, Y) = \begin{cases} \mathrm{card}(A) & when \; X = \emptyset \wedge Y \neq \emptyset \; or \; X \neq \emptyset \wedge Y = \emptyset \\ \frac{\mathrm{card}((X-Y) \cup (Y-X))}{\mathrm{card}(V_i)} & otherwise \end{cases}.$$

The presented above distance function is used in D^{3}RJ method for comparison of decision rules logical structure and for estimation of their similarity.

3   D^{3}RJ

The D^{3}RJ method is developed in the frameworks of Granular Computing and Rough-Mereology [9]. The processing consists of four phases called the data decomposition, decision rule induction, decision rule shortening and decision rule joining.

In the first phase the data that describe the whole investigated phenomenon is decomposed — partitioned into a number of subsets that describe, in a sense, parts of investigated phenomenon. Such procedure creates an overlapped, but non-exhaustive covering that consists of elements similar to the covered data. These elements are data subsets and parts in the mereological sense of the whole, i.e., the original data. The data decomposition method is aiming to avoid the problem of reasoning from data with incomplete object descriptions.

In the second phase information contained in parts, i.e., data subsets is transformed using inductive learning, to a set of decision rules. As it is explained later, we can apply any method of decision rule induction, including such ones that cannot deal with missing values. Often methods that make it possible to properly induce decision rules from data with missing values lead to inefficient algorithms or algorithms with low quality of classification. With help of a data decomposition all of them are applied to a data without missing values to take an advantage of their lower computational complexity and more precise decision rules.

Third phase is the rule shortening. It is very useful because it reduces complexity of rule set and improves classifier resistance to noise and data disturbations.

In the fourth phase the set of rule granules is converted to the smaller set of simplified but more powerful representation of decision rules. In this phase decision rules are clustered and joined to a coherent classifier. The constructed generalized rules can be treated as the higher level granules that represent knowledge extracted from several decision rules. The main objectives of the decision rule joining are reduction of classifier complexity and simplification of knowledge representation.

The D³RJ method returns a classifier that can be applied to a data with missing attribute values in both, learning and classifying.

4   Data Decomposition

The data decomposition should be done in accordance to regularities in a real-world interest domain. We expect the decomposition to reveal patterns of missing attribute values with a similar meaning for the investigated real-world problem.

With the help of introduced concept of total template it is possible to express the goal of the data decomposition phase in terms of total templates. The finite set of templates $S = \{t_1, \ldots, t_n\}$ determines uniquely a finite decomposition $D = \{B_{t_1}, \ldots, B_{t_n}\}$ of the decision table $\mathbb{A}$, where $B_{t_i} \in \mathcal{G}_{t_i}$ is a maximal decision sub-table related to template t_i. Such a transformation reduces problem of data decomposition to the problem of covering decision table with templates, which is frequently investigated (see e.g. [7, 8]) and we can use an advantage of broad experience in this area. In almost all approaches a greedy heuristic to this problem is employed. The greedy algorithm generates the best template for a decision table with respect to a defined criterion and removes all objects that are covered by generated template. In subsequent iterations the decision table is reduced in size by objects that are already covered and the generation of the next best template is repeated. The algorithm continues until all objects are covered, i.e., all objects were removed in previous iterations.

The problem of selecting the optimal criterion for defining the best template with respect to the data decomposition with total templates was already investigated (cf. [4]). There were proposed some template evaluation functions that perform efficiently at reasonable computational cost. The simplest template

evaluation function is based only on size of maximal decision table related to this total template.

$$q_1(t) = w(t) \cdot h(t) \tag{1}$$

The $w(t)$ in above formula is the number of attributes in maximal decision table related to template t (width), thus the number of attributes that appear in descriptors of template t. The $h(t)$ is the number of objects in maximal decision table related to template t (height). A more complex template evaluation function incorporates also a heuristical template evaluation:

$$q_2(t) = w(t) \cdot h(t) \cdot G(t)^{\beta}, \tag{2}$$

where parameter β controls the importance factor of an rough-set heuristical evaluation function and

$$G(t) = \sum_{i=1}^{K} \frac{\max_{c \in V_d} \mathrm{card}(\{y \in [x^i]_{\mathrm{IND}_t} : d(y) = c\})}{\mathrm{card}([x^i]_{\mathrm{IND}_t})}. \tag{3}$$

In above formula K is the number of indiscernibility classes (classes of abstraction of the indiscernibility relation IND_t) and $[x^i]_{\mathrm{IND}_t}$ denotes the i-th indiscernibility class. The indiscernibility relation IND_t is based on the maximal complete decision sub-table related to the template t.

5   Decision Rule Induction

The data decomposition phase delivers a number of data tables free from missing values. Such data tables enable us to apply any classifier induction method. In particular, the methods for inducing decision rules, that frequently suffer from lack of possibility to induce rules from data with missing values can be used. On each data table returned from the decomposition phase we apply an algorithm for decision rule induction.

In D^3RJ we use a method inducing all possible consistent decision rules, called also optimal decision rules. This method induces decision rules based on indiscernibility matrix (see e.g. [3, 12, 11]).

When the rules are too detailed, they are difficult to join and joining problem become to very time consuming. We can overcome this problem using shortening algorithm for decision rules (see, e.g., [6, 14]).

In the obtained set of all decision rules one object is usually covered by many rules. In D^3RJ method the problem of redundant decision rules is solved by rule joining. Such a proceeding delivers even less decision rules than without using the data decomposition and provide stable and accurate classifier with a compact concept description.

6   Decision Rule Joining

In the decision rule joining we allow to join only rules from the same decision class. It is possible to join two rules that have different decisions but it will make our system more complicated.

First, let us consider an example of simple rule joining. Rules to be joined:

$$\mathbb{R}_1 : (\{1\}, \{3\}, \emptyset, \{1\}, \{2\}, \emptyset, \{2\}) \Rightarrow d,$$
$$\mathbb{R}_2 : (\{2\}, \{3\}, \emptyset, \{2\}, \{2\}, \emptyset, \{3\}) \Rightarrow d.$$

After joining we obtain a generalized decision rule:

$$\mathbb{R} : (\{1,2\}, \{3\}, \{\emptyset\}, \{1,2\}, \{2\}, \{\emptyset\}, \{2,3\}) \Rightarrow d.$$

Moreover, we can join the generalized rules exactly in the same way as classical ones. Formally speaking a new rule obtained from $\mathbb{R}_m$ and $\mathbb{R}_n$ have a form $\left\{ r^a_{\mathbb{R}_m + \mathbb{R}_n} \right\}_{a \in A} \Rightarrow d$, where $r^a_{\mathbb{R}_m + \mathbb{R}_n} := r^a_m \cup r^a_n$. The D^{3}RJ method utilizes a decision rule joining algorithm as described in following points.

1. Let $X^{\mathbb{R}}$ be a set of all induced rules. We can assume that it is a set of generalized rules, because every classical rule can be interpreted as a generalized rule.
2. Let $\mathbb{R}_m \in X^{\mathbb{R}}$ and $\mathbb{R}_n \in X^{\mathbb{R}}$ be such, that $d^{\mathbb{R}_m} = d^{\mathbb{R}_n}$ and

$$\text{dist}(\mathbb{R}_m, \mathbb{R}_n) = \min_{i,j} \{ \text{dist}(\mathbb{R}_i, \mathbb{R}_j) : \mathbb{R}_i, \mathbb{R}_j \in X^{\mathbb{R}} \wedge d^{\mathbb{R}_i} = d^{\mathbb{R}_j} \}.$$

3. If there exist $\mathbb{R}_m$ and $\mathbb{R}_n$ in $X^{\mathbb{R}}$ such that $\text{dist}(\mathbb{R}_m, \mathbb{R}_n) < \varepsilon$ then the set of rules $X^{\mathbb{R}}$ is modified as follows:

$$X^{\mathbb{R}} := X^{\mathbb{R}} - \{\mathbb{R}_m, \mathbb{R}_n\},$$

$$X^{\mathbb{R}} := X^{\mathbb{R}} \cup \{\mathbb{R}_{\mathbb{R}_m + \mathbb{R}_n}\},$$

   where $\mathbb{R}_{\mathbb{R}_m + \mathbb{R}_n}$ is a new rule obtained by joining $\mathbb{R}_m$ and $\mathbb{R}_n$.
4. If the set $X^{\mathbb{R}}$ has been changed then we go back to step 2, otherwise the algorithm is finished.

We can assume that, for example, $\varepsilon = 1$. The algorithm ends when in the set $X^{\mathbb{R}}$ are no two rules from the same decision class that are close enough.

Presented method called Linear Rule Joining (LRJ) is very simple and efficient in time.

7   Empirical Evaluation

There were carried out some experiments in order to evaluate the D^{3}RJ method. Results were obtained using the ten-fold Cross-Validation (CV10) evaluation. The experiments were performed with different decomposition approaches as well as without using decomposition method at all. In experiments we used following data tables: att, ech, edu, hco, head, hep, hur2, hyp, inf2, pid2 and smo2. All data tables used in evaluation of the decomposition method were taken from *Recursive-Partitioning.com* [5] and contain from 10.5% up to 100% cases with missing values of natural origin. In presented results the exhaustive rule induction method was used to induce classifiers from the decision subtables. This

Table 1. Classification accuracy of the classical exhaustive decision rule induction and the D^{3}RJ method. Results are averaged using ten fold cross-validation over 11 data tables used in experiments and α is the rule shortening factor.

α	No decomposition	$w \cdot h$	$w{\cdot}h{\cdot}G$	$w{\cdot}h{\cdot}G^8$
1.0	70.15%	69.60%	70.57%	69.57%
0.9	71.66%	69.83%	70.42%	70.20%
0.8	71.36%	69.74%	70.13%	69.45%
0.7	71.87%	68.44%	69.57%	70.12%
0.6	69.72%	67.78%	69.07%	68.91%
0.5	67.93%	68.32%	69.39%	69.38%
0.4	66.81%	67.88%	68.60%	68.62%
0.3	68.28%	68.30%	68.78%	68.83%
0.2	66.47%	69.31%	69.46%	69.79%
0.1	66.14%	69.77%	69.15%	68.97%

Table 2. Number of decision rules using the classical exhaustive decision rule induction and the D^{3}RJ method.

α	No decomposition	$w \cdot h$	$w{\cdot}h{\cdot}G$	$w{\cdot}h{\cdot}G^8$
1.0	9970.54	1101.51	979.35	841.60
0.9	8835.55	1029.09	912.40	795.65
0.8	6672.00	893.29	798.13	701.93
0.7	4945.65	759.05	679.10	601.71
0.6	3114.22	564.05	510.35	460.64
0.5	1682.63	372.40	377.78	369.90
0.4	1158.45	253.72	246.65	236.31
0.3	661.78	146.45	147.34	144.81
0.2	366.80	75.36	84.64	87.41
0.1	227.59	40.12	45.45	45.11

method is implemented in the *RSES-Lib* software (see [2]). The data decomposition was done with the help of a genetic algorithm for best template generation.

Table 1 presents a general comparison of the classification accuracy using the exhaustive decision rule induction with the D^{3}RJ method using various decomposition criteria and shortening factor values. Table contains the classification accuracy averaged over 11 tested data sets. In the Table 2 the similar comparison is presented with respect to the number of decision rules. The detailed results are presented in next tables. From averages presented in Table 1 one can see that in general the classification accuracy of the D^{3}RJ method is slightly worse than standard decision rules at the top of the table, but slightly better at the bottom of it, where the shortening factor is lower. Table 2 that present number of decision rules, shows that the D^{3}RJ method requires 5–10 times less decision rules than the standard exhaustive decision rules, called also optimal decision rule. Thus, the reduction of the classification abilities is not as high as the reduction of the model size.

Table 3. The detailed empirical evaluation of the D^{3}RJ method using the shortening factor $\alpha = 1.00$, and template evaluation function $q = w{\cdot}h{\cdot}G$.

Table	Before joining		After joining		Profit	
	Accuracy	# Rules	Accuracy	# Rules	Com-pres-sion	Imp-rove-ment
att	56.09 ±3.23	2924.0 ±698.65	53.39 ±3.24	408.0 ±9.35	5.58	-2.70
ech	61.89 ±8.24	317.3 ±56.14	63.61 ±9.55	45.4 ±6.30	3.46	1.72
edu	46.22 ±5.28	4146.0 ±52.83	53.22 ±3.93	716.3 ±14.51	5.70	7.00
hco	79.90 ±8.25	2313.7 ±702.85	78.79 ±5.39	496.5 ±63.07	4.28	-1.11
head	64.20 ±3.59	703.4 ±18.29	67.31 ±6.06	224.2 ±7.30	2.63	3.11
hep	82.40 ±5.59	2496.1 ±148.23	80.08 ±6.16	579.7 ±154.07	1.31	-2.32
hur2	79.57 ±7.76	322.3 ±59.52	78.02 ±7.56	19.8 ±4.47	3.90	-1.55
hyp	97.63 ±1.06	1381.8 ±113.04	97.82 ±0.96	166.1 ±10.79	3.39	0.19
inf2	64.10 ±9.98	5003.7 ±237.77	65.24 ±9.74	4769.8 ±481.91	1.02	1.14
pid2	70.71 ±4.64	4056.0 ±226.20	72.39 ±2.35	109.3 ±5.18	17.87	1.68
smo2	53.03 ±2.17	8252.7 ±51.57	66.41 ±2.55	1217.1 ±17.11	6.49	13.38
avg	68.70 ±5.44	2901.55 ±215.01	70.57 ±5.23	979.35 ±67.96	2.96	1.87

Table 3 presents detailed experimental results of D^{3}RJ method with use of template evaluation function $q = w{\cdot}h{\cdot}G$ and shortening factor α in range $[0.1, 1.0]$. The results are presented for the standard decomposition method without the decision rule joining as well as with the decision rule joining. The decomposition method without the decision rule joining uses the standard voting over all decision rules induced from sub-tables. The compression ratio presented in this table is the ratio of the number of decision rules without the decision rule joining to the number of decision rules with the decision rule joining. The improvement is the difference of the classification accuracy between classification without and with decision rule joining. As we can see the decision rule joining not only reduces the number of decision rules, but also improves the classification accuracy.

8   Conclusions

The presented method consists of two main steps. The first one, called the decomposition step, makes it possible to split decision table with missing attribute values into more tables without missing values. In the second step one classifier (decision system) is induced from decision tables returned from the first step by joining some smaller subsystems of decision rules.

In the consequence we obtained a simple strategy for building decision systems for data tables with missing attribute values. Moreover, we can use in this task a parallel computing because created subsystems are independent. It seems that in this way it is possible to solve many hard classification problems in relatively short time. The obtained results showed that the presented method is very promising for classification problems with missing attribute values in data sets.

Acknowledgments

References

1. Alpigini, J.J., Peters, J.F., Skowron, A., Zhong, N., eds.: Rough Sets and Current Trends in Computing, Third International Conference, RSCTC 2002, Malvern, PA, USA, October 14–16, 2002, Proceedings. LNCS 2475, Springer (2002)
2. Bazan, J.G., Szczuka, M.S., Wróblewski, J.: A new version of rough set exploration system. [1] 397–404
3. Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A.: Rough sets: A tutorial. In Pal, S.K., Skowron, A., eds.: Rough Fuzzy Hybridization. A New Trend in Decision Making, Singapore, Springer (1999) 3–98
4. Latkowski, R.: On decomposition for incomplete data. Fundamenta Informaticae **54** (2003) 1–16
5. Lim, T.: Missing covariate values and classification trees. http://www.recursive-partitioning.com/mv.shtml, Recursive-Partitioning.com (2000)
6. Møllestad, T., Skowron, A.: A rough set framework for data mining of propositional default rules. In Raś, Z.W., Michalewicz, M., eds.: Foundations of Intelligent Systems — ISMIS 1996. LNCS 1079, Springer (1996) 448–457
7. Nguyen, S.H.: Regularity Analysis and its Application in Data Mining. PhD thesis, Warsaw University, Faculty of Mathematics, Computer Science and Mechanics (1999)
8. Nguyen, S.H., Skowron, A., Synak, P.: Discovery of data patterns with applications to decomposition and classification problems. In Polkowski, L., Skowron, A., eds.: Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems, Physica-Verlag (1998) 55–97
9. Pal, S.K., Polkowski, L., Skowron, A., eds.: Rough-Neural Computing: Techniques for Computing with Words. Springer (2004)
10. Pawlak, Z.: Rough sets: Theoretical aspects of reasoning about data. Kluwer, Dordrecht (1991)
11. Skowron, A.: Boolean reasoning for decision rules generation. In Komorowski, H.J., Raś, Z.W., eds.: Methodologies for Intelligent Systems — ISMIS 1993. LNCS 689, Springer (1993) 295–305
12. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In Słowiński, R., ed.: Intelligent Decision Support. Handbook of Applications and Advances in Rough Sets Theory, Dordrecht, Kluwer (1992) 331–362
13. Wang, H., Düntsh, I., Gediga, G., Skowron, A.: Hyperrelations in version space. Journal of Approximate Reasoning (to appear, 2004)
14. Ziarko, W.: Variable precision rough sets model. Journal of Computer and System Sciences **46** (1993) 39–59